```
<ht
ml>
    <head>
    <title>Pongo Bahasa (Construct
    2 preview)</title>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-
    Compatible"
    content="IE=edge,chrome=1" />
    <meta name="viewport"
    content="width=device-width,
    initial-scale=1.0, maximum-
    scale=1.0, minimum-scale=1.0,
    user-scalable=no, minimal-ui"
    />
    <meta name="apple-mobile-web-
    app-capable" content="yes" />
    <meta name="apple-mobile-web-
    app-status-bar-style"
    content="black" />
    <meta name="HandheldFriendly"
    content="true" />
    <style type="text/css">
    * {
    padding: 0;
    margin: 0;
    }
    body {
    background: #000;
    color: #fff;
    overflow: hidden;
    }
    canvas {
    touch-action-delay: none;
    touch-action: none;
    -ms-touch-action: none;
    }
    #inspect-outline {
    position: absolute;
    border: 2px dotted red;
    overflow: hidden;
    font-size: 8pt;
    font-family: Sans serif;
    color: #444;
    pointer-events: none;
    -webkit-animation: flash 2s
    linear 0s infinite;
    -moz-animation: flash 2s
    linear 0s infinite;
    -o-animation: flash 2s linear
    0s infinite;
    animation: flash 2s linear 0s
    infinite;
    background-image:
    url(debugger-inspect.png);
    }
    @-webkit-keyframes flash {
    0%, 75%, 100% {opacity: 1;}
    87% {opacity: 0.25;}
    }
    @-moz-keyframes flash {
    0%, 75%, 100% {opacity: 1;}
    87% {opacity: 0.25;}
    }
    @-o-keyframes flash {
    0%, 75%, 100% {opacity: 1;}
    87% {opacity: 0.25;}
    }
    @keyframes flash {
    0%, 75%, 100% {opacity: 1;}
    87% {opacity: 0.25;}
    }
    </style>
    <script type="text/javascript"
    src="jquery-
    2.0.0.min.js"></script>
    <script type="text/javascript"
    src="common_prelude.js"></scri
    pt>
    <script type="text/javascript"
    src="preview_prelude.js"></scr
    ipt>
    <script type="text/javascript"
    src="shaders.js"></script>
    <script type="text/javascript"
    src="glwrap.js"></script>
    <script
    src="Audio_common.js"></script
    >
    <script
    src="Audio_plugin.js"></script
    >
    <script
    src="Sprite_common.js"></scrip
    t>
    <script
    src="Sprite_plugin.js"></scrip
    t>
    <script
    src="Text_common.js"></script>
    <script
    src="Text_plugin.js"></script>
```

```
<script
src="TextBox_common.js"></scri
pt>
<script
src="TextBox_plugin.js"></scri
pt>
<script
src="Touch_common.js"></script
>
<script
src="Touch_plugin.js"></script
>
<script
src="DragnDrop_common.js"></sc
ript>
<script
src="DragnDrop_behavior.js"></
script>
<script
src="Flash_common.js"></script
>
<script
src="Flash_behavior.js"></scri
pt>
<script type="text/javascript"
src="data.js"></script>
<script type="text/javascript"
src="preview.js"></script>
<script type="text/javascript"
src="layout.js"></script>
<script type="text/javascript"
src="eveng.js"></script>
<script type="text/javascript"
src="expressions.js"></script>
<script type="text/javascript"
src="system.js"></script>
<script type="text/javascript"
src="commonace.js"></script>
<script
type="text/javascript">
jQuery(window).resize(function
() {
cr_sizeCanvas(jQuery(window).w
idth(),
jQuery(window).height());
});
// Start the project running
on window load
jQuery(document).ready(functio
n ()
{
// Create new runtime using
the c2canvas
cr.createRuntime("c2canvas");
});
// Pause and resume on page
becoming visible/invisible
function onVisibilityChanged()
{
if (document.hidden ||
document.mozHidden ||
document.webkitHidden ||
document.msHidden)
cr_setSuspended(true);
else
cr_setSuspended(false);
};
document.addEventListener("vis
ibilitychange",
onVisibilityChanged, false);
document.addEventListener("moz
visibilitychange",
onVisibilityChanged, false);
document.addEventListener("web
kitvisibilitychange",
onVisibilityChanged, false);
document.addEventListener("msv
isibilitychange",
onVisibilityChanged, false);
</script>
</head>
<body>
<div id="fb-root"></div>
<!-- The canvas must be inside
a div called c2canvasdiv -->
<div id="c2canvasdiv">
<canvas id="c2canvas"
width="1280" height="768">
<div
id="canvasNotSupportedBox">
<h1>This browser does not
appear to support HTML5. Try
upgrading your browser to the
latest version. <a
href="http://www.whatbrowser.o
rg">What is a browser?</a>
<br/><br/><a
href="http://www.microsoft.com
/windows/internet-
explorer/default.aspx">Microso
ft Internet Explorer</a><br/>
<a
href="http://www.mozilla.com/f
irefox/">Mozilla
Firefox</a><br/>
<a
href="http://www.google.com/ch
```

```
rome/">Google Chrome</a><br/>
<a
href="http://www.apple.com/saf
ari/download/">Apple
Safari</a><br/>
<a
href="http://www.google.com/ch
romeframe">Google Chrome Frame
for Internet Explorer</a><br/>
</h1>
</div>
</canvas>
</div>
</body>
</html>
```

```
<!DOCTYPE html>
<html
manifest="offline.appcache">
<head>
    <meta charset="UTF-8" />
     <meta http-equiv="X-UA-
Compatible"
content="IE=edge,chrome=1" />
     <title>Pongo
Bahasa</title>

     <!-- Allow fullscreen mode
on iOS devices. (These are
Apple specific meta tags.) -->
     <meta name="viewport"
content="width=device-width,
initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0,
user-scalable=no, minimal-ui"
/>
     <meta name="apple-mobile-
web-app-capable" content="yes"
/>
     <meta name="apple-mobile-
web-app-status-bar-style"
content="black" />
     <link rel="apple-touch-
icon" sizes="256x256"
href="icon-256.png" />
     <meta
name="HandheldFriendly"
content="true" />

     <!-- Chrome for Android
web app tags -->
     <meta name="mobile-web-
app-capable" content="yes" />
     <link rel="shortcut icon"
sizes="256x256" href="icon-
256.png" />

    <!-- All margins and
padding must be zero for the
canvas to fill the screen. -->
     <style type="text/css">
          * {
               padding: 0;
               margin: 0;
          }
          body {
               background:
#000;
               color: #fff;
               overflow:
hidden;
               -ms-touch-
action: none;
          }
          canvas {
               touch-action-
delay: none;
               touch-action:
none;
               -ms-touch-
action: none;
          }
    </style>


</head>

<body>
     <div id="fb-root"></div>

     <script>
     // Issue a warning if
trying to preview an exported
project on disk.
     (function(){
          // Check for running
exported on file protocol
          if
(window.location.protocol.subst
r(0, 4) === "file")
          {
               alert("Exported
games won't work until you
upload them. (When running on
the file:/// protocol, browsers
block many features from
working for security
reasons.)");
          }
     })();
     </script>

     <!-- The canvas must be
inside a div called c2canvasdiv
-->
     <div id="c2canvasdiv">

          <!-- The canvas the
project will render to.  If you
change its ID, don't forget to
change the
          ID the runtime looks
for in the jQuery events above
(ready() and cr_sizeCanvas()).
-->
```

```
            <canvas
id="c2canvas" width="1280"
height="768">
                <!-- This text
is displayed if the visitor's
browser does not support HTML5.
                You can change
it, but it is a good idea to
link to a description of a
browser
                and provide
some links to download some
popular HTML5-compatible
browsers. -->
                <h1>Your
browser does not appear to
support HTML5.  Try upgrading
your browser to the latest
version.  <a
href="http://www.whatbrowser.or
g">What is a browser?</a>
                <br/><br/><a
href="http://www.microsoft.com/
windows/internet-
explorer/default.aspx">Microsof
t Internet Explorer</a><br/>
                <a
href="http://www.mozilla.com/fi
refox/">Mozilla
Firefox</a><br/>
                <a
href="http://www.google.com/chr
ome/">Google Chrome</a><br/>
                <a
href="http://www.apple.com/safa
ri/download/">Apple
Safari</a><br/>
                <a
href="http://www.google.com/chr
omeframe">Google Chrome Frame
for Internet
Explorer</a><br/></h1>
            </canvas>

      </div>

      <!-- Pages load faster
with scripts at the bottom -->

      <!-- Construct 2 exported
games require jQuery. -->
      <script src="jquery-
2.0.0.min.js"></script>

    <!-- The runtime script.
You can rename it, but don't
forget to rename the reference
here as well.
    This file will have been
minified and obfuscated if you
enabled "Minify script" during
export. -->
      <script
src="c2runtime.js"></script>

    <script>
            // Size the canvas
to fill the browser viewport.

      jQuery(window).resize(func
tion() {

      cr_sizeCanvas(jQuery(windo
w).width(),
jQuery(window).height());
            });

            // Start the
Construct 2 project running on
window load.

      jQuery(document).ready(fun
ction ()
            {
                  // Create new
runtime using the c2canvas

      cr_createRuntime("c2canvas
");
            });

            // Pause and resume
on page becoming
visible/invisible
            function
onVisibilityChanged() {
                  if
(document.hidden ||
document.mozHidden ||
document.webkitHidden ||
document.msHidden)

      cr_setSuspended(true);
                  else

      cr_setSuspended(false);
            };


      document.addEventListener(
```

```
"visibilitychange",
onVisibilityChanged, false);

      document.addEventListener(
"mozvisibilitychange",
onVisibilityChanged, false);

      document.addEventListener(
"webkitvisibilitychange",
onVisibilityChanged, false);

      document.addEventListener(
"msvisibilitychange",
onVisibilityChanged, false);

    </script>
</body>
</html>
```

```
<!-- -->
var cr = {};
cr.plugins_ = {};
cr.behaviors = {};
if (typeof
Object.getPrototypeOf !==
"function")
{
	if (typeof
"test".__proto__ === "object")
	{

	Object.getPrototypeOf =
function(object) {
			return
object.__proto__;
		};
	}
	else
	{

	Object.getPrototypeOf =
function(object) {
			return
object.constructor.prototype;
		};
	}
}
(function(){
	cr.logexport = function
(msg)
	{
		if (window.console
&& window.console.log)

	window.console.log(msg);
	};
	cr.seal = function(x)
	{
		return x;
	};
	cr.freeze = function(x)
	{
		return x;
	};
	cr.is_undefined = function
(x)
	{
		return typeof x ===
"undefined";
	};
	cr.is_number = function
(x)
	{
		return typeof x ===
"number";
	};
	cr.is_string = function
(x)
	{
		return typeof x ===
"string";
	};
	cr.isPOT = function (x)
	{
		return x > 0 && ((x
- 1) & x) === 0;
	};
	cr.nextHighestPowerOfTwo =
function(x) {
		--x;
		for (var i = 1; i <
32; i <<= 1) {
			x = x | x >> i;
		}
		return x + 1;
	}
	cr.abs = function (x)
	{
		return (x < 0 ? -x :
x);
	};
	cr.max = function (a, b)
	{
		return (a > b ? a :
b);
	};
	cr.min = function (a, b)
	{
		return (a < b ? a :
b);
	};
	cr.PI = Math.PI;
	cr.round = function (x)
	{
		return (x + 0.5) |
0;
	};
	cr.floor = function (x)
	{
		if (x >= 0)
			return x | 0;
		else
			return (x | 0)
- 1;	// correctly round
down when negative
	};
	cr.ceil = function (x)
	{
		var f = x | 0;
		return (f === x ? f
: f + 1);
```

```
	};
	function Vector2(x, y)
	{
		this.x = x;
		this.y = y;
		cr.seal(this);
	};
	Vector2.prototype.offset =
function (px, py)
	{
		this.x += px;
		this.y += py;
		return this;
	};
	Vector2.prototype.mul =
function (px, py)
	{
		this.x *= px;
		this.y *= py;
		return this;
	};
	cr.vector2 = Vector2;
	cr.segments_intersect =
function(a1x, a1y, a2x, a2y,
b1x, b1y, b2x, b2y)
	{
		var max_ax, min_ax,
max_ay, min_ay, max_bx, min_bx,
max_by, min_by;
		if (a1x < a2x)
		{
			min_ax = a1x;
			max_ax = a2x;
		}
		else
		{
			min_ax = a2x;
			max_ax = a1x;
		}
		if (b1x < b2x)
		{
			min_bx = b1x;
			max_bx = b2x;
		}
		else
		{
			min_bx = b2x;
			max_bx = b1x;
		}
		if (max_ax < min_bx
|| min_ax > max_bx)
			return false;
		if (a1y < a2y)
		{
			min_ay = a1y;
			max_ay = a2y;
		}
		else
		{
			min_ay = a2y;
			max_ay = a1y;
		}
		if (b1y < b2y)
		{
			min_by = b1y;
			max_by = b2y;
		}
		else
		{
			min_by = b2y;
			max_by = b1y;
		}
		if (max_ay < min_by
|| min_ay > max_by)
			return false;
		var dpx = b1x - a1x
+ b2x - a2x;
		var dpy = b1y - a1y
+ b2y - a2y;
		var qax = a2x - a1x;
		var qay = a2y - a1y;
		var qbx = b2x - b1x;
		var qby = b2y - b1y;
		var d = cr.abs(qay *
qbx - qby * qax);
		var la = qbx * dpy -
qby * dpx;
		if (cr.abs(la) > d)
			return false;
		var lb = qax * dpy -
qay * dpx;
		return cr.abs(lb) <=
d;
	};
	function Rect(left, top,
right, bottom)
	{
		this.set(left, top,
right, bottom);
		cr.seal(this);
	};
	Rect.prototype.set =
function (left, top, right,
bottom)
	{
		this.left = left;
		this.top = top;
		this.right = right;
		this.bottom =
bottom;
	};
```

```
	Rect.prototype.copy =
function (r)
	{
		this.left = r.left;
		this.top = r.top;
		this.right =
r.right;
		this.bottom =
r.bottom;
	};
	Rect.prototype.width =
function ()
	{
		return this.right -
this.left;
	};
	Rect.prototype.height =
function ()
	{
		return this.bottom -
this.top;
	};
	Rect.prototype.offset =
function (px, py)
	{
		this.left += px;
		this.top += py;
		this.right += px;
		this.bottom += py;
		return this;
	};
	Rect.prototype.normalize =
function ()
	{
		var temp = 0;
		if (this.left >
this.right)
		{
			temp =
this.left;
			this.left =
this.right;
			this.right =
temp;
		}
		if (this.top >
this.bottom)
		{
			temp =
this.top;
			this.top =
this.bottom;
			this.bottom =
temp;
		}
	};
	Rect.prototype.intersects_
rect = function (rc)
	{
		return !(rc.right <
this.left || rc.bottom <
this.top || rc.left >
this.right || rc.top >
this.bottom);
	};
	Rect.prototype.intersects_
rect_off = function (rc, ox,
oy)
	{
		return !(rc.right +
ox < this.left || rc.bottom +
oy < this.top || rc.left + ox >
this.right || rc.top + oy >
this.bottom);
	};
	Rect.prototype.contains_pt
= function (x, y)
	{
		return (x >=
this.left && x <= this.right)
&& (y >= this.top && y <=
this.bottom);
	};
	Rect.prototype.equals =
function (r)
	{
		return this.left ===
r.left && this.top === r.top &&
this.right === r.right &&
this.bottom === r.bottom;
	};
	cr.rect = Rect;
	function Quad()
	{
		this.tlx = 0;
		this.tly = 0;
		this.trx = 0;
		this.try_ = 0;	//
is a keyword otherwise!
		this.brx = 0;
		this.bry = 0;
		this.blx = 0;
		this.bly = 0;
		cr.seal(this);
	};
	Quad.prototype.set_from_re
ct = function (rc)
	{
		this.tlx = rc.left;
		this.tly = rc.top;
		this.trx = rc.right;
		this.try_ = rc.top;
```

```
            this.brx = rc.right;
            this.bry =
rc.bottom;
            this.blx = rc.left;
            this.bly =
rc.bottom;
      };
      Quad.prototype.set_from_ro
tated_rect = function (rc, a)
      {
            if (a === 0)
            {

      this.set_from_rect(rc);
            }
            else
            {
                  var sin_a =
Math.sin(a);
                  var cos_a =
Math.cos(a);
                  var left_sin_a
= rc.left * sin_a;
                  var top_sin_a =
rc.top * sin_a;
                  var right_sin_a
= rc.right * sin_a;
                  var
bottom_sin_a = rc.bottom *
sin_a;
                  var left_cos_a
= rc.left * cos_a;
                  var top_cos_a =
rc.top * cos_a;
                  var right_cos_a
= rc.right * cos_a;
                  var
bottom_cos_a = rc.bottom *
cos_a;
                  this.tlx =
left_cos_a - top_sin_a;
                  this.tly =
top_cos_a + left_sin_a;
                  this.trx =
right_cos_a - top_sin_a;
                  this.try_ =
top_cos_a + right_sin_a;
                  this.brx =
right_cos_a - bottom_sin_a;
                  this.bry =
bottom_cos_a + right_sin_a;
                  this.blx =
left_cos_a - bottom_sin_a;
                  this.bly =
bottom_cos_a + left_sin_a;
            }
      };
      Quad.prototype.offset =
function (px, py)
      {
            this.tlx += px;
            this.tly += py;
            this.trx += px;
            this.try_ += py;
            this.brx += px;
            this.bry += py;
            this.blx += px;
            this.bly += py;
            return this;
      };
      var minresult = 0;
      var maxresult = 0;
      function minmax4(a, b, c,
d)
      {
            if (a < b)
            {
                  if (c < d)
                  {
                        if (a <
c)
      minresult = a;
                        else
      minresult = c;
                        if (b >
d)
      maxresult = b;
                        else
      maxresult = d;
                  }
                  else
                  {
                        if (a <
d)
      minresult = a;
                        else
      minresult = d;
                        if (b >
c)
      maxresult = b;
                        else
      maxresult = c;
                  }
            }
```

```
            else
            {
                  if (c < d)
                  {
                        if (b <
c)
      minresult = b;
                        else
      minresult = c;
                        if (a >
d)
      maxresult = a;
                        else
      maxresult = d;
                  }
                  else
                  {
                        if (b <
d)
      minresult = b;
                        else
      minresult = d;
                        if (a >
c)
      maxresult = a;
                        else
      maxresult = c;
                  }
            }
      };
      Quad.prototype.bounding_bo
x = function (rc)
      {
            minmax4(this.tlx,
this.trx, this.brx, this.blx);
            rc.left = minresult;
            rc.right =
maxresult;
            minmax4(this.tly,
this.try_, this.bry, this.bly);
            rc.top = minresult;
            rc.bottom =
maxresult;
      };
      Quad.prototype.contains_pt
= function (x, y)
      {
            var v0x = this.trx -
this.tlx;
            var v0y = this.try_
- this.tly;
            var v1x = this.brx -
this.tlx;
            var v1y = this.bry -
this.tly;
            var v2x = x -
this.tlx;
            var v2y = y -
this.tly;
            var dot00 = v0x *
v0x + v0y * v0y
            var dot01 = v0x *
v1x + v0y * v1y
            var dot02 = v0x *
v2x + v0y * v2y
            var dot11 = v1x *
v1x + v1y * v1y
            var dot12 = v1x *
v2x + v1y * v2y
            var invDenom = 1.0 /
(dot00 * dot11 - dot01 *
dot01);
            var u = (dot11 *
dot02 - dot01 * dot12) *
invDenom;
            var v = (dot00 *
dot12 - dot01 * dot02) *
invDenom;
            if ((u >= 0.0) && (v
> 0.0) && (u + v < 1))
                  return true;
            v0x = this.blx -
this.tlx;
            v0y = this.bly -
this.tly;
            var dot00 = v0x *
v0x + v0y * v0y
            var dot01 = v0x *
v1x + v0y * v1y
            var dot02 = v0x *
v2x + v0y * v2y
            invDenom = 1.0 /
(dot00 * dot11 - dot01 *
dot01);
            u = (dot11 * dot02 -
dot01 * dot12) * invDenom;
            v = (dot00 * dot12 -
dot01 * dot02) * invDenom;
            return (u >= 0.0) &&
(v > 0.0) && (u + v < 1);
      };
      Quad.prototype.at =
function (i, xory)
```

```
	{
		if (xory)
		{
			switch (i)
			{
				case 0:
return this.tlx;
				case 1:
return this.trx;
				case 2:
return this.brx;
				case 3:
return this.blx;
				case 4:
return this.tlx;
				default:
return this.tlx;
			}
		}
		else
		{
			switch (i)
			{
				case 0:
return this.tly;
				case 1:
return this.try_;
				case 2:
return this.bry;
				case 3:
return this.bly;
				case 4:
return this.tly;
				default:
return this.tly;
			}
		}
	};
	Quad.prototype.midX =
function ()
	{
		return (this.tlx +
this.trx  + this.brx +
this.blx) / 4;
	};
	Quad.prototype.midY =
function ()
	{
		return (this.tly +
this.try_ + this.bry +
this.bly) / 4;
	};
	Quad.prototype.intersects_
segment = function (x1, y1, x2,
y2)
	{
		if
(this.contains_pt(x1, y1) ||
this.contains_pt(x2, y2))
			return true;
		var a1x, a1y, a2x,
a2y;
		var i;
		for (i = 0; i < 4;
i++)
		{
			a1x =
this.at(i, true);
			a1y =
this.at(i, false);
			a2x = this.at(i
+ 1, true);
			a2y = this.at(i
+ 1, false);
			if
(cr.segments_intersect(x1, y1,
x2, y2, a1x, a1y, a2x, a2y))
				return
true;
		}
		return false;
	};
	Quad.prototype.intersects_
quad = function (rhs)
	{
		var midx =
rhs.midX();
		var midy =
rhs.midY();
		if
(this.contains_pt(midx, midy))
			return true;
		midx = this.midX();
		midy = this.midY();
		if
(rhs.contains_pt(midx, midy))
			return true;
		var a1x, a1y, a2x,
a2y, b1x, b1y, b2x, b2y;
		var i, j;
		for (i = 0; i < 4;
i++)
		{
			for (j = 0; j <
4; j++)
			{
				a1x =
this.at(i, true);
				a1y =
this.at(i, false);
				a2x =
this.at(i + 1, true);
```

```
                        a2y =
this.at(i + 1, false);
                        b1x =
rhs.at(j, true);
                        b1y =
rhs.at(j, false);
                        b2x =
rhs.at(j + 1, true);
                        b2y =
rhs.at(j + 1, false);
                        if
(cr.segments_intersect(a1x,
a1y, a2x, a2y, b1x, b1y, b2x,
b2y))

      return true;
                  }
            }
            return false;
      };
      cr.quad = Quad;
      cr.RGB = function (red,
green, blue)
      {
            return
Math.max(Math.min(red, 255), 0)
                  |
(Math.max(Math.min(green, 255),
0) << 8)
                  |
(Math.max(Math.min(blue, 255),
0) << 16);
      };
      cr.GetRValue = function
(rgb)
      {
            return rgb & 0xFF;
      };
      cr.GetGValue = function
(rgb)
      {
            return (rgb &
0xFF00) >> 8;
      };
      cr.GetBValue = function
(rgb)
      {
            return (rgb &
0xFF0000) >> 16;
      };
      cr.shallowCopy = function
(a, b, allowOverwrite)
      {
            var attr;
            for (attr in b)
            {
                  if
(b.hasOwnProperty(attr))
                  {
;
                        a[attr] =
b[attr];
                  }
            }
            return a;
      };
      cr.arrayRemove = function
(arr, index)
      {
            var i, len;
            index =
cr.floor(index);
            if (index < 0 ||
index >= arr.length)
                  return;

      // index out of bounds
            if (index === 0)
                        //
removing first item
                  arr.shift();
            else if (index ===
arr.length - 1)  // removing
last item
                  arr.pop();
            else
            {
                  for (i = index,
len = arr.length - 1; i < len;
i++)
                        arr[i] =
arr[i + 1];
                  arr.length =
len;
            }
      };
      cr.shallowAssignArray =
function (dest, src)
      {
            dest.length =
src.length;
            var i, len;
            for (i = 0, len =
src.length; i < len; i++)
                  dest[i] =
src[i];
      };
      cr.appendArray = function
(a, b)
      {
            a.push.apply(a, b);
      };
```

```
	cr.arrayFindRemove = function (arr, item)
	{
		var index = arr.indexOf(item);
		if (index !== -1)
	cr.arrayRemove(arr, index);
	};
	cr.clamp = function(x, a, b)
	{
		if (x < a)
			return a;
		else if (x > b)
			return b;
		else
			return x;
	};
	cr.to_radians = function(x)
	{
		return x / (180.0 / cr.PI);
	};
	cr.to_degrees = function(x)
	{
		return x * (180.0 / cr.PI);
	};
	cr.clamp_angle_degrees = function (a)
	{
		a %= 360;	// now in (-360, 360) range
		if (a < 0)
			a += 360;	// now in [0, 360) range
		return a;
	};
	cr.clamp_angle = function (a)
	{
		a %= 2 * cr.PI;	// now in (-2pi, 2pi) range
		if (a < 0)
			a += 2 * cr.PI;	// now in [0, 2pi) range
		return a;
	};
	cr.to_clamped_degrees = function (x)
	{
		return cr.clamp_angle_degrees(cr.to_degrees(x));
	};
	cr.to_clamped_radians = function (x)
	{
		return cr.clamp_angle(cr.to_radians(x));
	};
	cr.angleTo = function(x1, y1, x2, y2)
	{
		var dx = x2 - x1;
	    var dy = y2 - y1;
		return Math.atan2(dy, dx);
	};
	cr.angleDiff = function (a1, a2)
	{
		if (a1 === a2)
			return 0;
		var s1 = Math.sin(a1);
		var c1 = Math.cos(a1);
		var s2 = Math.sin(a2);
		var c2 = Math.cos(a2);
		var n = s1 * s2 + c1 * c2;
		if (n >= 1)
			return 0;
		if (n <= -1)
			return cr.PI;
		return Math.acos(n);
	};
	cr.angleRotate = function (start, end, step)
	{
		var ss = Math.sin(start);
		var cs = Math.cos(start);
		var se = Math.sin(end);
		var ce = Math.cos(end);
		if (Math.acos(ss * se + cs * ce) > step)
		{
			if (cs * se - ss * ce > 0)
```

```
                        return
cr.clamp_angle(start + step);
                else
                        return
cr.clamp_angle(start - step);
            }
            else
                return
cr.clamp_angle(end);
        };
        cr.angleClockwise =
function (a1, a2)
        {
            var s1 =
Math.sin(a1);
            var c1 =
Math.cos(a1);
            var s2 =
Math.sin(a2);
            var c2 =
Math.cos(a2);
            return c1 * s2 - s1
* c2 <= 0;
        };
        cr.rotatePtAround =
function (px, py, a, ox, oy,
getx)
        {
            if (a === 0)
                return getx ?
px : py;
            var sin_a =
Math.sin(a);
            var cos_a =
Math.cos(a);
            px -= ox;
            py -= oy;
            var left_sin_a = px
* sin_a;
            var top_sin_a = py *
sin_a;
            var left_cos_a = px
* cos_a;
            var top_cos_a = py *
cos_a;
            px = left_cos_a -
top_sin_a;
            py = top_cos_a +
left_sin_a;
            px += ox;
            py += oy;
            return getx ? px :
py;
        }
        cr.distanceTo =
function(x1, y1, x2, y2)
        {
            var dx = x2 - x1;
          var dy = y2 - y1;
            return
Math.sqrt(dx*dx + dy*dy);
        };
        cr.xor = function (x, y)
        {
            return !x !== !y;
        };
        cr.lerp = function (a, b,
x)
        {
            return a + (b - a) *
x;
        };
        cr.hasAnyOwnProperty =
function (o)
        {
            var p;
            for (p in o)
            {
                if
(o.hasOwnProperty(p))
                    return
true;
            }
            return false;
        };
        cr.wipe = function (obj)
        {
            var p;
            for (p in obj)
            {
                if
(obj.hasOwnProperty(p))
                    delete
obj[p];
            }
        };
        var startup_time = +(new
Date());
        cr.performance_now =
function()
        {
            if (typeof
window["performance"] !==
"undefined")
            {
                var winperf =
window["performance"];
                if (typeof
winperf.now !== "undefined")
                    return
winperf.now();
```

```
                else if (typeof
winperf["webkitNow"] !==
"undefined")
                        return
winperf["webkitNow"]();
                else if (typeof
winperf["mozNow"] !==
"undefined")
                        return
winperf["mozNow"]();
                else if (typeof
winperf["msNow"] !==
"undefined")
                        return
winperf["msNow"]();
            }
            return Date.now() -
startup_time;
        };
        var supports_set = (typeof
Set !== "undefined" && typeof
Set.prototype["forEach"] !==
"undefined");
        function ObjectSet_()
        {
            this.s = null;
            this.items = null;
            this.item_count = 0;
            if (supports_set)
            {
                this.s = new
Set();
            }
            else
            {
                this.items =
{};
            }
            this.values_cache =
[];
            this.cache_valid =
true;
            cr.seal(this);
        };
        ObjectSet_.prototype.conta
ins = function (x)
        {
            if (supports_set)
                return
this.s["has"](x);
            else
                return
this.items.hasOwnProperty(x.toS
tring());
        };
        ObjectSet_.prototype.add =
function (x)
        {
            if (supports_set)
            {
                if
(!this.s["has"](x))
                {
        this.s["add"](x);
        this.cache_valid = false;
                }
            }
            else
            {
                var str =
x.toString();
                if
(!this.items.hasOwnProperty(str
))
                {
        this.items[str] = x;
        this.item_count++;
        this.cache_valid = false;
                }
            }
            return this;
        };
        ObjectSet_.prototype.remov
e = function (x)
        {
            if (supports_set)
            {
                if
(this.s["has"](x))
                {
        this.s["delete"](x);
        this.cache_valid = false;
                }
            }
            else
            {
                var str =
x.toString();
                if
(this.items.hasOwnProperty(str)
)
                {
                    delete
this.items[str];
```

```
	this.item_count--;

	this.cache_valid = false;
		}
	}
	return this;
};
ObjectSet_.prototype.clear
= function ()
{
	if (supports_set)
	{

	this.s["clear"]();
	}
	else
	{
		this.items =
{};
		this.item_count
= 0;
	}

	this.values_cache.length =
0;
	this.cache_valid =
true;
	return this;
};
ObjectSet_.prototype.isEmp
ty = function ()
{
	if (supports_set)
		return
this.s["size"] === 0;
	else
		return
this.item_count === 0;
};
ObjectSet_.prototype.count
= function ()
{
	if (supports_set)
		return
this.s["size"];
	else
		return
this.item_count;
};
var current_arr = null;
var current_index = 0;
function
set_append_to_arr(x)
{
	current_arr[current_index+
+] = x;
};
ObjectSet_.prototype.updat
e_cache = function ()
{
	if
(this.cache_valid)
		return;
	if (supports_set)
	{

	this.values_cache.length =
this.s["size"];
		current_arr =
this.values_cache;
		current_index =
0;

	this.s["forEach"](set_appe
nd_to_arr);
;
		current_arr =
null;
		current_index =
0;
	}
	else
	{

	this.values_cache.length =
this.item_count;
		var p, n = 0;
		for (p in
this.items)
		{
			if
(this.items.hasOwnProperty(p))

	this.values_cache[n++] =
this.items[p];
		}
;
	}
	this.cache_valid =
true;
};
ObjectSet_.prototype.value
s = function ()
{
	this.update_cache();
	return
this.values_cache.slice(0);
};
```

```
	ObjectSet_.prototype.value
sRef = function ()
	{
		this.update_cache();
		return
this.values_cache;
	};
	cr.ObjectSet = ObjectSet_;
	function KahanAdder_()
	{
		this.c = 0;
	      this.y = 0;
	      this.t = 0;
	      this.sum = 0;
		cr.seal(this);
	};
	KahanAdder_.prototype.add
= function (v)
	{
		this.y = v - this.c;
	       this.t = this.sum +
this.y;
	       this.c = (this.t -
this.sum) - this.y;
	       this.sum = this.t;
	};
    KahanAdder_.prototype.reset
= function ()
    {
	      this.c = 0;
	      this.y = 0;
	      this.t = 0;
	      this.sum = 0;
    };
	cr.KahanAdder =
KahanAdder_;
	cr.regexp_escape =
function(text)
	{
		return
text.replace(/[-
[\]{}()*+?.,\\^$|#\s]/g,
"\\$&");
	};
	function
CollisionPoly_(pts_array_)
	{
		this.pts_cache = [];
		this.bboxLeft = 0;
		this.bboxTop = 0;
		this.bboxRight = 0;
		this.bboxBottom = 0;
		this.convexpolys =
null;	// for physics
behavior to cache separated
polys
	this.set_pts(pts_array_);
		cr.seal(this);
	};
	CollisionPoly_.prototype.s
et_pts = function(pts_array_)
	{
		this.pts_array =
pts_array_;
		this.pts_count =
pts_array_.length / 2;
		// x, y, x, y... in
array

	this.pts_cache.length =
pts_array_.length;
		this.cache_width = -
1;
		this.cache_height =
-1;
		this.cache_angle =
0;
	};
	CollisionPoly_.prototype.i
s_empty = function()
	{
		return
!this.pts_array.length;
	};
	CollisionPoly_.prototype.u
pdate_bbox = function ()
	{
		var myptscache =
this.pts_cache;
		var bboxLeft_ =
myptscache[0];
		var bboxRight_ =
bboxLeft_;
		var bboxTop_ =
myptscache[1];
		var bboxBottom_ =
bboxTop_;
		var x, y, i = 1, i2,
len = this.pts_count;
		for ( ; i < len;
++i)
		{
			i2 = i*2;
			x =
myptscache[i2];
			y =
myptscache[i2+1];
			if (x <
bboxLeft_)
				bboxLeft_
= x;
```

```
                if (x >
bboxRight_)
     bboxRight_ = x;
                if (y <
bboxTop_)
                     bboxTop_
= y;
                if (y >
bboxBottom_)

     bboxBottom_ = y;
           }
           this.bboxLeft =
bboxLeft_;
           this.bboxRight =
bboxRight_;
           this.bboxTop =
bboxTop_;
           this.bboxBottom =
bboxBottom_;
     };
     CollisionPoly_.prototype.s
et_from_rect = function(rc,
offx, offy)
     {

     this.pts_cache.length = 8;
           this.pts_count = 4;
           var myptscache =
this.pts_cache;
           myptscache[0] =
rc.left - offx;
           myptscache[1] =
rc.top - offy;
           myptscache[2] =
rc.right - offx;
           myptscache[3] =
rc.top - offy;
           myptscache[4] =
rc.right - offx;
           myptscache[5] =
rc.bottom - offy;
           myptscache[6] =
rc.left - offx;
           myptscache[7] =
rc.bottom - offy;
           this.cache_width =
rc.right - rc.left;
           this.cache_height =
rc.bottom - rc.top;
           this.update_bbox();
     };
     CollisionPoly_.prototype.s
et_from_quad = function(q,
offx, offy, w, h)
     {

     this.pts_cache.length = 8;
           this.pts_count = 4;
           var myptscache =
this.pts_cache;
           myptscache[0] =
q.tlx - offx;
           myptscache[1] =
q.tly - offy;
           myptscache[2] =
q.trx - offx;
           myptscache[3] =
q.try_ - offy;
           myptscache[4] =
q.brx - offx;
           myptscache[5] =
q.bry - offy;
           myptscache[6] =
q.blx - offx;
           myptscache[7] =
q.bly - offy;
           this.cache_width =
w;
           this.cache_height =
h;
           this.update_bbox();
     };
     CollisionPoly_.prototype.s
et_from_poly = function (r)
     {
           this.pts_count =
r.pts_count;

     cr.shallowAssignArray(this
.pts_cache, r.pts_cache);
           this.bboxLeft =
r.bboxLeft;
           this.bboxTop -
r.bboxTop;
           this.bboxRight =
r.bboxRight;
           this.bboxBottom =
r.bboxBottom;
     };
     CollisionPoly_.prototype.c
ache_poly = function(w, h, a)
     {
           if (this.cache_width
=== w && this.cache_height ===
h && this.cache_angle === a)
                 return;
     // cache up-to-date
           this.cache_width =
w;
```

```
            this.cache_height =
h;
            this.cache_angle =
a;
            var i, i2, i21, len,
x, y;
            var sina = 0;
            var cosa = 1;
            var myptsarray =
this.pts_array;
            var myptscache =
this.pts_cache;
            if (a !== 0)
            {
                  sina =
Math.sin(a);
                  cosa =
Math.cos(a);
            }
            for (i = 0, len =
this.pts_count; i < len; i++)
            {
                  i2 = i*2;
                  i21 = i2+1;
                  x =
myptsarray[i2] * w;
                  y =
myptsarray[i21] * h;
                  myptscache[i2]
= (x * cosa) - (y * sina);
                  myptscache[i21]
= (y * cosa) + (x * sina);
            }
            this.update_bbox();
      };
      CollisionPoly_.prototype.c
ontains_pt = function (a2x,
a2y)
      {
            var myptscache =
this.pts_cache;
            if (a2x ===
myptscache[0] && a2y ===
myptscache[1])
                  return true;
            var i, i2, imod, len
= this.pts_count;
            var a1x =
this.bboxLeft - 110;
            var a1y =
this.bboxTop - 101;
            var a3x =
this.bboxRight + 131
            var a3y =
this.bboxBottom + 120;
            var b1x, b1y, b2x,
b2y;
            var count1 = 0,
count2 = 0;
            for (i = 0; i < len;
i++)
            {
                  i2 = i*2;
                  imod =
((i+1)%len)*2;
                  b1x =
myptscache[i2];
                  b1y =
myptscache[i2+1];
                  b2x =
myptscache[imod];
                  b2y =
myptscache[imod+1];
                  if
(cr.segments_intersect(a1x,
a1y, a2x, a2y, b1x, b1y, b2x,
b2y))
                        count1++;
                  if
(cr.segments_intersect(a3x,
a3y, a2x, a2y, b1x, b1y, b2x,
b2y))
                        count2++;
            }
            return (count1 % 2
=== 1) || (count2 % 2 === 1);
      };
      CollisionPoly_.prototype.i
ntersects_poly = function (rhs,
offx, offy)
      {
            var rhspts =
rhs.pts_cache;
            var mypts =
this.pts_cache;
            if
(this.contains_pt(rhspts[0] +
offx, rhspts[1] + offy))
                  return true;
            if
(rhs.contains_pt(mypts[0] -
offx, mypts[1] - offy))
                  return true;
            var i, i2, imod,
leni, j, j2, jmod, lenj;
            var a1x, a1y, a2x,
a2y, b1x, b1y, b2x, b2y;
            for (i = 0, leni =
this.pts_count; i < leni; i++)
            {
                  i2 = i*2;
```

```
                imod =
((i+1)%leni)*2;
                a1x =
mypts[i2];
                a1y =
mypts[i2+1];
                a2x =
mypts[imod];
                a2y =
mypts[imod+1];
                for (j = 0,
lenj = rhs.pts_count; j < lenj;
j++)
                {
                      j2 = j*2;
                      jmod =
((j+1)%lenj)*2;
                      b1x =
rhspts[j2] + offx;
                      b1y =
rhspts[j2+1] + offy;
                      b2x =
rhspts[jmod] + offx;
                      b2y =
rhspts[jmod+1] + offy;
                      if
(cr.segments_intersect(a1x,
a1y, a2x, a2y, b1x, b1y, b2x,
b2y))

      return true;
                }
            }
            return false;
      };
      CollisionPoly_.prototype.i
ntersects_segment = function
(offx, offy, x1, y1, x2, y2)
      {
            var mypts =
this.pts_cache;
            if
(this.contains_pt(x1 - offx, y1
- offy))
                return true;
            var i, leni, i2,
imod;
            var a1x, a1y, a2x,
a2y;
            for (i = 0, leni =
this.pts_count; i < leni; i++)
            {
                i2 = i*2;
                imod =
((i+1)%leni)*2;
                a1x = mypts[i2]
+ offx;
                a1y =
mypts[i2+1] + offy;
                a2x =
mypts[imod] + offx;
                a2y =
mypts[imod+1] + offy;
                if
(cr.segments_intersect(x1, y1,
x2, y2, a1x, a1y, a2x, a2y))
                      return
true;
            }
            return false;
      };
      CollisionPoly_.prototype.m
irror = function (px)
      {
            var i, leni, i2;
            for (i = 0, leni =
this.pts_count; i < leni; ++i)
            {
                i2 = i*2;

      this.pts_cache[i2] = px *
2 - this.pts_cache[i2];
            }
      };
      CollisionPoly_.prototype.f
lip = function (py)
      {
            var i, leni, i21;
            for (i = 0, leni =
this.pts_count; i < leni; ++i)
            {
                i21 = i*2+1;

      this.pts_cache[i21] = py *
2 - this.pts_cache[i21];
            }
      };
      CollisionPoly_.prototype.d
iag = function ()
      {
            var i, leni, i2,
i21, temp;
            for (i = 0, leni =
this.pts_count; i < leni; ++i)
            {
                i2 = i*2;
                i21 = i2+1;
                temp =
this.pts_cache[i2];
```

```
	this.pts_cache[i2] = 
this.pts_cache[i21];

	this.pts_cache[i21] = 
temp;
		}
	};
	cr.CollisionPoly = 
CollisionPoly_;
	function 
SparseGrid_(cellwidth_, 
cellheight_)
	{
		this.cellwidth = 
cellwidth_;
		this.cellheight = 
cellheight_;
		this.cells = {};
	};
	SparseGrid_.prototype.tota
lCellCount = 0;
	SparseGrid_.prototype.getC
ell = function (x_, y_, 
create_if_missing)
	{
		var ret;
		var col = 
this.cells[x_];
		if (!col)
		{
			if 
(create_if_missing)
			{
				ret = 
allocGridCell(this, x_, y_);

	this.cells[x_] = {};

	this.cells[x_][y_] = ret;
				return 
ret;
			}
			else 
				return 
null;
		}
		ret = col[y_];
		if (ret)
			return ret;
		else if 
(create_if_missing)
		{
			ret = 
allocGridCell(this, x_, y_);

	this.cells[x_][y_] = ret;
				return ret;
			}
			else 
				return null;
	};
	SparseGrid_.prototype.XToC
ell = function (x_)
	{
		return cr.floor(x_ / 
this.cellwidth);
	};
	SparseGrid_.prototype.YToC
ell = function (y_)
	{
		return cr.floor(y_ / 
this.cellheight);
	};
	SparseGrid_.prototype.upda
te = function (inst, oldrange, 
newrange)
	{
		var x, lenx, y, 
leny, cell;
		if (oldrange)
		{
			for (x = 
oldrange.left, lenx = 
oldrange.right; x <= lenx; ++x)
			{
				for (y = 
oldrange.top, leny = 
oldrange.bottom; y <= leny; 
++y)
				{
					if 
(newrange && 
newrange.contains_pt(x, y))

	continue;  // is still in 
this cell
					cell 
= this.getCell(x, y, false);
	// don't create if missing
					if 
(!cell)

	continue;  // cell does 
not exist yet

	cell.remove(inst);
					if 
(cell.isEmpty())
					{
```

```
freeGridCell(cell);

this.cells[x][y] = null;
                        }
                    }
                }
            }
            if (newrange)
            {
                for (x =
newrange.left, lenx =
newrange.right; x <= lenx; ++x)
                {
                    for (y =
newrange.top, leny =
newrange.bottom; y <= leny;
++y)
                    {
                        if
(oldrange &&
oldrange.contains_pt(x, y))

continue;  // is still in
this cell

this.getCell(x, y,
true).insert(inst);
                    }
                }
            }
        };
        SparseGrid_.prototype.quer
yRange = function (rc, result)
        {
            var x, lenx, ystart,
y, leny, cell;
            x =
this.XToCell(rc.left);
            ystart =
this.YToCell(rc.top);
            lenx =
this.XToCell(rc.right);
            leny =
this.YToCell(rc.bottom);
            for ( ; x <= lenx;
++x)
            {
                for (y =
ystart; y <= leny; ++y)
                {
                    cell =
this.getCell(x, y, false);
                    if
(!cell)
continue;

cell.dump(result);
                    }
                }
            };
            cr.SparseGrid =
SparseGrid_;
            var gridcellcache = [];
            function
allocGridCell(grid_, x_, y_)
            {
                var ret;

SparseGrid_.prototype.tota
lCellCount++;
                if
(gridcellcache.length)
                {
                    ret =
gridcellcache.pop();
                    ret.grid =
grid_;
                    ret.x = x_;
                    ret.y = y_;
                    return ret;
                }
                else
                    return new
cr.GridCell(grid_, x_, y_);
            };
            function freeGridCell(c)
            {

SparseGrid_.prototype.tota
lCellCount--;
                c.objects.clear();
                if
(gridcellcache.length < 1000)

gridcellcache.push(c);
            };
            function GridCell_(grid_,
x_, y_)
            {
                this.grid = grid_;
                this.x = x_;
                this.y = y_;
                this.objects = new
cr.ObjectSet();
            };
            GridCell_.prototype.isEmpt
y = function ()
            {
```

```
            return
this.objects.isEmpty();
      };
      GridCell_.prototype.insert
= function (inst)
      {

      this.objects.add(inst);
      };
      GridCell_.prototype.remove
= function (inst)
      {

      this.objects.remove(inst);
      };
      GridCell_.prototype.dump =
function (result)
      {

      cr.appendArray(result,
this.objects.valuesRef());
      };
      cr.GridCell = GridCell_;
      var fxNames = [ "lighter",

      "xor",

      "copy",

      "destination-over",

      "source-in",

      "destination-in",

      "source-out",

      "destination-out",

      "source-atop",

      "destination-atop"];
      cr.effectToCompositeOp =
function(effect)
      {
            if (effect <= 0 ||
effect >= 11)
                  return "source-
over";
            return
fxNames[effect - 1];  // not
including "none" so offset by 1
      };
      cr.setGLBlend =
function(this_, effect, gl)
      {
            if (!gl)
                  return;
            this_.srcBlend =
gl.ONE;
            this_.destBlend =
gl.ONE_MINUS_SRC_ALPHA;
            switch (effect) {
            case 1:         //
lighter (additive)
                  this_.srcBlend
= gl.ONE;
                  this_.destBlend
= gl.ONE;
                  break;
            case 2:         //
xor
                  break;     //
todo
            case 3:         //
copy
                  this_.srcBlend
= gl.ONE;
                  this_.destBlend
= gl.ZERO;
                  break;
            case 4:         //
destination-over
                  this_.srcBlend
= gl.ONE_MINUS_DST_ALPHA;
                  this_.destBlend
= gl.ONE;
                  break;
            case 5:         //
source-in
                  this_.srcBlend
= gl.DST_ALPHA;
                  this_.destBlend
= gl.ZERO;
                  break;
            case 6:         //
destination-in
                  this_.srcBlend
= gl.ZERO;
                  this_.destBlend
= gl.SRC_ALPHA;
                  break;
            case 7:         //
source-out
                  this_.srcBlend
= gl.ONE_MINUS_DST_ALPHA;
                  this_.destBlend
= gl.ZERO;
                  break;
            case 8:         //
destination-out
```

```
                this_.srcBlend
= gl.ZERO;
                this_.destBlend
= gl.ONE_MINUS_SRC_ALPHA;
                break;
           case 9:         //
source-atop
                this_.srcBlend
= gl.DST_ALPHA;
                this_.destBlend
= gl.ONE_MINUS_SRC_ALPHA;
                break;
           case 10:   //
destination-atop
                this_.srcBlend
= gl.ONE_MINUS_DST_ALPHA;
                this_.destBlend
= gl.SRC_ALPHA;
                break;
           }
      };
      cr.round6dp = function (x)
      {
           return cr.round(x *
1000000) / 1000000;
      };
      /*
      var localeCompare_options
= {
           "usage": "search",
           "sensitivity":
"accent"
      };
      var has_localeCompare =
!!"a".localeCompare;
      var localeCompare_works1 =
(has_localeCompare &&
"a".localeCompare("A",
undefined,
localeCompare_options) === 0);
      var localeCompare_works2 =
(has_localeCompare &&
"a".localeCompare("á",
undefined,
localeCompare_options) !== 0);
      var supports_localeCompare
= (has_localeCompare &&
localeCompare_works1 &&
localeCompare_works2);
      */
      cr.equals_nocase =
function (a, b)
      {
           if (typeof a !==
"string" || typeof b !==
"string")
                return false;
           if (a.length !==
b.length)
                return false;
           if (a === b)
                return true;
           /*
           if
(supports_localeCompare)
           {
                return
(a.localeCompare(b, undefined,
localeCompare_options) === 0);
           }
           else
           {
           */
                return
a.toLowerCase() ===
b.toLowerCase();
      };
}());
var MatrixArray=typeof
Float32Array!=="undefined"?Floa
t32Array:Array,glMatrixArrayTyp
e=MatrixArray,vec3={},mat3={},m
at4={},quat4={};vec3.create=fun
ction(a){var b=new
MatrixArray(3);a&&(b[0]=a[0],b[
1]=a[1],b[2]=a[2]);return
b};vec3.set=function(a,b){b[0]=
a[0];b[1]=a[1];b[2]=a[2];return
b};vec3.add=function(a,b,c){if(
!c||a===c)return
a[0]+=b[0],a[1]+=b[1],a[2]+=b[2
],a;c[0]=a[0]+b[0];c[1]=a[1]+b[
1];c[2]=a[2]+b[2];return c};
vec3.subtract=function(a,b,c){i
f(!c||a===c)return a[0]-
=b[0],a[1]-=b[1],a[2]-
=b[2],a;c[0]=a[0]-
b[0];c[1]=a[1]-b[1];c[2]=a[2]-
b[2];return
c};vec3.negate=function(a,b){b|
|(b=a);b[0]=-a[0];b[1]=-
a[1];b[2]=-a[2];return
b};vec3.scale=function(a,b,c){i
f(!c||a===c)return
a[0]*=b,a[1]*=b,a[2]*=b,a;c[0]=
a[0]*b;c[1]=a[1]*b;c[2]=a[2]*b;
return c};
vec3.normalize=function(a,b){b|
|(b=a);var
c=a[0],d=a[1],e=a[2],g=Math.sqr
t(c*c+d*d+e*e);if(g){if(g===1)r
eturn
```

```
b[0]=c,b[1]=d,b[2]=e,b}else
return
b[0]=0,b[1]=0,b[2]=0,b;g=1/g;b[
0]=c*g;b[1]=d*g;b[2]=e*g;return
b};vec3.cross=function(a,b,c){c
||(c=a);var
d=a[0],e=a[1],a=a[2],g=b[0],f=b
[1],b=b[2];c[0]=e*b-
a*f;c[1]=a*g-d*b;c[2]=d*f-
e*g;return
c};vec3.length=function(a){var
b=a[0],c=a[1],a=a[2];return
Math.sqrt(b*b+c*c+a*a)};vec3.do
t=function(a,b){return
a[0]*b[0]+a[1]*b[1]+a[2]*b[2]};
vec3.direction=function(a,b,c){
c||(c=a);var d=a[0]-
b[0],e=a[1]-b[1],a=a[2]-
b[2],b=Math.sqrt(d*d+e*e+a*a);i
f(!b)return
c[0]=0,c[1]=0,c[2]=0,c;b=1/b;c[
0]=d*b;c[1]=e*b;c[2]=a*b;return
c};vec3.lerp=function(a,b,c,d){
d||(d=a);d[0]=a[0]+c*(b[0]-
a[0]);d[1]=a[1]+c*(b[1]-
a[1]);d[2]=a[2]+c*(b[2]-
a[2]);return
d};vec3.str=function(a){return"
["+a[0]+", "+a[1]+",
"+a[2]+"]"};
mat3.create=function(a){var
b=new
MatrixArray(9);a&&(b[0]=a[0],b[
1]=a[1],b[2]=a[2],b[3]=a[3],b[4
]=a[4],b[5]=a[5],b[6]=a[6],b[7]
=a[7],b[8]=a[8]);return
b};mat3.set=function(a,b){b[0]=
a[0];b[1]=a[1];b[2]=a[2];b[3]=a
[3];b[4]=a[4];b[5]=a[5];b[6]=a[
6];b[7]=a[7];b[8]=a[8];return
b};mat3.identity=function(a){a[
0]=1;a[1]=0;a[2]=0;a[3]=0;a[4]=
1;a[5]=0;a[6]=0;a[7]=0;a[8]=1;r
eturn a};
mat3.transpose=function(a,b){if
(!b||a===b){var
c=a[1],d=a[2],e=a[5];a[1]=a[3];
a[2]=a[6];a[3]=c;a[5]=a[7];a[6]
=d;a[7]=e;return
a}b[0]=a[0];b[1]=a[3];b[2]=a[6]
;b[3]=a[1];b[4]=a[4];b[5]=a[7];
b[6]=a[2];b[7]=a[5];b[8]=a[8];r
eturn
b};mat3.toMat4=function(a,b){b|
|(b=mat4.create());b[15]=1;b[14
]=0;b[13]=0;b[12]=0;b[11]=0;b[1
0]=a[8];b[9]=a[7];b[8]=a[6];b[7
]=0;b[6]=a[5];b[5]=a[4];b[4]=a[
3];b[3]=0;b[2]=a[2];b[1]=a[1];b
[0]=a[0];return b};
mat3.str=function(a){return"["+
a[0]+", "+a[1]+", "+a[2]+",
"+a[3]+", "+a[4]+", "+a[5]+",
"+a[6]+", "+a[7]+",
"+a[8]+"]"};mat4.create=functio
n(a){var b=new
MatrixArray(16);a&&(b[0]=a[0],b
[1]=a[1],b[2]=a[2],b[3]=a[3],b[
4]=a[4],b[5]=a[5],b[6]=a[6],b[7
]=a[7],b[8]=a[8],b[9]=a[9],b[10
]=a[10],b[11]=a[11],b[12]=a[12]
,b[13]=a[13],b[14]=a[14],b[15]=
a[15]);return b};
mat4.set=function(a,b){b[0]=a[0
];b[1]=a[1];b[2]=a[2];b[3]=a[3]
;b[4]=a[4];b[5]=a[5];b[6]=a[6];
b[7]=a[7];b[8]=a[8];b[9]=a[9];b
[10]=a[10];b[11]=a[11];b[12]=a[
12];b[13]=a[13];b[14]=a[14];b[1
5]=a[15];return
b};mat4.identity=function(a){a[
0]=1;a[1]=0;a[2]=0;a[3]=0;a[4]=
0;a[5]=1;a[6]=0;a[7]=0;a[8]=0;a
[9]=0;a[10]=1;a[11]=0;a[12]=0;a
[13]=0;a[14]=0;a[15]=1;return
a};
mat4.transpose=function(a,b){if
(!b||a===b){var
c=a[1],d=a[2],e=a[3],g=a[6],f=a
[7],h=a[11];a[1]=a[4];a[2]=a[8]
;a[3]=a[12];a[4]=c;a[6]=a[9];a[
7]=a[13];a[8]=d;a[9]=g;a[11]=a[
14];a[12]=e;a[13]=f;a[14]=h;ret
urn
a}b[0]=a[0];b[1]=a[4];b[2]=a[8]
;b[3]=a[12];b[4]=a[1];b[5]=a[5]
;b[6]=a[9];b[7]=a[13];b[8]=a[2]
;b[9]=a[6];b[10]=a[10];b[11]=a[
14];b[12]=a[3];b[13]=a[7];b[14]
=a[11];b[15]=a[15];return b};
mat4.determinant=function(a){va
r
b=a[0],c=a[1],d=a[2],e=a[3],g=a
[4],f=a[5],h=a[6],i=a[7],j=a[8]
,k=a[9],l=a[10],n=a[11],o=a[12]
,m=a[13],p=a[14],a=a[15];return
o*k*h*e-j*m*h*e-
o*f*l*e+g*m*l*e+j*f*p*e-
g*k*p*e-
o*k*d*i+j*m*d*i+o*c*l*i-
b*m*l*i-
j*c*p*i+b*k*p*i+o*f*d*n-
```

```
g*m*d*n-
o*c*h*n+b*m*h*n+g*c*p*n-
b*f*p*n-
j*f*d*a+g*k*d*a+j*c*h*a-
b*k*h*a-g*c*l*a+b*f*l*a};
mat4.inverse=function(a,b){b||(
b=a);var
c=a[0],d=a[1],e=a[2],g=a[3],f=a
[4],h=a[5],i=a[6],j=a[7],k=a[8]
,l=a[9],n=a[10],o=a[11],m=a[12]
,p=a[13],r=a[14],s=a[15],A=c*h-
d*f,B=c*i-e*f,t=c*j-g*f,u=d*i-
e*h,v=d*j-g*h,w=e*j-g*i,x=k*p-
l*m,y=k*r-n*m,z=k*s-o*m,C=l*r-
n*p,D=l*s-o*p,E=n*s-
o*r,q=1/(A*E-B*D+t*C+u*z-
v*y+w*x);b[0]=(h*E-
i*D+j*C)*q;b[1]=(-d*E+e*D-
g*C)*q;b[2]=(p*w-
r*v+s*u)*q;b[3]=(-l*w+n*v-
o*u)*q;b[4]=(-f*E+i*z-
j*y)*q;b[5]=(c*E-
e*z+g*y)*q;b[6]=(-m*w+r*t-
s*B)*q;b[7]=(k*w-
n*t+o*B)*q;b[8]=(f*D-
h*z+j*x)*q;
b[9]=(-c*D+d*z-
g*x)*q;b[10]=(m*v-
p*t+s*A)*q;b[11]=(-k*v+l*t-
o*A)*q;b[12]=(-f*C+h*y-
i*x)*q;b[13]=(c*C-
d*y+e*x)*q;b[14]=(-m*u+p*B-
r*A)*q;b[15]=(k*u-
l*B+n*A)*q;return
b};mat4.toRotationMat=function(
a,b){b||(b=mat4.create());b[0]=
a[0];b[1]=a[1];b[2]=a[2];b[3]=a
[3];b[4]=a[4];b[5]=a[5];b[6]=a[
6];b[7]=a[7];b[8]=a[8];b[9]=a[9
];b[10]=a[10];b[11]=a[11];b[12]
=0;b[13]=0;b[14]=0;b[15]=1;retu
rn b};
mat4.toMat3=function(a,b){b||(b
=mat3.create());b[0]=a[0];b[1]=
a[1];b[2]=a[2];b[3]=a[4];b[4]=a
[5];b[5]=a[6];b[6]=a[8];b[7]=a[
9];b[8]=a[10];return
b};mat4.toInverseMat3=function(
a,b){var
c=a[0],d=a[1],e=a[2],g=a[4],f=a
[5],h=a[6],i=a[8],j=a[9],k=a[10
],l=k*f-h*j,n=-k*g+h*i,o=j*g-
f*i,m=c*l+d*n+e*o;if(!m)return
null;m=1/m;b||(b=mat3.create())
;b[0]=l*m;b[1]=(-
k*d+e*j)*m;b[2]=(h*d-
e*f)*m;b[3]=n*m;b[4]=(k*c-
e*i)*m;b[5]=(-
h*c+e*g)*m;b[6]=o*m;b[7]=(-
j*c+d*i)*m;b[8]=(f*c-
d*g)*m;return b};
mat4.multiply=function(a,b,c){c
||(c=a);var
d=a[0],e=a[1],g=a[2],f=a[3],h=a
[4],i=a[5],j=a[6],k=a[7],l=a[8]
,n=a[9],o=a[10],m=a[11],p=a[12]
,r=a[13],s=a[14],a=a[15],A=b[0]
,B=b[1],t=b[2],u=b[3],v=b[4],w=
b[5],x=b[6],y=b[7],z=b[8],C=b[9
],D=b[10],E=b[11],q=b[12],F=b[1
3],G=b[14],b=b[15];c[0]=A*d+B*h
+t*l+u*p;c[1]=A*e+B*i+t*n+u*r;c
[2]=A*g+B*j+t*o+u*s;c[3]=A*f+B*
k+t*m+u*a;c[4]=v*d+w*h+x*l+y*p;
c[5]=v*e+w*i+x*n+y*r;c[6]=v*g+w
*j+x*o+y*s;c[7]=v*f+w*k+x*m+y*a
;c[8]=z*d+C*h+D*l+E*p;c[9]=z*e+
C*i+D*n+E*r;c[10]=z*g+C*
j+D*o+E*s;c[11]=z*f+C*k+D*m+E*a
;c[12]=q*d+F*h+G*l+b*p;c[13]=q*
e+F*i+G*n+b*r;c[14]=q*g+F*j+G*o
+b*s;c[15]=q*f+F*k+G*m+b*a;retu
rn
c};mat4.multiplyVec3=function(a
,b,c){c||(c=b);var
d=b[0],e=b[1],b=b[2];c[0]=a[0]*
d+a[4]*e+a[8]*b+a[12];c[1]=a[1]
*d+a[5]*e+a[9]*b+a[13];c[2]=a[2
]*d+a[6]*e+a[10]*b+a[14];return
c};
mat4.multiplyVec4=function(a,b,
c){c||(c=b);var
d=b[0],e=b[1],g=b[2],b=b[3];c[0
]=a[0]*d+a[4]*e+a[8]*g+a[12]*b;
c[1]=a[1]*d+a[5]*e+a[9]*g+a[13]
*b;c[2]=a[2]*d+a[6]*e+a[10]*g+a
[14]*b;c[3]=a[3]*d+a[7]*e+a[11]
*g+a[15]*b;return c};
mat4.translate=function(a,b,c){
var
d=b[0],e=b[1],b=b[2],g,f,h,i,j,
k,l,n,o,m,p,r;if(!c||a===c)retu
rn
a[12]=a[0]*d+a[4]*e+a[8]*b+a[12
],a[13]=a[1]*d+a[5]*e+a[9]*b+a[
13],a[14]=a[2]*d+a[6]*e+a[10]*b
+a[14],a[15]=a[3]*d+a[7]*e+a[11
]*b+a[15],a;g=a[0];f=a[1];h=a[2
];i=a[3];j=a[4];k=a[5];l=a[6];n
=a[7];o=a[8];m=a[9];p=a[10];r=a
[11];c[0]=g;c[1]=f;c[2]=h;c[3]=
i;c[4]=j;c[5]=k;c[6]=l;c[7]=n;c
```

```
[8]=o;c[9]=m;c[10]=p;c[11]=r;c[
12]=g*d+j*e+o*b+a[12];c[13]=f*d
+k*e+m*b+a[13];c[14]=h*d+l*e+p*
b+a[14];c[15]=i*d+n*e+r*b+a[15]
;
return
c};mat4.scale=function(a,b,c){v
ar
d=b[0],e=b[1],b=b[2];if(!c||a==
=c)return
a[0]*=d,a[1]*=d,a[2]*=d,a[3]*=d
,a[4]*=e,a[5]*=e,a[6]*=e,a[7]*=
e,a[8]*=b,a[9]*=b,a[10]*=b,a[11
]*=b,a;c[0]=a[0]*d;c[1]=a[1]*d;
c[2]=a[2]*d;c[3]=a[3]*d;c[4]=a[
4]*e;c[5]=a[5]*e;c[6]=a[6]*e;c[
7]=a[7]*e;c[8]=a[8]*b;c[9]=a[9]
*b;c[10]=a[10]*b;c[11]=a[11]*b;
c[12]=a[12];c[13]=a[13];c[14]=a
[14];c[15]=a[15];return c};
mat4.rotate=function(a,b,c,d){v
ar
e=c[0],g=c[1],c=c[2],f=Math.sqr
t(e*e+g*g+c*c),h,i,j,k,l,n,o,m,
p,r,s,A,B,t,u,v,w,x,y,z;if(!f)r
eturn
null;f!==1&&(f=1/f,e*=f,g*=f,c*
=f);h=Math.sin(b);i=Math.cos(b)
;j=1-
i;b=a[0];f=a[1];k=a[2];l=a[3];n
=a[4];o=a[5];m=a[6];p=a[7];r=a[
8];s=a[9];A=a[10];B=a[11];t=e*e
*j+i;u=g*e*j+c*h;v=c*e*j-
g*h;w=e*g*j-
c*h;x=g*g*j+i;y=c*g*j+e*h;z=e*c
*j+g*h;e=g*c*j-
e*h;g=c*c*j+i;d?a!==d&&(d[12]=a
[12],d[13]=a[13],d[14]=a[14],d[
15]=a[15]):d=a;d[0]=b*t+n*u+r*v
;d[1]=f*t+o*u+s*v;d[2]=k*t+m*u+
A*
v;d[3]=l*t+p*u+B*v;d[4]=b*w+n*x
+r*y;d[5]=f*w+o*x+s*y;d[6]=k*w+
m*x+A*y;d[7]=l*w+p*x+B*y;d[8]=b
*z+n*e+r*g;d[9]=f*z+o*e+s*g;d[1
0]=k*z+m*e+A*g;d[11]=l*z+p*e+B*
g;return
d};mat4.rotateX=function(a,b,c)
{var
d=Math.sin(b),b=Math.cos(b),e=a
[4],g=a[5],f=a[6],h=a[7],i=a[8]
,j=a[9],k=a[10],l=a[11];c?a!==c
&&(c[0]=a[0],c[1]=a[1],c[2]=a[2
],c[3]=a[3],c[12]=a[12],c[13]=a
[13],c[14]=a[14],c[15]=a[15]):c
=a;c[4]=e*b+i*d;c[5]=g*b+j*d;c[
6]=f*b+k*d;c[7]=h*b+l*d;c[8]=e*
-d+i*b;c[9]=g*-d+j*b;c[10]=f*-
d+k*b;c[11]=h*-d+l*b;return c};
mat4.rotateY=function(a,b,c){va
r
d=Math.sin(b),b=Math.cos(b),e=a
[0],g=a[1],f=a[2],h=a[3],i=a[8]
,j=a[9],k=a[10],l=a[11];c?a!==c
&&(c[4]=a[4],c[5]=a[5],c[6]=a[6
],c[7]=a[7],c[12]=a[12],c[13]=a
[13],c[14]=a[14],c[15]=a[15]):c
=a;c[0]=e*b+i*-d;c[1]=g*b+j*-
d;c[2]=f*b+k*-d;c[3]=h*b+l*-
d;c[8]=e*d+i*b;c[9]=g*d+j*b;c[1
0]=f*d+k*b;c[11]=h*d+l*b;return
c};
mat4.rotateZ=function(a,b,c){va
r
d=Math.sin(b),b=Math.cos(b),e=a
[0],g=a[1],f=a[2],h=a[3],i=a[4]
,j=a[5],k=a[6],l=a[7];c?a!==c&&
(c[8]=a[8],c[9]=a[9],c[10]=a[10
],c[11]=a[11],c[12]=a[12],c[13]
=a[13],c[14]=a[14],c[15]=a[15])
:c=a;c[0]=e*b+i*d;c[1]=g*b+j*d;
c[2]=f*b+k*d;c[3]=h*b+l*d;c[4]=
e*-d+i*b;c[5]=g*-d+j*b;c[6]=f*-
d+k*b;c[7]=h*-d+l*b;return c};
mat4.frustum=function(a,b,c,d,e
,g,f){f||(f=mat4.create());var
h=b-a,i=d-c,j=g-
e;f[0]=e*2/h;f[1]=0;f[2]=0;f[3]
=0;f[4]=0;f[5]=e*2/i;f[6]=0;f[7
]=0;f[8]=(b+a)/h;f[9]=(d+c)/i;f
[10]=-(g+e)/j;f[11]=-
1;f[12]=0;f[13]=0;f[14]=-
(g*e*2)/j;f[15]=0;return
f};mat4.perspective=function(a,
b,c,d,e){a=c*Math.tan(a*Math.PI
/360);b*=a;return
mat4.frustum(-b,b,-a,a,c,d,e)};
mat4.ortho=function(a,b,c,d,e,g
,f){f||(f=mat4.create());var
h=b-a,i=d-c,j=g-
e;f[0]=2/h;f[1]=0;f[2]=0;f[3]=0
;f[4]=0;f[5]=2/i;f[6]=0;f[7]=0;
f[8]=0;f[9]=0;f[10]=-
2/j;f[11]=0;f[12]=-
(a+b)/h;f[13]=-(d+c)/i;f[14]=-
(g+e)/j;f[15]=1;return f};
mat4.lookAt=function(a,b,c,d){d
||(d=mat4.create());var
e,g,f,h,i,j,k,l,n=a[0],o=a[1],a
=a[2];g=c[0];f=c[1];e=c[2];c=b[
1];j=b[2];if(n===b[0]&&o===c&&a
===j)return
```

```
mat4.identity(d);c=n-b[0];j=o-
b[1];k=a-
b[2];l=1/Math.sqrt(c*c+j*j+k*k)
;c*=l;j*=l;k*=l;b=f*k-
e*j;e=e*c-g*k;g=g*j-
f*c;(l=Math.sqrt(b*b+e*e+g*g))?
(l=1/l,b*=l,e*=l,g*=l):g=e=b=0;
f=j*g-k*e;h=k*b-c*g;i=c*e-
j*b;(l=Math.sqrt(f*f+h*h+i*i))?
(l=1/l,f*=l,h*=l,i*=l):i=h=f=0;
d[0]=b;d[1]=f;d[2]=c;d[3]=0;d[4
]=e;d[5]=h;d[6]=j;d[7]=0;d[8]=g
;d[9]=i;d[10]=k;d[11]=
0;d[12]=-(b*n+e*o+g*a);d[13]=-
(f*n+h*o+i*a);d[14]=-
(c*n+j*o+k*a);d[15]=1;return
d};mat4.fromRotationTranslation
=function(a,b,c){c||(c=mat4.cre
ate());var
d=a[0],e=a[1],g=a[2],f=a[3],h=d
+d,i=e+e,j=g+g,a=d*h,k=d*i;d*=j
;var
l=e*i;e*=j;g*=j;h*=f;i*=f;f*=j;
c[0]=1-(l+g);c[1]=k+f;c[2]=d-
i;c[3]=0;c[4]=k-f;c[5]=1-
(a+g);c[6]=e+h;c[7]=0;c[8]=d+i;
c[9]=e-h;c[10]=1-
(a+l);c[11]=0;c[12]=b[0];c[13]=
b[1];c[14]=b[2];c[15]=1;return
c};
mat4.str=function(a){return"["+
a[0]+", "+a[1]+", "+a[2]+",
"+a[3]+", "+a[4]+", "+a[5]+",
"+a[6]+", "+a[7]+", "+a[8]+",
"+a[9]+", "+a[10]+", "+a[11]+",
"+a[12]+", "+a[13]+",
"+a[14]+",
"+a[15]+"]"};quat4.create=funct
ion(a){var b=new
MatrixArray(4);a&&(b[0]=a[0],b[
1]=a[1],b[2]=a[2],b[3]=a[3]);re
turn
b};quat4.set=function(a,b){b[0]
=a[0];b[1]=a[1];b[2]=a[2];b[3]=
a[3];return b};
quat4.calculateW=function(a,b){
var
c=a[0],d=a[1],e=a[2];if(!b||a==
=b)return a[3]=-
Math.sqrt(Math.abs(1-c*c-d*d-
e*e)),a;b[0]=c;b[1]=d;b[2]=e;b[
3]=-Math.sqrt(Math.abs(1-c*c-
d*d-e*e));return
b};quat4.inverse=function(a,b){
if(!b||a===b)return a[0]*=-
1,a[1]*=-1,a[2]*=-1,a;b[0]=-
a[0];b[1]=-a[1];b[2]=-
a[2];b[3]=a[3];return
b};quat4.length=function(a){var
b=a[0],c=a[1],d=a[2],a=a[3];ret
urn
Math.sqrt(b*b+c*c+d*d+a*a)};
quat4.normalize=function(a,b){b
||(b=a);var
c=a[0],d=a[1],e=a[2],g=a[3],f=M
ath.sqrt(c*c+d*d+e*e+g*g);if(f=
==0)return
b[0]=0,b[1]=0,b[2]=0,b[3]=0,b;f
=1/f;b[0]=c*f;b[1]=d*f;b[2]=e*f
;b[3]=g*f;return
b};quat4.multiply=function(a,b,
c){c||(c=a);var
d=a[0],e=a[1],g=a[2],a=a[3],f=b
[0],h=b[1],i=b[2],b=b[3];c[0]=d
*b+a*f+e*i-
g*h;c[1]=e*b+a*h+g*f-
d*i;c[2]=g*b+a*i+d*h-
e*f;c[3]=a*b-d*f-e*h-g*i;return
c};
quat4.multiplyVec3=function(a,b
,c){c||(c=b);var
d=b[0],e=b[1],g=b[2],b=a[0],f=a
[1],h=a[2],a=a[3],i=a*d+f*g-
h*e,j=a*e+h*d-b*g,k=a*g+b*e-
f*d,d=-b*d-f*e-h*g;c[0]=i*a+d*-
b+j*-h-k*-f;c[1]=j*a+d*-f+k*-b-
i*-h;c[2]=k*a+d*-h+i*-f-j*-
b;return
c};quat4.toMat3=function(a,b){b
||(b=mat3.create());var
c=a[0],d=a[1],e=a[2],g=a[3],f=c
+c,h=d+d,i=e+e,j=c*f,k=c*h;c*=i
;var
l=d*h;d*=i;e*=i;f*=g;h*=g;g*=i;
b[0]=1-(l+e);b[1]=k+g;b[2]=c-
h;b[3]=k-g;b[4]=1-
(j+e);b[5]=d+f;b[6]=c+h;b[7]=d-
f;b[8]=1-(j+l);return b};
quat4.toMat4=function(a,b){b||(
b=mat4.create());var
c=a[0],d=a[1],e=a[2],g=a[3],f=c
+c,h=d+d,i=e+e,j=c*f,k=c*h;c*=i
;var
l=d*h;d*=i;e*=i;f*=g;h*=g;g*=i;
b[0]=1-(l+e);b[1]=k+g;b[2]=c-
h;b[3]=0;b[4]=k-g;b[5]=1-
(j+e);b[6]=d+f;b[7]=0;b[8]=c+h;
b[9]=d-f;b[10]=1-
(j+l);b[11]=0;b[12]=0;b[13]=0;b
[14]=0;b[15]=1;return b};
quat4.slerp=function(a,b,c,d){d
||(d=a);var
```

```
e=a[0]*b[0]+a[1]*b[1]+a[2]*b[2]
+a[3]*b[3],g,f;if(Math.abs(e)>=
1)return
d!==a&&(d[0]=a[0],d[1]=a[1],d[2
]=a[2],d[3]=a[3]),d;g=Math.acos
(e);f=Math.sqrt(1-
e*e);if(Math.abs(f)<0.001)retur
n
d[0]=a[0]*0.5+b[0]*0.5,d[1]=a[1
]*0.5+b[1]*0.5,d[2]=a[2]*0.5+b[
2]*0.5,d[3]=a[3]*0.5+b[3]*0.5,d
;e=Math.sin((1-
c)*g)/f;c=Math.sin(c*g)/f;d[0]=
a[0]*e+b[0]*c;d[1]=a[1]*e+b[1]*
c;d[2]=a[2]*e+b[2]*c;d[3]=a[3]*
e+b[3]*c;return d};
quat4.str=function(a){return"["
+a[0]+", "+a[1]+", "+a[2]+",
"+a[3]+"]"};
(function()
{
	var MAX_VERTICES = 8000;

	// equates to 2500 objects
being drawn
	var MAX_INDICES =
(MAX_VERTICES / 2) * 3;
	// 6 indices for every 4
vertices
	var MAX_POINTS = 8000;
	var MULTI_BUFFERS = 4;

		// cycle 4 buffers
to try and avoid blocking
	var BATCH_NULL = 0;
	var BATCH_QUAD = 1;
	var BATCH_SETTEXTURE = 2;
	var BATCH_SETOPACITY = 3;
	var BATCH_SETBLEND = 4;
	var BATCH_UPDATEMODELVIEW
= 5;
	var BATCH_RENDERTOTEXTURE
= 6;
	var BATCH_CLEAR = 7;
	var BATCH_POINTS = 8;
	var BATCH_SETPROGRAM = 9;
	var
BATCH_SETPROGRAMPARAMETERS =
10;
	var BATCH_SETTEXTURE1 =
11;
	function GLWrap_(gl,
isMobile)
	{
		this.isIE =
/msie/i.test(navigator.userAgen
t) ||
/trident/i.test(navigator.userA
gent);
		this.width = 0;
	// not yet known, wait for
call to setSize()
		this.height = 0;
		this.cam =
vec3.create([0, 0, 100]);
		// camera position
		this.look =
vec3.create([0, 0, 0]);
			// lookat
position
		this.up =
vec3.create([0, 1, 0]);
			// up vector
		this.worldScale =
vec3.create([1, 1, 1]);
	// world scaling factor
		this.matP =
mat4.create();
			// perspective
matrix
		this.matMV =
mat4.create();
			// model view
matrix
		this.lastMV =
mat4.create();
		this.currentMV =
mat4.create();
		this.gl = gl;
		this.initState();
	};
	GLWrap_.prototype.initStat
e = function ()
	{
		var gl = this.gl;
		var i, len;
		this.lastOpacity =
1;
		this.lastTexture0 =
null;		// last bound
to TEXTURE0
		this.lastTexture1 =
null;		// last bound
to TEXTURE1
		this.currentOpacity
= 1;
		gl.clearColor(0, 0,
0, 0);

	gl.clear(gl.COLOR_BUFFER_B
IT);
		gl.enable(gl.BLEND);
```

```
        gl.blendFunc(gl.ONE,
gl.ONE_MINUS_SRC_ALPHA);

     gl.disable(gl.CULL_FACE);

     gl.disable(gl.DEPTH_TEST);
           this.maxTextureSize
=
gl.getParameter(gl.MAX_TEXTURE_
SIZE);
           this.lastSrcBlend =
gl.ONE;
           this.lastDestBlend =
gl.ONE_MINUS_SRC_ALPHA;
           this.pointBuffer =
gl.createBuffer();

     gl.bindBuffer(gl.ARRAY_BUF
FER, this.pointBuffer);
           this.vertexBuffers =
new Array(MULTI_BUFFERS);
           this.texcoordBuffers
= new Array(MULTI_BUFFERS);
           for (i = 0; i <
MULTI_BUFFERS; i++)
           {

     this.vertexBuffers[i] =
gl.createBuffer();

     gl.bindBuffer(gl.ARRAY_BUF
FER, this.vertexBuffers[i]);

     this.texcoordBuffers[i] =
gl.createBuffer();

     gl.bindBuffer(gl.ARRAY_BUF
FER, this.texcoordBuffers[i]);
           }
           this.curBuffer = 0;
           this.indexBuffer =
gl.createBuffer();

     gl.bindBuffer(gl.ELEMENT_A
RRAY_BUFFER, this.indexBuffer);
           this.vertexData =
new Float32Array(MAX_VERTICES *
2);
           this.texcoordData =
new Float32Array(MAX_VERTICES *
2);
           this.pointData = new
Float32Array(MAX_POINTS * 4);
           var indexData = new
Uint16Array(MAX_INDICES);
           i = 0, len =
MAX_INDICES;
           var fv = 0;
           while (i < len)
           {
                 indexData[i++]
= fv;      // top left
                 indexData[i++]
= fv + 1;  // top right
                 indexData[i++]
= fv + 2;  // bottom right
(first tri)
                 indexData[i++]
= fv;      // top left
                 indexData[i++]
= fv + 2;  // bottom right
                 indexData[i++]
= fv + 3;  // bottom left
                 fv += 4;
           }

     gl.bufferData(gl.ELEMENT_A
RRAY_BUFFER, indexData,
gl.STATIC_DRAW);
           this.vertexPtr = 0;
           this.pointPtr = 0;
           var fsSource,
vsSource;
           this.shaderPrograms
= [];
           fsSource = [
                 "varying
mediump vec2 vTex;",
                 "uniform lowp
float opacity;",
                 "uniform lowp
sampler2D samplerFront;",
                 "void
main(void) {",
                 "
     gl_FragColor =
texture2D(samplerFront,
vTex);",
                 "
     gl_FragColor *= opacity;",
                 "}"
           ].join("\n");
           vsSource = [
                 "attribute
highp vec2 aPos;",
                 "attribute
mediump vec2 aTex;",
                 "varying
mediump vec2 vTex;",
                 "uniform highp
mat4 matP;",
```

```
                "uniform highp
mat4 matMV;",
                "void
main(void) {",
                "
     gl_Position = matP * matMV
* vec4(aPos.x, aPos.y, 0.0,
1.0);",
                "    vTex =
aTex;",
                "}"
           ].join("\n");
           var shaderProg =
this.createShaderProgram({src:
fsSource}, vsSource,
"<default>");
;

     this.shaderPrograms.push(s
haderProg);           //
Default shader is always shader
0
           fsSource = [
                "uniform
mediump sampler2D
samplerFront;",
                "varying lowp
float opacity;",
                "void
main(void) {",
                "
     gl_FragColor =
texture2D(samplerFront,
gl_PointCoord);",
                "
     gl_FragColor *= opacity;",
                "}"
           ].join("\n");
           var pointVsSource =
[
                "attribute vec4
aPos;",
                "varying float
opacity;",
                "uniform mat4
matP;",
                "uniform mat4
matMV;",
                "void
main(void) {",
                "
     gl_Position = matP * matMV
* vec4(aPos.x, aPos.y, 0.0,
1.0);",
                "
     gl_PointSize = aPos.z;",
                "    opacity =
aPos.w;",
                "}"
           ].join("\n");
           shaderProg =
this.createShaderProgram({src:
fsSource}, pointVsSource,
"<point>");
;

     this.shaderPrograms.push(s
haderProg);           // Point
shader is always shader 1
           for (var shader_name
in cr.shaders)
           {
                if
(cr.shaders.hasOwnProperty(shad
er_name))

     this.shaderPrograms.push(t
his.createShaderProgram(cr.shad
ers[shader_name], vsSource,
shader_name));
           }

     gl.activeTexture(gl.TEXTUR
E0);

     gl.bindTexture(gl.TEXTURE_
2D, null);
           this.batch = [];
           this.batchPtr = 0;
           this.hasQuadBatchTop
= false;

     this.hasPointBatchTop =
false;
           this.lastProgram = -
1;                    // start
-1 so first switchProgram can
do work
           this.currentProgram
= -1;            // current
program during batch execution
           this.currentShader =
null;
           this.fbo =
gl.createFramebuffer();
           this.renderToTex =
null;
           this.tmpVec3 =
vec3.create([0, 0, 0]);
;
;
```

```
            var pointsizes =
gl.getParameter(gl.ALIASED_POIN
T_SIZE_RANGE);
            this.minPointSize =
pointsizes[0];
            this.maxPointSize =
pointsizes[1];
;

      this.switchProgram(0);
            cr.seal(this);
      };
      function
GLShaderProgram(gl,
shaderProgram, name)
      {
            this.gl = gl;
            this.shaderProgram =
shaderProgram;
            this.name = name;
            this.locAPos =
gl.getAttribLocation(shaderProg
ram, "aPos");
            this.locATex =
gl.getAttribLocation(shaderProg
ram, "aTex");
            this.locMatP =
gl.getUniformLocation(shaderPro
gram, "matP");
            this.locMatMV =
gl.getUniformLocation(shaderPro
gram, "matMV");
            this.locOpacity =
gl.getUniformLocation(shaderPro
gram, "opacity");
            this.locSamplerFront
=
gl.getUniformLocation(shaderPro
gram, "samplerFront");
            this.locSamplerBack
=
gl.getUniformLocation(shaderPro
gram, "samplerBack");
            this.locDestStart =
gl.getUniformLocation(shaderPro
gram, "destStart");
            this.locDestEnd =
gl.getUniformLocation(shaderPro
gram, "destEnd");
            this.locSeconds =
gl.getUniformLocation(shaderPro
gram, "seconds");
            this.locPixelWidth =
gl.getUniformLocation(shaderPro
gram, "pixelWidth");
            this.locPixelHeight
=
gl.getUniformLocation(shaderPro
gram, "pixelHeight");
            this.locLayerScale =
gl.getUniformLocation(shaderPro
gram, "layerScale");
            if (this.locOpacity)

      gl.uniform1f(this.locOpaci
ty, 1);
            if
(this.locSamplerFront)

      gl.uniform1i(this.locSampl
erFront, 0);
            if
(this.locSamplerBack)

      gl.uniform1i(this.locSampl
erBack, 1);
            if
(this.locDestStart)

      gl.uniform2f(this.locDestS
tart, 0.0, 0.0);
            if (this.locDestEnd)

      gl.uniform2f(this.locDestE
nd, 1.0, 1.0);
            this.hasCurrentMatMV
= false;         // matMV needs
updating
      };
      GLWrap_.prototype.createSh
aderProgram =
function(shaderEntry, vsSource,
name)
      {
            var gl = this.gl;
            var fragmentShader =
gl.createShader(gl.FRAGMENT_SHA
DER);

      gl.shaderSource(fragmentSh
ader, shaderEntry.src);

      gl.compileShader(fragmentS
hader);
            if
(!gl.getShaderParameter(fragmen
tShader, gl.COMPILE_STATUS))
            {
;
```

```
	gl.deleteShader(fragmentSh
ader);
		return null;
	}
	var vertexShader =
gl.createShader(gl.VERTEX_SHADE
R);

	gl.shaderSource(vertexShad
er, vsSource);

	gl.compileShader(vertexSha
der);
	if
(!gl.getShaderParameter(vertexS
hader, gl.COMPILE_STATUS))
	{
;

	gl.deleteShader(fragmentSh
ader);

	gl.deleteShader(vertexShad
er);
		return null;
	}
	var shaderProgram =
gl.createProgram();

	gl.attachShader(shaderProg
ram, fragmentShader);

	gl.attachShader(shaderProg
ram, vertexShader);

	gl.linkProgram(shaderProgr
am);
	if
(!gl.getProgramParameter(shader
Program, gl.LINK_STATUS))
	{
;

	gl.deleteShader(fragmentSh
ader);

	gl.deleteShader(vertexShad
er);

	gl.deleteProgram(shaderPro
gram);
		return null;
	}

	gl.useProgram(shaderProgra
m);
;

	gl.deleteShader(fragmentSh
ader);

	gl.deleteShader(vertexShad
er);
	var ret = new
GLShaderProgram(gl,
shaderProgram, name);

	ret.extendBoxHorizontal =
shaderEntry.extendBoxHorizontal
|| 0;

	ret.extendBoxVertical =
shaderEntry.extendBoxVertical
|| 0;
	ret.crossSampling =
!!shaderEntry.crossSampling;
	ret.animated =
!!shaderEntry.animated;
	ret.parameters =
shaderEntry.parameters || [];
	var i, len;
	for (i = 0, len =
ret.parameters.length; i < len;
i++)
	{

	ret.parameters[i][1] =
gl.getUniformLocation(shaderPro
gram, ret.parameters[i][0]);

	gl.uniform1f(ret.parameter
s[i][1], 0);
	}
	cr.seal(ret);
	return ret;
	};
	GLWrap_.prototype.getShade
rIndex = function(name_)
	{
	var i, len;
	for (i = 0, len =
this.shaderPrograms.length; i <
len; i++)
	{
		if
(this.shaderPrograms[i].name
=== name_)
		return i;
	}
```

```
		return -1;
	};
	GLWrap_.prototype.project
= function (x, y, out)
	{
		var mv = this.matMV;
		var proj =
this.matP;
		var fTempo = [0, 0,
0, 0, 0, 0, 0, 0];
		fTempo[0] =
mv[0]*x+mv[4]*y+mv[12];
		fTempo[1] =
mv[1]*x+mv[5]*y+mv[13];
		fTempo[2] =
mv[2]*x+mv[6]*y+mv[14];
		fTempo[3] =
mv[3]*x+mv[7]*y+mv[15];
		fTempo[4] =
proj[0]*fTempo[0]+proj[4]*fTemp
o[1]+proj[8]*fTempo[2]+proj[12]
*fTempo[3];
		fTempo[5] =
proj[1]*fTempo[0]+proj[5]*fTemp
o[1]+proj[9]*fTempo[2]+proj[13]
*fTempo[3];
		fTempo[6] =
proj[2]*fTempo[0]+proj[6]*fTemp
o[1]+proj[10]*fTempo[2]+proj[14
]*fTempo[3];
		fTempo[7] = -
fTempo[2];
		if(fTempo[7]===0.0)
	//The w value
			return;

	fTempo[7]=1.0/fTempo[7];

	fTempo[4]*=fTempo[7];

	fTempo[5]*=fTempo[7];

	fTempo[6]*=fTempo[7];

	out[0]=(fTempo[4]*0.5+0.5)
*this.width;

	out[1]=(fTempo[5]*0.5+0.5)
*this.height;
	};
	GLWrap_.prototype.setSize
= function(w, h, force)
	{
		if (this.width === w
&& this.height === h && !force)
			return;
		this.endBatch();
		this.width = w;
		this.height = h;
		this.gl.viewport(0,
0, w, h);
		mat4.perspective(45,
w / h, 1, 1000, this.matP);

	mat4.lookAt(this.cam,
this.look, this.up,
this.matMV);
		var tl = [0, 0];
		var br = [0, 0];
		this.project(0, 0,
tl);
		this.project(1, 1,
br);
		this.worldScale[0] =
1 / (br[0] - tl[0]);
		this.worldScale[1] =
-1 / (br[1] - tl[1]);
		var i, len, s;
		for (i = 0, len =
this.shaderPrograms.length; i <
len; i++)
		{
			s =
this.shaderPrograms[i];

	s.hasCurrentMatMV = false;
			if (s.locMatP)
			{

	this.gl.useProgram(s.shade
rProgram);

	this.gl.uniformMatrix4fv(s
.locMatP, false, this.matP);
			}
		}

	this.gl.useProgram(this.sh
aderPrograms[this.lastProgram].
shaderProgram);

	this.gl.bindTexture(this.g
l.TEXTURE_2D, null);

	this.gl.activeTexture(this
.gl.TEXTURE1);

	this.gl.bindTexture(this.g
l.TEXTURE_2D, null);

	this.gl.activeTexture(this
.gl.TEXTURE0);
```

```
		this.lastTexture0 =
null;
		this.lastTexture1 =
null;
	};
	GLWrap_.prototype.resetMod
elView = function ()
	{

	mat4.lookAt(this.cam,
this.look, this.up,
this.matMV);

	mat4.scale(this.matMV,
this.worldScale);
	};
	GLWrap_.prototype.translat
e = function (x, y)
	{
		if (x === 0 && y ===
0)
			return;
		this.tmpVec3[0] =
x;// * this.worldScale[0];
		this.tmpVec3[1] =
y;// * this.worldScale[1];
		this.tmpVec3[2] = 0;

	mat4.translate(this.matMV,
this.tmpVec3);
	};
	GLWrap_.prototype.scale =
function (x, y)
	{
		if (x === 1 && y ===
1)
			return;
		this.tmpVec3[0] = x;
		this.tmpVec3[1] = y;
		this.tmpVec3[2] = 1;

	mat4.scale(this.matMV,
this.tmpVec3);
	};
	GLWrap_.prototype.rotateZ
= function (a)
	{
		if (a === 0)
			return;

	mat4.rotateZ(this.matMV,
a);
	};
	GLWrap_.prototype.updateMo
delView = function()
	{
		var anydiff = false;
		for (var i = 0; i <
16; i++)
		{
			if
(this.lastMV[i] !==
this.matMV[i])
			{
				anydiff =
true;
				break;
			}
		}
		if (!anydiff)
			return;
		var b =
this.pushBatch();
		b.type =
BATCH_UPDATEMODELVIEW;
		if (b.mat4param)

	mat4.set(this.matMV,
b.mat4param);
		else
			b.mat4param =
mat4.create(this.matMV);
		mat4.set(this.matMV,
this.lastMV);
		this.hasQuadBatchTop
= false;

	this.hasPointBatchTop =
false;
	};
	/*
	var debugBatch = false;
	jQuery(document).mousedown
(
		function(info) {
			if (info.which
=== 2)

	debugBatch = true;
		}
	);
	*/
	function GLBatchJob(type_,
glwrap_)
	{
		this.type = type_;
		this.glwrap =
glwrap_;
		this.gl =
glwrap_.gl;
		this.opacityParam =
0;	// for setOpacity()
```

```
            this.startIndex = 0;
            // for quad()
            this.indexCount = 0;
            // "
            this.texParam =
null;       // for setTexture()
            this.mat4param =
null;       // for
updateModelView()
            this.shaderParams =
[];         // for user
parameters
            cr.seal(this);
      };
      GLBatchJob.prototype.doSet
Texture = function ()
      {

      this.gl.bindTexture(this.g
l.TEXTURE_2D, this.texParam);
      };
      GLBatchJob.prototype.doSet
Texture1 = function ()
      {
            var gl = this.gl;

      gl.activeTexture(gl.TEXTUR
E1);

      gl.bindTexture(gl.TEXTURE_
2D, this.texParam);

      gl.activeTexture(gl.TEXTUR
E0);
      };
      GLBatchJob.prototype.doSet
Opacity = function ()
      {
            var o =
this.opacityParam;
            var glwrap =
this.glwrap;

      glwrap.currentOpacity = o;
            var curProg =
glwrap.currentShader;
            if
(curProg.locOpacity)

      this.gl.uniform1f(curProg.
locOpacity, o);
      };
      GLBatchJob.prototype.doQua
d = function ()
      {

      this.gl.drawElements(this.
gl.TRIANGLES, this.indexCount,
this.gl.UNSIGNED_SHORT,
this.startIndex * 2);
      };
      GLBatchJob.prototype.doSet
Blend = function ()
      {

      this.gl.blendFunc(this.sta
rtIndex, this.indexCount);
      };
      GLBatchJob.prototype.doUpd
ateModelView = function ()
      {
            var i, len, s,
shaderPrograms =
this.glwrap.shaderPrograms,
currentProgram =
this.glwrap.currentProgram;
            for (i = 0, len =
shaderPrograms.length; i < len;
i++)
            {
                  s =
shaderPrograms[i];
                  if (i ===
currentProgram && s.locMatMV)
                  {

      this.gl.uniformMatrix4fv(s
.locMatMV, false,
this.mat4param);

      s.hasCurrentMatMV = true;
                  }
                  else

      s.hasCurrentMatMV = false;
            }

      mat4.set(this.mat4param,
this.glwrap.currentMV);
      };
      GLBatchJob.prototype.doRen
derToTexture = function ()
      {
            var gl = this.gl;
            var glwrap =
this.glwrap;
            if (this.texParam)
            {
                  if
(glwrap.lastTexture1 ===
this.texParam)
```

```
                {

     gl.activeTexture(gl.TEXTUR
E1);

     gl.bindTexture(gl.TEXTURE_
2D, null);

     glwrap.lastTexture1 =
null;

     gl.activeTexture(gl.TEXTUR
E0);
                }

     gl.bindFramebuffer(gl.FRAM
EBUFFER, glwrap.fbo);

     gl.framebufferTexture2D(gl
.FRAMEBUFFER,
gl.COLOR_ATTACHMENT0,
gl.TEXTURE_2D, this.texParam,
0);
           }
           else
           {

     gl.framebufferTexture2D(gl
.FRAMEBUFFER,
gl.COLOR_ATTACHMENT0,
gl.TEXTURE_2D, null, 0);

     gl.bindFramebuffer(gl.FRAM
EBUFFER, null);
           }
     };
     GLBatchJob.prototype.doCle
ar = function ()
     {
           var gl = this.gl;
           if (this.startIndex
=== 0)          // clear whole
surface
           {

     gl.clearColor(this.mat4par
am[0], this.mat4param[1],
this.mat4param[2],
this.mat4param[3]);

     gl.clear(gl.COLOR_BUFFER_B
IT);
           }
           else
                       // clear
rectangle
           {

     gl.enable(gl.SCISSOR_TEST)
;

     gl.scissor(this.mat4param[
0], this.mat4param[1],
this.mat4param[2],
this.mat4param[3]);

     gl.clearColor(0, 0, 0, 0);

     gl.clear(this.gl.COLOR_BUF
FER_BIT);

     gl.disable(gl.SCISSOR_TEST
);
           }
     };
     GLBatchJob.prototype.doPoi
nts = function ()
     {
           var gl = this.gl;
           var glwrap =
this.glwrap;
           var s =
glwrap.shaderPrograms[1];

     gl.useProgram(s.shaderProg
ram);
           if
(!s.hasCurrentMatMV &&
s.locMatMV)
           {

     gl.uniformMatrix4fv(s.locM
atMV, false, glwrap.currentMV);

     s.hasCurrentMatMV = true;
           }

     gl.enableVertexAttribArray
(s.locAPos);

     gl.bindBuffer(gl.ARRAY_BUF
FER, glwrap.pointBuffer);

     gl.vertexAttribPointer(s.l
ocAPos, 4, gl.FLOAT, false, 0,
0);

     gl.drawArrays(gl.POINTS,
this.startIndex / 4,
this.indexCount);
           s =
glwrap.currentShader;
```

```
	gl.useProgram(s.shaderProg
ram);
		if (s.locAPos >= 0)
		{

	gl.enableVertexAttribArray
(s.locAPos);

	gl.bindBuffer(gl.ARRAY_BUF
FER,
glwrap.vertexBuffers[glwrap.cur
Buffer]);

	gl.vertexAttribPointer(s.l
ocAPos, 2, gl.FLOAT, false, 0,
0);
		}
		if (s.locATex >= 0)
		{

	gl.enableVertexAttribArray
(s.locATex);

	gl.bindBuffer(gl.ARRAY_BUF
FER,
glwrap.texcoordBuffers[glwrap.c
urBuffer]);

	gl.vertexAttribPointer(s.l
ocATex, 2, gl.FLOAT, false, 0,
0);
		}
	};
	GLBatchJob.prototype.doSet
Program = function ()
	{
		var gl = this.gl;
		var glwrap =
this.glwrap;
		var s =
glwrap.shaderPrograms[this.star
tIndex];	// recycled
param to save memory

	glwrap.currentProgram =
this.startIndex;		//
current batch program
		glwrap.currentShader
= s;

	gl.useProgram(s.shaderProg
ram);
	// switch to
```

```
		if
(!s.hasCurrentMatMV &&
s.locMatMV)
		{

	gl.uniformMatrix4fv(s.locM
atMV, false, glwrap.currentMV);

	s.hasCurrentMatMV = true;
		}
		if (s.locOpacity)

	gl.uniform1f(s.locOpacity,
glwrap.currentOpacity);
		if (s.locAPos >= 0)
		{

	gl.enableVertexAttribArray
(s.locAPos);

	gl.bindBuffer(gl.ARRAY_BUF
FER,
glwrap.vertexBuffers[glwrap.cur
Buffer]);

	gl.vertexAttribPointer(s.l
ocAPos, 2, gl.FLOAT, false, 0,
0);
		}
		if (s.locATex >= 0)
		{

	gl.enableVertexAttribArray
(s.locATex);

	gl.bindBuffer(gl.ARRAY_BUF
FER,
glwrap.texcoordBuffers[glwrap.c
urBuffer]);

	gl.vertexAttribPointer(s.l
ocATex, 2, gl.FLOAT, false, 0,
0);
		}
	}
	GLBatchJob.prototype.doSet
ProgramParameters = function ()
	{
		var i, len, s =
this.glwrap.currentShader;
		var gl = this.gl;
		if (s.locSamplerBack
&& this.glwrap.lastTexture1 !==
this.texParam)
		{
```

```
	gl.activeTexture(gl.TEXTUR
E1);

	gl.bindTexture(gl.TEXTURE_
2D, this.texParam);

	this.glwrap.lastTexture1 =
this.texParam;

	gl.activeTexture(gl.TEXTUR
E0);
		}
		if (s.locPixelWidth)

	gl.uniform1f(s.locPixelWid
th, this.mat4param[0]);
		if
(s.locPixelHeight)

	gl.uniform1f(s.locPixelHei
ght, this.mat4param[1]);
		if (s.locDestStart)

	gl.uniform2f(s.locDestStar
t, this.mat4param[2],
this.mat4param[3]);
		if (s.locDestEnd)

	gl.uniform2f(s.locDestEnd,
this.mat4param[4],
this.mat4param[5]);
		if (s.locLayerScale)

	gl.uniform1f(s.locLayerSca
le, this.mat4param[6]);
		if (s.locSeconds)

	gl.uniform1f(s.locSeconds,
cr.performance_now() / 1000.0);
		if
(s.parameters.length)
		{
			for (i = 0, len
= s.parameters.length; i < len;
i++)
			{

	gl.uniform1f(s.parameters[
i][1], this.shaderParams[i]);
			}
		}
	};
	GLWrap_.prototype.pushBatc
h = function ()
	{
		if (this.batchPtr
=== this.batch.length)

	this.batch.push(new
GLBatchJob(BATCH_NULL, this));
		return
this.batch[this.batchPtr++];
	};
	GLWrap_.prototype.endBatch
= function ()
	{
		if (this.batchPtr
=== 0)
			return;
		if
(this.gl.isContextLost())
			return;
		var gl = this.gl;
		if (this.pointPtr >
0)
		{

	gl.bindBuffer(gl.ARRAY_BUF
FER, this.pointBuffer);

	gl.bufferData(gl.ARRAY_BUF
FER, this.pointData.subarray(0,
this.pointPtr),
gl.STREAM_DRAW);
			if (s &&
s.locAPos >= 0 && s.name ===
"<point>")

	gl.vertexAttribPointer(s.l
ocAPos, 4, gl.FLOAT, false, 0,
0);
		}
		if (this.vertexPtr >
0)
		{
			var s =
this.currentShader;

	gl.bindBuffer(gl.ARRAY_BUF
FER,
this.vertexBuffers[this.curBuff
er]);

	gl.bufferData(gl.ARRAY_BUF
FER,
this.vertexData.subarray(0,
this.vertexPtr),
gl.STREAM_DRAW);
			if (s &&
s.locAPos >= 0 && s.name !==
"<point>")
```

```
     gl.vertexAttribPointer(s.l
ocAPos, 2, gl.FLOAT, false, 0,
0);

     gl.bindBuffer(gl.ARRAY_BUF
FER,
this.texcoordBuffers[this.curBu
ffer]);

     gl.bufferData(gl.ARRAY_BUF
FER,
this.texcoordData.subarray(0,
this.vertexPtr),
gl.STREAM_DRAW);
                if (s &&
s.locATex >= 0 && s.name !==
"<point>")

     gl.vertexAttribPointer(s.l
ocATex, 2, gl.FLOAT, false, 0,
0);
           }
           var i, len, b;
           for (i = 0, len =
this.batchPtr; i < len; i++)
           {
                b =
this.batch[i];
                switch (b.type)
{
                case
BATCH_QUAD:

     b.doQuad();
                     break;
                case
BATCH_SETTEXTURE:

     b.doSetTexture();
                     break;
                case
BATCH_SETOPACITY:

     b.doSetOpacity();
                     break;
                case
BATCH_SETBLEND:

     b.doSetBlend();
                     break;
                case
BATCH_UPDATEMODELVIEW:

     b.doUpdateModelView();
                     break;
                case
BATCH_RENDERTOTEXTURE:

     b.doRenderToTexture();
                     break;
                case
BATCH_CLEAR:

     b.doClear();
                     break;
                case
BATCH_POINTS:

     b.doPoints();
                     break;
                case
BATCH_SETPROGRAM:

     b.doSetProgram();
                     break;
                case
BATCH_SETPROGRAMPARAMETERS:

     b.doSetProgramParameters()
;
                     break;
                case
BATCH_SETTEXTURE1:

     b.doSetTexture1();
                     break;
                }
           }
           this.batchPtr = 0;
           this.vertexPtr = 0;
           this.pointPtr = 0;
           this.hasQuadBatchTop
= false;

     this.hasPointBatchTop =
false;
           this.curBuffer++;
           if (this.curBuffer
>= MULTI_BUFFERS)
                this.curBuffer
= 0;
     };
     GLWrap_.prototype.setOpaci
ty = function (op)
     {
           if (op ===
this.lastOpacity)
                return;
           var b =
this.pushBatch();
```

```
		b.type =
BATCH_SETOPACITY;
		b.opacityParam = op;
		this.lastOpacity =
op;
		this.hasQuadBatchTop
= false;

	this.hasPointBatchTop =
false;
	};
	GLWrap_.prototype.setTextu
re = function (tex)
	{
		if (tex ===
this.lastTexture0)
			return;
;
		var b =
this.pushBatch();
		b.type =
BATCH_SETTEXTURE;
		b.texParam = tex;
		this.lastTexture0 =
tex;
		this.hasQuadBatchTop
= false;

	this.hasPointBatchTop =
false;
	};
	GLWrap_.prototype.setBlend
= function (s, d)
	{
		if (s ===
this.lastSrcBlend && d ===
this.lastDestBlend)
			return;
		var b =
this.pushBatch();
		b.type =
BATCH_SETBLEND;
		b.startIndex = s;
		// recycle params to
save memory
		b.indexCount = d;
		this.lastSrcBlend =
s;
		this.lastDestBlend =
d;
		this.hasQuadBatchTop
= false;

	this.hasPointBatchTop =
false;
	};
	GLWrap_.prototype.setAlpha
Blend = function ()
	{

	this.setBlend(this.gl.ONE,
this.gl.ONE_MINUS_SRC_ALPHA);
	};
	var LAST_VERTEX =
MAX_VERTICES * 2 - 8;
	GLWrap_.prototype.quad =
function(tlx, tly, trx, try_,
brx, bry, blx, bly)
	{
		if (this.vertexPtr
>= LAST_VERTEX)

	this.endBatch();
		var v =
this.vertexPtr;		//
vertex cursor
		var vd =
this.vertexData;	// vertex
data array
		var td =
this.texcoordData;		//
texture coord data array
		if
(this.hasQuadBatchTop)
		{

	this.batch[this.batchPtr -
1].indexCount += 6;
		}
		else
		{
			var b =
this.pushBatch();
			b.type =
BATCH_QUAD;
			b.startIndex =
(v / 4) * 3;
			b.indexCount =
6;

	this.hasQuadBatchTop =
true;

	this.hasPointBatchTop =
false;
		}
		vd[v] = tlx;
		td[v++] = 0;
		vd[v] = tly;
		td[v++] = 0;
		vd[v] = trx;
		td[v++] = 1;
```

```
            vd[v] = try_;
            td[v++] = 0;
            vd[v] = brx;
            td[v++] = 1;
            vd[v] = bry;
            td[v++] = 1;
            vd[v] = blx;
            td[v++] = 0;
            vd[v] = bly;
            td[v++] = 1;
            this.vertexPtr = v;
      };
      GLWrap_.prototype.quadTex
= function(tlx, tly, trx, try_,
brx, bry, blx, bly, rcTex)
      {
            if (this.vertexPtr
>= LAST_VERTEX)

      this.endBatch();
            var v =
this.vertexPtr;            //
vertex cursor
            var vd =
this.vertexData;      // vertex
data array
            var td =
this.texcoordData;         //
texture coord data array
            if
(this.hasQuadBatchTop)
            {

      this.batch[this.batchPtr -
1].indexCount += 6;
            }
            else
            {
                  var b =
this.pushBatch();
                  b.type =
BATCH_QUAD;
                  b.startIndex =
(v / 4) * 3;
                  b.indexCount =
6;

      this.hasQuadBatchTop =
true;

      this.hasPointBatchTop =
false;
            }
            var rc_left =
rcTex.left;
            var rc_top =
rcTex.top;
            var rc_right =
rcTex.right;
            var rc_bottom =
rcTex.bottom;
            vd[v] = tlx;
            td[v++] = rc_left;
            vd[v] = tly;
            td[v++] = rc_top;
            vd[v] = trx;
            td[v++] = rc_right;
            vd[v] = try_;
            td[v++] = rc_top;
            vd[v] = brx;
            td[v++] = rc_right;
            vd[v] = bry;
            td[v++] = rc_bottom;
            vd[v] = blx;
            td[v++] = rc_left;
            vd[v] = bly;
            td[v++] = rc_bottom;
            this.vertexPtr = v;
      };
      GLWrap_.prototype.quadTexU
V = function(tlx, tly, trx,
try_, brx, bry, blx, bly, tlu,
tlv, tru, trv, bru, brv, blu,
blv)
      {
            if (this.vertexPtr
>= LAST_VERTEX)

      this.endBatch();
            var v =
this.vertexPtr;            //
vertex cursor
            var vd =
this.vertexData;      // vertex
data array
            var td =
this.texcoordData;         //
texture coord data array
            if
(this.hasQuadBatchTop)
            {

      this.batch[this.batchPtr -
1].indexCount += 6;
            }
            else
            {
                  var b =
this.pushBatch();
                  b.type =
BATCH_QUAD;
```

```
                b.startIndex =
(v / 4) * 3;
                b.indexCount =
6;

     this.hasQuadBatchTop =
true;

     this.hasPointBatchTop =
false;
          }
          vd[v] = tlx;
          td[v++] = tlu;
          vd[v] = tly;
          td[v++] = tlv;
          vd[v] = trx;
          td[v++] = tru;
          vd[v] = try_;
          td[v++] = trv;
          vd[v] = brx;
          td[v++] = bru;
          vd[v] = bry;
          td[v++] = brv;
          vd[v] = blx;
          td[v++] = blu;
          vd[v] = bly;
          td[v++] = blv;
          this.vertexPtr = v;
     };
     var LAST_POINT =
MAX_POINTS - 4;
     GLWrap_.prototype.point =
function(x_, y_, size_,
opacity_)
     {
          if (this.pointPtr >=
LAST_POINT)

     this.endBatch();
          var p =
this.pointPtr;              //
point cursor
          var pd =
this.pointData;       // point
data array
          if
(this.hasPointBatchTop)
          {

     this.batch[this.batchPtr -
1].indexCount++;
          }
          else
          {
                var b =
this.pushBatch();
                b.type =
BATCH_POINTS;
                b.startIndex =
p;
                b.indexCount =
1;

     this.hasPointBatchTop =
true;

     this.hasQuadBatchTop =
false;
          }
          pd[p++] = x_;
          pd[p++] = y_;
          pd[p++] = size_;
          pd[p++] = opacity_;
          this.pointPtr = p;
     };
     GLWrap_.prototype.switchPr
ogram = function (progIndex)
     {
          if (this.lastProgram
=== progIndex)
                return;
          // no change
          var shaderProg =
this.shaderPrograms[progIndex];
          if (!shaderProg)
          {
                if
(this.lastProgram === 0)
                      return;

                // already on
default shader
                progIndex = 0;
                shaderProg =
this.shaderPrograms[0];
          }
          var b =
this.pushBatch();
          b.type =
BATCH_SETPROGRAM;
          b.startIndex =
progIndex;
          this.lastProgram =
progIndex;
          this.hasQuadBatchTop
= false;

     this.hasPointBatchTop =
false;
     };
     GLWrap_.prototype.programU
sesDest = function (progIndex)
```

```
	{
		var s =
this.shaderPrograms[progIndex];
		return
!!(s.locDestStart ||
s.locDestEnd);
	};
	GLWrap_.prototype.programU
sesCrossSampling = function
(progIndex)
	{
		var s =
this.shaderPrograms[progIndex];
		return
!!(s.locDestStart ||
s.locDestEnd ||
s.crossSampling);
	};
	GLWrap_.prototype.programE
xtendsBox = function
(progIndex)
	{
		var s =
this.shaderPrograms[progIndex];
		return
s.extendBoxHorizontal !== 0 ||
s.extendBoxVertical !== 0;
	};
	GLWrap_.prototype.getProgr
amBoxExtendHorizontal =
function (progIndex)
	{
		return
this.shaderPrograms[progIndex].
extendBoxHorizontal;
	};
	GLWrap_.prototype.getProgr
amBoxExtendVertical = function
(progIndex)
	{
		return
this.shaderPrograms[progIndex].
extendBoxVertical;
	};
	GLWrap_.prototype.getProgr
amParameterType = function
(progIndex, paramIndex)
	{
		return
this.shaderPrograms[progIndex].
parameters[paramIndex][2];
	};
	GLWrap_.prototype.programI
sAnimated = function
(progIndex)
	{
		return
this.shaderPrograms[progIndex].
animated;
	};
	GLWrap_.prototype.setProgr
amParameters = function
(backTex, pixelWidth,
pixelHeight, destStartX,
destStartY, destEndX, destEndY,
layerScale, params)
	{
		var i, len, s =
this.shaderPrograms[this.lastPr
ogram];
		if (s.locPixelWidth
|| s.locPixelHeight ||
s.locSeconds ||
s.locSamplerBack ||
			s.locDestStart
|| s.locDestEnd ||
s.locLayerScale ||
params.length)
		{
			var b =
this.pushBatch();
			b.type =
BATCH_SETPROGRAMPARAMETERS;
			if
(b.mat4param)

	mat4.set(this.matMV,
b.mat4param);
			else

	b.mat4param =
mat4.create();
			b.mat4param[0]
= pixelWidth;
			b.mat4param[1]
= pixelHeight;
			b.mat4param[2]
= destStartX;
			b.mat4param[3]
= destStartY;
			b.mat4param[4]
= destEndX;
			b.mat4param[5]
= destEndY;
			b.mat4param[6]
= layerScale;
			if
(s.locSamplerBack)
			{
;

	b.texParam = backTex;
```

```
                }
                else
     b.texParam = null;
                if
(params.length)
                {

     b.shaderParams.length =
params.length;
                        for (i =
0, len = params.length; i <
len; i++)

     b.shaderParams[i] =
params[i];
                }

     this.hasQuadBatchTop =
false;

     this.hasPointBatchTop =
false;
           }
     };
     GLWrap_.prototype.clear =
function (r, g, b_, a)
     {
           var b =
this.pushBatch();
           b.type =
BATCH_CLEAR;
           b.startIndex = 0;
                              //
clear all mode
           if (!b.mat4param)
                b.mat4param =
mat4.create();
           b.mat4param[0] = r;
           b.mat4param[1] = g;
           b.mat4param[2] = b_;
           b.mat4param[3] = a;
           this.hasQuadBatchTop
= false;

     this.hasPointBatchTop =
false;
     };
     GLWrap_.prototype.clearRec
t = function (x, y, w, h)
     {
           var b =
this.pushBatch();
           b.type =
BATCH_CLEAR;
           b.startIndex = 1;
                              //
clear rect mode
           if (!b.mat4param)
                b.mat4param =
mat4.create();
           b.mat4param[0] = x;
           b.mat4param[1] = y;
           b.mat4param[2] = w;
           b.mat4param[3] = h;
           this.hasQuadBatchTop
= false;

     this.hasPointBatchTop =
false;
     };
     GLWrap_.prototype.present
= function ()
     {
           this.endBatch();
           this.gl.flush();
           /*
           if (debugBatch)
           {
;
                debugBatch =
false;
           }
           */
     };
     function
nextHighestPowerOfTwo(x) {
           --x;
           for (var i = 1; i <
32; i <<= 1) {
                x = x | x >> i;
           }
           return x + 1;
     }
     var all_textures = [];
     var textures_by_src = {};
     var BF_RGBA8 = 0;
     var BF_RGB8 = 1;
     var BF_RGBA4 = 2;
     var BF_RGB5_A1 = 3;
     var BF_RGB565 = 4;
     GLWrap_.prototype.loadText
ure = function (img, tiling,
linearsampling, pixelformat,
tiletype)
     {
           tiling = !!tiling;
           linearsampling =
!!linearsampling;
           var tex_key =
img.src + "," + tiling + "," +
```

```
linearsampling + (tiling ? (","
+ tiletype) : ""));
            var webGL_texture =
null;
            if (typeof img.src
!== "undefined" &&
textures_by_src.hasOwnProperty(
tex_key))
            {
                  webGL_texture =
textures_by_src[tex_key];

      webGL_texture.c2refcount++
;
                  return
webGL_texture;
            }
            this.endBatch();
;
            var gl = this.gl;
            var isPOT =
(cr.isPOT(img.width) &&
cr.isPOT(img.height));
            webGL_texture =
gl.createTexture();

      gl.bindTexture(gl.TEXTURE_
2D, webGL_texture);

      gl.pixelStorei(gl["UNPACK_
PREMULTIPLY_ALPHA_WEBGL"],
true);
            var internalformat =
gl.RGBA;
            var format =
gl.RGBA;
            var type =
gl.UNSIGNED_BYTE;
            if (pixelformat &&
!this.isIE)
            {
                  switch
(pixelformat) {
                  case BF_RGB8:

      internalformat = gl.RGB;
                        format =
gl.RGB;
                        break;
                  case BF_RGBA4:
                        type =
gl.UNSIGNED_SHORT_4_4_4_4;
                        break;
                  case
BF_RGB5_A1:
                        type =
gl.UNSIGNED_SHORT_5_5_5_1;
                        break;
                  case BF_RGB565:

      internalformat = gl.RGB;
                        format =
gl.RGB;
                        type =
gl.UNSIGNED_SHORT_5_6_5;
                        break;
                  }
            }
            if (!isPOT &&
tiling)
            {
                  var canvas =
document.createElement("canvas"
);
                  canvas.width =
cr.nextHighestPowerOfTwo(img.wi
dth);
                  canvas.height =
cr.nextHighestPowerOfTwo(img.he
ight);
                  var ctx =
canvas.getContext("2d");

      ctx.drawImage(img,

        0, 0, img.width,
img.height,

        0, 0, canvas.width,
canvas.height);

      gl.texImage2D(gl.TEXTURE_2
D, 0, internalformat, format,
type, canvas);
            }
            else

      gl.texImage2D(gl.TEXTURE_2
D, 0, internalformat, format,
type, img);
            if (tiling)
            {
                  if (tiletype
=== "repeat-x")
                  {

      gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_WRAP_S,
gl.REPEAT);

      gl.texParameteri(gl.TEXTUR
```

```
E_2D, gl.TEXTURE_WRAP_T,
gl.CLAMP_TO_EDGE);
                }
                else if
(tiletype === "repeat-y")
                {

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_WRAP_S,
gl.CLAMP_TO_EDGE);

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_WRAP_T,
gl.REPEAT);
                }
                else
                {

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_WRAP_S,
gl.REPEAT);

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_WRAP_T,
gl.REPEAT);
                }
           }
           else
           {

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_WRAP_S,
gl.CLAMP_TO_EDGE);

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_WRAP_T,
gl.CLAMP_TO_EDGE);
           }
           if (linearsampling)
           {

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_MAG_FILTER,
gl.LINEAR);
                if (isPOT)
                {

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_MIN_FILTER,
gl.LINEAR_MIPMAP_LINEAR);

     gl.generateMipmap(gl.TEXTU
RE_2D);
                }
                else

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_MIN_FILTER,
gl.LINEAR);
           }
           else
           {

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_MAG_FILTER,
gl.NEAREST);

     gl.texParameteri(gl.TEXTUR
E_2D, gl.TEXTURE_MIN_FILTER,
gl.NEAREST);
           }

     gl.bindTexture(gl.TEXTURE_
2D, null);
           this.lastTexture0 =
null;

     webGL_texture.c2width =
img.width;

     webGL_texture.c2height =
img.height;

     webGL_texture.c2refcount =
1;

     webGL_texture.c2texkey =
tex_key;

     all_textures.push(webGL_te
xture);

     textures_by_src[tex_key] =
webGL_texture;
           return
webGL_texture;
     };
     GLWrap_.prototype.createEm
ptyTexture = function (w, h,
linearsampling, _16bit, tiling)
     {
           this.endBatch();
           var gl = this.gl;
           if (this.isIE)
                 _16bit = false;
           var webGL_texture =
gl.createTexture();

     gl.bindTexture(gl.TEXTURE_
2D, webGL_texture);
```

```
	gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, w, h, 0, gl.RGBA, _16bit ? gl.UNSIGNED_SHORT_4_4_4_4 : gl.UNSIGNED_BYTE, null);
		if (tiling)
		{

	gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_S, gl.REPEAT);

	gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_T, gl.REPEAT);
		}
		else
		{

	gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_S, gl.CLAMP_TO_EDGE);

	gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_WRAP_T, gl.CLAMP_TO_EDGE);
		}

	gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER, linearsampling ? gl.LINEAR : gl.NEAREST);

	gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, linearsampling ? gl.LINEAR : gl.NEAREST);

	gl.bindTexture(gl.TEXTURE_2D, null);
		this.lastTexture0 = null;

	webGL_texture.c2width = w;

	webGL_texture.c2height = h;

	all_textures.push(webGL_texture);
		return webGL_texture;
	};
	GLWrap_.prototype.videoToTexture = function (video_, texture_, _16bit)
	{
		this.endBatch();
		var gl = this.gl;
		if (this.isIE)
			_16bit = false;

	gl.bindTexture(gl.TEXTURE_2D, texture_);

	gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, _16bit ? gl.UNSIGNED_SHORT_4_4_4_4 : gl.UNSIGNED_BYTE, video_);

	gl.bindTexture(gl.TEXTURE_2D, null);
		this.lastTexture0 = null;
	};
	GLWrap_.prototype.deleteTexture = function (tex)
	{
		if (!tex)
			return;
		if (typeof tex.c2refcount !== "undefined" && tex.c2refcount > 1)
		{
			tex.c2refcount--;
			return;
		}
		this.endBatch();
		if (tex === this.lastTexture0)
		{

	this.gl.bindTexture(this.gl.TEXTURE_2D, null);

	this.lastTexture0 = null;
		}
		if (tex === this.lastTexture1)
		{

	this.gl.activeTexture(this.gl.TEXTURE1);

	this.gl.bindTexture(this.gl.TEXTURE_2D, null);
```

```
	this.gl.activeTexture(this
.gl.TEXTURE0);

	this.lastTexture1 = null;
		}

	cr.arrayFindRemove(all_tex
tures, tex);
		if (typeof
tex.c2texkey !== "undefined")
			delete
textures_by_src[tex.c2texkey];

	this.gl.deleteTexture(tex)
;
	};
	GLWrap_.prototype.estimate
VRAM = function ()
	{
		var total =
this.width * this.height * 4 *
2;
		var i, len, t;
		for (i = 0, len =
all_textures.length; i < len;
i++)
		{
			t =
all_textures[i];
			total +=
(t.c2width * t.c2height * 4);
		}
		return total;
	};
	GLWrap_.prototype.textureC
ount = function ()
	{
		return
all_textures.length;
	};
	GLWrap_.prototype.setRende
ringToTexture = function (tex)
	{
		if (tex ===
this.renderToTex)
			return;
;
		var b =
this.pushBatch();
		b.type =
BATCH_RENDERTOTEXTURE;
		b.texParam = tex;
		this.renderToTex =
tex;
		this.hasQuadBatchTop
= false;

	this.hasPointBatchTop =
false;
	};
	cr.GLWrap = GLWrap_;
}());
;
(function()
{
	function Runtime(canvas)
	{
		if (!canvas ||
(!canvas.getContext &&
!canvas["dc"]))
			return;
		if
(canvas["c2runtime"])
			return;
		else

	canvas["c2runtime"] =
this;
		var self = this;
		this.isCrosswalk =
/crosswalk/i.test(navigator.use
rAgent) ||
/xwalk/i.test(navigator.userAge
nt) || !!(typeof
window["c2isCrosswalk"] !==
"undefined" &&
window["c2isCrosswalk"]);
		this.isPhoneGap =
(!this.isCrosswalk && (typeof
window["device"] !==
"undefined" && (typeof
window["device"]["cordova"] !==
"undefined" || typeof
window["device"]["phonegap"]
!== "undefined")));
		this.isDirectCanvas
= !!canvas["dc"];
		this.isAppMobi =
(typeof window["AppMobi"] !==
"undefined" ||
this.isDirectCanvas);
		this.isCocoonJs =
!!window["c2cocoonjs"];
		if (this.isCocoonJs)
		{

	CocoonJS["App"]["onSuspend
ed"].addEventListener(function(
) {
```

```
	self["setSuspended"](true);
		});

	CocoonJS["App"]["onActivated"].addEventListener(function() {

	self["setSuspended"](false);
		});
		}
		this.isDomFree = this.isDirectCanvas || this.isCocoonJs;
		this.isTizen = /tizen/i.test(navigator.userAgent);
		this.isAndroid = /android/i.test(navigator.userAgent) && !this.isTizen;
	// tizen says "like Android"
		this.isIE = /msie/i.test(navigator.userAgent) || /trident/i.test(navigator.userAgent);
		this.isiPhone = /iphone/i.test(navigator.userAgent) || /ipod/i.test(navigator.userAgent);	// treat ipod as an iphone
		this.isiPad = /ipad/i.test(navigator.userAgent);
		this.isiOS = this.isiPhone || this.isiPad;
		this.isChrome = /chrome/i.test(navigator.userAgent) || /chromium/i.test(navigator.userAgent);
		this.isAmazonWebApp = /amazonwebappplatform/i.test(navigator.userAgent);
		this.isFirefox = /firefox/i.test(navigator.userAgent);
		this.isSafari = !this.isChrome && /safari/i.test(navigator.userAgent);	// Chrome includes Safari in UA
		this.isWindows = /windows/i.test(navigator.userAgent);
		this.isNodeWebkit = (typeof window["c2nodewebkit"] !== "undefined" || /nodewebkit/i.test(navigator.userAgent));
		this.isArcade = (typeof window["is_scirra_arcade"] !== "undefined");
		this.isWindows8App = !!(typeof window["c2isWindows8"] !== "undefined" && window["c2isWindows8"]);
		this.isWindowsPhone8 = !!(typeof window["c2isWindowsPhone8"] !== "undefined" && window["c2isWindowsPhone8"]);
		this.isBlackberry10 = !!(typeof window["c2isBlackberry10"] !== "undefined" && window["c2isBlackberry10"]);

	this.isAndroidStockBrowser = (this.isAndroid && !this.isChrome && !this.isFirefox && !this.isAmazonWebApp && !this.isDomFree);

	this.devicePixelRatio = 1;
		this.isMobile = (this.isPhoneGap || this.isCrosswalk || this.isAppMobi || this.isCocoonJs || this.isAndroid || this.isiOS || this.isWindowsPhone8 || this.isBlackberry10 || this.isTizen);
		if (!this.isMobile)
			this.isMobile = /(blackberry|bb10|playbook|palm|symbian|nokia|windows\s+ce|phone|mobile|tablet)/i.test(navigator.userAgent);
		if (typeof cr_is_preview !== "undefined" && !this.isNodeWebkit && (window.location.search === "?nw" ||
```

```
/nodewebkit/i.test(navigator.us
erAgent)))
           {

     this.isNodeWebkit = true;
           }
           this.isDebug =
(typeof cr_is_preview !==
"undefined" &&
window.location.search.indexOf(
"debug") > -1)
           this.canvas =
canvas;
           this.canvasdiv =
document.getElementById("c2canv
asdiv");
           this.gl = null;
           this.glwrap = null;
           this.ctx = null;

     this.fullscreenOldMarginCs
s = "";

     this.firstInFullscreen =
false;
           this.oldWidth = 0;
           // for restoring
non-fullscreen canvas after
fullscreen
           this.oldHeight = 0;

     this.canvas.oncontextmenu
= function (e) { if
(e.preventDefault)
e.preventDefault(); return
false; };

     this.canvas.onselectstart
= function (e) { if
(e.preventDefault)
e.preventDefault(); return
false; };
           if
(this.isDirectCanvas)

     window["c2runtime"] =
this;
           if
(this.isNodeWebkit)
           {

     window.ondragover =
function(e) {
e.preventDefault(); return
false; };
                window.ondrop =
function(e) {
e.preventDefault(); return
false; };

     require("nw.gui")["App"]["
clearCache"]();
           }
           this.width =
canvas.width;
           this.height =
canvas.height;
           this.draw_width =
this.width;
           this.draw_height =
this.height;
           this.cssWidth =
this.width;
           this.cssHeight =
this.height;
           this.lastWindowWidth
= window.innerWidth;

     this.lastWindowHeight =
window.innerHeight;
           this.redraw = true;
           this.isSuspended =
false;
           if (!Date.now) {
             Date.now =
function now() {
                return +new
Date();
             };
           }
           this.plugins = [];
           this.types = {};
           this.types_by_index
= [];
           this.behaviors = [];
           this.layouts = {};

     this.layouts_by_index =
[];
           this.eventsheets =
{};

     this.eventsheets_by_index
= [];

     this.wait_for_textures =
[];        // for blocking
until textures loaded

     this.triggers_to_postinit
= [];
```

```
            this.all_global_vars
= [];
            this.all_local_vars
= [];
            this.solidBehavior =
null;

      this.jumpthruBehavior =
null;
            this.deathRow = new
cr.ObjectSet();

      this.isInClearDeathRow =
false;
            this.isInOnDestroy =
0;                          //
needs to support recursion so
increments and decrements and
is true if > 0
            this.isRunningEvents
= false;
            this.createRow = [];
            this.isLoadingState
= false;
            this.saveToSlot =
"";
            this.loadFromSlot =
"";
            this.loadFromJson =
"";
            this.lastSaveJson =
"";

      this.signalledContinuousPr
eview = false;
            this.suspendDrawing
= false;        // for hiding
display until continuous
preview loads
            this.dt = 0;
        this.dt1 = 0;
            this.logictime = 0;
                  // used to
calculate CPUUtilisation
            this.cpuutilisation
= 0;
            this.zeroDtCount =
0;
        this.timescale = 1.0;
        this.kahanTime = new
cr.KahanAdder();
            this.last_tick_time
= 0;
            this.measuring_dt =
true;
            this.fps = 0;
            this.last_fps_time =
0;
            this.tickcount = 0;
            this.execcount = 0;
            this.framecount = 0;
// for fps
            this.objectcount =
0;
            this.changelayout =
null;

      this.destroycallbacks =
[];
            this.event_stack =
[];

      this.event_stack_index = -
1;
            this.localvar_stack
= [[]];

      this.localvar_stack_index
= 0;
            this.trigger_depth =
0;         // recursion depth
for triggers

      this.pushEventStack(null);
            this.loop_stack =
[];

      this.loop_stack_index = -
1;
            this.next_uid = 0;
            this.next_puid = 0;
            // permanent unique
ids

      this.layout_first_tick =
true;
            this.family_count =
0;
            this.suspend_events
= [];
            this.raf_id = 0;
            this.timeout_id = 0;
            this.isloading =
true;
            this.loadingprogress
= 0;

      this.isNodeFullscreen =
false;
            this.stackLocalCount
= 0;  // number of stack-based
local vars for recursion
```

```
     this.halfFramerateMode =
false;
           this.lastRafTime =
0;         // time of last
requestAnimationFrame call
           this.ranLastRaf =
false;     // false if last
requestAnimationFrame was
skipped for half framerate mode
           this.had_a_click =
false;

     this.isInUserInputEvent =
false;
        this.objects_to_tick =
new cr.ObjectSet();

     this.objects_to_tick2 =
new cr.ObjectSet();

     this.registered_collisions
= [];
           this.temp_poly = new
cr.CollisionPoly([]);
           this.temp_poly2 =
new cr.CollisionPoly([]);
           this.allGroups = [];
                      // array
of all event groups
        this.activeGroups = {};
                      // event
group activation states
           this.cndsBySid = {};
           this.actsBySid = {};
           this.varsBySid = {};
           this.blocksBySid =
{};
           this.running_layout
= null;               //
currently running layout
           this.layer_canvas =
null;            // for layers
"render-to-texture"
           this.layer_ctx =
null;
           this.layer_tex =
null;
           this.layout_tex =
null;
           this.layout_canvas =
null;
           this.layout_ctx =
null;

     this.is_WebGL_context_lost
= false;

     this.uses_background_blend
ing = false;     // if any
shader uses background
blending, so entire layout
renders to texture
           this.fx_tex = [null,
null];

     this.fullscreen_scaling =
0;
           this.files_subfolder
= "";            // path with
project files
           this.objectsByUid =
{};                    // maps
every in-use UID (as a string)
to its instance
           this.loaderlogo =
null;
           this.snapshotCanvas
= null;
           this.snapshotData =
"";
           this.load();
           this.isRetina =
(!this.isDomFree &&
this.useHighDpi &&
!this.isAndroidStockBrowser);

     this.devicePixelRatio =
(this.isRetina ?
(window["devicePixelRatio"] ||
window["webkitDevicePixelRatio"
] ||
window["mozDevicePixelRatio"]
|| window["msDevicePixelRatio"]
|| 1) : 1);

     this.ClearDeathRow();
           var attribs;
           var alpha_canvas =
this.alphaBackground &&
!(this.isNodeWebkit ||
this.isWindows8App ||
this.isWindowsPhone8);
           if
(this.fullscreen_mode > 0)

     this["setSize"](window.inn
erWidth, window.innerHeight,
true);
           try {
```

```
                if
(this.enableWebGL &&
(this.isCocoonJs ||
!this.isDomFree))
                {
                      attribs =
{
     "alpha": alpha_canvas,

     "depth": false,

     "antialias": false,

     "failIfMajorPerformanceCav
eat": true
                      };
                      this.gl =
(canvas.getContext("webgl",
attribs) ||
canvas.getContext("experimental
-webgl", attribs));
                }
           }
           catch (e) {
           }
           if (this.gl)
           {
;
                if
(!this.isDomFree)
                {

     this.overlay_canvas =
document.createElement("canvas"
);

     jQuery(this.overlay_canvas
).appendTo(this.canvas.parentNo
de);

     this.overlay_canvas.oncont
extmenu = function (e) { return
false; };

     this.overlay_canvas.onsele
ctstart = function (e) { return
false; };

     this.overlay_canvas.width
= this.cssWidth;

     this.overlay_canvas.height
= this.cssHeight;

     jQuery(this.overlay_canvas
).css({"width": this.cssWidth +
"px",


           "height":
this.cssHeight + "px"});

     this.positionOverlayCanvas
();

     this.overlay_ctx =
this.overlay_canvas.getContext(
"2d");
                }
                this.glwrap =
new cr.GLWrap(this.gl,
this.isMobile);

     this.glwrap.setSize(canvas
.width, canvas.height);
                this.ctx =
null;

     this.canvas.addEventListen
er("webglcontextlost", function
(ev) {

     ev.preventDefault();

     self.onContextLost();

     console.log("[Construct 2]
WebGL context lost");

     window["cr_setSuspended"](
true);          // stop
rendering
                }, false);

     this.canvas.addEventListen
er("webglcontextrestored",
function (ev) {

     self.glwrap.initState();

     self.glwrap.setSize(self.g
lwrap.width,
self.glwrap.height, true);

     self.layer_tex = null;

     self.layout_tex = null;

     self.fx_tex[0] = null;

     self.fx_tex[1] = null;
```

```
self.onContextRestored();

self.redraw = true;

console.log("[Construct 2] WebGL context restored");

window["cr_setSuspended"](false);          // resume rendering
                }, false);
                var i, len, j, lenj, k, lenk, t, s, l, y;
                for (i = 0, len = this.types_by_index.length; i < len; i++)
                {
                        t = this.types_by_index[i];
                        for (j = 0, lenj = t.effect_types.length; j < lenj; j++)
                        {
                                s = t.effect_types[j];

s.shaderindex = this.glwrap.getShaderIndex(s.id);

this.uses_background_blending = this.uses_background_blending || this.glwrap.programUsesDest(s.shaderindex);
                        }
                }
                for (i = 0, len = this.layouts_by_index.length; i < len; i++)
                {
                        l = this.layouts_by_index[i];
                        for (j = 0, lenj = l.effect_types.length; j < lenj; j++)
                        {
                                s = l.effect_types[j];

s.shaderindex = this.glwrap.getShaderIndex(s.id);
                        }
                        for (j = 0, lenj = l.layers.length; j < lenj; j++)
                        {
                                y = l.layers[j];
                                for (k = 0, lenk = y.effect_types.length; k < lenk; k++)
                                {

s = y.effect_types[k];

s.shaderindex = this.glwrap.getShaderIndex(s.id);

this.uses_background_blending = this.uses_background_blending || this.glwrap.programUsesDest(s.shaderindex);
                                }
                        }
                }
        }
        else
        {
                if (this.fullscreen_mode > 0 && this.isDirectCanvas)
                {
;

this.canvas = null;

document.oncontextmenu = function (e) { return false; };

document.onselectstart = function (e) { return false; };
                        this.ctx = AppMobi["canvas"]["getContext"]("2d");
                        try {

this.ctx["samplingMode"] = this.linearSampling ? "smooth" : "sharp";
```

```
                                    hingEnabled"] =
      this.ctx["globalScale"] =     this.linearSampling;
1;
                                    this.ctx["mozImageSmoothin
      this.ctx["HTML5Compatibili    gEnabled"] =
tyMode"] = true;                    this.linearSampling;

      this.ctx["imageSmoothingEn    this.ctx["msImageSmoothing
abled"] = this.linearSampling;      Enabled"] =
                        }           this.linearSampling;
catch(e){}
                        if          this.ctx["imageSmoothingEn
(this.width !== 0 &&                abled"] = this.linearSampling;
this.height !== 0)                                  }
                        {
                                    this.overlay_canvas =
      this.ctx.width =              null;
this.width;
                                    this.overlay_ctx = null;
      this.ctx.height =                       }
this.height;                                this.tickFunc =
                        }           function () { self.tick(); };
                  }                         if (window !=
                  if (!this.ctx)    window.top && !this.isDomFree
                  {                 && !this.isWindows8App)
;                                           {
                        if
(this.isCocoonJs)                   document.addEventListener(
                        {           "mousedown", function () {

      attribs = {                   window.focus();
                                                  }, true);
      "antialias":
!!this.linearSampling,              document.addEventListener(
                                    "touchstart", function () {
      "alpha": alpha_canvas
                              };    window.focus();
                                                  }, true);
      this.ctx =                                }
canvas.getContext("2d",                     if (typeof
attribs);                           cr_is_preview !== "undefined")
                        }                         {
                        else                      if
                        {           (this.isCocoonJs)

      attribs = {                   console.log("[Construct 2]
                                    In preview-over-wifi via
      "alpha": alpha_canvas         CocoonJS mode");
                              }                   if
                                    (window.location.search.indexOf
      this.ctx =                    ("continuous") > -1)
canvas.getContext("2d",                           {
attribs);
                        }           cr.logexport("Reloading
                                    for continuous preview");
      this.ctx["webkitImageSmoot
```

```
	this.loadFromSlot =
"__c2_continuouspreview";

	this.suspendDrawing =
true;
		}
		if
(this.pauseOnBlur &&
!this.isMobile)
		{

	jQuery(window).focus(funct
ion ()
			{

	self["setSuspended"](false
);
			});

	jQuery(window).blur(functi
on ()
			{

	self["setSuspended"](true)
;
			});
		}
	}
	if
(this.fullscreen_mode === 0 &&
this.isRetina &&
this.devicePixelRatio > 1)
	{

	this["setSize"](this.origi
nal_width,
this.original_height, true);
	}

	this.tryLockOrientation();
	this.go();
	// run loading screen
	this.extra = {};
	cr.seal(this);
	};
	var webkitRepaintFlag =
false;
	Runtime.prototype["setSize
"] = function (w, h, force)
	{
	var offx = 0, offy =
0;
	var neww = 0, newh =
0, intscale = 0;
	if
(this.lastWindowWidth === w &&
this.lastWindowHeight === h &&
!force)
		return;
	this.lastWindowWidth
= w;

	this.lastWindowHeight = h;
	var mode =
this.fullscreen_mode;
	var orig_aspect,
cur_aspect;
	var isfullscreen =
(document["mozFullScreen"] ||
document["webkitIsFullScreen"]
||
!!document["msFullscreenElement
"] || document["fullScreen"] ||
this.isNodeFullscreen);
	if (!isfullscreen &&
this.fullscreen_mode === 0 &&
!force)
		return;
	// ignore size
events when not fullscreen and
not using a fullscreen-in-
browser mode
	if (isfullscreen &&
this.fullscreen_scaling > 0)
		mode =
this.fullscreen_scaling;
	if (mode >= 4)
	{
		orig_aspect =
this.original_width /
this.original_height;
		cur_aspect = w
/ h;
		if (cur_aspect
> orig_aspect)
		{
			neww = h
* orig_aspect;
			if (mode
=== 5)	// integer scaling
			{

	intscale = neww /
this.original_width;
				if
(intscale > 1)

	intscale =
Math.floor(intscale);
```

```
                        else
if (intscale < 1)

     intscale = 1 / Math.ceil(1
/ intscale);
                        neww
= this.original_width *
intscale;
                        newh
= this.original_height *
intscale;
                        offx
= (w - neww) / 2;
                        offy
= (h - newh) / 2;
                        w =
neww;
                        h =
newh;
                    }
                    else
                    {
                        offx
= (w - neww) / 2;
                        w =
neww;
                    }
                }
                else
                {
                    newh = w
/ orig_aspect;
                    if (mode
=== 5)     // integer scaling
                    {

     intscale = newh /
this.original_height;
                        if
(intscale > 1)

     intscale =
Math.floor(intscale);
                        else
if (intscale < 1)

     intscale = 1 / Math.ceil(1
/ intscale);
                        neww
= this.original_width *
intscale;
                        newh
= this.original_height *
intscale;
                        offx
= (w - neww) / 2;
                        offy
= (h - newh) / 2;
                        w =
neww;
                        h =
newh;
                    }
                    else
                    {
                        offy
= (h - newh) / 2;
                        h =
newh;
                    }
                }
                if
(isfullscreen &&
!this.isNodeWebkit)
                {
                    offx = 0;
                    offy = 0;
                }
                offx =
Math.floor(offx);
                offy =
Math.floor(offy);
                w =
Math.floor(w);
                h =
Math.floor(h);
            }
            else if
(this.isNodeWebkit &&
this.isNodeFullscreen &&
this.fullscreen_mode_set === 0)
            {
                offx =
Math.floor((w -
this.original_width) / 2);
                offy =
Math.floor((h -
this.original_height) / 2);
                w =
this.original_width;
                h =
this.original_height;
            }
            if (mode < 2)

     this.aspect_scale =
this.devicePixelRatio;
            if (this.isRetina &&
this.isiPad &&
this.devicePixelRatio > 1)  //
don't apply to iPad 1-2
            {
```

```
                if (w >= 1024)
                      w = 1023;
           // 2046 retina
pixels
                if (h >= 1024)
                      h = 1023;
           }
           var multiplier =
this.devicePixelRatio;
           this.cssWidth = w;
           this.cssHeight = h;
           this.width =
Math.round(w * multiplier);
           this.height =
Math.round(h * multiplier);
           this.redraw = true;
           if
(this.wantFullscreenScalingQual
ity)
           {
                this.draw_width
= this.width;

     this.draw_height =
this.height;

     this.fullscreenScalingQual
ity = true;
           }
           else
           {
                if ((this.width
< this.original_width &&
this.height <
this.original_height) || mode
=== 1)
                {

     this.draw_width =
this.width;

     this.draw_height =
this.height;

     this.fullscreenScalingQual
ity = true;
                }
                else
                {

     this.draw_width =
this.original_width;

     this.draw_height =
this.original_height;

     this.fullscreenScalingQual
ity = false;
                      /*var
orig_aspect =
this.original_width /
this.original_height;
                      var
cur_aspect = this.width /
this.height;
                      if
((this.fullscreen_mode !== 2 &&
cur_aspect > orig_aspect) ||
(this.fullscreen_mode === 2 &&
cur_aspect < orig_aspect))

     this.aspect_scale =
this.height /
this.original_height;
                      else

     this.aspect_scale =
this.width /
this.original_width;*/
                      if (mode
=== 2)          // scale inner
                      {

     orig_aspect =
this.original_width /
this.original_height;

     cur_aspect =
this.lastWindowWidth /
this.lastWindowHeight;
                            if
(cur_aspect < orig_aspect)

     this.draw_width =
this.draw_height * cur_aspect;
                            else
if (cur_aspect > orig_aspect)

     this.draw_height =
this.draw_width / cur_aspect;
                      }
                      else if
(mode === 3)
                      {

     orig_aspect =
this.original_width /
this.original_height;

     cur_aspect =
```

```
this.lastWindowWidth /
this.lastWindowHeight;
                            if
(cur_aspect > orig_aspect)

     this.draw_width =
this.draw_height * cur_aspect;
                            else
if (cur_aspect < orig_aspect)

     this.draw_height =
this.draw_width / cur_aspect;
                        }
                 }
           }
           if (this.canvasdiv
&& !this.isDomFree)
           {

     jQuery(this.canvasdiv).css
({"width": w + "px",


     "height": h + "px",


     "margin-left": offx,


     "margin-top": offy});
                 if (typeof
cr_is_preview !== "undefined")
                 {

     jQuery("#borderwrap").css(
{"width": w + "px",


     "height": h + "px"});
                 }
           }
           if (this.canvas)
           {

     this.canvas.width =
Math.round(w * multiplier);

     this.canvas.height =
Math.round(h * multiplier);
                 if
(this.isRetina)
                 {

     jQuery(this.canvas).css({"
width": w + "px",

     "height": h + "px"});
                 }
           }
           if
(this.overlay_canvas)
           {

     this.overlay_canvas.width
= w;

     this.overlay_canvas.height
= h;

     jQuery(this.overlay_canvas
).css({"width": w + "px",


     "height": h + "px"});
           }
           if (this.glwrap)
           {

     this.glwrap.setSize(Math.r
ound(w * multiplier),
Math.round(h * multiplier));
           }
           if
(this.isDirectCanvas &&
this.ctx)
           {
                 this.ctx.width
= w;
                 this.ctx.height
= h;
           }
           if (this.ctx)
           {

     this.ctx["webkitImageSmoot
hingEnabled"] =
this.linearSampling;

     this.ctx["mozImageSmoothin
gEnabled"] =
this.linearSampling;

     this.ctx["msImageSmoothing
Enabled"] =
this.linearSampling;

     this.ctx["imageSmoothingEn
abled"] = this.linearSampling;
           }
```

```
	this.tryLockOrientation();
	};
	Runtime.prototype.tryLockO
rientation = function ()
	{
		if
(!this.autoLockOrientation ||
this.orientations === 0)
			return;
		var orientation =
"portrait";
		if
(this.orientations === 2)
			orientation =
"landscape";
		if
(screen["lockOrientation"])

	screen["lockOrientation"](
orientation);
		else if
(screen["webkitLockOrientation"
])

	screen["webkitLockOrientat
ion"](orientation);
		else if
(screen["mozLockOrientation"])

	screen["mozLockOrientation
"](orientation);
		else if
(screen["msLockOrientation"])

	screen["msLockOrientation"
](orientation);
	};
	Runtime.prototype.onContex
tLost = function ()
	{

	this.is_WebGL_context_lost
= true;
		var i, len, t;
		for (i = 0, len =
this.types_by_index.length; i <
len; i++)
		{
			t =
this.types_by_index[i];
			if
(t.onLostWebGLContext)

	t.onLostWebGLContext();
		}
	};
	Runtime.prototype.onContex
tRestored = function ()
	{

	this.is_WebGL_context_lost
= false;
		var i, len, t;
		for (i = 0, len =
this.types_by_index.length; i <
len; i++)
		{
			t =
this.types_by_index[i];
			if
(t.onRestoreWebGLContext)

	t.onRestoreWebGLContext();
		}
	};
	Runtime.prototype.position
OverlayCanvas = function()
	{
		if (this.isDomFree)
			return;
		var isfullscreen =
(document["mozFullScreen"] ||
document["webkitIsFullScreen"]
|| document["fullScreen"] ||
!!document["msFullscreenElement
"] || this.isNodeFullscreen);
		var overlay_position
= isfullscreen ?
jQuery(this.canvas).offset() :
jQuery(this.canvas).position();

	overlay_position.position
= "absolute";

	jQuery(this.overlay_canvas
).css(overlay_position);
	};
	var caf =
window["cancelAnimationFrame"]
||

window["mozCancelAnimationFrame
"]	||

window["webkitCancelAnimationFr
ame"] ||

window["msCancelAnimationFrame"
]	||
```

```
window["oCancelAnimationFrame"]
;
     Runtime.prototype["setSusp
ended"] = function (s)
     {
          var i, len;
          if (s &&
!this.isSuspended)
          {

     cr.logexport("[Construct
2] Suspending");

     this.isSuspended = true;
               // next tick
will be last
               if (this.raf_id
!== 0 && caf)         // note:
CocoonJS does not implement
cancelAnimationFrame

     caf(this.raf_id);
               if
(this.timeout_id !== 0)

     clearTimeout(this.timeout_
id);
               for (i = 0, len
= this.suspend_events.length; i
< len; i++)

     this.suspend_events[i](tru
e);
          }
          else if (!s &&
this.isSuspended)
          {

     cr.logexport("[Construct
2] Resuming");

     this.isSuspended = false;

     this.last_tick_time =
cr.performance_now(); // ensure
first tick is a zero-dt one

     this.last_fps_time =
cr.performance_now(); // reset
FPS counter
               this.framecount
= 0;
               this.logictime
= 0;
               for (i = 0, len
= this.suspend_events.length; i
< len; i++)

     this.suspend_events[i](fal
se);
               this.tick();

     // kick off runtime again
          }
     };
     Runtime.prototype.addSuspe
ndCallback = function (f)
     {

     this.suspend_events.push(f
);
     };
     Runtime.prototype.load =
function ()
     {
;
          var pm =
cr.getProjectModel();
          this.name = pm[0];
          this.first_layout =
pm[1];
          this.fullscreen_mode
= pm[11];  // 0 = off, 1 =
crop, 2 = scale inner, 3 =
scale outer, 4 = letterbox
scale, 5 = integer letterbox
scale

     this.fullscreen_mode_set =
pm[11];
          this.original_width
= pm[9];
          this.original_height
= pm[10];

     this.parallax_x_origin =
this.original_width / 2;

     this.parallax_y_origin =
this.original_height / 2;
          if (this.isDomFree
&& (pm[11] >= 4 || pm[11] ===
0))
          {

     cr.logexport("[Construct
2] Letterbox scale fullscreen
modes are not supported on this
platform - falling back to
'Scale outer'");
```

```
	this.fullscreen_mode = 3;

	this.fullscreen_mode_set = 
3;
		}

	this.uses_loader_layout = 
pm[17];
		this.loaderstyle = 
pm[18];
		if (this.loaderstyle 
=== 0)
		{
			this.loaderlogo 
= new Image();

	this.loaderlogo.src = 
"loading-logo.png";
		}
		this.next_uid = 
pm[20];
		this.system = new 
cr.system_object(this);
		var i, len, j, lenj, 
k, lenk, idstr, m, b, t, f;
		var plugin, 
plugin_ctor;
		for (i = 0, len = 
pm[2].length; i < len; i++)
		{
			m = pm[2][i];
;

	cr.add_common_aces(m);
			plugin = new 
m[0](this);

	plugin.singleglobal = 
m[1];
			plugin.is_world 
= m[2];

	plugin.must_predraw = 
m[9];
			if 
(plugin.onCreate)

	plugin.onCreate();  // 
opportunity to override default 
ACEs

	cr.seal(plugin);

	this.plugins.push(plugin);
		}
		pm = 
cr.getProjectModel();
		for (i = 0, len = 
pm[3].length; i < len; i++)
		{
			m = pm[3][i];
			plugin_ctor = 
m[1];
;
			plugin = null;
			for (j = 0, 
lenj = this.plugins.length; j < 
lenj; j++)
			{
				if 
(this.plugins[j] instanceof 
plugin_ctor)
				{

	plugin = this.plugins[j];

	break;
				}
			}
;
;
			var type_inst = 
new plugin.Type(plugin);
;
			type_inst.name 
= m[0];

	type_inst.is_family = 
m[2];

	type_inst.instvar_sids = 
m[3].slice(0);

	type_inst.vars_count = 
m[3].length;

	type_inst.behs_count = 
m[4];

	type_inst.fx_count = m[5];
			type_inst.sid = 
m[11];
			if 
(type_inst.is_family)
			{

	type_inst.members = [];
				// types 
in this family
```

```
type_inst.family_index =
this.family_count++;

type_inst.families = null;
            }
            else
            {

type_inst.members = null;

type_inst.family_index = -
1;

type_inst.families = [];
            // families
this type belongs to
            }

type_inst.family_var_map =
null;

type_inst.family_beh_map =
null;

type_inst.family_fx_map =
null;

type_inst.is_contained =
false;

type_inst.container =
null;
            if (m[6])
            {

type_inst.texture_file =
m[6][0];

type_inst.texture_filesize
= m[6][1];

type_inst.texture_pixelfor
mat = m[6][2];
            }
            else
            {

type_inst.texture_file =
null;

type_inst.texture_filesize
= 0;

type_inst.texture_pixelfor
mat = 0;        // rgba8
            }
            if (m[7])
            {

type_inst.animations =
m[7];
            }
            else
            {

type_inst.animations =
null;
            }
            type_inst.index
= i;
// save index in to types array
in type

type_inst.instances = [];
// all instances of this type

type_inst.deadCache = [];

      // destroyed
instances to recycle next
create

type_inst.solstack = [new
cr.selection(type_inst)]; //
initialise SOL stack with one
empty SOL

type_inst.cur_sol = 0;

type_inst.default_instance
= null;

type_inst.stale_iids =
true;

type_inst.updateIIDs =
cr.type_updateIIDs;

type_inst.getFirstPicked =
cr.type_getFirstPicked;

type_inst.getPairedInstanc
e = cr.type_getPairedInstance;

type_inst.getCurrentSol =
cr.type_getCurrentSol;

type_inst.pushCleanSol =
cr.type_pushCleanSol;
```

```
type_inst.pushCopySol = cr.type_pushCopySol;

type_inst.popSol = cr.type_popSol;

type_inst.getBehaviorByName = cr.type_getBehaviorByName;

type_inst.getBehaviorIndexByName = cr.type_getBehaviorIndexByName;

type_inst.getEffectIndexByName = cr.type_getEffectIndexByName;

type_inst.applySolToContainer = cr.type_applySolToContainer;

type_inst.getInstanceByIID = cr.type_getInstanceByIID;

type_inst.collision_grid = new cr.SparseGrid(this.original_width, this.original_height);

type_inst.any_bbox_changed = true;

type_inst.any_instance_parallaxed = false;
type_inst.extra = {};

type_inst.toString = cr.type_toString;

type_inst.behaviors = [];
for (j = 0, lenj = m[8].length; j < lenj; j++)
{
b = m[8][j];
var behavior_ctor = b[1];
var behavior_plugin = null;
for (k = 0, lenk = this.behaviors.length; k < lenk; k++)
{
if (this.behaviors[k] instanceof behavior_ctor)
{

behavior_plugin = this.behaviors[k];

break;
}
}
if (!behavior_plugin)
{

behavior_plugin = new behavior_ctor(this);

behavior_plugin.my_types = [];
// types using this behavior

behavior_plugin.my_instances = new cr.ObjectSet();   // instances of this behavior
if (behavior_plugin.onCreate)

behavior_plugin.onCreate();

cr.seal(behavior_plugin);

this.behaviors.push(behavior_plugin);
if (cr.behaviors.solid && behavior_plugin instanceof cr.behaviors.solid)

this.solidBehavior = behavior_plugin;
if (cr.behaviors.jumpthru && behavior_plugin instanceof cr.behaviors.jumpthru)

this.jumpthruBehavior = behavior_plugin;
}
if (behavior_plugin.my_types.indexOf(type_inst) === -1)
```

```
behavior_plugin.my_types.push(type_inst);
var behavior_type = new behavior_plugin.Type(behavior_plugin, type_inst);

behavior_type.name = b[0];

behavior_type.sid = b[2];

behavior_type.onCreate();

cr.seal(behavior_type);

type_inst.behaviors.push(behavior_type);
}

type_inst.global = m[9];

type_inst.isOnLoaderLayout = m[10];

type_inst.effect_types = [];
for (j = 0, lenj = m[12].length; j < lenj; j++)
{

type_inst.effect_types.push({
id: m[12][j][0],

name: m[12][j][1],

shaderindex: -1,

active: true,

index: j
});
}

type_inst.tile_poly_data = m[13];
if (!this.uses_loader_layout || type_inst.is_family || type_inst.isOnLoaderLayout || !plugin.is_world)
{

type_inst.onCreate();

cr.seal(type_inst);
}
if (type_inst.name)

this.types[type_inst.name] = type_inst;

this.types_by_index.push(type_inst);
if (plugin.singleglobal)
{
var instance = new plugin.Instance(type_inst);

instance.uid = this.next_uid++;

instance.puid = this.next_puid++;

instance.iid = 0;

instance.get_iid = cr.inst_get_iid;

instance.toString = cr.inst_toString;

instance.properties = m[14];

instance.onCreate();

cr.seal(instance);

type_inst.instances.push(instance);

this.objectsByUid[instance.uid.toString()] = instance;
}
}
for (i = 0, len = pm[4].length; i < len; i++)
{
var familydata = pm[4][i];
var familytype =
```

```
this.types_by_index[familydata[
0]];
                var
familymember;
                for (j = 1,
lenj = familydata.length; j <
lenj; j++)
                {

     familymember =
this.types_by_index[familydata[
j]];

     familymember.families.push
(familytype);

     familytype.members.push(fa
milymember);
                }
           }
           for (i = 0, len =
pm[23].length; i < len; i++)
           {
                var
containerdata = pm[23][i];
                var
containertypes = [];
                for (j = 0,
lenj = containerdata.length; j
< lenj; j++)

     containertypes.push(this.t
ypes_by_index[containerdata[j]]
);
                for (j = 0,
lenj = containertypes.length; j
< lenj; j++)
                {

     containertypes[j].is_conta
ined = true;

     containertypes[j].containe
r = containertypes;
                }
           }
           if
(this.family_count > 0)
           {
                for (i = 0, len
= this.types_by_index.length; i
< len; i++)
                {
                     t =
this.types_by_index[i];
                     if
(t.is_family ||
!t.families.length)

     continue;

     t.family_var_map = new
Array(this.family_count);

     t.family_beh_map = new
Array(this.family_count);

     t.family_fx_map = new
Array(this.family_count);
                     var
all_fx = [];
                     var
varsum = 0;
                     var
behsum = 0;
                     var fxsum
= 0;
                     for (j =
0, lenj = t.families.length; j
< lenj; j++)
                     {
                          f =
t.families[j];

     t.family_var_map[f.family_
index] = varsum;

     varsum += f.vars_count;

     t.family_beh_map[f.family_
index] = behsum;

     behsum += f.behs_count;

     t.family_fx_map[f.family_i
ndex] = fxsum;

     fxsum += f.fx_count;
                          for
(k = 0, lenk =
f.effect_types.length; k <
lenk; k++)

     all_fx.push(cr.shallowCopy
({}, f.effect_types[k]));
                     }

     t.effect_types =
all_fx.concat(t.effect_types);
                     for (j =
0, lenj =
```

```
t.effect_types.length; j <
lenj; j++)

     t.effect_types[j].index =
j;
                }
           }
           for (i = 0, len =
pm[5].length; i < len; i++)
           {
                m = pm[5][i];
                var layout =
new cr.layout(this, m);

     cr.seal(layout);

     this.layouts[layout.name]
= layout;

     this.layouts_by_index.push
(layout);
           }
           for (i = 0, len =
pm[6].length; i < len; i++)
           {
                m = pm[6][i];
                var sheet = new
cr.eventsheet(this, m);
                cr.seal(sheet);

     this.eventsheets[sheet.nam
e] = sheet;

     this.eventsheets_by_index.
push(sheet);
           }
           for (i = 0, len =
this.eventsheets_by_index.lengt
h; i < len; i++)

     this.eventsheets_by_index[
i].postInit();
           for (i = 0, len =
this.triggers_to_postinit.lengt
h; i < len; i++)

     this.triggers_to_postinit[
i].postInit();

     this.triggers_to_postinit.
length = 0;
           this.files_subfolder
= pm[7];
           this.pixel_rounding
= pm[8];
           this.aspect_scale =
1.0;
           this.enableWebGL =
pm[12];
           this.linearSampling
= pm[13];
           this.alphaBackground
= pm[14];
           this.versionstr =
pm[15];
           this.useHighDpi =
pm[16];
           this.orientations =
pm[19];          // 0 = any, 1 =
portrait, 2 = landscape

     this.autoLockOrientation =
(this.orientations > 0);
           this.pauseOnBlur =
pm[21];

     this.wantFullscreenScaling
Quality = pm[22];          //
false = low quality, true =
high quality

     this.fullscreenScalingQual
ity =
this.wantFullscreenScalingQuali
ty;
           this.start_time =
Date.now();
     };
     Runtime.prototype.findWait
ingTexture = function (src_)
     {
           var i, len;
           for (i = 0, len =
this.wait_for_textures.length;
i < len; i++)
           {
                if
(this.wait_for_textures[i].cr_s
rc === src_)
                     return
this.wait_for_textures[i];
           }
           return null;
     };
     Runtime.prototype.areAllTe
xturesLoaded = function ()
     {
           var totalsize = 0;
           var completedsize =
0;
           var ret = true;
```

```
		var i, len;
		for (i = 0, len = this.wait_for_textures.length; i < len; i++)
		{
			var filesize = this.wait_for_textures[i].cr_filesize;
			if (!filesize || filesize <= 0)
				filesize = 50000;
			totalsize += filesize;
			if (this.wait_for_textures[i].complete || this.wait_for_textures[i]["loaded"])
	completedsize += filesize;
			else
				ret = false;	// not all textures loaded
		}
		if (totalsize == 0)
			this.progress = 0;
		else
			this.progress = (completedsize / totalsize);
		return ret;
	};
	Runtime.prototype.go = function ()
	{
		if (!this.ctx && !this.glwrap)
			return;
		var ctx = this.ctx || this.overlay_ctx;
		if (this.overlay_canvas)
	this.positionOverlayCanvas();
		this.progress = 0;
		this.last_progress = -1;
		if (this.areAllTexturesLoaded())
	this.go_textures_done();
		else
		{
			var ms_elapsed = Date.now() - this.start_time;
			if (ctx)
			{
				var overlay_width = this.width;
				var overlay_height = this.height;
				var multiplier = this.devicePixelRatio;
				if (this.overlay_canvas)
				{
	overlay_width = this.cssWidth;
	overlay_height = this.cssHeight;
	multiplier = 1;
				}
				if (this.loaderstyle !== 3 && ms_elapsed >= 500 && this.last_progress != this.progress)
				{
	ctx.clearRect(0, 0, overlay_width, overlay_height);
					var mx = overlay_width / 2;
					var my = overlay_height / 2;
					var haslogo = (this.loaderstyle === 0 && this.loaderlogo.complete);
					var hlw = 40 * multiplier;
					var hlh = 0;
					var logowidth = 80 * multiplier;
					var logoheight;
					if (haslogo)
					{
	logowidth = this.loaderlogo.width * multiplier;
	logoheight =
```

```
this.loaderlogo.height *
multiplier;

      hlw = logowidth / 2;

      hlh = logoheight / 2;

      ctx.drawImage(this.loaderl
ogo, cr.floor(mx - hlw),
cr.floor(my - hlh), logowidth,
logoheight);
                              }
                              if
(this.loaderstyle <= 1)
                              {

      my += hlh + (haslogo ? 12
* multiplier : 0);

      mx -= hlw;

      mx = cr.floor(mx) + 0.5;

      my = cr.floor(my) + 0.5;

      ctx.fillStyle =
"DodgerBlue";

      ctx.fillRect(mx, my,
Math.floor(logowidth *
this.progress), 6 *
multiplier);

      ctx.strokeStyle = "black";

      ctx.strokeRect(mx, my,
logowidth, 6 * multiplier);

      ctx.strokeStyle = "white";

      ctx.strokeRect(mx - 1 *
multiplier, my - 1 *
multiplier, logowidth + 2 *
multiplier, 8 * multiplier);
                              }
                              else
if (this.loaderstyle === 2)
                              {

      ctx.font = "12pt Arial";

      ctx.fillStyle = "#999";

      ctx.textBaseLine =
"middle";

      var percent_text =
Math.round(this.progress * 100)
+ "%";

      var text_dim =
ctx.measureText ?
ctx.measureText(percent_text) :
null;

      var text_width = text_dim
? text_dim.width : 0;

      ctx.fillText(percent_text,
mx - (text_width / 2), my);
                              }
                        }

      this.last_progress =
this.progress;
                  }

      setTimeout((function
(self) { return function () {
self.go(); }; })(this), 100);
            }
      };
      Runtime.prototype.go_textu
res_done = function ()
      {
            if
(this.overlay_canvas)
            {

      this.canvas.parentNode.rem
oveChild(this.overlay_canvas);

      this.overlay_ctx = null;

      this.overlay_canvas =
null;
            }
            this.start_time =
Date.now();
            this.last_fps_time =
cr.performance_now();       //
for counting framerate
            var i, len, t;
            if
(this.uses_loader_layout)
            {
                  for (i = 0, len
= this.types_by_index.length; i
< len; i++)
                  {
```

```
                        t =
this.types_by_index[i];
                        if
(!t.is_family &&
!t.isOnLoaderLayout &&
t.plugin.is_world)
                        {

     t.onCreate();

     cr.seal(t);
                        }
                   }
              }
              else
                   this.isloading
= false;
              for (i = 0, len =
this.layouts_by_index.length; i
< len; i++)
              {

     this.layouts_by_index[i].c
reateGlobalNonWorlds();
              }
              if
(this.fullscreen_mode >= 2)
              {
                   var orig_aspect
= this.original_width /
this.original_height;
                   var cur_aspect
= this.width / this.height;
                   if
((this.fullscreen_mode !== 2 &&
cur_aspect > orig_aspect) ||
(this.fullscreen_mode === 2 &&
cur_aspect < orig_aspect))

     this.aspect_scale =
this.height /
this.original_height;
                   else

     this.aspect_scale =
this.width /
this.original_width;
              }
              if
(this.first_layout)

     this.layouts[this.first_la
yout].startRunning();
              else

     this.layouts_by_index[0].s
tartRunning();
;
              if
(!this.uses_loader_layout)
              {

     this.loadingprogress = 1;

     this.trigger(cr.system_obj
ect.prototype.cnds.OnLoadFinish
ed, null);
              }
              if
(navigator["splashscreen"] &&
navigator["splashscreen"]["hide
"])

     navigator["splashscreen"][
"hide"]();
              this.tick();
              if
(this.isDirectCanvas)

     AppMobi["webview"]["execut
e"]("onGameReady();");
     };
     var raf =
window["requestAnimationFrame"]
||

window["mozRequestAnimationFram
e"]    ||

window["webkitRequestAnimationF
rame"] ||

window["msRequestAnimationFrame
"]     ||

window["oRequestAnimationFrame"
];
     Runtime.prototype.tick =
function ()
     {
              if
(!this.running_layout)
                   return;
              var logic_start =
cr.performance_now();
              if
(this.halfFramerateMode &&
this.ranLastRaf)
              {
```

```
                if (logic_start
- this.lastRafTime < 29)
                {

     this.ranLastRaf = false;

     this.lastRafTime =
logic_start;
                      if (raf)

     this.raf_id =
raf(this.tickFunc,
this.canvas);
                      else //
no idea if this works without
raf/hi res timers but let's
hope for the best

     this.timeout_id =
setTimeout(this.tickFunc,
this.isMobile ? 1 : 16);
                      return;
           // skipped this
frame
                }
           }
           this.ranLastRaf =
true;
           this.lastRafTime =
logic_start;
           var fsmode =
this.fullscreen_mode;
           var isfullscreen =
(document["mozFullScreen"] ||
document["webkitIsFullScreen"]
|| document["fullScreen"] ||
!!document["msFullscreenElement
"]);
           if ((isfullscreen ||
this.isNodeFullscreen) &&
this.fullscreen_scaling > 0)
                fsmode =
this.fullscreen_scaling;
           if (fsmode > 0 &&
!this.isiPhone)
           {
                var curwidth =
window.innerWidth;
                var curheight =
window.innerHeight;
                if
(this.lastWindowWidth !==
curwidth ||
this.lastWindowHeight !==
curheight)
                {

     this["setSize"](curwidth,
curheight);
                }
           }
           if (!this.isDomFree)
           {
                if
(isfullscreen)
                {
                     if
(!this.firstInFullscreen)
                     {

     this.fullscreenOldMarginCs
s =
jQuery(this.canvas).css("margin
") || "0";

     this.firstInFullscreen =
true;
                     }
                     if
(!this.isChrome &&
!this.isNodeWebkit)
                     {

     jQuery(this.canvas).css({

     "margin-left": "" +
Math.floor((screen.width -
(this.width /
this.devicePixelRatio)) / 2) +
"px",

     "margin-top": "" +
Math.floor((screen.height -
(this.height /
this.devicePixelRatio)) / 2) +
"px"
                          });
                     }
                }
                else
                {
                     if
(this.firstInFullscreen)
                     {
                          if
(!this.isChrome &&
!this.isNodeWebkit)
                          {

     jQuery(this.canvas).css("m
argin",
this.fullscreenOldMarginCss);
```

```
                        }
     this.fullscreenOldMarginCs
s = "";

     this.firstInFullscreen =
false;
                        if
(this.fullscreen_mode === 0)
                        {

     this["setSize"](Math.round
(this.oldWidth /
this.devicePixelRatio),
Math.round(this.oldHeight /
this.devicePixelRatio), true);
                        }
                    }
                    else
                    {

     this.oldWidth =
this.width;

     this.oldHeight =
this.height;
                    }
                }
            }
            if (this.isloading)
            {
                var done =
this.areAllTexturesLoaded();
            // updates
this.progress

     this.loadingprogress =
this.progress;
                if (done)
                {

     this.isloading = false;

     this.progress = 1;

     this.trigger(cr.system_obj
ect.prototype.cnds.OnLoadFinish
ed, null);
                }
            }
            this.logic();
            if ((this.redraw ||
this.isCocoonJs) &&
!this.is_WebGL_context_lost &&
!this.suspendDrawing)
            {
                this.redraw =
false;
                if
(this.glwrap)

     this.drawGL();
                else

     this.draw();
                if
(this.snapshotCanvas)
                {
                    if
(this.canvas &&
this.canvas.toDataURL)
                    {

     this.snapshotData =
this.canvas.toDataURL(this.snap
shotCanvas[0],
this.snapshotCanvas[1]);

     this.trigger(cr.system_obj
ect.prototype.cnds.OnCanvasSnap
shot, null);
                    }

     this.snapshotCanvas =
null;
                }
            }
            if
(!this.hit_breakpoint)
            {

     this.tickcount++;

     this.execcount++;

     this.framecount++;
            }
            this.logictime +=
cr.performance_now() -
logic_start;
            if
(this.isSuspended)
                return;
            if (raf)
                this.raf_id =
raf(this.tickFunc,
this.canvas);
            else
            {
                this.timeout_id
= setTimeout(this.tickFunc,
this.isMobile ? 1 : 16);
```

```
            }
      };
      Runtime.prototype.logic =
function ()
      {
            var i, leni, j,
lenj, k, lenk, type, inst,
binst;
            var cur_time =
cr.performance_now();
            if (cur_time -
this.last_fps_time >= 1000)  //
every 1 second
            {

      this.last_fps_time +=
1000;
                  this.fps =
this.framecount;
                  this.framecount
= 0;

      this.cpuutilisation =
this.logictime;
                  this.logictime
= 0;
            }
            if
(this.measuring_dt)
            {
                  if
(this.last_tick_time !== 0)
                  {
                        var
ms_diff = cur_time -
this.last_tick_time;
                        if
(ms_diff === 0 &&
!this.isDebug)
                        {

      this.zeroDtCount++;
                              if
(this.zeroDtCout >= 10)

      this.measuring_dt = false;

      this.dt1 = 1.0 / 60.0;
// 60fps assumed (0.01666...)
                        }
                        else
                        {

      this.dt1 = ms_diff /
1000.0; // dt measured in
seconds
                              if
(this.dt1 > 0.5)

      this.dt1 = 0;
                              else
if (this.dt1 > 0.1)

      this.dt1 = 0.1;
                        }
                  }

      this.last_tick_time =
cur_time;
            }
         this.dt = this.dt1 *
this.timescale;

this.kahanTime.add(this.dt);
            var isfullscreen =
(document["mozFullScreen"] ||
document["webkitIsFullScreen"]
|| document["fullScreen"] ||
!!document["msFullscreenElement
"] || this.isNodeFullscreen);
            if
(this.fullscreen_mode >= 2 /*
scale */ || (isfullscreen &&
this.fullscreen_scaling > 0))
            {
                  var orig_aspect
= this.original_width /
this.original_height;
                  var cur_aspect
= this.width / this.height;
                  var mode =
this.fullscreen_mode;
                  if
(isfullscreen &&
this.fullscreen_scaling > 0)
                        mode =
this.fullscreen_scaling;
                  if ((mode !== 2
&& cur_aspect > orig_aspect) ||
(mode === 2 && cur_aspect <
orig_aspect))
                  {

      this.aspect_scale =
this.height /
this.original_height;
                  }
                  else
                  {

      this.aspect_scale =
```

```
this.width /
this.original_width;
                }
                if
(this.running_layout)
                {

     this.running_layout.scroll
ToX(this.running_layout.scrollX
);

     this.running_layout.scroll
ToY(this.running_layout.scrollY
);
                }
           }
           else

     this.aspect_scale =
(this.isRetina ?
this.devicePixelRatio : 1);

     this.ClearDeathRow();

     this.isInOnDestroy++;

     this.system.runWaits();
           // prevent instance
list changing
           this.isInOnDestroy--
;

     this.ClearDeathRow();
     // allow instance list
changing

     this.isInOnDestroy++;
           for (i = 0, leni =
this.types_by_index.length; i <
leni; i++)
           {
                type =
this.types_by_index[i];
                if
(type.is_family ||
(!type.behaviors.length &&
!type.families.length))
                     continue;
                for (j = 0,
lenj = type.instances.length; j
< lenj; j++)
                {
                     inst =
type.instances[j];
                     for (k =
0, lenk =
inst.behavior_insts.length; k <
lenk; k++)
                     {

     inst.behavior_insts[k].tic
k();
                     }
                }
           }
           for (i = 0, leni =
this.types_by_index.length; i <
leni; i++)
           {
                type =
this.types_by_index[i];
                if
(type.is_family ||
(!type.behaviors.length &&
!type.families.length))
                     continue;
     // type doesn't have any
behaviors
                for (j = 0,
lenj = type.instances.length; j
< lenj; j++)
                {
                     inst =
type.instances[j];
                     for (k =
0, lenk =
inst.behavior_insts.length; k <
lenk; k++)
                     {

     binst =
inst.behavior_insts[k];
                          if
(binst.posttick)

     binst.posttick();
                     }
                }
           }
        var tickarr =
this.objects_to_tick.valuesRef(
);
        for (i = 0, leni =
tickarr.length; i < leni; i++)
            tickarr[i].tick();
           this.isInOnDestroy--
;          // end preventing
instance lists from being
changed

     this.handleSaveLoad();
```

```
            // save/load now if
queued
            i = 0;
            while
(this.changelayout && i++ < 10)
            {
      this.doChangeLayout(this.c
hangelayout);
            }
         for (i = 0, leni =
this.eventsheets_by_index.lengt
h; i < leni; i++)
this.eventsheets_by_index[i].ha
sRun = false;
            if
(this.running_layout.event_shee
t)
      this.running_layout.event_
sheet.run();
      this.registered_collisions
.length = 0;
      this.layout_first_tick =
false;
      this.isInOnDestroy++;
      // prevent instance lists
from being changed
            for (i = 0, leni =
this.types_by_index.length; i <
leni; i++)
            {
                  type =
this.types_by_index[i];
                  if
(type.is_family ||
(!type.behaviors.length &&
!type.families.length))
                        continue;
      // type doesn't have any
behaviors
                  for (j = 0,
lenj = type.instances.length; j
< lenj; j++)
                  {
                        var inst
= type.instances[j];
                        for (k =
0, lenk =
inst.behavior_insts.length; k <
lenk; k++)
                        {
      binst =
inst.behavior_insts[k];
                              if
(binst.tick2)
      binst.tick2();
                              }
                        }
                  }
            tickarr =
this.objects_to_tick2.valuesRef
();
            for (i = 0, leni =
tickarr.length; i < leni; i++)
                  tickarr[i].tick2();
            this.isInOnDestroy--
;           // end preventing
instance lists from being
changed
      };
      Runtime.prototype.doChange
Layout = function
(changeToLayout)
      {
;
            var prev_layout =
this.running_layout;
      this.running_layout.stopRu
nning();
            var i, len, j, lenj,
k, lenk, type, inst, binst;
            if (this.glwrap)
            {
                  for (i = 0, len
= this.types_by_index.length; i
< len; i++)
                  {
                        type =
this.types_by_index[i];
                        if
(type.is_family)
      continue;
                        if
(type.unloadTextures &&
(!type.global ||
type.instances.length === 0) &&
changeToLayout.initial_types.in
dexOf(type) === -1)
                        {
      type.unloadTextures();
                        }
                  }
```

```
            }
            if (prev_layout ==
changeToLayout)

      this.system.waits.length =
0;

      changeToLayout.startRunnin
g();
            for (i = 0, len =
this.types_by_index.length; i <
len; i++)
            {
                  type =
this.types_by_index[i];
                  if
(!type.global &&
!type.plugin.singleglobal)
                        continue;
                  for (j = 0,
lenj = type.instances.length; j
< lenj; j++)
                  {
                        inst =
type.instances[j];
                        if
(inst.onLayoutChange)

      inst.onLayoutChange();
                        if
(inst.behavior_insts)
                        {
                              for
(k = 0, lenk =
inst.behavior_insts.length; k <
lenk; k++)
                              {

      binst =
inst.behavior_insts[k];

      if (binst.onLayoutChange)


      binst.onLayoutChange();
                              }
                        }
                  }
            }
            this.redraw = true;

      this.layout_first_tick =
true;

      this.ClearDeathRow();
      };
```

```
    Runtime.prototype.tickMe =
function (inst)
    {

this.objects_to_tick.add(inst);
    };
      Runtime.prototype.untickMe
= function (inst)
      {

      this.objects_to_tick.remov
e(inst);
      };
      Runtime.prototype.tick2Me
= function (inst)
    {

this.objects_to_tick2.add(inst)
;
    };
      Runtime.prototype.untick2M
e = function (inst)
      {

      this.objects_to_tick2.remo
ve(inst);
      };
    Runtime.prototype.getDt =
function (inst)
    {
        if (!inst ||
inst.my_timescale === -1.0)
            return this.dt;
        return this.dt1 *
inst.my_timescale;
    };
      Runtime.prototype.draw =
function ()
      {

      this.running_layout.draw(t
his.ctx);
            if
(this.isDirectCanvas)

      this.ctx["present"]();
      };
      Runtime.prototype.drawGL =
function ()
      {

      this.running_layout.drawGL
(this.glwrap);

      this.glwrap.present();
      };
```

```
	Runtime.prototype.addDestr
oyCallback = function (f)
	{
		if (f)

	this.destroycallbacks.push
(f);
	};
	Runtime.prototype.removeDe
stroyCallback = function (f)
	{

	cr.arrayFindRemove(this.de
stroycallbacks, f);
	};
	Runtime.prototype.getObjec
tByUID = function (uid_)
	{
;
		return
this.objectsByUid[uid_.toString
()];
	};
	Runtime.prototype.DestroyI
nstance = function (inst)
	{
		var i, len;
		if
(!this.deathRow.contains(inst))
		{

	this.deathRow.add(inst);
			if
(inst.is_contained)
			{
				for (i =
0, len = inst.siblings.length;
i < len; i++)
				{

	this.DestroyInstance(inst.
siblings[i]);
				}
			}
			if
(this.isInClearDeathRow)

	this.deathRow.values_cache
.push(inst);

	this.isInOnDestroy++;
	// support recursion

	this.trigger(Object.getPro
totypeOf(inst.type.plugin).cnds
.OnDestroyed, inst);

	this.isInOnDestroy--;
		}
	};
	Runtime.prototype.ClearDea
thRow = function ()
	{
		var inst, index,
type, instances, binst;
		var i, j, k, leni,
lenj, lenk;
		var w, f;

	this.isInClearDeathRow =
true;
		for (i = 0, leni =
this.createRow.length; i <
leni; i++)
		{
			inst =
this.createRow[i];
			type =
inst.type;

	type.instances.push(inst);
			for (j = 0,
lenj = type.families.length; j
< lenj; j++)
			{

	type.families[j].instances
.push(inst);

	type.families[j].stale_iid
s = true;
			}
		}

	this.createRow.length = 0;
		var arr =
this.deathRow.valuesRef();  //
get array of items from set
		for (i = 0; i <
arr.length; i++)	// check
array length every time in case
it changes
		{
			inst = arr[i];
			type =
inst.type;
			instances =
type.instances;
			for (j = 0,
lenj =
this.destroycallbacks.length; j
< lenj; j++)
```

```
this.destroycallbacks[j](inst);
cr.arrayFindRemove(instances, inst);
if (instances.length === 0)
type.any_instance_parallaxed = false;
if (inst.collcells)
{
type.collision_grid.update(inst, inst.collcells, null);
}
if (inst.layer)
{
cr.arrayRemove(inst.layer.instances, inst.get_zindex());
inst.layer.zindices_stale = true;
}
for (j = 0, lenj = type.families.length; j < lenj; j++)
{
cr.arrayFindRemove(type.families[j].instances, inst);
type.families[j].stale_iids = true;
}
if (inst.behavior_insts)
{
for (j = 0, lenj = inst.behavior_insts.length; j < lenj; j++)
{
binst = inst.behavior_insts[j];
if (binst.onDestroy)
binst.onDestroy();
binst.behavior.my_instances.remove(inst);
}
}
this.objects_to_tick.remove(inst);
this.objects_to_tick2.remove(inst);
for (j = 0, lenj = this.system.waits.length; j < lenj; j++)
{
w = this.system.waits[j];
if (w.sols.hasOwnProperty(type.index))
cr.arrayFindRemove(w.sols[type.index].insts, inst);
if (!type.is_family)
{
for (k = 0, lenk = type.families.length; k < lenk; k++)
{
f = type.families[k];
if (w.sols.hasOwnProperty(f.index))
cr.arrayFindRemove(w.sols[f.index].insts, inst);
}
}
}
if (inst.onDestroy)
inst.onDestroy();
if (this.objectsByUid.hasOwnProperty(inst.uid.toString()))
delete this.objectsByUid[inst.uid.toString()];
this.objectcount--;
if (type.deadCache.length < 64)
```

```
	type.deadCache.push(inst);
		type.stale_iids
= true;
		}
		if
(!this.deathRow.isEmpty())
		this.redraw =
true;

	this.deathRow.clear();

	this.isInClearDeathRow =
false;
	};
	Runtime.prototype.createIn
stance = function (type, layer,
sx, sy)
	{
		if (type.is_family)
		{
			var i =
cr.floor(Math.random() *
type.members.length);
			return
this.createInstance(type.member
s[i], layer, sx, sy);
		}
		if
(!type.default_instance)
		{
			return null;
		}
		return
this.createInstanceFromInit(typ
e.default_instance, layer,
false, sx, sy, false);
	};
	var all_behaviors = [];
	Runtime.prototype.createIn
stanceFromInit = function
(initial_inst, layer,
is_startup_instance, sx, sy,
skip_siblings)
	{
		var i, len, j, lenj,
p, effect_fallback, x, y;
		if (!initial_inst)
			return null;
		var type =
this.types_by_index[initial_ins
t[1]];
;
;
		var is_world =
type.plugin.is_world;
;
		if (this.isloading
&& is_world &&
!type.isOnLoaderLayout)
			return null;
		if (is_world &&
!this.glwrap &&
initial_inst[0][11] === 11)
			return null;
		var original_layer =
layer;
		if (!is_world)
			layer = null;
		var inst;
		if
(type.deadCache.length)
		{
			inst =
type.deadCache.pop();
			inst.recycled =
true;

	type.plugin.Instance.call(
inst, type);
		}
		else
		{
			inst = new
type.plugin.Instance(type);
			inst.recycled =
false;
		}
		if
(is_startup_instance &&
!skip_siblings)
			inst.uid =
initial_inst[2];
		else
			inst.uid =
this.next_uid++;

	this.objectsByUid[inst.uid
.toString()] = inst;
		inst.puid =
this.next_puid++;
		inst.iid =
type.instances.length;
		for (i = 0, len =
this.createRow.length; i < len;
++i)
		{
			if
(this.createRow[i].type ===
type)

	inst.iid++;
```

```
            }
            inst.get_iid =
cr.inst_get_iid;
            var initial_vars =
initial_inst[3];
            if (inst.recycled)
            {

      cr.wipe(inst.extra);
            }
            else
            {
                  inst.extra =
{};
                  if (typeof
cr_is_preview !== "undefined")
                  {

      inst.instance_var_names =
[];

      inst.instance_var_names.le
ngth = initial_vars.length;
                        for (i =
0, len = initial_vars.length; i
< len; i++)

      inst.instance_var_names[i]
= initial_vars[i][1];
                  }

      inst.instance_vars = [];

      inst.instance_vars.length
= initial_vars.length;
            }
            for (i = 0, len =
initial_vars.length; i < len;
i++)

      inst.instance_vars[i] =
initial_vars[i][0];
            if (is_world)
            {
                  var wm =
initial_inst[0];
;
                  inst.x =
cr.is_undefined(sx) ? wm[0] :
sx;
                  inst.y =
cr.is_undefined(sy) ? wm[1] :
sy;
                  inst.z = wm[2];
                  inst.width =
wm[3];
                  inst.height =
wm[4];
                  inst.depth =
wm[5];
                  inst.angle =
wm[6];
                  inst.opacity =
wm[7];
                  inst.hotspotX =
wm[8];
                  inst.hotspotY =
wm[9];
                  inst.blend_mode
= wm[10];
                  effect_fallback
= wm[11];
                  if
(!this.glwrap &&
type.effect_types.length)   //
no WebGL renderer and shaders
used

      inst.blend_mode =
effect_fallback;            //
use fallback blend mode -
destroy mode was handled above

      inst.compositeOp =
cr.effectToCompositeOp(inst.ble
nd_mode);
                  if (this.gl)

      cr.setGLBlend(inst,
inst.blend_mode, this.gl);
                  if
(inst.recycled)
                  {
                        for (i =
0, len = wm[12].length; i <
len; i++)
                        {
                              for
(j = 0, lenj =
wm[12][i].length; j < lenj;
j++)

      inst.effect_params[i][j] =
wm[12][i][j];
                        }

      inst.bbox.set(0, 0, 0, 0);

      inst.collcells.set(0, 0, -
1, -1);
```

```
inst.bquad.set_from_rect(inst.bbox);

inst.bbox_changed_callbacks.length = 0;
}
else
{

inst.effect_params = wm[12].slice(0);
for (i = 0, len = inst.effect_params.length; i < len; i++)

inst.effect_params[i] = wm[12][i].slice(0);

inst.active_effect_types = [];

inst.active_effect_flags = [];

inst.active_effect_flags.length = type.effect_types.length;
inst.bbox = new cr.rect(0, 0, 0, 0);

inst.collcells = new cr.rect(0, 0, -1, -1);

inst.bquad = new cr.quad();

inst.bbox_changed_callbacks = [];

inst.set_bbox_changed = cr.set_bbox_changed;

inst.add_bbox_changed_callback = cr.add_bbox_changed_callback;

inst.contains_pt = cr.inst_contains_pt;

inst.update_bbox = cr.update_bbox;

inst.get_zindex = cr.inst_get_zindex;
}

inst.tilemap_exists = false;

inst.tilemap_width = 0;

inst.tilemap_height = 0;

inst.tilemap_data = null;
if (wm.length === 14)
{

inst.tilemap_exists = true;

inst.tilemap_width = wm[13][0];

inst.tilemap_height = wm[13][1];

inst.tilemap_data = wm[13][2];
}
for (i = 0, len = type.effect_types.length; i < len; i++)

inst.active_effect_flags[i] = true;

inst.updateActiveEffects = cr.inst_updateActiveEffects;

inst.updateActiveEffects();

inst.uses_shaders = !!inst.active_effect_types.length;

inst.bbox_changed = true;

type.any_bbox_changed = true;
inst.visible = true;
inst.my_timescale = -1.0;
inst.layer = layer;
inst.zindex = layer.instances.length;    //
```

```
will be placed at top of
current layer
                if (typeof
inst.collision_poly ===
"undefined")

     inst.collision_poly =
null;

     inst.collisionsEnabled =
true;
                this.redraw =
true;
           }
           inst.toString =
cr.inst_toString;
           var initial_props,
binst;
           all_behaviors.length
= 0;
           for (i = 0, len =
type.families.length; i < len;
i++)
           {

     all_behaviors.push.apply(a
ll_behaviors,
type.families[i].behaviors);
           }

     all_behaviors.push.apply(a
ll_behaviors, type.behaviors);
           if (inst.recycled)
           {
                for (i = 0, len
= all_behaviors.length; i <
len; i++)
                {
                     var btype
= all_behaviors[i];
                     binst =
inst.behavior_insts[i];

     binst.recycled = true;

     btype.behavior.Instance.ca
ll(binst, btype, inst);

     initial_props =
initial_inst[4][i];
                     for (j =
0, lenj = initial_props.length;
j < lenj; j++)

     binst.properties[j] =
initial_props[j];

     binst.onCreate();

     btype.behavior.my_instance
s.add(inst);
                }
           }
           else
           {

     inst.behavior_insts = [];
                for (i = 0, len
= all_behaviors.length; i <
len; i++)
                {
                     var btype
= all_behaviors[i];
                     var binst
= new
btype.behavior.Instance(btype,
inst);

     binst.recycled = false;

     binst.properties =
initial_inst[4][i].slice(0);

     binst.onCreate();

     cr.seal(binst);

     inst.behavior_insts.push(b
inst);

     btype.behavior.my_instance
s.add(inst);
                }
           }
           initial_props =
initial_inst[5];
           if (inst.recycled)
           {
                for (i = 0, len
= initial_props.length; i <
len; i++)

     inst.properties[i] =
initial_props[i];
           }
           else
                inst.properties
= initial_props.slice(0);

     this.createRow.push(inst);
           if (layer)
           {
```

```
;
     layer.instances.push(inst)
;
                if
(layer.parallaxX !== 1 ||
layer.parallaxY !== 1)

     type.any_instance_parallax
ed = true;
           }
           this.objectcount++;
           if
(type.is_contained)
           {

     inst.is_contained = true;
                if
(inst.recycled)

     inst.siblings.length = 0;
                else

     inst.siblings = [];
           // note: should not
include self in siblings
                if
(!is_startup_instance &&
!skip_siblings)  // layout links
initial instances
                {
                      for (i =
0, len = type.container.length;
i < len; i++)
                      {
                            if
(type.container[i] === type)

     continue;
                            if
(!type.container[i].default_ins
tance)
                            {

     return null;
                            }

     inst.siblings.push(this.cr
eateInstanceFromInit(type.conta
iner[i].default_instance,
original_layer, false, is_world
? inst.x : sx, is_world ?
inst.y : sy, true));
                      }
                      for (i =
0, len = inst.siblings.length;
i < len; i++)
                      {

     inst.siblings[i].siblings.
push(inst);
                            for
(j = 0; j < len; j++)
                            {

     if (i !== j)


     inst.siblings[i].siblings.
push(inst.siblings[j]);
                            }
                      }
                }
           }
           else
           {

     inst.is_contained = false;
                inst.siblings =
null;
           }
           inst.onCreate();
           if (!inst.recycled)
                cr.seal(inst);
           for (i = 0, len =
inst.behavior_insts.length; i <
len; i++)
           {
                if
(inst.behavior_insts[i].postCre
ate)

     inst.behavior_insts[i].pos
tCreate();
           }
           return inst;
     };
     Runtime.prototype.getLayer
ByName = function (layer_name)
     {
           var i, len;
           for (i = 0, len =
this.running_layout.layers.leng
th; i < len; i++)
           {
                var layer =
this.running_layout.layers[i];
                if
(cr.equals_nocase(layer.name,
layer_name))
```

```
                        return
layer;
            }
            return null;
      };
      Runtime.prototype.getLayer
ByNumber = function (index)
      {
            index =
cr.floor(index);
            if (index < 0)
                  index = 0;
            if (index >=
this.running_layout.layers.leng
th)
                  index =
this.running_layout.layers.leng
th - 1;
            return
this.running_layout.layers[inde
x];
      };
      Runtime.prototype.getLayer
= function (l)
      {
            if (cr.is_number(l))
                  return
this.getLayerByNumber(l);
            else
                  return
this.getLayerByName(l.toString(
));
      };
      Runtime.prototype.clearSol
= function (solModifiers)
      {
            var i, len;
            for (i = 0, len =
solModifiers.length; i < len;
i++)
            {

      solModifiers[i].getCurrent
Sol().select_all = true;
            }
      };
      Runtime.prototype.pushClea
nSol = function (solModifiers)
      {
            var i, len;
            for (i = 0, len =
solModifiers.length; i < len;
i++)
            {

      solModifiers[i].pushCleanS
ol();
            }
      };
      Runtime.prototype.pushCopy
Sol = function (solModifiers)
      {
            var i, len;
            for (i = 0, len =
solModifiers.length; i < len;
i++)
            {

      solModifiers[i].pushCopySo
l();
            }
      };
      Runtime.prototype.popSol =
function (solModifiers)
      {
            var i, len;
            for (i = 0, len =
solModifiers.length; i < len;
i++)
            {

      solModifiers[i].popSol();
            }
      };
      Runtime.prototype.updateAl
lBBoxes = function (type)
      {
            if
(!type.any_bbox_changed)
                  return;
      // all instances must
already be up-to-date
            var i, len,
instances = type.instances;
            for (i = 0, len =
instances.length; i < len; ++i)
            {

      instances[i].update_bbox()
;
            }

      type.any_bbox_changed =
false;
      };
      Runtime.prototype.getColli
sionCandidates = function
(layer, rtype, bbox,
candidates)
      {
```

```
		var i, len, t;
		var is_parallaxed =
(layer ? (layer.parallaxX !== 1
|| layer.parallaxY !== 1) :
false);
		if (rtype.is_family)
		{
			for (i = 0, len
= rtype.members.length; i <
len; ++i)
			{
				t =
rtype.members[i];
				if
(is_parallaxed ||
t.any_instance_parallaxed)
				{

	cr.appendArray(candidates,
t.instances);
				}
				else
				{

	this.updateAllBBoxes(t);

	t.collision_grid.queryRang
e(bbox, candidates);
				}
			}
		}
		else
		{
			if
(is_parallaxed ||
rtype.any_instance_parallaxed)
			{

	cr.appendArray(candidates,
rtype.instances);
			}
			else
			{

	this.updateAllBBoxes(rtype
);

	rtype.collision_grid.query
Range(bbox, candidates);
			}
		}
	};
	Runtime.prototype.getTypes
CollisionCandidates = function
(layer, types, bbox,
candidates)
	{
		var i, len;
		for (i = 0, len =
types.length; i < len; ++i)
		{

	this.getCollisionCandidate
s(layer, types[i], bbox,
candidates);
		}
	};
	Runtime.prototype.getSolid
CollisionCandidates = function
(layer, bbox, candidates)
	{
		var solid =
this.getSolidBehavior();
		if (!solid)
			return null;

	this.getTypesCollisionCand
idates(layer, solid.my_types,
bbox, candidates);
	};
	Runtime.prototype.getJumpt
hruCollisionCandidates =
function (layer, bbox,
candidates)
	{
		var jumpthru =
this.getJumpthruBehavior();
		if (!jumpthru)
			return null;

	this.getTypesCollisionCand
idates(layer,
jumpthru.my_types, bbox,
candidates);
	};
	Runtime.prototype.testAndS
electCanvasPointOverlap =
function (type, ptx, pty,
inverted)
	{
		var sol =
type.getCurrentSol();
		var i, j, inst, len;
		var lx, ly;
		if (sol.select_all)
		{
			if (!inverted)
			{

	sol.select_all = false;
```

```
     sol.instances.length = 0;
// clear contents
                }
                for (i = 0, len
= type.instances.length; i <
len; i++)
                {
                     inst =
type.instances[i];

     inst.update_bbox();
                     lx =
inst.layer.canvasToLayer(ptx,
pty, true);
                     ly =
inst.layer.canvasToLayer(ptx,
pty, false);
                     if
(inst.contains_pt(lx, ly))
                     {
                          if
(inverted)

     return false;
                          else

     sol.instances.push(inst);
                     }
                }
           }
           else
           {
                j = 0;
                for (i = 0, len
= sol.instances.length; i <
len; i++)
                {
                     inst =
sol.instances[i];

     inst.update_bbox();
                     lx =
inst.layer.canvasToLayer(ptx,
pty, true);
                     ly =
inst.layer.canvasToLayer(ptx,
pty, false);
                     if
(inst.contains_pt(lx, ly))
                     {
                          if
(inverted)

     return false;
                          else
                                                  {
     sol.instances[j] =
sol.instances[i];

     j++;
                                                  }
                                             }
                                        }
                                        if (!inverted)

     sol.instances.length = j;
                                   }

     type.applySolToContainer()
;
                                   if (inverted)
                                        return true;
                                   // did not find
anything overlapping
                                   else
                                        return
sol.hasObjects();
     };
     Runtime.prototype.testOver
lap = function (a, b)
     {
                                   if (!a || !b || a
=== b || !a.collisionsEnabled
|| !b.collisionsEnabled)
                                        return false;
                                   a.update_bbox();
                                   b.update_bbox();
                                   var layera =
a.layer;
                                   var layerb =
b.layer;
                                   var different_layers
= (layera !== layerb &&
(layera.parallaxX !==
layerb.parallaxX ||
layerb.parallaxY !==
layerb.parallaxY ||
layera.scale !== layerb.scale
|| layera.angle !==
layerb.angle || layera.zoomRate
!== layerb.zoomRate));
                                   var i, len, i2, i21,
x, y, haspolya, haspolyb,
polya, polyb;
                                   if
(!different_layers)   // same
layers: easy check
                                   {
```

```
                if
(!a.bbox.intersects_rect(b.bbox
))
                        return
false;
                if
(!a.bquad.intersects_quad(b.bqu
ad))
                        return
false;
                if
(a.tilemap_exists &&
b.tilemap_exists)
                        return
false;
                if
(a.tilemap_exists)
                        return
this.testTilemapOverlap(a, b);
                if
(b.tilemap_exists)
                        return
this.testTilemapOverlap(b, a);
                haspolya =
(a.collision_poly &&
!a.collision_poly.is_empty());
                haspolyb =
(b.collision_poly &&
!b.collision_poly.is_empty());
                if (!haspolya
&& !haspolyb)
                        return
true;
                if (haspolya)
                {

     a.collision_poly.cache_pol
y(a.width, a.height, a.angle);
                        polya =
a.collision_poly;
                }
                else
                {

     this.temp_poly.set_from_qu
ad(a.bquad, a.x, a.y, a.width,
a.height);
                        polya =
this.temp_poly;
                }
                if (haspolyb)
                {

     b.collision_poly.cache_pol
y(b.width, b.height, b.angle);
                        polyb =
b.collision_poly;
                }
                else
                {

     this.temp_poly.set_from_qu
ad(b.bquad, b.x, b.y, b.width,
b.height);
                        polyb =
this.temp_poly;
                }
                return
polya.intersects_poly(polyb,
b.x - a.x, b.y - a.y);
            }
            else // different
layers: need to do full
translated check
            {
                haspolya =
(a.collision_poly &&
!a.collision_poly.is_empty());
                haspolyb =
(b.collision_poly &&
!b.collision_poly.is_empty());
                if (haspolya)
                {

     a.collision_poly.cache_pol
y(a.width, a.height, a.angle);

     this.temp_poly.set_from_po
ly(a.collision_poly);
                }
                else
                {

     this.temp_poly.set_from_qu
ad(a.bquad, a.x, a.y, a.width,
a.height);
                }
                polya =
this.temp_poly;
                if (haspolyb)
                {

     b.collision_poly.cache_pol
y(b.width, b.height, b.angle);

     this.temp_poly2.set_from_p
oly(b.collision_poly);
                }
                else
                {
```

```
	this.temp_poly2.set_from_quad(b.bquad, b.x, b.y, b.width, b.height);
		}
		polyb = this.temp_poly2;
		for (i = 0, len = polya.pts_count; i < len; i++)
		{
			i2 = i * 2;
			i21 = i2 + 1;
			x = polya.pts_cache[i2];
			y = polya.pts_cache[i21];

	polya.pts_cache[i2] = layera.layerToCanvas(x + a.x, y + a.y, true);

	polya.pts_cache[i21] = layera.layerToCanvas(x + a.x, y + a.y, false);
		}

	polya.update_bbox();
		for (i = 0, len = polyb.pts_count; i < len; i++)
		{
			i2 = i * 2;
			i21 = i2 + 1;
			x = polyb.pts_cache[i2];
			y = polyb.pts_cache[i21];

	polyb.pts_cache[i2] = layerb.layerToCanvas(x + b.x, y + b.y, true);

	polyb.pts_cache[i21] = layerb.layerToCanvas(x + b.x, y + b.y, false);
		}

	polyb.update_bbox();
		return polya.intersects_poly(polyb, 0, 0);
		}
	};
	var tmpQuad = new cr.quad();
	var tmpRect = new cr.rect(0, 0, 0, 0);
	Runtime.prototype.testTilemapOverlap = function (tm, a)
	{
		var collrects = tm.collision_rects;
		var i, len, c, rc;
		var bbox = a.bbox;
		var tmx = tm.x;
		var tmy = tm.y;
		var haspolya = (a.collision_poly && !a.collision_poly.is_empty());
		for (i = 0, len = collrects.length; i < len; ++i)
		{
			c = collrects[i];
			rc = c.rc;
			if (bbox.intersects_rect_off(rc, tmx, tmy))
			{

	tmpQuad.set_from_rect(rc);

	tmpQuad.offset(tmx, tmy);
				if (tmpQuad.intersects_quad(a.bquad))
				{
					if (haspolya)
					{

	a.collision_poly.cache_poly(a.width, a.height, a.angle);

	if (c.poly)

	{

		if (c.poly.intersects_poly(a.collision_poly, a.x - (tmx + rc.left), a.y - (tmy + rc.top)))

		{

			return true;
```

```
            }

      }

      else

      {


      this.temp_poly.set_from_qu
ad(tmpQuad, 0, 0, rc.right -
rc.left, rc.bottom - rc.top);

            if
(this.temp_poly.intersects_poly
(a.collision_poly, a.x, a.y))

                  return true;

      }
                              }
                              else
                              {

      if (c.poly)

      {


      this.temp_poly.set_from_qu
ad(a.bquad, 0, 0, a.width,
a.height);

            if
(c.poly.intersects_poly(this.te
mp_poly, -(tmx + rc.left), -
(tmy + rc.top)))

            {

                  return true;

            }

      }

      else

            return true;
                              }
                        }
                  }
            }
            return false;
      };
```

```
      Runtime.prototype.testRect
Overlap = function (r, b)
      {
            if (!b ||
!b.collisionsEnabled)
                  return false;
            b.update_bbox();
            var layerb =
b.layer;
            var haspolyb, polyb;
            if
(!b.bbox.intersects_rect(r))
                  return false;
            if
(b.tilemap_exists)
            {
                  var collrects =
b.collision_rects;
                  var i, len, c,
tilerc;
                  var tmx = b.x;
                  var tmy = b.y;
                  for (i = 0, len
= collrects.length; i < len;
++i)
                  {
                        c =
collrects[i];
                        tilerc =
c.rc;
                        if
(r.intersects_rect_off(tilerc,
tmx, tmy))
                        {
                              if
(c.poly)
                              {

      this.temp_poly.set_from_re
ct(r, 0, 0);

      if
(c.poly.intersects_poly(this.te
mp_poly, -(tmx + tilerc.left),
-(tmy + tilerc.top)))

            return true;
                              }
                              else

      return true;
                        }
                  }
                  return false;
            }
            else
```

```
		{
	tmpQuad.set_from_rect(r);
			if (!b.bquad.intersects_quad(tmpQuad))
				return false;
			haspolyb = (b.collision_poly && !b.collision_poly.is_empty());
			if (!haspolyb)
				return true;
	b.collision_poly.cache_poly(b.width, b.height, b.angle);
	tmpQuad.offset(-r.left, -r.top);
	this.temp_poly.set_from_quad(tmpQuad, 0, 0, 1, 1);
			return b.collision_poly.intersects_poly(this.temp_poly, r.left - b.x, r.top - b.y);
		}
	};
	Runtime.prototype.testSegmentOverlap = function (x1, y1, x2, y2, b)
	{
		if (!b || !b.collisionsEnabled)
			return false;
		b.update_bbox();
		var layerb = b.layer;
		var haspolyb, polyb;
	tmpRect.set(cr.min(x1, x2), cr.min(y1, y2), cr.max(x1, x2), cr.max(y1, y2));
		if (!b.bbox.intersects_rect(tmpRect))
			return false;
		if (b.tilemap_exists)
		{
			var collrects = b.collision_rects;
			var i, len, c, tilerc;
			var tmx = b.x;
			var tmy = b.y;
			for (i = 0, len = collrects.length; i < len; ++i)
			{
				c = collrects[i];
				tilerc = c.rc;
				if (tmpRect.intersects_rect_off(tilerc, tmx, tmy))
				{
	tmpQuad.set_from_rect(tilerc);
	tmpQuad.offset(tmx, tmy);
					if (tmpQuad.intersects_segment(x1, y1, x2, y2))
					{
	if (c.poly)
	{
		if (c.poly.intersects_segment(tmx + tilerc.left, tmy + tilerc.top, x1, y1, x2, y2))
		{
			return true;
		}
	}
	else
		return true;
					}
				}
			}
			return false;
		}
		else
		{
			if (!b.bquad.intersects_segment(x1, y1, x2, y2))
				return false;
```

```
                haspolyb =
(b.collision_poly &&
!b.collision_poly.is_empty());
                if (!haspolyb)
                      return
true;

     b.collision_poly.cache_pol
y(b.width, b.height, b.angle);
                return
b.collision_poly.intersects_seg
ment(b.x, b.y, x1, y1, x2, y2);
           }
     };
     Runtime.prototype.typeHasB
ehavior = function (t, b)
     {
           if (!b)
                return false;
           var i, len, j, lenj,
f;
           for (i = 0, len =
t.behaviors.length; i < len;
i++)
           {
                if
(t.behaviors[i].behavior
instanceof b)
                      return
true;
           }
           if (!t.is_family)
           {
                for (i = 0, len
= t.families.length; i < len;
i++)
                {
                      f =
t.families[i];
                      for (j =
0, lenj = f.behaviors.length; j
< lenj; j++)
                      {
                            if
(f.behaviors[j].behavior
instanceof b)

     return true;
                      }
                }
           }
           return false;
     };
     Runtime.prototype.typeHasN
oSaveBehavior = function (t)
     {
           return
this.typeHasBehavior(t,
cr.behaviors.NoSave);
     };
     Runtime.prototype.typeHasP
ersistBehavior = function (t)
     {
           return
this.typeHasBehavior(t,
cr.behaviors.Persist);
     };
     Runtime.prototype.getSolid
Behavior = function ()
     {
           return
this.solidBehavior;
     };
     Runtime.prototype.getJumpt
hruBehavior = function ()
     {
           return
this.jumpthruBehavior;
     };
     var candidates = [];
     Runtime.prototype.testOver
lapSolid = function (inst)
     {
           var i, len, s;
           inst.update_bbox();

     this.getSolidCollisionCand
idates(inst.layer, inst.bbox,
candidates);
           for (i = 0, len =
candidates.length; i < len;
++i)
           {
                s =
candidates[i];
                if
(!s.extra.solidEnabled)
                      continue;
                if
(this.testOverlap(inst, s))
                {

     candidates.length = 0;
                      return s;
                }
           }
           candidates.length =
0;
           return null;
     };
     Runtime.prototype.testRect
OverlapSolid = function (r)
```

```
	{
		var i, len, s;

	this.getSolidCollisionCand
idates(null, r, candidates);
		for (i = 0, len =
candidates.length; i < len;
++i)
		{
			s =
candidates[i];
			if
(!s.extra.solidEnabled)
				continue;
			if
(this.testRectOverlap(r, s))
			{

	candidates.length = 0;
				return s;
			}
		}
		candidates.length =
0;
		return null;
	};
	var jumpthru_array_ret =
[];
	Runtime.prototype.testOver
lapJumpThru = function (inst,
all)
	{
		var ret = null;
		if (all)
		{
			ret =
jumpthru_array_ret;
			ret.length = 0;
		}
		inst.update_bbox();

	this.getJumpthruCollisionC
andidates(inst.layer,
inst.bbox, candidates);
		var i, len, j;
		for (i = 0, len =
candidates.length; i < len;
++i)
		{
			j =
candidates[i];
			if
(!j.extra.jumpthruEnabled)
				continue;
			if
(this.testOverlap(inst, j))
			{
				if (all)

	ret.push(j);
				else
				{

	candidates.length = 0;

	return j;
				}
			}
		}
		candidates.length =
0;
		return ret;
	};
	Runtime.prototype.pushOutS
olid = function (inst, xdir,
ydir, dist, include_jumpthrus,
specific_jumpthru)
	{
		var push_dist = dist
|| 50;
		var oldx = inst.x
		var oldy = inst.y;
		var i;
		var last_overlapped
= null, secondlast_overlapped =
null;
		for (i = 0; i <
push_dist; i++)
		{
			inst.x = (oldx
+ (xdir * i));
			inst.y = (oldy
+ (ydir * i));

	inst.set_bbox_changed();
			if
(!this.testOverlap(inst,
last_overlapped))
			{

	last_overlapped =
this.testOverlapSolid(inst);
				if
(last_overlapped)

	secondlast_overlapped =
last_overlapped;
				if
(!last_overlapped)
				{
					if
(include_jumpthrus)
```

```
                        {
     if (specific_jumpthru)

          last_overlapped =
(this.testOverlap(inst,
specific_jumpthru) ?
specific_jumpthru : null);

     else

          last_overlapped =
this.testOverlapJumpThru(inst);

     if (last_overlapped)


     secondlast_overlapped =
last_overlapped;
                        }
                        if
(!last_overlapped)
                        {

     if (secondlast_overlapped)


     this.pushInFractional(inst
, xdir, ydir,
secondlast_overlapped, 16);

     return true;
                        }
                    }
               }
          }
          inst.x = oldx;
          inst.y = oldy;

     inst.set_bbox_changed();
          return false;
     };
     Runtime.prototype.pushOut
= function (inst, xdir, ydir,
dist, otherinst)
     {
          var push_dist = dist
|| 50;
          var oldx = inst.x
          var oldy = inst.y;
          var i;
          for (i = 0; i <
push_dist; i++)
          {
               inst.x = (oldx
+ (xdir * i));
               inst.y = (oldy
+ (ydir * i));

     inst.set_bbox_changed();
               if
(!this.testOverlap(inst,
otherinst))
                    return
true;
          }
          inst.x = oldx;
          inst.y = oldy;

     inst.set_bbox_changed();
          return false;
     };
     Runtime.prototype.pushInFr
actional = function (inst,
xdir, ydir, obj, limit)
     {
          var divisor = 2;
          var frac;
          var forward = false;
          var overlapping =
false;
          var bestx = inst.x;
          var besty = inst.y;
          while (divisor <=
limit)
          {
               frac = 1 /
divisor;
               divisor *= 2;
               inst.x += xdir
* frac * (forward ? 1 : -1);
               inst.y += ydir
* frac * (forward ? 1 : -1);

     inst.set_bbox_changed();
               if
(this.testOverlap(inst, obj))
               {
                    forward =
true;

     overlapping = true;
               }
               else
               {
                    forward =
false;

     overlapping = false;
                    bestx =
inst.x;
```

```
                    besty =
inst.y;
                }
            }
            if (overlapping)
            {
                inst.x = bestx;
                inst.y = besty;

      inst.set_bbox_changed();
            }
      };
      Runtime.prototype.pushOutS
olidNearest = function (inst,
max_dist_)
      {
            var max_dist =
(cr.is_undefined(max_dist_) ?
100 : max_dist_);
            var dist = 0;
            var oldx = inst.x
            var oldy = inst.y;
            var dir = 0;
            var dx = 0, dy = 0;
            var last_overlapped
= this.testOverlapSolid(inst);
            if
(!last_overlapped)
                return true;
            // already clear of
solids
            while (dist <=
max_dist)
            {
                switch (dir) {
                case 0:
      dx = 0; dy = -1; dist++;
break;
                case 1:
      dx = 1; dy = -1; break;
                case 2:
      dx = 1; dy = 0; break;
                case 3:
      dx = 1; dy = 1; break;
                case 4:
      dx = 0; dy = 1; break;
                case 5:
      dx = -1; dy = 1; break;
                case 6:
      dx = -1; dy = 0; break;
                case 7:
      dx = -1; dy = -1; break;
                }
                dir = (dir + 1)
% 8;
                inst.x =
cr.floor(oldx + (dx * dist));
                inst.y =
cr.floor(oldy + (dy * dist));

      inst.set_bbox_changed();
                if
(!this.testOverlap(inst,
last_overlapped))
                {

      last_overlapped =
this.testOverlapSolid(inst);
                    if
(!last_overlapped)

      return true;
                }
            }
            inst.x = oldx;
            inst.y = oldy;

      inst.set_bbox_changed();
            return false;
      };
      Runtime.prototype.register
Collision = function (a, b)
      {
            if
(!a.collisionsEnabled ||
!b.collisionsEnabled)
                return;

      this.registered_collisions
.push([a, b]);
      };
      Runtime.prototype.checkReg
isteredCollision = function (a,
b)
      {
            var i, len, x;
            for (i = 0, len =
this.registered_collisions.leng
th; i < len; i++)
            {
                x =
this.registered_collisions[i];
                if ((x[0] == a
&& x[1] == b) || (x[0] == b &&
x[1] == a))
                    return
true;
            }
            return false;
      };
```

```
Runtime.prototype.calculat
eSolidBounceAngle =
function(inst, startx, starty,
obj)
{
        var objx = inst.x;
        var objy = inst.y;
        var radius =
cr.max(10,
cr.distanceTo(startx, starty,
objx, objy));
        var startangle =
cr.angleTo(startx, starty,
objx, objy);
        var firstsolid = obj
|| this.testOverlapSolid(inst);
        if (!firstsolid)
            return
cr.clamp_angle(startangle +
cr.PI);
        var cursolid =
firstsolid;
        var i, curangle,
anticlockwise_free_angle,
clockwise_free_angle;
        var increment =
cr.to_radians(5);     // 5
degree increments
        for (i = 1; i < 36;
i++)
        {
            curangle =
startangle - i * increment;
            inst.x = startx
+ Math.cos(curangle) * radius;
            inst.y = starty
+ Math.sin(curangle) * radius;

inst.set_bbox_changed();
            if
(!this.testOverlap(inst,
cursolid))
            {
                cursolid
= obj ? null :
this.testOverlapSolid(inst);
                if
(!cursolid)
                {

anticlockwise_free_angle =
curangle;

break;
                }
            }
        }
        if (i === 36)

anticlockwise_free_angle =
cr.clamp_angle(startangle +
cr.PI);
        var cursolid =
firstsolid;
        for (i = 1; i < 36;
i++)
        {
            curangle =
startangle + i * increment;
            inst.x = startx
+ Math.cos(curangle) * radius;
            inst.y = starty
+ Math.sin(curangle) * radius;

inst.set_bbox_changed();
            if
(!this.testOverlap(inst,
cursolid))
            {
                cursolid
= obj ? null :
this.testOverlapSolid(inst);
                if
(!cursolid)
                {

clockwise_free_angle =
curangle;

break;
                }
            }
        }
        if (i === 36)

clockwise_free_angle =
cr.clamp_angle(startangle +
cr.PI);
        inst.x = objx;
        inst.y = objy;

inst.set_bbox_changed();
        if
(clockwise_free_angle ===
anticlockwise_free_angle)
            return
clockwise_free_angle;
        var half_diff =
cr.angleDiff(clockwise_free_ang
le, anticlockwise_free_angle) /
2;
        var normal;
```

```
            if
(cr.angleClockwise(clockwise_fr
ee_angle,
anticlockwise_free_angle))
            {
                  normal =
cr.clamp_angle(anticlockwise_fr
ee_angle + half_diff + cr.PI);
            }
            else
            {
                  normal =
cr.clamp_angle(clockwise_free_a
ngle + half_diff);
            }
;
            var vx =
Math.cos(startangle);
            var vy =
Math.sin(startangle);
            var nx =
Math.cos(normal);
            var ny =
Math.sin(normal);
            var v_dot_n = vx *
nx + vy * ny;
            var rx = vx - 2 *
v_dot_n * nx;
            var ry = vy - 2 *
v_dot_n * ny;
            return cr.angleTo(0,
0, rx, ry);
      };
      var triggerSheetStack =
[];
      var triggerSheetIndex = -
1;
      Runtime.prototype.trigger
= function (method, inst, value
/* for fast triggers */)
      {
;
            if
(!this.running_layout)
                  return false;
            var sheet =
this.running_layout.event_sheet
;
            if (!sheet)
                  return false;
// no event sheet active;
nothing to trigger
            triggerSheetIndex++;
            if
(triggerSheetIndex ===
triggerSheetStack.length)
      triggerSheetStack.push(new
cr.ObjectSet());
            else

      triggerSheetStack[triggerS
heetIndex].clear();
        var ret =
this.triggerOnSheet(method,
inst, sheet, value);
            triggerSheetIndex--;
            return ret;
    };

Runtime.prototype.triggerOnShee
t = function (method, inst,
sheet, value)
    {
            var
alreadyTriggeredSheets =
triggerSheetStack[triggerSheetI
ndex];
        if
(alreadyTriggeredSheets.contain
s(sheet))
             return false;

alreadyTriggeredSheets.add(shee
t);
        var includes =
sheet.includes.valuesRef();
        var ret = false;
            var i, leni, r;
        for (i = 0, leni =
includes.length; i < leni; i++)
        {
                  if
(includes[i].isActive())
                  {
                        r =
this.triggerOnSheet(method,
inst,
includes[i].include_sheet,
value);
                        ret = ret
|| r;
                  }
        }
            if (!inst)
            {
                  r =
this.triggerOnSheetForTypeName(
method, inst, "system", sheet,
value);
                  ret = ret || r;
            }
```

```
            else
            {
                  r =
this.triggerOnSheetForTypeName(
method, inst, inst.type.name,
sheet, value);
                  ret = ret || r;
                  for (i = 0,
leni =
inst.type.families.length; i <
leni; i++)
                  {
                        r =
this.triggerOnSheetForTypeName(
method, inst,
inst.type.families[i].name,
sheet, value);
                        ret = ret
|| r;
                  }
            }
            return ret;
// true if anything got
triggered
      };
      Runtime.prototype.triggerO
nSheetForTypeName = function
(method, inst, type_name,
sheet, value)
      {
            var i, leni;
            var ret = false,
ret2 = false;
            var trig, index;
            var fasttrigger =
(typeof value !== "undefined");
            var triggers =
(fasttrigger ?
sheet.fasttriggers :
sheet.triggers);
            var obj_entry =
triggers[type_name];
            if (!obj_entry)
                  return ret;
            var triggers_list =
null;
            for (i = 0, leni =
obj_entry.length; i < leni;
i++)
            {
                  if
(obj_entry[i].method == method)
                  {

      triggers_list =
obj_entry[i].evs;
                        break;
                  }
            }
            if (!triggers_list)
                  return ret;
            var
triggers_to_fire;
            if (fasttrigger)
            {

      triggers_to_fire =
triggers_list[value];
            }
            else
            {

      triggers_to_fire =
triggers_list;
            }
            if
(!triggers_to_fire)
                  return null;
            for (i = 0, leni =
triggers_to_fire.length; i <
leni; i++)
            {
                  trig =
triggers_to_fire[i][0];
                  index =
triggers_to_fire[i][1];
                  ret2 =
this.executeSingleTrigger(inst,
type_name, trig, index);
                  ret = ret ||
ret2;
            }
            return ret;
      };
      Runtime.prototype.executeS
ingleTrigger = function (inst,
type_name, trig, index)
      {
            var i, leni;
            var ret = false;

      this.trigger_depth++;
            var current_event =
this.getCurrentEventStack().cur
rent_event;
            if (current_event)

      this.pushCleanSol(current_
event.solModifiersIncludingPare
nts);
            var isrecursive =
(this.trigger_depth > 1);
```

```
     // calling trigger from
inside another trigger

     this.pushCleanSol(trig.sol
ModifiersIncludingParents);
           if (isrecursive)

     this.pushLocalVarStack();
           var event_stack =
this.pushEventStack(trig);

     event_stack.current_event
= trig;
           if (inst)
           {
                 var sol =
this.types[type_name].getCurren
tSol();
                 sol.select_all
= false;

     sol.instances.length = 1;

     sol.instances[0] = inst;

     this.types[type_name].appl
ySolToContainer();
           }
           var ok_to_run =
true;
           if (trig.parent)
           {
                 var
temp_parents_arr =
event_stack.temp_parents_arr;
                 var cur_parent
= trig.parent;
                 while
(cur_parent)
                 {

     temp_parents_arr.push(cur_
parent);

     cur_parent =
cur_parent.parent;
                 }

     temp_parents_arr.reverse()
;
                 for (i = 0,
leni = temp_parents_arr.length;
i < leni; i++)
                 {
                       if
(!temp_parents_arr[i].run_pretr
igger())   // parent event
failed
                       {

     ok_to_run = false;

     break;
                       }
                 }
           }
           if (ok_to_run)
           {

     this.execcount++;
                 if
(trig.orblock)

     trig.run_orblocktrigger(in
dex);
                 else

     trig.run();
                 ret = ret ||
event_stack.last_event_true;
           }

     this.popEventStack();
           if (isrecursive)

     this.popLocalVarStack();

     this.popSol(trig.solModifi
ersIncludingParents);
           if (current_event)

     this.popSol(current_event.
solModifiersIncludingParents);
           if
(this.isInOnDestroy === 0 &&
triggerSheetIndex === 0 &&
!this.isRunningEvents &&
(!this.deathRow.isEmpty() ||
this.createRow.length))
           {

     this.ClearDeathRow();
           }
           this.trigger_depth--
;
           return ret;
     };
     Runtime.prototype.getCurre
ntCondition = function ()
     {
           var evinfo =
this.getCurrentEventStack();
```

```
            return
evinfo.current_event.conditions
[evinfo.cndindex];
      };
      Runtime.prototype.getCurre
ntAction = function ()
      {
            var evinfo =
this.getCurrentEventStack();
            return
evinfo.current_event.actions[ev
info.actindex];
      };
      Runtime.prototype.pushLoca
lVarStack = function ()
      {

      this.localvar_stack_index+
+;
            if
(this.localvar_stack_index >=
this.localvar_stack.length)

      this.localvar_stack.push([
]);
      };
      Runtime.prototype.popLocal
VarStack = function ()
      {
;

      this.localvar_stack_index-
-;
      };
      Runtime.prototype.getCurre
ntLocalVarStack = function ()
      {
            return
this.localvar_stack[this.localv
ar_stack_index];
      };
      Runtime.prototype.pushEven
tStack = function (cur_event)
      {

      this.event_stack_index++;
            if
(this.event_stack_index >=
this.event_stack.length)

      this.event_stack.push(new
cr.eventStackFrame());
            var ret =
this.getCurrentEventStack();

      ret.reset(cur_event);
            return ret;
      };
      Runtime.prototype.popEvent
Stack = function ()
      {
;

      this.event_stack_index--;
      };
      Runtime.prototype.getCurre
ntEventStack = function ()
      {
            return
this.event_stack[this.event_sta
ck_index];
      };
      Runtime.prototype.pushLoop
Stack = function (name_)
      {

      this.loop_stack_index++;
            if
(this.loop_stack_index >=
this.loop_stack.length)
            {

      this.loop_stack.push(cr.se
al({ name: name_, index: 0,
stopped: false }));
            }
            var ret =
this.getCurrentLoop();
            ret.name = name_;
            ret.index = 0;
            ret.stopped = false;
            return ret;
      };
      Runtime.prototype.popLoopS
tack = function ()
      {
;

      this.loop_stack_index--;
      };
      Runtime.prototype.getCurre
ntLoop = function ()
      {
            return
this.loop_stack[this.loop_stack
_index];
      };
      Runtime.prototype.getEvent
VariableByName = function
(name, scope)
      {
```

```
		var i, leni, j,
lenj, sheet, e;
		while (scope)
		{
			for (i = 0,
leni = scope.subevents.length;
i < leni; i++)
			{
				e =
scope.subevents[i];
				if (e
instanceof cr.eventvariable &&
cr.equals_nocase(name, e.name))

	return e;
			}
			scope =
scope.parent;
		}
		for (i = 0, leni =
this.eventsheets_by_index.lengt
h; i < leni; i++)
		{
			sheet =
this.eventsheets_by_index[i];
			for (j = 0,
lenj = sheet.events.length; j <
lenj; j++)
			{
				e =
sheet.events[j];
				if (e
instanceof cr.eventvariable &&
cr.equals_nocase(name, e.name))

	return e;
			}
		}
		return null;
	};
	Runtime.prototype.getLayou
tBySid = function (sid_)
	{
		var i, len;
		for (i = 0, len =
this.layouts_by_index.length; i
< len; i++)
		{
			if
(this.layouts_by_index[i].sid
=== sid_)
				return
this.layouts_by_index[i];
		}
		return null;
	};
	Runtime.prototype.getObjec
tTypeBySid = function (sid_)
	{
		var i, len;
		for (i = 0, len =
this.types_by_index.length; i <
len; i++)
		{
			if
(this.types_by_index[i].sid ===
sid_)
				return
this.types_by_index[i];
		}
		return null;
	};
	Runtime.prototype.getGroup
BySid = function (sid_)
	{
		var i, len;
		for (i = 0, len =
this.allGroups.length; i < len;
i++)
		{
			if
(this.allGroups[i].sid ===
sid_)
				return
this.allGroups[i];
		}
		return null;
	};
	function makeSaveDb(e)
	{
		var db =
e.target.result;

	db.createObjectStore("save
s", { keyPath: "slot" });
	};
	function
IndexedDB_WriteSlot(slot_,
data_, oncomplete_, onerror_)
	{
		var request =
indexedDB.open("_C2SaveStates")
;

	request.onupgradeneeded =
makeSaveDb;
		request.onerror =
onerror_;
		request.onsuccess =
function (e)
		{
```

```
                var db =
e.target.result;
                db.onerror =
onerror_;
                var transaction
= db.transaction(["saves"],
"readwrite");
                var objectStore
=
transaction.objectStore("saves"
);
                var putReq =
objectStore.put({"slot": slot_,
"data": data_ });

     putReq.onsuccess =
oncomplete_;
           };
     };
     function
IndexedDB_ReadSlot(slot_,
oncomplete_, onerror_)
     {
           var request =
indexedDB.open("_C2SaveStates")
;

     request.onupgradeneeded =
makeSaveDb;
           request.onerror =
onerror_;
           request.onsuccess =
function (e)
           {
                var db =
e.target.result;
                db.onerror =
onerror_;
                var transaction
= db.transaction(["saves"]);
                var objectStore
=
transaction.objectStore("saves"
);
                var readReq =
objectStore.get(slot_);

     readReq.onsuccess =
function (e)
                {
                      if
(readReq.result)

     oncomplete_(readReq.result
["data"]);
                      else

     oncomplete_(null);
                      };
           };
     };
     Runtime.prototype.signalCo
ntinuousPreview = function ()
     {

     this.signalledContinuousPr
eview = true;
     };
     function
doContinuousPreviewReload()
     {

     cr.logexport("Reloading
for continuous preview");
           if
(!!window["c2cocoonjs"])
           {

     CocoonJS["App"]["reload"](
);
           }
           else
           {
                 if
(window.location.search.indexOf
("continuous") > -1)

     window.location.reload(tru
e);
                 else

     window.location =
window.location +
"?continuous";
           }
     };
     Runtime.prototype.handleSa
veLoad = function ()
     {
           var self = this;
           var savingToSlot =
this.saveToSlot;
           var savingJson =
this.lastSaveJson;
           var loadingFromSlot
= this.loadFromSlot;
           var continuous =
false;
           if
(this.signalledContinuousPrevie
w)
           {
```

```
                continuous =
true;
                savingToSlot =
"__c2_continuouspreview";

     this.signalledContinuousPr
eview = false;
           }
           if
(savingToSlot.length)
           {

     this.ClearDeathRow();
                savingJson =
this.saveToJSONString();
                if
(window.indexedDB &&
!this.isCocoonJs)
                {

     IndexedDB_WriteSlot(saving
ToSlot, savingJson, function ()
                      {

     cr.logexport("Saved state
to IndexedDB storage (" +
savingJson.length + " bytes)");

     self.lastSaveJson =
savingJson;

     self.trigger(cr.system_obj
ect.prototype.cnds.OnSaveComple
te, null);

     self.lastSaveJson = "";
                           if
(continuous)

     doContinuousPreviewReload(
);
                      },
function (e)
                      {
                           try
{

     localStorage.setItem("__c2
save_" + savingToSlot,
savingJson);

     cr.logexport("Saved state
to WebStorage (" +
savingJson.length + " bytes)");

     self.lastSaveJson =
savingJson;

     self.trigger(cr.system_obj
ect.prototype.cnds.OnSaveComple
te, null);

     self.lastSaveJson = "";

     if (continuous)


     doContinuousPreviewReload(
);
                           }

     catch (f)
                           {

     cr.logexport("Failed to
save game state: " + e + "; " +
f);
                           }
                      });
                }
                else
                {
                      try {

     localStorage.setItem("__c2
save_" + savingToSlot,
savingJson);

     cr.logexport("Saved state
to WebStorage (" +
savingJson.length + " bytes)");

     self.lastSaveJson =
savingJson;

     this.trigger(cr.system_obj
ect.prototype.cnds.OnSaveComple
te, null);

     self.lastSaveJson = "";
                           if
(continuous)

     doContinuousPreviewReload(
);
                      }
                      catch (e)
                      {
```

```
cr.logexport("Error saving
to WebStorage: " + e);
                    }
                }
                this.saveToSlot
= "";

this.loadFromSlot = "";

this.loadFromJson = "";
        }
        if
(loadingFromSlot.length)
        {
                if
(window.indexedDB &&
!this.isCocoonJs)
                {

IndexedDB_ReadSlot(loading
FromSlot, function (result_)
                    {
                        if
(result_)
                        {

self.loadFromJson =
result_;

cr.logexport("Loaded state
from IndexedDB storage (" +
self.loadFromJson.length + "
bytes)");
                        }
                        else
                        {

self.loadFromJson =
localStorage.getItem("__c2save_
" + loadingFromSlot) || "";

cr.logexport("Loaded state
from WebStorage (" +
self.loadFromJson.length + "
bytes)");
                        }

self.suspendDrawing =
false;
                        if
(!self.loadFromJson.length)

self.trigger(cr.system_obj
ect.prototype.cnds.OnLoadFailed
, null);
                    },
function (e)
                    {

self.loadFromJson =
localStorage.getItem("__c2save_
" + loadingFromSlot) || "";

cr.logexport("Loaded state
from WebStorage (" +
self.loadFromJson.length + "
bytes)");

self.suspendDrawing =
false;
                        if
(!self.loadFromJson.length)

self.trigger(cr.system_obj
ect.prototype.cnds.OnLoadFailed
, null);
                    });
                }
                else
                {

this.loadFromJson =
localStorage.getItem("__c2save_
" + loadingFromSlot) || "";

cr.logexport("Loaded state
from WebStorage (" +
this.loadFromJson.length + "
bytes)");

this.suspendDrawing =
false;
                    if
(!self.loadFromJson.length)

this.trigger(cr.system_obj
ect.prototype.cnds.OnLoadFailed
, null);
                }

this.loadFromSlot = "";
                this.saveToSlot
= "";
            }
            if
(this.loadFromJson.length)
            {

this.ClearDeathRow();
```

```
	this.loadFromJSONString(th
is.loadFromJson);

	this.lastSaveJson =
this.loadFromJson;

	this.trigger(cr.system_obj
ect.prototype.cnds.OnLoadComple
te, null);

	this.lastSaveJson = "";

	this.loadFromJson = "";
		}
	};
	function
CopyExtraObject(extra)
	{
		var p, ret = {};
		for (p in extra)
		{
			if
(extra.hasOwnProperty(p))
			{
				if
(extra[p] instanceof
cr.ObjectSet)

	continue;
				if
(extra[p] && typeof
extra[p].c2userdata !==
"undefined")

	continue;
				ret[p] =
extra[p];
			}
		}
		return ret;
	};
	Runtime.prototype.saveToJS
ONString = function()
	{
		var i, len, j, lenj,
type, layout, typeobj, g, c, a,
v, p;
		var o = {
			"c2save":
			true,
			"version":
			1,
			"rt": {
				"time":
	this.kahanTime.sum,

	"timescale":
	this.timescale,

	"tickcount":
	this.tickcount,

	"execcount":
	this.execcount,

	"next_uid":
	this.next_uid,

	"running_layout":
	this.running_layout.sid,

	"start_time_offset":
(Date.now() - this.start_time)
				},
				"types": {},
				"layouts": {},
				"events": {
					"groups":
{},
					"cnds":
{},
					"acts":
{},
					"vars":
{}
				}
			};
			for (i = 0, len =
this.types_by_index.length; i <
len; i++)
			{
				type =
this.types_by_index[i];
				if
(type.is_family ||
this.typeHasNoSaveBehavior(type
))
					continue;
				typeobj = {

	"instances": []
				};
				if
(cr.hasAnyOwnProperty(type.extr
a))

	typeobj["ex"] =
CopyExtraObject(type.extra);
```

```
                for (j = 0,
lenj = type.instances.length; j
< lenj; j++)
                {

     typeobj["instances"].push(
this.saveInstanceToJSON(type.in
stances[j]));
                }

     o["types"][type.sid.toStri
ng()] = typeobj;
           }
           for (i = 0, len =
this.layouts_by_index.length; i
< len; i++)
           {
                layout =
this.layouts_by_index[i];

     o["layouts"][layout.sid.to
String()] =
layout.saveToJSON();
           }
           var ogroups =
o["events"]["groups"];
           for (i = 0, len =
this.allGroups.length; i < len;
i++)
           {
                g =
this.allGroups[i];

     ogroups[g.sid.toString()]
=
!!this.activeGroups[g.group_nam
e];
           }
           var ocnds =
o["events"]["cnds"];
           for (p in
this.cndsBySid)
           {
                if
(this.cndsBySid.hasOwnProperty(
p))
                {
                     c =
this.cndsBySid[p];
                     if
(cr.hasAnyOwnProperty(c.extra))

     ocnds[p] = { "ex":
CopyExtraObject(c.extra) };
                }
           }
           var oacts =
o["events"]["acts"];
           for (p in
this.actsBySid)
           {
                if
(this.actsBySid.hasOwnProperty(
p))
                {
                     a =
this.actsBySid[p];
                     if
(cr.hasAnyOwnProperty(a.extra))

     oacts[p] = { "ex": a.extra
};
                }
           }
           var ovars =
o["events"]["vars"];
           for (p in
this.varsBySid)
           {
                if
(this.varsBySid.hasOwnProperty(
p))
                {
                     v =
this.varsBySid[p];
                     if
(!v.is_constant && (!v.parent
|| v.is_static))

     ovars[p] = v.data;
                }
           }
           o["system"] =
this.system.saveToJSON();
           return
JSON.stringify(o);
      };
      Runtime.prototype.refreshU
idMap = function ()
      {
           var i, len, type, j,
lenj, inst;
           this.objectsByUid =
{};
           for (i = 0, len =
this.types_by_index.length; i <
len; i++)
           {
                type =
this.types_by_index[i];
                if
(type.is_family)
```

```
                    continue;
                for (j = 0,
lenj = type.instances.length; j
< lenj; j++)
                {
                    inst =
type.instances[j];

     this.objectsByUid[inst.uid
.toString()] = inst;
                }
            }
     };
     Runtime.prototype.loadFrom
JSONString = function (str)
     {
            var o =
JSON.parse(str);
            if (!o["c2save"])
                return;
     // probably not a c2 save
state
            if (o["version"] >
1)
                return;
     // from future version of
c2; assume not compatible
            var rt = o["rt"];

     this.kahanTime.reset();
            this.kahanTime.sum =
rt["time"];
            this.timescale =
rt["timescale"];
            this.tickcount =
rt["tickcount"];
            this.execcount =
rt["execcount"];
            this.start_time =
Date.now() -
rt["start_time_offset"];
            var layout_sid =
rt["running_layout"];
            if (layout_sid !==
this.running_layout.sid)
            {
                var
changeToLayout =
this.getLayoutBySid(layout_sid)
;
                if
(changeToLayout)

     this.doChangeLayout(change
ToLayout);
                else
                    return;
            // layout that was
saved on has gone missing
(deleted?)
            }
            this.isLoadingState
= true;
            var i, len, j, lenj,
k, lenk, p, type,
existing_insts, load_insts,
inst, binst, layout, layer, g,
iid, t;
            var otypes =
o["types"];
            for (p in otypes)
            {
                if
(otypes.hasOwnProperty(p))
                {
                    type =
this.getObjectTypeBySid(parseIn
t(p, 10));
                    if (!type
|| type.is_family ||
this.typeHasNoSaveBehavior(type
))

     continue;
                    if
(otypes[p]["ex"])

     type.extra =
otypes[p]["ex"];
                    else

     cr.wipe(type.extra);

     existing_insts =
type.instances;

     load_insts =
otypes[p]["instances"];
                    for (i =
0, len =
cr.min(existing_insts.length,
load_insts.length); i < len;
i++)
                    {

     this.loadInstanceFromJSON(
existing_insts[i],
load_insts[i]);
                    }
                    for (i =
load_insts.length, len =
```

```
existing_insts.length; i < len;
i++)

     this.DestroyInstance(exist
ing_insts[i]);
                       for (i =
existing_insts.length, len =
load_insts.length; i < len;
i++)
                       {

     layer = null;
                            if
(type.plugin.is_world)
                            {

     layer =
this.running_layout.getLayerByS
id(load_insts[i]["w"]["l"]);

     if (!layer)

           continue;
                            }
                            inst
=
this.createInstanceFromInit(typ
e.default_instance, layer,
false, 0, 0, true);

     this.loadInstanceFromJSON(
inst, load_insts[i]);
                       }

     type.stale_iids = true;
                 }
           }

     this.ClearDeathRow();

     this.refreshUidMap();
           var olayouts =
o["layouts"];
           for (p in olayouts)
           {
                 if
(olayouts.hasOwnProperty(p))
                 {
                       layout =
this.getLayoutBySid(parseInt(p,
10));
                       if
(!layout)

     continue;       //
must've gone missing

     layout.loadFromJSON(olayou
ts[p]);
                 }
           }
           var ogroups =
o["events"]["groups"];
           for (p in ogroups)
           {
                 if
(ogroups.hasOwnProperty(p))
                 {
                       g =
this.getGroupBySid(parseInt(p,
10));
                       if (g)

     this.activeGroups[g.group_
name] = ogroups[p];
                 }
           }
           var ocnds =
o["events"]["cnds"];
           for (p in ocnds)
           {
                 if
(ocnds.hasOwnProperty(p) &&
this.cndsBySid.hasOwnProperty(p
))
                 {

     this.cndsBySid[p].extra =
ocnds[p]["ex"];
                 }
           }
           var oacts =
o["events"]["acts"];
           for (p in oacts)
           {
                 if
(oacts.hasOwnProperty(p) &&
this.actsBySid.hasOwnProperty(p
))
                 {

     this.actsBySid[p].extra =
oacts[p]["ex"];
                 }
           }
           var ovars =
o["events"]["vars"];
           for (p in ovars)
           {
                 if
(ovars.hasOwnProperty(p) &&
```

```
this.varsBySid.hasOwnProperty(p
))
                {
     this.varsBySid[p].data =
ovars[p];
                }
           }
           this.next_uid =
rt["next_uid"];
           this.isLoadingState
= false;
     this.system.loadFromJSON(o
["system"]);
           for (i = 0, len =
this.types_by_index.length; i <
len; i++)
           {
                type =
this.types_by_index[i];
                if
(type.is_family)
                     continue;
                for (j = 0,
lenj = type.instances.length; j
< lenj; j++)
                {
                     inst =
type.instances[j];
                     if
(type.is_contained)
                     {
                          iid
= inst.get_iid();
     inst.siblings.length = 0;
                          for
(k = 0, lenk =
type.container.length; k <
lenk; k++)
                          {
     t = type.container[k];
     if (type === t)
           continue;
;
     inst.siblings.push(t.insta
nces[iid]);
                          }
                     }
                     if
(inst.afterLoad)
     inst.afterLoad();
                     if
(inst.behavior_insts)
                     {
                          for
(k = 0, lenk =
inst.behavior_insts.length; k <
lenk; k++)
                          {
     binst =
inst.behavior_insts[k];
     if (binst.afterLoad)
           binst.afterLoad();
                          }
                     }
                }
           }
           this.redraw = true;
     };
     Runtime.prototype.saveInst
anceToJSON = function(inst,
state_only)
     {
           var i, len, world,
behinst, et;
           var type =
inst.type;
           var plugin =
type.plugin;
           var o = {};
           if (state_only)
                o["c2"] = true;
           // mark as known
json data from Construct 2
           else
                o["uid"] =
inst.uid;
           if
(cr.hasAnyOwnProperty(inst.extr
a))
                o["ex"] =
CopyExtraObject(inst.extra);
           if
(inst.instance_vars &&
inst.instance_vars.length)
           {
                o["ivs"] = {};
                for (i = 0, len
= inst.instance_vars.length; i
< len; i++)
                {
```

```
o["ivs"][inst.type.instvar
_sids[i].toString()] =
inst.instance_vars[i];
                }
            }
            if (plugin.is_world)
            {
                world = {
                    "x":
inst.x,
                    "y":
inst.y,
                    "w":
inst.width,
                    "h":
inst.height,
                    "l":
inst.layer.sid,
                    "zi":
inst.get_zindex()
                };
                if (inst.angle
!== 0)

world["a"] = inst.angle;
                if
(inst.opacity !== 1)

world["o"] = inst.opacity;
                if
(inst.hotspotX !== 0.5)

world["hX"] =
inst.hotspotX;
                if
(inst.hotspotY !== 0.5)

world["hY"] =
inst.hotspotY;
                if
(inst.blend_mode !== 0)

world["bm"] =
inst.blend_mode;
                if
(!inst.visible)

world["v"] = inst.visible;
                if
(!inst.collisionsEnabled)

world["ce"] =
inst.collisionsEnabled;
                if
(inst.my_timescale !== -1)

world["mts"] =
inst.my_timescale;
                if
(type.effect_types.length)
                {

world["fx"] = [];
                    for (i =
0, len =
type.effect_types.length; i <
len; i++)
                    {
                        et =
type.effect_types[i];

world["fx"].push({"name":
et.name,


"active":
inst.active_effect_flags[et.ind
ex],


"params":
inst.effect_params[et.index]
});
                    }
                }
                o["w"] = world;
            }
            if
(inst.behavior_insts &&
inst.behavior_insts.length)
            {
                o["behs"] = {};
                for (i = 0, len
= inst.behavior_insts.length; i
< len; i++)
                {
                    behinst =
inst.behavior_insts[i];
                    if
(behinst.saveToJSON)

o["behs"][behinst.type.sid
.toString()] =
behinst.saveToJSON();
                }
            }
            if (inst.saveToJSON)
                o["data"] =
inst.saveToJSON();
            return o;
        };
```

```
	Runtime.prototype.getInsta
nceVarIndexBySid = function
(type, sid_)
	{
		var i, len;
		for (i = 0, len =
type.instvar_sids.length; i <
len; i++)
		{
			if
(type.instvar_sids[i] === sid_)
				return i;
		}
		return -1;
	};
	Runtime.prototype.getBehav
iorIndexBySid = function (inst,
sid_)
	{
		var i, len;
		for (i = 0, len =
inst.behavior_insts.length; i <
len; i++)
		{
			if
(inst.behavior_insts[i].type.si
d === sid_)
				return i;
		}
		return -1;
	};
	Runtime.prototype.loadInst
anceFromJSON = function(inst,
o, state_only)
	{
		var p, i, len, iv,
oivs, world, fxindex, obehs,
behindex;
		var oldlayer;
		var type =
inst.type;
		var plugin =
type.plugin;
		if (state_only)
		{
			if (!o["c2"])
				return;
		}
		else
			inst.uid =
o["uid"];
		if (o["ex"])
			inst.extra =
o["ex"];
		else
		cr.wipe(inst.extra);
			oivs = o["ivs"];
			if (oivs)
			{
				for (p in oivs)
				{
					if
(oivs.hasOwnProperty(p))
					{
						iv =
this.getInstanceVarIndexBySid(t
ype, parseInt(p, 10));
						if
(iv < 0 || iv >=
inst.instance_vars.length)

		continue;	//
must've gone missing

		inst.instance_vars[iv] =
oivs[p];
					}
				}
			}
			if (plugin.is_world)
			{
				world = o["w"];
				if
(inst.layer.sid !== world["l"])
				{
					oldlayer
= inst.layer;

		inst.layer =
this.running_layout.getLayerBySid(world["l"]);
					if
(inst.layer)
					{

		inst.layer.instances.push(
inst);

		inst.layer.zindices_stale
= true;

		cr.arrayFindRemove(oldlaye
r.instances, inst);

		oldlayer.zindices_stale =
true;
					}
					else
					{
```

```
                    }
                }
                inst.x =
world["x"];
                inst.y =
world["y"];
                inst.width =
world["w"];
                inst.height =
world["h"];
                inst.zindex =
world["zi"];
                inst.angle =
world.hasOwnProperty("a") ?
world["a"] : 0;
                inst.opacity =
world.hasOwnProperty("o") ?
world["o"] : 1;
                inst.hotspotX =
world.hasOwnProperty("hX") ?
world["hX"] : 0.5;
                inst.hotspotY =
world.hasOwnProperty("hY") ?
world["hY"] : 0.5;
                inst.visible =
world.hasOwnProperty("v") ?
world["v"] : true;

     inst.collisionsEnabled =
world.hasOwnProperty("ce") ?
world["ce"] : true;

     inst.my_timescale =
world.hasOwnProperty("mts") ?
world["mts"] : -1;
                inst.blend_mode
= world.hasOwnProperty("bm") ?
world["bm"] : 0;;

     inst.compositeOp =
cr.effectToCompositeOp(inst.ble
nd_mode);
                if (this.gl)

     cr.setGLBlend(inst,
inst.blend_mode, this.gl);

     inst.set_bbox_changed();
                if
(world.hasOwnProperty("fx"))
                {
                    for (i =
0, len = world["fx"].length; i
< len; i++)
                    {

     fxindex =
type.getEffectIndexByName(world
["fx"][i]["name"]);
                        if
(fxindex < 0)

     continue;       //
must've gone missing

     inst.active_effect_flags[f
xindex] =
world["fx"][i]["active"];

     inst.effect_params[fxindex
] = world["fx"][i]["params"];
                    }
                }

     inst.updateActiveEffects()
;
            }
            obehs = o["behs"];
            if (obehs)
            {
                for (p in
obehs)
                {
                    if
(obehs.hasOwnProperty(p))
                    {

     behindex =
this.getBehaviorIndexBySid(inst
, parseInt(p, 10));
                        if
(behindex < 0)

     continue;       //
must've gone missing

     inst.behavior_insts[behind
ex].loadFromJSON(obehs[p]);
                    }
                }
            }
            if (o["data"])

     inst.loadFromJSON(o["data"
]);
     };
     cr.runtime = Runtime;
```

```
	cr.createRuntime =
function (canvasid)
	{
		return new
Runtime(document.getElementById
(canvasid));
	};
	cr.createDCRuntime =
function (w, h)
	{
		return new Runtime({
"dc": true, "width": w,
"height": h });
	};
	window["cr_createRuntime"]
= cr.createRuntime;
	window["cr_createDCRuntime
"] = cr.createDCRuntime;
	window["createCocoonJSRunt
ime"] = function ()
	{
		window["c2cocoonjs"]
= true;
		var canvas =
document.createElement("screenc
anvas") ||
document.createElement("canvas"
);
		canvas.screencanvas
= true;

	document.body.appendChild(
canvas);
		var rt = new
Runtime(canvas);
		window["c2runtime"]
= rt;

	window.addEventListener("o
rientationchange", function ()
{

	window["c2runtime"]["setSi
ze"](window.innerWidth,
window.innerHeight);
		});

	window["c2runtime"]["setSi
ze"](window.innerWidth,
window.innerHeight);
		return rt;
	};
}());
window["cr_getC2Runtime"] =
function()
{
	var canvas =
document.getElementById("c2canv
as");
	if (canvas)
		return
canvas["c2runtime"];
	else if
(window["c2runtime"])
		return
window["c2runtime"];
	else
		return null;
}
window["cr_sizeCanvas"] =
function(w, h)
{
	if (w === 0 || h === 0)
		return;
	var runtime =
window["cr_getC2Runtime"]();
	if (runtime)

	runtime["setSize"](w, h);
}
window["cr_setSuspended"] =
function(s)
{
	var runtime =
window["cr_getC2Runtime"]();
	if (runtime)

	runtime["setSuspended"](s)
;
}
;
(function()
{
	function Layout(runtime,
m)
	{
		this.runtime =
runtime;
		this.event_sheet =
null;
		this.scrollX =
(this.runtime.original_width /
2);
		this.scrollY =
(this.runtime.original_height /
2);
		this.scale = 1.0;
		this.angle = 0;
		this.first_visit =
true;
		this.name = m[0];
		this.width = m[1];
```

```
            this.height = m[2];
      this.unbounded_scrolling =
m[3];
            this.sheetname =
m[4];
            this.sid = m[5];
            var lm = m[6];
            var i, len;
            this.layers = [];
            this.initial_types =
[];
            for (i = 0, len =
lm.length; i < len; i++)
            {
                  var layer = new
cr.layer(this, lm[i]);
                  layer.number =
i;
                  cr.seal(layer);
      this.layers.push(layer);
            }
            var im = m[7];
      this.initial_nonworld =
[];
            for (i = 0, len =
im.length; i < len; i++)
            {
                  var inst =
im[i];
                  var type =
this.runtime.types_by_index[ins
t[1]];
;
                  if
(!type.default_instance)
      type.default_instance =
inst;
      this.initial_nonworld.push
(inst);
                  if
(this.initial_types.indexOf(typ
e) === -1)
      this.initial_types.push(ty
pe);
            }
            this.effect_types =
[];
      this.active_effect_types =
[];
            this.effect_params =
[];
            for (i = 0, len =
m[8].length; i < len; i++)
            {
      this.effect_types.push({
                        id:
m[8][i][0],
                        name:
m[8][i][1],
      shaderindex: -1,
                        active:
true,
                        index: i
                  });
      this.effect_params.push(m[
8][i][2].slice(0));
            }
      this.updateActiveEffects()
;
            this.rcTex = new
cr.rect(0, 0, 1, 1);
            this.rcTex2 = new
cr.rect(0, 0, 1, 1);
            this.persist_data =
{};
      };
      Layout.prototype.saveObjec
tToPersist = function (inst)
      {
            var sidStr =
inst.type.sid.toString();
            if
(!this.persist_data.hasOwnPrope
rty(sidStr))
      this.persist_data[sidStr]
= [];
            var type_persist =
this.persist_data[sidStr];
      type_persist.push(this.run
time.saveInstanceToJSON(inst));
      };
      Layout.prototype.hasOpaque
BottomLayer = function ()
      {
            var layer =
this.layers[0];
            return
!layer.transparent &&
layer.opacity === 1.0 &&
```

```
!layer.forceOwnTexture &&
layer.visible;
      };
      Layout.prototype.updateAct
iveEffects = function ()
      {

      this.active_effect_types.l
ength = 0;
            var i, len, et;
            for (i = 0, len =
this.effect_types.length; i <
len; i++)
            {
                  et =
this.effect_types[i];
                  if (et.active)

      this.active_effect_types.p
ush(et);
            }
      };
      Layout.prototype.getEffect
ByName = function (name_)
      {
            var i, len, et;
            for (i = 0, len =
this.effect_types.length; i <
len; i++)
            {
                  et =
this.effect_types[i];
                  if (et.name ===
name_)
                        return
et;
            }
            return null;
      };
      var created_instances =
[];
      Layout.prototype.startRunn
ing = function ()
      {
            if (this.sheetname)
            {

      this.event_sheet =
this.runtime.eventsheets[this.s
heetname];
;
            }

      this.runtime.running_layou
t = this;
            this.scrollX =
(this.runtime.original_width /
2);
            this.scrollY =
(this.runtime.original_height /
2);
            var i, k, len, lenk,
type, type_instances, inst,
iid, t, s, p, q, type_data,
layer;
            for (i = 0, len =
this.runtime.types_by_index.len
gth; i < len; i++)
            {
                  type =
this.runtime.types_by_index[i];
                  if
(type.is_family)
                        continue;
            // instances are
only transferred for their real
type
                  type_instances
= type.instances;
                  for (k = 0,
lenk = type_instances.length; k
< lenk; k++)
                  {
                        inst =
type_instances[k];
                        if
(inst.layer)
                        {
                              var
num = inst.layer.number;
                              if
(num >= this.layers.length)

      num = this.layers.length -
1;

      inst.layer =
this.layers[num];

      inst.layer.instances.push(
inst);

      inst.layer.zindices_stale
= true;
                        }
                  }
            }
            var layer;

      created_instances.length =
0;
```

```
this.boundScrolling();
for (i = 0, len = this.layers.length; i < len; i++)
{
layer = this.layers[i];
layer.createInitialInstances();	// fills created_instances
layer.disableAngle = true;
var px = layer.canvasToLayer(0, 0, true, true);
var py = layer.canvasToLayer(0, 0, false, true);
layer.disableAngle = false;
if (this.runtime.pixel_rounding)
{
px = (px + 0.5) | 0;
py = (py + 0.5) | 0;
}
layer.rotateViewport(px, py, null);
}
var uids_changed = false;
if (!this.first_visit)
{
for (p in this.persist_data)
{
if (this.persist_data.hasOwnProperty(p))
{
type = this.runtime.getObjectTypeBySid(parseInt(p, 10));
if (!type || type.is_family || !this.runtime.typeHasPersistBehavior(type))
continue;
type_data = this.persist_data[p];
for (i = 0, len = type_data.length; i < len; i++)
{
layer = null;
if (type.plugin.is_world)
{
layer = this.getLayerBySid(type_data[i]["w"]["l"]);
if (!layer)
continue;
}
inst = this.runtime.createInstanceFromInit(type.default_instance, layer, false, 0, 0, true);
this.runtime.loadInstanceFromJSON(inst, type_data[i]);
uids_changed = true;
created_instances.push(inst);
}
type_data.length = 0;
}
}
for (i = 0, len = this.layers.length; i < len; i++)
{
this.layers[i].instances.sort(sortInstanceByZIndex);
this.layers[i].zindices_stale = true;	// in case of duplicates/holes
}
}
```

```
            if (uids_changed)
            {
      this.runtime.ClearDeathRow
();
      this.runtime.refreshUidMap
();
            }
            for (i = 0; i <
created_instances.length; i++)
            {
                  inst =
created_instances[i];
                  if
(!inst.type.is_contained)
                        continue;
                  iid =
inst.get_iid();
                  for (k = 0,
lenk =
inst.type.container.length; k <
lenk; k++)
                  {
                        t =
inst.type.container[k];
                        if
(inst.type === t)
      continue;
                        if
(t.instances.length > iid)
      inst.siblings.push(t.insta
nces[iid]);
                        else
                        {
                              if
(!t.default_instance)
                              {
                              }
                              else
                              {
      s =
this.runtime.createInstanceFrom
Init(t.default_instance,
inst.layer, true, inst.x,
inst.y, true);
      this.runtime.ClearDeathRow
();
      t.updateIIDs();
      inst.siblings.push(s);
      created_instances.push(s);
            // come back around
and link up its own instances
too
                              }
                        }
                  }
            }
            for (i = 0, len =
this.initial_nonworld.length; i
< len; i++)
            {
                  inst =
this.runtime.createInstanceFrom
Init(this.initial_nonworld[i],
null, true);
;
            }
      this.runtime.changelayout
= null;
      this.runtime.ClearDeathRow
();
            if (this.runtime.ctx
&& !this.runtime.isDomFree)
            {
                  for (i = 0, len
=
this.runtime.types_by_index.len
gth; i < len; i++)
                  {
                        t =
this.runtime.types_by_index[i];
                        if
(t.is_family ||
!t.instances.length ||
!t.preloadCanvas2D)
      continue;
      t.preloadCanvas2D(this.run
time.ctx);
                  }
            }
            /*
            if
(this.runtime.glwrap)
            {
      console.log("Estimated
VRAM at layout start: " +
this.runtime.glwrap.textureCoun
t() + " textures, approx. " +
Math.round(this.runtime.glwrap.
```

```
estimateVRAM() / 1024) + "
kb");
            }
            */
            for (i = 0, len =
created_instances.length; i <
len; i++)
            {
                  inst =
created_instances[i];

      this.runtime.trigger(Objec
t.getPrototypeOf(inst.type.plug
in).cnds.OnCreated, inst);
            }

      created_instances.length =
0;

      this.runtime.trigger(cr.sy
stem_object.prototype.cnds.OnLa
youtStart, null);
            this.first_visit =
false;
      };
      Layout.prototype.createGlo
balNonWorlds = function ()
      {
            var i, k, len,
initial_inst, inst, type;
            for (i = 0, k = 0,
len =
this.initial_nonworld.length; i
< len; i++)
            {
                  initial_inst =
this.initial_nonworld[i];
                  type =
this.runtime.types_by_index[ini
tial_inst[1]];
                  if
(type.global)
                        inst =
this.runtime.createInstanceFrom
Init(initial_inst, null, true);
                  else
                  {

      this.initial_nonworld[k] =
initial_inst;
                        k++;
                  }
            }

      this.initial_nonworld.leng
th = k;
      };
      Layout.prototype.stopRunni
ng = function ()
      {
;
            /*
            if
(this.runtime.glwrap)
            {

      console.log("Estimated
VRAM at layout end: " +
this.runtime.glwrap.textureCoun
t() + " textures, approx. " +
Math.round(this.runtime.glwrap.
estimateVRAM() / 1024) + "
kb");
            }
            */

      this.runtime.trigger(cr.sy
stem_object.prototype.cnds.OnLa
youtEnd, null);

      this.runtime.system.waits.
length = 0;
            var i, leni, j,
lenj;
            var layer_instances,
inst, type;
            for (i = 0, leni =
this.layers.length; i < leni;
i++)
            {
                  layer_instances
= this.layers[i].instances;
                  for (j = 0,
lenj = layer_instances.length;
j < lenj; j++)
                  {
                        inst =
layer_instances[j];
                        if
(!inst.type.global)
                        {
                              if
(this.runtime.typeHasPersistBeh
avior(inst.type))

      this.saveObjectToPersist(i
nst);

      this.runtime.DestroyInstan
ce(inst);
                        }
                  }
```

```
	this.runtime.ClearDeathRow
();

	layer_instances.length =
0;

	this.layers[i].zindices_st
ale = true;
		}
		for (i = 0, leni =
this.runtime.types_by_index.len
gth; i < leni; i++)
		{
			type =
this.runtime.types_by_index[i];
			if (type.global
|| type.plugin.is_world ||
type.plugin.singleglobal ||
type.is_family)
				continue;
			for (j = 0,
lenj = type.instances.length; j
< lenj; j++)

	this.runtime.DestroyInstan
ce(type.instances[j]);

	this.runtime.ClearDeathRow
();
		}
	};
	Layout.prototype.draw =
function (ctx)
	{
		var layout_canvas;
		var layout_ctx =
ctx;
		var ctx_changed =
false;
		var render_offscreen
=
!this.runtime.fullscreenScaling
Quality;
		if
(render_offscreen)
		{
			if
(!this.runtime.layout_canvas)
			{

	this.runtime.layout_canvas
=
document.createElement("canvas"
);

	layout_canvas =
this.runtime.layout_canvas;

	layout_canvas.width =
this.runtime.draw_width;

	layout_canvas.height =
this.runtime.draw_height;

	this.runtime.layout_ctx =
layout_canvas.getContext("2d");

	ctx_changed = true;
			}
			layout_canvas =
this.runtime.layout_canvas;
			layout_ctx =
this.runtime.layout_ctx;
			if
(layout_canvas.width !==
this.runtime.draw_width)
			{

	layout_canvas.width =
this.runtime.draw_width;

	ctx_changed = true;
			}
			if
(layout_canvas.height !==
this.runtime.draw_height)
			{

	layout_canvas.height =
this.runtime.draw_height;

	ctx_changed = true;
			}
			if
(ctx_changed)
			{

	layout_ctx["webkitImageSmo
othingEnabled"] =
this.runtime.linearSampling;

	layout_ctx["mozImageSmooth
ingEnabled"] =
this.runtime.linearSampling;

	layout_ctx["msImageSmoothi
ngEnabled"] =
this.runtime.linearSampling;

	layout_ctx["imageSmoothing
```

```
Enabled"] =
this.runtime.linearSampling;
                }
            }

      layout_ctx.globalAlpha =
1;

      layout_ctx.globalComposite
Operation = "source-over";
            if
(this.runtime.alphaBackground
&&
!this.hasOpaqueBottomLayer())

      layout_ctx.clearRect(0, 0,
this.runtime.draw_width,
this.runtime.draw_height);
            var i, len, l;
            for (i = 0, len =
this.layers.length; i < len;
i++)
            {
                  l =
this.layers[i];
                  if (l.visible
&& l.opacity > 0 &&
l.blend_mode !== 11)

      l.draw(layout_ctx);
            }
            if
(render_offscreen)
            {

      ctx.drawImage(layout_canva
s, 0, 0, this.runtime.width,
this.runtime.height);
            }
      };
      Layout.prototype.drawGL =
function (glw)
      {
            var
render_to_texture =
(this.active_effect_types.lengt
h > 0 ||


this.runtime.uses_background_bl
ending ||


!this.runtime.fullscreenScaling
Quality);
            if
(render_to_texture)
            {
                  if
(!this.runtime.layout_tex)
                  {

      this.runtime.layout_tex =
glw.createEmptyTexture(this.run
time.draw_width,
this.runtime.draw_height,
this.runtime.linearSampling);
                  }
                  if
(this.runtime.layout_tex.c2widt
h !== this.runtime.draw_width
||
this.runtime.layout_tex.c2heigh
t !== this.runtime.draw_height)
                  {

      glw.deleteTexture(this.run
time.layout_tex);

      this.runtime.layout_tex =
glw.createEmptyTexture(this.run
time.draw_width,
this.runtime.draw_height,
this.runtime.linearSampling);
                  }

      glw.setRenderingToTexture(
this.runtime.layout_tex);
                  if
(!this.runtime.fullscreenScalin
gQuality)
                  {

      glw.setSize(this.runtime.d
raw_width,
this.runtime.draw_height);
                  }
            }
            else
            {
                  if
(this.runtime.layout_tex)
                  {

      glw.setRenderingToTexture(
null);

      glw.deleteTexture(this.run
time.layout_tex);
```

```
     this.runtime.layout_tex =
null;
                }
           }
           if
(this.runtime.alphaBackground
&&
!this.hasOpaqueBottomLayer())
                glw.clear(0, 0,
0, 0);
           var i, len;
           for (i = 0, len =
this.layers.length; i < len;
i++)
           {
                if
(this.layers[i].visible &&
this.layers[i].opacity > 0)

     this.layers[i].drawGL(glw)
;
           }
           if
(render_to_texture)
           {
                if
(this.active_effect_types.lengt
h === 0 ||

     (this.active_effect_types.
length === 1 &&
this.runtime.fullscreenScalingQ
uality))
                {
                     if
(this.active_effect_types.lengt
h === 1)
                     {
                          var
etindex =
this.active_effect_types[0].ind
ex;

     glw.switchProgram(this.act
ive_effect_types[0].shaderindex
);

     glw.setProgramParameters(n
ull,
                // backTex


      1.0 /
this.runtime.draw_width,
     // pixelWidth

      1.0 /
this.runtime.draw_height,   //
pixelHeight


      0.0, 0.0,
                         //
destStart


      1.0, 1.0,
                         //
destEnd


      this.scale,
                         //
layerScale



this.effect_params[etindex]);
           // fx parameters
                              if
(glw.programIsAnimated(this.act
ive_effect_types[0].shaderindex
))

     this.runtime.redraw =
true;
                         }
                         else

     glw.switchProgram(0);
                         if
(!this.runtime.fullscreenScalin
gQuality)
                         {

     glw.setSize(this.runtime.w
idth, this.runtime.height);
                         }

     glw.setRenderingToTexture(
null);                        //
to backbuffer

     glw.setOpacity(1);

     glw.setTexture(this.runtim
e.layout_tex);

     glw.setAlphaBlend();
```

```
glw.resetModelView();

glw.updateModelView();
var halfw = this.runtime.width / 2;
var halfh = this.runtime.height / 2;

glw.quad(-halfw, halfh, halfw, halfh, halfw, -halfh, -halfw, -halfh);

glw.setTexture(null);
}
else
{

this.renderEffectChain(glw, null, null, null);
}
}
};
Layout.prototype.getRenderTarget = function()
{
return (this.active_effect_types.length > 0 ||

this.runtime.uses_background_blending ||

!this.runtime.fullscreenScalingQuality) ? this.runtime.layout_tex : null;
};
Layout.prototype.getMinLayerScale = function ()
{
var m = this.layers[0].getScale();
var i, len, l;
for (i = 1, len = this.layers.length; i < len; i++)
{
l = this.layers[i];
if (l.parallaxX === 0 && l.parallaxY === 0)
continue;
if (l.getScale() < m)
m = l.getScale();
}
return m;
};
Layout.prototype.scrollToX = function (x)
{
if (!this.unbounded_scrolling)
{
var widthBoundary = (this.runtime.draw_width * (1 / this.getMinLayerScale()) / 2);
if (x > this.width - widthBoundary)
x = this.width - widthBoundary;
if (x < widthBoundary)
x = widthBoundary;
}
if (this.scrollX !== x)
{
this.scrollX = x;

this.runtime.redraw = true;
}
};
Layout.prototype.scrollToY = function (y)
{
if (!this.unbounded_scrolling)
{
var heightBoundary = (this.runtime.draw_height * (1 / this.getMinLayerScale()) / 2);
if (y > this.height - heightBoundary)
y = this.height - heightBoundary;
if (y < heightBoundary)
y = heightBoundary;
}
if (this.scrollY !== y)
{
```

```
                this.scrollY =
y;

     this.runtime.redraw =
true;
          }
     };
     Layout.prototype.boundScro
lling = function ()
     {

     this.scrollToX(this.scroll
X);

     this.scrollToY(this.scroll
Y);
     };
     Layout.prototype.renderEff
ectChain = function (glw,
layer, inst, rendertarget)
     {
          var
active_effect_types = inst ?


     inst.active_effect_types :

          layer ?


     layer.active_effect_types
:


     this.active_effect_types;
          var layerScale =
inst ? inst.layer.getScale() :

          layer ?
layer.getScale() : 1;
          var fx_tex =
this.runtime.fx_tex;
          var i, len, last,
temp, fx_index = 0,
other_fx_index = 1;
          var y, h;
          var windowWidth =
this.runtime.draw_width;
          var windowHeight =
this.runtime.draw_height;
          var halfw =
windowWidth / 2;
          var halfh =
windowHeight / 2;
          var rcTex = layer ?
layer.rcTex : this.rcTex;
          var rcTex2 = layer ?
layer.rcTex2 : this.rcTex2;
          var screenleft = 0,
clearleft = 0;
          var screentop = 0,
cleartop = 0;
          var screenright =
windowWidth, clearright =
windowWidth;
          var screenbottom =
windowHeight, clearbottom =
windowHeight;
          var
boxExtendHorizontal = 0;
          var
boxExtendVertical = 0;
          var inst_layer_angle
= inst ? inst.layer.getAngle()
: 0;
          if (inst)
          {
               for (i = 0, len
= active_effect_types.length; i
< len; i++)
               {

     boxExtendHorizontal +=
glw.getProgramBoxExtendHorizont
al(active_effect_types[i].shade
rindex);

     boxExtendVertical +=
glw.getProgramBoxExtendVertical
(active_effect_types[i].shaderi
ndex);
               }
               var bbox =
inst.bbox;
               screenleft =
layer.layerToCanvas(bbox.left,
bbox.top, true, true);
               screentop =
layer.layerToCanvas(bbox.left,
bbox.top, false, true);
               screenright =
layer.layerToCanvas(bbox.right,
bbox.bottom, true, true);
               screenbottom =
layer.layerToCanvas(bbox.right,
bbox.bottom, false, true);
               if
(inst_layer_angle !== 0)
               {
                    var
screentrx =
```

```
layer.layerToCanvas(bbox.right,
bbox.top, true, true);
                        var
screentry =
layer.layerToCanvas(bbox.right,
bbox.top, false, true);
                        var
screenblx =
layer.layerToCanvas(bbox.left,
bbox.bottom, true, true);
                        var
screenbly =
layer.layerToCanvas(bbox.left,
bbox.bottom, false, true);
                        temp =
Math.min(screenleft,
screenright, screentrx,
screenblx);

      screenright =
Math.max(screenleft,
screenright, screentrx,
screenblx);

      screenleft = temp;
                        temp =
Math.min(screentop,
screenbottom, screentry,
screenbly);

      screenbottom =
Math.max(screentop,
screenbottom, screentry,
screenbly);
                        screentop
= temp;
                }
                screenleft -=
boxExtendHorizontal;
                screentop -=
boxExtendVertical;
                screenright +=
boxExtendHorizontal;
                screenbottom +=
boxExtendVertical;
                rcTex2.left =
screenleft / windowWidth;
                rcTex2.top = 1
- screentop / windowHeight;
                rcTex2.right =
screenright / windowWidth;
                rcTex2.bottom =
1 - screenbottom /
windowHeight;
                clearleft =
screenleft =
cr.floor(screenleft);
                cleartop =
screentop =
cr.floor(screentop);
                clearright =
screenright =
cr.ceil(screenright);
                clearbottom =
screenbottom =
cr.ceil(screenbottom);
                clearleft -=
boxExtendHorizontal;
                cleartop -=
boxExtendVertical;
                clearright +=
boxExtendHorizontal;
                clearbottom +=
boxExtendVertical;
                if (screenleft
< 0)
      screenleft = 0;
                if (screentop <
0)
      screentop = 0;
                if (screenright
> windowWidth)
      screenright = windowWidth;
                if
(screenbottom > windowHeight)
      screenbottom =
windowHeight;
                if (clearleft <
0)
      clearleft = 0;
                if (cleartop <
0)
      cleartop = 0;
                if (clearright
> windowWidth)
      clearright = windowWidth;
                if (clearbottom
> windowHeight)
      clearbottom =
windowHeight;
                rcTex.left =
screenleft / windowWidth;
                rcTex.top = 1 -
screentop / windowHeight;
                rcTex.right =
screenright / windowWidth;
                rcTex.bottom =
1 - screenbottom /
windowHeight;
            }
```

```
            else
            {
                  rcTex.left =
rcTex2.left = 0;
                  rcTex.top =
rcTex2.top = 0;
                  rcTex.right =
rcTex2.right = 1;
                  rcTex.bottom =
rcTex2.bottom = 1;
            }
            var pre_draw = (inst
&& (((inst.angle ||
inst_layer_angle) &&
glw.programUsesDest(active_effe
ct_types[0].shaderindex)) ||
boxExtendHorizontal !== 0 ||
boxExtendVertical !== 0 ||
inst.opacity !== 1 ||
inst.type.plugin.must_predraw))
|| (layer && !inst &&
layer.opacity !== 1);
            glw.setAlphaBlend();
            if (pre_draw)
            {
                  if
(!fx_tex[fx_index])
                  {

      fx_tex[fx_index] =
glw.createEmptyTexture(windowWi
dth, windowHeight,
this.runtime.linearSampling);
                  }
                  if
(fx_tex[fx_index].c2width !==
windowWidth ||
fx_tex[fx_index].c2height !==
windowHeight)
                  {

      glw.deleteTexture(fx_tex[f
x_index]);

      fx_tex[fx_index] =
glw.createEmptyTexture(windowWi
dth, windowHeight,
this.runtime.linearSampling);
                  }

      glw.switchProgram(0);

      glw.setRenderingToTexture(
fx_tex[fx_index]);
                  h = clearbottom
- cleartop;
                  y =
(windowHeight - cleartop) - h;

      glw.clearRect(clearleft,
y, clearright - clearleft, h);
                  if (inst)
                  {

      inst.drawGL(glw);
                  }
                  else
                  {

      glw.setTexture(this.runtim
e.layer_tex);

      glw.setOpacity(layer.opaci
ty);

      glw.resetModelView();

      glw.translate(-halfw, -
halfh);

      glw.updateModelView();

      glw.quadTex(screenleft,
screenbottom, screenright,
screenbottom, screenright,
screentop, screenleft,
screentop, rcTex);
                  }
                  rcTex2.left =
rcTex2.top = 0;
                  rcTex2.right =
rcTex2.bottom = 1;
                  if (inst)
                  {
                        temp =
rcTex.top;
                        rcTex.top
= rcTex.bottom;

      rcTex.bottom = temp;
                  }
                  fx_index = 1;
                  other_fx_index
= 0;
            }
            glw.setOpacity(1);
            var last =
active_effect_types.length - 1;
            var post_draw =
glw.programUsesCrossSampling(ac
tive_effect_types[last].shaderi
ndex) ||
```

```
(!layer && !inst &&
!this.runtime.fullscreenScaling
Quality);
          var etindex = 0;
          for (i = 0, len =
active_effect_types.length; i <
len; i++)
          {
               if
(!fx_tex[fx_index])
               {

     fx_tex[fx_index] =
glw.createEmptyTexture(windowWi
dth, windowHeight,
this.runtime.linearSampling);
               }
               if
(fx_tex[fx_index].c2width !==
windowWidth ||
fx_tex[fx_index].c2height !==
windowHeight)
               {

     glw.deleteTexture(fx_tex[f
x_index]);

     fx_tex[fx_index] =
glw.createEmptyTexture(windowWi
dth, windowHeight,
this.runtime.linearSampling);
               }

     glw.switchProgram(active_e
ffect_types[i].shaderindex);
               etindex =
active_effect_types[i].index;
               if
(glw.programIsAnimated(active_e
ffect_types[i].shaderindex))

     this.runtime.redraw =
true;
               if (i == 0 &&
!pre_draw)
               {

     glw.setRenderingToTexture(
fx_tex[fx_index]);
                    h =
clearbottom - cleartop;
                    y =
(windowHeight - cleartop) - h;

     glw.clearRect(clearleft,
y, clearright - clearleft, h);
                    if (inst)
                    {

     glw.setProgramParameters(r
endertarget,
          // backTex

      1.0 / inst.width,
                // pixelWidth

      1.0 / inst.height,
                // pixelHeight

      rcTex2.left, rcTex2.top,
          // destStart

      rcTex2.right,
rcTex2.bottom,   // destEnd

      layerScale,

inst.effect_params[etindex]);
     // fx params

     inst.drawGL(glw);
                    }
                    else
                    {

     glw.setProgramParameters(r
endertarget,
          // backTex

      1.0 / windowWidth,
                // pixelWidth

      1.0 / windowHeight,
          // pixelHeight

      0.0, 0.0,
                // destStart
```

```
1.0, 1.0,
          // destEnd
layerScale,
layer ?
          // fx params
layer.effect_params[etinde
x] :
this.effect_params[etindex
]);
glw.setTexture(layer ?
this.runtime.layer_tex :
this.runtime.layout_tex);
glw.resetModelView();
glw.translate(-halfw, -
halfh);
glw.updateModelView();
glw.quadTex(screenleft,
screenbottom, screenright,
screenbottom, screenright,
screentop, screenleft,
screentop, rcTex);
                    }
rcTex2.left = rcTex2.top =
0;
rcTex2.right =
rcTex2.bottom = 1;
                    if (inst
&& !post_draw)
                    {
                         temp
= screenbottom;
screenbottom = screentop;
screentop = temp;
                    }
          }
          else
          {
```

```
glw.setProgramParameters(r
endertarget,
                    // backTex
                                   1.0
/ windowWidth,
          // pixelWidth
                                   1.0
/ windowHeight,
          // pixelHeight
rcTex2.left, rcTex2.top,
          // destStart
rcTex2.right, rcTex2.bottom,
          // destEnd
layerScale,
inst ?
                    // fx
params
          inst.effect_params[etindex
] :
          layer ?
          layer.effect_params[etinde
x] :
          this.effect_params[etindex
]);
          glw.setTexture(null);
                    if (i ===
last && !post_draw)
                    {
                         if
(inst)
          glw.setBlend(inst.srcBlend
, inst.destBlend);
```

```
                                else
if (layer)
     glw.setBlend(layer.srcBlen
d, layer.destBlend);

     glw.setRenderingToTexture(
rendertarget);
                        }
                        else
                        {

     glw.setRenderingToTexture(
fx_tex[fx_index]);
                                h =
clearbottom - cleartop;
                                y =
(windowHeight - cleartop) - h;

     glw.clearRect(clearleft,
y, clearright - clearleft, h);
                        }

     glw.setTexture(fx_tex[othe
r_fx_index]);

     glw.resetModelView();

     glw.translate(-halfw, -
halfh);

     glw.updateModelView();

     glw.quadTex(screenleft,
screenbottom, screenright,
screenbottom, screenright,
screentop, screenleft,
screentop, rcTex);
                        if (i ===
last && !post_draw)

     glw.setTexture(null);
                    }
                    fx_index =
(fx_index === 0 ? 1 : 0);
                    other_fx_index
= (fx_index === 0 ? 1 : 0);
     // will be opposite to
fx_index since it was just
assigned
               }
               if (post_draw)
               {

     glw.switchProgram(0);
                    if (inst)

     glw.setBlend(inst.srcBlend
, inst.destBlend);
                    else if (layer)

     glw.setBlend(layer.srcBlen
d, layer.destBlend);
                    else
                    {
                         if
(!this.runtime.fullscreenScalin
gQuality)
                         {

     glw.setSize(this.runtime.w
idth, this.runtime.height);

     halfw = this.runtime.width
/ 2;

     halfh =
this.runtime.height / 2;

     screenleft = 0;

     screentop = 0;

     screenright =
this.runtime.width;

     screenbottom =
this.runtime.height;
                         }
                    }

     glw.setRenderingToTexture(
rendertarget);

     glw.setTexture(fx_tex[othe
r_fx_index]);

     glw.resetModelView();
                    glw.translate(-
halfw, -halfh);

     glw.updateModelView();
                    if (inst &&
active_effect_types.length ===
1 && !pre_draw)

     glw.quadTex(screenleft,
screentop, screenright,
screentop, screenright,
screenbottom, screenleft,
screenbottom, rcTex);
                    else
```

```
	glw.quadTex(screenleft,
screenbottom, screenright,
screenbottom, screenright,
screentop, screenleft,
screentop, rcTex);

	glw.setTexture(null);
		}
	};
	Layout.prototype.getLayerB
ySid = function (sid_)
	{
		var i, len;
		for (i = 0, len =
this.layers.length; i < len;
i++)
		{
			if
(this.layers[i].sid === sid_)
				return
this.layers[i];
		}
		return null;
	};
	Layout.prototype.saveToJSO
N = function ()
	{
		var i, len, layer,
et;
		var o = {
			"sx":
this.scrollX,
			"sy":
this.scrollY,
			"s":
this.scale,
			"a":
this.angle,
			"w":
this.width,
			"h":
this.height,
			"fv":
this.first_visit,
	// added r127
			"persist":
this.persist_data,
			"fx": [],
			"layers": {}
		};
		for (i = 0, len =
this.effect_types.length; i <
len; i++)
		{
			et =
this.effect_types[i];

	o["fx"].push({"name":
et.name, "active": et.active,
"params":
this.effect_params[et.index]
});
		}
		for (i = 0, len =
this.layers.length; i < len;
i++)
		{
			layer =
this.layers[i];

	o["layers"][layer.sid.toSt
ring()] = layer.saveToJSON();
		}
		return o;
	};
	Layout.prototype.loadFromJ
SON = function (o)
	{
		var i, len, fx, p,
layer;
		this.scrollX =
o["sx"];
		this.scrollY =
o["sy"];
		this.scale = o["s"];
		this.angle = o["a"];
		this.width = o["w"];
		this.height =
o["h"];
		this.persist_data =
o["persist"];
		if (typeof o["fv"]
!== "undefined")

	this.first_visit =
o["fv"];
		var ofx = o["fx"];
		for (i = 0, len =
ofx.length; i < len; i++)
		{
			fx =
this.getEffectByName(ofx[i]["na
me"]);
			if (!fx)
				continue;
		// must've gone
missing
			fx.active =
ofx[i]["active"];
```

```
	this.effect_params[fx.inde
x] = ofx[i]["params"];
		}

	this.updateActiveEffects()
;
		var olayers =
o["layers"];
		for (p in olayers)
		{
			if
(olayers.hasOwnProperty(p))
			{
				layer =
this.getLayerBySid(parseInt(p,
10));
				if
(!layer)

	continue;	//
must've gone missing

	layer.loadFromJSON(olayers
[p]);
			}
		}
	};
	cr.layout = Layout;
	function Layer(layout, m)
	{
		this.layout =
layout;
		this.runtime =
layout.runtime;
		this.instances = [];
// running instances
		this.scale = 1.0;
		this.angle = 0;
		this.disableAngle =
false;
		this.tmprect = new
cr.rect(0, 0, 0, 0);
		this.tmpquad = new
cr.quad();
		this.viewLeft = 0;
		this.viewRight = 0;
		this.viewTop = 0;
		this.viewBottom = 0;
		this.zindices_stale
= false;
		this.name = m[0];
		this.index = m[1];
		this.sid = m[2];
		this.visible = m[3];
		// initially visible
	this.background_color =
m[4];
		this.transparent =
m[5];
		this.parallaxX =
m[6];
		this.parallaxY =
m[7];
		this.opacity = m[8];
		this.forceOwnTexture
= m[9];
		this.zoomRate =
m[10];
		this.blend_mode =
m[11];
		this.effect_fallback
= m[12];
		this.compositeOp =
"source-over";
		this.srcBlend = 0;
		this.destBlend = 0;

	this.render_offscreen =
false;
		var im = m[13];
		var i, len;

	this.initial_instances =
[];
		for (i = 0, len =
im.length; i < len; i++)
		{
			var inst =
im[i];
			var type =
this.runtime.types_by_index[ins
t[1]];
;
			if
(!type.default_instance)

	type.default_instance =
inst;

	this.initial_instances.pus
h(inst);
			if
(this.layout.initial_types.inde
xOf(type) === -1)

	this.layout.initial_types.
push(type);
		}
		this.effect_types =
[];
```

```
	this.active_effect_types = [];
		this.effect_params = [];
		for (i = 0, len = m[14].length; i < len; i++)
		{
	this.effect_types.push({
				id: m[14][i][0],
				name: m[14][i][1],
	shaderindex: -1,
				active: true,
				index: i
			});
	this.effect_params.push(m[14][i][2].slice(0));
		}
	this.updateActiveEffects();
		this.rcTex = new cr.rect(0, 0, 1, 1);
		this.rcTex2 = new cr.rect(0, 0, 1, 1);
	};
	Layer.prototype.updateActiveEffects = function ()
	{
	this.active_effect_types.length = 0;
		var i, len, et;
		for (i = 0, len = this.effect_types.length; i < len; i++)
		{
			et = this.effect_types[i];
			if (et.active)
	this.active_effect_types.push(et);
		}
	};
	Layer.prototype.getEffectByName = function (name_)
	{
		var i, len, et;
		for (i = 0, len = this.effect_types.length; i < len; i++)
		{
			et = this.effect_types[i];
			if (et.name === name_)
				return et;
		}
		return null;
	};
	Layer.prototype.createInitialInstances = function ()
	{
		var i, k, len, inst, initial_inst, type, keep, hasPersistBehavior;
		for (i = 0, k = 0, len = this.initial_instances.length; i < len; i++)
		{
			initial_inst = this.initial_instances[i];
			type = this.runtime.types_by_index[initial_inst[1]];
;

	hasPersistBehavior = this.runtime.typeHasPersistBehavior(type);
			keep = true;
			if (!hasPersistBehavior || this.layout.first_visit)
			{
				inst = this.runtime.createInstanceFromInit(initial_inst, this, true);
;

	created_instances.push(inst);
				if (inst.type.global)
					keep = false;
			}
			if (keep)
			{

	this.initial_instances[k] = this.initial_instances[i];
```

```
                    k++;
                }
            }

        this.initial_instances.length = k;

        this.runtime.ClearDeathRow();        // flushes creation row so IIDs will be correct
            if (!this.runtime.glwrap && this.effect_types.length)   // no WebGL renderer and shaders used
                this.blend_mode = this.effect_fallback;
                // use fallback blend mode
            this.compositeOp = cr.effectToCompositeOp(this.blend_mode);
            if (this.runtime.gl)

        cr.setGLBlend(this, this.blend_mode, this.runtime.gl);
        };
        Layer.prototype.updateZIndices = function ()
        {
            if (!this.zindices_stale)
                return;
            var i, len;
            for (i = 0, len = this.instances.length; i < len; i++)
            {
;
;

        this.instances[i].zindex = i;
            }
            this.zindices_stale = false;
        };
        Layer.prototype.getScale = function (include_aspect)
        {
            return this.getNormalScale() * (this.runtime.fullscreenScalingQuality || include_aspect ? this.runtime.aspect_scale : 1);
        };
        Layer.prototype.getNormalScale = function ()
        {
            return ((this.scale * this.layout.scale) - 1) * this.zoomRate + 1;
        };
        Layer.prototype.getAngle = function ()
        {
            if (this.disableAngle)
                return 0;
            return cr.clamp_angle(this.layout.angle + this.angle);
        };
        Layer.prototype.draw = function (ctx)
        {

        this.render_offscreen = (this.forceOwnTexture || this.opacity !== 1.0 || this.blend_mode !== 0);
            var layer_canvas = this.runtime.canvas;
            var layer_ctx = ctx;
            var ctx_changed = false;
            ctx.globalAlpha = 1;

        ctx.globalCompositeOperation = "source-over";
            if (this.render_offscreen)
            {
                if (!this.runtime.layer_canvas)
                {

        this.runtime.layer_canvas = document.createElement("canvas");
;

        layer_canvas = this.runtime.layer_canvas;

        layer_canvas.width = this.runtime.draw_width;

        layer_canvas.height = this.runtime.draw_height;
```

```
	this.runtime.layer_ctx = layer_canvas.getContext("2d");
;

	ctx_changed = true;
		}
		layer_canvas = this.runtime.layer_canvas;
		layer_ctx = this.runtime.layer_ctx;
		if (layer_canvas.width !== this.runtime.draw_width)
		{

	layer_canvas.width = this.runtime.draw_width;

	ctx_changed = true;
		}
		if (layer_canvas.height !== this.runtime.draw_height)
		{

	layer_canvas.height = this.runtime.draw_height;

	ctx_changed = true;
		}
		if (ctx_changed)
		{

	layer_ctx["webkitImageSmoothingEnabled"] = this.runtime.linearSampling;

	layer_ctx["mozImageSmoothingEnabled"] = this.runtime.linearSampling;

	layer_ctx["msImageSmoothingEnabled"] = this.runtime.linearSampling;

	layer_ctx["imageSmoothingEnabled"] = this.runtime.linearSampling;
		}
		if (this.transparent)

	layer_ctx.clearRect(0, 0, this.runtime.draw_width, this.runtime.draw_height);
		}
		if (!this.transparent)
		{

	layer_ctx.fillStyle = "rgb(" + this.background_color[0] + "," + this.background_color[1] + "," + this.background_color[2] + ")";

	layer_ctx.fillRect(0, 0, this.runtime.draw_width, this.runtime.draw_height);
		}
		layer_ctx.save();
		this.disableAngle = true;
		var px = this.canvasToLayer(0, 0, true, true);
		var py = this.canvasToLayer(0, 0, false, true);
		this.disableAngle = false;
		if (this.runtime.pixel_rounding)
		{
			px = (px + 0.5) | 0;
			py = (py + 0.5) | 0;
		}

	this.rotateViewport(px, py, layer_ctx);
		var myscale = this.getScale();

	layer_ctx.scale(myscale, myscale);

	layer_ctx.translate(-px, -py);
		var i, len, inst, bbox;
		for (i = 0, len = this.instances.length; i < len; i++)
		{
			inst = this.instances[i];
```

```
                if
(!inst.visible || inst.width
=== 0 || inst.height === 0)
                        continue;

     inst.update_bbox();
                bbox =
inst.bbox;
                if (bbox.right
< this.viewLeft || bbox.bottom
< this.viewTop || bbox.left >
this.viewRight || bbox.top >
this.viewBottom)
                        continue;

     layer_ctx.globalCompositeO
peration = inst.compositeOp;

     inst.draw(layer_ctx);
           }
           layer_ctx.restore();
           if
(this.render_offscreen)
           {

     ctx.globalCompositeOperati
on = this.compositeOp;
                ctx.globalAlpha
= this.opacity;

     ctx.drawImage(layer_canvas
, 0, 0);
           }
     };
     Layer.prototype.rotateView
port = function (px, py, ctx)
     {
           var myscale =
this.getScale();
           this.viewLeft = px;
           this.viewTop = py;
           this.viewRight = px
+ (this.runtime.draw_width * (1
/ myscale));
           this.viewBottom = py
+ (this.runtime.draw_height *
(1 / myscale));
           var myAngle =
this.getAngle();
           if (myAngle !== 0)
           {
                if (ctx)
                {

     ctx.translate(this.runtime
.draw_width / 2,
this.runtime.draw_height / 2);

     ctx.rotate(-myAngle);

     ctx.translate(this.runtime
.draw_width / -2,
this.runtime.draw_height / -2);
                }

     this.tmprect.set(this.view
Left, this.viewTop,
this.viewRight,
this.viewBottom);

     this.tmprect.offset((this.
viewLeft + this.viewRight) / -
2, (this.viewTop +
this.viewBottom) / -2);

     this.tmpquad.set_from_rota
ted_rect(this.tmprect,
myAngle);

     this.tmpquad.bounding_box(
this.tmprect);

     this.tmprect.offset((this.
viewLeft + this.viewRight) / 2,
(this.viewTop +
this.viewBottom) / 2);
                this.viewLeft =
this.tmprect.left;
                this.viewTop =
this.tmprect.top;
                this.viewRight
= this.tmprect.right;
                this.viewBottom
= this.tmprect.bottom;
           }
     }
     Layer.prototype.drawGL =
function (glw)
     {
           var windowWidth =
this.runtime.draw_width;
           var windowHeight =
this.runtime.draw_height;
           var shaderindex = 0;
           var etindex = 0;

     this.render_offscreen =
(this.forceOwnTexture ||
this.opacity !== 1.0 ||
this.active_effect_types.length
> 0 || this.blend_mode !== 0);
```

```
            if
(this.render_offscreen)
            {
                  if
(!this.runtime.layer_tex)
                  {

      this.runtime.layer_tex =
glw.createEmptyTexture(this.run
time.draw_width,
this.runtime.draw_height,
this.runtime.linearSampling);
                  }
                  if
(this.runtime.layer_tex.c2width
!== this.runtime.draw_width ||
this.runtime.layer_tex.c2height
!== this.runtime.draw_height)
                  {

      glw.deleteTexture(this.run
time.layer_tex);

      this.runtime.layer_tex =
glw.createEmptyTexture(this.run
time.draw_width,
this.runtime.draw_height,
this.runtime.linearSampling);
                  }

      glw.setRenderingToTexture(
this.runtime.layer_tex);
                  if
(this.transparent)

      glw.clear(0, 0, 0, 0);
            }
            if
(!this.transparent)
            {

      glw.clear(this.background_
color[0] / 255,
this.background_color[1] / 255,
this.background_color[2] / 255,
1);
            }
            this.disableAngle =
true;
            var px =
this.canvasToLayer(0, 0, true,
true);
            var py =
this.canvasToLayer(0, 0, false,
true);
            this.disableAngle =
false;
            if
(this.runtime.pixel_rounding)
            {
                  px = (px + 0.5)
| 0;
                  py = (py + 0.5)
| 0;
            }

      this.rotateViewport(px,
py, null);
            var myscale =
this.getScale();

      glw.resetModelView();
            glw.scale(myscale,
myscale);
            glw.rotateZ(-
this.getAngle());

      glw.translate((this.viewLe
ft + this.viewRight) / -2,
(this.viewTop +
this.viewBottom) / -2);

      glw.updateModelView();
            var i, len, inst,
bbox;
            for (i = 0, len =
this.instances.length; i < len;
i++)
            {
                  inst =
this.instances[i];
                  if
(!inst.visible || inst.width
=== 0 || inst.height === 0)
                        continue;

      inst.update_bbox();
                  bbox =
inst.bbox;
                  if (bbox.right
< this.viewLeft || bbox.bottom
< this.viewTop || bbox.left >
this.viewRight || bbox.top >
this.viewBottom)
                        continue;
                  if
(inst.uses_shaders)
                  {

      shaderindex =
```

```
inst.active_effect_types[0].sha
derindex;
                        etindex =
inst.active_effect_types[0].ind
ex;
                        if
(inst.active_effect_types.lengt
h === 1 &&
!glw.programUsesCrossSampling(s
haderindex) &&

      !glw.programExtendsBox(sha
derindex) && ((!inst.angle &&
!inst.layer.getAngle()) ||
!glw.programUsesDest(shaderinde
x)) &&

      inst.opacity === 1 &&
!inst.type.plugin.must_predraw)
                        {

      glw.switchProgram(shaderin
dex);

      glw.setBlend(inst.srcBlend
, inst.destBlend);
                              if
(glw.programIsAnimated(shaderin
dex))

      this.runtime.redraw =
true;
                              var
destStartX = 0, destStartY = 0,
destEndX = 0, destEndY = 0;
                              if
(glw.programUsesDest(shaderinde
x))
                              {

      var bbox = inst.bbox;

      var screenleft =
this.layerToCanvas(bbox.left,
bbox.top, true, true);

      var screentop =
this.layerToCanvas(bbox.left,
bbox.top, false, true);

      var screenright =
this.layerToCanvas(bbox.right,
bbox.bottom, true, true);

      var screenbottom =
this.layerToCanvas(bbox.right,
bbox.bottom, false, true);

      destStartX = screenleft /
windowWidth;

      destStartY = 1 - screentop
/ windowHeight;

      destEndX = screenright /
windowWidth;

      destEndY = 1 -
screenbottom / windowHeight;
                              }

      glw.setProgramParameters(t
his.render_offscreen ?
this.runtime.layer_tex :
this.layout.getRenderTarget(),
// backTex


       1.0 / inst.width,
            // pixelWidth


       1.0 / inst.height,
            // pixelHeight


       destStartX, destStartY,


       destEndX, destEndY,


       this.getScale(),



inst.effect_params[etindex]);

      inst.drawGL(glw);
                        }
                        else
                        {

      this.layout.renderEffectCh
ain(glw, this, inst,
this.render_offscreen ?
this.runtime.layer_tex :
this.layout.getRenderTarget());

      glw.resetModelView();
```

```
     glw.scale(myscale,
myscale);

     glw.rotateZ(-
this.getAngle());

     glw.translate((this.viewLe
ft + this.viewRight) / -2,
(this.viewTop +
this.viewBottom) / -2);

     glw.updateModelView();
                        }
                }
                else
                {

     glw.switchProgram(0);
     // un-set any previously
set shader

     glw.setBlend(inst.srcBlend
, inst.destBlend);

     inst.drawGL(glw);
                }
           }
           if
(this.render_offscreen)
           {
                shaderindex =
this.active_effect_types.length
?
this.active_effect_types[0].sha
derindex : 0;
                etindex =
this.active_effect_types.length
?
this.active_effect_types[0].ind
ex : 0;
                if
(this.active_effect_types.lengt
h === 0 ||
(this.active_effect_types.lengt
h === 1 &&

     !glw.programUsesCrossSampl
ing(shaderindex) &&
this.opacity === 1))
                {
                        if
(this.active_effect_types.lengt
h === 1)
                        {

     glw.switchProgram(shaderin
dex);

     glw.setProgramParameters(t
his.layout.getRenderTarget(),
           // backTex


      1.0 /
this.runtime.draw_width,
     // pixelWidth


      1.0 /
this.runtime.draw_height,   //
pixelHeight


      0.0, 0.0,
                        //
destStart


      1.0, 1.0,
                        //
destEnd


      this.getScale(),
                        //
layerScale



this.effect_params[etindex]);
           // fx parameters
                             if
(glw.programIsAnimated(shaderin
dex))

     this.runtime.redraw =
true;
                        }
                        else

     glw.switchProgram(0);

     glw.setRenderingToTexture(
this.layout.getRenderTarget());

     glw.setOpacity(this.opacit
y);

     glw.setTexture(this.runtim
e.layer_tex);
```

```
	glw.setBlend(this.srcBlend
, this.destBlend);

	glw.resetModelView();

	glw.updateModelView();
				var halfw
= this.runtime.draw_width / 2;
				var halfh
= this.runtime.draw_height / 2;

	glw.quad(-halfw, halfh,
halfw, halfh, halfw, -halfh, -
halfw, -halfh);

	glw.setTexture(null);
			}
			else
			{

	this.layout.renderEffectCh
ain(glw, this, null,
this.layout.getRenderTarget());
			}
		}
	};
	Layer.prototype.canvasToLa
yer = function (ptx, pty, getx,
using_draw_area)
	{
		var multiplier =
this.runtime.devicePixelRatio;
		if
(this.runtime.isRetina)
		{
			ptx *=
multiplier;
			pty *=
multiplier;
		}
		var ox =
this.runtime.parallax_x_origin;
		var oy =
this.runtime.parallax_y_origin;
		var x =
((this.layout.scrollX - ox) *
this.parallaxX) + ox;
		var y =
((this.layout.scrollY - oy) *
this.parallaxY) + oy;
		var invScale = 1 /
this.getScale(!using_draw_area)
;
		if (using_draw_area)
		{
			x -=
(this.runtime.draw_width *
invScale) / 2;
			y -=
(this.runtime.draw_height *
invScale) / 2;
		}
		else
		{
			x -=
(this.runtime.width * invScale)
/ 2;
			y -=
(this.runtime.height *
invScale) / 2;
		}
		x += ptx * invScale;
		y += pty * invScale;
		var a =
this.getAngle();
		if (a !== 0)
		{
			x -=
this.layout.scrollX;
			y -=
this.layout.scrollY;
			var cosa =
Math.cos(a);
			var sina =
Math.sin(a);
			var x_temp = (x
* cosa) - (y * sina);
			y = (y * cosa)
+ (x * sina);
			x = x_temp;
			x +=
this.layout.scrollX;
			y +=
this.layout.scrollY;
		}
		return getx ? x : y;
	};
	Layer.prototype.layerToCan
vas = function (ptx, pty, getx,
using_draw_area)
	{
		var a =
this.getAngle();
		if (a !== 0)
		{
			ptx -=
this.layout.scrollX;
			pty -=
this.layout.scrollY;
			var cosa =
Math.cos(-a);
```

```
                var sina =
Math.sin(-a);
                var x_temp =
(ptx * cosa) - (pty * sina);
                pty = (pty *
cosa) + (ptx * sina);
                ptx = x_temp;
                ptx +=
this.layout.scrollX;
                pty +=
this.layout.scrollY;
           }
           var ox =
this.runtime.parallax_x_origin;
           var oy =
this.runtime.parallax_y_origin;
           var x =
((this.layout.scrollX - ox) *
this.parallaxX) + ox;
           var y =
((this.layout.scrollY - oy) *
this.parallaxY) + oy;
           var invScale = 1 /
this.getScale(!using_draw_area)
;
           if (using_draw_area)
           {
                x -=
(this.runtime.draw_width *
invScale) / 2;
                y -=
(this.runtime.draw_height *
invScale) / 2;
           }
           else
           {
                x -=
(this.runtime.width * invScale)
/ 2;
                y -=
(this.runtime.height *
invScale) / 2;
           }
           x = (ptx - x) /
invScale;
           y = (pty - y) /
invScale;
           var multiplier =
this.runtime.devicePixelRatio;
           if
(this.runtime.isRetina)
           {
                x /=
multiplier;
                y /=
multiplier;
           }
           return getx ? x : y;
      };
      Layer.prototype.rotatePt =
function (x_, y_, getx)
      {
           if (this.getAngle()
=== 0)
                return getx ?
x_ : y_;
           var nx =
this.layerToCanvas(x_, y_,
true);
           var ny =
this.layerToCanvas(x_, y_,
false);
           this.disableAngle =
true;
           var px =
this.canvasToLayer(nx, ny,
true);
           var py =
this.canvasToLayer(nx, ny,
true);
           this.disableAngle =
false;
           return getx ? px :
py;
      };
      Layer.prototype.saveToJSON
= function ()
      {
           var i, len, et;
           var o = {
                "s":
this.scale,
                "a":
this.angle,
                "vl":
this.viewLeft,
                "vt":
this.viewTop,
                "vr":
this.viewRight,
                "vb":
this.viewBottom,
                "v":
this.visible,
                "bc":
this.background_color,
                "t":
this.transparent,
                "px":
this.parallaxX,
                "py":
this.parallaxY,
```

```
                "o":
this.opacity,
                "zr":
this.zoomRate,
                "fx": [],
                "instances": []
            };
            for (i = 0, len =
this.effect_types.length; i <
len; i++)
            {
                et =
this.effect_types[i];

        o["fx"].push({"name":
et.name, "active": et.active,
"params":
this.effect_params[et.index]
});
            }
            return o;
        };
        function
sortInstanceByZIndex(a, b)
        {
            return a.zindex -
b.zindex;
        };
        Layer.prototype.loadFromJS
ON = function (o)
        {
            var i, len, p, inst,
fx;
            this.scale = o["s"];
            this.angle = o["a"];
            this.viewLeft =
o["vl"];
            this.viewTop =
o["vt"];
            this.viewRight =
o["vr"];
            this.viewBottom =
o["vb"];
            this.visible =
o["v"];

        this.background_color =
o["bc"];
            this.transparent =
o["t"];
            this.parallaxX =
o["px"];
            this.parallaxY =
o["py"];
            this.opacity =
o["o"];
            this.zoomRate =
o["zr"];
            var ofx = o["fx"];
            for (i = 0, len =
ofx.length; i < len; i++)
            {
                fx =
this.getEffectByName(ofx[i]["na
me"]);
                if (!fx)
                    continue;
            // must've gone
missing
                fx.active =
ofx[i]["active"];

        this.effect_params[fx.inde
x] = ofx[i]["params"];
            }

        this.updateActiveEffects()
;

        this.instances.sort(sortIn
stanceByZIndex);
            this.zindices_stale
= true;
        };
        cr.layer = Layer;
}()));
;
(function()
{
        var allUniqueSolModifiers
= [];
        function
testSolsMatch(arr1, arr2)
        {
            var i, len =
arr1.length;
            switch (len) {
            case 0:
                return true;
            case 1:
                return arr1[0]
=== arr2[0];
            case 2:
                return arr1[0]
=== arr2[0] && arr1[1] ===
arr2[1];
            default:
                for (i = 0; i <
len; i++)
                {
                    if
(arr1[i] !== arr2[i])
```

```
	return false;
		}
		return true;
	}
};
function solArraySorter(t1, t2)
{
	return t1.index - t2.index;
};
function findMatchingSolModifier(arr)
{
	var i, len, u, temp, subarr;
	if (arr.length === 2)
	{
		if (arr[0].index > arr[1].index)
		{
			temp = arr[0];
			arr[0] = arr[1];
			arr[1] = temp;
		}
	}
	else if (arr.length > 2)

	arr.sort(solArraySorter);
	// so testSolsMatch compares in same order
	if (arr.length >= allUniqueSolModifiers.length)

	allUniqueSolModifiers.length = arr.length + 1;
	if (!allUniqueSolModifiers[arr.length])

	allUniqueSolModifiers[arr.length] = [];
	subarr = allUniqueSolModifiers[arr.length];
	for (i = 0, len = subarr.length; i < len; i++)
	{
		u = subarr[i];
		if (testSolsMatch(arr, u))
			return u;
	}
	subarr.push(arr);
	return arr;
};
function EventSheet(runtime, m)
{
	this.runtime = runtime;
	this.triggers = {};
	this.fasttriggers = {};
	this.hasRun = false;
	this.includes = new cr.ObjectSet(); // all event sheets included by this sheet, at first-level indirection only
	this.name = m[0];
	var em = m[1];
// events model
	this.events = [];
// triggers won't make it to this array
	var i, len;
	for (i = 0, len = em.length; i < len; i++)

	this.init_event(em[i], null, this.events);
};

EventSheet.prototype.toString = function ()
{
	return this.name;
};
EventSheet.prototype.init_event = function (m, parent, nontriggers)
{
	switch (m[0]) {
	case 0:	// event block
	{
		var block = new cr.eventblock(this, parent, m);
		cr.seal(block);
		if (block.orblock)
		{

	nontriggers.push(block);
```

```
                    var i,
len;
                    for (i =
0, len =
block.conditions.length; i <
len; i++)
                    {
                         if
(block.conditions[i].trigger)

     this.init_trigger(block,
i);
                    }
               }
               else
               {
                    if
(block.is_trigger())

     this.init_trigger(block,
0);
                    else

     nontriggers.push(block);
               }
               break;
          }
          case 1: // variable
          {
               var v = new
cr.eventvariable(this, parent,
m);
               cr.seal(v);

     nontriggers.push(v);
               break;
          }
        case 2:  // include
        {
            var inc = new
cr.eventinclude(this, parent,
m);
               cr.seal(inc);

nontriggers.push(inc);
               break;
        }
          default:
;
          }
     };
     EventSheet.prototype.postI
nit = function ()
     {
          var i, len;
          for (i = 0, len =
this.events.length; i < len;
i++)
          {

     this.events[i].postInit(i
< len - 1 && this.events[i +
1].is_else_block);
          }
     };
     EventSheet.prototype.run =
function (from_include)
     {
          if
(!this.runtime.resuming_breakpo
int)
          {
               this.hasRun =
true;
               if
(!from_include)

     this.runtime.isRunningEven
ts = true;
          }
          var i, len;
          for (i = 0, len =
this.events.length; i < len;
i++)
          {
               var ev =
this.events[i];
               ev.run();

     this.runtime.clearSol(ev.s
olModifiers);
                    if
(!this.runtime.deathRow.isEmpty
() ||
this.runtime.createRow.length)

     this.runtime.ClearDeathRow
();
          }
               if
(!from_include)

     this.runtime.isRunningEven
ts = false;
     };
     EventSheet.prototype.init_
trigger = function (trig,
index)
     {
          if (!trig.orblock)
```

```
     this.runtime.triggers_to_p
ostinit.push(trig);   // needs
to be postInit'd later
           var i, len;
           var cnd =
trig.conditions[index];
           var type_name;
           if (cnd.type)
                 type_name =
cnd.type.name;
           else
                 type_name =
"system";
           var fasttrigger =
cnd.fasttrigger;
           var triggers =
(fasttrigger ?
this.fasttriggers :
this.triggers);
           if
(!triggers[type_name])

     triggers[type_name] = [];
           var obj_entry =
triggers[type_name];
           var method =
cnd.func;
           if (fasttrigger)
           {
                 if
(!cnd.parameters.length)
                 // no
parameters
                      return;
                 var firstparam
= cnd.parameters[0];
                 if
(firstparam.type !== 1 ||
           // not a string
param

     firstparam.expression.type
!== 2)     // not a string
literal node
                 {
                      return;
                 }
                 var fastevs;
                 var firstvalue
=
firstparam.expression.value.toL
owerCase();
                 var i, len;
                 for (i = 0, len
= obj_entry.length; i < len;
i++)
                 {
                      if
(obj_entry[i].method == method)
                      {

     fastevs =
obj_entry[i].evs;
                           if
(!fastevs[firstvalue])

     fastevs[firstvalue] =
[[trig, index]];
                           else

     fastevs[firstvalue].push([
trig, index]);

     return;
                      }
                 }
                 fastevs = {};

     fastevs[firstvalue] =
[[trig, index]];

     obj_entry.push({ method:
method, evs: fastevs });
           }
           else
           {
                 for (i = 0, len
= obj_entry.length; i < len;
i++)
                 {
                      if
(obj_entry[i].method == method)
                      {

     obj_entry[i].evs.push([tri
g, index]);

     return;
                      }
                 }

     obj_entry.push({ method:
method, evs: [[trig, index]]});
           }
     };
     cr.eventsheet =
EventSheet;
     function Selection(type)
     {
```

```
		this.type = type;
		this.instances = [];
// subset of picked instances
		this.else_instances
= []; // subset of unpicked
instances
		this.select_all =
true;
	};
	Selection.prototype.hasObj
ects = function ()
	{
		if (this.select_all)
			return
this.type.instances.length;
		else
			return
this.instances.length;
	};
	Selection.prototype.getObj
ects = function ()
	{
		if (this.select_all)
			return
this.type.instances;
		else
			return
this.instances;
	};
	/*
	Selection.prototype.ensure
_picked = function (inst,
skip_siblings)
	{
		var i, len;
		var orblock =
inst.runtime.getCurrentEventSta
ck().current_event.orblock;
		if (this.select_all)
		{
			this.select_all
= false;
			if (orblock)
			{

	cr.shallowAssignArray(this
.else_instances,
inst.type.instances);

	cr.arrayFindRemove(this.el
se_instances, inst);
			}

	this.instances.length = 1;

	this.instances[0] = inst;
		}
		else
		{
			if (orblock)
			{
				i =
this.else_instances.indexOf(ins
t);
				if (i !==
-1)
				{

	this.instances.push(this.e
lse_instances[i]);

	this.else_instances.splice
(i, 1);
				}
			}
			else
			{
				if
(this.instances.indexOf(inst)
=== -1)

	this.instances.push(inst);
			}
		}
		if (!skip_siblings)
		{
		}
	};
	*/
	Selection.prototype.pick_o
ne = function (inst)
	{
		if (!inst)
			return;
		if
(inst.runtime.getCurrentEventSt
ack().current_event.orblock)
		{
			if
(this.select_all)
			{

	this.instances.length = 0;

	cr.shallowAssignArray(this
.else_instances,
inst.type.instances);

	this.select_all = false;
			}
```

```
                var i =
this.else_instances.indexOf(ins
t);
                if (i !== -1)
                {

     this.instances.push(this.e
lse_instances[i]);

     this.else_instances.splice
(i, 1);
                }
           }
           else
           {
                this.select_all
= false;

     this.instances.length = 1;

     this.instances[0] = inst;
           }
     };
     cr.selection = Selection;
     function EventBlock(sheet,
parent, m)
     {
           this.sheet = sheet;
           this.parent =
parent;
           this.runtime =
sheet.runtime;
           this.solModifiers =
[];

     this.solModifiersIncluding
Parents = [];

     this.solWriterAfterCnds =
false;     // block does not
change SOL after running its
conditions
           this.group = false;
                              //
is group of events

     this.initially_activated =
false;     // if a group, is
active on startup
           this.toplevelevent =
false;                // is an
event block parented only by a
top-level group
           this.toplevelgroup =
false;                // is
parented only by other groups
or is top-level (i.e. not in a
subevent)
           this.has_else_block
= false;        // is followed
by else
;
           this.conditions =
[];
           this.actions = [];
           this.subevents = [];
        if (m[1])
        {
                this.group_name
= m[1][1].toLowerCase();
                this.group =
true;

     this.initially_activated =
!!m[1][0];

     this.runtime.allGroups.pus
h(this);

this.runtime.activeGroups[(/*th
is.sheet.name + "|" +
*/this.group_name).toLowerCase(
)] = this.initially_activated;
        }
           else
           {
                this.group_name
= "";
                this.group =
false;

     this.initially_activated =
false;
           }
           this.orblock = m[2];
           this.sid = m[4];
           if (!this.group)

     this.runtime.blocksBySid[t
his.sid.toString()] = this;
           var i, len;
           var cm = m[5];
           for (i = 0, len =
cm.length; i < len; i++)
           {
                var cnd = new
cr.condition(this, cm[i]);
                cnd.index = i;
                cr.seal(cnd);

     this.conditions.push(cnd);
                /*
```

```
                if
(cnd.is_logical())

     this.is_logical = true;
                if (cnd.type &&
!cnd.type.plugin.singleglobal
&&
this.cndReferences.indexOf(cnd.
type) === -1)

     this.cndReferences.push(cn
d.type);
                */

     this.addSolModifier(cnd.ty
pe);
           }
           var am = m[6];
           for (i = 0, len =
am.length; i < len; i++)
           {
                var act = new
cr.action(this, am[i]);
                act.index = i;
                cr.seal(act);

     this.actions.push(act);
           }
           if (m.length === 8)
           {
                var em = m[7];
                for (i = 0, len
= em.length; i < len; i++)

     this.sheet.init_event(em[i
], this, this.subevents);
           }
           this.is_else_block =
false;
           if
(this.conditions.length)
           {

     this.is_else_block =
(this.conditions[0].type ==
null && this.conditions[0].func
==
cr.system_object.prototype.cnds
.Else);
           }
     };
     window["_c2hh_"] = "";
     EventBlock.prototype.postI
nit = function (hasElse/*,
prevBlock_*/)
     {
           var i, len;
           var p = this.parent;
           if (this.group)
           {

     this.toplevelgroup = true;
                while (p)
                {
                     if
(!p.group)
                     {

     this.toplevelgroup =
false;

     break;
                     }
                     p =
p.parent;
                }
           }
           this.toplevelevent =
!this.is_trigger() &&
(!this.parent ||
(this.parent.group &&
this.parent.toplevelgroup));
           this.has_else_block
= !!hasElse;

     this.solModifiersIncluding
Parents =
this.solModifiers.slice(0);
           p = this.parent;
           while (p)
           {
                for (i = 0, len
= p.solModifiers.length; i <
len; i++)

     this.addParentSolModifier(
p.solModifiers[i]);
                p = p.parent;
           }
           this.solModifiers =
findMatchingSolModifier(this.so
lModifiers);

     this.solModifiersIncluding
Parents =
findMatchingSolModifier(this.so
lModifiersIncludingParents);
           var i, len/*, s*/;
           for (i = 0, len =
this.conditions.length; i <
len; i++)
```

```
	this.conditions[i].postInit();
		for (i = 0, len = this.actions.length; i < len; i++)
	this.actions[i].postInit();
		for (i = 0, len = this.subevents.length; i < len; i++)
		{
	this.subevents[i].postInit(i < len - 1 && this.subevents[i + 1].is_else_block);
		}
		/*
		if (this.is_else_block && this.prev_block)
		{
			for (i = 0, len = this.prev_block.solModifiers.length; i < len; i++)
			{
				s = this.prev_block.solModifiers[i];
				if (this.solModifiers.indexOf(s) === -1)
	this.solModifiers.push(s);
			}
		}
		*/
	}
	function addSolModifierToList(type, arr)
	{
		var i, len, t;
		if (!type)
			return;
		if (arr.indexOf(type) === -1)
			arr.push(type);
		if (type.is_contained)
		{
			for (i = 0, len = type.container.length; i < len; i++)
			{
				t = type.container[i];
				if (type === t)
	continue;	// already handled
				if (arr.indexOf(t) === -1)
	arr.push(t);
			}
		}
	};
	EventBlock.prototype.addSolModifier = function (type)
	{
	addSolModifierToList(type, this.solModifiers);
	};
	EventBlock.prototype.addParentSolModifier = function (type)
	{
	addSolModifierToList(type, this.solModifiersIncludingParents);
	};
	EventBlock.prototype.setSolWriterAfterCnds = function ()
	{
	this.solWriterAfterCnds = true;
		if (this.parent)
	this.parent.setSolWriterAfterCnds();
	};
	EventBlock.prototype.is_trigger = function ()
	{
		if (!this.conditions.length)	// no conditions
			return false;
		else
			return this.conditions[0].trigger;
	};
	EventBlock.prototype.run = function ()
	{
```

```
var i, len, any_true
= false, cnd_result;
var evinfo =
this.runtime.getCurrentEventSta
ck();
evinfo.current_event
= this;
if
(!this.is_else_block)

evinfo.else_branch_ran =
false;
if (this.orblock)
{
if
(this.conditions.length === 0)
any_true
= true;          // be sure to
run if empty block

evinfo.cndindex = 0
for (len =
this.conditions.length;
evinfo.cndindex < len;
evinfo.cndindex++)
{
if
(this.conditions[evinfo.cndinde
x].trigger)          // skip
triggers when running OR block

continue;

cnd_result =
this.conditions[evinfo.cndindex
].run();
if
(cnd_result)               //
make sure all conditions run
and run if any were true

any_true = true;
}

evinfo.last_event_true =
any_true;
if (any_true)

this.run_actions_and_subev
ents();
}
else
{

evinfo.cndindex = 0
for (len =
this.conditions.length;
evinfo.cndindex < len;
evinfo.cndindex++)
{

cnd_result =
this.conditions[evinfo.cndindex
].run();
if
(!cnd_result)    // condition
failed
{

evinfo.last_event_true =
false;
if
(this.toplevelevent &&
(!this.runtime.deathRow.isEmpty
() ||
this.runtime.createRow.length))

this.runtime.ClearDeathRow
();

return;          // bail
out now
}
}

evinfo.last_event_true =
true;

this.run_actions_and_subev
ents();
}

this.end_run(evinfo);
};
EventBlock.prototype.end_r
un = function (evinfo)
{
if
(evinfo.last_event_true &&
this.has_else_block)

evinfo.else_branch_ran =
true;
if
(this.toplevelevent &&
(!this.runtime.deathRow.isEmpty
() ||
this.runtime.createRow.length))

this.runtime.ClearDeathRow
();
```

```
	};
	EventBlock.prototype.run_o
rblocktrigger = function
(index)
	{
		var evinfo =
this.runtime.getCurrentEventSta
ck();
		evinfo.current_event
= this;
		if
(this.conditions[index].run())
		{

	this.run_actions_and_subev
ents();

	this.runtime.getCurrentEve
ntStack().last_event_true =
true;
		}
	};
	EventBlock.prototype.run_a
ctions_and_subevents = function
()
	{
		var evinfo =
this.runtime.getCurrentEventSta
ck();
		var len;
		for (evinfo.actindex
= 0, len = this.actions.length;
evinfo.actindex < len;
evinfo.actindex++)
		{
			if
(this.actions[evinfo.actindex].
run())
				return;
		}

	this.run_subevents();
	};
	EventBlock.prototype.resum
e_actions_and_subevents =
function ()
	{
		var evinfo =
this.runtime.getCurrentEventSta
ck();
		var len;
		for (len =
this.actions.length;
evinfo.actindex < len;
evinfo.actindex++)
		{
			if
(this.actions[evinfo.actindex].
run())
				return;
		}

	this.run_subevents();
	};
	EventBlock.prototype.run_s
ubevents = function ()
	{
		if
(!this.subevents.length)
			return;
		var i, len, subev,
pushpop/*, skipped_pop = false,
pop_modifiers = null*/;
		var last =
this.subevents.length - 1;

	this.runtime.pushEventStac
k(this);
		if
(this.solWriterAfterCnds)
		{
			for (i = 0, len
= this.subevents.length; i <
len; i++)
			{
				subev =
this.subevents[i];

	pushpop =
(!this.toplevelgroup ||
(!this.group && i < last));
					if
(pushpop)

	this.runtime.pushCopySol(s
ubev.solModifiers);

	subev.run();
					if
(pushpop)

	this.runtime.popSol(subev.
solModifiers);
					else

	this.runtime.clearSol(sube
v.solModifiers);
			}
		}
		else
		{
```

```
                for (i = 0, len
= this.subevents.length; i <
len; i++)
                {

     this.subevents[i].run();
                }
           }

     this.runtime.popEventStack
();
     };
     EventBlock.prototype.run_p
retrigger = function ()
     {
           var evinfo =
this.runtime.getCurrentEventSta
ck();
           evinfo.current_event
= this;
           var any_true =
false;
           var i, len;
           for (evinfo.cndindex
= 0, len =
this.conditions.length;
evinfo.cndindex < len;
evinfo.cndindex++)
           {
;
                if
(this.conditions[evinfo.cndinde
x].run())
                     any_true
= true;
                else if
(!this.orblock)            //
condition failed (let OR blocks
run all conditions anyway)
                     return
false;               // bail
out
           }
           return this.orblock
? any_true : true;
     };
     EventBlock.prototype.retri
gger = function ()
     {

     this.runtime.execcount++;
           var prevcndindex =
this.runtime.getCurrentEventSta
ck().cndindex;
           var len;
           var evinfo =
this.runtime.pushEventStack(thi
s);
           if (!this.orblock)
           {
                for
(evinfo.cndindex = prevcndindex
+ 1, len =
this.conditions.length;
evinfo.cndindex < len;
evinfo.cndindex++)
                {
                     if
(!this.conditions[evinfo.cndind
ex].run())    // condition
failed
                     {

     this.runtime.popEventStack
();               // moving up
level of recursion

     return false;
// bail out
                     }
                }
           }

     this.run_actions_and_subev
ents();

     this.runtime.popEventStack
();
           return true;
     // ran an iteration
     };
     EventBlock.prototype.isFir
stConditionOfType = function
(cnd)
     {
           var cndindex =
cnd.index;
           if (cndindex === 0)
                return true;
           --cndindex;
           for ( ; cndindex >=
0; --cndindex)
           {
                if
(this.conditions[cndindex].type
=== cnd.type)
                     return
false;
           }
           return true;
     };
```

```
	cr.eventblock =
EventBlock;
	function Condition(block,
m)
	{
		this.block = block;
		this.sheet =
block.sheet;
		this.runtime =
block.runtime;
		this.parameters =
[];
		this.results = [];
		this.extra = {};
	// for plugins to stow
away some custom info
		this.index = -1;
		this.func = m[1];
;
		this.trigger = (m[3]
> 0);
		this.fasttrigger =
(m[3] === 2);
		this.looping = m[4];
		this.inverted =
m[5];
		this.isstatic =
m[6];
		this.sid = m[7];

	this.runtime.cndsBySid[thi
s.sid.toString()] = this;
		if (m[0] === -1)
	// system object
		{
			this.type =
null;
			this.run =
this.run_system;

	this.behaviortype = null;
			this.beh_index
= -1;
		}
		else
		{
			this.type =
this.runtime.types_by_index[m[0
]];
;
			if
(this.isstatic)
				this.run
= this.run_static;
			else
				this.run
= this.run_object;
			if (m[2])
			{

	this.behaviortype =
this.type.getBehaviorByName(m[2
]);
;

	this.beh_index =
this.type.getBehaviorIndexByNam
e(m[2]);
;
			}
			else
			{

	this.behaviortype = null;

	this.beh_index = -1;
			}
			if
(this.block.parent)

	this.block.parent.setSolWr
iterAfterCnds();
		}
		if
(this.fasttrigger)
			this.run =
this.run_true;
		if (m.length === 10)
		{
			var i, len;
			var em = m[9];
			for (i = 0, len
= em.length; i < len; i++)
			{
				var param
= new cr.parameter(this,
em[i]);

	cr.seal(param);

	this.parameters.push(param
);
			}

	this.results.length =
em.length;
		}
	};
	Condition.prototype.postIn
it = function ()
	{
```

```
		var i, len;
		for (i = 0, len = this.parameters.length; i < len; i++)
	this.parameters[i].postInit();
	};
	/*
	Condition.prototype.is_logical = function ()
	{
		return !this.type || this.type.plugin.singleglobal;
	};
	*/
	Condition.prototype.run_true = function ()
	{
		return true;
	};
	Condition.prototype.run_system = function ()
	{
		var i, len;
		for (i = 0, len = this.parameters.length; i < len; i++)
			this.results[i] = this.parameters[i].get();
		return cr.xor(this.func.apply(this.runtime.system, this.results), this.inverted);
	};
	Condition.prototype.run_static = function ()
	{
		var i, len;
		for (i = 0, len = this.parameters.length; i < len; i++)
			this.results[i] = this.parameters[i].get(i);
		var ret = this.func.apply(this.behaviortype ? this.behaviortype : this.type, this.results);
	this.type.applySolToContainer();
		return ret;
	};
	Condition.prototype.run_object = function ()
	{
		var i, j, leni, lenj, ret, met, inst, s, sol2;
		var sol = this.type.getCurrentSol();
		var is_orblock = this.block.orblock && !this.trigger;	// triggers in OR blocks need to work normally
		var offset = 0;
		var is_contained = this.type.is_contained;
		if (sol.select_all) {
	sol.instances.length = 0; // clear contents
	sol.else_instances.length = 0;
			for (i = 0, leni = this.type.instances.length; i < leni; i++)
			{
				inst = this.type.instances[i];
;
				for (j = 0, lenj = this.parameters.length; j < lenj; j++)
	this.results[j] = this.parameters[j].get(i); // default SOL index is current object
				if (this.beh_index > -1)
				{
					if (this.type.is_family)
					{
	offset = inst.type.family_beh_map[this.type.family_index];
					}
					ret = this.func.apply(inst.behavior_insts[this.beh_index + offset], this.results);
				}
				else
```

```
                        ret
= this.func.apply(inst,
this.results);
                    met =
cr.xor(ret, this.inverted);
                    if (met)

     sol.instances.push(inst);
                    else if
(is_orblock)
          // in OR blocks,
keep the instances not meeting
the condition for subsequent
testing

     sol.else_instances.push(in
st);
               }
               if
(this.type.finish)

     this.type.finish(true);
               sol.select_all
= false;

     this.type.applySolToContai
ner();
               return
sol.hasObjects();
          }
          else {
               var k = 0;
               var
using_else_instances =
(is_orblock &&
!this.block.isFirstConditionOfT
ype(this));
               var arr =
(using_else_instances ?
sol.else_instances :
sol.instances);
               var any_true =
false;
               for (i = 0,
leni = arr.length; i < leni;
i++)
               {
                    inst =
arr[i];
;
                    for (j =
0, lenj =
this.parameters.length; j <
lenj; j++)

     this.results[j] =
this.parameters[j].get(i);
// default SOL index is current
object
                    if
(this.beh_index > -1)
                    {
                         if
(this.type.is_family)
                         {

     offset =
inst.type.family_beh_map[this.t
ype.family_index];
                         }
                         ret
=
this.func.apply(inst.behavior_i
nsts[this.beh_index + offset],
this.results);
                    }
                    else
                         ret
= this.func.apply(inst,
this.results);
                    if
(cr.xor(ret, this.inverted))
                    {

     any_true = true;
                         if
(using_else_instances)
                         {

     sol.instances.push(inst);

     if (is_contained)

     {

          for (j = 0, lenj =
inst.siblings.length; j < lenj;
j++)

          {

               s =
inst.siblings[j];


     s.type.getCurrentSol().ins
tances.push(s);

          }

     }
                         }
```

```
                        else
                        {
arr[k] = inst;
if (is_contained)
{
        for (j = 0, lenj =
inst.siblings.length; j < lenj;
j++)
        {
                s =
inst.siblings[j];

s.type.getCurrentSol().ins
tances[k] = s;
        }
}
k++;
                        }
                    }
                    else
                    {
                        if
(using_else_instances)
                        {
arr[k] = inst;
if (is_contained)
{
        for (j = 0, lenj =
inst.siblings.length; j < lenj;
j++)
        {
                s =
inst.siblings[j];

s.type.getCurrentSol().els
e_instances[k] = s;
        }
}
k++;
                        }
                        else
if (is_orblock)
                        {
sol.else_instances.push(in
st);
if (is_contained)
{
        for (j = 0, lenj =
inst.siblings.length; j < lenj;
j++)
        {
                s =
inst.siblings[j];

s.type.getCurrentSol().els
e_instances.push(s);
        }
}
                        }
                    }
                }
                arr.length = k;
                if
(is_contained)
                {
                    for (i =
0, leni =
this.type.container.length; i <
leni; i++)
                    {
                        sol2
=
this.type.container[i].getCurre
ntSol();
                        if
(using_else_instances)
sol2.else_instances.length
= k;
                        else
sol2.instances.length = k;
```

```
                        }
                    }
                    var
pick_in_finish = any_true;
     // don't pick in finish()
if we're only doing the logic
test below
                    if
(using_else_instances &&
!any_true)
                    {
                        for (i =
0, leni = sol.instances.length;
i < leni; i++)
                        {
                            inst
= sol.instances[i];
                            for
(j = 0, lenj =
this.parameters.length; j <
lenj; j++)

     this.results[j] =
this.parameters[j].get(i);
                            if
(this.beh_index > -1)

     ret =
this.func.apply(inst.behavior_i
nsts[this.beh_index],
this.results);
                            else

     ret =
this.func.apply(inst,
this.results);
                            if
(cr.xor(ret, this.inverted))
                            {

     any_true = true;

     break;          // got
our flag, don't need to test
any more
                            }
                        }
                    }
                    if
(this.type.finish)

     this.type.finish(pick_in_f
inish || is_orblock);
                    return
is_orblock ? any_true :
sol.hasObjects();
                }
            };
            cr.condition = Condition;
            function Action(block, m)
            {
                this.block = block;
                this.sheet =
block.sheet;
                this.runtime =
block.runtime;
                this.parameters =
[];
                this.results = [];
                this.extra = {};
            // for plugins to stow
away some custom info
                this.index = -1;
                this.func = m[1];
;
                if (m[0] === -1) //
system
                {
                    this.type =
null;
                    this.run =
this.run_system;

            this.behaviortype = null;
                    this.beh_index
= -1;
                }
                else
                {
                    this.type =
this.runtime.types_by_index[m[0
]];
;
                    this.run =
this.run_object;
                    if (m[2])
                    {

            this.behaviortype =
this.type.getBehaviorByName(m[2
]);
;

            this.beh_index =
this.type.getBehaviorIndexByNam
e(m[2]);
;
                    }
                    else
                    {

            this.behaviortype = null;
```

```
	this.beh_index = -1;
		}
	}
	this.sid = m[3];

	this.runtime.actsBySid[this.sid.toString()] = this;
	if (m.length === 6)
	{
		var i, len;
		var em = m[5];
		for (i = 0, len = em.length; i < len; i++)
		{
			var param = new cr.parameter(this, em[i]);

	cr.seal(param);

	this.parameters.push(param);
		}

	this.results.length = em.length;
	}
	};
	Action.prototype.postInit = function ()
	{
	var i, len;
	for (i = 0, len = this.parameters.length; i < len; i++)

	this.parameters[i].postInit();
	};
	Action.prototype.run_system = function ()
	{
	var i, len;
	for (i = 0, len = this.parameters.length; i < len; i++)
		this.results[i] = this.parameters[i].get();
	return this.func.apply(this.runtime.system, this.results);
	};
	Action.prototype.run_object = function ()
	{
	var instances = this.type.getCurrentSol().getObjects();
	var i, j, leni, lenj, inst;
	for (i = 0, leni = instances.length; i < leni; i++)
	{
		inst = instances[i];
		for (j = 0, lenj = this.parameters.length; j < lenj; j++)

	this.results[j] = this.parameters[j].get(i); // pass i to use as default SOL index
		if (this.beh_index > -1)
		{
			var offset = 0;
			if (this.type.is_family)
			{

	offset = inst.type.family_beh_map[this.type.family_index];
			}

	this.func.apply(inst.behavior_insts[this.beh_index + offset], this.results);
		}
		else

	this.func.apply(inst, this.results);
	}
	return false;
	};
	cr.action = Action;
	var tempValues = [];
	var tempValuesPtr = -1;
	function Parameter(owner, m)
	{
	this.owner = owner;
	this.block = owner.block;
	this.sheet = owner.sheet;
```

```
            this.runtime =
owner.runtime;
            this.type = m[0];
            this.expression =
null;
            this.solindex = 0;
            this.combosel = 0;
            this.layout = null;
            this.key = 0;
            this.object = null;
            this.index = 0;
            this.varname = null;
            this.eventvar =
null;
            this.fileinfo =
null;
            this.subparams =
null;
            this.variadicret =
null;
            var i, len, param;
            switch (m[0])
            {
                  case 0:
      // number
                  case 7:
      // any

      this.expression = new
cr.expNode(this, m[1]);

      this.solindex = 0;
                        this.get
= this.get_exp;
                        break;
                  case 1:
      // string

      this.expression = new
cr.expNode(this, m[1]);

      this.solindex = 0;
                        this.get
= this.get_exp_str;
                        break;
                  case 5:
      // layer

      this.expression = new
cr.expNode(this, m[1]);

      this.solindex = 0;
                        this.get
= this.get_layer;
                        break;
                  case 3:
      // combo
                  case 8:
      // cmp

      this.combosel = m[1];
                        this.get
= this.get_combosel;
                        break;
                  case 6:
      // layout

      this.layout =
this.runtime.layouts[m[1]];
;
                        this.get
= this.get_layout;
                        break;
                  case 9:
      // keyb
                        this.key
= m[1];
                        this.get
= this.get_key;
                        break;
                  case 4:
      // object

      this.object =
this.runtime.types_by_index[m[1
]];
;
                        this.get
= this.get_object;

      this.block.addSolModifier(
this.object);
                        if
(this.owner instanceof
cr.action)

      this.block.setSolWriterAft
erCnds();
                        else if
(this.block.parent)

      this.block.parent.setSolWr
iterAfterCnds();
                        break;
                  case 10:   //
instvar

      this.index = m[1];
                        if
(owner.type.is_family)
```

```
	this.get =
this.get_familyvar;
                        else

	this.get =
this.get_instvar;
                        break;
                case 11:   //
eventvar

	this.varname = m[1];

	this.eventvar = null;
                        this.get
= this.get_eventvar;
                        break;
                case 2:
	// audiofile    ["name",
ismusic]
                case 12:   //
fileinfo        "name"

	this.fileinfo = m[1];
                        this.get
= this.get_audiofile;
                        break;
                case 13:   //
variadic
                        this.get
= this.get_variadic;

	this.subparams = [];

	this.variadicret = [];
                        for (i =
1, len = m.length; i < len;
i++)
                        {

	param = new
cr.parameter(this.owner, m[i]);

	cr.seal(param);

	this.subparams.push(param)
;

	this.variadicret.push(0);
                        }
                        break;
                default:
;
            }
	};
```

```
	Parameter.prototype.postIn
it = function ()
	{
            var i, len;
            if (this.type ===
11)         // eventvar
            {
                this.eventvar =
this.runtime.getEventVariableBy
Name(this.varname,
this.block.parent);
;
            }
            else if (this.type
=== 13)     // variadic,
postInit all sub-params
            {
                for (i = 0, len
= this.subparams.length; i <
len; i++)

	this.subparams[i].postInit
();
            }
            if (this.expression)

	this.expression.postInit()
;
	};
	Parameter.prototype.pushTe
mpValue = function ()
	{
            tempValuesPtr++;
            if
(tempValues.length ===
tempValuesPtr)

	tempValues.push(new
cr.expvalue());
            return
tempValues[tempValuesPtr];
	};
	Parameter.prototype.popTem
pValue = function ()
	{
            tempValuesPtr--;
	};
	Parameter.prototype.get_ex
p = function (solindex)
	{
            this.solindex =
solindex || 0;   // default SOL
index to use
            var temp =
this.pushTempValue();
```

```
	this.expression.get(temp);
		this.popTempValue();
		return temp.data;
			// return
actual JS value, not expvalue
	};
	Parameter.prototype.get_ex
p_str = function (solindex)
	{
		this.solindex =
solindex || 0;	// default SOL
index to use
		var temp =
this.pushTempValue();

	this.expression.get(temp);
		this.popTempValue();
		if
(cr.is_string(temp.data))
			return
temp.data;
		else
			return "";
	};
	Parameter.prototype.get_ob
ject = function ()
	{
		return this.object;
	};
	Parameter.prototype.get_co
mbosel = function ()
	{
		return
this.combosel;
	};
	Parameter.prototype.get_la
yer = function (solindex)
	{
		this.solindex =
solindex || 0;	// default SOL
index to use
		var temp =
this.pushTempValue();

	this.expression.get(temp);
		this.popTempValue();
		if
(temp.is_number())
			return
this.runtime.getLayerByNumber(t
emp.data);
		else
			return
this.runtime.getLayerByName(tem
p.data);
	}
	Parameter.prototype.get_la
yout = function ()
	{
		return this.layout;
	};
	Parameter.prototype.get_ke
y = function ()
	{
		return this.key;
	};
	Parameter.prototype.get_in
stvar = function ()
	{
		return this.index;
	};
	Parameter.prototype.get_fa
milyvar = function (solindex)
	{
		var familytype =
this.owner.type;
		var realtype = null;
		var sol =
familytype.getCurrentSol();
		var objs =
sol.getObjects();
		if (objs.length)
			realtype =
objs[solindex %
objs.length].type;
		else
		{
;
			realtype =
sol.else_instances[solindex %
sol.else_instances.length].type
;
		}
		return this.index +
realtype.family_var_map[familyt
ype.family_index];
	};
	Parameter.prototype.get_ev
entvar = function ()
	{
		return
this.eventvar;
	};
	Parameter.prototype.get_au
diofile = function ()
	{
		return
this.fileinfo;
	};
	Parameter.prototype.get_va
riadic = function ()
```

```
	{
		var i, len;
		for (i = 0, len = 
this.subparams.length; i < len; 
i++)
		{

	this.variadicret[i] = 
this.subparams[i].get();
		}
		return 
this.variadicret;
	};
	cr.parameter = Parameter;
	function 
EventVariable(sheet, parent, m)
	{
		this.sheet = sheet;
		this.parent = 
parent;
		this.runtime = 
sheet.runtime;
		this.solModifiers = 
[];
		this.name = m[1];
		this.vartype = m[2];
		this.initial = m[3];
		this.is_static = 
!!m[4];
		this.is_constant = 
!!m[5];
		this.sid = m[6];

	this.runtime.varsBySid[thi
s.sid.toString()] = this;
		this.data = 
this.initial;	// note: also 
stored in event stack frame for 
local nonstatic nonconst vars
		if (this.parent)
		// local var
		{
			if 
(this.is_static || 
this.is_constant)

	this.localIndex = -1;
			else

	this.localIndex = 
this.runtime.stackLocalCount++;

	this.runtime.all_local_var
s.push(this);
		}
		else
			// global var
		{
			this.localIndex 
= -1;

	this.runtime.all_global_va
rs.push(this);
		}
	};
	EventVariable.prototype.po
stInit = function ()
	{
		this.solModifiers = 
findMatchingSolModifier(this.so
lModifiers);
	};
	EventVariable.prototype.se
tValue = function (x)
	{
;
		var lvs = 
this.runtime.getCurrentLocalVar
Stack();
		if (!this.parent || 
this.is_static || !lvs)
			this.data = x;
		else // local 
nonstatic variable: use event 
stack to keep value at this 
level of recursion
		{
			if 
(this.localIndex >= lvs.length)

	lvs.length = 
this.localIndex + 1;

	lvs[this.localIndex] = x;
		}
	};
	EventVariable.prototype.ge
tValue = function ()
	{
		var lvs = 
this.runtime.getCurrentLocalVar
Stack();
		if (!this.parent || 
this.is_static || !lvs || 
this.is_constant)
			return 
this.data;
		else // local 
nonstatic variable
		{
```

```
                if
(this.localIndex >= lvs.length)
                {
;
                     return
this.initial;
                }
                if (typeof
lvs[this.localIndex] ===
"undefined")
                {
;
                     return
this.initial;
                }
                return
lvs[this.localIndex];
           }
      };
      EventVariable.prototype.ru
n = function ()
      {
                if (this.parent
&& !this.is_static &&
!this.is_constant)

      this.setValue(this.initial
);
      };
      cr.eventvariable =
EventVariable;
      function
EventInclude(sheet, parent, m)
      {
           this.sheet = sheet;
           this.parent =
parent;
           this.runtime =
sheet.runtime;
           this.solModifiers =
[];
           this.include_sheet =
null;      // determined in
postInit

      this.include_sheet_name =
m[1];
      };
      EventInclude.prototype.toS
tring = function ()
      {
           return "include:" +
this.include_sheet.toString();
      };
      EventInclude.prototype.pos
tInit = function ()
      {
         this.include_sheet =
this.runtime.eventsheets[this.i
nclude_sheet_name];
;
;

this.sheet.includes.add(this);
           this.solModifiers =
findMatchingSolModifier(this.so
lModifiers);
      };
      EventInclude.prototype.run
= function ()
      {
                if
(this.parent)

      this.runtime.pushCleanSol(
this.runtime.types_by_index);
         if
(!this.include_sheet.hasRun)

this.include_sheet.run(true);
                // from include
                if
(this.parent)

      this.runtime.popSol(this.r
untime.types_by_index);
      };
      EventInclude.prototype.isA
ctive = function ()
      {
           var p = this.parent;
           while (p)
           {
                if (p.group)
                {
                     if
(!this.runtime.activeGroups[p.g
roup_name.toLowerCase()])

      return false;
                }
                p = p.parent;
           }
           return true;
      };
      cr.eventinclude =
EventInclude;
      function EventStackFrame()
      {

      this.temp_parents_arr =
[];
```

```
            this.reset(null);
            cr.seal(this);
      };
      EventStackFrame.prototype.
reset = function (cur_event)
      {
            this.current_event =
cur_event;
            this.cndindex = 0;
            this.actindex = 0;

      this.temp_parents_arr.leng
th = 0;
            this.last_event_true
= false;
            this.else_branch_ran
= false;
            this.any_true_state
= false;
      };
      EventStackFrame.prototype.
isModifierAfterCnds = function
()
      {
            if
(this.current_event.solWriterAf
terCnds)
                  return true;
            if (this.cndindex <
this.current_event.conditions.l
ength - 1)
                  return
!!this.current_event.solModifie
rs.length;
            return false;
      };
      cr.eventStackFrame =
EventStackFrame;
}());
(function()
{
      function ExpNode(owner_,
m)
      {
            this.owner = owner_;
            this.runtime =
owner_.runtime;
            this.type = m[0];
;
            this.get =
[this.eval_int,

      this.eval_float,

      this.eval_string,
```

```
            this.eval_unaryminus,

            this.eval_add,

            this.eval_subtract,

            this.eval_multiply,

            this.eval_divide,

            this.eval_mod,

            this.eval_power,

            this.eval_and,

            this.eval_or,

            this.eval_equal,

            this.eval_notequal,

            this.eval_less,

            this.eval_lessequal,

            this.eval_greater,

            this.eval_greaterequal,

            this.eval_conditional,

            this.eval_system_exp,

            this.eval_object_behavior_
exp,

            this.eval_instvar_exp,

            this.eval_object_behavior_
exp,

            this.eval_eventvar_exp][th
is.type];
                  var paramsModel =
null;
                  this.value = null;
                  this.first = null;
                  this.second = null;
                  this.third = null;
                  this.func = null;
                  this.results = null;
                  this.parameters =
null;
```

```
            this.object_type =
null;
            this.beh_index = -1;
            this.instance_expr =
null;
            this.varindex = -1;
            this.behavior_type =
null;
            this.varname = null;
            this.eventvar =
null;
            this.return_string =
false;
            switch (this.type) {
            case 0:          //
int
            case 1:          //
float
            case 2:          //
string
                  this.value =
m[1];
                  break;
            case 3:          //
unaryminus
                  this.first =
new cr.expNode(owner_, m[1]);
                  break;
            case 18:   //
conditional
                  this.first =
new cr.expNode(owner_, m[1]);
                  this.second =
new cr.expNode(owner_, m[2]);
                  this.third =
new cr.expNode(owner_, m[3]);
                  break;
            case 19:   //
system_exp
                  this.func =
m[1];
;
                  this.results =
[];
                  this.parameters
= [];
                  if (m.length
=== 3)
                  {

      paramsModel = m[2];

      this.results.length =
paramsModel.length + 1;     //
must also fit 'ret'
                  }
                  else

      this.results.length = 1;
// to fit 'ret'
                  break;
            case 20:   //
object_exp

      this.object_type =
this.runtime.types_by_index[m[1
]];
;
                  this.beh_index
= -1;
                  this.func =
m[2];

      this.return_string = m[3];
                  if (m[4])

      this.instance_expr = new
cr.expNode(owner_, m[4]);
                  else

      this.instance_expr = null;
                  this.results =
[];
                  this.parameters
= [];
                  if (m.length
=== 6)
                  {

      paramsModel = m[5];

      this.results.length =
paramsModel.length + 1;
                  }
                  else

      this.results.length = 1;
      // to fit 'ret'
                  break;
            case 21:         //
instvar_exp

      this.object_type =
this.runtime.types_by_index[m[1
]];
;

      this.return_string = m[2];
                  if (m[3])

      this.instance_expr = new
cr.expNode(owner_, m[3]);
```

```
                else
     this.instance_expr = null;
                this.varindex =
m[4];
                break;
           case 22:        //
behavior_exp
     this.object_type =
this.runtime.types_by_index[m[1
]];
;
     this.behavior_type =
this.object_type.getBehaviorByN
ame(m[2]);
;
                this.beh_index
=
this.object_type.getBehaviorInd
exByName(m[2]);
                this.func =
m[3];
     this.return_string = m[4];
                if (m[5])
     this.instance_expr = new
cr.expNode(owner_, m[5]);
                else
     this.instance_expr = null;
                this.results =
[];
                this.parameters
= [];
                if (m.length
=== 7)
                {
     paramsModel = m[6];
     this.results.length =
paramsModel.length + 1;
                }
                else
     this.results.length = 1;
     // to fit 'ret'
                break;
           case 23:        //
eventvar_exp
                this.varname =
m[1];
                this.eventvar =
null; // assigned in postInit
                break;
           }
           if (this.type >= 4
&& this.type <= 17)
           {
                this.first =
new cr.expNode(owner_, m[1]);
                this.second =
new cr.expNode(owner_, m[2]);
           }
           if (paramsModel)
           {
                var i, len;
                for (i = 0, len
= paramsModel.length; i < len;
i++)
     this.parameters.push(new
cr.expNode(owner_,
paramsModel[i]));
           }
           cr.seal(this);
      };
      ExpNode.prototype.postInit
= function ()
      {
           if (this.type ===
23)   // eventvar_exp
           {
                this.eventvar =
this.owner.runtime.getEventVari
ableByName(this.varname,
this.owner.block.parent);
;
           }
           if (this.first)
     this.first.postInit();
           if (this.second)
     this.second.postInit();
           if (this.third)
     this.third.postInit();
           if
(this.instance_expr)
     this.instance_expr.postIni
t();
           if (this.parameters)
           {
                var i, len;
```

```
                for (i = 0, len
= this.parameters.length; i <
len; i++)

     this.parameters[i].postIni
t();
           }
     };
     ExpNode.prototype.eval_sys
tem_exp = function (ret)
     {
           this.results[0] =
ret;
           var temp =
this.owner.pushTempValue();
           var i, len;
           for (i = 0, len =
this.parameters.length; i <
len; i++)
           {

     this.parameters[i].get(tem
p);
                this.results[i
+ 1] = temp.data;   // passing
actual javascript value as
argument instead of expvalue
           }

     this.owner.popTempValue();

     this.func.apply(this.runti
me.system, this.results);
     };
     ExpNode.prototype.eval_obj
ect_behavior_exp = function
(ret)
     {
           var sol =
this.object_type.getCurrentSol(
);
           var instances =
sol.getObjects();
           if
(!instances.length)
           {
                if
(sol.else_instances.length)
                     instances
= sol.else_instances;
                else
                {
                     if
(this.return_string)

     ret.set_string("");
                     else

     ret.set_int(0);
                     return;
                }
           }
           this.results[0] =
ret;
           ret.object_class =
this.object_type;          //
so expression can access family
type if need be
           var temp =
this.owner.pushTempValue();
           var i, len;
           for (i = 0, len =
this.parameters.length; i <
len; i++) {

     this.parameters[i].get(tem
p);
                this.results[i
+ 1] = temp.data;   // passing
actual javascript value as
argument instead of expvalue
           }
           var index =
this.owner.solindex;
           if
(this.instance_expr) {

     this.instance_expr.get(tem
p);
                if
(temp.is_number()) {
                     index =
temp.data;
                     instances
= this.object_type.instances;
// pick from all instances, not
SOL
                }
           }

     this.owner.popTempValue();
           index %=
instances.length;      //
wraparound
           if (index < 0)
                index +=
instances.length;
           var returned_val;
           var inst =
instances[index];
           if (this.beh_index >
-1)
```

```
		{
			var offset = 0;
			if
(this.object_type.is_family)
			{
				offset =
inst.type.family_beh_map[this.o
bject_type.family_index];
			}
			returned_val =
this.func.apply(inst.behavior_i
nsts[this.beh_index + offset],
this.results);
		}
		else
			returned_val =
this.func.apply(inst,
this.results);
;
	};
	ExpNode.prototype.eval_ins
tvar_exp = function (ret)
	{
		var sol =
this.object_type.getCurrentSol(
);
		var instances =
sol.getObjects();
		if
(!instances.length)
		{
			if
(sol.else_instances.length)
				instances
= sol.else_instances;
			else
			{
				if
(this.return_string)

	ret.set_string("");
				else

	ret.set_int(0);
				return;
			}
		}
		var index =
this.owner.solindex;
		if
(this.instance_expr)
		{
			var temp =
this.owner.pushTempValue();

	this.instance_expr.get(tem
p);
			if
(temp.is_number())
			{
				index =
temp.data;
				var
type_instances =
this.object_type.instances;
				index %=
type_instances.length;	//
wraparound
				if (index
< 0)		//
offset

	index +=
type_instances.length;
				var
to_ret =
type_instances[index].instance_
vars[this.varindex];
				if
(cr.is_string(to_ret))

	ret.set_string(to_ret);
				else

	ret.set_float(to_ret);

	this.owner.popTempValue();
				return;
// done
			}

	this.owner.popTempValue();
		}
		index %=
instances.length;	//
wraparound
		if (index < 0)
			index +=
instances.length;
		var inst =
instances[index];
		var offset = 0;
		if
(this.object_type.is_family)
		{
			offset =
inst.type.family_var_map[this.o
bject_type.family_index];
		}
```

```
		var to_ret =
inst.instance_vars[this.varinde
x + offset];
		if
(cr.is_string(to_ret))

	ret.set_string(to_ret);
		else

	ret.set_float(to_ret);
	};
	ExpNode.prototype.eval_int
= function (ret)
	{
		ret.type =
cr.exptype.Integer;
		ret.data =
this.value;
	};
	ExpNode.prototype.eval_flo
at = function (ret)
	{
		ret.type =
cr.exptype.Float;
		ret.data =
this.value;
	};
	ExpNode.prototype.eval_str
ing = function (ret)
	{
		ret.type =
cr.exptype.String;
		ret.data =
this.value;
	};
	ExpNode.prototype.eval_una
ryminus = function (ret)
	{
		this.first.get(ret);
// retrieve operand
		if (ret.is_number())
			ret.data = -
ret.data;
	};
	ExpNode.prototype.eval_add
= function (ret)
	{
		this.first.get(ret);
// left operand
		var temp =
this.owner.pushTempValue();

	this.second.get(temp);
			// right
operand
		if (ret.is_number()
&& temp.is_number())
		{
			ret.data +=
temp.data;		// both
operands numbers: add
			if
(temp.is_float())

	ret.make_float();
		}

	this.owner.popTempValue();
	};
	ExpNode.prototype.eval_sub
tract = function (ret)
	{
		this.first.get(ret);
// left operand
		var temp =
this.owner.pushTempValue();

	this.second.get(temp);
			// right
operand
		if (ret.is_number()
&& temp.is_number())
		{
			ret.data -=
temp.data;		// both
operands numbers: subtract
			if
(temp.is_float())

	ret.make_float();
		}

	this.owner.popTempValue();
	};
	ExpNode.prototype.eval_mul
tiply = function (ret)
	{
		this.first.get(ret);
// left operand
		var temp =
this.owner.pushTempValue();

	this.second.get(temp);
			// right
operand
		if (ret.is_number()
&& temp.is_number())
		{
			ret.data *=
temp.data;		// both
operands numbers: multiply
```

```
                if
(temp.is_float())

     ret.make_float();
          }

     this.owner.popTempValue();
     };
     ExpNode.prototype.eval_div
ide = function (ret)
     {
          this.first.get(ret);
// left operand
          var temp =
this.owner.pushTempValue();

     this.second.get(temp);
                // right
operand
          if (ret.is_number()
&& temp.is_number())
          {
                ret.data /=
temp.data;          // both
operands numbers: divide

     ret.make_float();
          }

     this.owner.popTempValue();
     };
     ExpNode.prototype.eval_mod
= function (ret)
     {
          this.first.get(ret);
// left operand
          var temp =
this.owner.pushTempValue();

     this.second.get(temp);
                // right
operand
          if (ret.is_number()
&& temp.is_number())
          {
                ret.data %=
temp.data;          // both
operands numbers: modulo
                if
(temp.is_float())

     ret.make_float();
          }

     this.owner.popTempValue();
     };
     ExpNode.prototype.eval_pow
er = function (ret)
     {
          this.first.get(ret);
// left operand
          var temp =
this.owner.pushTempValue();

     this.second.get(temp);
                // right
operand
          if (ret.is_number()
&& temp.is_number())
          {
                ret.data =
Math.pow(ret.data, temp.data);
// both operands numbers: raise
to power
                if
(temp.is_float())

     ret.make_float();
          }

     this.owner.popTempValue();
     };
     ExpNode.prototype.eval_and
= function (ret)
     {
          this.first.get(ret);
// left operand
          var temp =
this.owner.pushTempValue();

     this.second.get(temp);
                // right
operand
          if (ret.is_number())
          {
                if
(temp.is_string())
                {

     ret.set_string(ret.data.to
String() + temp.data);
                }
                else
                {
                      if
(ret.data && temp.data)

     ret.set_int(1);
                      else

     ret.set_int(0);
                }
```

```
		}
		else if
(ret.is_string())
		{
			if
(temp.is_string())
				ret.data
+= temp.data;
			else
			{
				ret.data
+= (Math.round(temp.data *
1e10) / 1e10).toString();
			}
		}

	this.owner.popTempValue();
	};
	ExpNode.prototype.eval_or
= function (ret)
	{
		this.first.get(ret);
// left operand
		var temp =
this.owner.pushTempValue();

	this.second.get(temp);
			// right
operand
		if (ret.is_number()
&& temp.is_number())
		{
			if (ret.data ||
temp.data)

	ret.set_int(1);
			else

	ret.set_int(0);
		}

	this.owner.popTempValue();
	};
	ExpNode.prototype.eval_con
ditional = function (ret)
	{
		this.first.get(ret);
// condition operand
		if (ret.data)
// is true

	this.second.get(ret);
// evaluate second operand to
ret
		else

	this.third.get(ret);
// evaluate third operand to
ret
	};
	ExpNode.prototype.eval_equ
al = function (ret)
	{
		this.first.get(ret);
// left operand
		var temp =
this.owner.pushTempValue();

	this.second.get(temp);
			// right
operand
		ret.set_int(ret.data
=== temp.data ? 1 : 0);

	this.owner.popTempValue();
	};
	ExpNode.prototype.eval_not
equal = function (ret)
	{
		this.first.get(ret);
// left operand
		var temp =
this.owner.pushTempValue();

	this.second.get(temp);
			// right
operand
		ret.set_int(ret.data
!== temp.data ? 1 : 0);

	this.owner.popTempValue();
	};
	ExpNode.prototype.eval_les
s = function (ret)
	{
		this.first.get(ret);
// left operand
		var temp =
this.owner.pushTempValue();

	this.second.get(temp);
			// right
operand
		ret.set_int(ret.data
< temp.data ? 1 : 0);

	this.owner.popTempValue();
	};
	ExpNode.prototype.eval_les
sequal = function (ret)
	{
```

```
            this.first.get(ret);
// left operand
            var temp =
this.owner.pushTempValue();

      this.second.get(temp);
                  // right
operand
            ret.set_int(ret.data
<= temp.data ? 1 : 0);

      this.owner.popTempValue();
      };
      ExpNode.prototype.eval_gre
ater = function (ret)
      {
            this.first.get(ret);
// left operand
            var temp =
this.owner.pushTempValue();

      this.second.get(temp);
                  // right
operand
            ret.set_int(ret.data
> temp.data ? 1 : 0);

      this.owner.popTempValue();
      };
      ExpNode.prototype.eval_gre
aterequal = function (ret)
      {
            this.first.get(ret);
// left operand
            var temp =
this.owner.pushTempValue();

      this.second.get(temp);
                  // right
operand
            ret.set_int(ret.data
>= temp.data ? 1 : 0);

      this.owner.popTempValue();
      };
      ExpNode.prototype.eval_eve
ntvar_exp = function (ret)
      {
            var val =
this.eventvar.getValue();
            if
(cr.is_number(val))

      ret.set_float(val);
            else

      ret.set_string(val);
      };
      cr.expNode = ExpNode;
      function ExpValue(type,
data)
      {
            this.type = type ||
cr.exptype.Integer;
            this.data = data ||
0;
            this.object_class =
null;
;
;
;
            if (this.type ==
cr.exptype.Integer)
                  this.data =
Math.floor(this.data);
            cr.seal(this);
      };
      ExpValue.prototype.is_int
= function ()
      {
            return this.type ===
cr.exptype.Integer;
      };
      ExpValue.prototype.is_floa
t = function ()
      {
            return this.type ===
cr.exptype.Float;
      };
      ExpValue.prototype.is_numb
er = function ()
      {
            return this.type ===
cr.exptype.Integer || this.type
=== cr.exptype.Float;
      };
      ExpValue.prototype.is_stri
ng = function ()
      {
            return this.type ===
cr.exptype.String;
      };
      ExpValue.prototype.make_in
t = function ()
      {
            if (!this.is_int())
            {
                  if
(this.is_float())
```

```
                        this.data
= Math.floor(this.data);
// truncate float
                    else if
(this.is_string())
                        this.data
= parseInt(this.data, 10);
                    this.type =
cr.exptype.Integer;
                }
        };
        ExpValue.prototype.make_fl
oat = function ()
        {
                if
(!this.is_float())
                {
                    if
(this.is_string())
                        this.data
= parseFloat(this.data);
                    this.type =
cr.exptype.Float;
                }
        };
        ExpValue.prototype.make_st
ring = function ()
        {
                if
(!this.is_string())
                {
                    this.data =
this.data.toString();
                    this.type =
cr.exptype.String;
                }
        };
        ExpValue.prototype.set_int
= function (val)
        {
;
                this.type =
cr.exptype.Integer;
                this.data =
Math.floor(val);
        };
        ExpValue.prototype.set_flo
at = function (val)
        {
;
                this.type =
cr.exptype.Float;
                this.data = val;
        };
        ExpValue.prototype.set_str
ing = function (val)
        {
;
                this.type =
cr.exptype.String;
                this.data = val;
        };
        ExpValue.prototype.set_any
= function (val)
        {
                if
(cr.is_number(val))
                {
                    this.type =
cr.exptype.Float;
                    this.data =
val;
                }
                else if
(cr.is_string(val))
                {
                    this.type =
cr.exptype.String;
                    this.data =
val.toString();
                }
                else
                {
                    this.type =
cr.exptype.Integer;
                    this.data = 0;
                }
        };
        cr.expvalue = ExpValue;
        cr.exptype = {
                Integer: 0,     //
emulated; no native integer
support in javascript
                Float: 1,
                String: 2
        };
}());
;
cr.system_object = function
(runtime)
{
    this.runtime = runtime;
        this.waits = [];
};
cr.system_object.prototype.save
ToJSON = function ()
{
        var o = {};
        var i, len, j, lenj, p, w,
t, sobj;
        o["waits"] = [];
        var owaits = o["waits"];
```

```
	var waitobj;
	for (i = 0, len =
this.waits.length; i < len;
i++)
	{
		w = this.waits[i];
		waitobj = {
			"t": w.time,
			"ev": w.ev.sid,
			"sm": [],
			"sols": {}
		};
		if
(w.ev.actions[w.actindex])
			waitobj["act"]
= w.ev.actions[w.actindex].sid;
		for (j = 0, lenj =
w.solModifiers.length; j <
lenj; j++)

	waitobj["sm"].push(w.solMo
difiers[j].sid);
		for (p in w.sols)
		{
			if
(w.sols.hasOwnProperty(p))
			{
				t =
this.runtime.types_by_index[par
seInt(p, 10)];
;
				sobj = {

	"sa": w.sols[p].sa,

	"insts": []
				};
				for (j =
0, lenj =
w.sols[p].insts.length; j <
lenj; j++)

	sobj["insts"].push(w.sols[
p].insts[j].uid);

	waitobj["sols"][t.sid.toSt
ring()] = sobj;
			}
		}

	owaits.push(waitobj);
	}
	return o;
};
cr.system_object.prototype.load
FromJSON = function (o)
{
	var owaits = o["waits"];
	var i, len, j, lenj, p, w,
addWait, e, aindex, t,
savedsol, nusol, inst;
	this.waits.length = 0;
	for (i = 0, len =
owaits.length; i < len; i++)
	{
		w = owaits[i];
		e =
this.runtime.blocksBySid[w["ev"
].toString()];
		if (!e)
			continue;  //
event must've gone missing
		aindex = -1;
		for (j = 0, lenj =
e.actions.length; j < lenj;
j++)
		{
			if
(e.actions[j].sid === w["act"])
			{
				aindex =
j;
				break;
			}
		}
		if (aindex === -1)
			continue;  //
action must've gone missing
		addWait = {};
		addWait.sols = {};
		addWait.solModifiers
= [];
		addWait.deleteme =
false;
		addWait.time =
w["t"];
		addWait.ev = e;
		addWait.actindex =
aindex;
		for (j = 0, lenj =
w["sm"].length; j < lenj; j++)
		{
			t =
this.runtime.getObjectTypeBySid
(w["sm"][j]);
			if (t)

	addWait.solModifiers.push(
t);
		}
		for (p in w["sols"])
		{
```

```
                if
(w["sols"].hasOwnProperty(p))
                {
                        t =
this.runtime.getObjectTypeBySid
(parseInt(p, 10));
                        if (!t)

     continue;          // type
must've been deleted
                        savedsol
= w["sols"][p];
                        nusol = {
                              sa:
savedsol["sa"],

     insts: []
                        };
                        for (j =
0, lenj =
savedsol["insts"].length; j <
lenj; j++)
                        {
                              inst
=
this.runtime.getObjectByUID(sav
edsol["insts"][j]);
                              if
(inst)

     nusol.insts.push(inst);
                        }

     addWait.sols[t.index.toStr
ing()] = nusol;
                }
           }

     this.waits.push(addWait);
     }
};
(function ()
{
     var sysProto =
cr.system_object.prototype;
     function SysCnds() {};
    SysCnds.prototype.EveryTick
= function()
    {
        return true;
    };

SysCnds.prototype.OnLayoutStart
= function()
    {
        return true;
    };

SysCnds.prototype.OnLayoutEnd =
function()
    {
        return true;
    };
    SysCnds.prototype.Compare =
function(x, cmp, y)
    {
        return cr.do_cmp(x,
cmp, y);
    };

SysCnds.prototype.CompareTime =
function (cmp, t)
    {
        var elapsed =
this.runtime.kahanTime.sum;
        if (cmp === 0)
        {
            var cnd =
this.runtime.getCurrentConditio
n();
            if
(!cnd.extra.CompareTime_execute
d)
            {
                if (elapsed >=
t)
                {

cnd.extra.CompareTime_executed
= true;
                    return
true;
                }
            }
            return false;
        }
        return
cr.do_cmp(elapsed, cmp, t);
    };

SysCnds.prototype.LayerVisible
= function (layer)
    {
        if (!layer)
            return false;
        else
            return
layer.visible;
    };
     SysCnds.prototype.LayerCmp
Opacity = function (layer, cmp,
opacity_)
```

```
    {
        if (!layer)
            return false;
        return
cr.do_cmp(layer.opacity * 100,
cmp, opacity_);
    };
  SysCnds.prototype.Repeat =
function (count)
  {
        var current_frame =
this.runtime.getCurrentEventSta
ck();
      var current_event =
current_frame.current_event;
        var
solModifierAfterCnds =
current_frame.isModifierAfterCn
ds();
      var current_loop =
this.runtime.pushLoopStack();
      var i;
        if
(solModifierAfterCnds)
        {
            for (i = 0; i <
count && !current_loop.stopped;
i++)
            {

    this.runtime.pushCopySol(c
urrent_event.solModifiers);

    current_loop.index = i;

    current_event.retrigger();

    this.runtime.popSol(curren
t_event.solModifiers);
            }
        }
        else
        {
            for (i = 0; i <
count && !current_loop.stopped;
i++)
            {

    current_loop.index = i;

    current_event.retrigger();
            }
        }

this.runtime.popLoopStack();
        return false;
  };
    SysCnds.prototype.While =
function (count)
  {
        var current_frame =
this.runtime.getCurrentEventSta
ck();
      var current_event =
current_frame.current_event;
        var
solModifierAfterCnds =
current_frame.isModifierAfterCn
ds();
      var current_loop =
this.runtime.pushLoopStack();
      var i;
        if
(solModifierAfterCnds)
        {
            for (i = 0;
!current_loop.stopped; i++)
            {

    this.runtime.pushCopySol(c
urrent_event.solModifiers);

    current_loop.index = i;
                if
(!current_event.retrigger())
        // one of the other
conditions returned false

    current_loop.stopped =
true; // break

    this.runtime.popSol(curren
t_event.solModifiers);
            }
        }
        else
        {
            for (i = 0;
!current_loop.stopped; i++)
            {

    current_loop.index = i;
                if
(!current_event.retrigger())

    current_loop.stopped =
true;
            }
        }

this.runtime.popLoopStack();
        return false;
```

```
    };
    SysCnds.prototype.For =
function (name, start, end)
    {
        var current_frame =
this.runtime.getCurrentEventSta
ck();
        var current_event =
current_frame.current_event;
           var
solModifierAfterCnds =
current_frame.isModifierAfterCn
ds();
        var current_loop =
this.runtime.pushLoopStack(name
);
        var i;
           if (end < start)
           {
                 if
(solModifierAfterCnds)
                 {
                       for (i =
start; i >= end &&
!current_loop.stopped; --i)  //
inclusive to end
                       {

     this.runtime.pushCopySol(c
urrent_event.solModifiers);

     current_loop.index = i;

     current_event.retrigger();

     this.runtime.popSol(curren
t_event.solModifiers);
                       }
                 }
                 else
                 {
                       for (i =
start; i >= end &&
!current_loop.stopped; --i)  //
inclusive to end
                       {

     current_loop.index = i;

     current_event.retrigger();
                       }
                 }
           }
           else
           {
                 if
(solModifierAfterCnds)
                 {
                       for (i =
start; i <= end &&
!current_loop.stopped; ++i)  //
inclusive to end
                       {

     this.runtime.pushCopySol(c
urrent_event.solModifiers);

     current_loop.index = i;

     current_event.retrigger();

     this.runtime.popSol(curren
t_event.solModifiers);
                       }
                 }
                 else
                 {
                       for (i =
start; i <= end &&
!current_loop.stopped; ++i)  //
inclusive to end
                       {

     current_loop.index = i;

     current_event.retrigger();
                       }
                 }
           }

this.runtime.popLoopStack();
           return false;
    };
     var foreach_instancestack
= [];
     var foreach_instanceptr =
-1;
    SysCnds.prototype.ForEach =
function (obj)
    {
        var sol =
obj.getCurrentSol();

     foreach_instanceptr++;
           if
(foreach_instancestack.length
=== foreach_instanceptr)

     foreach_instancestack.push
([]);
```

```
            var instances =
foreach_instancestack[foreach_i
nstanceptr];

      cr.shallowAssignArray(inst
ances, sol.getObjects());
         var current_frame =
this.runtime.getCurrentEventSta
ck();
         var current_event =
current_frame.current_event;
            var
solModifierAfterCnds =
current_frame.isModifierAfterCn
ds();
         var current_loop =
this.runtime.pushLoopStack();
         var i, len, j, lenj,
inst, s, sol2;
            var is_contained =
obj.is_contained;
            if
(solModifierAfterCnds)
            {
                  for (i = 0, len
= instances.length; i < len &&
!current_loop.stopped; i++)
                  {

      this.runtime.pushCopySol(c
urrent_event.solModifiers);
                        inst =
instances[i];
                        sol =
obj.getCurrentSol();

      sol.select_all = false;

      sol.instances.length = 1;

      sol.instances[0] = inst;
                        if
(is_contained)
                        {
                              for
(j = 0, lenj =
inst.siblings.length; j < lenj;
j++)
                              {

      s = inst.siblings[j];

      sol2 =
s.type.getCurrentSol();

      sol2.select_all = false;

      sol2.instances.length = 1;

      sol2.instances[0] = s;
                              }
                        }

      current_loop.index = i;

      current_event.retrigger();

      this.runtime.popSol(curren
t_event.solModifiers);
                  }
            }
            else
            {
                  sol.select_all
= false;

      sol.instances.length = 1;
                  for (i = 0, len
= instances.length; i < len &&
!current_loop.stopped; i++)
                  {
                        inst =
instances[i];

      sol.instances[0] = inst;
                        if
(is_contained)
                        {
                              for
(j = 0, lenj =
inst.siblings.length; j < lenj;
j++)
                              {

      s = inst.siblings[j];

      sol2 =
s.type.getCurrentSol();

      sol2.select_all = false;

      sol2.instances.length = 1;

      sol2.instances[0] = s;
                              }
                        }

      current_loop.index = i;

      current_event.retrigger();
                  }
            }
```

```
            instances.length =
0;

this.runtime.popLoopStack();
            foreach_instanceptr-
-;
            return false;
    };
      function
foreach_sortinstances(a, b)
      {
            var va =
a.extra.c2_foreachordered_val;
            var vb =
b.extra.c2_foreachordered_val;
            if (cr.is_number(va)
&& cr.is_number(vb))
                  return va - vb;
            else
            {
                  va = "" + va;
                  vb = "" + vb;
                  if (va < vb)
                        return -
1;
                  else if (va >
vb)
                        return 1;
                  else
                        return 0;
            }
      };
      SysCnds.prototype.ForEachO
rdered = function (obj, exp,
order)
    {
        var sol =
obj.getCurrentSol();

      foreach_instanceptr++;
            if
(foreach_instancestack.length
=== foreach_instanceptr)

      foreach_instancestack.push
([]);
            var instances =
foreach_instancestack[foreach_i
nstanceptr];

      cr.shallowAssignArray(inst
ances, sol.getObjects());
        var current_frame =
this.runtime.getCurrentEventSta
ck();
        var current_event =
current_frame.current_event;
            var
current_condition =
this.runtime.getCurrentConditio
n();
            var
solModifierAfterCnds =
current_frame.isModifierAfterCn
ds();
        var current_loop =
this.runtime.pushLoopStack();
            var i, len, j, lenj,
inst, s, sol2;
            for (i = 0, len =
instances.length; i < len; i++)
            {

      instances[i].extra.c2_fore
achordered_val =
current_condition.parameters[1]
.get(i);
            }

      instances.sort(foreach_sor
tinstances);
            if (order === 1)

      instances.reverse();
            var is_contained =
obj.is_contained;
            if
(solModifierAfterCnds)
            {
                  for (i = 0, len
= instances.length; i < len &&
!current_loop.stopped; i++)
                  {

      this.runtime.pushCopySol(c
urrent_event.solModifiers);
                        inst =
instances[i];
                        sol =
obj.getCurrentSol();

      sol.select_all = false;

      sol.instances.length = 1;

      sol.instances[0] = inst;
                        if
(is_contained)
                        {
                              for
(j = 0, lenj =
```

```
inst.siblings.length; j < lenj;
j++)
                              {

      s = inst.siblings[j];

      sol2 =
s.type.getCurrentSol();

      sol2.select_all = false;

      sol2.instances.length = 1;

      sol2.instances[0] = s;
                              }
                        }

      current_loop.index = i;

      current_event.retrigger();

      this.runtime.popSol(curren
t_event.solModifiers);
                  }
            }
            else
            {
                  sol.select_all
= false;

      sol.instances.length = 1;
                  for (i = 0, len
= instances.length; i < len &&
!current_loop.stopped; i++)
                  {
                        inst =
instances[i];

      sol.instances[0] = inst;
                        if
(is_contained)
                        {
                              for
(j = 0, lenj =
inst.siblings.length; j < lenj;
j++)
                              {

      s = inst.siblings[j];

      sol2 =
s.type.getCurrentSol();

      sol2.select_all = false;

      sol2.instances.length = 1;

      sol2.instances[0] = s;
                              }
                        }

      current_loop.index = i;

      current_event.retrigger();
                  }
            }
            instances.length =
0;

this.runtime.popLoopStack();
            foreach_instanceptr-
-;
            return false;
      };
      SysCnds.prototype.PickByCo
mparison = function (obj_,
exp_, cmp_, val_)
      {
            var i, len, k, inst;
            if (!obj_)
                  return;

      foreach_instanceptr++;
            if
(foreach_instancestack.length
=== foreach_instanceptr)

      foreach_instancestack.push
([]);
            var tmp_instances =
foreach_instancestack[foreach_i
nstanceptr];
            var sol =
obj_.getCurrentSol();

      cr.shallowAssignArray(tmp_
instances, sol.getObjects());
            if (sol.select_all)

      sol.else_instances.length
= 0;
            var
current_condition =
this.runtime.getCurrentConditio
n();
            for (i = 0, k = 0,
len = tmp_instances.length; i <
len; i++)
            {
                  inst =
tmp_instances[i];
```

```
     tmp_instances[k] = inst;
                exp_ =
current_condition.parameters[1]
.get(i);
                val_ =
current_condition.parameters[3]
.get(i);
                if
(cr.do_cmp(exp_, cmp_, val_))
                {
                     k++;
                }
                else
                {

     sol.else_instances.push(in
st);
                }
           }
           tmp_instances.length
= k;
           sol.select_all =
false;

     cr.shallowAssignArray(sol.
instances, tmp_instances);
           tmp_instances.length
= 0;
           foreach_instanceptr-
-;

     obj_.applySolToContainer()
;
           return
!!sol.instances.length;
     };
     SysCnds.prototype.PickByEv
aluate = function (obj_, exp_)
     {
           var i, len, k, inst;
           if (!obj_)
                return;

     foreach_instanceptr++;
           if
(foreach_instancestack.length
=== foreach_instanceptr)

     foreach_instancestack.push
([]);
           var tmp_instances =
foreach_instancestack[foreach_i
nstanceptr];
           var sol =
obj_.getCurrentSol();
```

```
     cr.shallowAssignArray(tmp_
instances, sol.getObjects());
           if (sol.select_all)

     sol.else_instances.length
= 0;
           var
current_condition =
this.runtime.getCurrentConditio
n();
           for (i = 0, k = 0,
len = tmp_instances.length; i <
len; i++)
           {
                inst =
tmp_instances[i];

     tmp_instances[k] = inst;
                exp_ =
current_condition.parameters[1]
.get(i);
                if (exp_)
                {
                     k++;
                }
                else
                {

     sol.else_instances.push(in
st);
                }
           }
           tmp_instances.length
= k;
           sol.select_all =
false;

     cr.shallowAssignArray(sol.
instances, tmp_instances);
           tmp_instances.length
= 0;
           foreach_instanceptr-
-;

     obj_.applySolToContainer()
;
           return
!!sol.instances.length;
     };

SysCnds.prototype.TriggerOnce =
function ()
    {
```

```
        var cndextra =
this.runtime.getCurrentConditio
n().extra;
           if (typeof
cndextra.TriggerOnce_lastTick
=== "undefined")

     cndextra.TriggerOnce_lastT
ick = -1;
        var last_tick =
cndextra.TriggerOnce_lastTick;
        var cur_tick =
this.runtime.tickcount;

cndextra.TriggerOnce_lastTick =
cur_tick;
        return
this.runtime.layout_first_tick
|| last_tick !== cur_tick - 1;
    };
    SysCnds.prototype.Every =
function (seconds)
    {
        var cnd =
this.runtime.getCurrentConditio
n();
        var last_time =
cnd.extra.Every_lastTime || 0;
        var cur_time =
this.runtime.kahanTime.sum;
           if (typeof
cnd.extra.Every_seconds ===
"undefined")

     cnd.extra.Every_seconds =
seconds;
           var this_seconds =
cnd.extra.Every_seconds;
        if (cur_time >=
last_time + this_seconds)
        {

cnd.extra.Every_lastTime =
last_time + this_seconds;
                if (cur_time >=
cnd.extra.Every_lastTime +
this_seconds)

     cnd.extra.Every_lastTime =
cur_time;

     cnd.extra.Every_seconds =
seconds;
            return true;
        }
        else
            return false;
    };
    SysCnds.prototype.PickNth =
function (obj, index)
    {
        if (!obj)
            return false;
        var sol =
obj.getCurrentSol();
        var instances =
sol.getObjects();
           index =
cr.floor(index);
        if (index < 0 || index
>= instances.length)
            return false;
           var inst =
instances[index];
        sol.pick_one(inst);

     obj.applySolToContainer();
        return true;
    };
     SysCnds.prototype.PickRand
om = function (obj)
    {
        if (!obj)
            return false;
        var sol =
obj.getCurrentSol();
        var instances =
sol.getObjects();
           var index =
cr.floor(Math.random() *
instances.length);
        if (index >=
instances.length)
            return false;
           var inst =
instances[index];
        sol.pick_one(inst);

     obj.applySolToContainer();
        return true;
    };
     SysCnds.prototype.CompareV
ar = function (v, cmp, val)
    {
        return
cr.do_cmp(v.getValue(), cmp,
val);
    };

SysCnds.prototype.IsGroupActive
= function (group)
    {
```

```
        return
this.runtime.activeGroups[(/*th
is.runtime.getCurrentCondition(
).sheet.name + "|" +
*/group).toLowerCase()];
    };
     SysCnds.prototype.IsPrevie
w = function ()
     {
          return typeof
cr_is_preview !== "undefined";
     };
     SysCnds.prototype.PickAll
= function (obj)
    {
        if (!obj)
            return false;
          if
(!obj.instances.length)
                return false;
        var sol =
obj.getCurrentSol();
        sol.select_all = true;

     obj.applySolToContainer();
        return true;
    };
     SysCnds.prototype.IsMobile
= function ()
     {
          return
this.runtime.isMobile;
     };
     SysCnds.prototype.CompareB
etween = function (x, a, b)
     {
          return x >= a && x
<= b;
     };
     SysCnds.prototype.Else =
function ()
     {
          var current_frame =
this.runtime.getCurrentEventSta
ck();
          if
(current_frame.else_branch_ran)
                return false;
          // another event in
this else-if chain has run
          else
                return
!current_frame.last_event_true;
          /*
          var current_frame =
this.runtime.getCurrentEventSta
ck();
       var current_event =
current_frame.current_event;
          var prev_event =
current_event.prev_block;
          if (!prev_event)
                return false;
          if
(prev_event.is_logical)
                return
!this.runtime.last_event_true;
          var i, len, j, lenj,
s, sol, temp, inst, any_picked
= false;
          for (i = 0, len =
prev_event.cndReferences.length
; i < len; i++)
          {
                s =
prev_event.cndReferences[i];
                sol =
s.getCurrentSol();
                if
(sol.select_all ||
sol.instances.length ===
s.instances.length)
                {

     sol.select_all = false;

     sol.instances.length = 0;
                }
                else
                {
                      if
(sol.instances.length === 1 &&
sol.else_instances.length === 0
&& s.instances.length >= 2)
                      {
                            inst
= sol.instances[0];

     sol.instances.length = 0;
                            for
(j = 0, lenj =
s.instances.length; j < lenj;
j++)
                            {

     if (s.instances[j] !=
inst)
```

```
	sol.instances.push(s.insta
nces[j]);
				}

	any_picked = true;
				}
				else
				{
					temp
= sol.instances;

	sol.instances =
sol.else_instances;

	sol.else_instances = temp;

	any_picked = true;
				}
			}
		}
		return any_picked;
		*/
	};
	SysCnds.prototype.OnLoadFi
nished = function ()
	{
		return true;
	};
	SysCnds.prototype.OnCanvas
Snapshot = function ()
	{
		return true;
	};
	SysCnds.prototype.EffectsS
upported = function ()
	{
		return
!!this.runtime.glwrap;
	};
	SysCnds.prototype.OnSaveCo
mplete = function ()
	{
		return true;
	};
	SysCnds.prototype.OnLoadCo
mplete = function ()
	{
		return true;
	};
	SysCnds.prototype.OnLoadFa
iled = function ()
	{
		return true;
	};
	SysCnds.prototype.ObjectUI
DExists = function (u)
	{
		return
!!this.runtime.getObjectByUID(u
);
	};
	SysCnds.prototype.IsOnPlat
form = function (p)
	{
		var rt =
this.runtime;
		switch (p) {
		case 0:		//
HTML5 website
			return
!rt.isDomFree &&
!rt.isNodeWebkit &&
!rt.isPhoneGap &&
!rt.isCrosswalk &&
!rt.isWindows8App &&
!rt.isWindowsPhone8 &&
!rt.isBlackberry10 &&
!rt.isAmazonWebApp;
		case 1:		//
iOS
			return
rt.isiOS;
		case 2:		//
Android
			return
rt.isAndroid;
		case 3:		//
Windows 8
			return
rt.isWindows8App;
		case 4:		//
Windows Phone 8
			return
rt.isWindowsPhone8;
		case 5:		//
Blackberry 10
			return
rt.isBlackberry10;
		case 6:		//
Tizen
			return
rt.isTizen;
		case 7:		//
CocoonJS
			return
rt.isCocoonJs;
		case 8:		//
PhoneGap
			return
rt.isPhoneGap;
		case 9:	// Scirra
Arcade
```

```
                return
rt.isArcade;
            case 10:   // node-
webkit
                return
rt.isNodeWebkit;
            case 11:   //
crosswalk
                return
rt.isCrosswalk;
            case 12:   // amazon
webapp
                return
rt.isAmazonWebApp;
            default:   // should
not be possible
                return false;
            }
        };
        var cacheRegex = null;
        var lastRegex = "";
        var lastFlags = "";
        function getRegex(regex_,
flags_)
        {
            if (!cacheRegex ||
regex_ !== lastRegex || flags_
!== lastFlags)
            {
                cacheRegex =
new RegExp(regex_, flags_);
                lastRegex =
regex_;
                lastFlags =
flags_;
            }
            cacheRegex.lastIndex
= 0;        // reset
            return cacheRegex;
        };
        SysCnds.prototype.RegexTes
t = function (str_, regex_,
flags_)
        {
            var regex =
getRegex(regex_, flags_);
            return
regex.test(str_);
        };
        var tmp_arr = [];
        SysCnds.prototype.PickOver
lappingPoint = function (obj_,
x_, y_)
        {
            if (!obj_)
             return false;
            var sol =
obj_.getCurrentSol();
            var instances =
sol.getObjects();
                var current_event =
this.runtime.getCurrentEventSta
ck().current_event;
                var orblock =
current_event.orblock;
                var cnd =
this.runtime.getCurrentConditio
n();
                var i, len, inst,
pick;
                if (sol.select_all)
                {

        cr.shallowAssignArray(tmp_
arr, instances);

        sol.else_instances.length
= 0;
                    sol.select_all
= false;

        sol.instances.length = 0;
                }
                else
                {
                    if (orblock)
                    {

        cr.shallowAssignArray(tmp_
arr, sol.else_instances);

        sol.else_instances.length
= 0;
                    }
                    else
                    {

        cr.shallowAssignArray(tmp_
arr, instances);

        sol.instances.length = 0;
                    }
                }
                for (i = 0, len =
tmp_arr.length; i < len; ++i)
                {
                    inst =
tmp_arr[i];
                    pick =
cr.xor(inst.contains_pt(x_,
y_), cnd.inverted);
                    if (pick)
```

```
      sol.instances.push(inst);
                else
      sol.else_instances.push(in
st);
            }

      obj_.applySolToContainer()
;
            return
cr.xor(!!sol.instances.length,
cnd.inverted);
      };
      sysProto.cnds = new
SysCnds();
    function SysActs() {};

SysActs.prototype.GoToLayout =
function(to)
    {
            if
(this.runtime.isloading)
                return;
      // cannot change layout
while loading on loader layout
            if
(this.runtime.changelayout)
                return;
      // already changing to a
different layout
;

this.runtime.changelayout = to;
    };

SysActs.prototype.CreateObject
= function (obj, layer, x, y)
    {
        if (!layer || !obj)
            return;
        var inst =
this.runtime.createInstance(obj
, layer, x, y);
            if (!inst)
                return;

      this.runtime.isInOnDestroy
++;
            var i, len, s;

      this.runtime.trigger(Objec
t.getPrototypeOf(obj.plugin).cn
ds.OnCreated, inst);
            if
(inst.is_contained)
            {
                for (i = 0, len
= inst.siblings.length; i <
len; i++)
                {
                    s =
inst.siblings[i];

      this.runtime.trigger(Objec
t.getPrototypeOf(s.type.plugin)
.cnds.OnCreated, s);
                }
            }

      this.runtime.isInOnDestroy
--;
        var sol =
obj.getCurrentSol();
        sol.select_all = false;
            sol.instances.length
= 1;
            sol.instances[0] =
inst;
            if
(inst.is_contained)
            {
                for (i = 0, len
= inst.siblings.length; i <
len; i++)
                {
                    s =
inst.siblings[i];
                    sol =
s.type.getCurrentSol();

      sol.select_all = false;

      sol.instances.length = 1;

      sol.instances[0] = s;
                }
            }
    };

SysActs.prototype.SetLayerVisib
le = function (layer, visible_)
    {
        if (!layer)
            return;
            if (layer.visible
!== visible_)
            {
                layer.visible =
visible_;
```

```
	this.runtime.redraw =
true;
		}
	};
	SysActs.prototype.SetLayer
Opacity = function (layer,
opacity_)
	{
		if (!layer)
			return;
		opacity_ =
cr.clamp(opacity_ / 100, 0, 1);
		if (layer.opacity
!== opacity_)
		{
			layer.opacity =
opacity_;

	this.runtime.redraw =
true;
		}
	};
	SysActs.prototype.SetLayer
ScaleRate = function (layer,
sr)
	{
		if (!layer)
			return;
		if (layer.zoomRate
!== sr)
		{
			layer.zoomRate
= sr;

	this.runtime.redraw =
true;
		}
	};
	SysActs.prototype.SetLayou
tScale = function (s)
	{
		if
(!this.runtime.running_layout)
			return;
		if
(this.runtime.running_layout.sc
ale !== s)
		{

	this.runtime.running_layou
t.scale = s;

	this.runtime.running_layou
t.boundScrolling();

	this.runtime.redraw =
true;
		}
	};
	SysActs.prototype.ScrollX =
function(x)
	{

this.runtime.running_layout.scr
ollToX(x);
	};
	SysActs.prototype.ScrollY =
function(y)
	{

this.runtime.running_layout.scr
ollToY(y);
	};
	SysActs.prototype.Scroll =
function(x, y)
	{

this.runtime.running_layout.scr
ollToX(x);

this.runtime.running_layout.scr
ollToY(y);
	};

SysActs.prototype.ScrollToObjec
t = function(obj)
	{
		var inst =
obj.getFirstPicked();
		if (inst)
		{

this.runtime.running_layout.scr
ollToX(inst.x);

this.runtime.running_layout.scr
ollToY(inst.y);
		}
	};
	SysActs.prototype.SetVar =
function(v, x)
	{
;
		if (v.vartype === 0)
		{
			if
(cr.is_number(x))

	v.setValue(x);
			else
```

```
     v.setValue(parseFloat(x));
          }
          else if (v.vartype
=== 1)

     v.setValue(x.toString());
     };
     SysActs.prototype.AddVar =
function(v, x)
     {
;
          if (v.vartype === 0)
          {
               if
(cr.is_number(x))

     v.setValue(v.getValue() +
x);
               else

     v.setValue(v.getValue() +
parseFloat(x));
          }
          else if (v.vartype
=== 1)

     v.setValue(v.getValue() +
x.toString());
     };
     SysActs.prototype.SubVar =
function(v, x)
     {
;
          if (v.vartype === 0)
          {
               if
(cr.is_number(x))

     v.setValue(v.getValue() -
x);
               else

     v.setValue(v.getValue() -
parseFloat(x));
          }
     };

SysActs.prototype.SetGroupActiv
e = function (group, active)
    {
          var activeGroups =
this.runtime.activeGroups;
          var groupkey =
(/*this.runtime.getCurrentActio
n().sheet.name + "|" +
*/group).toLowerCase();
          switch (active) {
          case 0:

     activeGroups[groupkey] =
false;
               break;
          case 1:

     activeGroups[groupkey] =
true;
               break;
          case 2:

     activeGroups[groupkey] =
!activeGroups[groupkey];
               break;
          }
    };

SysActs.prototype.SetTimescale
= function (ts_)
    {
        var ts = ts_;
        if (ts < 0)
            ts = 0;
        this.runtime.timescale
= ts;
    };

SysActs.prototype.SetObjectTime
scale = function (obj, ts_)
    {
        var ts = ts_;
        if (ts < 0)
            ts = 0;
        if (!obj)
            return;
        var sol =
obj.getCurrentSol();
        var instances =
sol.getObjects();
        var i, len;
        for (i = 0, len =
instances.length; i < len; i++)
        {

instances[i].my_timescale = ts;
        }
    };

SysActs.prototype.RestoreObject
Timescale = function (obj)
    {
        if (!obj)
```

```
            return false;
        var sol =
obj.getCurrentSol();
        var instances =
sol.getObjects();
        var i, len;
        for (i = 0, len =
instances.length; i < len; i++)
        {

instances[i].my_timescale = -
1.0;
        }
    };
     var waitobjrecycle = [];
     function allocWaitObject()
     {
          var w;
          if
(waitobjrecycle.length)
               w =
waitobjrecycle.pop();
          else
          {
               w = {};
               w.sols = {};
               w.solModifiers
= [];
          }
          w.deleteme = false;
          return w;
     };
     function freeWaitObject(w)
     {
          cr.wipe(w.sols);

     w.solModifiers.length = 0;

     waitobjrecycle.push(w);
     };
     var solstateobjects = [];
     function
allocSolStateObject()
     {
          var s;
          if
(solstateobjects.length)
               s =
solstateobjects.pop();
          else
          {
               s = {};
               s.insts = [];
          }
          s.sa = false;
          return s;
     };
     function
freeSolStateObject(s)
     {
          s.insts.length = 0;

     solstateobjects.push(s);
     };
     SysActs.prototype.Wait =
function (seconds)
     {
          if (seconds < 0)
               return;
          var i, len, s, t,
ss;
          var evinfo =
this.runtime.getCurrentEventSta
ck();
          var waitobj =
allocWaitObject();
          waitobj.time =
this.runtime.kahanTime.sum +
seconds;
          waitobj.ev =
evinfo.current_event;
          waitobj.actindex =
evinfo.actindex + 1;  //
pointing at next action
          for (i = 0, len =
this.runtime.types_by_index.len
gth; i < len; i++)
          {
               t =
this.runtime.types_by_index[i];
               s =
t.getCurrentSol();
               if
(s.select_all &&
evinfo.current_event.solModifie
rs.indexOf(t) === -1)
                    continue;

     waitobj.solModifiers.push(
t);
               ss =
allocSolStateObject();
               ss.sa =
s.select_all;

     cr.shallowAssignArray(ss.i
nsts, s.instances);

     waitobj.sols[i.toString()]
= ss;
          }
```

```
     this.waits.push(waitobj);
          return true;
     };
     SysActs.prototype.SetLayer
Scale = function (layer, scale)
    {
        if (!layer)
            return;
           if (layer.scale ===
scale)
                return;
        layer.scale = scale;
        this.runtime.redraw =
true;
    };
     SysActs.prototype.ResetGlo
bals = function ()
     {
          var i, len, g;
          for (i = 0, len =
this.runtime.all_global_vars.le
ngth; i < len; i++)
          {
               g =
this.runtime.all_global_vars[i]
;
               g.data =
g.initial;
          }
     };
     SysActs.prototype.SetLayou
tAngle = function (a)
     {
          a =
cr.to_radians(a);
          a =
cr.clamp_angle(a);
          if
(this.runtime.running_layout)
          {
               if
(this.runtime.running_layout.an
gle !== a)
               {

     this.runtime.running_layou
t.angle = a;

     this.runtime.redraw =
true;
               }
          }
     };
     SysActs.prototype.SetLayer
Angle = function (layer, a)
    {
        if (!layer)
            return;
           a =
cr.to_radians(a);
           a =
cr.clamp_angle(a);
           if (layer.angle ===
a)
                return;
        layer.angle = a;
        this.runtime.redraw =
true;
    };
     SysActs.prototype.SetLayer
Parallax = function (layer, px,
py)
    {
        if (!layer)
            return;
           if (layer.parallaxX
=== px / 100 && layer.parallaxY
=== py / 100)
                return;
        layer.parallaxX = px /
100;
           layer.parallaxY = py
/ 100;
           if (layer.parallaxX
!== 1 || layer.parallaxY !== 1)
           {
                var i, len,
instances = layer.instances;
                for (i = 0, len
= instances.length; i < len;
++i)
                {

     instances[i].type.any_inst
ance_parallaxed = true;
                }
           }
        this.runtime.redraw =
true;
    };
     SysActs.prototype.SetLayer
Background = function (layer,
c)
    {
        if (!layer)
            return;
           var r =
cr.GetRValue(c);
           var g =
cr.GetGValue(c);
```

```
            var b =
cr.GetBValue(c);
            if
(layer.background_color[0] ===
r && layer.background_color[1]
=== g &&
layer.background_color[2] ===
b)
                return;

layer.background_color[0] = r;

      layer.background_color[1]
= g;

      layer.background_color[2]
= b;
        this.runtime.redraw =
true;
    };
      SysActs.prototype.SetLayer
Transparent = function (layer,
t)
    {
        if (!layer)
            return;
            if (!!t ===
!!layer.transparent)
                return;
            layer.transparent =
!!t;
        this.runtime.redraw =
true;
    };
      SysActs.prototype.StopLoop
= function ()
      {
            if
(this.runtime.loop_stack_index
< 0)
                return;
      // no loop currently
running

      this.runtime.getCurrentLoo
p().stopped = true;
      };
      SysActs.prototype.GoToLayo
utByName = function
(layoutname)
      {
            if
(this.runtime.isloading)
                return;
      // cannot change layout
while loading on loader layout
            if
(this.runtime.changelayout)
                return;
      // already changing to
different layout
;
            var l;
            for (l in
this.runtime.layouts)
            {
                  if
(this.runtime.layouts.hasOwnPro
perty(l) && cr.equals_nocase(l,
layoutname))
                  {

      this.runtime.changelayout
= this.runtime.layouts[l];
                        return;
                  }
            }
      };
      SysActs.prototype.RestartL
ayout = function (layoutname)
      {
            if
(this.runtime.isloading)
                return;
      // cannot restart loader
layouts
            if
(this.runtime.changelayout)
                return;
      // already changing to a
different layout
;
            if
(!this.runtime.running_layout)
                return;

      this.runtime.changelayout
= this.runtime.running_layout;
            var i, len, g;
            for (i = 0, len =
this.runtime.allGroups.length;
i < len; i++)
            {
                  g =
this.runtime.allGroups[i];

      this.runtime.activeGroups[
g.group_name.toLowerCase()] =
g.initially_activated;
            }
      };
```

```
	SysActs.prototype.SnapshotCanvas = function (format_, quality_)
	{

	this.runtime.snapshotCanvas = [format_ === 0 ? "image/png" : "image/jpeg", quality_ / 100];
		this.runtime.redraw = true;		// force redraw so snapshot is always taken
	};
	SysActs.prototype.SetCanvasSize = function (w, h)
	{
		if (w <= 0 || h <= 0)
			return;
		var mode = this.runtime.fullscreen_mode;
		var isfullscreen = (document["mozFullScreen"] || document["webkitIsFullScreen"] || !!document["msFullscreenElement"] || document["fullScreen"] || this.runtime.isNodeFullscreen);
		if (isfullscreen && this.runtime.fullscreen_scaling > 0)
			mode = this.runtime.fullscreen_scaling;
		if (mode === 0)
		{

	this.runtime["setSize"](w, h, true);
		}
		else
		{

	this.runtime.original_width = w;

	this.runtime.original_height = h;

	this.runtime["setSize"](this.runtime.lastWindowWidth, this.runtime.lastWindowHeight, true);
		}
	};
	SysActs.prototype.SetLayoutEffectEnabled = function (enable_, effectname_)
	{
		if (!this.runtime.running_layout || !this.runtime.glwrap)
			return;
		var et = this.runtime.running_layout.getEffectByName(effectname_);
		if (!et)
			return;
	// effect name not found
		var enable = (enable_ === 1);
		if (et.active == enable)
			return;
	// no change
		et.active = enable;

	this.runtime.running_layout.updateActiveEffects();
		this.runtime.redraw = true;
	};
	SysActs.prototype.SetLayerEffectEnabled = function (layer, enable_, effectname_)
	{
		if (!layer || !this.runtime.glwrap)
			return;
		var et = layer.getEffectByName(effectname_);
		if (!et)
			return;
	// effect name not found
		var enable = (enable_ === 1);
		if (et.active == enable)
			return;
	// no change
		et.active = enable;

	layer.updateActiveEffects();
		this.runtime.redraw = true;
	};
	SysActs.prototype.SetLayoutEffectParam = function (effectname_, index_, value_)
```

```
	{
		if (!this.runtime.running_layout || !this.runtime.glwrap)
			return;
		var et = this.runtime.running_layout.getEffectByName(effectname_);
		if (!et)
			return;
	// effect name not found
		var params = this.runtime.running_layout.effect_params[et.index];
		index_ = Math.floor(index_);
		if (index_ < 0 || index_ >= params.length)
			return;
	// effect index out of bounds
		if (this.runtime.glwrap.getProgramParameterType(et.shaderindex, index_) === 1)
			value_ /= 100.0;
		if (params[index_] === value_)
			return;
	// no change
		params[index_] = value_;
		if (et.active)

	this.runtime.redraw = true;
	};
	SysActs.prototype.SetLayerEffectParam = function (layer, effectname_, index_, value_)
	{
		if (!layer || !this.runtime.glwrap)
			return;
		var et = layer.getEffectByName(effectname_);
		if (!et)
			return;
	// effect name not found
		var params = layer.effect_params[et.index];
		index_ = Math.floor(index_);
		if (index_ < 0 || index_ >= params.length)
			return;
	// effect index out of bounds
		if (this.runtime.glwrap.getProgramParameterType(et.shaderindex, index_) === 1)
			value_ /= 100.0;
		if (params[index_] === value_)
			return;
	// no change
		params[index_] = value_;
		if (et.active)

	this.runtime.redraw = true;
	};
	SysActs.prototype.SaveState = function (slot_)
	{

	this.runtime.saveToSlot = slot_;
	};
	SysActs.prototype.LoadState = function (slot_)
	{

	this.runtime.loadFromSlot = slot_;
	};
	SysActs.prototype.LoadStateJSON = function (jsonstr_)
	{

	this.runtime.loadFromJson = jsonstr_;
	};
	SysActs.prototype.SetHalfFramerateMode = function (set_)
	{

	this.runtime.halfFramerateMode = (set_ !== 0);
	};
	SysActs.prototype.SetFullscreenQuality = function (q)
	{
		var isfullscreen = (document["mozFullScreen"] || document["webkitIsFullScreen"]
```

```
||
!!document["msFullscreenElement
"] || document["fullScreen"] ||
this.isNodeFullscreen);
            if (!isfullscreen &&
this.runtime.fullscreen_mode
=== 0)
                return;

      this.runtime.wantFullscree
nScalingQuality = (q !== 0);

      this.runtime["setSize"](th
is.runtime.lastWindowWidth,
this.runtime.lastWindowHeight,
true);
      };
      sysProto.acts = new
SysActs();
    function SysExps() {};
    SysExps.prototype["int"] =
function(ret, x)
    {
        if (cr.is_string(x))
        {

ret.set_int(parseInt(x, 10));
            if
(isNaN(ret.data))
                ret.data = 0;
        }
        else
            ret.set_int(x);
    };
    SysExps.prototype["float"]
= function(ret, x)
    {
        if (cr.is_string(x))
        {

ret.set_float(parseFloat(x));
            if
(isNaN(ret.data))
                ret.data = 0;
        }
        else
            ret.set_float(x);
    };
    SysExps.prototype.str =
function(ret, x)
    {
        if (cr.is_string(x))
            ret.set_string(x);
        else

ret.set_string(x.toString());
    };
    SysExps.prototype.len =
function(ret, x)
    {
        ret.set_int(x.length ||
0);
    };
    SysExps.prototype.random =
function (ret, a, b)
    {
        if (b === undefined)
        {

ret.set_float(Math.random() *
a);
        }
        else
        {

ret.set_float(Math.random() *
(b - a) + a);
        }
    };
    SysExps.prototype.sqrt =
function(ret, x)
    {

ret.set_float(Math.sqrt(x));
    };
    SysExps.prototype.abs =
function(ret, x)
    {

ret.set_float(Math.abs(x));
    };
    SysExps.prototype.round =
function(ret, x)
    {

ret.set_int(Math.round(x));
    };
    SysExps.prototype.floor =
function(ret, x)
    {

ret.set_int(Math.floor(x));
    };
    SysExps.prototype.ceil =
function(ret, x)
    {

ret.set_int(Math.ceil(x));
    };
    SysExps.prototype.sin =
function(ret, x)
    {
```

```
ret.set_float(Math.sin(cr.to_ra
dians(x)));
    };
    SysExps.prototype.cos =
function(ret, x)
    {

ret.set_float(Math.cos(cr.to_ra
dians(x)));
    };
    SysExps.prototype.tan =
function(ret, x)
    {

ret.set_float(Math.tan(cr.to_ra
dians(x)));
    };
    SysExps.prototype.asin =
function(ret, x)
    {

ret.set_float(cr.to_degrees(Mat
h.asin(x)));
    };
    SysExps.prototype.acos =
function(ret, x)
    {

ret.set_float(cr.to_degrees(Mat
h.acos(x)));
    };
    SysExps.prototype.atan =
function(ret, x)
    {

ret.set_float(cr.to_degrees(Mat
h.atan(x)));
    };
    SysExps.prototype.exp =
function(ret, x)
    {

ret.set_float(Math.exp(x));
    };
    SysExps.prototype.ln =
function(ret, x)
    {

ret.set_float(Math.log(x));
    };
    SysExps.prototype.log10 =
function(ret, x)
    {

ret.set_float(Math.log(x) /
Math.LN10);
    };
    SysExps.prototype.max =
function(ret)
    {
            var max_ =
arguments[1];
            var i, len;
            for (i = 2, len =
arguments.length; i < len; i++)
            {
                  if (max_ <
arguments[i])
                        max_ =
arguments[i];
            }
            ret.set_float(max_);
    };
    SysExps.prototype.min =
function(ret)
    {
        var min_ =
arguments[1];
            var i, len;
            for (i = 2, len =
arguments.length; i < len; i++)
            {
                  if (min_ >
arguments[i])
                        min_ =
arguments[i];
            }
            ret.set_float(min_);
    };
    SysExps.prototype.dt =
function(ret)
    {

ret.set_float(this.runtime.dt);
    };
    SysExps.prototype.timescale
= function(ret)
    {

ret.set_float(this.runtime.time
scale);
    };

SysExps.prototype.wallclocktime
= function(ret)
    {

ret.set_float((Date.now() -
```

```
this.runtime.start_time) /
1000.0);
    };
    SysExps.prototype.time =
function(ret)
    {

ret.set_float(this.runtime.kaha
nTime.sum);
    };
    SysExps.prototype.tickcount
= function(ret)
    {

ret.set_int(this.runtime.tickco
unt);
    };

SysExps.prototype.objectcount =
function(ret)
    {

ret.set_int(this.runtime.object
count);
    };
    SysExps.prototype.fps =
function(ret)
    {

ret.set_int(this.runtime.fps);
    };
    SysExps.prototype.loopindex
= function(ret, name_)
    {
            var loop, i, len;
        if
(!this.runtime.loop_stack.lengt
h)
        {
            ret.set_int(0);
            return;
        }
        if (name_)
        {
            for (i = 0, len =
this.runtime.loop_stack.length;
i < len; i++)
            {
                loop =
this.runtime.loop_stack[i];
                if (loop.name
=== name_)
                {

ret.set_int(loop.index);
                    return;
                }
            }
            ret.set_int(0);
        }
        else
        {
                loop =
this.runtime.getCurrentLoop();

     ret.set_int(loop ?
loop.index : -1);
        }
    };
    SysExps.prototype.distance
= function(ret, x1, y1, x2, y2)
    {

ret.set_float(cr.distanceTo(x1,
y1, x2, y2));
    };
    SysExps.prototype.angle =
function(ret, x1, y1, x2, y2)
    {

ret.set_float(cr.to_degrees(cr.
angleTo(x1, y1, x2, y2)));
    };
    SysExps.prototype.scrollx =
function(ret)
    {

ret.set_float(this.runtime.runn
ing_layout.scrollX);
    };
    SysExps.prototype.scrolly =
function(ret)
    {

ret.set_float(this.runtime.runn
ing_layout.scrollY);
    };
    SysExps.prototype.newline =
function(ret)
    {
        ret.set_string("\n");
    };
    SysExps.prototype.lerp =
function(ret, a, b, x)
    {

ret.set_float(cr.lerp(a, b,
x));
    };

SysExps.prototype.windowwidth =
function(ret)
```

```
    {
ret.set_int(this.runtime.width)
;
    };
SysExps.prototype.windowheight
= function(ret)
    {
ret.set_int(this.runtime.height
);
    };
     SysExps.prototype.uppercas
e = function(ret, str)
     {
     ret.set_string(cr.is_strin
g(str) ? str.toUpperCase() :
"");
     };
     SysExps.prototype.lowercas
e = function(ret, str)
     {
     ret.set_string(cr.is_strin
g(str) ? str.toLowerCase() :
"");
     };
     SysExps.prototype.clamp =
function(ret, x, l, u)
     {
          if (x < l)
     ret.set_float(l);
          else if (x > u)
     ret.set_float(u);
          else
     ret.set_float(x);
     };
     SysExps.prototype.layersca
le = function (ret, layerparam)
     {
          var layer =
this.runtime.getLayer(layerpara
m);
          if (!layer)
     ret.set_float(0);
          else
     ret.set_float(layer.scale)
;
     };
     SysExps.prototype.layeropa
city = function (ret,
layerparam)
     {
          var layer =
this.runtime.getLayer(layerpara
m);
          if (!layer)
     ret.set_float(0);
          else
     ret.set_float(layer.opacit
y * 100);
     };
     SysExps.prototype.layersca
lerate = function (ret,
layerparam)
     {
          var layer =
this.runtime.getLayer(layerpara
m);
          if (!layer)
     ret.set_float(0);
          else
     ret.set_float(layer.zoomRa
te);
     };
     SysExps.prototype.layerpar
allaxx = function (ret,
layerparam)
     {
          var layer =
this.runtime.getLayer(layerpara
m);
          if (!layer)
     ret.set_float(0);
          else
     ret.set_float(layer.parall
axX * 100);
     };
     SysExps.prototype.layerpar
allaxy = function (ret,
layerparam)
     {
          var layer =
this.runtime.getLayer(layerpara
m);
          if (!layer)
     ret.set_float(0);
          else
```

```
	ret.set_float(layer.parallaxY * 100);
	};
	SysExps.prototype.layoutscale = function (ret)
	{
		if (this.runtime.running_layout)
	ret.set_float(this.runtime.running_layout.scale);
		else
	ret.set_float(0);
	};
	SysExps.prototype.layoutangle = function (ret)
	{
	ret.set_float(cr.to_degrees(this.runtime.running_layout.angle));
	};
	SysExps.prototype.layerangle = function (ret, layerparam)
	{
		var layer = this.runtime.getLayer(layerparam);
		if (!layer)
	ret.set_float(0);
		else
	ret.set_float(cr.to_degrees(layer.angle));
	};
	SysExps.prototype.layoutwidth = function (ret)
	{
	ret.set_int(this.runtime.running_layout.width);
	};
	SysExps.prototype.layoutheight = function (ret)
	{
	ret.set_int(this.runtime.running_layout.height);
	};
	SysExps.prototype.find = function (ret, text, searchstr)
	{
		if (cr.is_string(text) && cr.is_string(searchstr))
	ret.set_int(text.search(new RegExp(cr.regexp_escape(searchstr), "i")));
		else
			ret.set_int(-1);
	};
	SysExps.prototype.left = function (ret, text, n)
	{
	ret.set_string(cr.is_string(text) ? text.substr(0, n) : "");
	};
	SysExps.prototype.right = function (ret, text, n)
	{
	ret.set_string(cr.is_string(text) ? text.substr(text.length - n) : "");
	};
	SysExps.prototype.mid = function (ret, text, index_, length_)
	{
	ret.set_string(cr.is_string(text) ? text.substr(index_, length_) : "");
	};
	SysExps.prototype.tokenat = function (ret, text, index_, sep)
	{
		if (cr.is_string(text) && cr.is_string(sep))
		{
			var arr = text.split(sep);
			var i = cr.floor(index_);
			if (i < 0 || i >= arr.length)
	ret.set_string("");
			else
```

```
	ret.set_string(arr[i]);
		}
		else

	ret.set_string("");
	};
	SysExps.prototype.tokencou
nt = function (ret, text, sep)
	{
		if
(cr.is_string(text) &&
text.length)

	ret.set_int(text.split(sep
).length);
		else
			ret.set_int(0);
	};
	SysExps.prototype.replace
= function (ret, text, find_,
replace_)
	{
		if
(cr.is_string(text) &&
cr.is_string(find_) &&
cr.is_string(replace_))

	ret.set_string(text.replac
e(new
RegExp(cr.regexp_escape(find_),
"gi"), replace_));
		else

	ret.set_string(cr.is_strin
g(text) ? text : "");
	};
	SysExps.prototype.trim =
function (ret, text)
	{

	ret.set_string(cr.is_strin
g(text) ? text.trim() : "");
	};
	SysExps.prototype.pi =
function (ret)
	{

	ret.set_float(cr.PI);
	};
	SysExps.prototype.layoutna
me = function (ret)
	{
		if
(this.runtime.running_layout)

	ret.set_string(this.runtim
e.running_layout.name);
		else

	ret.set_string("");
	};
	SysExps.prototype.renderer
= function (ret)
	{

	ret.set_string(this.runtim
e.gl ? "webgl" : "canvas2d");
	};
	SysExps.prototype.angledif
f = function (ret, a, b)
	{

	ret.set_float(cr.to_degree
s(cr.angleDiff(cr.to_radians(a)
, cr.to_radians(b))));
	};
	SysExps.prototype.choose =
function (ret)
	{
		var index =
cr.floor(Math.random() *
(arguments.length - 1));

	ret.set_any(arguments[inde
x + 1]);
	};
	SysExps.prototype.rgb =
function (ret, r, g, b)
	{

	ret.set_int(cr.RGB(r, g,
b));
	};
	SysExps.prototype.projectv
ersion = function (ret)
	{

	ret.set_string(this.runtim
e.versionstr);
	};
	SysExps.prototype.angleler
p = function (ret, a, b, x)
	{
		a =
cr.to_radians(a);
		b =
cr.to_radians(b);
		var diff =
cr.angleDiff(a, b);
```

```
            if
(cr.angleClockwise(b, a))
            {

      ret.set_float(cr.to_clampe
d_degrees(a + diff * x));
            }
            else
            {

      ret.set_float(cr.to_clampe
d_degrees(a - diff * x));
            }
      };
      SysExps.prototype.anglerot
ate = function (ret, a, b, c)
      {
            a =
cr.to_radians(a);
            b =
cr.to_radians(b);
            c =
cr.to_radians(c);

      ret.set_float(cr.to_clampe
d_degrees(cr.angleRotate(a, b,
c)));
      };
      SysExps.prototype.zeropad
= function (ret, n, d)
      {
            var s = (n < 0 ? "-"
: "");
            if (n < 0) n = -n;
            var zeroes = d -
n.toString().length;
            for (var i = 0; i <
zeroes; i++)
                  s += "0";
            ret.set_string(s +
n.toString());
      };
      SysExps.prototype.cpuutili
sation = function (ret)
      {

      ret.set_float(this.runtime
.cpuutilisation / 1000);
      };
      SysExps.prototype.viewport
left = function (ret,
layerparam)
      {
            var layer =
this.runtime.getLayer(layerpara
m);
            ret.set_float(layer
? layer.viewLeft : 0);
      };
      SysExps.prototype.viewport
top = function (ret,
layerparam)
      {
            var layer =
this.runtime.getLayer(layerpara
m);
            ret.set_float(layer
? layer.viewTop : 0);
      };
      SysExps.prototype.viewport
right = function (ret,
layerparam)
      {
            var layer =
this.runtime.getLayer(layerpara
m);
            ret.set_float(layer
? layer.viewRight : 0);
      };
      SysExps.prototype.viewport
bottom = function (ret,
layerparam)
      {
            var layer =
this.runtime.getLayer(layerpara
m);
            ret.set_float(layer
? layer.viewBottom : 0);
      };
      SysExps.prototype.loadingp
rogress = function (ret)
      {

      ret.set_float(this.runtime
.loadingprogress);
      };
      SysExps.prototype.unlerp =
function(ret, a, b, y)
    {
        ret.set_float((y - a) /
(b - a));
    };
      SysExps.prototype.canvassn
apshot = function (ret)
      {

      ret.set_string(this.runtim
e.snapshotData);
      };
      SysExps.prototype.urlencod
e = function (ret, s)
      {
```

```
	ret.set_string(encodeURICo
mponent(s));
	};
	SysExps.prototype.urldecod
e = function (ret, s)
	{

	ret.set_string(decodeURICo
mponent(s));
	};
	SysExps.prototype.canvasto
layerx = function (ret,
layerparam, x, y)
	{
		var layer =
this.runtime.getLayer(layerpara
m);
		ret.set_float(layer
? layer.canvasToLayer(x, y,
true) : 0);
	};
	SysExps.prototype.canvasto
layery = function (ret,
layerparam, x, y)
	{
		var layer =
this.runtime.getLayer(layerpara
m);
		ret.set_float(layer
? layer.canvasToLayer(x, y,
false) : 0);
	};
	SysExps.prototype.layertoc
anvasx = function (ret,
layerparam, x, y)
	{
		var layer =
this.runtime.getLayer(layerpara
m);
		ret.set_float(layer
? layer.layerToCanvas(x, y,
true) : 0);
	};
	SysExps.prototype.layertoc
anvasy = function (ret,
layerparam, x, y)
	{
		var layer =
this.runtime.getLayer(layerpara
m);
		ret.set_float(layer
? layer.layerToCanvas(x, y,
false) : 0);
	};
	SysExps.prototype.savestat
ejson = function (ret)
	{

	ret.set_string(this.runtim
e.lastSaveJson);
	};
	SysExps.prototype.imagemem
oryusage = function (ret)
	{
		if
(this.runtime.glwrap)

	ret.set_float(Math.round(1
00 *
this.runtime.glwrap.estimateVRA
M() / (1024 * 1024)) / 100);
		else

	ret.set_float(0);
	};
	SysExps.prototype.regexsea
rch = function (ret, str_,
regex_, flags_)
	{
		var regex =
getRegex(regex_, flags_);
		ret.set_int(str_ ?
str_.search(regex) : -1);
	};
	SysExps.prototype.regexrep
lace = function (ret, str_,
regex_, flags_, replace_)
	{
		var regex =
getRegex(regex_, flags_);
		ret.set_string(str_
? str_.replace(regex, replace_)
: "");
	};
	var regexMatches = [];
	var lastMatchesStr = "";
	var lastMatchesRegex = "";
	var lastMatchesFlags = "";
	function
updateRegexMatches(str_,
regex_, flags_)
	{
		if (str_ ===
lastMatchesStr && regex_ ===
lastMatchesRegex && flags_ ===
lastMatchesFlags)
			return;
		var regex =
getRegex(regex_, flags_);
```

```
		regexMatches =
str_.match(regex);
		lastMatchesStr =
str_;
		lastMatchesRegex =
regex_;
		lastMatchesFlags =
flags_;
	};
	SysExps.prototype.regexmat
chcount = function (ret, str_,
regex_, flags_)
	{
		var regex =
getRegex(regex_, flags_);

	updateRegexMatches(str_,
regex_, flags_);

	ret.set_int(regexMatches ?
regexMatches.length : 0);
	};
	SysExps.prototype.regexmat
chat = function (ret, str_,
regex_, flags_, index_)
	{
		index_ =
Math.floor(index_);
		var regex =
getRegex(regex_, flags_);

	updateRegexMatches(str_,
regex_, flags_);
		if (!regexMatches ||
index_ < 0 || index_ >=
regexMatches.length)

	ret.set_string("");
		else

	ret.set_string(regexMatche
s[index_]);
	};
	SysExps.prototype.infinity
= function (ret)
	{

	ret.set_float(Infinity);
	};
	sysProto.exps = new
SysExps();
	sysProto.runWaits =
function ()
	{
		var i, j, len, w, k,
s, ss;
		var evinfo =
this.runtime.getCurrentEventSta
ck();
		for (i = 0, len =
this.waits.length; i < len;
i++)
		{
			w =
this.waits[i];
			if (w.time >
this.runtime.kahanTime.sum)
				continue;

	evinfo.current_event =
w.ev;
			evinfo.actindex
= w.actindex;
			evinfo.cndindex
= 0;
			for (k in
w.sols)
			{
				if
(w.sols.hasOwnProperty(k))
				{
					s =
this.runtime.types_by_index[par
seInt(k, 10)].getCurrentSol();
					ss =
w.sols[k];

	s.select_all = ss.sa;

	cr.shallowAssignArray(s.in
stances, ss.insts);

	freeSolStateObject(ss);
				}
			}

	w.ev.resume_actions_and_su
bevents();

	this.runtime.clearSol(w.so
lModifiers);
			w.deleteme =
true;
		}
		for (i = 0, j = 0,
len = this.waits.length; i <
len; i++)
		{
			w =
this.waits[i];
			this.waits[j] =
w;
```

```
                if (w.deleteme)
     freeWaitObject(w);
                else
                     j++;
           }
           this.waits.length =
j;
     };
}());
;
(function () {
     cr.add_common_aces =
function (m)
     {
           var pluginProto =
m[0].prototype;
           var singleglobal_ =
m[1];
           var position_aces =
m[3];
           var size_aces =
m[4];
           var angle_aces =
m[5];
           var appearance_aces
= m[6];
           var zorder_aces =
m[7];
           var effects_aces =
m[8];
           if
(!pluginProto.cnds)

     pluginProto.cnds = {};
           if
(!pluginProto.acts)

     pluginProto.acts = {};
           if
(!pluginProto.exps)

     pluginProto.exps = {};
           var cnds =
pluginProto.cnds;
           var acts =
pluginProto.acts;
           var exps =
pluginProto.exps;
           if (position_aces)
           {
                cnds.CompareX =
function (cmp, x)
                {
                     return
cr.do_cmp(this.x, cmp, x);
                };
                cnds.CompareY =
function (cmp, y)
                {
                     return
cr.do_cmp(this.y, cmp, y);
                };
                cnds.IsOnScreen
= function ()
                {
                     var layer
= this.layer;

     this.update_bbox();
                     var bbox
= this.bbox;
                     return
!(bbox.right < layer.viewLeft
|| bbox.bottom < layer.viewTop
|| bbox.left > layer.viewRight
|| bbox.top >
layer.viewBottom);
                };

     cnds.IsOutsideLayout =
function ()
                {

     this.update_bbox();
                     var bbox
= this.bbox;
                     var
layout =
this.runtime.running_layout;
                     return
(bbox.right < 0 || bbox.bottom
< 0 || bbox.left > layout.width
|| bbox.top > layout.height);
                };

     cnds.PickDistance =
function (which, x, y)
                {
                     var sol =
this.getCurrentSol();
                     var
instances = sol.getObjects();
                     if
(!instances.length)

     return false;
                     var inst
= instances[0];
                     var
pickme = inst;
```

```
                    var dist
= cr.distanceTo(inst.x, inst.y,
x, y);
                    var i,
len, d;
                    for (i =
1, len = instances.length; i <
len; i++)
                    {
                         inst
= instances[i];
                         d =
cr.distanceTo(inst.x, inst.y,
x, y);
                         if
((which === 0 && d < dist) ||
(which === 1 && d > dist))
                         {

     dist = d;

     pickme = inst;
                         }
                    }

     sol.pick_one(pickme);
                    return
true;
               };
               acts.SetX =
function (x)
               {
                    if
(this.x !== x)
                    {

     this.x = x;

     this.set_bbox_changed();
                    }
               };
               acts.SetY =
function (y)
               {
                    if
(this.y !== y)
                    {

     this.y = y;

     this.set_bbox_changed();
                    }
               };
               acts.SetPos =
function (x, y)
               {
                    if
(this.x !== x || this.y !== y)
                    {

     this.x = x;

     this.y = y;

     this.set_bbox_changed();
                    }
               };

     acts.SetPosToObject =
function (obj, imgpt)
               {
                    var inst
= obj.getPairedInstance(this);
                    if
(!inst)

     return;
                    var newx,
newy;
                    if
(inst.getImagePoint)
                    {
                         newx
= inst.getImagePoint(imgpt,
true);
                         newy
= inst.getImagePoint(imgpt,
false);
                    }
                    else
                    {
                         newx
= inst.x;
                         newy
= inst.y;
                    }
                    if
(this.x !== newx || this.y !==
newy)
                    {

     this.x = newx;

     this.y = newy;

     this.set_bbox_changed();
                    }
               };

     acts.MoveForward =
function (dist)
               {
```

```
                    if (dist
!== 0)
                    {

     this.x +=
Math.cos(this.angle) * dist;

     this.y +=
Math.sin(this.angle) * dist;

     this.set_bbox_changed();
                    }
                };

     acts.MoveAtAngle =
function (a, dist)
                {
                    if (dist
!== 0)
                    {

     this.x +=
Math.cos(cr.to_radians(a)) *
dist;

     this.y +=
Math.sin(cr.to_radians(a)) *
dist;

     this.set_bbox_changed();
                    }
                };
                exps.X =
function (ret)
                {

     ret.set_float(this.x);
                };
                exps.Y =
function (ret)
                {

     ret.set_float(this.y);
                };
                exps.dt =
function (ret)
                {

     ret.set_float(this.runtime
.getDt(this));
                };
            }
            if (size_aces)
            {

     cnds.CompareWidth =
function (cmp, w)
                {
                    return
cr.do_cmp(this.width, cmp, w);
                };

     cnds.CompareHeight =
function (cmp, h)
                {
                    return
cr.do_cmp(this.height, cmp, h);
                };
                acts.SetWidth =
function (w)
                {
                    if
(this.width !== w)
                    {

     this.width = w;

     this.set_bbox_changed();
                    }
                };
                acts.SetHeight
= function (h)
                {
                    if
(this.height !== h)
                    {

     this.height = h;

     this.set_bbox_changed();
                    }
                };
                acts.SetSize =
function (w, h)
                {
                    if
(this.width !== w ||
this.height !== h)
                    {

     this.width = w;

     this.height = h;

     this.set_bbox_changed();
                    }
                };
                exps.Width =
function (ret)
                {
```

```
ret.set_float(this.width);
                };
                exps.Height =
function (ret)
                {

ret.set_float(this.height)
;
                };
                exps.BBoxLeft =
function (ret)
                {

this.update_bbox();

ret.set_float(this.bbox.le
ft);
                };
                exps.BBoxTop =
function (ret)
                {

this.update_bbox();

ret.set_float(this.bbox.to
p);
                };
                exps.BBoxRight
= function (ret)
                {

this.update_bbox();

ret.set_float(this.bbox.ri
ght);
                };
                exps.BBoxBottom
= function (ret)
                {

this.update_bbox();

ret.set_float(this.bbox.bo
ttom);
                };
            }
            if (angle_aces)
            {

cnds.AngleWithin =
function (within, a)
                {
                      return
cr.angleDiff(this.angle,
cr.to_radians(a)) <=
cr.to_radians(within);
                };

cnds.IsClockwiseFrom =
function (a)
                {
                      return
cr.angleClockwise(this.angle,
cr.to_radians(a));
                };

cnds.IsBetweenAngles =
function (a, b)
                {
                      var lower
= cr.to_clamped_radians(a);
                      var upper
= cr.to_clamped_radians(b);
                      var angle
= cr.clamp_angle(this.angle);
                      var
obtuse =
(!cr.angleClockwise(upper,
lower));
                      if
(obtuse)

return
!(!cr.angleClockwise(angle,
lower) &&
cr.angleClockwise(angle,
upper));
                      else

return
cr.angleClockwise(angle, lower)
&& !cr.angleClockwise(angle,
upper);
                };
                acts.SetAngle =
function (a)
                {
                      var
newangle =
cr.to_radians(cr.clamp_angle_de
grees(a));
                      if
(isNaN(newangle))

return;
                      if
(this.angle !== newangle)
                      {

this.angle = newangle;
```

```
this.set_bbox_changed();
                }
            };

acts.RotateClockwise =
function (a)
            {
                if (a !==
0 && !isNaN(a))
                {

this.angle +=
cr.to_radians(a);

this.angle =
cr.clamp_angle(this.angle);

this.set_bbox_changed();
                }
            };

acts.RotateCounterclockwis
e = function (a)
            {
                if (a !==
0 && !isNaN(a))
                {

this.angle -=
cr.to_radians(a);

this.angle =
cr.clamp_angle(this.angle);

this.set_bbox_changed();
                }
            };

acts.RotateTowardAngle =
function (amt, target)
            {
                var
newangle =
cr.angleRotate(this.angle,
cr.to_radians(target),
cr.to_radians(amt));
                if
(isNaN(newangle))

return;
                if
(this.angle !== newangle)
                {

this.angle = newangle;

this.set_bbox_changed();
                }
            };

acts.RotateTowardPosition
= function (amt, x, y)
            {
                var dx =
x - this.x;
                var dy =
y - this.y;
                var
target = Math.atan2(dy, dx);
                var
newangle =
cr.angleRotate(this.angle,
target, cr.to_radians(amt));
                if
(isNaN(newangle))

return;
                if
(this.angle !== newangle)
                {

this.angle = newangle;

this.set_bbox_changed();
                }
            };

acts.SetTowardPosition =
function (x, y)
            {
                var dx =
x - this.x;
                var dy =
y - this.y;
                var
newangle = Math.atan2(dy, dx);
                if
(isNaN(newangle))

return;
                if
(this.angle !== newangle)
                {

this.angle = newangle;

this.set_bbox_changed();
                }
            };
            exps.Angle =
function (ret)
```

```
                {
     ret.set_float(cr.to_clampe
d_degrees(this.angle));
                };
           }
           if (!singleglobal_)
           {

     cnds.CompareInstanceVar =
function (iv, cmp, val)
                {
                     return
cr.do_cmp(this.instance_vars[iv
], cmp, val);
                };

     cnds.IsBoolInstanceVarSet
= function (iv)
                {
                     return
this.instance_vars[iv];
                };

     cnds.PickInstVarHiLow =
function (which, iv)
                {
                     var sol =
this.getCurrentSol();
                     var
instances = sol.getObjects();
                     if
(!instances.length)

     return false;
                     var inst
= instances[0];
                     var
pickme = inst;
                     var val =
inst.instance_vars[iv];
                     var i,
len, v;
                     for (i =
1, len = instances.length; i <
len; i++)
                     {
                          inst
= instances[i];
                          v =
inst.instance_vars[iv];
                          if
((which === 0 && v < val) ||
(which === 1 && v > val))
                          {
                               val = v;

                               pickme = inst;
                          }
                     }

                     sol.pick_one(pickme);
                     return
true;
                };
                cnds.PickByUID
= function (u)
                {
                     var i,
len, j, inst, families,
instances, sol;
                     var cnd =
this.runtime.getCurrentConditio
n();
                     if
(cnd.inverted)
                     {
                          sol
= this.getCurrentSol();
                          if
(sol.select_all)
                          {

                               sol.select_all = false;

                               sol.instances.length = 0;

                               sol.else_instances.length
= 0;

                               instances =
this.instances;

                               for (i = 0, len =
instances.length; i < len; i++)

                               {

                                    inst = instances[i];

                                    if (inst.uid === u)

                               sol.else_instances.push(in
st);

                                    else

                               sol.instances.push(inst);
```

```
}
this.applySolToContainer();
return !!sol.instances.length;
}
else
{
for (i = 0, j = 0, len = sol.instances.length; i < len; i++)
{
inst = sol.instances[i];
sol.instances[j] = inst;
if (inst.uid === u)
{
sol.else_instances.push(inst);
}
else
j++;
}
sol.instances.length = j;
this.applySolToContainer();
return !!sol.instances.length;
}
}
else
{
inst = this.runtime.getObjectByUID(u);
if (!inst)
return false;
sol = this.getCurrentSol();
if (!sol.select_all && sol.instances.indexOf(inst) === -1)
return false;
// not picked
if (this.is_family)
{
families = inst.type.families;
for (i = 0, len = families.length; i < len; i++)
{
if (families[i] === this)
{
sol.pick_one(inst);
this.applySolToContainer();
return true;
}
}
}
else if (inst.type === this)
{
sol.pick_one(inst);
this.applySolToContainer();
return true;
}
return false;
}
};
```

```
                cnds.OnCreated
= function ()
                {
                     return
true;
                };

     cnds.OnDestroyed =
function ()
                {
                     return
true;
                };

     acts.SetInstanceVar =
function (iv, val)
                {
                     var
myinstvars =
this.instance_vars;
                     if
(cr.is_number(myinstvars[iv]))
                     {
                          if
(cr.is_number(val))

     myinstvars[iv] = val;
                          else

     myinstvars[iv] =
parseFloat(val);
                     }
                     else if
(cr.is_string(myinstvars[iv]))
                     {
                          if
(cr.is_string(val))

     myinstvars[iv] = val;
                          else

     myinstvars[iv] =
val.toString();
                     }
                     else
;
                };

     acts.AddInstanceVar =
function (iv, val)
                {
                     var
myinstvars =
this.instance_vars;
                     if
(cr.is_number(myinstvars[iv]))
                     {
                          if
(cr.is_number(val))

     myinstvars[iv] += val;
                          else

     myinstvars[iv] +=
parseFloat(val);
                     }
                     else if
(cr.is_string(myinstvars[iv]))
                     {
                          if
(cr.is_string(val))

     myinstvars[iv] += val;
                          else

     myinstvars[iv] +=
val.toString();
                     }
                     else
;
                };

     acts.SubInstanceVar =
function (iv, val)
                {
                     var
myinstvars =
this.instance_vars;
                     if
(cr.is_number(myinstvars[iv]))
                     {
                          if
(cr.is_number(val))

     myinstvars[iv] -= val;
                          else

     myinstvars[iv] -=
parseFloat(val);
                     }
                     else
;
                };

     acts.SetBoolInstanceVar =
function (iv, val)
                {

     this.instance_vars[iv] =
val ? 1 : 0;
                };
```

```
acts.ToggleBoolInstanceVar = function (iv)
{
this.instance_vars[iv] = 1 - this.instance_vars[iv];
};
acts.Destroy = function ()
{
this.runtime.DestroyInstance(this);
};
if (!acts.LoadFromJsonString)
{
acts.LoadFromJsonString = function (str_)
{
var o, i, len, binst;
try {
o = JSON.parse(str_);
}
catch (e) {
return;
}
this.runtime.loadInstanceFromJSON(this, o, true);
if (this.afterLoad)
this.afterLoad();
if (this.behavior_insts)
{
for (i = 0, len = this.behavior_insts.length; i < len; ++i)
{
binst = this.behavior_insts[i];
if (binst.afterLoad)
binst.afterLoad();
}
}
};
}
exps.Count = function (ret)
{
var count = ret.object_class.instances.length;
var i, len, inst;
for (i = 0, len = this.runtime.createRow.length; i < len; i++)
{
inst = this.runtime.createRow[i];
if (ret.object_class.is_family)
{
if (inst.type.families.indexOf(ret.object_class) >= 0)
count++;
}
else
{
if (inst.type === ret.object_class)
count++;
}
}
ret.set_int(count);
};
exps.PickedCount = function (ret)
{
ret.set_int(ret.object_class.getCurrentSol().getObjects().length);
};
```

```
                exps.UID =
function (ret)
                {

     ret.set_int(this.uid);
                };
                exps.IID =
function (ret)
                {

     ret.set_int(this.get_iid()
);
                };
                if
(!exps.AsJSON)
                {

     exps.AsJSON = function
(ret)
                     {

     ret.set_string(JSON.string
ify(this.runtime.saveInstanceTo
JSON(this, true)));
                     };
                }
           }
           if (appearance_aces)
           {
                cnds.IsVisible
= function ()
                {
                     return
this.visible;
                };
                acts.SetVisible
= function (v)
                {
                     if (!v
!== !this.visible)
                     {

     this.visible = v;

     this.runtime.redraw =
true;
                     }
                };

     cnds.CompareOpacity =
function (cmp, x)
                {
                     return
cr.do_cmp(cr.round6dp(this.opac
ity * 100), cmp, x);
                };
                acts.SetOpacity
= function (x)
                {
                     var
new_opacity = x / 100.0;
                     if
(new_opacity < 0)

     new_opacity = 0;
                     else if
(new_opacity > 1)

     new_opacity = 1;
                     if
(new_opacity !== this.opacity)
                     {

     this.opacity =
new_opacity;

     this.runtime.redraw =
true;
                     }
                };
                exps.Opacity =
function (ret)
                {

     ret.set_float(cr.round6dp(
this.opacity * 100.0));
                };
           }
           if (zorder_aces)
           {
                cnds.IsOnLayer
= function (layer_)
                {
                     if
(!layer_)

     return false;
                     return
this.layer === layer_;
                };

     cnds.PickTopBottom =
function (which_)
                {
                     var sol =
this.getCurrentSol();
                     var
instances = sol.getObjects();
                     if
(!instances.length)

     return false;
```

```
                    var inst
= instances[0];
                    var
pickme = inst;
                    var i,
len;
                    for (i =
1, len = instances.length; i <
len; i++)
                    {
                         inst
= instances[i];
                         if
(which_ === 0)
                         {

     if (inst.layer.index >
pickme.layer.index ||
(inst.layer.index ===
pickme.layer.index &&
inst.get_zindex() >
pickme.get_zindex()))

     {

          pickme = inst;

     }
                         }
                         else
                         {

     if (inst.layer.index <
pickme.layer.index ||
(inst.layer.index ===
pickme.layer.index &&
inst.get_zindex() <
pickme.get_zindex()))

     {

          pickme = inst;

     }
                         }
                    }

     sol.pick_one(pickme);
                    return
true;
               };
               acts.MoveToTop
= function ()
               {
                    var
zindex = this.get_zindex();
                         if
(zindex ===
this.layer.instances.length -
1)

     return;

     cr.arrayRemove(this.layer.
instances, zindex);

     this.layer.instances.push(
this);

     this.runtime.redraw =
true;

     this.layer.zindices_stale
= true;
                    };

     acts.MoveToBottom =
function ()
                    {
                         var
zindex = this.get_zindex();
                         if
(zindex === 0)

     return;

     cr.arrayRemove(this.layer.
instances, zindex);

     this.layer.instances.unshi
ft(this);

     this.runtime.redraw =
true;

     this.layer.zindices_stale
= true;
                    };

     acts.MoveToLayer =
function (layerMove)
                    {
                         if
(!layerMove || layerMove ==
this.layer)

     return;

     cr.arrayRemove(this.layer.
instances, this.get_zindex());
```

```
this.layer.zindices_stale = true;

this.layer = layerMove;

this.zindex = layerMove.instances.length;

layerMove.instances.push(this);

this.runtime.redraw = true;
};

acts.ZMoveToObject = function (where_, obj_)
{
var isafter = (where_ === 0);
if (!obj_)

return;
var other = obj_.getFirstPicked(this);
if (!other || other.uid === this.uid)

return;
if (this.layer.index !== other.layer.index)
{

cr.arrayRemove(this.layer.instances, this.get_zindex());

this.layer.zindices_stale = true;

this.layer = other.layer;

this.zindex = other.layer.instances.length;

other.layer.instances.push(this);
}
var myZ = this.get_zindex();
var insertZ = other.get_zindex();

cr.arrayRemove(this.layer.instances, myZ);
if (myZ < insertZ)

insertZ--;
if (isafter)

insertZ++;
if (insertZ === this.layer.instances.length)

this.layer.instances.push(this);
else

this.layer.instances.splice(insertZ, 0, this);

this.layer.zindices_stale = true;

this.runtime.redraw = true;
};

exps.LayerNumber = function (ret)
{

ret.set_int(this.layer.number);
};
exps.LayerName = function (ret)
{

ret.set_string(this.layer.name);
};
exps.ZIndex = function (ret)
{

ret.set_int(this.get_zindex());
};
}
if (effects_aces)
{

acts.SetEffectEnabled = function (enable_, effectname_)
```

```
                {
                        if
(!this.runtime.glwrap)

     return;
                        var i =
this.type.getEffectIndexByName(
effectname_);
                        if (i <
0)

     return;            // effect
name not found
                        var
enable = (enable_ === 1);
                        if
(this.active_effect_flags[i]
=== enable)

     return;            // no
change

     this.active_effect_flags[i
] = enable;

     this.updateActiveEffects()
;

     this.runtime.redraw =
true;
                };

     acts.SetEffectParam =
function (effectname_, index_,
value_)
                {
                        if
(!this.runtime.glwrap)

     return;
                        var i =
this.type.getEffectIndexByName(
effectname_);
                        if (i <
0)

     return;            // effect
name not found
                        var et =
this.type.effect_types[i];
                        var
params = this.effect_params[i];
                        index_ =
Math.floor(index_);
                        if
(index_ < 0 || index_ >=
params.length)

     return;            // effect
index out of bounds
                        if
(this.runtime.glwrap.getProgram
ParameterType(et.shaderindex,
index_) === 1)

     value_ /= 100.0;
                        if
(params[index_] === value_)

     return;            // no
change

     params[index_] = value_;
                        if
(et.active)

     this.runtime.redraw =
true;
                }
            }
     };
     cr.set_bbox_changed =
function ()
     {
            this.bbox_changed =
true;                   // will
recreate next time box
requested

     this.type.any_bbox_changed
= true;    // avoid unnecessary
updateAllBBox() calls
            this.runtime.redraw
= true;          // assume
runtime needs to redraw
            var i, len,
callbacks =
this.bbox_changed_callbacks;
            for (i = 0, len =
callbacks.length; i < len; ++i)
            {

     callbacks[i](this);
            }
     };
     cr.add_bbox_changed_callba
ck = function (f)
     {
            if (f)
            {
```

```
	this.bbox_changed_callback
s.push(f);
		}
	};
	var tmprc = new cr.rect(0,
0, 0, 0);
	cr.update_bbox = function
()
	{
		if
(!this.bbox_changed)
			return;
// bounding box not changed
		var bbox =
this.bbox;
		var bquad =
this.bquad;
		bbox.set(this.x,
this.y, this.x + this.width,
this.y + this.height);
		bbox.offset(-
this.hotspotX * this.width, -
this.hotspotY * this.height);
		if (!this.angle)
		{

	bquad.set_from_rect(bbox);
// make bounding quad from box
		}
		else
		{
			bbox.offset(-
this.x, -this.y);
		// translate to
origin

	bquad.set_from_rotated_rec
t(bbox, this.angle);  // rotate
around origin

	bquad.offset(this.x,
this.y);
	// translate back to
original position

	bquad.bounding_box(bbox);
		}
		bbox.normalize();
		this.bbox_changed =
false;  // bounding box up to
date
		if
(this.collisionsEnabled)
		{
```

```
		var mygrid =
this.type.collision_grid;
		var collcells =
this.collcells;

	tmprc.set(mygrid.XToCell(b
box.left),
mygrid.YToCell(bbox.top),
mygrid.XToCell(bbox.right),
mygrid.YToCell(bbox.bottom));
		if
(!collcells.equals(tmprc))
		{
			if
(collcells.right <
collcells.left)

	mygrid.update(this, null,
tmprc);		// first
insertion with invalid rect:
don't provide old range
			else

	mygrid.update(this,
collcells, tmprc);

	collcells.copy(tmprc);
		}
		}
	};
	cr.inst_contains_pt =
function (x, y)
	{
		if
(!this.bbox.contains_pt(x, y))
		return false;
		if
(!this.bquad.contains_pt(x, y))
		return false;
		if
(this.collision_poly &&
!this.collision_poly.is_empty()
)
		{

	this.collision_poly.cache_
poly(this.width, this.height,
this.angle);
		return
this.collision_poly.contains_pt
(x - this.x, y - this.y);
		}
		else
		return true;
	};
```

```
	cr.inst_get_iid = function
()
	{

	this.type.updateIIDs();
		return this.iid;
	};
	cr.inst_get_zindex =
function ()
	{

	this.layer.updateZIndices(
);
		return this.zindex;
	};
	cr.inst_updateActiveEffect
s = function ()
	{

	this.active_effect_types.l
ength = 0;
		var i, len, et,
inst;
		for (i = 0, len =
this.active_effect_flags.length
; i < len; i++)
		{
			if
(this.active_effect_flags[i])

	this.active_effect_types.p
ush(this.type.effect_types[i]);
		}
		this.uses_shaders =
!!this.active_effect_types.leng
th;
	};
	cr.inst_toString =
function ()
	{
		return "Inst" +
this.puid;
	};
	cr.type_getFirstPicked =
function (frominst)
	{
		if (frominst &&
frominst.is_contained &&
frominst.type != this)
		{
			var i, len, s;
			for (i = 0, len
= frominst.siblings.length; i <
len; i++)
			{
				s =
frominst.siblings[i];
				if
(s.type == this)

	return s;
			}
		}
		var instances =
this.getCurrentSol().getObjects
();
		if
(instances.length)
			return
instances[0];
		else
			return null;
	};
	cr.type_getPairedInstance
= function (inst)
	{
		var instances =
this.getCurrentSol().getObjects
();
		if
(instances.length)
			return
instances[inst.get_iid() %
instances.length];
		else
			return null;
	};
	cr.type_updateIIDs =
function ()
	{
		if (!this.stale_iids
|| this.is_family)
			return;
	// up to date or is family
- don't want family to
overwrite IIDs
		var i, len;
		for (i = 0, len =
this.instances.length; i < len;
i++)

	this.instances[i].iid = i;
		var next_iid = i;
		var createRow =
this.runtime.createRow;
		for (i = 0, len =
createRow.length; i < len; ++i)
		{
			if
(createRow[i].type === this)
```

```
	createRow[i].iid =
next_iid++;
		}
		this.stale_iids =
false;
	};
	cr.type_getInstanceByIID =
function (i)
	{
		if (i <
this.instances.length)
			return
this.instances[i];
		i -=
this.instances.length;
		var createRow =
this.runtime.createRow;
		var j, lenj;
		for (j = 0, lenj =
createRow.length; j < lenj;
++j)
		{
			if
(createRow[j].type === this)
			{
				if (i ===
0)

	return createRow[j];
				--i;
			}
		}
;
		return null;
	};
	cr.type_getCurrentSol =
function ()
	{
		return
this.solstack[this.cur_sol];
	};
	cr.type_pushCleanSol =
function ()
	{
		this.cur_sol++;
		if (this.cur_sol ===
this.solstack.length)

	this.solstack.push(new
cr.selection(this));
		else

	this.solstack[this.cur_sol
].select_all = true;  // else
clear next SOL
	};
	cr.type_pushCopySol =
function ()
	{
		this.cur_sol++;
		if (this.cur_sol ===
this.solstack.length)

	this.solstack.push(new
cr.selection(this));
		var clonesol =
this.solstack[this.cur_sol];
		var prevsol =
this.solstack[this.cur_sol -
1];
		if
(prevsol.select_all)

	clonesol.select_all =
true;
		else
		{

	clonesol.select_all =
false;

	cr.shallowAssignArray(clon
esol.instances,
prevsol.instances);

	cr.shallowAssignArray(clon
esol.else_instances,
prevsol.else_instances);
		}
	};
	cr.type_popSol = function
()
	{
;
		this.cur_sol--;
	};
	cr.type_getBehaviorByName
= function (behname)
	{
		var i, len, j, lenj,
f, index = 0;
		if (!this.is_family)
		{
			for (i = 0, len
= this.families.length; i <
len; i++)
			{
				f =
this.families[i];
```

```
                        for (j =
0, lenj = f.behaviors.length; j
< lenj; j++)
                        {
                              if
(behname ===
f.behaviors[j].name)
                              {

      this.extra.lastBehIndex =
index;

      return f.behaviors[j];
                              }

      index++;
                        }
                  }
            }
            for (i = 0, len =
this.behaviors.length; i < len;
i++) {
                  if (behname ===
this.behaviors[i].name)
                  {

      this.extra.lastBehIndex =
index;
                        return
this.behaviors[i];
                  }
                  index++;
            }
            return null;
      };
      cr.type_getBehaviorIndexBy
Name = function (behname)
      {
            var b =
this.getBehaviorByName(behname)
;
            if (b)
                  return
this.extra.lastBehIndex;
            else
                  return -1;
      };
      cr.type_getEffectIndexByNa
me = function (name_)
      {
            var i, len;
            for (i = 0, len =
this.effect_types.length; i <
len; i++)
            {
                  if
(this.effect_types[i].name ===
name_)
                        return i;
            }
            return -1;
      };
      cr.type_applySolToContaine
r = function ()
      {
            if
(!this.is_contained ||
this.is_family)
                  return;
            var i, len, j, lenj,
t, sol, sol2;
            this.updateIIDs();
            sol =
this.getCurrentSol();
            var select_all =
sol.select_all;
            var es =
this.runtime.getCurrentEventSta
ck();
            var orblock = es &&
es.current_event &&
es.current_event.orblock;
            for (i = 0, len =
this.container.length; i < len;
i++)
            {
                  t =
this.container[i];
                  if (t === this)
                        continue;
                  t.updateIIDs();
                  sol2 =
t.getCurrentSol();
                  sol2.select_all
= select_all;
                  if
(!select_all)
                  {

      sol2.instances.length =
sol.instances.length;
                        for (j =
0, lenj = sol.instances.length;
j < lenj; j++)

      sol2.instances[j] =
t.getInstanceByIID(sol.instance
s[j].iid);
                        if
(orblock)
                        {
```

```
	sol2.else_instances.length = sol.else_instances.length;
	for (j = 0, lenj = sol.else_instances.length; j < lenj; j++)

	sol2.else_instances[j] = t.getInstanceByIID(sol.else_instances[j].iid);
	}
	}
	}
	};
	cr.type_toString = function ()
	{
	return "Type" + this.sid;
	};
	cr.do_cmp = function (x, cmp, y)
	{
	if (typeof x === "undefined" || typeof y === "undefined")
	return false;
	switch (cmp)
	{
	case 0:	// equal
	return x === y;
	case 1:	// not equal
	return x !== y;
	case 2:	// less
	return x < y;
	case 3:	// less/equal
	return x <= y;
	case 4:	// greater
	return x > y;
	case 5:	// greater/equal
	return x >= y;
	default:
;
	return false;
	}
	};
})();
cr.shaders = {};
;
;
cr.plugins_.Audio = function(runtime)
{
	this.runtime = runtime;
};
(function ()
{
	var pluginProto = cr.plugins_.Audio.prototype;
	pluginProto.Type = function(plugin)
	{
	this.plugin = plugin;
	this.runtime = plugin.runtime;
	};
	var typeProto = pluginProto.Type.prototype;
	typeProto.onCreate = function()
	{
	};
	var audRuntime = null;
	var audInst = null;
	var audTag = "";
	var appPath = "";	// for PhoneGap only
	var API_HTML5 = 0;
	var API_WEBAUDIO = 1;
	var API_PHONEGAP = 2;
	var API_APPMOBI = 3;
	var api = API_HTML5;
	var context = null;
	var audioBuffers = [];	// cache of buffers
	var audioInstances = [];	// cache of instances
	var lastAudio = null;
	var useOgg = false;	// determined at create time
	var timescale_mode = 0;
	var silent = false;
	var masterVolume = 1;
	var listenerX = 0;
	var listenerY = 0;
```

```
	var panningModel = 1;
	// HRTF
	var distanceModel = 1;
		// Inverse
	var refDistance = 10;
	var maxDistance = 10000;
	var rolloffFactor = 1;
	var micSource = null;
	var micTag = "";
	var isMusicWorkaround =
false;
	var musicPlayNextTouch =
[];
	function dbToLinear(x)
	{
		var v =
dbToLinear_nocap(x);
		if (v < 0)
			v = 0;
		if (v > 1)
			v = 1;
		return v;
	};
	function linearToDb(x)
	{
		if (x < 0)
			x = 0;
		if (x > 1)
			x = 1;
		return
linearToDb_nocap(x);
	};
	function
dbToLinear_nocap(x)
	{
		return Math.pow(10,
x / 20);
	};
	function
linearToDb_nocap(x)
	{
		return (Math.log(x)
/ Math.log(10)) * 20;
	};
	var effects = {};
	function
getDestinationForTag(tag)
	{
		tag =
tag.toLowerCase();
		if
(effects.hasOwnProperty(tag))
		{
			if
(effects[tag].length)
				return
effects[tag][0].getInputNode();
		}
		return
context["destination"];
	};
	function createGain()
	{
		if
(context["createGain"])
			return
context["createGain"]();
		else
			return
context["createGainNode"]();
	};
	function createDelay(d)
	{
		if
(context["createDelay"])
			return
context["createDelay"](d);
		else
			return
context["createDelayNode"](d);
	};
	function startSource(s)
	{
		if (s["start"])
			s["start"](0);
		else
			s["noteOn"](0);
	};
	function startSourceAt(s,
x, d)
	{
		if (s["start"])
			s["start"](0,
x);
		else

	s["noteGrainOn"](0, x, d -
x);
	};
	function stopSource(s)
	{
		try {
			if (s["stop"])

	s["stop"](0);
			else

	s["noteOff"](0);
		}
		catch (e) {}
	};
```

```
	function setAudioParam(ap, value, ramp, time)
	{
		if (!ap)
			return;
	// iOS is missing some parameters

	ap["cancelScheduledValues"](0);
		if (time === 0)
		{
			ap["value"] = value;
			return;
		}
		var curTime = context["currentTime"];
		time += curTime;
		switch (ramp) {
		case 0:		// step

	ap["setValueAtTime"](value, time);
			break;
		case 1:		// linear

	ap["setValueAtTime"](ap["value"], curTime);		// to set what to ramp from

	ap["linearRampToValueAtTime"](value, time);
			break;
		case 2:		// exponential

	ap["setValueAtTime"](ap["value"], curTime);		// to set what to ramp from

	ap["exponentialRampToValueAtTime"](value, time);
			break;
		}
	};
	var filterTypes = ["lowpass", "highpass", "bandpass", "lowshelf", "highshelf", "peaking", "notch", "allpass"];
	function FilterEffect(type, freq, detune, q, gain, mix)
	{
		this.type = "filter";
		this.params = [type, freq, detune, q, gain, mix];
		this.inputNode = createGain();
		this.wetNode = createGain();

	this.wetNode["gain"]["value"] = mix;
		this.dryNode = createGain();

	this.dryNode["gain"]["value"] = 1 - mix;
		this.filterNode = context["createBiquadFilter"]();
		if (typeof this.filterNode["type"] === "number")

	this.filterNode["type"] = type;
		else

	this.filterNode["type"] = filterTypes[type];

	this.filterNode["frequency"]["value"] = freq;
		if (this.filterNode["detune"])
	// iOS 6 doesn't have detune yet

	this.filterNode["detune"]["value"] = detune;

	this.filterNode["Q"]["value"] = q;

	this.filterNode["gain"]["value"] = gain;

	this.inputNode["connect"](this.filterNode);

	this.inputNode["connect"](this.dryNode);

	this.filterNode["connect"](this.wetNode);
	};
```

```
	FilterEffect.prototype.connectTo = function (node)
	{
	this.wetNode["disconnect"]();
	this.wetNode["connect"](node);
	this.dryNode["disconnect"]();
	this.dryNode["connect"](node);
	};
	FilterEffect.prototype.remove = function ()
	{
	this.inputNode["disconnect"]();
	this.filterNode["disconnect"]();
	this.wetNode["disconnect"]();
	this.dryNode["disconnect"]();
	};
	FilterEffect.prototype.getInputNode = function ()
	{
		return this.inputNode;
	};
	FilterEffect.prototype.setParam = function(param, value, ramp, time)
	{
		switch (param) {
		case 0:		// mix
			value = value / 100;
			if (value < 0) value = 0;
			if (value > 1) value = 1;
			this.params[5] = value;

	setAudioParam(this.wetNode["gain"], value, ramp, time);

	setAudioParam(this.dryNode["gain"], 1 - value, ramp, time);
			break;
		case 1:		// filter frequency
			this.params[1] = value;

	setAudioParam(this.filterNode["frequency"], value, ramp, time);
			break;
		case 2:		// filter detune
			this.params[2] = value;

	setAudioParam(this.filterNode["detune"], value, ramp, time);
			break;
		case 3:		// filter Q
			this.params[3] = value;

	setAudioParam(this.filterNode["Q"], value, ramp, time);
			break;
		case 4:		// filter/delay gain (note value is in dB here)
			this.params[4] = value;

	setAudioParam(this.filterNode["gain"], value, ramp, time);
			break;
		}
	};
	function DelayEffect(delayTime, delayGain, mix)
	{
		this.type = "delay";
		this.params = [delayTime, delayGain, mix];
		this.inputNode = createGain();
		this.wetNode = createGain();
```

```
	this.wetNode["gain"]["value"] = mix;
		this.dryNode = createGain();

	this.dryNode["gain"]["value"] = 1 - mix;
		this.mainNode = createGain();
		this.delayNode = createDelay(delayTime);

	this.delayNode["delayTime"]["value"] = delayTime;
		this.delayGainNode = createGain();

	this.delayGainNode["gain"]["value"] = delayGain;

	this.inputNode["connect"](this.mainNode);

	this.inputNode["connect"](this.dryNode);

	this.mainNode["connect"](this.wetNode);

	this.mainNode["connect"](this.delayNode);

	this.delayNode["connect"](this.delayGainNode);

	this.delayGainNode["connect"](this.mainNode);
	};
	DelayEffect.prototype.connectTo = function (node)
	{

	this.wetNode["disconnect"]();

	this.wetNode["connect"](node);

	this.dryNode["disconnect"]();

	this.dryNode["connect"](node);
	};

	DelayEffect.prototype.remove = function ()
	{

	this.inputNode["disconnect"]();

	this.mainNode["disconnect"]();

	this.delayNode["disconnect"]();

	this.delayGainNode["disconnect"]();

	this.wetNode["disconnect"]();

	this.dryNode["disconnect"]();
	};
	DelayEffect.prototype.getInputNode = function ()
	{
		return this.inputNode;
	};
	DelayEffect.prototype.setParam = function(param, value, ramp, time)
	{
		switch (param) {
		case 0:		// mix
			value = value / 100;
			if (value < 0) value = 0;
			if (value > 1) value = 1;
			this.params[2] = value;

	setAudioParam(this.wetNode["gain"], value, ramp, time);

	setAudioParam(this.dryNode["gain"], 1 - value, ramp, time);
			break;
		case 4:		// filter/delay gain (note value is passed in dB but needs to be linear here)
```

```
                this.params[1]
= dbToLinear(value);

     setAudioParam(this.delayGa
inNode["gain"],
dbToLinear(value), ramp, time);
                break;
           case 5:         //
delay time
                this.params[0]
= value;

     setAudioParam(this.delayNo
de["delayTime"], value, ramp,
time);
                break;
           }
     };
     function
ConvolveEffect(buffer,
normalize, mix, src)
     {
           this.type =
"convolve";
           this.params =
[normalize, mix, src];
           this.inputNode =
createGain();
           this.wetNode =
createGain();

     this.wetNode["gain"]["valu
e"] = mix;
           this.dryNode =
createGain();

     this.dryNode["gain"]["valu
e"] = 1 - mix;
           this.convolveNode =
context["createConvolver"]();
           if (buffer)
           {

     this.convolveNode["normali
ze"] = normalize;

     this.convolveNode["buffer"
] = buffer;
           }

     this.inputNode["connect"](
this.convolveNode);

     this.inputNode["connect"](
this.dryNode);

     this.convolveNode["connect
"](this.wetNode);
     };
     ConvolveEffect.prototype.c
onnectTo = function (node)
     {

     this.wetNode["disconnect"]
();

     this.wetNode["connect"](no
de);

     this.dryNode["disconnect"]
();

     this.dryNode["connect"](no
de);
     };
     ConvolveEffect.prototype.r
emove = function ()
     {

     this.inputNode["disconnect
"]();

     this.convolveNode["disconn
ect"]();

     this.wetNode["disconnect"]
();

     this.dryNode["disconnect"]
();
     };
     ConvolveEffect.prototype.g
etInputNode = function ()
     {
          return
this.inputNode;
     };
     ConvolveEffect.prototype.s
etParam = function(param,
value, ramp, time)
     {
          switch (param) {
          case 0:         //
mix
               value = value /
100;
               if (value < 0)
value = 0;
               if (value > 1)
value = 1;
```

```
                this.params[1]
= value;

      setAudioParam(this.wetNode
["gain"], value, ramp, time);

      setAudioParam(this.dryNode
["gain"], 1 - value, ramp,
time);
                break;
            }
      };
      function
FlangerEffect(delay,
modulation, freq, feedback,
mix)
      {
            this.type =
"flanger";
            this.params =
[delay, modulation, freq,
feedback, mix];
            this.inputNode =
createGain();
            this.dryNode =
createGain();

      this.dryNode["gain"]["valu
e"] = 1 - (mix / 2);
            this.wetNode =
createGain();

      this.wetNode["gain"]["valu
e"] = mix / 2;
            this.feedbackNode =
createGain();

      this.feedbackNode["gain"][
"value"] = feedback;
            this.delayNode =
createDelay(delay +
modulation);

      this.delayNode["delayTime"
]["value"] = delay;
            this.oscNode =
context["createOscillator"]();

      this.oscNode["frequency"][
"value"] = freq;
            this.oscGainNode =
createGain();

      this.oscGainNode["gain"]["
value"] = modulation;

      this.inputNode["connect"](
this.delayNode);

      this.inputNode["connect"](
this.dryNode);

      this.delayNode["connect"](
this.wetNode);

      this.delayNode["connect"](
this.feedbackNode);

      this.feedbackNode["connect
"](this.delayNode);

      this.oscNode["connect"](th
is.oscGainNode);

      this.oscGainNode["connect"
](this.delayNode["delayTime"]);

      startSource(this.oscNode);
      };
      FlangerEffect.prototype.co
nnectTo = function (node)
      {

      this.dryNode["disconnect"]
();

      this.dryNode["connect"](no
de);

      this.wetNode["disconnect"]
();

      this.wetNode["connect"](no
de);
      };
      FlangerEffect.prototype.re
move = function ()
      {

      this.inputNode["disconnect
"]();

      this.delayNode["disconnect
"]();

      this.oscNode["disconnect"]
();

      this.oscGainNode["disconne
ct"]();
```

```
	this.dryNode["disconnect"]
();

	this.wetNode["disconnect"]
();

	this.feedbackNode["disconn
ect"]();
	};
	FlangerEffect.prototype.ge
tInputNode = function ()
	{
		return
this.inputNode;
	};
	FlangerEffect.prototype.se
tParam = function(param, value,
ramp, time)
	{
		switch (param) {
		case 0:		//
mix
			value = value /
100;
			if (value < 0)
value = 0;
			if (value > 1)
value = 1;
			this.params[4]
= value;

	setAudioParam(this.wetNode
["gain"], value / 2, ramp,
time);

	setAudioParam(this.dryNode
["gain"], 1 - (value / 2),
ramp, time);
			break;
		case 6:		//
modulation
			this.params[1]
= value / 1000;

	setAudioParam(this.oscGain
Node["gain"], value / 1000,
ramp, time);
			break;
		case 7:		//
modulation frequency
			this.params[2]
= value;

	setAudioParam(this.oscNode
["frequency"], value, ramp,
time);
			break;
		case 8:		//
feedback
			this.params[3]
= value / 100;

	setAudioParam(this.feedbac
kNode["gain"], value / 100,
ramp, time);
			break;
		}
	};
	function
PhaserEffect(freq, detune, q,
modulation, modfreq, mix)
	{
		this.type =
"phaser";
		this.params = [freq,
detune, q, modulation, modfreq,
mix];
		this.inputNode =
createGain();
		this.dryNode =
createGain();

	this.dryNode["gain"]["valu
e"] = 1 - (mix / 2);
		this.wetNode =
createGain();

	this.wetNode["gain"]["valu
e"] = mix / 2;
		this.filterNode =
context["createBiquadFilter"]()
;
		if (typeof
this.filterNode["type"] ===
"number")

	this.filterNode["type"] =
7;	// all-pass
		else

	this.filterNode["type"] =
"allpass";

	this.filterNode["frequency
"]["value"] = freq;
		if
(this.filterNode["detune"])
	// iOS 6 doesn't have
detune yet
```

```
	this.filterNode["detune"][
"value"] = detune;

	this.filterNode["Q"]["valu
e"] = q;
		this.oscNode =
context["createOscillator"]();

	this.oscNode["frequency"][
"value"] = modfreq;
		this.oscGainNode =
createGain();

	this.oscGainNode["gain"]["
value"] = modulation;

	this.inputNode["connect"](
this.filterNode);

	this.inputNode["connect"](
this.dryNode);

	this.filterNode["connect"]
(this.wetNode);

	this.oscNode["connect"](th
is.oscGainNode);

	this.oscGainNode["connect"
](this.filterNode["frequency"])
;

	startSource(this.oscNode);
	};
	PhaserEffect.prototype.con
nectTo = function (node)
	{

	this.dryNode["disconnect"]
();

	this.dryNode["connect"](no
de);

	this.wetNode["disconnect"]
();

	this.wetNode["connect"](no
de);
	};
	PhaserEffect.prototype.rem
ove = function ()
	{

	this.inputNode["disconnect
"]();

	this.filterNode["disconnec
t"]();

	this.oscNode["disconnect"]
();

	this.oscGainNode["disconne
ct"]();

	this.dryNode["disconnect"]
();

	this.wetNode["disconnect"]
();
	};
	PhaserEffect.prototype.get
InputNode = function ()
	{
		return
this.inputNode;
	};
	PhaserEffect.prototype.set
Param = function(param, value,
ramp, time)
	{
		switch (param) {
		case 0:		//
mix
			value = value /
100;
			if (value < 0)
value = 0;
			if (value > 1)
value = 1;
			this.params[5]
= value;

	setAudioParam(this.wetNode
["gain"], value / 2, ramp,
time);

	setAudioParam(this.dryNode
["gain"], 1 - (value / 2),
ramp, time);
			break;
		case 1:		//
filter frequency
			this.params[0]
= value;

	setAudioParam(this.filterN
```

```
ode["frequency"], value, ramp,
time);
                break;
            case 2:         //
filter detune
                this.params[1]
= value;

      setAudioParam(this.filterN
ode["detune"], value, ramp,
time);
                break;
            case 3:         //
filter Q
                this.params[2]
= value;

      setAudioParam(this.filterN
ode["Q"], value, ramp, time);
                break;
            case 6:         //
modulation
                this.params[3]
= value;

      setAudioParam(this.oscGain
Node["gain"], value, ramp,
time);
                break;
            case 7:         //
modulation frequency
                this.params[4]
= value;

      setAudioParam(this.oscNode
["frequency"], value, ramp,
time);
                break;
            }
      };
      function GainEffect(g)
      {
            this.type = "gain";
            this.params = [g];
            this.node =
createGain();

      this.node["gain"]["value"]
= g;
      };
      GainEffect.prototype.conne
ctTo = function (node_)
      {

      this.node["disconnect"]();

      this.node["connect"](node_
);
      };
      GainEffect.prototype.remov
e = function ()
      {

      this.node["disconnect"]();
      };
      GainEffect.prototype.getIn
putNode = function ()
      {
            return this.node;
      };
      GainEffect.prototype.setPa
ram = function(param, value,
ramp, time)
      {
            switch (param) {
            case 4:         //
gain
                  this.params[0]
= dbToLinear(value);

      setAudioParam(this.node["g
ain"], dbToLinear(value), ramp,
time);
                  break;
            }
      };
      function
TremoloEffect(freq, mix)
      {
            this.type =
"tremolo";
            this.params = [freq,
mix];
            this.node =
createGain();

      this.node["gain"]["value"]
= 1 - (mix / 2);
            this.oscNode =
context["createOscillator"]();

      this.oscNode["frequency"][
"value"] = freq;
            this.oscGainNode =
createGain();

      this.oscGainNode["gain"]["
value"] = mix / 2;

      this.oscNode["connect"](th
is.oscGainNode);
```

```
	this.oscGainNode["connect"](this.node["gain"]);

	startSource(this.oscNode);
	};
	TremoloEffect.prototype.connectTo = function (node_)
	{

	this.node["disconnect"]();

	this.node["connect"](node_);
	};
	TremoloEffect.prototype.remove = function ()
	{

	this.oscNode["disconnect"]();

	this.oscGainNode["disconnect"]();

	this.node["disconnect"]();
	};
	TremoloEffect.prototype.getInputNode = function ()
	{
		return this.node;
	};
	TremoloEffect.prototype.setParam = function(param, value, ramp, time)
	{
		switch (param) {
		case 0:		// mix
			value = value / 100;
			if (value < 0) value = 0;
			if (value > 1) value = 1;
			this.params[1] = value;

	setAudioParam(this.node["gain"]["value"], 1 - (value / 2), ramp, time);

	setAudioParam(this.oscGainNode["gain"]["value"], value / 2, ramp, time);
			break;
		case 7:		// modulation frequency
			this.params[0] = value;

	setAudioParam(this.oscNode["frequency"], value, ramp, time);
			break;
		}
	};
	function RingModulatorEffect(freq, mix)
	{
		this.type = "ringmod";
		this.params = [freq, mix];
		this.inputNode = createGain();
		this.wetNode = createGain();

	this.wetNode["gain"]["value"] = mix;
		this.dryNode = createGain();

	this.dryNode["gain"]["value"] = 1 - mix;
		this.ringNode = createGain();

	this.ringNode["gain"]["value"] = 0;
		this.oscNode = context["createOscillator"]();

	this.oscNode["frequency"]["value"] = freq;

	this.oscNode["connect"](this.ringNode["gain"]);

	startSource(this.oscNode);

	this.inputNode["connect"](this.ringNode);

	this.inputNode["connect"](this.dryNode);

	this.ringNode["connect"](this.wetNode);
	};
```

```
	RingModulatorEffect.prototype.connectTo = function (node_)
	{
	this.wetNode["disconnect"]();
	this.wetNode["connect"](node_);
	this.dryNode["disconnect"]();
	this.dryNode["connect"](node_);
	};
	RingModulatorEffect.prototype.remove = function ()
	{
	this.oscNode["disconnect"]();
	this.ringNode["disconnect"]();
	this.inputNode["disconnect"]();
	this.wetNode["disconnect"]();
	this.dryNode["disconnect"]();
	};
	RingModulatorEffect.prototype.getInputNode = function ()
	{
		return this.inputNode;
	};
	RingModulatorEffect.prototype.setParam = function(param, value, ramp, time)
	{
		switch (param) {
		case 0:		// mix
			value = value / 100;
			if (value < 0) value = 0;
			if (value > 1) value = 1;
			this.params[1] = value;

	setAudioParam(this.wetNode["gain"], value, ramp, time);

	setAudioParam(this.dryNode["gain"], 1 - value, ramp, time);
			break;
		case 7:		// modulation frequency
			this.params[0] = value;

	setAudioParam(this.oscNode["frequency"], value, ramp, time);
			break;
		}
	};
	function DistortionEffect(threshold, headroom, drive, makeupgain, mix)
	{
		this.type = "distortion";
		this.params = [threshold, headroom, drive, makeupgain, mix];
		this.inputNode = createGain();
		this.preGain = createGain();
		this.postGain = createGain();
		this.setDrive(drive, dbToLinear_nocap(makeupgain));
		this.wetNode = createGain();

	this.wetNode["gain"]["value"] = mix;
		this.dryNode = createGain();

	this.dryNode["gain"]["value"] = 1 - mix;
		this.waveShaper = context["createWaveShaper"]();
		this.curve = new Float32Array(65536);

	this.generateColortouchCurve(threshold, headroom);
```

```
	this.waveShaper.curve = this.curve;

	this.inputNode["connect"](this.preGain);

	this.inputNode["connect"](this.dryNode);

	this.preGain["connect"](this.waveShaper);

	this.waveShaper["connect"](this.postGain);

	this.postGain["connect"](this.wetNode);
	};
	DistortionEffect.prototype.setDrive = function (drive, makeupgain)
	{
		if (drive < 0.01)
			drive = 0.01;

	this.preGain["gain"]["value"] = drive;

	this.postGain["gain"]["value"] = Math.pow(1 / drive, 0.6) * makeupgain;
	};
	function e4(x, k)
	{
		return 1.0 - Math.exp(-k * x);
	}
	DistortionEffect.prototype.shape = function (x, linearThreshold, linearHeadroom)
	{
		var maximum = 1.05 * linearHeadroom * linearThreshold;
		var kk = (maximum - linearThreshold);
		var sign = x < 0 ? -1 : +1;
		var absx = x < 0 ? -x : x;
		var shapedInput = absx < linearThreshold ? absx : linearThreshold + kk * e4(absx - linearThreshold, 1.0 / kk);
		shapedInput *= sign;
		return shapedInput;
	};
	DistortionEffect.prototype.generateColortouchCurve = function (threshold, headroom)
	{
		var linearThreshold = dbToLinear_nocap(threshold);
		var linearHeadroom = dbToLinear_nocap(headroom);
		var n = 65536;
		var n2 = n / 2;
		var x = 0;
		for (var i = 0; i < n2; ++i) {
			x = i / n2;
			x = this.shape(x, linearThreshold, linearHeadroom);
			this.curve[n2 + i] = x;
			this.curve[n2 - i - 1] = -x;
		}
	};
	DistortionEffect.prototype.connectTo = function (node)
	{

	this.wetNode["disconnect"]();

	this.wetNode["connect"](node);

	this.dryNode["disconnect"]();

	this.dryNode["connect"](node);
	};
	DistortionEffect.prototype.remove = function ()
	{

	this.inputNode["disconnect"]();

	this.preGain["disconnect"]();

	this.waveShaper["disconnect"]();
```

```
	this.postGain["disconnect"
]();

	this.wetNode["disconnect"]
();

	this.dryNode["disconnect"]
();
	};
	DistortionEffect.prototype
.getInputNode = function ()
	{
		return
this.inputNode;
	};
	DistortionEffect.prototype
.setParam = function(param,
value, ramp, time)
	{
		switch (param) {
		case 0:		//
mix
			value = value /
100;
			if (value < 0)
value = 0;
			if (value > 1)
value = 1;
			this.params[4]
= value;

	setAudioParam(this.wetNode
["gain"], value, ramp, time);

	setAudioParam(this.dryNode
["gain"], 1 - value, ramp,
time);
			break;
		}
	};
	function
CompressorEffect(threshold,
knee, ratio, attack, release)
	{
		this.type =
"compressor";
		this.params =
[threshold, knee, ratio,
attack, release];
		this.node =
context["createDynamicsCompress
or"]();

	this.node["threshold"]["va
lue"] = threshold;

	this.node["knee"]["value"]
= knee;

	this.node["ratio"]["value"
] = ratio;

	this.node["attack"]["value
"] = attack;

	this.node["release"]["valu
e"] = release;
	};
	CompressorEffect.prototype
.connectTo = function (node_)
	{

	this.node["disconnect"]();

	this.node["connect"](node_
);
	};
	CompressorEffect.prototype
.remove = function ()
	{

	this.node["disconnect"]();
	};
	CompressorEffect.prototype
.getInputNode = function ()
	{
		return this.node;
	};
	CompressorEffect.prototype
.setParam = function(param,
value, ramp, time)
	{
	};
	function
AnalyserEffect(fftSize,
smoothing)
	{
		this.type =
"analyser";
		this.params =
[fftSize, smoothing];
		this.node =
context["createAnalyser"]();
		this.node["fftSize"]
= fftSize;

	this.node["smoothingTimeCo
nstant"] = smoothing;
		this.freqBins = new
Float32Array(this.node["frequen
cyBinCount"]);
```

```
            this.signal = new
Uint8Array(fftSize);
            this.peak = 0;
            this.rms = 0;
      };
      AnalyserEffect.prototype.t
ick = function ()
      {

      this.node["getFloatFrequen
cyData"](this.freqBins);

      this.node["getByteTimeDoma
inData"](this.signal);
            var fftSize =
this.node["fftSize"];
            var i = 0;
            this.peak = 0;
            var rmsSquaredSum =
0;
            var s = 0;
            for ( ; i < fftSize;
i++)
            {
                  s =
(this.signal[i] - 128) / 128;
                  if (s < 0)
                        s = -s;
                  if (this.peak <
s)
                        this.peak
= s;
                  rmsSquaredSum
+= s * s;
            }
            this.peak =
linearToDb(this.peak);
            this.rms =
linearToDb(Math.sqrt(rmsSquared
Sum / fftSize));
      };
      AnalyserEffect.prototype.c
onnectTo = function (node_)
      {

      this.node["disconnect"]();

      this.node["connect"](node_
);
      };
      AnalyserEffect.prototype.r
emove = function ()
      {

      this.node["disconnect"]();
      };
      AnalyserEffect.prototype.g
etInputNode = function ()
      {
            return this.node;
      };
      AnalyserEffect.prototype.s
etParam = function(param,
value, ramp, time)
      {
      };
      var OT_POS_SAMPLES = 4;
      function ObjectTracker()
      {
            this.obj = null;
            this.loadUid = 0;
            this.speeds = [];
            this.lastX = 0;
            this.lastY = 0;
            this.moveAngle = 0;
      };
      ObjectTracker.prototype.se
tObject = function (obj_)
      {
            this.obj = obj_;
            if (this.obj)
            {
                  this.lastX =
this.obj.x;
                  this.lastY =
this.obj.y;
            }
            this.speeds.length =
0;
      };
      ObjectTracker.prototype.ha
sObject = function ()
      {
            return !!this.obj;
      };
      ObjectTracker.prototype.ti
ck = function (dt)
      {
            if (!this.obj || dt
=== 0)
                  return;
            this.moveAngle =
cr.angleTo(this.lastX,
this.lastY, this.obj.x,
this.obj.y);
            var s =
cr.distanceTo(this.lastX,
this.lastY, this.obj.x,
this.obj.y) / dt;
            if
(this.speeds.length <
OT_POS_SAMPLES)
```

```
	this.speeds.push(s);
		else
		{

	this.speeds.shift();

	this.speeds.push(s);
		}
		this.lastX =
this.obj.x;
		this.lastY =
this.obj.y;
	};
	ObjectTracker.prototype.ge
tSpeed = function ()
	{
		if
(!this.speeds.length)
			return 0;
		var i, len, sum = 0;
		for (i = 0, len =
this.speeds.length; i < len;
i++)
		{
			sum +=
this.speeds[i];
		}
		return sum /
this.speeds.length;
	};
	ObjectTracker.prototype.ge
tVelocityX = function ()
	{
		return
Math.cos(this.moveAngle) *
this.getSpeed();
	};
	ObjectTracker.prototype.ge
tVelocityY = function ()
	{
		return
Math.sin(this.moveAngle) *
this.getSpeed();
	};
	var iOShadtouch = false;
	// has had touch input on
iOS to work around web audio
API muting
	function
C2AudioBuffer(src_, is_music)
	{
		this.src = src_;
		this.myapi = api;
		this.is_music =
is_music;

	this.added_end_listener =
false;
		var self = this;
		this.outNode = null;
		this.mediaSourceNode
= null;
		this.panWhenReady =
[];	// for web audio API
positioned sounds
		this.seekWhenReady =
0;
		this.pauseWhenReady
= false;

	this.supportWebAudioAPI =
false;
		if (api ===
API_WEBAUDIO && is_music)
		{
			this.myapi =
API_HTML5;
			this.outNode =
createGain();
		}
		this.bufferObject =
null;		// actual audio
object
		this.audioData =
null;			// web
audio api: ajax request result
(compressed audio that needs
decoding)
		var request;
		switch (this.myapi)
{
		case API_HTML5:

	this.bufferObject = new
Audio();
			if (api ===
API_WEBAUDIO &&
context["createMediaElementSour
ce"] && !audRuntime.isFirefox)
			{

	this.supportWebAudioAPI =
true;	// can be routed
through web audio api

	this.bufferObject.addEvent
Listener("canplay", function ()
				{
					if
(!self.mediaSourceNode)
```

```
	// protect against this
event firing twice
	{

	self.mediaSourceNode =
context["createMediaElementSour
ce"](self.bufferObject);

	self.mediaSourceNode["conn
ect"](self.outNode);
	}
		});
		}

	this.bufferObject.autoplay
= false;	// this is only a
source buffer, not an instance

	this.bufferObject.preload
= "auto";

	this.bufferObject.src =
src_;
		break;
		case API_WEBAUDIO:
			request = new
XMLHttpRequest();

	request.open("GET", src_,
true);

	request.responseType =
"arraybuffer";
			request.onload
= function () {

	self.audioData =
request.response;

	self.decodeAudioBuffer();
			};
			request.send();
			break;
		case API_PHONEGAP:

	this.bufferObject = true;
			break;
		case API_APPMOBI:

	this.bufferObject = true;
			break;
		}
	};
	C2AudioBuffer.prototype.de
codeAudioBuffer = function ()
	{

		if
(this.bufferObject ||
!this.audioData)
			return;
	// audio already decoded
or AJAX request not yet
complete
		var self = this;
		if
(context["decodeAudioData"])
		{

	context["decodeAudioData"]
(this.audioData, function
(buffer) {

	self.bufferObject =
buffer;
			var
p, i, len, a;
			if
(!cr.is_undefined(self.playTagW
henReady) && !silent)
			{

	if
(self.panWhenReady.length)

	{

		for (i = 0, len =
self.panWhenReady.length; i <
len; i++)

		{

			p =
self.panWhenReady[i];

			a = new
C2AudioInstance(self,
p.thistag);


	a.setPannerEnabled(true);

			if (typeof
p.objUid !== "undefined")

			{

				p.obj =
audRuntime.getObjectByUID(p.obj
Uid);
```

```
                    if
(!p.obj)

     continue;

                }

                if (p.obj)

                {

                    var px =
cr.rotatePtAround(p.obj.x,
p.obj.y, -
p.obj.layer.getAngle(),
listenerX, listenerY, true);

                    var py =
cr.rotatePtAround(p.obj.x,
p.obj.y, -
p.obj.layer.getAngle(),
listenerX, listenerY, false);

     a.setPan(px, py,
cr.to_degrees(p.obj.angle -
p.obj.layer.getAngle()), p.ia,
p.oa, p.og);

     a.setObject(p.obj);

                }

                else

                {

     a.setPan(p.x, p.y, p.a,
p.ia, p.oa, p.og);

                }

     a.play(self.loopWhenReady,
self.volumeWhenReady,
self.seekWhenReady);

                if
(self.pauseWhenReady)

     a.pause();

     audioInstances.push(a);

          }

     self.panWhenReady.length =
0;

     }

     else

     {

          a = new
C2AudioInstance(self,
self.playTagWhenReady);

     a.play(self.loopWhenReady,
self.volumeWhenReady,
self.seekWhenReady);

          if
(self.pauseWhenReady)

               a.pause();

     audioInstances.push(a);

     }
                         }
                         else
if
(!cr.is_undefined(self.convolve
WhenReady))
                         {

     var convolveNode =
self.convolveWhenReady.convolve
Node;

     convolveNode["normalize"]
= self.normalizeWhenReady;

     convolveNode["buffer"] =
buffer;
                         }
               });
          }
          else
          {
```

```
	this.bufferObject = context["createBuffer"](this.audioData, false);
		if (!cr.is_undefined(this.playTagWhenReady) && !silent)
		{
			var a = new C2AudioInstance(this, this.playTagWhenReady);

	a.play(this.loopWhenReady, this.volumeWhenReady, this.seekWhenReady);
			if (this.pauseWhenReady)

	a.pause();

	audioInstances.push(a);
		}
		else if (!cr.is_undefined(this.convolveWhenReady))
		{
			var convolveNode = this.convolveWhenReady.convolveNode;

	convolveNode["normalize"] = this.normalizeWhenReady;

	convolveNode["buffer"] = this.bufferObject;
		}
	}
};
C2AudioBuffer.prototype.isLoaded = function ()
{
	switch (this.myapi) {
	case API_HTML5:
		return this.bufferObject["readyState"] === 4;	// HAVE_ENOUGH_DATA
	case API_WEBAUDIO:
		return !!this.audioData;
	// null until AJAX request completes
	case API_PHONEGAP:
		return true;
	case API_APPMOBI:
		return true;
	}
	return false;
};
function C2AudioInstance(buffer_, tag_)
{
	var self = this;
	this.tag = tag_;
	this.fresh = true;
	this.stopped = true;
	this.src = buffer_.src;
	this.buffer = buffer_;
	this.myapi = api;
	this.is_music = buffer_.is_music;
	this.playbackRate = 1;
	this.pgended = true;		// for PhoneGap only: ended flag
	this.resume_me = false;		// make sure resumes when leaving suspend
	this.is_paused = false;
	this.resume_position = 0;	// for web audio api to resume from correct playback position
	this.looping = false;
	this.is_muted = false;
	this.is_silent = false;
	this.volume = 1;
	this.mutevol = 1;
	this.startTime = audRuntime.kahanTime.sum;
	this.gainNode = null;
	this.pannerNode = null;
	this.pannerEnabled = false;
	this.objectTracker = null;
	this.panX = 0;
	this.panY = 0;
	this.panAngle = 0;
	this.panConeInner = 0;
```

```
            this.panConeOuter =
0;

      this.panConeOuterGain = 0;
            this.instanceObject
= null;
            var add_end_listener
= false;
            if (this.myapi ===
API_WEBAUDIO &&
this.buffer.myapi === API_HTML5
&&
!this.buffer.supportWebAudioAPI
)
                  this.myapi =
API_HTML5;
            switch (this.myapi)
{
            case API_HTML5:
                  if
(this.is_music)
                  {

      this.instanceObject =
buffer_.bufferObject;

      add_end_listener =
!buffer_.added_end_listener;

      buffer_.added_end_listener
= true;
                  }
                  else
                  {

      this.instanceObject = new
Audio();

      this.instanceObject.autopl
ay = false;

      this.instanceObject.src =
buffer_.bufferObject.src;

      add_end_listener = true;
                  }
                  if
(add_end_listener)
                  {

      this.instanceObject.addEve
ntListener('ended', function ()
{

      audTag = self.tag;

      self.stopped = true;

      audRuntime.trigger(cr.plug
ins_.Audio.prototype.cnds.OnEnd
ed, audInst);
                        });
                  }
                  break;
            case API_WEBAUDIO:
                  this.gainNode =
createGain();

      this.gainNode["connect"](g
etDestinationForTag(tag_));
                  if
(this.buffer.myapi ===
API_WEBAUDIO)
                  {
                        if
(buffer_.bufferObject)
                        {

      this.instanceObject =
context["createBufferSource"]()
;

      this.instanceObject["buffe
r"] = buffer_.bufferObject;

      this.instanceObject["conne
ct"](this.gainNode);
                        }
                  }
                  else
                  {

      this.instanceObject =
this.buffer.bufferObject;
      // reference the audio
element

      this.buffer.outNode["conne
ct"](this.gainNode);
                  }
                  break;
            case API_PHONEGAP:

      this.instanceObject = new
window["Media"](appPath +
this.src, null, null, function
(status) {
                              if
(status ===
window["Media"]["MEDIA_STOPPED"
])
```

```
                                {
      self.pgended = true;

      self.stopped = true;

      audTag = self.tag;

      audRuntime.trigger(cr.plug
ins_.Audio.prototype.cnds.OnEnd
ed, audInst);
                                }
                  });
                  break;
            case API_APPMOBI:

      this.instanceObject =
true;
                  break;
            }
      };
      C2AudioInstance.prototype.
hasEnded = function ()
      {
            switch (this.myapi)
{
            case API_HTML5:
                  return
this.instanceObject.ended;
            case API_WEBAUDIO:
                  if
(this.buffer.myapi ===
API_WEBAUDIO)
                  {
                        if
(!this.fresh && !this.stopped
&& this.instanceObject["loop"])

      return false;
                        if
(this.is_paused)

      return false;
                        return
(audRuntime.kahanTime.sum -
this.startTime) >
this.buffer.bufferObject["durat
ion"];
                  }
                  else
                        return
this.instanceObject.ended;
            case API_PHONEGAP:
                  return
this.pgended;
            case API_APPMOBI:
                        true; //
recycling an AppMobi sound does
not matter because it will just
do another throwaway playSound
            }
            return true;
      };
      C2AudioInstance.prototype.
canBeRecycled = function ()
      {
            if (this.fresh ||
this.stopped)
                        return true;
            // not yet used or
is not playing
            return
this.hasEnded();
      };
      C2AudioInstance.prototype.
setPannerEnabled = function
(enable_)
      {
            if (api !==
API_WEBAUDIO)
                        return;
            if
(!this.pannerEnabled &&
enable_)
            {
                        if
(!this.gainNode)
                              return;
                        if
(!this.pannerNode)
                        {

      this.pannerNode =
context["createPanner"]();
                              if
(typeof
this.pannerNode["panningModel"]
=== "number")

      this.pannerNode["panningMo
del"] = panningModel;
                              else

      this.pannerNode["panningMo
del"] = ["equalpower", "HRTF",
"soundfield"][panningModel];
                              if
(typeof
this.pannerNode["distanceModel"
] === "number")
```

```
	this.pannerNode["distanceM
odel"] = distanceModel;
                        else

	this.pannerNode["distanceM
odel"] = ["linear", "inverse",
"exponential"][distanceModel];

	this.pannerNode["refDistan
ce"] = refDistance;

	this.pannerNode["maxDistan
ce"] = maxDistance;

	this.pannerNode["rolloffFa
ctor"] = rolloffFactor;
                }

	this.gainNode["disconnect"
]();

	this.gainNode["connect"](t
his.pannerNode);

	this.pannerNode["connect"]
(getDestinationForTag(this.tag)
);

	this.pannerEnabled = true;
            }
            else if
(this.pannerEnabled &&
!enable_)
            {
                if
(!this.gainNode)
                        return;

	this.pannerNode["disconnec
t"]();

	this.gainNode["disconnect"
]();

	this.gainNode["connect"](g
etDestinationForTag(this.tag));

	this.pannerEnabled =
false;
            }
	};
	C2AudioInstance.prototype.
setPan = function (x, y, angle,
innerangle, outerangle,
outergain)
	{
            if
(!this.pannerEnabled || api !==
API_WEBAUDIO)
                return;

	this.pannerNode["setPositi
on"](x, y, 0);

	this.pannerNode["setOrient
ation"](Math.cos(cr.to_radians(
angle)),
Math.sin(cr.to_radians(angle)),
0);

	this.pannerNode["coneInner
Angle"] = innerangle;

	this.pannerNode["coneOuter
Angle"] = outerangle;

	this.pannerNode["coneOuter
Gain"] = outergain;
            this.panX = x;
            this.panY = y;
            this.panAngle =
angle;
            this.panConeInner =
innerangle;
            this.panConeOuter =
outerangle;

	this.panConeOuterGain =
outergain;
	};
	C2AudioInstance.prototype.
setObject = function (o)
	{
            if
(!this.pannerEnabled || api !==
API_WEBAUDIO)
                return;
            if
(!this.objectTracker)

	this.objectTracker = new
ObjectTracker();

	this.objectTracker.setObje
ct(o);
	};
	C2AudioInstance.prototype.
tick = function (dt)
	{
            if
(!this.pannerEnabled || api !==
```

```
API_WEBAUDIO ||
!this.objectTracker ||
!this.objectTracker.hasObject()
|| !this.isPlaying())
            {
                  return;
            }

      this.objectTracker.tick(dt
);
            var inst =
this.objectTracker.obj;
            var px =
cr.rotatePtAround(inst.x,
inst.y, -inst.layer.getAngle(),
listenerX, listenerY, true);
            var py =
cr.rotatePtAround(inst.x,
inst.y, -inst.layer.getAngle(),
listenerX, listenerY, false);

      this.pannerNode["setPositi
on"](px, py, 0);
            var a = 0;
            if (typeof
this.objectTracker.obj.angle
!== "undefined")
            {
                  a = inst.angle
- inst.layer.getAngle();

      this.pannerNode["setOrient
ation"](Math.cos(a),
Math.sin(a), 0);
            }

      this.pannerNode["setVeloci
ty"](this.objectTracker.getVelo
cityX(),
this.objectTracker.getVelocityY
(), 0);
      };
      C2AudioInstance.prototype.
play = function (looping, vol,
fromPosition)
      {
            var instobj =
this.instanceObject;
            this.looping =
looping;
            this.volume = vol;
            var seekPos =
fromPosition || 0;
            switch (this.myapi)
{
            case API_HTML5:
                  if
(instobj.playbackRate !== 1.0)

      instobj.playbackRate =
1.0;
                  if
(instobj.volume !== vol *
masterVolume)

      instobj.volume = vol *
masterVolume;
                  if
(instobj.loop !== looping)

      instobj.loop = looping;
                  if
(instobj.muted)

      instobj.muted = false;
                  if
(instobj.currentTime !==
seekPos)
                  {
                        try {

      instobj.currentTime =
seekPos;
                        }
                        catch
(err)
                        {
;
                        }
                  }
                  if
(this.is_music &&
isMusicWorkaround &&
!audRuntime.isInUserInputEvent)

      musicPlayNextTouch.push(th
is);
                  else

      this.instanceObject.play()
;
                  break;
            case API_WEBAUDIO:
                  this.muted =
false;
                  this.mutevol =
1;
                  if
(this.buffer.myapi ===
API_WEBAUDIO)
                  {
```

```
                        if
(!this.fresh)
                        {

     this.instanceObject =
context["createBufferSource"]()
;

     this.instanceObject["buffe
r"] = this.buffer.bufferObject;

     this.instanceObject["conne
ct"](this.gainNode);
                        }

     this.instanceObject.loop =
looping;

     this.gainNode["gain"]["val
ue"] = vol * masterVolume;
                        if
(seekPos === 0)

     startSource(this.instanceO
bject);
                        else

     startSourceAt(this.instanc
eObject, seekPos,
this.getDuration());
                  }
                  else
                  {
                        if
(instobj.playbackRate !== 1.0)

     instobj.playbackRate =
1.0;
                        if
(instobj.loop !== looping)

     instobj.loop = looping;

     this.gainNode["gain"]["val
ue"] = vol * masterVolume;
                        if
(instobj.currentTime !==
seekPos)
                        {
                              try
{

     instobj.currentTime =
seekPos;
                              }
     catch (err)
                              {
;
                              }
                        }
                        if
(this.is_music &&
isMusicWorkaround &&
!audRuntime.isInUserInputEvent)

     musicPlayNextTouch.push(th
is);
                        else

     instobj.play();
                  }
                  break;
            case API_PHONEGAP:
                  if
((!this.fresh && this.stopped)
|| seekPos !== 0)

     instobj["seekTo"](seekPos)
;

     instobj["play"]();
                  this.pgended =
false;
                  break;
            case API_APPMOBI:
                  if
(audRuntime.isDirectCanvas)

     AppMobi["context"]["playSo
und"](this.src, looping);
                  else

     AppMobi["player"]["playSou
nd"](this.src, looping);
                  break;
            }
            this.playbackRate =
1;
            this.startTime =
audRuntime.kahanTime.sum -
seekPos;
            this.fresh = false;
            this.stopped =
false;
            this.is_paused =
false;
      };
      C2AudioInstance.prototype.
stop = function ()
      {
```

```
		switch (this.myapi)
{
		case API_HTML5:
			if
(!this.instanceObject.paused)

	this.instanceObject.pause(
);
			break;
		case API_WEBAUDIO:
			if
(this.buffer.myapi ===
API_WEBAUDIO)

	stopSource(this.instanceOb
ject);
			else
			{
				if
(!this.instanceObject.paused)

	this.instanceObject.pause(
);
			}
			break;
		case API_PHONEGAP:

	this.instanceObject["stop"
]();
			break;
		case API_APPMOBI:
			if
(audRuntime.isDirectCanvas)

	AppMobi["context"]["stopSo
und"](this.src);
			break;
		}
		this.stopped = true;
		this.is_paused =
false;
	};
	C2AudioInstance.prototype.
pause = function ()
	{
		if (this.fresh ||
this.stopped || this.hasEnded()
|| this.is_paused)
			return;
		switch (this.myapi)
{
		case API_HTML5:
			if
(!this.instanceObject.paused)

	this.instanceObject.pause(
);
			break;
		case API_WEBAUDIO:
			if
(this.buffer.myapi ===
API_WEBAUDIO)
			{

	this.resume_position =
this.getPlaybackTime();
				if
(this.looping)

	this.resume_position =
this.resume_position %
this.getDuration();

	stopSource(this.instanceOb
ject);
			}
			else
			{
				if
(!this.instanceObject.paused)

	this.instanceObject.pause(
);
			}
			break;
		case API_PHONEGAP:

	this.instanceObject["pause
"]();
			break;
		case API_APPMOBI:
			if
(audRuntime.isDirectCanvas)

	AppMobi["context"]["stopSo
und"](this.src);
			break;
		}
		this.is_paused =
true;
	};
	C2AudioInstance.prototype.
resume = function ()
	{
		if (this.fresh ||
this.stopped || this.hasEnded()
|| !this.is_paused)
			return;
		switch (this.myapi)
{
```

```
            case API_HTML5:
      this.instanceObject.play()
;
                  break;
            case API_WEBAUDIO:
                  if
(this.buffer.myapi ===
API_WEBAUDIO)
                  {

      this.instanceObject =
context["createBufferSource"]()
;

      this.instanceObject["buffe
r"] = this.buffer.bufferObject;

      this.instanceObject["conne
ct"](this.gainNode);

      this.instanceObject.loop =
this.looping;

      this.gainNode["gain"]["val
ue"] = masterVolume *
this.volume * this.mutevol;

      this.startTime =
audRuntime.kahanTime.sum -
this.resume_position;

      startSourceAt(this.instanc
eObject, this.resume_position,
this.getDuration());
                  }
                  else
                  {

      this.instanceObject.play()
;
                  }
                  break;
            case API_PHONEGAP:

      this.instanceObject["play"
]();
                  break;
            case API_APPMOBI:
                  if
(audRuntime.isDirectCanvas)

      AppMobi["context"]["resume
Sound"](this.src);
                  break;
            }
```

```
            this.is_paused =
false;
      };
      C2AudioInstance.prototype.
seek = function (pos)
      {
            if (this.fresh ||
this.stopped ||
this.hasEnded())
                  return;
            switch (this.myapi)
{
            case API_HTML5:
                  try {

      this.instanceObject.curren
tTime = pos;
                  }
                  catch (e) {}
                  break;
            case API_WEBAUDIO:
                  if
(this.buffer.myapi ===
API_WEBAUDIO)
                  {
                        if
(this.is_paused)

      this.resume_position =
pos;
                        else
                        {

      this.pause();

      this.resume_position =
pos;

      this.resume();
                        }
                  }
                  else
                  {
                        try {

      this.instanceObject.curren
tTime = pos;
                        }
                        catch (e)
{}
                  }
                  break;
            case API_PHONEGAP:
                  break;
            case API_APPMOBI:
```

```
                if
(audRuntime.isDirectCanvas)

     AppMobi["context"]["seekSo
und"](this.src, pos);
                break;
           }
     };
     C2AudioInstance.prototype.
reconnect = function (toNode)
     {
           if (this.myapi !==
API_WEBAUDIO)
                return;
           if
(this.pannerEnabled)
           {

     this.pannerNode["disconnec
t"]();

     this.pannerNode["connect"]
(toNode);
           }
           else
           {

     this.gainNode["disconnect"
]();

     this.gainNode["connect"](t
oNode);
           }
     };
     C2AudioInstance.prototype.
getDuration = function ()
     {
           switch (this.myapi)
{
           case API_HTML5:
                if (typeof
this.instanceObject.duration
!== "undefined")
                      return
this.instanceObject.duration;
                else
                      return 0;
           case API_WEBAUDIO:
                return
this.buffer.bufferObject["durat
ion"];
           case API_PHONEGAP:
                return
this.instanceObject["getDuratio
n"]();
           case API_APPMOBI:
                if
(audRuntime.isDirectCanvas)
                      return
AppMobi["context"]["getDuration
Sound"](this.src);
                else
                      return 0;
           }
           return 0;
     };
     C2AudioInstance.prototype.
getPlaybackTime = function ()
     {
           var duration =
this.getDuration();
           var ret = 0;
           switch (this.myapi)
{
           case API_HTML5:
                if (typeof
this.instanceObject.currentTime
!== "undefined")
                      ret =
this.instanceObject.currentTime
;
                break;
           case API_WEBAUDIO:
                if
(this.buffer.myapi ===
API_WEBAUDIO)
                {
                      if
(this.is_paused)

     return
this.resume_position;
                      else
                            ret
= audRuntime.kahanTime.sum -
this.startTime;
                }
                else if (typeof
this.instanceObject.currentTime
!== "undefined")
                      ret =
this.instanceObject.currentTime
;
                break;
           case API_PHONEGAP:
                break;
           case API_APPMOBI:
                if
(audRuntime.isDirectCanvas)
                      ret =
AppMobi["context"]["getPlayback
TimeSound"](this.src);
```

```
                break;
            }
            if (!this.looping &&
ret > duration)
                ret = duration;
            return ret;
        };
        C2AudioInstance.prototype.
isPlaying = function ()
        {
            return
!this.is_paused && !this.fresh
&& !this.stopped &&
!this.hasEnded();
        };
        C2AudioInstance.prototype.
setVolume = function (v)
        {
            this.volume = v;
            this.updateVolume();
        };
        C2AudioInstance.prototype.
updateVolume = function ()
        {
            var volToSet =
this.volume * masterVolume;
            switch (this.myapi)
{
            case API_HTML5:
                if
(this.instanceObject.volume &&
this.instanceObject.volume !==
volToSet)

        this.instanceObject.volume
= volToSet;
                break;
            case API_WEBAUDIO:

        this.gainNode["gain"]["val
ue"] = volToSet * this.mutevol;
                break;
            case API_PHONEGAP:
                break;
            case API_APPMOBI:
                break;
            }
        };
        C2AudioInstance.prototype.
getVolume = function ()
        {
            return this.volume;
        };
        C2AudioInstance.prototype.
doSetMuted = function (m)
        {
            switch (this.myapi)
{
            case API_HTML5:
                if
(this.instanceObject.muted !==
!!m)

        this.instanceObject.muted
= !!m;
                break;
            case API_WEBAUDIO:
                this.mutevol =
(m ? 0 : 1);

        this.gainNode["gain"]["val
ue"] = masterVolume *
this.volume * this.mutevol;
                break;
            case API_PHONEGAP:
                break;
            case API_APPMOBI:
                break;
            }
        };
        C2AudioInstance.prototype.
setMuted = function (m)
        {
            this.is_muted = !!m;

        this.doSetMuted(this.is_mu
ted || this.is_silent);
        };
        C2AudioInstance.prototype.
setSilent = function (m)
        {
            this.is_silent =
!!m;

        this.doSetMuted(this.is_mu
ted || this.is_silent);
        };
        C2AudioInstance.prototype.
setLooping = function (l)
        {
            this.looping = l;
            switch (this.myapi)
{
            case API_HTML5:
                if
(this.instanceObject.loop !==
!!l)

        this.instanceObject.loop =
!!l;
                break;
            case API_WEBAUDIO:
```

```
                if
(this.instanceObject.loop !==
!!l)

     this.instanceObject.loop =
!!l;
                break;
           case API_PHONEGAP:
                break;
           case API_APPMOBI:
                if
(audRuntime.isDirectCanvas)

     AppMobi["context"]["setLoo
pingSound"](this.src, l);
                break;
           }
     };
     C2AudioInstance.prototype.
setPlaybackRate = function (r)
     {
           this.playbackRate =
r;

     this.updatePlaybackRate();
     };
     C2AudioInstance.prototype.
updatePlaybackRate = function
()
     {
           var r =
this.playbackRate;
           if ((timescale_mode
=== 1 && !this.is_music) ||
timescale_mode === 2)
                r *=
audRuntime.timescale;
           switch (this.myapi)
{
           case API_HTML5:
                if
(this.instanceObject.playbackRa
te !== r)

     this.instanceObject.playba
ckRate = r;
                break;
           case API_WEBAUDIO:
                if
(this.buffer.myapi ===
API_WEBAUDIO)
                {
                     if
(this.instanceObject["playbackR
ate"]["value"] !== r)

     this.instanceObject["playb
ackRate"]["value"] = r;
                }
                else
                {
                     if
(this.instanceObject.playbackRa
te !== r)

     this.instanceObject.playba
ckRate = r;
                }
                break;
           case API_PHONEGAP:
                break;
           case API_APPMOBI:
                break;
           }
     };
     C2AudioInstance.prototype.
setSuspended = function (s)
     {
           switch (this.myapi)
{
           case API_HTML5:
                if (s)
                {
                     if
(this.isPlaying())
                     {

     this.instanceObject["pause
"]();

     this.resume_me = true;
                     }
                     else

     this.resume_me = false;
                }
                else
                {
                     if
(this.resume_me)

     this.instanceObject["play"
]();
                }
                break;
           case API_WEBAUDIO:
                if (s)
                {
                     if
(this.isPlaying())
                     {
```

```
                        if
(this.buffer.myapi ===
API_WEBAUDIO)
                        {

     this.resume_position =
this.getPlaybackTime();

     if (this.looping)

          this.resume_position
= this.resume_position %
this.getDuration();

     stopSource(this.instanceOb
ject);
                        }
                        else

     this.instanceObject["pause
"]();

     this.resume_me = true;
                    }
                    else

     this.resume_me = false;
               }
               else
               {
                    if
(this.resume_me)
                    {
                        if
(this.buffer.myapi ===
API_WEBAUDIO)
                        {

     this.instanceObject =
context["createBufferSource"]()
;

     this.instanceObject["buffe
r"] = this.buffer.bufferObject;

     this.instanceObject["conne
ct"](this.gainNode);

     this.instanceObject.loop =
this.looping;

     this.gainNode["gain"]["val
ue"] = masterVolume *
this.volume * this.mutevol;

     this.startTime =
audRuntime.kahanTime.sum -
this.resume_position;

     startSourceAt(this.instanc
eObject, this.resume_position,
this.getDuration());
                        }
                        else
                        {

     this.instanceObject["play"
]();
                        }
                    }
                }
                break;
            case API_PHONEGAP:
                if (s)
                {
                    if
(this.isPlaying())
                    {

     this.instanceObject["pause
"]();

     this.resume_me = true;
                    }
                    else

     this.resume_me = false;
                }
                else
                {
                    if
(this.resume_me)

     this.instanceObject["play"
]();
                }
                break;
            case API_APPMOBI:
                break;
            }
        };
        pluginProto.Instance =
function(type)
        {
            this.type = type;
            this.runtime =
type.runtime;
            audRuntime =
this.runtime;
            audInst = this;
            this.listenerTracker
= null;
```

```
          this.listenerZ = -
600;
          if
((this.runtime.isiOS ||
(this.runtime.isAndroid &&
(this.runtime.isChrome ||
this.runtime.isAndroidStockBrow
ser))) &&
!this.runtime.isCrosswalk &&
!this.runtime.isDomFree)
          {

     isMusicWorkaround = true;
          }
          context = null;
          if (typeof
AudioContext !== "undefined")
          {
               api =
API_WEBAUDIO;
               context = new
AudioContext();
          }
          else if (typeof
webkitAudioContext !==
"undefined")
          {
               api =
API_WEBAUDIO;
               context = new
webkitAudioContext();
          }
          if
((this.runtime.isiOS && api ===
API_WEBAUDIO) ||
isMusicWorkaround)
          {

     document.addEventListener(
"touchstart", function ()
               {
                    var i,
len, m;
                    if
(!iOShadtouch && context)
                    {
                         var
buffer =
context["createBuffer"](1, 1,
22050);
                         var
source =
context["createBufferSource"]()
;

     source["buffer"] = buffer;

     source["connect"](context[
"destination"]);

     startSource(source);

     iOShadtouch = true;
                    }
                    if
(isMusicWorkaround)
                    {
                         if
(!silent)
                         {

     for (i = 0, len =
musicPlayNextTouch.length; i <
len; ++i)

     {

          m =
musicPlayNextTouch[i];

          if (!m.stopped &&
!m.is_paused)


     m.instanceObject.play();

     }
                         }

     musicPlayNextTouch.length
= 0;
                    }
               }, true);
          }
          if (api !==
API_WEBAUDIO)
          {
               if
(this.runtime.isPhoneGap)
                    api =
API_PHONEGAP;
               else if
(this.runtime.isAppMobi)
                    api =
API_APPMOBI;
          }
          if (api ===
API_PHONEGAP)
          {
               appPath =
location.href;
```

```
                var i =
appPath.lastIndexOf("/");
                if (i > -1)
                     appPath =
appPath.substr(0, i + 1);
                appPath =
appPath.replace("file://", "");
           }
           if
(this.runtime.isSafari &&
this.runtime.isWindows &&
typeof Audio === "undefined")
           {
                alert("It looks
like you're using Safari for
Windows without Quicktime.
Audio cannot be played until
Quicktime is installed.");

     this.runtime.DestroyInstan
ce(this);
           }
           else
           {
                if
(this.runtime.isDirectCanvas)
                     useOgg =
this.runtime.isAndroid;
     // AAC on iOS, OGG on
Android
                else
                {
                     try {

     useOgg = !!(new
Audio().canPlayType('audio/ogg;
codecs="vorbis"'));
                     }
                     catch (e)
                     {

     useOgg = false;
                     }
                }
                switch (api) {
                case API_HTML5:
;
                     break;
                case
API_WEBAUDIO:
;
                     break;
                case
API_PHONEGAP:
;
                     break;
                case
API_APPMOBI:
;
                     break;
                default:
;
                }

     this.runtime.tickMe(this);
           }
     };
     var instanceProto =
pluginProto.Instance.prototype;
     instanceProto.onCreate =
function ()
     {
           timescale_mode =
this.properties[0];   // 0 =
off, 1 = sounds only, 2 = all
           this.saveload =
this.properties[1];        // 0
= all, 1 = sounds only, 2 =
music only, 3 = none
           panningModel =
this.properties[2];        // 0
= equalpower, 1 = hrtf, 3 =
soundfield
           distanceModel =
this.properties[3];        // 0
= linear, 1 = inverse, 2 =
exponential
           this.listenerZ = -
this.properties[4];
           refDistance =
this.properties[5];
           maxDistance =
this.properties[6];
           rolloffFactor =
this.properties[7];
           this.listenerTracker
= new ObjectTracker();
           if (api ===
API_WEBAUDIO)
           {

     context["listener"]["speed
OfSound"] = this.properties[8];

     context["listener"]["doppl
erFactor"] =
this.properties[9];

     context["listener"]["setPo
sition"](this.runtime.draw_widt
h / 2, this.runtime.draw_height
/ 2, this.listenerZ);
```

```
context["listener"]["setOr
ientation"](0, 0, 1, 0, -1, 0);

window["c2OnAudioMicStream
"] = function
(localMediaStream, tag)
{
if
(micSource)

micSource["disconnect"]();
micTag =
tag.toLowerCase();
micSource
=
context["createMediaStreamSourc
e"](localMediaStream);

micSource["connect"](getDe
stinationForTag(micTag));
};
}

this.runtime.addSuspendCal
lback(function(s)
{

audInst.onSuspend(s);
});
var self = this;

this.runtime.addDestroyCal
lback(function (inst)
{

self.onInstanceDestroyed(i
nst);
});
};
instanceProto.onInstanceDe
stroyed = function (inst)
{
var i, len, a;
for (i = 0, len =
audioInstances.length; i < len;
i++)
{
a =
audioInstances[i];
if
(a.objectTracker)
{
if
(a.objectTracker.obj === inst)
{
a.objectTracker.obj =
null;
if
(a.pannerEnabled &&
a.isPlaying() && a.looping)

a.stop();
}
}
}
if
(this.listenerTracker.obj ===
inst)

this.listenerTracker.obj =
null;
};
instanceProto.saveToJSON =
function ()
{
var o = {
"silent":
silent,
"masterVolume":
masterVolume,
"listenerZ":
this.listenerZ,
"listenerUid":
this.listenerTracker.hasObject(
) ?
this.listenerTracker.obj.uid :
-1,
"playing": [],
"effects": {}
};
var playingarr =
o["playing"];
var i, len, a, d, p,
panobj, playbackTime;
for (i = 0, len =
audioInstances.length; i < len;
i++)
{
a =
audioInstances[i];
if
(!a.isPlaying())
continue;
// no
need to save stopped sounds
if
(this.saveload === 3) // not
saving/loading any sounds/music
continue;
```

```
                if (a.is_music
&& this.saveload === 1)     //
not saving/loading music
                      continue;
                if (!a.is_music
&& this.saveload === 2)     //
not saving/loading sound
                      continue;
                playbackTime =
a.getPlaybackTime();
                if (a.looping)

     playbackTime =
playbackTime % a.getDuration();
                d = {
                      "tag":
a.tag,

     "buffersrc": a.buffer.src,

     "is_music": a.is_music,

     "playbackTime":
playbackTime,
                      "volume":
a.volume,

     "looping": a.looping,
                      "muted":
a.is_muted,

     "playbackRate":
a.playbackRate,
                      "paused":
a.is_paused,

     "resume_position":
a.resume_position
                };
                if
(a.pannerEnabled)
                {
                      d["pan"]
= {};
                      panobj =
d["pan"];
                      if
(a.objectTracker &&
a.objectTracker.hasObject())
                      {

     panobj["objUid"] =
a.objectTracker.obj.uid;
                      }
                      else
                      {

     panobj["x"] = a.panX;

     panobj["y"] = a.panY;

     panobj["a"] = a.panAngle;
                      }

     panobj["ia"] =
a.panConeInner;

     panobj["oa"] =
a.panConeOuter;

     panobj["og"] =
a.panConeOuterGain;
                }

     playingarr.push(d);
           }
           var fxobj =
o["effects"];
           var fxarr;
           for (p in effects)
           {
                 if
(effects.hasOwnProperty(p))
                 {
                      fxarr =
[];
                      for (i =
0, len = effects[p].length; i <
len; i++)
                      {

     fxarr.push({ "type":
effects[p][i].type, "params":
effects[p][i].params });
                      }
                      fxobj[p]
= fxarr;
                 }
           }
           return o;
     };
     var
objectTrackerUidsToLoad = [];
     instanceProto.loadFromJSON
= function (o)
     {
           var setSilent =
o["silent"];
           masterVolume =
o["masterVolume"];
           this.listenerZ =
o["listenerZ"];
```

```
this.listenerTracker.setOb
ject(null);
            var listenerUid =
o["listenerUid"];
            if (listenerUid !==
-1)
            {

      this.listenerTracker.loadU
id = listenerUid;

      objectTrackerUidsToLoad.pu
sh(this.listenerTracker);
            }
            var playingarr =
o["playing"];
            var i, len, d, src,
is_music, tag, playbackTime,
looping, vol, b, a, p, pan,
panObjUid;
            if (this.saveload
!== 3)
            {
                  for (i = 0, len
= audioInstances.length; i <
len; i++)
                  {
                        a =
audioInstances[i];
                        if
(a.is_music && this.saveload
=== 1)

      continue;       // only
saving/loading sound: leave
music playing
                        if
(!a.is_music && this.saveload
=== 2)

      continue;       // only
saving/loading music: leave
sound playing
                        a.stop();
                  }
            }
            var fxarr, fxtype,
fxparams, fx;
            for (p in effects)
            {
                  if
(effects.hasOwnProperty(p))
                  {
                        for (i =
0, len = effects[p].length; i <
len; i++)

      effects[p][i].remove();
                  }
            }
            cr.wipe(effects);
            for (p in
o["effects"])
            {
                  if
(o["effects"].hasOwnProperty(p)
)
                  {
                        fxarr =
o["effects"][p];
                        for (i =
0, len = fxarr.length; i < len;
i++)
                        {

      fxtype = fxarr[i]["type"];

      fxparams =
fxarr[i]["params"];

      switch (fxtype) {
                              case
"filter":

      addEffectForTag(p, new
FilterEffect(fxparams[0],
fxparams[1], fxparams[2],
fxparams[3], fxparams[4],
fxparams[5]));

      break;
                              case
"delay":

      addEffectForTag(p, new
DelayEffect(fxparams[0],
fxparams[1], fxparams[2]));

      break;
                              case
"convolve":

      src = fxparams[2];

      b =
this.getAudioBuffer(src,
false);

      if (b.bufferObject)
```

```
{
        fx = new ConvolveEffect(b.bufferObject, fxparams[0], fxparams[1], src);
}
else
{
        fx = new ConvolveEffect(null, fxparams[0], fxparams[1], src);
        b.normalizeWhenReady = fxparams[0];
        b.convolveWhenReady = fx;
}
addEffectForTag(p, fx);
break;
                        case "flanger":
addEffectForTag(p, new FlangerEffect(fxparams[0], fxparams[1], fxparams[2], fxparams[3], fxparams[4]));
break;
                        case "phaser":
addEffectForTag(p, new PhaserEffect(fxparams[0], fxparams[1], fxparams[2], fxparams[3], fxparams[4], fxparams[5]));
break;
                        case "gain":
addEffectForTag(p, new GainEffect(fxparams[0]));
break;
                        case "tremolo":
addEffectForTag(p, new TremoloEffect(fxparams[0], fxparams[1]));
break;
                        case "ringmod":
addEffectForTag(p, new RingModulatorEffect(fxparams[0], fxparams[1]));
break;
                        case "distortion":
addEffectForTag(p, new DistortionEffect(fxparams[0], fxparams[1], fxparams[2], fxparams[3], fxparams[4]));
break;
                        case "compressor":
addEffectForTag(p, new CompressorEffect(fxparams[0], fxparams[1], fxparams[2], fxparams[3], fxparams[4]));
break;
                        case "analyser":
addEffectForTag(p, new AnalyserEffect(fxparams[0], fxparams[1]));
break;
                        }
                    }
                }
            }
            for (i = 0, len = playingarr.length; i < len; i++)
            {
                if (this.saveload === 3) // not saving/loading any sounds/music
                    continue;
                d = playingarr[i];
                src = d["buffersrc"];
```

```
                is_music =
d["is_music"];
                tag = d["tag"];
                playbackTime =
d["playbackTime"];
                looping =
d["looping"];
                vol =
d["volume"];
                pan = d["pan"];
                panObjUid =
(pan &&
pan.hasOwnProperty("objUid")) ?
pan["objUid"] : -1;
                if (is_music &&
this.saveload === 1)  // not
saving/loading music
                      continue;
                if (!is_music
&& this.saveload === 2)    //
not saving/loading sound
                      continue;
                a =
this.getAudioInstance(src, tag,
is_music, looping, vol);
                if (!a)
                {
                      b =
this.getAudioBuffer(src,
is_music);

     b.seekWhenReady =
playbackTime;

     b.pauseWhenReady =
d["paused"];
                      if (pan)
                      {
                            if
(panObjUid !== -1)
                            {

     b.panWhenReady.push({
objUid: panObjUid, ia:
pan["ia"], oa: pan["oa"], og:
pan["og"], thistag: tag });
                            }
                            else
                            {

     b.panWhenReady.push({ x:
pan["x"], y: pan["y"], a:
pan["a"], ia: pan["ia"], oa:
pan["oa"], og: pan["og"],
thistag: tag });
                            }
                                  }
                                  continue;
                            }

     a.resume_position =
d["resume_position"];

     a.setPannerEnabled(!!pan);
                      a.play(looping,
vol, playbackTime);

     a.updatePlaybackRate();

     a.updateVolume();

     a.doSetMuted(a.is_muted ||
a.is_silent);
                      if
(d["paused"])

     a.pause();
                      if (d["muted"])

     a.setMuted(true);

     a.doSetMuted(a.is_muted ||
a.is_silent);
                      if (pan)
                      {
                            if
(panObjUid !== -1)
                            {

     a.objectTracker =
a.objectTracker || new
ObjectTracker();

     a.objectTracker.loadUid =
panObjUid;

     objectTrackerUidsToLoad.pu
sh(a.objectTracker);
                            }
                            else
                            {

     a.setPan(pan["x"],
pan["y"], pan["a"], pan["ia"],
pan["oa"], pan["og"]);
                            }
                      }
                }
                if (setSilent &&
!silent)              //
setting silent
                {
```

```
                for (i = 0, len
= audioInstances.length; i <
len; i++)

     audioInstances[i].setSilen
t(true);
                silent = true;
           }
           else if (!setSilent
&& silent)      // setting not
silent
           {
                for (i = 0, len
= audioInstances.length; i <
len; i++)

     audioInstances[i].setSilen
t(false);
                silent = false;
           }
     };
     instanceProto.afterLoad =
function ()
     {
           var i, len, ot,
inst;
           for (i = 0, len =
objectTrackerUidsToLoad.length;
i < len; i++)
           {
                ot =
objectTrackerUidsToLoad[i];
                inst =
this.runtime.getObjectByUID(ot.
loadUid);

     ot.setObject(inst);
                ot.loadUid = -
1;
                if (inst)
                {
                     listenerX
= inst.x;
                     listenerY
= inst.y;
                }
           }

     objectTrackerUidsToLoad.le
ngth = 0;
     };
     instanceProto.onSuspend =
function (s)
     {
           var i, len;
           for (i = 0, len =
audioInstances.length; i < len;
i++)

     audioInstances[i].setSuspe
nded(s);
     };
     instanceProto.tick =
function ()
     {
           var dt =
this.runtime.dt;
           var i, len, a;
           for (i = 0, len =
audioInstances.length; i < len;
i++)
           {
                a =
audioInstances[i];
                a.tick(dt);
                if (a.myapi !==
API_HTML5 && a.myapi !==
API_APPMOBI)
                {
                     if
(!a.fresh && !a.stopped &&
a.hasEnded())
                     {

     a.stopped = true;

     audTag = a.tag;

     audRuntime.trigger(cr.plug
ins_.Audio.prototype.cnds.OnEnd
ed, audInst);
                     }
                }
                if
(timescale_mode !== 0)

     a.updatePlaybackRate();
           }
           var p, arr, f;
           for (p in effects)
           {
                if
(effects.hasOwnProperty(p))
                {
                     arr =
effects[p];
                     for (i =
0, len = arr.length; i < len;
i++)
                     {
```

```
                                f =
arr[i];
                                if
(f.tick)

     f.tick();
                        }
                  }
            }
            if (api ===
API_WEBAUDIO &&
this.listenerTracker.hasObject(
))
            {

     this.listenerTracker.tick(
dt);
                  listenerX =
this.listenerTracker.obj.x;
                  listenerY =
this.listenerTracker.obj.y;

     context["listener"]["setPo
sition"](this.listenerTracker.o
bj.x,
this.listenerTracker.obj.y,
this.listenerZ);

     context["listener"]["setVe
locity"](this.listenerTracker.g
etVelocityX(),
this.listenerTracker.getVelocit
yY(), 0);
            }
      };
      instanceProto.getAudioBuff
er = function (src_, is_music)
      {
            var i, len, a, ret =
null, j, k, lenj, ai;
            for (i = 0, len =
audioBuffers.length; i < len;
i++)
            {
                  a =
audioBuffers[i];
                  if (a.src ===
src_)
                  {
                        ret = a;
                        break;
                  }
            }
            if (!ret)
            {
                  ret = new
C2AudioBuffer(src_, is_music);

      audioBuffers.push(ret);
            }
            return ret;
      };
      instanceProto.getAudioInst
ance = function (src_, tag,
is_music, looping, vol)
      {
            var i, len, a;
            for (i = 0, len =
audioInstances.length; i < len;
i++)
            {
                  a =
audioInstances[i];
                  if (a.src ===
src_ && (a.canBeRecycled() ||
is_music))
                  {
                        a.tag =
tag;
                        return a;
                  }
            }
            var b =
this.getAudioBuffer(src_,
is_music);
            if (!b.bufferObject)
            {
                  if (tag !==
"<preload>")
                  {

      b.playTagWhenReady = tag;

      b.loopWhenReady = looping;

      b.volumeWhenReady = vol;
                  }
                  return null;
            }
            a = new
C2AudioInstance(b, tag);

      audioInstances.push(a);
            return a;
      };
      var taggedAudio = [];
      function
getAudioByTag(tag)
      {
            taggedAudio.length =
0;
```

```
            if (!tag.length)
            {
                  if (!lastAudio
|| lastAudio.hasEnded())
                        return;
                  else
                  {

      taggedAudio.length = 1;

      taggedAudio[0] =
lastAudio;
                        return;
                  }
            }
            var i, len, a;
            for (i = 0, len =
audioInstances.length; i < len;
i++)
            {
                  a =
audioInstances[i];
                  if
(cr.equals_nocase(tag, a.tag))

      taggedAudio.push(a);
            }
      };
      function
reconnectEffects(tag)
      {
            var i, len, arr, n,
toNode =
context["destination"];
            if
(effects.hasOwnProperty(tag))
            {
                  arr =
effects[tag];
                  if (arr.length)
                  {
                        toNode =
arr[0].getInputNode();
                        for (i =
0, len = arr.length; i < len;
i++)
                        {
                              n =
arr[i];
                              if
(i + 1 === len)

      n.connectTo(context["desti
nation"]);
                              else

      n.connectTo(arr[i +
1].getInputNode());
                        }
                  }
            }
            getAudioByTag(tag);
            for (i = 0, len =
taggedAudio.length; i < len;
i++)

      taggedAudio[i].reconnect(t
oNode);
            if (micSource &&
micTag === tag)
            {

      micSource["disconnect"]();

      micSource["connect"](toNod
e);
            }
      };
      function
addEffectForTag(tag, fx)
      {
            if
(!effects.hasOwnProperty(tag))
                  effects[tag] =
[fx];
            else

      effects[tag].push(fx);

      reconnectEffects(tag);
      };
      function Cnds() {};
      Cnds.prototype.OnEnded =
function (t)
      {
            return
cr.equals_nocase(audTag, t);
      };
      Cnds.prototype.PreloadsCom
plete = function ()
      {
            var i, len;
            for (i = 0, len =
audioBuffers.length; i < len;
i++)
            {
                  if
(!audioBuffers[i].isLoaded())
                        return
false;
            }
```

```
		return true;
	};
	Cnds.prototype.AdvancedAud
ioSupported = function ()
	{
		return api ===
API_WEBAUDIO;
	};
	Cnds.prototype.IsSilent =
function ()
	{
		return silent;
	};
	Cnds.prototype.IsAnyPlayin
g = function ()
	{
		var i, len;
		for (i = 0, len =
audioInstances.length; i < len;
i++)
		{
			if
(audioInstances[i].isPlaying())
				return
true;
		}
		return false;
	};
	Cnds.prototype.IsTagPlayin
g = function (tag)
	{
		getAudioByTag(tag);
		var i, len;
		for (i = 0, len =
taggedAudio.length; i < len;
i++)
		{
			if
(taggedAudio[i].isPlaying())
				return
true;
		}
		return false;
	};
	pluginProto.cnds = new
Cnds();
	function Acts() {};
	Acts.prototype.Play =
function (file, looping, vol,
tag)
	{
		if (silent)
			return;
		var v =
dbToLinear(vol);
		var is_music =
file[1];
		var src =
this.runtime.files_subfolder +
file[0] + (useOgg ? ".ogg" :
".m4a");
		lastAudio =
this.getAudioInstance(src, tag,
is_music, looping!==0, v);
		if (!lastAudio)
			return;

	lastAudio.setPannerEnabled
(false);

	lastAudio.play(looping!==0
, v);
	};
	Acts.prototype.PlayAtPosit
ion = function (file, looping,
vol, x_, y_, angle_,
innerangle_, outerangle_,
outergain_, tag)
	{
		if (silent)
			return;
		var v =
dbToLinear(vol);
		var is_music =
file[1];
		var src =
this.runtime.files_subfolder +
file[0] + (useOgg ? ".ogg" :
".m4a");
		lastAudio =
this.getAudioInstance(src, tag,
is_music, looping!==0, v);
		if (!lastAudio)
		{
			var b =
this.getAudioBuffer(src,
is_music);

	b.panWhenReady.push({ x:
x_, y: y_, a: angle_, ia:
innerangle_, oa: outerangle_,
og: dbToLinear(outergain_),
thistag: tag });
			return;
		}

	lastAudio.setPannerEnabled
(true);
		lastAudio.setPan(x_,
y_, angle_, innerangle_,
```

```
outerangle_,
dbToLinear(outergain_));

	lastAudio.play(looping!==0
, v);
	};
	Acts.prototype.PlayAtObjec
t = function (file, looping,
vol, obj, innerangle,
outerangle, outergain, tag)
	{
		if (silent || !obj)
			return;
		var inst =
obj.getFirstPicked();
		if (!inst)
			return;
		var v =
dbToLinear(vol);
		var is_music =
file[1];
		var src =
this.runtime.files_subfolder +
file[0] + (useOgg ? ".ogg" :
".m4a");
		lastAudio =
this.getAudioInstance(src, tag,
is_music, looping!==0, v);
		if (!lastAudio)
		{
			var b =
this.getAudioBuffer(src,
is_music);

	b.panWhenReady.push({ obj:
inst, ia: innerangle, oa:
outerangle, og:
dbToLinear(outergain), thistag:
tag });
			return;
		}

	lastAudio.setPannerEnabled
(true);
		var px =
cr.rotatePtAround(inst.x,
inst.y, -inst.layer.getAngle(),
listenerX, listenerY, true);
		var py =
cr.rotatePtAround(inst.x,
inst.y, -inst.layer.getAngle(),
listenerX, listenerY, false);
		lastAudio.setPan(px,
py, cr.to_degrees(inst.angle -
inst.layer.getAngle()),
innerangle, outerangle,
dbToLinear(outergain));

	lastAudio.setObject(inst);

	lastAudio.play(looping!==0
, v);
	};
	Acts.prototype.PlayByName
= function (folder, filename,
looping, vol, tag)
	{
		if (silent)
			return;
		var v =
dbToLinear(vol);
		var is_music =
(folder === 1);
		var src =
this.runtime.files_subfolder +
filename.toLowerCase() +
(useOgg ? ".ogg" : ".m4a");
		lastAudio =
this.getAudioInstance(src, tag,
is_music, looping!==0, v);
		if (!lastAudio)
			return;

	lastAudio.setPannerEnabled
(false);

	lastAudio.play(looping!==0
, v);
	};
	Acts.prototype.PlayAtPosit
ionByName = function (folder,
filename, looping, vol, x_, y_,
angle_, innerangle_,
outerangle_, outergain_, tag)
	{
		if (silent)
			return;
		var v =
dbToLinear(vol);
		var is_music =
(folder === 1);
		var src =
this.runtime.files_subfolder +
filename.toLowerCase() +
(useOgg ? ".ogg" : ".m4a");
		lastAudio =
this.getAudioInstance(src, tag,
is_music, looping!==0, v);
		if (!lastAudio)
		{
```

```
                var b =
this.getAudioBuffer(src,
is_music);

     b.panWhenReady.push({ x:
x_, y: y_, a: angle_, ia:
innerangle_, oa: outerangle_,
og: dbToLinear(outergain_),
thistag: tag });
                return;
           }

     lastAudio.setPannerEnabled
(true);
           lastAudio.setPan(x_,
y_, angle_, innerangle_,
outerangle_,
dbToLinear(outergain_));

     lastAudio.play(looping!==0
, v);
     };
     Acts.prototype.PlayAtObjec
tByName = function (folder,
filename, looping, vol, obj,
innerangle, outerangle,
outergain, tag)
     {
           if (silent || !obj)
                return;
           var inst =
obj.getFirstPicked();
           if (!inst)
                return;
           var v =
dbToLinear(vol);
           var is_music =
(folder === 1);
           var src =
this.runtime.files_subfolder +
filename.toLowerCase() +
(useOgg ? ".ogg" : ".m4a");
           lastAudio =
this.getAudioInstance(src, tag,
is_music, looping!==0, v);
           if (!lastAudio)
           {
                var b =
this.getAudioBuffer(src,
is_music);

     b.panWhenReady.push({ obj:
inst, ia: innerangle, oa:
outerangle, og:
dbToLinear(outergain), thistag:
tag });
                return;
           }

     lastAudio.setPannerEnabled
(true);
           var px =
cr.rotatePtAround(inst.x,
inst.y, -inst.layer.getAngle(),
listenerX, listenerY, true);
           var py =
cr.rotatePtAround(inst.x,
inst.y, -inst.layer.getAngle(),
listenerX, listenerY, false);
           lastAudio.setPan(px,
py, cr.to_degrees(inst.angle -
inst.layer.getAngle()),
innerangle, outerangle,
dbToLinear(outergain));

     lastAudio.setObject(inst);

     lastAudio.play(looping!==0
, v);
     };
     Acts.prototype.SetLooping
= function (tag, looping)
     {
           getAudioByTag(tag);
           var i, len;
           for (i = 0, len =
taggedAudio.length; i < len;
i++)

     taggedAudio[i].setLooping(
looping === 0);
     };
     Acts.prototype.SetMuted =
function (tag, muted)
     {
           getAudioByTag(tag);
           var i, len;
           for (i = 0, len =
taggedAudio.length; i < len;
i++)

     taggedAudio[i].setMuted(mu
ted === 0);
     };
     Acts.prototype.SetVolume =
function (tag, vol)
     {
           getAudioByTag(tag);
           var v =
dbToLinear(vol);
           var i, len;
```

```
		for (i = 0, len =
taggedAudio.length; i < len;
i++)
	taggedAudio[i].setVolume(v
);
	};
	Acts.prototype.Preload =
function (file)
	{
		if (silent)
			return;
		var is_music =
file[1];
		var src =
this.runtime.files_subfolder +
file[0] + (useOgg ? ".ogg" :
".m4a");
		if (api ===
API_APPMOBI)
		{
			if
(this.runtime.isDirectCanvas)

	AppMobi["context"]["loadSo
und"](src);
			else

	AppMobi["player"]["loadSou
nd"](src);
			return;
		}
		else if (api ===
API_PHONEGAP)
		{
			return;
		}

	this.getAudioInstance(src,
"<preload>", is_music, false);
	};
	Acts.prototype.PreloadByNa
me = function (folder,
filename)
	{
		if (silent)
			return;
		var is_music =
(folder === 1);
		var src =
this.runtime.files_subfolder +
filename.toLowerCase() +
(useOgg ? ".ogg" : ".m4a");
		if (api ===
API_APPMOBI)
		{
```

```
			if
(this.runtime.isDirectCanvas)

	AppMobi["context"]["loadSo
und"](src);
			else

	AppMobi["player"]["loadSou
nd"](src);
			return;
		}
		else if (api ===
API_PHONEGAP)
		{
			return;
		}

	this.getAudioInstance(src,
"<preload>", is_music, false);
	};
	Acts.prototype.SetPlayback
Rate = function (tag, rate)
	{
		getAudioByTag(tag);
		if (rate < 0.0)
			rate = 0;
		var i, len;
		for (i = 0, len =
taggedAudio.length; i < len;
i++)

	taggedAudio[i].setPlayback
Rate(rate);
	};
	Acts.prototype.Stop =
function (tag)
	{
		getAudioByTag(tag);
		var i, len;
		for (i = 0, len =
taggedAudio.length; i < len;
i++)

	taggedAudio[i].stop();
	};
	Acts.prototype.StopAll =
function ()
	{
		var i, len;
		for (i = 0, len =
audioInstances.length; i < len;
i++)

	audioInstances[i].stop();
	};
```

```
	Acts.prototype.SetPaused = function (tag, state)
	{
		getAudioByTag(tag);
		var i, len;
		for (i = 0, len = taggedAudio.length; i < len; i++)
		{
			if (state === 0)
	taggedAudio[i].pause();
			else
	taggedAudio[i].resume();
		}
	};
	Acts.prototype.Seek = function (tag, pos)
	{
		getAudioByTag(tag);
		var i, len;
		for (i = 0, len = taggedAudio.length; i < len; i++)
		{
	taggedAudio[i].seek(pos);
		}
	};
	Acts.prototype.SetSilent = function (s)
	{
		var i, len;
		if (s === 2)		// toggling
			s = (silent ? 1 : 0);	// choose opposite state
		if (s === 0 && !silent)		// setting silent
		{
			for (i = 0, len = audioInstances.length; i < len; i++)
	audioInstances[i].setSilent(true);
			silent = true;
		}
		else if (s === 1 && silent)	// setting not silent
		{
			for (i = 0, len = audioInstances.length; i < len; i++)
	audioInstances[i].setSilent(false);
			silent = false;
		}
	};
	Acts.prototype.SetMasterVolume = function (vol)
	{
		masterVolume = dbToLinear(vol);
		var i, len;
		for (i = 0, len = audioInstances.length; i < len; i++)
	audioInstances[i].updateVolume();
	};
	Acts.prototype.AddFilterEffect = function (tag, type, freq, detune, q, gain, mix)
	{
		if (api !== API_WEBAUDIO || type < 0 || type >= filterTypes.length)
			return;
		tag = tag.toLowerCase();
		mix = mix / 100;
		if (mix < 0) mix = 0;
		if (mix > 1) mix = 1;
		addEffectForTag(tag, new FilterEffect(type, freq, detune, q, gain, mix));
	};
	Acts.prototype.AddDelayEffect = function (tag, delay, gain, mix)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		mix = mix / 100;
		if (mix < 0) mix = 0;
		if (mix > 1) mix = 1;
```

```
		addEffectForTag(tag, new DelayEffect(delay, dbToLinear(gain), mix));
	};
	Acts.prototype.AddFlangerEffect = function (tag, delay, modulation, freq, feedback, mix)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		mix = mix / 100;
		if (mix < 0) mix = 0;
		if (mix > 1) mix = 1;
		addEffectForTag(tag, new FlangerEffect(delay / 1000, modulation / 1000, freq, feedback / 100, mix));
	};
	Acts.prototype.AddPhaserEffect = function (tag, freq, detune, q, mod, modfreq, mix)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		mix = mix / 100;
		if (mix < 0) mix = 0;
		if (mix > 1) mix = 1;
		addEffectForTag(tag, new PhaserEffect(freq, detune, q, mod, modfreq, mix));
	};
	Acts.prototype.AddConvolutionEffect = function (tag, file, norm, mix)
	{
		if (api !== API_WEBAUDIO)
			return;
		var doNormalize = (norm === 0);
		var src = this.runtime.files_subfolder + file[0] + (useOgg ? ".ogg" : ".m4a");
		var b = this.getAudioBuffer(src, false);
		tag = tag.toLowerCase();
		mix = mix / 100;
		if (mix < 0) mix = 0;
		if (mix > 1) mix = 1;
		var fx;
		if (b.bufferObject)
		{
			fx = new ConvolveEffect(b.bufferObject, doNormalize, mix, src);
		}
		else
		{
			fx = new ConvolveEffect(null, doNormalize, mix, src);

	b.normalizeWhenReady = doNormalize;

	b.convolveWhenReady = fx;
		}
		addEffectForTag(tag, fx);
	};
	Acts.prototype.AddGainEffect = function (tag, g)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		addEffectForTag(tag, new GainEffect(dbToLinear(g)));
	};
	Acts.prototype.AddMuteEffect = function (tag)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		addEffectForTag(tag, new GainEffect(0));	// re-use gain effect with 0 gain
	};
```

```
	Acts.prototype.AddTremoloEffect = function (tag, freq, mix)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		mix = mix / 100;
		if (mix < 0) mix = 0;
		if (mix > 1) mix = 1;
		addEffectForTag(tag, new TremoloEffect(freq, mix));
	};
	Acts.prototype.AddRingModEffect = function (tag, freq, mix)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		mix = mix / 100;
		if (mix < 0) mix = 0;
		if (mix > 1) mix = 1;
		addEffectForTag(tag, new RingModulatorEffect(freq, mix));
	};
	Acts.prototype.AddDistortionEffect = function (tag, threshold, headroom, drive, makeupgain, mix)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		mix = mix / 100;
		if (mix < 0) mix = 0;
		if (mix > 1) mix = 1;
		addEffectForTag(tag, new DistortionEffect(threshold, headroom, drive, makeupgain, mix));
	};
	Acts.prototype.AddCompressorEffect = function (tag, threshold, knee, ratio, attack, release)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		addEffectForTag(tag, new CompressorEffect(threshold, knee, ratio, attack / 1000, release / 1000));
	};
	Acts.prototype.AddAnalyserEffect = function (tag, fftSize, smoothing)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		addEffectForTag(tag, new AnalyserEffect(fftSize, smoothing));
	};
	Acts.prototype.RemoveEffects = function (tag)
	{
		if (api !== API_WEBAUDIO)
			return;
		tag = tag.toLowerCase();
		var i, len, arr;
		if (effects.hasOwnProperty(tag))
		{
			arr = effects[tag];
			if (arr.length)
			{
				for (i = 0, len = arr.length; i < len; i++)

	arr[i].remove();

	arr.length = 0;

	reconnectEffects(tag);
			}
		}
	};
```

```
	Acts.prototype.SetEffectPa
rameter = function (tag, index,
param, value, ramp, time)
	{
		if (api !==
API_WEBAUDIO)
			return;
		tag =
tag.toLowerCase();
		index =
Math.floor(index);
		var arr;
		if
(!effects.hasOwnProperty(tag))
			return;
		arr = effects[tag];
		if (index < 0 ||
index >= arr.length)
			return;

	arr[index].setParam(param,
value, ramp, time);
	};
	Acts.prototype.SetListener
Object = function (obj_)
	{
		if (!obj_ || api !==
API_WEBAUDIO)
			return;
		var inst =
obj_.getFirstPicked();
		if (!inst)
			return;

	this.listenerTracker.setOb
ject(inst);
		listenerX = inst.x;
		listenerY = inst.y;
	};
	Acts.prototype.SetListener
Z = function (z)
	{
		this.listenerZ = z;
	};
	pluginProto.acts = new
Acts();
	function Exps() {};
	Exps.prototype.Duration =
function (ret, tag)
	{
		getAudioByTag(tag);
		if
(taggedAudio.length)

	ret.set_float(taggedAudio[
0].getDuration());
		else

	ret.set_float(0);
	};
	Exps.prototype.PlaybackTim
e = function (ret, tag)
	{
		getAudioByTag(tag);
		if
(taggedAudio.length)

	ret.set_float(taggedAudio[
0].getPlaybackTime());
		else

	ret.set_float(0);
	};
	Exps.prototype.Volume =
function (ret, tag)
	{
		getAudioByTag(tag);
		if
(taggedAudio.length)
		{
			var v =
taggedAudio[0].getVolume();

	ret.set_float(linearToDb(v
));
		}
		else

	ret.set_float(0);
	};
	Exps.prototype.MasterVolum
e = function (ret)
	{

	ret.set_float(masterVolume
);
	};
	Exps.prototype.EffectCount
= function (ret, tag)
	{
		tag =
tag.toLowerCase();
		var arr = null;
		if
(effects.hasOwnProperty(tag))
			arr =
effects[tag];
		ret.set_int(arr ?
arr.length : 0);
	};
	function getAnalyser(tag,
index)
```

```
	{
		var arr = null;
		if
(effects.hasOwnProperty(tag))
			arr =
effects[tag];
		if (arr && index >=
0 && index < arr.length &&
arr[index].freqBins)
			return
arr[index];
		else
			return null;
	};
	Exps.prototype.AnalyserFre
qBinCount = function (ret, tag,
index)
	{
		tag =
tag.toLowerCase();
		index =
Math.floor(index);
		var analyser =
getAnalyser(tag, index);
		ret.set_int(analyser
?
analyser.node["frequencyBinCoun
t"] : 0);
	};
	Exps.prototype.AnalyserFre
qBinAt = function (ret, tag,
index, bin)
	{
		tag =
tag.toLowerCase();
		index =
Math.floor(index);
		bin =
Math.floor(bin);
		var analyser =
getAnalyser(tag, index);
		if (!analyser)

	ret.set_float(0);
		else if (bin < 0 ||
bin >=
analyser.node["frequencyBinCoun
t"])

	ret.set_float(0);
		else

	ret.set_float(analyser.fre
qBins[bin]);
	};
	Exps.prototype.AnalyserPea
kLevel = function (ret, tag,
index)
	{
		tag =
tag.toLowerCase();
		index =
Math.floor(index);
		var analyser =
getAnalyser(tag, index);
		if (analyser)

	ret.set_float(analyser.pea
k);
		else

	ret.set_float(0);
	};
	Exps.prototype.AnalyserRMS
Level = function (ret, tag,
index)
	{
		tag =
tag.toLowerCase();
		index =
Math.floor(index);
		var analyser =
getAnalyser(tag, index);
		if (analyser)

	ret.set_float(analyser.rms
);
		else

	ret.set_float(0);
	};
	pluginProto.exps = new
Exps();
}());
;
;
cr.plugins_.Sprite =
function(runtime)
{
	this.runtime = runtime;
};
(function ()
{
	var pluginProto =
cr.plugins_.Sprite.prototype;
	pluginProto.Type =
function(plugin)
	{
		this.plugin =
plugin;
```

```
		this.runtime =
plugin.runtime;
	};
	var typeProto =
pluginProto.Type.prototype;
	function
frame_getDataUri()
	{
		if
(this.datauri.length === 0)
		{
			var tmpcanvas =
document.createElement("canvas"
);
			tmpcanvas.width
= this.width;

	tmpcanvas.height =
this.height;
			var tmpctx =
tmpcanvas.getContext("2d");
			if
(this.spritesheeted)
			{

	tmpctx.drawImage(this.text
ure_img, this.offx, this.offy,
this.width, this.height,

						0,
0, this.width, this.height);
			}
			else
			{

	tmpctx.drawImage(this.text
ure_img, 0, 0, this.width,
this.height);
			}
			this.datauri =
tmpcanvas.toDataURL("image/png"
);
		}
		return this.datauri;
	};
	typeProto.onCreate =
function()
	{
		if (this.is_family)
			return;
		var i, leni, j,
lenj;
		var anim, frame,
animobj, frameobj, wt, uv;
		this.all_frames =
[];

	this.has_loaded_textures =
false;
		for (i = 0, leni =
this.animations.length; i <
leni; i++)
		{
			anim =
this.animations[i];
			animobj = {};
			animobj.name =
anim[0];
			animobj.speed =
anim[1];
			animobj.loop =
anim[2];

	animobj.repeatcount =
anim[3];

	animobj.repeatto =
anim[4];

	animobj.pingpong =
anim[5];
			animobj.sid =
anim[6];
			animobj.frames
= [];
			for (j = 0,
lenj = anim[7].length; j <
lenj; j++)
			{
				frame =
anim[7][j];
				frameobj
= {};

	frameobj.texture_file =
frame[0];

	frameobj.texture_filesize
= frame[1];

	frameobj.offx = frame[2];

	frameobj.offy = frame[3];

	frameobj.width = frame[4];

	frameobj.height =
frame[5];

	frameobj.duration =
frame[6];
```

```
	frameobj.hotspotX =
frame[7];

	frameobj.hotspotY =
frame[8];

	frameobj.image_points =
frame[9];

	frameobj.poly_pts =
frame[10];

	frameobj.pixelformat =
frame[11];

	frameobj.spritesheeted =
(frameobj.width !== 0);

	frameobj.datauri = "";
		// generated on
demand and cached

	frameobj.getDataUri =
frame_getDataUri;
				uv = {};
				uv.left =
0;
				uv.top =
0;
				uv.right
= 1;
				uv.bottom
= 1;

	frameobj.sheetTex = uv;

	frameobj.webGL_texture =
null;
				wt =
this.runtime.findWaitingTexture
(frame[0]);
				if (wt)
				{

	frameobj.texture_img = wt;
				}
				else
				{

	frameobj.texture_img = new
Image();

	frameobj.texture_img["idtk
LoadDisposed"] = true;

	frameobj.texture_img.src =
frame[0];

	frameobj.texture_img.cr_sr
c = frame[0];

	frameobj.texture_img.cr_fi
lesize = frame[1];

	frameobj.texture_img.c2web
GL_texture = null;

	this.runtime.wait_for_text
ures.push(frameobj.texture_img)
;
				}

	cr.seal(frameobj);

	animobj.frames.push(frameo
bj);

	this.all_frames.push(frame
obj);
			}

	cr.seal(animobj);

	this.animations[i] =
animobj;		// swap array
data for object
		}
	};
	typeProto.updateAllCurrent
Texture = function ()
	{
		var i, len, inst;
		for (i = 0, len =
this.instances.length; i < len;
i++)
		{
			inst =
this.instances[i];

	inst.curWebGLTexture =
inst.curFrame.webGL_texture;
		}
	};
	typeProto.onLostWebGLConte
xt = function ()
	{
		if (this.is_family)
			return;
		var i, len, frame;
```

```
		for (i = 0, len = this.all_frames.length; i < len; ++i)
		{
			frame = this.all_frames[i];

	frame.texture_img.c2webGL_texture = null;

	frame.webGL_texture = null;
		}
	};
	typeProto.onRestoreWebGLContext = function ()
	{
		if (this.is_family || !this.instances.length)
			return;
		var i, len, frame;
		for (i = 0, len = this.all_frames.length; i < len; ++i)
		{
			frame = this.all_frames[i];

	frame.webGL_texture = this.runtime.glwrap.loadTexture(frame.texture_img, false, this.runtime.linearSampling, frame.pixelformat);
		}

	this.updateAllCurrentTexture();
	};
	typeProto.loadTextures = function ()
	{
		if (this.is_family || this.has_loaded_textures || !this.runtime.glwrap)
			return;
		var i, len, frame;
		for (i = 0, len = this.all_frames.length; i < len; ++i)
		{
			frame = this.all_frames[i];

	frame.webGL_texture = this.runtime.glwrap.loadTexture(frame.texture_img, false, this.runtime.linearSampling, frame.pixelformat);
		}

	this.has_loaded_textures = true;
	};
	typeProto.unloadTextures = function ()
	{
		if (this.is_family || this.instances.length || !this.has_loaded_textures)
			return;
		var i, len, frame;
		for (i = 0, len = this.all_frames.length; i < len; ++i)
		{
			frame = this.all_frames[i];

	this.runtime.glwrap.deleteTexture(frame.webGL_texture);
		}

	this.has_loaded_textures = false;
	};
	var already_drawn_images = [];
	typeProto.preloadCanvas2D = function (ctx)
	{
		var i, len, frameimg;

	already_drawn_images.length = 0;
		for (i = 0, len = this.all_frames.length; i < len; ++i)
		{
			frameimg = this.all_frames[i].texture_img;
			if (already_drawn_images.indexOf(frameimg) !== -1)

	continue;

	ctx.drawImage(frameimg, 0, 0);

	already_drawn_images.push(frameimg);
```

```
            }
      };
      pluginProto.Instance =
function(type)
      {
            this.type = type;
            this.runtime =
type.runtime;
            var poly_pts =
this.type.animations[0].frames[
0].poly_pts;
            if (this.recycled)

      this.collision_poly.set_pt
s(poly_pts);
            else

      this.collision_poly = new
cr.CollisionPoly(poly_pts);
      };
      var instanceProto =
pluginProto.Instance.prototype;
      instanceProto.onCreate =
function()
      {
            this.visible =
(this.properties[0] === 0); //
0=visible, 1=invisible
            this.isTicking =
false;
            this.inAnimTrigger =
false;

      this.collisionsEnabled =
(this.properties[3] !== 0);
            if
(!(this.type.animations.length
=== 1 &&
this.type.animations[0].frames.
length === 1) &&
this.type.animations[0].speed
!== 0)
            {

      this.runtime.tickMe(this);
                  this.isTicking
= true;
            }
            this.cur_animation =
this.getAnimationByName(this.pr
operties[1]) ||
this.type.animations[0];
            this.cur_frame =
this.properties[2];
            if (this.cur_frame <
0)
                  this.cur_frame
= 0;
            if (this.cur_frame
>=
this.cur_animation.frames.lengt
h)
                  this.cur_frame
=
this.cur_animation.frames.lengt
h - 1;
            var curanimframe =
this.cur_animation.frames[this.
cur_frame];

      this.collision_poly.set_pt
s(curanimframe.poly_pts);
            this.hotspotX =
curanimframe.hotspotX;
            this.hotspotY =
curanimframe.hotspotY;
            this.cur_anim_speed
= this.cur_animation.speed;
            if (this.recycled)

      this.animTimer.reset();
            else
                  this.animTimer
= new cr.KahanAdder();
            this.frameStart =
this.getNowTime();
            this.animPlaying =
true;
            this.animRepeats =
0;
            this.animForwards =
true;
            this.animTriggerName
= "";
            this.changeAnimName
= "";
            this.changeAnimFrom
= 0;
            this.changeAnimFrame
= -1;

      this.type.loadTextures();
            var i, leni, j,
lenj;
            var anim, frame, uv,
maintex;
            for (i = 0, leni =
this.type.animations.length; i
< leni; i++)
            {
                  anim =
this.type.animations[i];
```

```
                for (j = 0,
lenj = anim.frames.length; j <
lenj; j++)
                {
                      frame =
anim.frames[j];
                      if
(frame.width === 0)
                      {

     frame.width =
frame.texture_img.width;

     frame.height =
frame.texture_img.height;
                      }
                      if
(frame.spritesheeted)
                      {

     maintex =
frame.texture_img;
                            uv =
frame.sheetTex;

     uv.left = frame.offx /
maintex.width;

     uv.top = frame.offy /
maintex.height;

     uv.right = (frame.offx +
frame.width) / maintex.width;

     uv.bottom = (frame.offy +
frame.height) / maintex.height;
                            if
(frame.offx === 0 && frame.offy
=== 0 && frame.width ===
maintex.width && frame.height
=== maintex.height)
                            {

     frame.spritesheeted =
false;
                            }
                      }
                }
           }
           this.curFrame =
this.cur_animation.frames[this.
cur_frame];
           this.curWebGLTexture
= this.curFrame.webGL_texture;
     };
```

```
     instanceProto.saveToJSON =
function ()
     {
           var o = {
                 "a":
this.cur_animation.sid,
                 "f":
this.cur_frame,
                 "cas":
this.cur_anim_speed,
                 "fs":
this.frameStart,
                 "ar":
this.animRepeats,
                 "at":
this.animTimer.sum
           };
           if
(!this.animPlaying)
                 o["ap"] =
this.animPlaying;
           if
(!this.animForwards)
                 o["af"] =
this.animForwards;
           return o;
     };
     instanceProto.loadFromJSON
= function (o)
     {
           var anim =
this.getAnimationBySid(o["a"]);
           if (anim)

     this.cur_animation = anim;
           this.cur_frame =
o["f"];
           if (this.cur_frame <
0)
                 this.cur_frame
= 0;
           if (this.cur_frame
>=
this.cur_animation.frames.lengt
h)
                 this.cur_frame
=
this.cur_animation.frames.lengt
h - 1;
           this.cur_anim_speed
= o["cas"];
           this.frameStart =
o["fs"];
           this.animRepeats =
o["ar"];
```

```
	this.animTimer.reset();
		this.animTimer.sum = o["at"];
		this.animPlaying = o.hasOwnProperty("ap") ? o["ap"] : true;
		this.animForwards = o.hasOwnProperty("af") ? o["af"] : true;
		this.curFrame = this.cur_animation.frames[this.cur_frame];
		this.curWebGLTexture = this.curFrame.webGL_texture;

	this.collision_poly.set_pts(this.curFrame.poly_pts);
		this.hotspotX = this.curFrame.hotspotX;
		this.hotspotY = this.curFrame.hotspotY;
	};
	instanceProto.animationFinish = function (reverse)
	{
		this.cur_frame = reverse ? 0 : this.cur_animation.frames.length - 1;
		this.animPlaying = false;
		this.animTriggerName = this.cur_animation.name;
		this.inAnimTrigger = true;

	this.runtime.trigger(cr.plugins_.Sprite.prototype.cnds.OnAnyAnimFinished, this);

	this.runtime.trigger(cr.plugins_.Sprite.prototype.cnds.OnAnimFinished, this);
		this.inAnimTrigger = false;
		this.animRepeats = 0;
	};
	instanceProto.getNowTime = function()
	{
		return this.animTimer.sum;
	};
	instanceProto.tick = function()
	{

	this.animTimer.add(this.runtime.getDt(this));
		if (this.changeAnimName.length)

	this.doChangeAnim();
		if (this.changeAnimFrame >= 0)

	this.doChangeAnimFrame();
		var now = this.getNowTime();
		var cur_animation = this.cur_animation;
		var prev_frame = cur_animation.frames[this.cur_frame];
		var next_frame;
		var cur_frame_time = prev_frame.duration / this.cur_anim_speed;
		if (this.animPlaying && now >= this.frameStart + cur_frame_time)
		{
			if (this.animForwards)
			{

	this.cur_frame++;
			}
			else
			{

	this.cur_frame--;
			}
			this.frameStart += cur_frame_time;
			if (this.cur_frame >= cur_animation.frames.length)
			{
				if (cur_animation.pingpong)
				{

	this.animForwards = false;

	this.cur_frame = cur_animation.frames.length - 2;
				}
```

```
                    else if
(cur_animation.loop)
                    {

     this.cur_frame =
cur_animation.repeatto;
                    }
                    else
                    {

     this.animRepeats++;
                         if
(this.animRepeats >=
cur_animation.repeatcount)
                         {

     this.animationFinish(false
);
                         }
                         else
                         {

     this.cur_frame =
cur_animation.repeatto;
                         }
                    }
               }
               if
(this.cur_frame < 0)
               {
                    if
(cur_animation.pingpong)
                    {

     this.cur_frame = 1;

     this.animForwards = true;
                         if
(!cur_animation.loop)
                         {

     this.animRepeats++;

     if (this.animRepeats >=
cur_animation.repeatcount)

     {


     this.animationFinish(true)
;

     }
                         }
                    }
                    else
```

```
                    {
                         if
(cur_animation.loop)
                         {

     this.cur_frame =
cur_animation.repeatto;
                         }
                         else
                         {

     this.animRepeats++;

     if (this.animRepeats >=
cur_animation.repeatcount)

     {


     this.animationFinish(true)
;

     }

     else

     {

          this.cur_frame =
cur_animation.repeatto;

     }
                         }
                    }
               }
               if
(this.cur_frame < 0)

     this.cur_frame = 0;
               else if
(this.cur_frame >=
cur_animation.frames.length)

     this.cur_frame =
cur_animation.frames.length -
1;
               if (now >
this.frameStart +
(cur_animation.frames[this.cur_
frame].duration /
this.cur_anim_speed))
               {

     this.frameStart = now;
               }
```

```
                next_frame =
cur_animation.frames[this.cur_f
rame];

     this.OnFrameChanged(prev_f
rame, next_frame);

     this.runtime.redraw =
true;
           }
     };
     instanceProto.getAnimation
ByName = function (name_)
     {
           var i, len, a;
           for (i = 0, len =
this.type.animations.length; i
< len; i++)
           {
                a =
this.type.animations[i];
                if
(cr.equals_nocase(a.name,
name_))
                     return a;
           }
           return null;
     };
     instanceProto.getAnimation
BySid = function (sid_)
     {
           var i, len, a;
           for (i = 0, len =
this.type.animations.length; i
< len; i++)
           {
                a =
this.type.animations[i];
                if (a.sid ===
sid_)
                     return a;
           }
           return null;
     };
     instanceProto.doChangeAnim
= function ()
     {
           var prev_frame =
this.cur_animation.frames[this.
cur_frame];
           var anim =
this.getAnimationByName(this.ch
angeAnimName);
           this.changeAnimName
= "";
           if (!anim)
                return;
           if
(cr.equals_nocase(anim.name,
this.cur_animation.name) &&
this.animPlaying)
                return;
           this.cur_animation =
anim;
           this.cur_anim_speed
= anim.speed;
           if (this.cur_frame <
0)
                this.cur_frame
= 0;
           if (this.cur_frame
>=
this.cur_animation.frames.lengt
h)
                this.cur_frame
=
this.cur_animation.frames.lengt
h - 1;
           if
(this.changeAnimFrom === 1)
                this.cur_frame
= 0;
           this.animPlaying =
true;
           this.frameStart =
this.getNowTime();
           this.animForwards =
true;

     this.OnFrameChanged(prev_f
rame,
this.cur_animation.frames[this.
cur_frame]);
           this.runtime.redraw
= true;
     };
     instanceProto.doChangeAnim
Frame = function ()
     {
           var prev_frame =
this.cur_animation.frames[this.
cur_frame];
           var
prev_frame_number =
this.cur_frame;
           this.cur_frame =
cr.floor(this.changeAnimFrame);
           if (this.cur_frame <
0)
                this.cur_frame
= 0;
```

```
            if (this.cur_frame
>=
this.cur_animation.frames.lengt
h)
                  this.cur_frame
=
this.cur_animation.frames.lengt
h - 1;
            if
(prev_frame_number !==
this.cur_frame)
            {

      this.OnFrameChanged(prev_f
rame,
this.cur_animation.frames[this.
cur_frame]);
                  this.frameStart
= this.getNowTime();

      this.runtime.redraw =
true;
            }
            this.changeAnimFrame
= -1;
      };
      instanceProto.OnFrameChang
ed = function (prev_frame,
next_frame)
      {
            var oldw =
prev_frame.width;
            var oldh =
prev_frame.height;
            var neww =
next_frame.width;
            var newh =
next_frame.height;
            if (oldw != neww)
                  this.width *=
(neww / oldw);
            if (oldh != newh)
                  this.height *=
(newh / oldh);
            this.hotspotX =
next_frame.hotspotX;
            this.hotspotY =
next_frame.hotspotY;

      this.collision_poly.set_pt
s(next_frame.poly_pts);

      this.set_bbox_changed();
            this.curFrame =
next_frame;
            this.curWebGLTexture
= next_frame.webGL_texture;
            var i, len, b;
            for (i = 0, len =
this.behavior_insts.length; i <
len; i++)
            {
                  b =
this.behavior_insts[i];
                  if
(b.onSpriteFrameChanged)

      b.onSpriteFrameChanged(pre
v_frame, next_frame);
            }

      this.runtime.trigger(cr.pl
ugins_.Sprite.prototype.cnds.On
FrameChanged, this);
      };
      instanceProto.draw =
function(ctx)
      {
            ctx.globalAlpha =
this.opacity;
            var cur_frame =
this.curFrame;
            var spritesheeted =
cur_frame.spritesheeted;
            var cur_image =
cur_frame.texture_img;
            var myx = this.x;
            var myy = this.y;
            var w = this.width;
            var h = this.height;
            if (this.angle === 0
&& w >= 0 && h >= 0)
            {
                  myx -=
this.hotspotX * w;
                  myy -=
this.hotspotY * h;
                  if
(this.runtime.pixel_rounding)
                  {
                        myx =
(myx + 0.5) | 0;
                        myy =
(myy + 0.5) | 0;
                  }
                  if
(spritesheeted)
                  {

      ctx.drawImage(cur_image,
cur_frame.offx, cur_frame.offy,
```

```
cur_frame.width,
cur_frame.height,

myx, myy, w, h);
                }
                else
                {

     ctx.drawImage(cur_image,
myx, myy, w, h);
                }
           }
           else
           {
                if
(this.runtime.pixel_rounding)
                {
                     myx =
(myx + 0.5) | 0;
                     myy =
(myy + 0.5) | 0;
                }
                ctx.save();
                var widthfactor
= w > 0 ? 1 : -1;
                var
heightfactor = h > 0 ? 1 : -1;

     ctx.translate(myx, myy);
                if (widthfactor
!== 1 || heightfactor !== 1)

     ctx.scale(widthfactor,
heightfactor);

     ctx.rotate(this.angle *
widthfactor * heightfactor);
                var drawx = 0 -
(this.hotspotX * cr.abs(w))
                var drawy = 0 -
(this.hotspotY * cr.abs(h));
                if
(spritesheeted)
                {

     ctx.drawImage(cur_image,
cur_frame.offx, cur_frame.offy,
cur_frame.width,
cur_frame.height,

drawx, drawy, cr.abs(w),
cr.abs(h));
                }
                else
                {

     ctx.drawImage(cur_image,
drawx, drawy, cr.abs(w),
cr.abs(h));
                }
                ctx.restore();
           }
           /*
           ctx.strokeStyle =
"#f00";
           ctx.lineWidth = 3;
           ctx.beginPath();

     this.collision_poly.cache_
poly(this.width, this.height,
this.angle);
           var i, len, ax, ay,
bx, by;
           for (i = 0, len =
this.collision_poly.pts_count;
i < len; i++)
           {
                ax =
this.collision_poly.pts_cache[i
*2] + this.x;
                ay =
this.collision_poly.pts_cache[i
*2+1] + this.y;
                bx =
this.collision_poly.pts_cache[(
(i+1)%len)*2] + this.x;
                by =
this.collision_poly.pts_cache[(
(i+1)%len)*2+1] + this.y;
                ctx.moveTo(ax,
ay);
                ctx.lineTo(bx,
by);
           }
           ctx.stroke();
           ctx.closePath();
           */
           /*
           if
(this.behavior_insts.length >=
1 &&
this.behavior_insts[0].draw)
           {

     this.behavior_insts[0].dra
w(ctx);
           }
           */
     };
```

```
	instanceProto.drawGL = function(glw)
	{
		glw.setTexture(this.curWebGLTexture);
		glw.setOpacity(this.opacity);
		var cur_frame = this.curFrame;
		var q = this.bquad;
		if (this.runtime.pixel_rounding)
		{
			var ox = ((this.x + 0.5) | 0) - this.x;
			var oy = ((this.y + 0.5) | 0) - this.y;
			if (cur_frame.spritesheeted)
				glw.quadTex(q.tlx + ox, q.tly + oy, q.trx + ox, q.try_ + oy, q.brx + ox, q.bry + oy, q.blx + ox, q.bly + oy, cur_frame.sheetTex);
			else
				glw.quad(q.tlx + ox, q.tly + oy, q.trx + ox, q.try_ + oy, q.brx + ox, q.bry + oy, q.blx + ox, q.bly + oy);
		}
		else
		{
			if (cur_frame.spritesheeted)
				glw.quadTex(q.tlx, q.tly, q.trx, q.try_, q.brx, q.bry, q.blx, q.bly, cur_frame.sheetTex);
			else
				glw.quad(q.tlx, q.tly, q.trx, q.try_, q.brx, q.bry, q.blx, q.bly);
		}
	};
	instanceProto.getImagePointIndexByName = function(name_)
	{
		var cur_frame = this.curFrame;
		var i, len;
		for (i = 0, len = cur_frame.image_points.length; i < len; i++)
		{
			if (cr.equals_nocase(name_, cur_frame.image_points[i][0]))
				return i;
		}
		return -1;
	};
	instanceProto.getImagePoint = function(imgpt, getX)
	{
		var cur_frame = this.curFrame;
		var image_points = cur_frame.image_points;
		var index;
		if (cr.is_string(imgpt))
			index = this.getImagePointIndexByName(imgpt);
		else
			index = imgpt - 1;	// 0 is origin
		index = cr.floor(index);
		if (index < 0 || index >= image_points.length)
			return getX ? this.x : this.y;// return origin
		var x = (image_points[index][1] - cur_frame.hotspotX) * this.width;
		var y = image_points[index][2];
		y = (y - cur_frame.hotspotY) * this.height;
		var cosa = Math.cos(this.angle);
		var sina = Math.sin(this.angle);
		var x_temp = (x * cosa) - (y * sina);
		y = (y * cosa) + (x * sina);
		x = x_temp;
		x += this.x;
		y += this.y;
		return getX ? x : y;
	};
```

```
	function Cnds() {};
	var arrCache = [];
	function allocArr()
	{
		if (arrCache.length)
			return
arrCache.pop();
		else
			return [0, 0,
0];
	};
	function freeArr(a)
	{
		a[0] = 0;
		a[1] = 0;
		a[2] = 0;
		arrCache.push(a);
	};
	function makeCollKey(a, b)
	{
		if (a < b)
			return "" + a +
"," + b;
		else
			return "" + b +
"," + a;
	};
	function
collmemory_add(collmemory, a,
b, tickcount)
	{
		var a_uid = a.uid;
		var b_uid = b.uid;
		var key =
makeCollKey(a_uid, b_uid);
		if
(collmemory.hasOwnProperty(key)
)
		{

	collmemory[key][2] =
tickcount;
			return;
		}
		var arr =
allocArr();
		arr[0] = a_uid;
		arr[1] = b_uid;
		arr[2] = tickcount;
		collmemory[key] =
arr;
	};
	function
collmemory_remove(collmemory,
a, b)
	{
		var key =
makeCollKey(a.uid, b.uid);
		if
(collmemory.hasOwnProperty(key)
)
		{

	freeArr(collmemory[key]);
			delete
collmemory[key];
		}
	};
	function
collmemory_removeInstance(collm
emory, inst)
	{
		var uid = inst.uid;
		var p, entry;
		for (p in
collmemory)
		{
			if
(collmemory.hasOwnProperty(p))
			{
				entry =
collmemory[p];
				if
(entry[0] === uid || entry[1]
=== uid)
				{

	freeArr(collmemory[p]);

	delete collmemory[p];
				}
			}
		}
	};
	var last_coll_tickcount =
-2;
	function
collmemory_has(collmemory, a,
b)
	{
		var key =
makeCollKey(a.uid, b.uid);
		if
(collmemory.hasOwnProperty(key)
)
		{

	last_coll_tickcount =
collmemory[key][2];
			return true;
		}
		else
```

```
		{
	last_coll_tickcount = -2;
			return false;
		}
	};
	var candidates = [];
	Cnds.prototype.OnCollision
= function (rtype)
	{
		if (!rtype)
			return false;
		var runtime =
this.runtime;
		var cnd =
runtime.getCurrentCondition();
		var ltype =
cnd.type;
		if
(!cnd.extra.collmemory)
		{
	cnd.extra.collmemory = {};
	runtime.addDestroyCallback
((function (collmemory) {
				return
function(inst) {
	collmemory_removeInstance(
collmemory, inst);
				};
	})(cnd.extra.collmemory));
		}
		var collmemory =
cnd.extra.collmemory;
		var lsol =
ltype.getCurrentSol();
		var rsol =
rtype.getCurrentSol();
		var linstances =
lsol.getObjects();
		var rinstances;
		var l, linst, r,
rinst;
		var curlsol,
currsol;
		var tickcount =
this.runtime.tickcount;
		var lasttickcount =
tickcount - 1;
		var exists, run;
		var current_event =
runtime.getCurrentEventStack().
current_event;
		var orblock =
current_event.orblock;
		for (l = 0; l <
linstances.length; l++)
		{
			linst =
linstances[l];
			if
(rsol.select_all)
			{
	linst.update_bbox();
	this.runtime.getCollisionC
andidates(linst.layer, rtype,
linst.bbox, candidates);
	rinstances = candidates;
			}
			else
	rinstances =
rsol.getObjects();
			for (r = 0; r <
rinstances.length; r++)
			{
				rinst =
rinstances[r];
				if
(runtime.testOverlap(linst,
rinst) ||
runtime.checkRegisteredCollisio
n(linst, rinst))
				{
	exists =
collmemory_has(collmemory,
linst, rinst);
					run
= (!exists ||
(last_coll_tickcount <
lasttickcount));
	collmemory_add(collmemory,
linst, rinst, tickcount);
					if
(run)
					{
	runtime.pushCopySol(curren
t_event.solModifiers);
	curlsol =
ltype.getCurrentSol();
```

```
currsol =
rtype.getCurrentSol();

curlsol.select_all =
false;

currsol.select_all =
false;

if (ltype === rtype)

{


curlsol.instances.length =
2;   // just use lsol, is same
reference as rsol

curlsol.instances[0]
= linst;

curlsol.instances[1]
= rinst;


ltype.applySolToContainer(
);

}

else

{


curlsol.instances.length =
1;


currsol.instances.length =
1;

curlsol.instances[0]
= linst;

currsol.instances[0]
= rinst;


ltype.applySolToContainer(
);


rtype.applySolToContainer(
);
```

```
}

current_event.retrigger();

runtime.popSol(current_eve
nt.solModifiers);
                    }
                }
                else
                {

collmemory_remove(collmemo
ry, linst, rinst);
                }
            }

candidates.length = 0;
        }
        return false;
    };
    var rpicktype = null;
    var rtopick = new
cr.ObjectSet();
    var needscollisionfinish =
false;
    function
DoOverlapCondition(rtype, offx,
offy)
    {
        if (!rtype)
            return false;
        var do_offset =
(offx !== 0 || offy !== 0);
        var oldx, oldy, ret
= false, r, lenr, rinst;
        var cnd =
this.runtime.getCurrentConditio
n();
        var ltype =
cnd.type;
        var inverted =
cnd.inverted;
        var rsol =
rtype.getCurrentSol();
        var orblock =
this.runtime.getCurrentEventSta
ck().current_event.orblock;
        var rinstances;
        if (rsol.select_all)
        {

this.update_bbox();

this.runtime.getCollisionC
```

```
andidates(this.layer, rtype,
this.bbox, candidates);
                rinstances =
candidates;
           }
           else if (orblock)
                rinstances =
rsol.else_instances;
           else
                rinstances =
rsol.instances;
           rpicktype = rtype;
           needscollisionfinish
= (ltype !== rtype &&
!inverted);
           if (do_offset)
           {
                oldx = this.x;
                oldy = this.y;
                this.x += offx;
                this.y += offy;

     this.set_bbox_changed();
           }
           for (r = 0, lenr =
rinstances.length; r < lenr;
r++)
           {
                rinst =
rinstances[r];
                if
(this.runtime.testOverlap(this,
rinst))
                {
                     ret =
true;
                     if
(inverted)

     break;
                     if (ltype
!== rtype)

     rtopick.add(rinst);
                }
           }
           if (do_offset)
           {
                this.x = oldx;
                this.y = oldy;

     this.set_bbox_changed();
           }
           candidates.length =
0;
           return ret;
     };
     typeProto.finish =
function (do_pick)
     {
          if
(!needscollisionfinish)
               return;
          if (do_pick)
          {
               var orblock =
this.runtime.getCurrentEventSta
ck().current_event.orblock;
               var sol =
rpicktype.getCurrentSol();
               var topick =
rtopick.valuesRef();
               var i, len,
inst;
               if
(sol.select_all)
               {

     sol.select_all = false;

     sol.instances.length =
topick.length;
                    for (i =
0, len = topick.length; i <
len; i++)
                    {

     sol.instances[i] =
topick[i];
                    }
                    if
(orblock)
                    {

     sol.else_instances.length
= 0;
                         for
(i = 0, len =
rpicktype.instances.length; i <
len; i++)
                         {

     inst =
rpicktype.instances[i];

     if
(!rtopick.contains(inst))

     sol.else_instances.push(in
st);
                         }
```

```
                    }
                }
                else
                {
                    if
(orblock)
                    {
                        var
initsize =
sol.instances.length;

    sol.instances.length =
initsize + topick.length;
                        for
(i = 0, len = topick.length; i
< len; i++)
                        {

    sol.instances[initsize +
i] = topick[i];

    cr.arrayFindRemove(sol.els
e_instances, topick[i]);
                        }
                    }
                    else
                    {

    cr.shallowAssignArray(sol.
instances, topick);
                    }
                }

    rpicktype.applySolToContai
ner();
            }
            rtopick.clear();
            needscollisionfinish
= false;
    };
    Cnds.prototype.IsOverlappi
ng = function (rtype)
    {
        return
DoOverlapCondition.call(this,
rtype, 0, 0);
    };
    Cnds.prototype.IsOverlappi
ngOffset = function (rtype,
offx, offy)
    {
        return
DoOverlapCondition.call(this,
rtype, offx, offy);
    };
    Cnds.prototype.IsAnimPlayi
ng = function (animname)
    {
        if
(this.changeAnimName.length)
            return
cr.equals_nocase(this.changeAni
mName, animname);
        else
            return
cr.equals_nocase(this.cur_anima
tion.name, animname);
    };
    Cnds.prototype.CompareFram
e = function (cmp, framenum)
    {
        return
cr.do_cmp(this.cur_frame, cmp,
framenum);
    };
    Cnds.prototype.CompareAnim
Speed = function (cmp, x)
    {
        var s =
(this.animForwards ?
this.cur_anim_speed : -
this.cur_anim_speed);
        return cr.do_cmp(s,
cmp, x);
    };
    Cnds.prototype.OnAnimFinis
hed = function (animname)
    {
        return
cr.equals_nocase(this.animTrigg
erName, animname);
    };
    Cnds.prototype.OnAnyAnimFi
nished = function ()
    {
        return true;
    };
    Cnds.prototype.OnFrameChan
ged = function ()
    {
        return true;
    };
    Cnds.prototype.IsMirrored
= function ()
    {
        return this.width <
0;
    };
    Cnds.prototype.IsFlipped =
function ()
    {
```

```
            return this.height <
0;
      };
      Cnds.prototype.OnURLLoaded
= function ()
      {
            return true;
      };
      Cnds.prototype.IsCollision
Enabled = function ()
      {
            return
this.collisionsEnabled;
      };
      pluginProto.cnds = new
Cnds();
      function Acts() {};
      Acts.prototype.Spawn =
function (obj, layer, imgpt)
      {
            if (!obj || !layer)
                  return;
            var inst =
this.runtime.createInstance(obj
, layer,
this.getImagePoint(imgpt,
true),
this.getImagePoint(imgpt,
false));
            if (!inst)
                  return;
            if (typeof
inst.angle !== "undefined")
            {
                  inst.angle =
this.angle;

      inst.set_bbox_changed();
            }

      this.runtime.isInOnDestroy
++;
            var i, len, s;

      this.runtime.trigger(Objec
t.getPrototypeOf(obj.plugin).cn
ds.OnCreated, inst);
            if
(inst.is_contained)
            {
                  for (i = 0, len
= inst.siblings.length; i <
len; i++)
                  {
                        s =
inst.siblings[i];

      this.runtime.trigger(Objec
t.getPrototypeOf(s.type.plugin)
.cnds.OnCreated, s);
                  }
            }

      this.runtime.isInOnDestroy
--;
            var cur_act =
this.runtime.getCurrentAction()
;
            var reset_sol =
false;
            if
(cr.is_undefined(cur_act.extra.
Spawn_LastExec) ||
cur_act.extra.Spawn_LastExec <
this.runtime.execcount)
            {
                  reset_sol =
true;

      cur_act.extra.Spawn_LastEx
ec = this.runtime.execcount;
            }
            var sol;
            if (obj !=
this.type)
            {
                  sol =
obj.getCurrentSol();
                  sol.select_all
= false;
                  if (reset_sol)
                  {

      sol.instances.length = 1;

      sol.instances[0] = inst;
                  }
                  else

      sol.instances.push(inst);
                  if
(inst.is_contained)
                  {
                        for (i =
0, len = inst.siblings.length;
i < len; i++)
                        {
                              s =
inst.siblings[i];
                              sol
= s.type.getCurrentSol();
```

```
sol.select_all = false;
                        if
(reset_sol)
                        {
sol.instances.length = 1;
sol.instances[0] = s;
                        }
                        else

sol.instances.push(s);
                    }
                }
            }
        };
        Acts.prototype.SetEffect =
function (effect)
        {
            this.compositeOp =
cr.effectToCompositeOp(effect);
            cr.setGLBlend(this,
effect, this.runtime.gl);
            this.runtime.redraw
= true;
        };
        Acts.prototype.StopAnim =
function ()
        {
            this.animPlaying =
false;
        };
        Acts.prototype.StartAnim =
function (from)
        {
            this.animPlaying =
true;
            this.frameStart =
this.getNowTime();
            if (from === 1 &&
this.cur_frame !== 0)
            {
this.changeAnimFrame = 0;
                if
(!this.inAnimTrigger)

this.doChangeAnimFrame();
            }
            if (!this.isTicking)
            {
this.runtime.tickMe(this);
                this.isTicking
= true;
            }
        };
        Acts.prototype.SetAnim =
function (animname, from)
        {
            this.changeAnimName
= animname;
            this.changeAnimFrom
= from;
            if (!this.isTicking)
            {

this.runtime.tickMe(this);
                this.isTicking
= true;
            }
            if
(!this.inAnimTrigger)

this.doChangeAnim();
        };
        Acts.prototype.SetAnimFram
e = function (framenumber)
        {
            this.changeAnimFrame
= framenumber;
            if (!this.isTicking)
            {

this.runtime.tickMe(this);
                this.isTicking
= true;
            }
            if
(!this.inAnimTrigger)

this.doChangeAnimFrame();
        };
        Acts.prototype.SetAnimSpee
d = function (s)
        {
            this.cur_anim_speed
= cr.abs(s);
            this.animForwards =
(s >= 0);
            if (!this.isTicking)
            {

this.runtime.tickMe(this);
                this.isTicking
= true;
            }
        };
        Acts.prototype.SetMirrored
= function (m)
        {
```

```
		var neww =
cr.abs(this.width) * (m === 0 ?
-1 : 1);
		if (this.width ===
neww)
			return;
		this.width = neww;

	this.set_bbox_changed();
	};
	Acts.prototype.SetFlipped
= function (f)
	{
		var newh =
cr.abs(this.height) * (f === 0
? -1 : 1);
		if (this.height ===
newh)
			return;
		this.height = newh;

	this.set_bbox_changed();
	};
	Acts.prototype.SetScale =
function (s)
	{
		var cur_frame =
this.curFrame;
		var mirror_factor =
(this.width < 0 ? -1 : 1);
		var flip_factor =
(this.height < 0 ? -1 : 1);
		var new_width =
cur_frame.width * s *
mirror_factor;
		var new_height =
cur_frame.height * s *
flip_factor;
		if (this.width !==
new_width || this.height !==
new_height)
		{
			this.width =
new_width;
			this.height =
new_height;

	this.set_bbox_changed();
		}
	};
	Acts.prototype.LoadURL =
function (url_, resize_)
	{
		var img = new
Image();
		var self = this;
		var curFrame_ =
this.curFrame;
		img.onload =
function ()
		{
			if
(curFrame_.texture_img.src ===
img.src)
			{
				if
(self.runtime.glwrap &&
self.curFrame === curFrame_)

	self.curWebGLTexture =
curFrame_.webGL_texture;

	self.runtime.redraw =
true;

	self.runtime.trigger(cr.pl
ugins_.Sprite.prototype.cnds.On
URLLoaded, self);
				return;
			}

	curFrame_.texture_img =
img;
			curFrame_.offx
= 0;
			curFrame_.offy
= 0;
			curFrame_.width
= img.width;

	curFrame_.height =
img.height;

	curFrame_.spritesheeted =
false;

	curFrame_.datauri = "";
			if
(self.runtime.glwrap)
			{
				if
(curFrame_.webGL_texture)

	self.runtime.glwrap.delete
Texture(curFrame_.webGL_texture
);

	curFrame_.webGL_texture =
self.runtime.glwrap.loadTexture
(img, false,
self.runtime.linearSampling);
```

```
                    if
(self.curFrame === curFrame_)

     self.curWebGLTexture =
curFrame_.webGL_texture;

     self.type.updateAllCurrent
Texture();
                }
                if (resize_ ===
0)         // resize to image
size
                {

     self.width = img.width;

     self.height = img.height;

     self.set_bbox_changed();
                }

     self.runtime.redraw =
true;

     self.runtime.trigger(cr.pl
ugins_.Sprite.prototype.cnds.On
URLLoaded, self);
           };
           if (url_.substr(0,
5) !== "data:")
                img.crossOrigin
= 'anonymous';
           img.src = url_;
     };
     Acts.prototype.SetCollisio
ns = function (set_)
     {
           if
(this.collisionsEnabled ===
(set_ !== 0))
                return;
     // no change

     this.collisionsEnabled =
(set_ !== 0);
           if
(this.collisionsEnabled)

     this.set_bbox_changed();
           // needs to be added
back to cells
           else
           {
                if
(this.collcells.right >=
this.collcells.left)

     this.type.collision_grid.u
pdate(this, this.collcells,
null);

     this.collcells.set(0, 0, -
1, -1);
           }
     };
     pluginProto.acts = new
Acts();
     function Exps() {};
     Exps.prototype.AnimationFr
ame = function (ret)
     {

     ret.set_int(this.cur_frame
);
     };
     Exps.prototype.AnimationFr
ameCount = function (ret)
     {

     ret.set_int(this.cur_anima
tion.frames.length);
     };
     Exps.prototype.AnimationNa
me = function (ret)
     {

     ret.set_string(this.cur_an
imation.name);
     };
     Exps.prototype.AnimationSp
eed = function (ret)
     {

     ret.set_float(this.animFor
wards ? this.cur_anim_speed : -
this.cur_anim_speed);
     };
     Exps.prototype.ImagePointX
= function (ret, imgpt)
     {

     ret.set_float(this.getImag
ePoint(imgpt, true));
     };
     Exps.prototype.ImagePointY
= function (ret, imgpt)
     {

     ret.set_float(this.getImag
ePoint(imgpt, false));
     };
```

```
	Exps.prototype.ImagePointC
ount = function (ret)
	{

	ret.set_int(this.curFrame.
image_points.length);
	};
	Exps.prototype.ImageWidth
= function (ret)
	{

	ret.set_float(this.curFram
e.width);
	};
	Exps.prototype.ImageHeight
= function (ret)
	{

	ret.set_float(this.curFram
e.height);
	};
	pluginProto.exps = new
Exps();
}());
;
;
cr.plugins_.Text =
function(runtime)
{
	this.runtime = runtime;
};
(function ()
{
	var pluginProto =
cr.plugins_.Text.prototype;
	pluginProto.onCreate =
function ()
	{

	pluginProto.acts.SetWidth
= function (w)
		{
			if (this.width
!== w)
			{

	this.width = w;

	this.text_changed = true;
	// also recalculate text
wrapping

	this.set_bbox_changed();
			}
		};
	};
	pluginProto.Type =
function(plugin)
	{
		this.plugin =
plugin;
		this.runtime =
plugin.runtime;
	};
	var typeProto =
pluginProto.Type.prototype;
	typeProto.onCreate =
function()
	{
	};
	typeProto.onLostWebGLConte
xt = function ()
	{
		if (this.is_family)
			return;
		var i, len, inst;
		for (i = 0, len =
this.instances.length; i < len;
i++)
		{
			inst =
this.instances[i];
			inst.mycanvas =
null;
			inst.myctx =
null;
			inst.mytex =
null;
		}
	};
	pluginProto.Instance =
function(type)
	{
		this.type = type;
		this.runtime =
type.runtime;
		if (this.recycled)

	this.lines.length = 0;
		else
			this.lines =
[];	// for word wrapping
		this.text_changed =
true;
	};
	var instanceProto =
pluginProto.Instance.prototype;
	var requestedWebFonts =
{};	// already requested
web fonts have an entry here
	instanceProto.onCreate =
function()
```

```
	{
		this.text = this.properties[0];
		this.visible = (this.properties[1] === 0);	// 0=visible, 1=invisible
		this.font = this.properties[2];
		this.color = this.properties[3];
		this.halign = this.properties[4];	// 0=left, 1=center, 2=right
		this.valign = this.properties[5];	// 0=top, 1=center, 2=bottom
		this.wrapbyword = (this.properties[7] === 0);	// 0=word, 1=character
		this.lastwidth = this.width;
		this.lastwrapwidth = this.width;
		this.lastheight = this.height;

	this.line_height_offset = this.properties[8];
		this.facename = "";
		this.fontstyle = "";
		this.ptSize = 0;
		this.textWidth = 0;
		this.textHeight = 0;
		this.parseFont();
		this.mycanvas = null;
		this.myctx = null;
		this.mytex = null;

	this.need_text_redraw = false;

	this.last_render_tick = this.runtime.tickcount;
		if (this.recycled)

	this.rcTex.set(0, 0, 1, 1);
		else
			this.rcTex = new cr.rect(0, 0, 1, 1);
		if (this.runtime.glwrap)

	this.runtime.tickMe(this);
;
	};
	instanceProto.parseFont = function ()
	{
		var arr = this.font.split(" ");
		var i;
		for (i = 0; i < arr.length; i++)
		{
			if (arr[i].substr(arr[i].length - 2, 2) === "pt")
			{

	this.ptSize = parseInt(arr[i].substr(0, arr[i].length - 2));

	this.pxHeight = Math.ceil((this.ptSize / 72.0) * 96.0) + 4;	// assume 96dpi...
				if (i > 0)

	this.fontstyle = arr[i - 1];

	this.facename = arr[i + 1];
				for (i = i + 2; i < arr.length; i++)

	this.facename += " " + arr[i];
				break;
			}
		}
	};
	instanceProto.saveToJSON = function ()
	{
		return {
			"t": this.text,
			"f": this.font,
			"c": this.color,
			"ha": this.halign,
			"va": this.valign,
```

```
                "wr":
this.wrapbyword,
                "lho":
this.line_height_offset,
                "fn":
this.facename,
                "fs":
this.fontstyle,
                "ps":
this.ptSize,
                "pxh":
this.pxHeight,
                "tw":
this.textWidth,
                "th":
this.textHeight,
                "lrt":
this.last_render_tick
            };
      };
      instanceProto.loadFromJSON
= function (o)
      {
            this.text = o["t"];
            this.font = o["f"];
            this.color = o["c"];
            this.halign =
o["ha"];
            this.valign =
o["va"];
            this.wrapbyword =
o["wr"];

      this.line_height_offset =
o["lho"];
            this.facename =
o["fn"];
            this.fontstyle =
o["fs"];
            this.ptSize =
o["ps"];
            this.pxHeight =
o["pxh"];
            this.textWidth =
o["tw"];
            this.textHeight =
o["th"];

      this.last_render_tick =
o["lrt"];
            this.text_changed =
true;
            this.lastwidth =
this.width;
            this.lastwrapwidth =
this.width;
            this.lastheight =
this.height;
      };
      instanceProto.tick =
function ()
      {
            if
(this.runtime.glwrap &&
this.mytex &&
(this.runtime.tickcount -
this.last_render_tick >= 300))
            {
                  var layer =
this.layer;
             this.update_bbox();
             var bbox =
this.bbox;
             if (bbox.right <
layer.viewLeft || bbox.bottom <
layer.viewTop || bbox.left >
layer.viewRight || bbox.top >
layer.viewBottom)
                  {

      this.runtime.glwrap.delete
Texture(this.mytex);

      this.mytex = null;

      this.myctx = null;

      this.mycanvas = null;
                  }
            }
      };
      instanceProto.onDestroy =
function ()
      {
            this.myctx = null;
            this.mycanvas =
null;
            if
(this.runtime.glwrap &&
this.mytex)

      this.runtime.glwrap.delete
Texture(this.mytex);
            this.mytex = null;
      };
      instanceProto.updateFont =
function ()
      {
            this.font =
this.fontstyle + " " +
this.ptSize.toString() + "pt "
+ this.facename;
```

```
		this.text_changed =
true;
		this.runtime.redraw
= true;
	};
	instanceProto.draw =
function(ctx, glmode)
	{
		ctx.font =
this.font;
		ctx.textBaseline =
"top";
		ctx.fillStyle =
this.color;
		ctx.globalAlpha =
glmode ? 1 : this.opacity;
		var myscale = 1;
		if (glmode)
		{
			myscale =
this.layer.getScale();
			ctx.save();

	ctx.scale(myscale,
myscale);
		}
		if
(this.text_changed ||
this.width !==
this.lastwrapwidth)
		{

	this.type.plugin.WordWrap(
this.text, this.lines, ctx,
this.width, this.wrapbyword);

	this.text_changed = false;

	this.lastwrapwidth =
this.width;
		}
		this.update_bbox();
		var penX = glmode ?
0 : this.bquad.tlx;
		var penY = glmode ?
0 : this.bquad.tly;
		if
(this.runtime.pixel_rounding)
		{
			penX = (penX +
0.5) | 0;
			penY = (penY +
0.5) | 0;
		}
		if (this.angle !== 0
&& !glmode)
		{
			ctx.save();

	ctx.translate(penX, penY);

	ctx.rotate(this.angle);
			penX = 0;
			penY = 0;
		}
		var endY = penY +
this.height;
		var line_height =
this.pxHeight;
		line_height +=
this.line_height_offset;
		var drawX;
		var i;
		if (this.valign ===
1)		// center
			penY +=
Math.max(this.height / 2 -
(this.lines.length *
line_height) / 2, 0);
		else if (this.valign
=== 2)	// bottom
			penY +=
Math.max(this.height -
(this.lines.length *
line_height) - 2, 0);
		for (i = 0; i <
this.lines.length; i++)
		{
			drawX = penX;
			if (this.halign
=== 1)		// center
				drawX =
penX + (this.width -
this.lines[i].width) / 2;
			else if
(this.halign === 2)	// right
				drawX =
penX + (this.width -
this.lines[i].width);

	ctx.fillText(this.lines[i]
.text, drawX, penY);
			penY +=
line_height;
			if (penY >=
endY - line_height)
				break;
		}
		if (this.angle !== 0
|| glmode)
			ctx.restore();
```

```
	this.last_render_tick = this.runtime.tickcount;
	};
	instanceProto.drawGL = function(glw)
	{
		if (this.width < 1 || this.height < 1)
			return;
		var need_redraw = this.text_changed || this.need_text_redraw;

	this.need_text_redraw = false;
		var layer_scale = this.layer.getScale();
		var layer_angle = this.layer.getAngle();
		var rcTex = this.rcTex;
		var floatscaledwidth = layer_scale * this.width;
		var floatscaledheight = layer_scale * this.height;
		var scaledwidth = Math.ceil(floatscaledwidth);
		var scaledheight = Math.ceil(floatscaledheight);
		var halfw = this.runtime.draw_width / 2;
		var halfh = this.runtime.draw_height / 2;
		if (!this.myctx)
		{
			this.mycanvas = document.createElement("canvas");

	this.mycanvas.width = scaledwidth;

	this.mycanvas.height = scaledheight;
			this.lastwidth = scaledwidth;
			this.lastheight = scaledheight;
			need_redraw = true;
			this.myctx = this.mycanvas.getContext("2d");
		}
		if (scaledwidth !== this.lastwidth || scaledheight !== this.lastheight)
		{

	this.mycanvas.width = scaledwidth;

	this.mycanvas.height = scaledheight;
			if (this.mytex)
			{

	glw.deleteTexture(this.mytex);

	this.mytex = null;
			}
			need_redraw = true;
		}
		if (need_redraw)
		{

	this.myctx.clearRect(0, 0, scaledwidth, scaledheight);

	this.draw(this.myctx, true);
			if (!this.mytex)

	this.mytex = glw.createEmptyTexture(scaledwidth, scaledheight, this.runtime.linearSampling, this.runtime.isMobile);

	glw.videoToTexture(this.mycanvas, this.mytex, this.runtime.isMobile);
		}
		this.lastwidth = scaledwidth;
		this.lastheight = scaledheight;

	glw.setTexture(this.mytex);

	glw.setOpacity(this.opacity);

	glw.resetModelView();
		glw.translate(-halfw, -halfh);
```

```
	glw.updateModelView();
		var q = this.bquad;
		var old_dpr =
this.runtime.devicePixelRatio;

	this.runtime.devicePixelRa
tio = 1;
		var tlx =
this.layer.layerToCanvas(q.tlx,
q.tly, true, true);
		var tly =
this.layer.layerToCanvas(q.tlx,
q.tly, false, true);
		var trx =
this.layer.layerToCanvas(q.trx,
q.try_, true, true);
		var try_ =
this.layer.layerToCanvas(q.trx,
q.try_, false, true);
		var brx =
this.layer.layerToCanvas(q.brx,
q.bry, true, true);
		var bry =
this.layer.layerToCanvas(q.brx,
q.bry, false, true);
		var blx =
this.layer.layerToCanvas(q.blx,
q.bly, true, true);
		var bly =
this.layer.layerToCanvas(q.blx,
q.bly, false, true);

	this.runtime.devicePixelRa
tio = old_dpr;
		if
(this.runtime.pixel_rounding ||
(this.angle === 0 &&
layer_angle === 0))
		{
			var ox = ((tlx
+ 0.5) | 0) - tlx;
			var oy = ((tly
+ 0.5) | 0) - tly
			tlx += ox;
			tly += oy;
			trx += ox;
			try_ += oy;
			brx += ox;
			bry += oy;
			blx += ox;
			bly += oy;
		}
		if (this.angle === 0
&& layer_angle === 0)
		{
```

```
		trx = tlx +
scaledwidth;
		try_ = tly;
		brx = trx;
		bry = tly +
scaledheight;
		blx = tlx;
		bly = bry;
		rcTex.right =
1;
		rcTex.bottom =
1;
	}
	else
	{
		rcTex.right =
floatscaledwidth / scaledwidth;
		rcTex.bottom =
floatscaledheight /
scaledheight;
	}
	glw.quadTex(tlx,
tly, trx, try_, brx, bry, blx,
bly, rcTex);

	glw.resetModelView();

	glw.scale(layer_scale,
layer_scale);
	glw.rotateZ(-
this.layer.getAngle());

	glw.translate((this.layer.
viewLeft +
this.layer.viewRight) / -2,
(this.layer.viewTop +
this.layer.viewBottom) / -2);

	glw.updateModelView();

	this.last_render_tick =
this.runtime.tickcount;
	};
	var wordsCache = [];
	pluginProto.TokeniseWords
= function (text)
	{
	wordsCache.length =
0;
	var cur_word = "";
	var ch;
	var i = 0;
	while (i <
text.length)
	{
```

```
                ch =
text.charAt(i);
                if (ch ===
"\n")
                {
                     if
(cur_word.length)
                     {

     wordsCache.push(cur_word);

     cur_word = "";
                     }

     wordsCache.push("\n");
                     ++i;
                }
                else if (ch ===
" " || ch === "\t" || ch === "-
")
                {
                     do {

     cur_word +=
text.charAt(i);
                          i++;
                     }
                     while (i
< text.length &&
(text.charAt(i) === " " ||
text.charAt(i) === "\t"));

     wordsCache.push(cur_word);
                     cur_word
= "";
                }
                else if (i <
text.length)
                {
                     cur_word
+= ch;
                     i++;
                }
           }
           if (cur_word.length)

     wordsCache.push(cur_word);
     };
     var linesCache = [];
     function allocLine()
     {
           if
(linesCache.length)
                return
linesCache.pop();
           else
                return {};
     };
     function freeLine(l)
     {
           linesCache.push(l);
     };
     function freeAllLines(arr)
     {
           var i, len;
           for (i = 0, len =
arr.length; i < len; i++)
           {

     freeLine(arr[i]);
           }
           arr.length = 0;
     };
     pluginProto.WordWrap =
function (text, lines, ctx,
width, wrapbyword)
     {
           if (!text ||
!text.length)
           {

     freeAllLines(lines);
                return;
           }
           if (width <= 2.0)
           {

     freeAllLines(lines);
                return;
           }
           if (text.length <=
100 && text.indexOf("\n") === -
1)
           {
                var all_width =
ctx.measureText(text).width;
                if (all_width
<= width)
                {

     freeAllLines(lines);

     lines.push(allocLine());

     lines[0].text = text;

     lines[0].width =
all_width;
                     return;
                }
           }
```

```
            this.WrapText(text,
lines, ctx, width, wrapbyword);
      };
      pluginProto.WrapText =
function (text, lines, ctx,
width, wrapbyword)
      {
            var wordArray;
            if (wrapbyword)
            {

      this.TokeniseWords(text);
      // writes to wordsCache
                  wordArray =
wordsCache;
            }
            else
                  wordArray =
text;
            var cur_line = "";
            var prev_line;
            var line_width;
            var i;
            var lineIndex = 0;
            var line;
            for (i = 0; i <
wordArray.length; i++)
            {
                  if
(wordArray[i] === "\n")
                  {
                        if
(lineIndex >= lines.length)

      lines.push(allocLine());
                        line =
lines[lineIndex];
                        line.text
= cur_line;

      line.width =
ctx.measureText(cur_line).width
;

      lineIndex++;
                        cur_line
= "";
                        continue;
                  }
                  prev_line =
cur_line;
                  cur_line +=
wordArray[i];
                  line_width =
ctx.measureText(cur_line).width
;
                  if (line_width
>= width)
                  {
                        if
(lineIndex >= lines.length)

      lines.push(allocLine());
                        line =
lines[lineIndex];
                        line.text
= prev_line;

      line.width =
ctx.measureText(prev_line).widt
h;

      lineIndex++;
                        cur_line
= wordArray[i];
                        if
(!wrapbyword && cur_line === "
")

      cur_line = "";
                  }
            }
            if (cur_line.length)
            {
                  if (lineIndex
>= lines.length)

      lines.push(allocLine());
                  line =
lines[lineIndex];
                  line.text =
cur_line;
                  line.width =
ctx.measureText(cur_line).width
;
                  lineIndex++;
            }
            for (i = lineIndex;
i < lines.length; i++)

      freeLine(lines[i]);
            lines.length =
lineIndex;
      };
      function Cnds() {};
      Cnds.prototype.CompareText
= function(text_to_compare,
case_sensitive)
      {
            if (case_sensitive)
                  return
this.text == text_to_compare;
```

```
		else
			return
cr.equals_nocase(this.text,
text_to_compare);
	};
	pluginProto.cnds = new
Cnds();
	function Acts() {};
	Acts.prototype.SetText =
function(param)
	{
		if
(cr.is_number(param) && param <
1e9)
			param =
Math.round(param * 1e10) /
1e10; // round to nearest ten
billionth - hides floating
point errors
		var text_to_set =
param.toString();
		if (this.text !==
text_to_set)
		{
			this.text =
text_to_set;

	this.text_changed = true;

	this.runtime.redraw =
true;
		}
	};
	Acts.prototype.AppendText
= function(param)
	{
		if
(cr.is_number(param))
			param =
Math.round(param * 1e10) /
1e10; // round to nearest ten
billionth - hides floating
point errors
		var text_to_append =
param.toString();
		if (text_to_append)
	// not empty
		{
			this.text +=
text_to_append;

	this.text_changed = true;

	this.runtime.redraw =
true;
		}
	};
	Acts.prototype.SetFontFace
= function (face_, style_)
	{
		var newstyle = "";
		switch (style_) {
		case 1: newstyle =
"bold"; break;
		case 2: newstyle =
"italic"; break;
		case 3: newstyle =
"bold italic"; break;
		}
		if (face_ ===
this.facename && newstyle ===
this.fontstyle)
			return;
	// no change
		this.facename =
face_;
		this.fontstyle =
newstyle;
		this.updateFont();
	};
	Acts.prototype.SetFontSize
= function (size_)
	{
		if (this.ptSize ===
size_)
			return;
		this.ptSize = size_;
		this.pxHeight =
Math.ceil((this.ptSize / 72.0)
* 96.0) + 4;	// assume
96dpi...
		this.updateFont();
	};
	Acts.prototype.SetFontColo
r = function (rgb)
	{
		var newcolor =
"rgb(" +
cr.GetRValue(rgb).toString() +
"," +
cr.GetGValue(rgb).toString() +
"," +
cr.GetBValue(rgb).toString() +
")";
		if (newcolor ===
this.color)
			return;
		this.color =
newcolor;

	this.need_text_redraw =
true;
```

```
            this.runtime.redraw
= true;
      };
      Acts.prototype.SetWebFont
= function (familyname_,
cssurl_)
      {
            if
(this.runtime.isDomFree)
            {

      cr.logexport("[Construct
2] Text plugin: 'Set web font'
not supported on this platform
- the action has been
ignored");
                  return;
      // DC todo
            }
            var self = this;
            var refreshFunc =
(function () {

            self.runtime.redraw
= true;

            self.text_changed =
true;

      });
            if
(requestedWebFonts.hasOwnProper
ty(cssurl_))
            {
                  var newfacename
= "'" + familyname_ + "'";
                  if
(this.facename === newfacename)
                        return;
      // no change
                  this.facename =
newfacename;

      this.updateFont();
                  for (var i = 1;
i < 10; i++)
                  {

      setTimeout(refreshFunc, i
* 100);

      setTimeout(refreshFunc, i
* 1000);
                  }
                  return;
            }
            var wf =
document.createElement("link");
            wf.href = cssurl_;
            wf.rel =
"stylesheet";
            wf.type =
"text/css";
            wf.onload =
refreshFunc;

      document.getElementsByTagN
ame('head')[0].appendChild(wf);

      requestedWebFonts[cssurl_]
= true;
            this.facename = "'"
+ familyname_ + "'";
            this.updateFont();
            for (var i = 1; i <
10; i++)
            {

      setTimeout(refreshFunc, i
* 100);

      setTimeout(refreshFunc, i
* 1000);
            }
;
      };
      Acts.prototype.SetEffect =
function (effect)
      {
            this.compositeOp =
cr.effectToCompositeOp(effect);
            cr.setGLBlend(this,
effect, this.runtime.gl);
            this.runtime.redraw
= true;
      };
      pluginProto.acts = new
Acts();
      function Exps() {};
      Exps.prototype.Text =
function(ret)
      {

      ret.set_string(this.text);
      };
      Exps.prototype.FaceName =
function (ret)
      {

      ret.set_string(this.facena
me);
      };
```

```
	Exps.prototype.FaceSize =
function (ret)
	{

	ret.set_int(this.ptSize);
	};
	Exps.prototype.TextWidth =
function (ret)
	{
		var w = 0;
		var i, len, x;
		for (i = 0, len =
this.lines.length; i < len;
i++)
		{
			x =
this.lines[i].width;
			if (w < x)
				w = x;
		}
		ret.set_int(w);
	};
	Exps.prototype.TextHeight
= function (ret)
	{

	ret.set_int(this.lines.len
gth * (this.pxHeight +
this.line_height_offset) -
this.line_height_offset);
	};
	pluginProto.exps = new
Exps();
}());
;
;
cr.plugins_.TextBox =
function(runtime)
{
	this.runtime = runtime;
};
(function ()
{
	var pluginProto =
cr.plugins_.TextBox.prototype;
	pluginProto.Type =
function(plugin)
	{
		this.plugin =
plugin;
		this.runtime =
plugin.runtime;
	};
	var typeProto =
pluginProto.Type.prototype;
	typeProto.onCreate =
function()
	{
	};
	pluginProto.Instance =
function(type)
	{
		this.type = type;
		this.runtime =
type.runtime;
	};
	var instanceProto =
pluginProto.Instance.prototype;
	var elemTypes = ["text",
"password", "email", "number",
"tel", "url"];
	if
(navigator.userAgent.indexOf("M
SIE 9") > -1)
	{
		elemTypes[2] =
"text";
		elemTypes[3] =
"text";
		elemTypes[4] =
"text";
		elemTypes[5] =
"text";
	}
	instanceProto.onCreate =
function()
	{
		if
(this.runtime.isDomFree)
		{

	cr.logexport("[Construct
2] Textbox plugin not supported
on this platform - the object
will not be created");
			return;
		}
		if
(this.properties[7] === 6) //
textarea
		{
			this.elem =
document.createElement("textare
a");

	jQuery(this.elem).css("res
ize", "none");
		}
		else
		{
```

```
                this.elem =
document.createElement("input")
;
                this.elem.type
=
elemTypes[this.properties[7]];
            }
            this.elem.id =
this.properties[9];

     jQuery(this.elem).appendTo
(this.runtime.canvasdiv ?
this.runtime.canvasdiv :
"body");

     this.elem["autocomplete"]
= "off";
            this.elem.value =
this.properties[0];

     this.elem["placeholder"] =
this.properties[1];
            this.elem.title =
this.properties[2];
            this.elem.disabled =
(this.properties[4] === 0);

     this.elem["readOnly"] =
(this.properties[5] === 1);

     this.elem["spellcheck"] =
(this.properties[6] === 1);
            this.autoFontSize =
(this.properties[8] !== 0);
            this.element_hidden
= false;
            if
(this.properties[3] === 0)
            {

     jQuery(this.elem).hide();
                this.visible =
false;

     this.element_hidden =
true;
            }
            var onchangetrigger
= (function (self) {
                return
function() {

     self.runtime.trigger(cr.pl
ugins_.TextBox.prototype.cnds.O
nTextChanged, self);
                };
            })(this);
            this.elem["oninput"]
= onchangetrigger;
            if
(navigator.userAgent.indexOf("M
SIE") !== -1)

     this.elem["oncut"] =
onchangetrigger;
            this.elem.onclick =
(function (self) {
                return
function(e) {

     e.stopPropagation();

     self.runtime.isInUserInput
Event = true;

     self.runtime.trigger(cr.pl
ugins_.TextBox.prototype.cnds.O
nClicked, self);

     self.runtime.isInUserInput
Event = false;
                };
            })(this);
            this.elem.ondblclick
= (function (self) {
                return
function(e) {

     e.stopPropagation();

     self.runtime.isInUserInput
Event = true;

     self.runtime.trigger(cr.pl
ugins_.TextBox.prototype.cnds.O
nDoubleClicked, self);

     self.runtime.isInUserInput
Event = false;
                };
            })(this);

     this.elem.addEventListener
("touchstart", function (e) {

     e.stopPropagation();
            }, false);

     this.elem.addEventListener
("touchmove", function (e) {

     e.stopPropagation();
```

```
            }, false);

      this.elem.addEventListener
("touchend", function (e) {

      e.stopPropagation();
            }, false);

      jQuery(this.elem).mousedow
n(function (e) {

      e.stopPropagation();
            });

      jQuery(this.elem).mouseup(
function (e) {

      e.stopPropagation();
            });

      jQuery(this.elem).keydown(
function (e) {
                  if (e.which !==
13 && e.which != 27)  // allow
enter and escape

      e.stopPropagation();
            });

      jQuery(this.elem).keyup(fu
nction (e) {
                  if (e.which !==
13 && e.which != 27)  // allow
enter and escape

      e.stopPropagation();
            });
            this.lastLeft = 0;
            this.lastTop = 0;
            this.lastRight = 0;
            this.lastBottom = 0;
            this.lastWinWidth =
0;
            this.lastWinHeight =
0;

      this.updatePosition(true);

      this.runtime.tickMe(this);
      };
      instanceProto.saveToJSON =
function ()
      {
            return {
                  "text":
this.elem.value,
                  "placeholder":
this.elem.placeholder,
                  "tooltip":
this.elem.title,
                  "disabled":
!!this.elem.disabled,
                  "readonly":
!!this.elem.readOnly,
                  "spellcheck":
!!this.elem["spellcheck"]
            };
      };
      instanceProto.loadFromJSON
= function (o)
      {
            this.elem.value =
o["text"];

      this.elem.placeholder =
o["placeholder"];
            this.elem.title =
o["tooltip"];
            this.elem.disabled =
o["disabled"];
            this.elem.readOnly =
o["readonly"];

      this.elem["spellcheck"] =
o["spellcheck"];
      };
      instanceProto.onDestroy =
function ()
      {
            if
(this.runtime.isDomFree)
                  return;

      jQuery(this.elem).remove()
;
            this.elem = null;
      };
      instanceProto.tick =
function ()
      {

      this.updatePosition();
      };
      instanceProto.updatePositi
on = function (first)
      {
            if
(this.runtime.isDomFree)
                  return;
            var left =
this.layer.layerToCanvas(this.x
, this.y, true);
```

```
            var top =
this.layer.layerToCanvas(this.x
, this.y, false);
            var right =
this.layer.layerToCanvas(this.x
+ this.width, this.y +
this.height, true);
            var bottom =
this.layer.layerToCanvas(this.x
+ this.width, this.y +
this.height, false);
            if (!this.visible ||
!this.layer.visible || right <=
0 || bottom <= 0 || left >=
this.runtime.width || top >=
this.runtime.height)
            {
                  if
(!this.element_hidden)

      jQuery(this.elem).hide();

      this.element_hidden =
true;
                  return;
            }
            if (left < 1)
                  left = 1;
            if (top < 1)
                  top = 1;
            if (right >=
this.runtime.width)
                  right =
this.runtime.width - 1;
            if (bottom >=
this.runtime.height)
                  bottom =
this.runtime.height - 1;
            var curWinWidth =
window.innerWidth;
            var curWinHeight =
window.innerHeight;
            if (!first &&
this.lastLeft === left &&
this.lastTop === top &&
this.lastRight === right &&
this.lastBottom === bottom &&
this.lastWinWidth ===
curWinWidth &&
this.lastWinHeight ===
curWinHeight)
            {
                  if
(this.element_hidden)
                  {

      jQuery(this.elem).show();

      this.element_hidden =
false;
                  }
                  return;
            }
            this.lastLeft =
left;
            this.lastTop = top;
            this.lastRight =
right;
            this.lastBottom =
bottom;
            this.lastWinWidth =
curWinWidth;
            this.lastWinHeight =
curWinHeight;
            if
(this.element_hidden)
            {

      jQuery(this.elem).show();

      this.element_hidden =
false;
            }
            var offx =
Math.round(left) +
jQuery(this.runtime.canvas).off
set().left;
            var offy =
Math.round(top) +
jQuery(this.runtime.canvas).off
set().top;

      jQuery(this.elem).css("pos
ition", "absolute");

      jQuery(this.elem).offset({
left: offx, top: offy});

      jQuery(this.elem).width(Ma
th.round(right - left));

      jQuery(this.elem).height(M
ath.round(bottom - top));
            if
(this.autoFontSize)

      jQuery(this.elem).css("fon
t-size",
((this.layer.getScale() /
this.runtime.devicePixelRatio)
- 0.2) + "em");
```

```
	};
	instanceProto.draw =
function(ctx)
	{
	};
	instanceProto.drawGL =
function(glw)
	{
	};
	function Cnds() {};
	Cnds.prototype.CompareText
= function (text, case_)
	{
		if
(this.runtime.isDomFree)
			return false;
		if (case_ === 0) //
insensitive
			return
cr.equals_nocase(this.elem.valu
e, text);
		else
			return
this.elem.value === text;
	};
	Cnds.prototype.OnTextChang
ed = function ()
	{
		return true;
	};
	Cnds.prototype.OnClicked =
function ()
	{
		return true;
	};
	Cnds.prototype.OnDoubleCli
cked = function ()
	{
		return true;
	};
	pluginProto.cnds = new
Cnds();
	function Acts() {};
	Acts.prototype.SetText =
function (text)
	{
		if
(this.runtime.isDomFree)
			return;
		this.elem.value =
text;
	};
	Acts.prototype.SetPlacehol
der = function (text)
	{
		if
(this.runtime.isDomFree)
			return;

	this.elem.placeholder =
text;
	};
	Acts.prototype.SetTooltip
= function (text)
	{
		if
(this.runtime.isDomFree)
			return;
		this.elem.title =
text;
	};
	Acts.prototype.SetVisible
= function (vis)
	{
		if
(this.runtime.isDomFree)
			return;
		this.visible = (vis
!== 0);
	};
	Acts.prototype.SetEnabled
= function (en)
	{
		if
(this.runtime.isDomFree)
			return;
		this.elem.disabled =
(en === 0);
	};
	Acts.prototype.SetReadOnly
= function (ro)
	{
		if
(this.runtime.isDomFree)
			return;
		this.elem.readOnly =
(ro === 0);
	};
	Acts.prototype.SetFocus =
function ()
	{
		if
(this.runtime.isDomFree)
			return;
		this.elem.focus();
	};
	Acts.prototype.SetBlur =
function ()
	{
		if
(this.runtime.isDomFree)
```

```
                return;
            this.elem.blur();
        };
        Acts.prototype.SetCSSStyle
= function (p, v)
        {
            if
(this.runtime.isDomFree)
                return;

        jQuery(this.elem).css(p,
v);
        };
        pluginProto.acts = new
Acts();
        function Exps() {};
        Exps.prototype.Text =
function (ret)
        {
            if
(this.runtime.isDomFree)
            {

        ret.set_string("");
                return;
            }

        ret.set_string(this.elem.v
alue);
        };
        pluginProto.exps = new
Exps();
}());
;
;
cr.plugins_.Touch =
function(runtime)
{
        this.runtime = runtime;
};
(function ()
{
        var pluginProto =
cr.plugins_.Touch.prototype;
        pluginProto.Type =
function(plugin)
        {
            this.plugin =
plugin;
            this.runtime =
plugin.runtime;
        };
        var typeProto =
pluginProto.Type.prototype;
        typeProto.onCreate =
function()
        {
        };
        pluginProto.Instance =
function(type)
        {
            this.type = type;
            this.runtime =
type.runtime;
            this.touches = [];
            this.mouseDown =
false;
        };
        var instanceProto =
pluginProto.Instance.prototype;
        var dummyoffset = {left:
0, top: 0};
        instanceProto.findTouch =
function (id)
        {
            var i, len;
            for (i = 0, len =
this.touches.length; i < len;
i++)
            {
                if
(this.touches[i]["id"] === id)
                    return i;
            }
            return -1;
        };
        var appmobi_accx = 0;
        var appmobi_accy = 0;
        var appmobi_accz = 0;
        function
AppMobiGetAcceleration(evt)
        {
            appmobi_accx =
evt.x;
            appmobi_accy =
evt.y;
            appmobi_accz =
evt.z;
        };
        var pg_accx = 0;
        var pg_accy = 0;
        var pg_accz = 0;
        function
PhoneGapGetAcceleration(evt)
        {
            pg_accx = evt.x;
            pg_accy = evt.y;
            pg_accz = evt.z;
        };
        var theInstance = null;
        instanceProto.onCreate =
function()
```

```
	{
		theInstance = this;
		this.isWindows8 = !!(typeof window["c2isWindows8"] !== "undefined" && window["c2isWindows8"]);
		this.orient_alpha = 0;
		this.orient_beta = 0;
		this.orient_gamma = 0;
		this.acc_g_x = 0;
		this.acc_g_y = 0;
		this.acc_g_z = 0;
		this.acc_x = 0;
		this.acc_y = 0;
		this.acc_z = 0;
		this.curTouchX = 0;
		this.curTouchY = 0;
		this.trigger_index = 0;
		this.trigger_id = 0;
		this.useMouseInput = (this.properties[0] !== 0);
		var elem = (this.runtime.fullscreen_mode > 0) ? document : this.runtime.canvas;
		var elem2 = document;
		if (this.runtime.isDirectCanvas)
			elem2 = elem = window["Canvas"];
		else if (this.runtime.isCocoonJs)
			elem2 = elem = window;
		var self = this;
		if (window.navigator["pointerEnabled"])
		{
	elem.addEventListener("pointerdown",
	function(info) {
	self.onPointerStart(info);
				},
				false
			);
	elem.addEventListener("pointermove",
	function(info) {
	self.onPointerMove(info);
				},
				false
			);
	elem2.addEventListener("pointerup",
	function(info) {
	self.onPointerEnd(info);
				},
				false
			);
	elem2.addEventListener("pointercancel",
	function(info) {
	self.onPointerEnd(info);
				},
				false
			);
			if (this.runtime.canvas)
			{
	this.runtime.canvas.addEventListener("MSGestureHold", function(e) {
	e.preventDefault();
				},
	false);
	document.addEventListener("MSGestureHold", function(e) {
	e.preventDefault();
				},
	false);
	this.runtime.canvas.addEventListener("gesturehold", function(e) {
	e.preventDefault();
				},
	false);
```

```
	document.addEventListener(
"gesturehold", function(e) {

	e.preventDefault();
				},
false);
		}
	}
	else if
(window.navigator["msPointerEna
bled"])
	{

	elem.addEventListener("MSP
ointerDown",

	function(info) {

	self.onPointerStart(info);
				},
				false
		);

	elem.addEventListener("MSP
ointerMove",

	function(info) {

	self.onPointerMove(info);
				},
				false
		);

	elem2.addEventListener("MS
PointerUp",

	function(info) {

	self.onPointerEnd(info);
				},
				false
		);

	elem2.addEventListener("MS
PointerCancel",

	function(info) {

	self.onPointerEnd(info);
				},
				false
		);
		if
(this.runtime.canvas)
		{

	this.runtime.canvas.addEve
ntListener("MSGestureHold",
function(e) {

	e.preventDefault();
				},
false);

	document.addEventListener(
"MSGestureHold", function(e) {

	e.preventDefault();
				},
false);
		}
	}
	else
	{

	elem.addEventListener("tou
chstart",

	function(info) {

	self.onTouchStart(info);
				},
				false
		);

	elem.addEventListener("tou
chmove",

	function(info) {

	self.onTouchMove(info);
				},
				false
		);

	elem2.addEventListener("to
uchend",

	function(info) {

	self.onTouchEnd(info);
				},
				false
		);

	elem2.addEventListener("to
uchcancel",

	function(info) {

	self.onTouchEnd(info);
```

```
                    },
                    false
                );
            }
            if (this.isWindows8)
            {
                var
win8accelerometerFn =
function(e) {
                        var
reading = e["reading"];

     self.acc_x =
reading["accelerationX"];

     self.acc_y =
reading["accelerationY"];

     self.acc_z =
reading["accelerationZ"];
                    };
                var
win8inclinometerFn =
function(e) {
                        var
reading = e["reading"];

     self.orient_alpha =
reading["yawDegrees"];

     self.orient_beta =
reading["pitchDegrees"];

     self.orient_gamma =
reading["rollDegrees"];
                    };
                var
accelerometer =
Windows["Devices"]["Sensors"]["
Accelerometer"]["getDefault"]()
;
            if (accelerometer)
                {

accelerometer["reportInterval"]
=
Math.max(accelerometer["minimum
ReportInterval"], 16);

     accelerometer.addEventList
ener("readingchanged",
win8accelerometerFn);
            }
                var
inclinometer =
Windows["Devices"]["Sensors"]["
Inclinometer"]["getDefault"]();
                if
(inclinometer)
                {

     inclinometer["reportInterv
al"] =
Math.max(inclinometer["minimumR
eportInterval"], 16);

     inclinometer.addEventListe
ner("readingchanged",
win8inclinometerFn);
                }

     document.addEventListener(
"visibilitychange", function(e)
{
                    if
(document["hidden"] ||
document["msHidden"])
                    {
                        if
(accelerometer)

     accelerometer.removeEventL
istener("readingchanged",
win8accelerometerFn);
                        if
(inclinometer)

     inclinometer.removeEventLi
stener("readingchanged",
win8inclinometerFn);
                    }
                    else
                    {
                        if
(accelerometer)

     accelerometer.addEventList
ener("readingchanged",
win8accelerometerFn);
                        if
(inclinometer)

     inclinometer.addEventListe
ner("readingchanged",
win8inclinometerFn);
                    }
                }, false);
            }
            else
            {
```

```
window.addEventListener("d
eviceorientation", function
(eventData) {

self.orient_alpha =
eventData["alpha"] || 0;

self.orient_beta =
eventData["beta"] || 0;

self.orient_gamma =
eventData["gamma"] || 0;
}, false);

window.addEventListener("d
evicemotion", function
(eventData) {
if
(eventData["accelerationIncludi
ngGravity"])
{

self.acc_g_x =
eventData["accelerationIncludin
gGravity"]["x"];

self.acc_g_y =
eventData["accelerationIncludin
gGravity"]["y"];

self.acc_g_z =
eventData["accelerationIncludin
gGravity"]["z"];
}
if
(eventData["acceleration"])
{

self.acc_x =
eventData["acceleration"]["x"];

self.acc_y =
eventData["acceleration"]["y"];

self.acc_z =
eventData["acceleration"]["z"];
}
}, false);
}
if
(this.useMouseInput &&
!this.runtime.isDomFree)
{

jQuery(document).mousemove
(

function(info) {

self.onMouseMove(info);
}
);

jQuery(document).mousedown
(

function(info) {

self.onMouseDown(info);
}
);

jQuery(document).mouseup(

function(info) {

self.onMouseUp(info);
}
);
}
if
(this.runtime.isAppMobi &&
!this.runtime.isDirectCanvas)
{

AppMobi["accelerometer"]["
watchAcceleration"](AppMobiGetA
cceleration, { "frequency": 40,
"adjustForRotation": true });
}
if
(this.runtime.isPhoneGap)
{

navigator["accelerometer"]
["watchAcceleration"](PhoneGapG
etAcceleration, null, {
"frequency": 40 });
}

this.runtime.tick2Me(this)
;
};
instanceProto.onPointerMov
e = function (info)
{
if
(info["pointerType"] ===
info["MSPOINTER_TYPE_MOUSE"] ||
```

```
info["pointerType"] ===
"mouse")
                return;
            if
(info.preventDefault)

        info.preventDefault();
            var i =
this.findTouch(info["pointerId"
]);
            var nowtime =
cr.performance_now();
            if (i >= 0)
            {
                var offset =
this.runtime.isDomFree ?
dummyoffset :
jQuery(this.runtime.canvas).off
set();
                var t =
this.touches[i];
                if (nowtime -
t.time < 2)
                    return;
                t.lasttime =
t.time;
                t.lastx = t.x;
                t.lasty = t.y;
                t.time =
nowtime;
                t.x =
info.pageX - offset.left;
                t.y =
info.pageY - offset.top;
            }
        };
        instanceProto.onPointerSta
rt = function (info)
        {
            if
(info["pointerType"] ===
info["MSPOINTER_TYPE_MOUSE"] ||
info["pointerType"] ===
"mouse")
                return;
            if
(info.preventDefault)

        info.preventDefault();
            var offset =
this.runtime.isDomFree ?
dummyoffset :
jQuery(this.runtime.canvas).off
set();
            var touchx =
info.pageX - offset.left;
            var touchy =
info.pageY - offset.top;
            var nowtime =
cr.performance_now();
            this.trigger_index =
this.touches.length;
            this.trigger_id =
info["pointerId"];
            this.touches.push({
time: nowtime,

            x: touchx,

            y: touchy,

            lasttime: nowtime,

            lastx: touchx,

            lasty: touchy,

            "id":
info["pointerId"],

            startindex:
this.trigger_index

        });

        this.runtime.isInUserInput
Event = true;

        this.runtime.trigger(cr.pl
ugins_.Touch.prototype.cnds.OnN
thTouchStart, this);

        this.runtime.trigger(cr.pl
ugins_.Touch.prototype.cnds.OnT
ouchStart, this);
            this.curTouchX =
touchx;
            this.curTouchY =
touchy;

        this.runtime.trigger(cr.pl
ugins_.Touch.prototype.cnds.OnT
ouchObject, this);

        this.runtime.isInUserInput
Event = false;
        };
        instanceProto.onPointerEnd
= function (info)
        {
            if
(info["pointerType"] ===
```

```
info["MSPOINTER_TYPE_MOUSE"] ||
info["pointerType"] ===
"mouse")
                return;
            if
(info.preventDefault)

      info.preventDefault();
            var i =
this.findTouch(info["pointerId"
]);
            this.trigger_index =
(i >= 0 ?
this.touches[i].startindex : -
1);
            this.trigger_id = (i
>= 0 ? this.touches[i]["id"] :
-1);

      this.runtime.isInUserInput
Event = true;

      this.runtime.trigger(cr.pl
ugins_.Touch.prototype.cnds.OnN
thTouchEnd, this);

      this.runtime.trigger(cr.pl
ugins_.Touch.prototype.cnds.OnT
ouchEnd, this);

      this.runtime.isInUserInput
Event = false;
            if (i >= 0)
            {

      this.touches.splice(i, 1);
            }
      };
      instanceProto.onTouchMove
= function (info)
      {
            if
(info.preventDefault)

      info.preventDefault();
            var nowtime =
cr.performance_now();
            var i, len, t, u;
            for (i = 0, len =
info.changedTouches.length; i <
len; i++)
            {
                  t =
info.changedTouches[i];
                  var j =
this.findTouch(t["identifier"])
;
                  if (j >= 0)
                  {
                        var
offset = this.runtime.isDomFree
? dummyoffset :
jQuery(this.runtime.canvas).off
set();
                        u =
this.touches[j];
                        if
(nowtime - u.time < 2)

      continue;

      u.lasttime = u.time;
                        u.lastx =
u.x;
                        u.lasty =
u.y;
                        u.time =
nowtime;
                        u.x =
t.pageX - offset.left;
                        u.y =
t.pageY - offset.top;
                  }
            }
      };
      instanceProto.onTouchStart
= function (info)
      {
            if
(info.preventDefault)

      info.preventDefault();
            var offset =
this.runtime.isDomFree ?
dummyoffset :
jQuery(this.runtime.canvas).off
set();
            var nowtime =
cr.performance_now();

      this.runtime.isInUserInput
Event = true;
            var i, len, t, j;
            for (i = 0, len =
info.changedTouches.length; i <
len; i++)
            {
                  t =
info.changedTouches[i];
```

```
                j =
this.findTouch(t["identifier"])
;
                if (j !== -1)
                      continue;
                var touchx =
t.pageX - offset.left;
                var touchy =
t.pageY - offset.top;

     this.trigger_index =
this.touches.length;
                this.trigger_id
= t["identifier"];

     this.touches.push({ time:
nowtime,

                x: touchx,

                y: touchy,

                lasttime:
nowtime,

                lastx: touchx,

                lasty: touchy,

                "id":
t["identifier"],

                startindex:
this.trigger_index

           });

     this.runtime.trigger(cr.pl
ugins_.Touch.prototype.cnds.OnN
thTouchStart, this);

     this.runtime.trigger(cr.pl
ugins_.Touch.prototype.cnds.OnT
ouchStart, this);
                this.curTouchX
= touchx;
                this.curTouchY
= touchy;

     this.runtime.trigger(cr.pl
ugins_.Touch.prototype.cnds.OnT
ouchObject, this);
           }

     this.runtime.isInUserInput
Event = false;
     };
     instanceProto.onTouchEnd =
function (info)
     {
           if
(info.preventDefault)

     info.preventDefault();

     this.runtime.isInUserInput
Event = true;
           var i, len, t, j;
           for (i = 0, len =
info.changedTouches.length; i <
len; i++)
           {
                 t =
info.changedTouches[i];
                 j =
this.findTouch(t["identifier"])
;
                 if (j >= 0)
                 {

     this.trigger_index =
this.touches[j].startindex;

     this.trigger_id =
this.touches[j]["id"];

     this.runtime.trigger(cr.pl
ugins_.Touch.prototype.cnds.OnN
thTouchEnd, this);

     this.runtime.trigger(cr.pl
ugins_.Touch.prototype.cnds.OnT
ouchEnd, this);

     this.touches.splice(j, 1);
                 }
           }

     this.runtime.isInUserInput
Event = false;
     };
     instanceProto.getAlpha =
function ()
     {
           if
(this.runtime.isAppMobi &&
this.orient_alpha === 0 &&
appmobi_accz !== 0)
                 return
appmobi_accz * 90;
           else if
(this.runtime.isPhoneGap  &&
```

```
this.orient_alpha === 0 &&
pg_accz !== 0)
                return pg_accz
* 90;
           else
                return
this.orient_alpha;
      };
      instanceProto.getBeta =
function ()
      {
           if
(this.runtime.isAppMobi &&
this.orient_beta === 0 &&
appmobi_accy !== 0)
                return
appmobi_accy * -90;
           else if
(this.runtime.isPhoneGap  &&
this.orient_beta === 0 &&
pg_accy !== 0)
                return pg_accy
* -90;
           else
                return
this.orient_beta;
      };
      instanceProto.getGamma =
function ()
      {
           if
(this.runtime.isAppMobi &&
this.orient_gamma === 0 &&
appmobi_accx !== 0)
                return
appmobi_accx * 90;
           else if
(this.runtime.isPhoneGap  &&
this.orient_gamma === 0 &&
pg_accx !== 0)
                return pg_accx
* 90;
           else
                return
this.orient_gamma;
      };
      var noop_func =
function(){};
      instanceProto.onMouseDown
= function(info)
      {
           if
(info.preventDefault &&
this.runtime.had_a_click)

      info.preventDefault();
           var t = { pageX:
info.pageX, pageY: info.pageY,
"identifier": 0 };
           var fakeinfo = {
changedTouches: [t] };

      this.onTouchStart(fakeinfo
);
           this.mouseDown =
true;
      };
      instanceProto.onMouseMove
= function(info)
      {
           if
(info.preventDefault &&
this.runtime.had_a_click)

      info.preventDefault();
           if (!this.mouseDown)
                return;
           var t = { pageX:
info.pageX, pageY: info.pageY,
"identifier": 0 };
           var fakeinfo = {
changedTouches: [t] };

      this.onTouchMove(fakeinfo)
;
      };
      instanceProto.onMouseUp =
function(info)
      {
           if
(info.preventDefault &&
this.runtime.had_a_click)

      info.preventDefault();

      this.runtime.had_a_click =
true;
           var t = { pageX:
info.pageX, pageY: info.pageY,
"identifier": 0 };
           var fakeinfo = {
changedTouches: [t] };

      this.onTouchEnd(fakeinfo);
           this.mouseDown =
false;
      };
      instanceProto.tick2 =
function()
      {
           var i, len, t;
```

```
		var nowtime = cr.performance_now();
		for (i = 0, len = this.touches.length; i < len; i++)
		{
			t = this.touches[i];
			if (t.time <= nowtime - 50)

	t.lasttime = nowtime;
		}
	};
	function Cnds() {};
	Cnds.prototype.OnTouchStart = function ()
	{
		return true;
	};
	Cnds.prototype.OnTouchEnd = function ()
	{
		return true;
	};
	Cnds.prototype.IsInTouch = function ()
	{
		return this.touches.length;
	};
	Cnds.prototype.OnTouchObject = function (type)
	{
		if (!type)
			return false;
		return this.runtime.testAndSelectCanvasPointOverlap(type, this.curTouchX, this.curTouchY, false);
	};
	Cnds.prototype.IsTouchingObject = function (type)
	{
		if (!type)
			return false;
		var sol = type.getCurrentSol();
		var instances = sol.getObjects();
		var px, py;
		var touching = [];
		var i, leni, j, lenj;
		for (i = 0, leni = instances.length; i < leni; i++)
		{
			var inst = instances[i];

	inst.update_bbox();
			for (j = 0, lenj = this.touches.length; j < lenj; j++)
			{
				var touch = this.touches[j];
				px = inst.layer.canvasToLayer(touch.x, touch.y, true);
				py = inst.layer.canvasToLayer(touch.x, touch.y, false);
				if (inst.contains_pt(px, py))
				{

	touching.push(inst);

	break;
				}
			}
		}
		if (touching.length)
		{
			sol.select_all = false;
			sol.instances = touching;

	type.applySolToContainer();
			return true;
		}
		else
			return false;
	};
	Cnds.prototype.CompareTouchSpeed = function (index, cmp, s)
	{
		index = Math.floor(index);
		if (index < 0 || index >= this.touches.length)
			return false;
		var t = this.touches[index];
```

```
		var dist =
cr.distanceTo(t.x, t.y,
t.lastx, t.lasty);
		var timediff =
(t.time - t.lasttime) / 1000;
		var speed = 0;
		if (timediff > 0)
			speed = dist /
timediff;
		return
cr.do_cmp(speed, cmp, s);
	};
	Cnds.prototype.Orientation
Supported = function ()
	{
		return typeof
window["DeviceOrientationEvent"
] !== "undefined";
	};
	Cnds.prototype.MotionSuppo
rted = function ()
	{
		return typeof
window["DeviceMotionEvent"] !==
"undefined";
	};
	Cnds.prototype.CompareOrie
ntation = function
(orientation_, cmp_, angle_)
	{
		var v = 0;
		if (orientation_ ===
0)
			v =
this.getAlpha();
		else if
(orientation_ === 1)
			v =
this.getBeta();
		else
			v =
this.getGamma();
		return cr.do_cmp(v,
cmp_, angle_);
	};
	Cnds.prototype.CompareAcce
leration = function
(acceleration_, cmp_, angle_)
	{
		var v = 0;
		if (acceleration_
=== 0)
			v =
this.acc_g_x;
		else if
(acceleration_ === 1)
			v =
this.acc_g_y;
		else if
(acceleration_ === 2)
			v =
this.acc_g_z;
		else if
(acceleration_ === 3)
			v = this.acc_x;
		else if
(acceleration_ === 4)
			v = this.acc_y;
		else if
(acceleration_ === 5)
			v = this.acc_z;
		return cr.do_cmp(v,
cmp_, angle_);
	};
	Cnds.prototype.OnNthTouchS
tart = function (touch_)
	{
		touch_ =
Math.floor(touch_);
		return touch_ ===
this.trigger_index;
	};
	Cnds.prototype.OnNthTouchE
nd = function (touch_)
	{
		touch_ =
Math.floor(touch_);
		return touch_ ===
this.trigger_index;
	};
	Cnds.prototype.HasNthTouch
= function (touch_)
	{
		touch_ =
Math.floor(touch_);
		return
this.touches.length >= touch_ +
1;
	};
	pluginProto.cnds = new
Cnds();
	function Exps() {};
	Exps.prototype.TouchCount
= function (ret)
	{

	ret.set_int(this.touches.l
ength);
	};
	Exps.prototype.X =
function (ret, layerparam)
	{
```

```
            if
(this.touches.length)
            {
                  var layer,
oldScale, oldZoomRate,
oldParallaxX, oldAngle;
                  if
(cr.is_undefined(layerparam))
                  {
                        layer =
this.runtime.getLayerByNumber(0
);
                        oldScale
= layer.scale;

      oldZoomRate =
layer.zoomRate;

      oldParallaxX =
layer.parallaxX;
                        oldAngle
= layer.angle;

      layer.scale =
this.runtime.running_layout.sca
le;

      layer.zoomRate = 1.0;

      layer.parallaxX = 1.0;

      layer.angle =
this.runtime.running_layout.ang
le;

      ret.set_float(layer.canvas
ToLayer(this.touches[0].x,
this.touches[0].y, true));

      layer.scale = oldScale;

      layer.zoomRate =
oldZoomRate;

      layer.parallaxX =
oldParallaxX;

      layer.angle = oldAngle;
                  }
                  else
                  {
                        if
(cr.is_number(layerparam))

      layer =
this.runtime.getLayerByNumber(l
ayerparam);
                        else

      layer =
this.runtime.getLayerByName(lay
erparam);
                        if
(layer)

      ret.set_float(layer.canvas
ToLayer(this.touches[0].x,
this.touches[0].y, true));
                        else

      ret.set_float(0);
                  }
            }
            else

      ret.set_float(0);
      };
      Exps.prototype.XAt =
function (ret, index,
layerparam)
      {
            index =
Math.floor(index);
            if (index < 0 ||
index >= this.touches.length)
            {

      ret.set_float(0);
                  return;
            }
            var layer, oldScale,
oldZoomRate, oldParallaxX,
oldAngle;
            if
(cr.is_undefined(layerparam))
            {
                  layer =
this.runtime.getLayerByNumber(0
);
                  oldScale =
layer.scale;
                  oldZoomRate =
layer.zoomRate;
                  oldParallaxX =
layer.parallaxX;
                  oldAngle =
layer.angle;
                  layer.scale =
this.runtime.running_layout.sca
le;
```

```
                layer.zoomRate
= 1.0;
                layer.parallaxX
= 1.0;
                layer.angle =
this.runtime.running_layout.ang
le;

     ret.set_float(layer.canvas
ToLayer(this.touches[index].x,
this.touches[index].y, true));
                layer.scale =
oldScale;
                layer.zoomRate
= oldZoomRate;
                layer.parallaxX
= oldParallaxX;
                layer.angle =
oldAngle;
           }
           else
           {
                if
(cr.is_number(layerparam))
                     layer =
this.runtime.getLayerByNumber(l
ayerparam);
                else
                     layer =
this.runtime.getLayerByName(lay
erparam);
                if (layer)

     ret.set_float(layer.canvas
ToLayer(this.touches[index].x,
this.touches[index].y, true));
                else

     ret.set_float(0);
           }
     };
     Exps.prototype.XForID =
function (ret, id, layerparam)
     {
           var index =
this.findTouch(id);
           if (index < 0)
           {

     ret.set_float(0);
                return;
           }
           var touch =
this.touches[index];
           var layer, oldScale,
oldZoomRate, oldParallaxX,
oldAngle;
           if
(cr.is_undefined(layerparam))
           {
                layer =
this.runtime.getLayerByNumber(0
);
                oldScale =
layer.scale;
                oldZoomRate =
layer.zoomRate;
                oldParallaxX =
layer.parallaxX;
                oldAngle =
layer.angle;
                layer.scale =
this.runtime.running_layout.sca
le;
                layer.zoomRate
= 1.0;
                layer.parallaxX
= 1.0;
                layer.angle =
this.runtime.running_layout.ang
le;

     ret.set_float(layer.canvas
ToLayer(touch.x, touch.y,
true));
                layer.scale =
oldScale;
                layer.zoomRate
= oldZoomRate;
                layer.parallaxX
= oldParallaxX;
                layer.angle =
oldAngle;
           }
           else
           {
                if
(cr.is_number(layerparam))
                     layer =
this.runtime.getLayerByNumber(l
ayerparam);
                else
                     layer =
this.runtime.getLayerByName(lay
erparam);
                if (layer)

     ret.set_float(layer.canvas
ToLayer(touch.x, touch.y,
true));
```

```
                else

     ret.set_float(0);
          }
     };
     Exps.prototype.Y =
function (ret, layerparam)
     {
          if
(this.touches.length)
          {
                var layer,
oldScale, oldZoomRate,
oldParallaxY, oldAngle;
                if
(cr.is_undefined(layerparam))
                {
                      layer =
this.runtime.getLayerByNumber(0
);
                      oldScale
= layer.scale;

     oldZoomRate =
layer.zoomRate;

     oldParallaxY =
layer.parallaxY;
                      oldAngle
= layer.angle;

     layer.scale =
this.runtime.running_layout.sca
le;

     layer.zoomRate = 1.0;

     layer.parallaxY = 1.0;

     layer.angle =
this.runtime.running_layout.ang
le;

     ret.set_float(layer.canvas
ToLayer(this.touches[0].x,
this.touches[0].y, false));

     layer.scale = oldScale;

     layer.zoomRate =
oldZoomRate;

     layer.parallaxY =
oldParallaxY;

     layer.angle = oldAngle;
                }
                else
                {
                      if
(cr.is_number(layerparam))

     layer =
this.runtime.getLayerByNumber(l
ayerparam);
                      else

     layer =
this.runtime.getLayerByName(lay
erparam);
                      if
(layer)

     ret.set_float(layer.canvas
ToLayer(this.touches[0].x,
this.touches[0].y, false));
                      else

     ret.set_float(0);
                }
           }
           else

     ret.set_float(0);
     };
     Exps.prototype.YAt =
function (ret, index,
layerparam)
     {
           index =
Math.floor(index);
           if (index < 0 ||
index >= this.touches.length)
           {

     ret.set_float(0);
                return;
           }
           var layer, oldScale,
oldZoomRate, oldParallaxY,
oldAngle;
           if
(cr.is_undefined(layerparam))
           {
                layer =
this.runtime.getLayerByNumber(0
);
                oldScale =
layer.scale;
                oldZoomRate =
layer.zoomRate;
```

```
                oldParallaxY =
layer.parallaxY;
                oldAngle =
layer.angle;
                layer.scale =
this.runtime.running_layout.sca
le;
                layer.zoomRate
= 1.0;
                layer.parallaxY
= 1.0;
                layer.angle =
this.runtime.running_layout.ang
le;

     ret.set_float(layer.canvas
ToLayer(this.touches[index].x,
this.touches[index].y, false));
                layer.scale =
oldScale;
                layer.zoomRate
= oldZoomRate;
                layer.parallaxY
= oldParallaxY;
                layer.angle =
oldAngle;
           }
           else
           {
                if
(cr.is_number(layerparam))
                     layer =
this.runtime.getLayerByNumber(l
ayerparam);
                else
                     layer =
this.runtime.getLayerByName(lay
erparam);
                if (layer)

     ret.set_float(layer.canvas
ToLayer(this.touches[index].x,
this.touches[index].y, false));
                else

     ret.set_float(0);
           }
     };
     Exps.prototype.YForID =
function (ret, id, layerparam)
     {
           var index =
this.findTouch(id);
           if (index < 0)
           {
     ret.set_float(0);
                return;
           }
           var touch =
this.touches[index];
           var layer, oldScale,
oldZoomRate, oldParallaxY,
oldAngle;
           if
(cr.is_undefined(layerparam))
           {
                layer =
this.runtime.getLayerByNumber(0
);
                oldScale =
layer.scale;
                oldZoomRate =
layer.zoomRate;
                oldParallaxY =
layer.parallaxY;
                oldAngle =
layer.angle;
                layer.scale =
this.runtime.running_layout.sca
le;
                layer.zoomRate
= 1.0;
                layer.parallaxY
= 1.0;
                layer.angle =
this.runtime.running_layout.ang
le;

     ret.set_float(layer.canvas
ToLayer(touch.x, touch.y,
false));
                layer.scale =
oldScale;
                layer.zoomRate
= oldZoomRate;
                layer.parallaxY
= oldParallaxY;
                layer.angle =
oldAngle;
           }
           else
           {
                if
(cr.is_number(layerparam))
                     layer =
this.runtime.getLayerByNumber(l
ayerparam);
                else
```

```
                        layer =
this.runtime.getLayerByName(lay
erparam);
                if (layer)

     ret.set_float(layer.canvas
ToLayer(touch.x, touch.y,
false));
                else

     ret.set_float(0);
          }
     };
     Exps.prototype.AbsoluteX =
function (ret)
     {
          if
(this.touches.length)

     ret.set_float(this.touches
[0].x);
          else

     ret.set_float(0);
     };
     Exps.prototype.AbsoluteXAt
= function (ret, index)
     {
          index =
Math.floor(index);
          if (index < 0 ||
index >= this.touches.length)
          {

     ret.set_float(0);
                return;
          }

     ret.set_float(this.touches
[index].x);
     };
     Exps.prototype.AbsoluteXFo
rID = function (ret, id)
     {
          var index =
this.findTouch(id);
          if (index < 0)
          {

     ret.set_float(0);
                return;
          }
          var touch =
this.touches[index];

     ret.set_float(touch.x);
     };
     Exps.prototype.AbsoluteY =
function (ret)
     {
          if
(this.touches.length)

     ret.set_float(this.touches
[0].y);
          else

     ret.set_float(0);
     };
     Exps.prototype.AbsoluteYAt
= function (ret, index)
     {
          index =
Math.floor(index);
          if (index < 0 ||
index >= this.touches.length)
          {

     ret.set_float(0);
                return;
          }

     ret.set_float(this.touches
[index].y);
     };
     Exps.prototype.AbsoluteYFo
rID = function (ret, id)
     {
          var index =
this.findTouch(id);
          if (index < 0)
          {

     ret.set_float(0);
                return;
          }
          var touch =
this.touches[index];

     ret.set_float(touch.y);
     };
     Exps.prototype.SpeedAt =
function (ret, index)
     {
          index =
Math.floor(index);
          if (index < 0 ||
index >= this.touches.length)
          {

     ret.set_float(0);
                return;
```

```
		}
		var t = 
this.touches[index];
		var dist = 
cr.distanceTo(t.x, t.y, 
t.lastx, t.lasty);
		var timediff = 
(t.time - t.lasttime) / 1000;
		if (timediff === 0)

	ret.set_float(0);
		else

	ret.set_float(dist / 
timediff);
	};
	Exps.prototype.SpeedForID 
= function (ret, id)
	{
		var index = 
this.findTouch(id);
		if (index < 0)
		{

	ret.set_float(0);
			return;
		}
		var touch = 
this.touches[index];
		var dist = 
cr.distanceTo(touch.x, touch.y, 
touch.lastx, touch.lasty);
		var timediff = 
(touch.time - touch.lasttime) / 
1000;
		if (timediff === 0)

	ret.set_float(0);
		else

	ret.set_float(dist / 
timediff);
	};
	Exps.prototype.AngleAt = 
function (ret, index)
	{
		index = 
Math.floor(index);
		if (index < 0 || 
index >= this.touches.length)
		{

	ret.set_float(0);
			return;
		}
		var t = 
this.touches[index];

	ret.set_float(cr.to_degree
s(cr.angleTo(t.lastx, t.lasty, 
t.x, t.y)));
	};
	Exps.prototype.AngleForID 
= function (ret, id)
	{
		var index = 
this.findTouch(id);
		if (index < 0)
		{

	ret.set_float(0);
			return;
		}
		var touch = 
this.touches[index];

	ret.set_float(cr.to_degree
s(cr.angleTo(touch.lastx, 
touch.lasty, touch.x, 
touch.y)));
	};
	Exps.prototype.Alpha = 
function (ret)
	{

	ret.set_float(this.getAlph
a());
	};
	Exps.prototype.Beta = 
function (ret)
	{

	ret.set_float(this.getBeta
());
	};
	Exps.prototype.Gamma = 
function (ret)
	{

	ret.set_float(this.getGamm
a());
	};
	Exps.prototype.Acceleratio
nXWithG = function (ret)
	{

	ret.set_float(this.acc_g_x
);
	};
	Exps.prototype.Acceleratio
nYWithG = function (ret)
```

```
	{

	ret.set_float(this.acc_g_y
);
	};
	Exps.prototype.Acceleratio
nZWithG = function (ret)
	{

	ret.set_float(this.acc_g_z
);
	};
	Exps.prototype.Acceleratio
nX = function (ret)
	{

	ret.set_float(this.acc_x);
	};
	Exps.prototype.Acceleratio
nY = function (ret)
	{

	ret.set_float(this.acc_y);
	};
	Exps.prototype.Acceleratio
nZ = function (ret)
	{

	ret.set_float(this.acc_z);
	};
	Exps.prototype.TouchIndex
= function (ret)
	{

	ret.set_int(this.trigger_i
ndex);
	};
	Exps.prototype.TouchID =
function (ret)
	{

	ret.set_float(this.trigger
_id);
	};
	pluginProto.exps = new
Exps();
}());
;
;
cr.behaviors.DragnDrop =
function(runtime)
{
	this.runtime = runtime;
	var self = this;
	if
(!this.runtime.isDomFree)
	{

	jQuery(document).mousemove
(
		function(info)
{

	self.onMouseMove(info);
		}
	);

	jQuery(document).mousedown
(
		function(info)
{

	self.onMouseDown(info);
		}
	);

	jQuery(document).mouseup(
		function(info)
{

	self.onMouseUp(info);
		}
	);
	}
	var elem =
(this.runtime.fullscreen_mode >
0) ? document :
this.runtime.canvas;
	if
(this.runtime.isDirectCanvas)
		elem =
window["Canvas"];
	else if
(this.runtime.isCocoonJs)
		elem = window;
	if
(window.navigator["pointerEnabl
ed"])
	{

	elem.addEventListener("poi
nterdown",
		function(info)
{

	self.onPointerStart(info);
		},
		false
	);

	elem.addEventListener("poi
ntermove",
```

```
                function(info)
{

     self.onPointerMove(info);
                },
                false
           );

     elem.addEventListener("poi
nterup",
                function(info)
{

     self.onPointerEnd(info);
                },
                false
           );

     elem.addEventListener("poi
ntercancel",
                function(info)
{

     self.onPointerEnd(info);
                },
                false
           );
     }
     else if
(window.navigator["msPointerEna
bled"])
     {

     elem.addEventListener("MSP
ointerDown",
                function(info)
{

     self.onPointerStart(info);
                },
                false
           );

     elem.addEventListener("MSP
ointerMove",
                function(info)
{

     self.onPointerMove(info);
                },
                false
           );

     elem.addEventListener("MSP
ointerUp",
                function(info)
{

     self.onPointerEnd(info);
                },
                false
           );

     elem.addEventListener("MSP
ointerCancel",
                function(info)
{

     self.onPointerEnd(info);
                },
                false
           );
     }
     else
     {

     elem.addEventListener("tou
chstart",
                function(info)
{

     self.onTouchStart(info);
                },
                false
           );

     elem.addEventListener("tou
chmove",
                function(info)
{

     self.onTouchMove(info);
                },
                false
           );

     elem.addEventListener("tou
chend",
                function(info)
{

     self.onTouchEnd(info);
                },
                false
           );

     elem.addEventListener("tou
chcancel",
                function(info)
{
```

```
	self.onTouchEnd(info);
		},
		false
	);
	}
};
(function ()
{
	var behaviorProto =
cr.behaviors.DragnDrop.prototyp
e;
	var dummyoffset = {left:
0, top: 0};
	function
GetDragDropBehavior(inst)
	{
		var i, len;
		for (i = 0, len =
inst.behavior_insts.length; i <
len; i++)
		{
			if
(inst.behavior_insts[i]
instanceof
behaviorProto.Instance)
				return
inst.behavior_insts[i];
		}
		return null;
	};
	behaviorProto.onMouseDown
= function (info)
	{
		if (info.which !==
1)
			return;
	// not left mouse button

	this.onInputDown("leftmous
e", info.pageX, info.pageY);
	};
	behaviorProto.onMouseMove
= function (info)
	{
		if (info.which !==
1)
			return;
	// not left mouse button

	this.onInputMove("leftmous
e", info.pageX, info.pageY);
	};
	behaviorProto.onMouseUp =
function (info)
	{
		if (info.which !==
1)
			return;
	// not left mouse button

	this.onInputUp("leftmouse"
);
	};
	behaviorProto.onTouchStart
= function (info)
	{
		if
(info.preventDefault)

	info.preventDefault();
		var i, len, t, id;
		for (i = 0, len =
info.changedTouches.length; i <
len; i++)
		{
			t =
info.changedTouches[i];
			id =
t.identifier;

	this.onInputDown(id ?
id.toString() : "<none>",
t.pageX, t.pageY);
		}
	};
	behaviorProto.onTouchMove
= function (info)
	{
		if
(info.preventDefault)

	info.preventDefault();
		var i, len, t, id;
		for (i = 0, len =
info.changedTouches.length; i <
len; i++)
		{
			t =
info.changedTouches[i];
			id =
t.identifier;

	this.onInputMove(id ?
id.toString() : "<none>",
t.pageX, t.pageY);
		}
	};
	behaviorProto.onTouchEnd =
function (info)
	{
```

```
            if
(info.preventDefault)

      info.preventDefault();
            var i, len, t, id;
            for (i = 0, len =
info.changedTouches.length; i <
len; i++)
            {
                  t =
info.changedTouches[i];
                  id =
t.identifier;

      this.onInputUp(id ?
id.toString() : "<none>");
            }
      };
      behaviorProto.onPointerSta
rt = function (info)
      {
            if
(info["pointerType"] ===
info["MSPOINTER_TYPE_MOUSE"] ||
info["pointerType"] ===
"mouse")
                  return;
            if
(info.preventDefault)

      info.preventDefault();

      this.onInputDown(info["poi
nterId"].toString(),
info.pageX, info.pageY);
      };
      behaviorProto.onPointerMov
e = function (info)
      {
            if
(info["pointerType"] ===
info["MSPOINTER_TYPE_MOUSE"] ||
info["pointerType"] ===
"mouse")
                  return;
            if
(info.preventDefault)

      info.preventDefault();

      this.onInputMove(info["poi
nterId"].toString(),
info.pageX, info.pageY);
      };
      behaviorProto.onPointerEnd
= function (info)
      {
            if
(info["pointerType"] ===
info["MSPOINTER_TYPE_MOUSE"] ||
info["pointerType"] ===
"mouse")
                  return;
            if
(info.preventDefault)

      info.preventDefault();

      this.onInputUp(info["point
erId"].toString());
      };
      behaviorProto.onInputDown
= function (src, pageX, pageY)
      {
            var offset =
this.runtime.isDomFree ?
dummyoffset :
jQuery(this.runtime.canvas).off
set();
            var x = pageX -
offset.left;
            var y = pageY -
offset.top;
            var lx, ly, topx,
topy;
            var arr =
this.my_instances.valuesRef();
            var i, len, b, inst,
topmost = null;
            for (i = 0, len =
arr.length; i < len; i++)
            {
                  inst = arr[i];
                  b =
GetDragDropBehavior(inst);
                  if (!b.enabled
|| b.dragging)
                        continue;
            // don't consider
disabled or already-dragging
instances
                  lx =
inst.layer.canvasToLayer(x, y,
true);
                  ly =
inst.layer.canvasToLayer(x, y,
false);

      inst.update_bbox();
                  if
(!inst.contains_pt(lx, ly))
```

```
                        continue;
            // don't consider
instances not over this point
                if (!topmost)
                {
                        topmost =
inst;
                        topx =
lx;
                        topy =
ly;
                        continue;
                }
                if
(inst.layer.index >
topmost.layer.index)
                {
                        topmost =
inst;
                        topx =
lx;
                        topy =
ly;
                        continue;
                }
                if
(inst.layer.index ===
topmost.layer.index &&
inst.get_zindex() >
topmost.get_zindex())
                {
                        topmost =
inst;
                        topx =
lx;
                        topy =
ly;
                        continue;
                }
            }
            if (topmost)

      GetDragDropBehavior(topmos
t).onDown(src, topx, topy);
      };
      behaviorProto.onInputMove
= function (src, pageX, pageY)
      {
            var offset =
this.runtime.isDomFree ?
dummyoffset :
jQuery(this.runtime.canvas).off
set();
            var x = pageX -
offset.left;
            var y = pageY -
offset.top;
            var lx, ly;
            var arr =
this.my_instances.valuesRef();
            var i, len, b, inst;
            for (i = 0, len =
arr.length; i < len; i++)
            {
                  inst = arr[i];
                  b =
GetDragDropBehavior(inst);
                  if (!b.enabled
|| !b.dragging || (b.dragging
&& b.dragsource !== src))
                        continue;
            // don't consider
disabled, not-dragging, or
dragging by other sources
                  lx =
inst.layer.canvasToLayer(x, y,
true);
                  ly =
inst.layer.canvasToLayer(x, y,
false);
                  b.onMove(lx,
ly);
            }
      };
      behaviorProto.onInputUp =
function (src)
      {
            var arr =
this.my_instances.valuesRef();
            var i, len, b, inst;
            for (i = 0, len =
arr.length; i < len; i++)
            {
                  inst = arr[i];
                  b =
GetDragDropBehavior(inst);
                  if (b.dragging
&& b.dragsource === src)
                        b.onUp();
            }
      };
      behaviorProto.Type =
function(behavior, objtype)
      {
            this.behavior =
behavior;
            this.objtype =
objtype;
            this.runtime =
behavior.runtime;
      };
```

```
	var behtypeProto =
behaviorProto.Type.prototype;
	behtypeProto.onCreate =
function()
	{
	};
	behaviorProto.Instance =
function(type, inst)
	{
		this.type = type;
		this.behavior =
type.behavior;
		this.inst = inst;
				//
associated object instance to
modify
		this.runtime =
type.runtime;
	};
	var behinstProto =
behaviorProto.Instance.prototyp
e;
	behinstProto.onCreate =
function()
	{
		this.dragging =
false;
		this.dx = 0;
		this.dy = 0;
		this.dragsource =
"<none>";
		this.axes =
this.properties[0];
		this.enabled =
(this.properties[1] !== 0);
	};
	behinstProto.saveToJSON =
function ()
	{
		return { "enabled":
this.enabled };
	};
	behinstProto.loadFromJSON
= function (o)
	{
		this.enabled =
o["enabled"];
		this.dragging =
false;
	};
	behinstProto.onDown =
function(src, x, y)
	{
		this.dx = x -
this.inst.x;
		this.dy = y -
this.inst.y;
		this.dragging =
true;
		this.dragsource =
src;

	this.runtime.isInUserInput
Event = true;

	this.runtime.trigger(cr.be
haviors.DragnDrop.prototype.cnd
s.OnDragStart, this.inst);

	this.runtime.isInUserInput
Event = false;
	};
	behinstProto.onMove =
function(x, y)
	{
		var newx = x -
this.dx;
		var newy = y -
this.dy;
		if (this.axes === 0)
		// both
		{
			if (this.inst.x
!== newx || this.inst.y !==
newy)
			{

	this.inst.x = newx;

	this.inst.y = newy;

	this.inst.set_bbox_changed
();
			}
		}
		else if (this.axes
=== 1)	// horizontal
		{
			if (this.inst.x
!== newx)
			{

	this.inst.x = newx;

	this.inst.set_bbox_changed
();
			}
		}
		else if (this.axes
=== 2)	// vertical
		{
```

```
                if (this.inst.y
!== newy)
                {
     this.inst.y = newy;

     this.inst.set_bbox_changed
();
                }
           }
     };
     behinstProto.onUp =
function()
     {
           this.dragging =
false;

     this.runtime.isInUserInput
Event = true;

     this.runtime.trigger(cr.be
haviors.DragnDrop.prototype.cnd
s.OnDrop, this.inst);

     this.runtime.isInUserInput
Event = false;
     };
     behinstProto.tick =
function ()
     {
     };
     function Cnds() {};
     Cnds.prototype.IsDragging
= function ()
     {
           return
this.dragging;
     };
     Cnds.prototype.OnDragStart
= function ()
     {
           return true;
     };
     Cnds.prototype.OnDrop =
function ()
     {
           return true;
     };
     Cnds.prototype.IsEnabled =
function ()
     {
           return
!!this.enabled;
     };
     behaviorProto.cnds = new
Cnds();
     function Acts() {};
     Acts.prototype.SetEnabled
= function (s)
     {
           this.enabled = (s
!== 0);
           if (!this.enabled)
                 this.dragging =
false;
     };
     Acts.prototype.Drop =
function ()
     {
           if (this.dragging)
                 this.onUp();
     };
     behaviorProto.acts = new
Acts();
     function Exps() {};
     behaviorProto.exps = new
Exps();
}());
;
;
cr.behaviors.Flash =
function(runtime)
{
     this.runtime = runtime;
};
(function ()
{
     var behaviorProto =
cr.behaviors.Flash.prototype;
     behaviorProto.Type =
function(behavior, objtype)
     {
           this.behavior =
behavior;
           this.objtype =
objtype;
           this.runtime =
behavior.runtime;
     };
     var behtypeProto =
behaviorProto.Type.prototype;
     behtypeProto.onCreate =
function()
     {
     };
     behaviorProto.Instance =
function(type, inst)
     {
           this.type = type;
           this.behavior =
type.behavior;
```

```
		this.inst = inst;
		//
associated object instance to
modify
		this.runtime =
type.runtime;
	};
	var behinstProto =
behaviorProto.Instance.prototyp
e;
	behinstProto.onCreate =
function()
	{
		this.ontime = 0;
		this.offtime = 0;
		this.stage = 0;
		// 0 = on, 1 = off
		this.stagetimeleft =
0;
		this.timeleft = 0;
	};
	behinstProto.saveToJSON =
function ()
	{
		return {
			"ontime":
this.ontime,
			"offtime":
this.offtime,
			"stage":
this.stage,

	"stagetimeleft":
this.stagetimeleft,
			"timeleft":
this.timeleft
		};
	};
	behinstProto.loadFromJSON
= function (o)
	{
		this.ontime =
o["ontime"];
		this.offtime =
o["offtime"];
		this.stage =
o["stage"];
		this.stagetimeleft =
o["stagetimeleft"];
		this.timeleft =
o["timeleft"];
	};
	behinstProto.tick =
function ()
	{
		if (this.timeleft <=
0)
			return;
	// not flashing
		var dt =
this.runtime.getDt(this.inst);
		this.timeleft -= dt;
		if (this.timeleft <=
0)
		{
			this.timeleft =
0;

	this.inst.visible = true;

	this.runtime.redraw =
true;

	this.runtime.trigger(cr.be
haviors.Flash.prototype.cnds.On
FlashEnded, this.inst);
			return;
		}
		this.stagetimeleft -
= dt;
		if
(this.stagetimeleft <= 0)
		{
			if (this.stage
=== 0)
			{

	this.inst.visible = false;

	this.stage = 1;

	this.stagetimeleft +=
this.offtime;
			}
			else
			{

	this.inst.visible = true;

	this.stage = 0;

	this.stagetimeleft +=
this.ontime;
			}

	this.runtime.redraw =
true;
		}
	};
	function Cnds() {};
```

```
    Cnds.prototype.IsFlashing = function ()
    {
        return this.timeleft > 0;
    };
    Cnds.prototype.OnFlashEnded = function ()
    {
        return true;
    };
    behaviorProto.cnds = new Cnds();
    function Acts() {};
    Acts.prototype.Flash = function (on_, off_, dur_)
    {
        this.ontime = on_;
        this.offtime = off_;
        this.stage = 1;
    // always start off
        this.stagetimeleft = off_;
        this.timeleft = dur_;
        this.inst.visible = false;
        this.runtime.redraw = true;
    };
    Acts.prototype.StopFlashing = function ()
    {
        this.timeleft = 0;
        this.inst.visible = true;
        this.runtime.redraw = true;
        return;
    };
    behaviorProto.acts = new Acts();
    function Exps() {};
    behaviorProto.exps = new Exps();
}());
cr.getProjectModel = function() { return [
    null,
    null,
    [
    [
        cr.plugins_.Audio,
        true,
        false,
        false,
        false,
        false,
        false,
        false,
        false,
        false
    ]
  , [
        cr.plugins_.Sprite,
        false,
        true,
        true,
        true,
        true,
        true,
        true,
        true,
        false
    ]
  , [
        cr.plugins_.Text,
        false,
        true,
        true,
        true,
        true,
        true,
        true,
        true,
        false
    ]
  , [
        cr.plugins_.TextBox,
        false,
        true,
        true,
        true,
        false,
        false,
        false,
        false,
        false
    ]
  , [
        cr.plugins_.Touch,
        true,
        false,
        false,
        false,
        false,
        false,
        false,
        false,
        false
    ]
    ],
```

```
[
[
        "t0",
        cr.plugins_.Sprite,
        false,
        [],
        0,
        0,
        null,
        [
            [
            "Default",
            5,
            false,
            1,
            0,
            false,
425477981341332,
            [
["images/home-sheet0.png",
786611, 0, 0, 1279, 768, 1,
0.5003909468650818,
0.5,[],[],1]
            ]
            ]
        ],
        [
        ],
        false,
        false,
        5840415258166885,
        [],
        null
    ]
,   [
        "t1",
        cr.plugins_.Sprite,
        false,
        [],
        0,
        0,
        null,
        [
            [
            "Default",
            5,
            false,
            1,
            0,
            false,
2416608716326732,
            [
["images/btn_play-
sheet0.png", 8061, 0, 0, 234,
234, 1, 0.5, 0.5,[],[-
0.3547009825706482,-
0.3547009825706482,0,-
0.5,0.3547009825706482,-
0.3547009825706482,0.5,0,0.3547
009825706482,0.3547009825706482
,0,0.4957259893417358,-
0.3504270017147064,0.3504269719
12384,-0.4957264959812164,0],0]
            ]
            ]
        ],
        [
        ],
        false,
        false,
        7677619652895739,
        [],
        null
    ]
,   [
        "t2",
        cr.plugins_.Touch,
        false,
        [],
        0,
        0,
        null,
        null,
        [
        ],
        false,
        false,
        880647178207692,
        [],
        null
        ,[1]
    ]
,   [
        "t3",
        cr.plugins_.Sprite,
        false,
        [],
        0,
        0,
        null,
        [
            [
            "Default",
            5,
            false,
            1,
            0,
```

```
                false,
     1882400029739704,
                [
     ["images/bggame-
sheet0.png", 1144910, 0, 0,
1279, 768, 1,
0.5003909468650818,
0.5,[],[],1]
                ]
                ]
           ],
           [
           ],
           false,
           false,
           6359182693699608,
           [],
           null
     ]
,    [
           "t4",
           cr.plugins_.Sprite,
           false,
           [],
           0,
           0,
           null,
           [
                [
                "Default",
                5,
                false,
                1,
                0,
                false,
     6971045830002355,
                [
     ["images/lyrapel-
sheet0.png", 27667, 0, 0, 348,
467, 1, 0.5,
0.5010706782341003,[],[-
0.4367815852165222,-
0.4539614915847778,0,-
0.5010706782341003,0.4224140048
027039,-
0.4432548880577087,0.4770119786
262512,-
0.002141684293746948,0.43678200
24490356,0.4518203139305115,0,0
.4796572923660278,-
0.4367815852165222,0.4518203139
305115,-0.4741379022598267,-
0.002141684293746948],0]
                ]
                ]
           ],
           [
           ],
           false,
           false,
           7683529011140795,
           [],
           null
     ]
,    [
           "t5",
           cr.plugins_.Sprite,
           false,
           [],
           0,
           0,
           null,
           [
                [
                "Default",
                5,
                false,
                1,
                0,
                false,
     1097045232462277,
                [
     ["images/fldapel-
sheet0.png", 286, 0, 0, 400,
100, 1, 0.5, 0.5,[],[],1]
                ]
                ]
           ],
           [
           ],
           false,
           false,
           317323620133282,
           [],
           null
     ]
,    [
           "t6",
           cr.plugins_.Text,
           false,
     [3040197777531715,62818561
29631571,6184584796420218],
           1,
           0,
```

```
            null,
            null,
            [
            [
                  "DragDrop",
      cr.behaviors.DragnDrop,
      5879110412252368
            ]
            ],
            false,
            false,
            6856476541981364,
            [],
            null
      ]
,     [
            "t7",
            cr.plugins_.Sprite,
            false,
            [1754136777998757],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  0,
                  false,
                  1,
                  0,
                  false,
      5382256678376886,
                  [
      ["images/trgapel-
sheet1.png", 1666, 0, 0, 63,
67, 1, 0.5079365372657776,
0.5074626803398132,[],[-
0.3809525370597839,-
0.388059675693512,-
0.01587355136871338,-
0.5074626803398132,0.3809524774
551392,-
0.4029846787452698,0.4603174328
804016,-
0.01492568850517273,0.460317432
8804016,0.46268630027771,-
0.01587355136871338,0.462686300
27771,-
0.4126984477043152,0.4029853343
963623,-0.3809525370597839,-
0.01492568850517273],0],
      ["images/trgapel-
sheet0.png", 1804, 1, 1, 65,
92, 1, 0.5076923370361328,
0.5,[],[-0.4769231379032135,-
0.4782609045505524,-
0.01538434624671936,-
0.5,0.3692306876182556,-
0.4130434989929199,0.4769226908
683777,0,0.1076926589012146,0.2
282609939575195,-
0.01538434624671936,0.217391014
0991211,-
0.461538553237915,0.46739101409
91211,-
0.5076923370361328,0],0],
      ["images/trgapel-
sheet2.png", 1501, 0, 0, 63,
67, 1, 0.5079365372657776,
0.5074626803398132,[],[-
0.3650795221328735,-
0.3731346726417542,-
0.01587355136871338,-
0.5074626803398132,0.3650794625
282288,-
0.388059675693512,0.47619044780
7312,-
0.01492568850517273,0.380952477
4551392,0.3880593180656433,-
0.01587355136871338,0.477612316
608429,-
0.3650795221328735,0.3582093119
621277,-0.4920635223388672,-
0.01492568850517273],0],
      ["images/trgapel-
sheet0.png", 1804, 67, 1, 16,
92, 1, 0.5, 0.5,[],[-0.375,-
0.4782609045505524,0,-
0.5,0.375,-
0.4782609045505524,0.5,0,0.375,
0.4782609939575195,0,0.48913002
01416016,-
0.3125,0.4673910140991211,-
0.5,0],0]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
            9747953063296485,
            [],
            null
```

```
      ]
,     [
            "t8",
            cr.plugins_.Sprite,
            false,
            [],
            1,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      2333559381183121,
                  [

      ["images/lyrnextapel-
sheet0.png", 66681, 0, 0, 1062,
768, 1, 0.5, 0.5,[],[-
0.4161958992481232,-
0.3841150104999542,0,-
0.5,0.3851220011711121,-
0.3411459922790527,0.3851220011
711121,0.3411459922790527,0,0.5
,-
0.4161958992481232,0.3841149806
976318],0]
                  ]
                  ]
            ],
            [
            [
                  "Flash",

      cr.behaviors.Flash,

      4519254015321678
            ]
            ],
            false,
            false,
            5369679973524416,
            [],
            null
      ]
,     [
            "t9",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      4319131561567745,
                  [

      ["images/lyrnanas-
sheet0.png", 43915, 0, 0, 348,
467, 1, 0.5,
0.5010706782341003,[],[-
0.4367815852165222,-
0.4539614915847778,0,-
0.5010706782341003,0.4224140048
027039,-
0.4432548880577087,0.4770119786
262512,-
0.002141684293746948,0.43678200
24490356,0.4518203139305115,0,0
.4796572923660278,-
0.4367815852165222,0.4518203139
305115,-0.4741379022598267,-
0.002141684293746948],0]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
            3756653918776747,
            [],
            null
      ]
,     [
            "t10",
            cr.plugins_.Text,
            false,

      [1770899982106417,10458239
15022196,1698163072252259],
            1,
            0,
            null,
            null,
            [
            [
                  "DragDrop",
```

```
      cr.behaviors.DragnDrop,

      8826203057561886
            ]
            ],
            false,
            false,
            7698680524069486,
            [],
            null
      ]
,     [
            "t11",
            cr.plugins_.Sprite,
            false,
            [9850966054456417],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  0,
                  false,
                  1,
                  0,
                  false,

      8118244815754752,
                  [

      ["images/trgnanas-
sheet2.png", 1301, 1, 1, 61,
70, 1, 0.5081967115402222,
0.5,[],[-0.4754098057746887,-
0.4714286029338837,-
0.01639372110366821,-
0.4857142865657806,0.3934422731
399536,-
0.4142856895923615,0.4918032884
597778,0,0.4426233172416687,0.4
571430087089539,-
0.01639372110366821,-
0.3142859935760498,-
0.4590164124965668,0.4571430087
089539,-
0.5081967115402222,0],0],

      ["images/trgnanas-
sheet0.png", 2616, 1, 1, 66,
70, 0, 0.5, 0.5,[],[-
0.3787879943847656,-
0.3857139945030212,0,-
0.5,0.3939390182495117,-
0.4000000059604645,0.4545450210
571289,0,0.4696969985961914,0.4
714289903640747,0,0.47142899036
40747,-
0.4090909063816071,0.4142860174
179077,-
0.3787879943847656,0],0],

      ["images/trgnanas-
sheet2.png", 1301, 63, 1, 61,
70, 1, 0.5081967115402222,
0.5,[],[-0.4754098057746887,-
0.4714286029338837,-
0.01639372110366821,-
0.4857142865657806,0.3934422731
399536,-
0.4142856895923615,0.4918032884
597778,0,0.4426233172416687,0.4
571430087089539,-
0.01639372110366821,-
0.3142859935760498,-
0.4590164124965668,0.4571430087
089539,-
0.5081967115402222,0],0],

      ["images/trgnanas-
sheet1.png", 1663, 0, 0, 66,
70, 1, 0.5, 0.5,[],[-
0.3787879943847656,-
0.3857139945030212,0,-
0.5,0.3939390182495117,-
0.4000000059604645,0.4545450210
571289,0,0.4696969985961914,0.4
714289903640747,0,0.47142899036
40747,-
0.4090909063816071,0.4142860174
179077,-
0.3787879943847656,0],0],

      ["images/trgnanas-
sheet0.png", 2616, 68, 1, 59,
70, 1, 0.508474588394165,
0.5,[],[-0.3898305892944336,-
0.4000000059604645,-
0.01694959402084351,-
0.5,0.3728814125061035,-
0.4000000059604645,0.4067794084
54895,0,0.3728814125061035,0.39
99999761581421,-
0.01694959402084351,0.485714018
3448792,-
0.3898305892944336,0.3999999761
581421,-
0.2881355881690979,0],0]
                  ]
                  ]
            ],
```

```
            [
            ],
            false,
            false,
            2700925321776098,
            [],
            null
      ]
,     [
            "t12",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,
      3600402875484969,
                  [
      ["images/fldnanas-
sheet0.png", 158, 0, 0, 250,
50, 1, 0.5, 0.5,[],[],4]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
            7719522902219564,
            [],
            null
      ]
,     [
            "t13",
            cr.plugins_.Sprite,
            false,
            [],
            1,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,
      1435481829342922,
                  [
      ["images/lyrnextapel-
sheet0.png", 66681, 0, 0, 1062,
768, 1, 0.5, 0.5,[],[-
0.4161958992481232,-
0.3841150104999542,0,-
0.5,0.3851220011711121,-
0.3411459922790527,0.3851220011
711121,0.3411459922790527,0,0.5
,-
0.4161958992481232,0.3841149806
976318],0]
                  ]
                  ]
            ],
            [
            [
                  "Flash",
      cr.behaviors.Flash,
      9866409438454454
            ]
            ],
            false,
            false,
            6740515462359022,
            [],
            null
      ]
,     [
            "t14",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,
      9158612841462232,
                  [
      ["images/lyrmangga-
```

sheet0.png", 41735, 0, 0, 348,
467, 1, 0.5,
0.5010706782341003,[],[-
0.4367815852165222,-
0.4539614915847778,0,-
0.5010706782341003,0.4224140048
027039,-
0.4432548880577087,0.4770119786
262512,-
0.002141684293746948,0.43678200
24490356,0.4518203139305115,0,0
.4796572923660278,-
0.4367815852165222,0.4518203139
305115,-0.4741379022598267,-
0.002141684293746948],0]
]
]
],
[
],
false,
false,
2866995194144282,
[],
null
]
, [
"t15",
cr.plugins_.Text,
false,
[1088753767598059,77666669
98521413,3766110459897438],
1,
0,
null,
null,
[
[
"DragDrop",
cr.behaviors.DragnDrop,
9187704762661678
]
],
false,
false,
6595260274642995,
[],
null
]
, [
"t16",
cr.plugins_.Sprite,
false,
[],
0,
0,
null,
[
[
"Default",
5,
false,
1,
0,
false,
7734843672483473,
[
["images/fldmangga-
sheet0.png", 286, 0, 0, 300,
50, 1, 0.5, 0.5,[],[],0]
]
]
],
[
],
false,
false,
9990018426802101,
[],
null
]
, [
"t17",
cr.plugins_.Sprite,
false,
[2010510121261972],
0,
0,
null,
[
[
"Default",
0,
false,
1,
0,
false,
8266079252370754,
[
["images/trgmangga-
sheet0.png", 4549, 1, 1, 95,
67, 1, 0.5052631497383118,
0.5074626803398132,[],[-
0.4842105507850647,-
0.4776119887828827,-

0.0105261504650116,-
0.4776119887828827,0.4421048760
414124,-
0.4328357875347138,0.4947368502
616882,-
0.01492568850517273,0.463157832
6225281,0.4477612972259522,-
0.0105261504650116,0.4776123166
08429,-
0.4736842513084412,0.4477612972
259522,-0.5052631497383118,-
0.01492568850517273],0],

["images/trgmangga-
sheet0.png", 4549, 1, 69, 63,
67, 1, 0.5079365372657776,
0.5074626803398132,[],[-
0.3809525370597839,-
0.388059675693512,-
0.01587355136871338,-
0.5074626803398132,0.3809524774
551392,-
0.4029846787452698,0.4603174328
804016,-
0.01492568850517273,0.460317432
8804016,0.46268630027771,-
0.01587355136871338,0.462686300
27771,-
0.4126984477043152,0.4029853343
963623,-0.3809525370597839,-
0.01492568850517273],0],

["images/trgmangga-
sheet0.png", 4549, 129, 94, 59,
67, 1, 0.508474588394165,
0.5074626803398132,[],[-
0.4745762944221497,-
0.4776119887828827,-
0.01694959402084351,-
0.4925372898578644,0.3898304104
804993,-
0.4179104864597321,0.4915254116
05835,-
0.01492568850517273,0.457627415
6570435,0.46268630027771,-
0.01694959402084351,-
0.3283586800098419,-
0.4576270878314972,0.4477612972
259522,-0.508474588394165,-
0.01492568850517273],0],

["images/trgmangga-
sheet0.png", 4549, 97, 1, 65,
92, 1, 0.5076923370361328,
0.5,[],[-0.3846153318881989,-
0.4130434989929199,-
0.01538434624671936,-
0.5,0.4615386724472046,-
0.4782609045505524,0.4923076629
638672,0,0.3692306876182556,0.4
130430221557617,-
0.01538434624671936,0.489130020
1416016,-
0.3846153318881989,0.4130430221
557617,-
0.461538553237915,0],0],

["images/trgmangga-
sheet0.png", 4549, 163, 1, 65,
92, 1, 0.5076923370361328,
0.5,[],[-0.3846153318881989,-
0.4130434989929199,-
0.01538434624671936,-
0.5,0.4615386724472046,-
0.4782609045505524,0.4923076629
638672,0,0.3692306876182556,0.4
130430221557617,-
0.01538434624671936,0.489130020
1416016,-
0.3846153318881989,0.4130430221
557617,-
0.461538553237915,0],0],

["images/trgmangga-
sheet0.png", 4549, 65, 94, 63,
67, 1, 0.5079365372657776,
0.5074626803398132,[],[-
0.3809525370597839,-
0.388059675693512,-
0.01587355136871338,-
0.5074626803398132,0.3809524774
551392,-
0.4029846787452698,0.4603174328
804016,-
0.01492568850517273,0.460317432
8804016,0.46268630027771,-
0.01587355136871338,0.462686300
27771,-
0.4126984477043152,0.4029853343
963623,-0.3809525370597839,-
0.01492568850517273],0]
]
]
],
[
],
false,
false,
60485051374583,
[],
null
]

```
,     [
            "t18",
            cr.plugins_.Audio,
            false,
            [],
            0,
            0,
            null,
            null,
            [
            ],
            false,
            false,
            955594947136884 6,
            [],
            null

      ,[0,0,1,1,600,600,10000,1,
5000,1]
      ]
,     [
            "t19",
            cr.plugins_.Sprite,
            false,
            [],
            1,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      1015672393580554,
                  [

      ["images/lyrnextapel-
sheet0.png", 66681, 0, 0, 1062,
768, 1, 0.5, 0.5,[],[-
0.4161958992481232,-
0.3841150104999542,0,-
0.5,0.3851220011711121,-
0.3411459922790527,0.3851220011
711121,0.3411459922790527,0,0.5
,-
0.4161958992481232,0.3841149806
976318],0]
                  ]
                  ]
            ],
            [
            [
                  "Flash",

      cr.behaviors.Flash,

      9377903770588921
            ]
            ],
            false,
            false,
            3727137882210832,
            [],
            null
      ]
,     [
            "t20",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      549783743001853,
                  [

      ["images/lyrjeruk-
sheet0.png", 21188, 0, 0, 348,
467, 1, 0.5,
0.5010706782341003,[],[-
0.4367815852165222,-
0.4539614915847778,0,-
0.5010706782341003,0.4224140048
027039,-
0.4432548880577087,0.4770119786
262512,-
0.002141684293746948,0.43678200
24490356,0.4518203139305115,0,0
.4796572923660278,-
0.4367815852165222,0.4518203139
305115,-0.4741379022598267,-
0.002141684293746948],0]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
```

```
            3572809830062199,
            [],
            null
      ]
,     [
            "t21",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      4342157794196137,
                  [

      ["images/fldnanas-
sheet0.png", 158, 0, 0, 250,
50, 1, 0.5, 0.5,[],[],4]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
            8834763518672949,
            [],
            null
      ]
,     [
            "t22",
            cr.plugins_.Text,
            false,

      [1746301997510502,60652731
00053226,1450484917192008],
            1,
            0,
            null,
            null,
            [
            [
                  "DragDrop",

      cr.behaviors.DragnDrop,

      3033971319494005
            ]
            ],
            false,
            false,
            5878527486259275,
            [],
            null
      ]
,     [
            "t23",
            cr.plugins_.Sprite,
            false,
            [4698894073034772],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  0,
                  false,
                  1,
                  0,
                  false,

      3298218617242071,
                  [

      ["images/trgjeruk-
sheet1.png", 1737, 61, 1, 34,
116, 1, 0.5,
0.5,[],[0.2058820128440857,-
0.2931029796600342,0,0.32758599
51972961,0.4411759972572327,-
0.4827586114406586,0.5,0,0.3529
409766197205,0.4568970203399658
,0,0.4913790225982666,-
0.4117647111415863,0.4741380214
691162,0.02941197156906128,0],0
],

      ["images/trgjeruk-
sheet0.png", 2326, 59, 1, 63,
67, 1, 0.5079365372657776,
0.5074626803398132,[],[-
0.3650795221328735,-
0.3731346726417542,-
0.01587355136871338,-
0.5074626803398132,0.3650794625
282288,-
0.388059675693512,0.47619044780
7312,-
0.01492568850517273,0.380952477
4551392,0.3880593180656433,-
0.01587355136871338,0.477612316
608429,-
```

0.3650795221328735,0.3582093119
621277,-0.4920635223388672,-
0.01492568850517273],0],

["images/trgjeruk-
sheet2.png", 864, 0, 0, 45, 67,
1, 0.5111111402511597,
0.5074626803398132,[],[-
0.4666667282581329,-
0.4776119887828827,0.4222218394
27948,-0.4626865684986115,-
0.133333146572113,-
0.01492568850517273,-
0.133333146572113,0.07462733983
99353,-0.0222221314907074,-
0.2686566710472107,-
0.4444444477558136,0.4477612972
259522,-0.5111111402511597,-
0.01492568850517273],0],

["images/trgjeruk-
sheet1.png", 1737, 1, 1, 59,
67, 1, 0.508474588394165,
0.5074626803398132,[],[-
0.4745762944221497,-
0.4776119887828827,-
0.01694959402084351,0.298507332
8018189,0.4576274156570435,-
0.4776119887828827,0.4915254116
05835,-
0.01492568850517273,0.457627415
6570435,0.46268630027771,-
0.01694959402084351,0.462686300
27771,-
0.4067795872688294,0.4029853343
963623,-0.508474588394165,-
0.01492568850517273],0],

["images/trgjeruk-
sheet0.png", 2326, 1, 1, 57,
92, 1, 0.5087719559669495,
0.5,[],[-0.4561403393745422,-
0.467391312122345,-
0.01754394173622131,-
0.08695700764656067,-
0.2105259597301483,-
0.06521698832511902,0.192982017
993927,0,0.4561400413513184,0.4
782609939575195,-
0.01754394173622131,0.195652008
0566406,-
0.4561403393745422,0.4673910140
991211,-
0.5087719559669495,0],0]
]
]
],
[
],
false,
false,
7664865368749329,
[],
null
]
, [
"t24",
cr.plugins_.Sprite,
false,
[],
1,
0,
null,
[
[
"Default",
5,
false,
1,
0,
false,

8668220911063052,
[

["images/lyrnextapel-
sheet0.png", 66681, 0, 0, 1062,
768, 1, 0.5, 0.5,[],[-
0.4161958992481232,-
0.3841150104999542,0,-
0.5,0.3851220011711121,-
0.3411459922790527,0.3851220011
711121,0.3411459922790527,0,0.5
,-
0.4161958992481232,0.3841149806
976318],0]
]
]
],
[
[
"Flash",

cr.behaviors.Flash,

8120018502526793
]
],
false,
false,
881726586685883,
[],

```
            null
      ]
,     [
            "t25",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      367916512074035,
                  [

      ["images/lyrsemangka-
sheet0.png", 30882, 0, 0, 348,
467, 1, 0.5,
0.5010706782341003,[],[-
0.4367815852165222,-
0.4539614915847778,0,-
0.5010706782341003,0.4224140048
027039,-
0.4432548880577087,0.4770119786
262512,-
0.002141684293746948,0.43678200
24490356,0.4518203139305115,0,0
.4796572923660278,-
0.4367815852165222,0.4518203139
305115,-0.4741379022598267,-
0.002141684293746948],0]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
            6815082121797144,
            [],
            null
      ]
,     [
            "t26",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      8874253282125503,
                  [

      ["images/fldsemangka-
sheet0.png", 294, 0, 0, 400,
50, 1, 0.5, 0.5,[],[],0]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
            564578925566669,
            [],
            null
      ]
,     [
            "t27",
            cr.plugins_.Text,
            false,

      [6994292691584671,41030724
57877527,9400717653453503],
            1,
            0,
            null,
            null,
            [
            [
                  "DragDrop",

      cr.behaviors.DragnDrop,

      1009508697496344
            ]
            ],
            false,
            false,
            6365516867078737,
            [],
            null
      ]
,     [
            "t28",
            cr.plugins_.Sprite,
```

```
            false,
            [7532813498433082],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  0,
                  false,
                  1,
                  0,
                  false,

      9918288874226586,
                  [

      ["images/trgsemangka-
sheet0.png", 6163, 1, 137, 57,
67, 1, 0.5087719559669495,
0.5074626803398132,[],[-
0.4035089612007141,-
0.4179104864597321,-
0.01754394173622131,-
0.5074626803398132,0.3684210181
236267,-
0.4029846787452698,0.4035090208
053589,-
0.01492568850517273,0.368421018
1236267,0.3880593180656433,-
0.01754394173622131,0.477612316
608429,-
0.385964959859848,0.38805931806
56433,-0.2982459664344788,-
0.01492568850517273],0],

      ["images/trgsemangka-
sheet0.png", 6163, 1, 69, 63,
67, 1, 0.5079365372657776,
0.5074626803398132,[],[-
0.3650795221328735,-
0.3731346726417542,-
0.01587355136871338,-
0.5074626803398132,0.3650794625
282288,-
0.388059675693512,0.47619044780
7312,-
0.01492568850517273,0.380952477
4551392,0.3880593180656433,-
0.01587355136871338,0.477612316
608429,-
0.3650795221328735,0.3582093119
621277,-0.4920635223388672,-
0.01492568850517273],0],

      ["images/trgsemangka-
sheet0.png", 6163, 1, 1, 95,
67, 1, 0.5052631497383118,
0.5074626803398132,[],[-
0.4842105507850647,-
0.4776119887828827,-
0.0105261504650116,-
0.4776119887828827,0.4421048760
414124,-
0.4328357875347138,0.4947368502
616882,-
0.01492568850517273,0.463157832
6225281,0.4477612972259522,-
0.0105261504650116,0.4776123166
08429,-
0.4736842513084412,0.4477612972
259522,-0.5052631497383118,-
0.01492568850517273],0],

      ["images/trgsemangka-
sheet0.png", 6163, 65, 94, 63,
67, 1, 0.5079365372657776,
0.5074626803398132,[],[-
0.3809525370597839,-
0.388059675693512,-
0.01587355136871338,-
0.5074626803398132,0.3809524774
551392,-
0.4029846787452698,0.4603174328
804016,-
0.01492568850517273,0.460317432
8804016,0.46268630027771,-
0.01587355136871338,0.462686300
27771,-
0.4126984477043152,0.4029853343
963623,-0.3809525370597839,-
0.01492568850517273],0],

      ["images/trgsemangka-
sheet0.png", 6163, 193, 94, 59,
67, 1, 0.508474588394165,
0.5074626803398132,[],[-
0.4745762944221497,-
0.4776119887828827,-
0.01694959402084351,-
0.4925372898578644,0.3898304104
804993,-
0.4179104864597321,0.4915254116
05835,-
0.01492568850517273,0.457627415
6570435,0.46268630027771,-
0.01694959402084351,-
0.3283586800098419,-
0.4576270878314972,0.4477612972
259522,-0.508474588394165,-
0.01492568850517273],0],
```

```
["images/trgsemangka-sheet0.png", 6163, 97, 1, 65, 92, 1, 0.5076923370361328, 0.5,[],[-0.3846153318881989,-0.4130434989929199,-0.01538434624671936,-0.5,0.4615386724472046,-0.4782609045505524,0.4923076629638672,0,0.3692306876182556,0.4130430221557617,-0.01538434624671936,0.4891300201416016,-0.3846153318881989,0.4130430221557617,-0.461538553237915,0],0],

["images/trgsemangka-sheet0.png", 6163, 163, 1, 57, 92, 1, 0.5087719559669495, 0.5,[],[-0.4561403393745422,-0.467391312122345,-0.01754394173622131,-0.08695700764656067,-0.2105259597301483,-0.06521698832511902,0.192982017993927,0,0.4561400413513184,0.4782609939575195,-0.01754394173622131,0.1956520080566406,-0.4561403393745422,0.4673910140991211,-0.5087719559669495,0],0],

["images/trgsemangka-sheet0.png", 6163, 129, 94, 63, 67, 1, 0.5079365372657776, 0.5074626803398132,[],[-0.3809525370597839,-0.388059675693512,-0.01587355136871338,-0.5074626803398132,0.3809524774551392,-0.4029846787452698,0.4603174328804016,-0.01492568850517273,0.4603174328804016,0.46268630027771,-0.01587355136871338,0.46268630027771,-0.4126984477043152,0.4029853343963623,-0.3809525370597839,-0.01492568850517273],0]
                ]
                ]
            ],
            [
                ],
                false,
                false,
                9856103874356889,
                [],
                null
            ]
,           [
                "t29",
                cr.plugins_.Sprite,
                false,
                [],
                1,
                0,
                null,
                [
                    [
                    "Default",
                    5,
                    false,
                    1,
                    0,
                    false,

2061692081504526,
                    [

["images/lyrnextapel-sheet0.png", 66681, 0, 0, 1062, 768, 1, 0.5, 0.5,[],[-0.4161958992481232,-0.3841150104999542,0,-0.5,0.3851220011711121,-0.3411459922790527,0.3851220011711121,0.3411459922790527,0,0.5,-0.4161958992481232,0.3841149806976318],0]
                    ]
                    ]
                ],
                [
                [
                    "Flash",

cr.behaviors.Flash,

3077870764117402
                ]
                ],
                false,
                false,
                8531724105600084,
                [],
                null
            ]
```

```
,     [
            "t30",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      5592386291134397,
                  [

      ["images/btn_tm_game1_buah
-sheet0.png", 47578, 0, 0, 446,
651, 1, 0.5,
0.5007680654525757,[],[-
0.4529148042201996,-
0.4685099720954895,0,-
0.5007680654525757,0.4551569819
450378,-
0.4700460731983185,0.5,-
0.001536071300506592,0.45291501
28364563,0.4669739603996277,0,0
.4976959228515625,-
0.4529148042201996,0.4669739603
996277,-0.4977578520774841,-
0.001536071300506592],0]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
            6744480228427497,
            [],
            null
      ]
,     [
            "t31",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      7376612609103696,
                  [

      ["images/btn_tm_game1_hewa
n-sheet0.png", 43435, 0, 0,
446, 651, 1, 0.5,
0.5007680654525757,[],[-
0.4529148042201996,-
0.4685099720954895,0,-
0.5007680654525757,0.4551569819
450378,-
0.4700460731983185,0.5,-
0.001536071300506592,0.45291501
28364563,0.4669739603996277,0,0
.4976959228515625,-
0.4529148042201996,0.4669739603
996277,-0.4977578520774841,-
0.001536071300506592],0]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
            7500725029548548,
            [],
            null
      ]
,     [
            "t32",
            cr.plugins_.Text,
            false,

      [5705195306234939,85267992
24122885,9013674748388972],
            1,
            0,
            null,
            null,
            [
            [
                  "DragDrop",

      cr.behaviors.DragnDrop,

      6095081388051722
            ]
            ],
```

```
            false,
            false,
            2811655171871849,
            [],
            null
      ]
,     [
            "t33",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      8122614673758006,
                  [

      ["images/btn_help-
sheet0.png", 38887, 0, 0, 243,
243, 1, 0.5020576119422913,
0.5020576119422913,[],[-
0.3539096117019653,-
0.3539096117019653,-
0.00411561131477356,-
0.5020576119422913,0.3539093732
833862,-
0.3580245971679688,0.4979423880
577087,-
0.00411561131477356,0.349794387
8173828,0.3497943878173828,-
0.00411561131477356,0.493827402
5917053,-
0.3539096117019653,0.3497943878
173828,-0.4979423880577087,-
0.00411561131477356],0]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
            5196737681262576,
            [],
            null
      ]
,     [
            "t34",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      887738720642432,
                  [

      ["images/help-sheet0.png",
965300, 0, 0, 1276, 768, 1,
0.5, 0.5,[],[-0.5,-0.5,0.5,-
0.5,0.4992160201072693,0.498697
9961395264,-
0.4992162883281708,0.4986979961
395264],0]
                  ]
                  ]
            ],
            [
            ],
            false,
            false,
            3025354438314305,
            [],
            null
      ]
,     [
            "t35",
            cr.plugins_.Sprite,
            false,
            [],
            0,
            0,
            null,
            [
                  [
                  "Default",
                  5,
                  false,
                  1,
                  0,
                  false,

      8933314843283856,
                  [
```

```
                                        ]
     ["images/btn_home-             ,     [
sheet0.png", 13439, 0, 0, 178,                  "t38",
178, 1, 0.5, 0.5,[],[-                          cr.plugins_.Text,
0.3539330065250397,-                            false,
0.3539330065250397,0,-                          [],
0.5,0.3539329767227173,-                        0,
0.3539330065250397,0.5,0,0.3539                 0,
329767227173,0.3539329767227173                 null,
,0,0.4943820238113403,-                         null,
0.3483150005340576,0.3483150005                 [
340576,-0.5,0],0]                               ],
                ]                               false,
                ]                               false,
          ],                                    7496116753049275,
          [                                     [],
          ],                                    null
          false,                        ]
          false,                    ,     [
          6208148389876754,                     "t39",
          [],                                   cr.plugins_.TextBox,
          null                                  false,
     ]                                          [4083465937060341],
,    [                                          0,
          "t36",                                0,
          cr.plugins_.Text,                     null,
          false,                                null,
          [],                                   [
          0,                                    ],
          0,                                    false,
          null,                                 false,
          null,                                 525499153230424,
          [                                     [],
          ],                                    null
          false,                        ]
          false,                    ,     [
          9701153025611221,                     "t40",
          [],                                   cr.plugins_.Sprite,
          null                                  false,
     ]                                          [],
,    [                                          0,
          "t37",                                0,
          cr.plugins_.Text,                     null,
          false,                                [
          [],                                         [
          0,                                          "Default",
          0,                                          5,
          null,                                       false,
          null,                                       1,
          [                                           0,
          ],                                          false,
          false,
          false,                        3217694467403391,
          8974465090976181,                           [
          [],
          null                          ["images/bghasil-
```

sheet0.png", 1813, 0, 0, 1053,
368, 1, 0.5004748106002808,
0.5,[],[-0.4881291091442108,-
0.4646739065647125,-
0.0009498000144958496,-
0.5,0.488129198551178,-
0.467391312122345,0.48717916011
8103,0.4646739959716797,-
0.0009498000144958496,0.4972829
818725586,-
0.4881291091442108,0.4646739959
716797],0]
]
]
],
[
],
false,
false,
4135019519545213,
[],
null
]
,
[
"t41",
cr.plugins_.Text,
false,
[],
0,
0,
null,
null,
[
],
false,
false,
5510876156791825,
[],
null
]
,
[
"t42",
cr.plugins_.Text,
false,
[],
0,
0,
null,
null,
[
],
false,
false,
6641519981403827,
[],
null
]
,
[
"t43",
cr.plugins_.Sprite,
false,
[],
0,
0,
null,
[
[
"Default",
5,
false,
1,
0,
false,
6565958742586877,
[
["images/helpbahasa-
sheet0.png", 92293, 0, 0, 1018,
768, 1, 0.5, 0.5,[],[-
0.4990176856517792,-
0.4986979067325592,0.5,-
0.5,0.4990180134773254,0.498697
9961395264,-
0.4990176856517792,0.4986979961
395264],0]
]
]
],
[
],
false,
false,
6486365600201218,
[],
null
]
,
[
"t44",
cr.plugins_.Sprite,
false,
[],
0,
0,
null,
[
[
"Default",
5,
false,
1,
0,

```
                false,
      8880529327193613,
                [
      ["images/helpbahasa2-
sheet0.png", 93133, 0, 0, 1018,
768, 1, 0.5, 0.5,[],[-
0.4990176856517792,-
0.4986979067325592,0.5,-
0.5,0.4990180134773254,0.498697
9961395264,-
0.4990176856517792,0.4986979961
395264],0]
                ]
                ]
           ],
           [
           ],
           false,
           false,
           6808314721041855,
           [],
           null
      ]
,     [
           "t45",
           cr.plugins_.Sprite,
           false,
           [],
           0,
           0,
           null,
           [
                [
                "Default",
                5,
                false,
                1,
                0,
                false,
      7358018047393607,
                [
      ["images/btn_stop-
sheet0.png", 22190, 0, 0, 243,
243, 1, 0.5020576119422913,
0.5020576119422913,[],[-
0.3539096117019653,-
0.3539096117019653,-
0.00411561131477356,-
0.5020576119422913,0.3539093732
833862,-
0.3580245971679688,0.4979423880
577087,-
0.00411561131477356,0.349794387
8173828,0.3497943878173828,-
0.00411561131477356,0.493827402
5917053,-
0.3539096117019653,0.3497943878
173828,-0.4979423880577087,-
0.00411561131477356],0]
                ]
                ]
           ],
           [
           ],
           false,
           false,
           6740113895118955,
           [],
           null
      ]
      ],
      [
      ],
      [
      [
           "Home",
           1280,
           768,
           false,
           "eventHome",
           7340429581381607,
           [
           [
                "Layer 0",
                0,
      8891461113084533,
                true,
                [255, 255,
255],
                false,
                1,
                1,
                1,
                false,
                1,
                0,
                0,
                [
                [
                     [640.5,
384, 0, 1279, 768, 0, 0, 1,
0.5003909468650818, 0.5, 0, 0,
[]],
                     0,
                     0,
                     [
                     ],
```

```
                        [                                                   [""]
                        ],                                          ],
                        [                                           [
                              0,                                    ],
                                                                    [
      "Default",                                                          "",
                              0,                                          "",
                              1                                           "",
                        ]                                                 1,
                  ]                                                       1,
,                 [                                                       0,
                        [104,                                             0,
673, 0, 178.7736053466797,                                                0,
178.7736053466797, 0, 0, 1,                                               0,
0.5, 0.5, 0, 0, []],                                                      ""
                        1,                                          ]
                        1,                                    ]
                        [                 ,                   [
                        ],                                          [405,
                        [                 620, 0, 488, 72, 0, 0, 1, 0, 0,
                        ],                0, 0, []],
                        [                                           38,
                              0,                                    54,
                                                                    [
      "Default",                                                    ],
                              0,                                    [
                              1                                     ],
                        ]                                           [
                  ]
,                 [                               "masukan namamu",
                        [1176,                                            0,
666, 0, 178.1909942626953,
178.1909942626953, 0, 0, 1,                     "36pt Arial Rounded MT
0.5020576119422913,                       Bold",
0.5020576119422913, 0, 0, []],
                        33,                     "rgb(255,255,255)",
                        46,                                               0,
                        [                                                 0,
                        ],                                                0,
                        [                                                 0,
                        ],                                                0
                        [                                           ]
                              0,                            ]
                                                      ,                 [
      "Default",                                                    [-618,
                              0,          394, 0, 1018, 768, 0, 0, 1,
                              1           0.5, 0.5, 0, 0, []],
                        ]                                           44,
                  ]                                                 81,
,                 [                                                 [
                        [524,                                       ],
692, 0, 248, 50, 0, 0, 1, 0, 0,                                     [
0, 0, []],                                                          ],
                        39,                                         [
                        53,                                               0,
                        [
```

```
                                                                    [255, 255,
      "Default",                        255],
                                0,                                  false,
                                1                                   1,
                          ]                                         1,
                    ]                                               1,
,                   [                                               false,
                          [950,                                     1,
172, 0, 135, 72, 0, 0, 1, 0, 0,                                     0,
0, 0, []],                                                          0,
                          38,                                       [
                          114,                                      [
                          [                                               [640.5,
                          ],                        384, 0, 1279, 768, 0, 0, 1,
                          [                         0.5003909468650818, 0.5, 0, 0,
                          ],                        []],
                          [                                               3,
                                                                          3,
      "buah",                                                             [
                                0,                                        ],
                                                                          [
      "36pt Arial Rounded MT                                              ],
Bold",                                                                    [
                                                                                0,
      "rgb(255,255,255)",
                                0,                        "Default",
                                0,                                              0,
                                0,                                              1
                                0,                                        ]
                                0                                   ]
                          ]                         ,                 [
                    ]                                                     [633,
                    ],                              493, 0, 432, 116, 0, 0, 1, 0.5,
                    [                               0.5, 0, 0, []],
      ]                                                                   5,
            ]                                                             9,
            ],                                                            [
            [                                                             ],
            ],                                                            [
            []                                                            ],
      ]                                                                   [
,     [                                                                         0,
            "GameApel",
            1280,                                 "Default",
            768,                                                                0,
            false,                                                              1
            "eventGameApel",                                              ]
            8356392465445041,                                     ]
            [                               ,                     [
            [                                                             [214,
                  "Layer 0",                  406, 0, 348, 467, 0, 0, 1, 0.5,
                  0,                          0.5010706782341003, 0, 0, []],
                                                                          4,
      3154998609455729,                                                   4,
                  true,                                                   [
                                                                          ],
```

```
                        [                                       ],
                        ],                                      [
                        [                                       [
                              0,                                      0,
                                                                      1
      "Default",                                                ]
                              0,                                ],
                              1                                 [
                        ]                                             "p",
                  ]                                                   0,
,                 [
                        [618,                 "90pt Arial Rounded MT
239, 0, 79, 145, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        6,                    "rgb(255,0,0)",
                        5,                                            0,
                        [                                             0,
                                                                      0,
      [618],                                                          0,
                                                                      0
      [239],                                                    ]
                              [0]                         ]
                        ],                  ,                 [
                        [                                       [540,
                        [                     237, 0, 79, 145, 0, 0, 1, 0, 0,
                              0,              0, 0, []],
                              1                                 6,
                        ]                                       7,
                        ],                                      [
                        [
                              "a",            [540],
                              0,
                                              [237],
      "90pt Arial Rounded MT                                          [0]
Bold",                                                          ],
                                                                [
      "rgb(255,0,0)",                                           [
                              0,                                      0,
                              0,                                      1
                              0,                                ]
                              0,                                ],
                              0                                 [
                        ]                                             "l",
                  ]                                                   0,
,                 [
                        [422,                 "90pt Arial Rounded MT
241, 0, 83, 143, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        6,                    "rgb(255,0,0)",
                        6,                                            0,
                        [                                             0,
                                                                      0,
      [422],                                                          0,
                                                                      0
      [241],                                                    ]
                              [0]                         ]
```

```
,                [
                        [733,
238, 0, 79, 145, 0, 0, 1, 0, 0,
0, 0, []],
                        6,
                        8,
                        [
      [733],

      [238],
                              [0]
                        ],
                        [
                        [
                              0,
                              1
                        ]
                        ],
                        [
                              "e",
                              0,
      "90pt Arial Rounded MT
Bold",
      "rgb(255,0,0)",
                              0,
                              0,
                              0,
                              0,
                              0
                        ]
                  ]
,                [
                        [464,
494, 0, 63, 67, 0, 0,
0.2000000029802322,
0.5079365372657776,
0.5074626803398132, 0, 0, []],
                        7,
                        10,
                        [
      ["a"]
                        ],
                        [
                        ],
                        [
                              0,
      "Default",
                              0,
                              1
                        ]
                  ]
,                [
                        [680,
498, 0, 63, 67, 0, 0,
0.2000000029802322,
0.5079365372657776,
0.5074626803398132, 0, 0, []],
                              7,
                              11,
                              [
                        ["e"]
                              ],
                              [
                              ],
                              [
                                    0,
                        "Default",
                                    2,
                                    1
                              ]
                        ]
,                [
                              [557,
500, 0, 65, 92, 0, 0,
0.2000000029802322,
0.5076923370361328, 0.5, 0, 0,
[]],
                              7,
                              12,
                              [
                        ["p"]
                              ],
                              [
                              ],
                              [
                                    0,
                        "Default",
                                    1,
                                    1
                              ]
                        ]
,                [
                              [792,
485, 0, 16, 92, 0, 0,
0.2000000029802322, 0.5, 0.5,
0, 0, []],
                              7,
                              13,
                              [
                        ["l"]
                              ],
                              [
```

```
                        ],                                      [479, 23,
                        [               0, 613, 190, 0, 0, 1, 0, 0, 0,
                              0,        0, []],
                                                                37,
      "Default",                                                52,
                              3,                                [
                              1                                 ],
                        ]                                       [
                  ]                                             ],
,                 [                                             [
                        [-581,
377, 0, 1062, 768, 0, 0, 1,                 "nama",
0.5, 0.5, 0, 0, []],                                                    0,
                        8,
                        15,                       "italic 48pt Arial Rounded
                        [               MT Bold",
                        ],
                        [                         "rgb(51,51,255)",
                        [                                               0,
                        ]                                               0,
                        ],                                              0,
                        [                                               0,
                              0,                                        0
                                                                ]
      "Default",                                          ]
                              0,        ,                 [
                              1                                 [1209,
                        ]               69, 0, 103.9230499267578,
                  ]                     103.9230499267578, 0, 0, 1,
,                 [                     0.5020580291748047,
                        [306, 24,       0.5020580291748047, 0, 0, []],
0, 165, 83, 0, 0, 1, 0, 0, 0,                                   45,
0, []],                                                         82,
                        36,                                     [
                        51,                                     ],
                        [                                       [
                        ],                                      ],
                        [                                       [
                        ],                                            0,
                        [
                                                    "Default",
      "hallo",                                                        0,
                              0,                                      1
                                                                ]
      "48pt Arial Rounded MT                              ]
Bold",                                                    ],
                                                          [
      "rgb(255,0,0)",                             ]
                              0,                        ]
                              0,                        ],
                              0,                        [
                              0,                        ],
                              0                         []
                        ]                           ]
                  ]                           ,     [
,                 [                                   "GameNanas",
```

```
            1280,
            768,
            false,
            "eventGameNanas",
            2546695225403348,
            [
            [
                  "Layer 0",
                  0,
     158318900876142,
                  true,
                  [255, 255,
255],
                  false,
                  1,
                  1,
                  1,
                  false,
                  1,
                  0,
                  0,
                  [
                  [
                        [640.5,
384, 0, 1279, 768, 0, 0, 1,
0.5003909468650818, 0.5, 0, 0,
[]],
                        3,
                        16,
                        [
                        ],
                        [
                        ],
                        [
                              0,
     "Default",
                              0,
                              1
                        ]
                  ]
,                 [
                        [672,
576, 0, 531.130859375,
106.2261734008789, 0, 0, 1,
0.5, 0.5, 0, 0, []],
                        12,
                        24,
                        [
                        ],
                        [
                        ],
                        [
                              0,
                  "Default",
                                                      0,
                                                      1
                                                ]
                                          ]
,                                         [
                                                [197,
432, 0, 348, 467, 0, 0, 1, 0.5,
0.5010706782341003, 0, 0, []],
                                                9,
                                                17,
                                                [
                                                ],
                                                [
                                                ],
                                                [
                                                      0,
                  "Default",
                                                      0,
                                                      1
                                                ]
                                          ]
,                                         [
                                                [608,
224, 0, 89, 143, 0, 0, 1, 0, 0,
0, 0, []],
                                                10,
                                                18,
                                                [
                  [608],
                  [224],
                                                      [0]
                                                ],
                                                [
                                                [
                                                      0,
                                                      1
                                                ]
                                                ],
                                                [
                                                      "n",
                                                      0,
                  "90pt Arial Rounded MT
Bold",
                  "rgb(153,0,255)",
                                                      0,
                                                      0,
                                                      0,
                                                      0,
                                                      0
```

```
                        ]                                                   "n",
                  ]                                                         0,
,                 [
                        [608,                 "90pt Arial Rounded MT
352, 0, 89, 143, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        10,                   "rgb(153,0,255)",
                        19,                                                 0,
                        [                                                   0,
                                                                            0,
      [608],                                                                0,
                                                                            0
      [352],                                                        ]
                              [0]                           ]
                        ],                ,                   [
                        [                                             [800,
                        [                 224, 0, 89, 143, 0, 0, 1, 0, 0,
                              0,          0, 0, []],
                              1                                     10,
                        ]                                           21,
                        ],                                          [
                        [
                              "a",        [800],
                              0,
                                          [224],
      "90pt Arial Rounded MT                                                [0]
Bold",                                                              ],
                                                                    [
      "rgb(153,0,255)",                                             [
                              0,                                            0,
                              0,                                            1
                              0,                                    ]
                              0,                                    ],
                              0                                     [
                        ]                                                   "a",
                  ]                                                         0,
,                 [
                        [704,                 "90pt Arial Rounded MT
352, 0, 89, 143, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        10,                   "rgb(153,0,255)",
                        20,                                                 0,
                        [                                                   0,
                                                                            0,
      [704],                                                                0,
                                                                            0
      [352],                                                        ]
                              [0]                           ]
                        ],                ,                   [
                        [                                             [416,
                        [                 352, 0, 89, 143, 0, 0, 1, 0, 0,
                              0,          0, 0, []],
                              1                                     10,
                        ]                                           22,
                        ],                                          [
                        [
```

```
                                                                [
      [416],
                                                ["a"]
      [352],                                                    ],
                            [0]                                 [
                        ],                                      ],
                        [                                       [
                        [                                           0,
                              0,
                              1                 "Default",
                        ]                                           1,
                        ],                                          1
                        [                                       ]
                              "s",                        ]
                              0,        ,                 [
                                                                [672,
      "90pt Arial Rounded MT          576, 0, 61, 70, 0, 0,
Bold",                                0.2000000029802322,
                                      0.5081967115402222, 0.5, 0, 0,
      "rgb(153,0,255)",               []],
                              0,                                11,
                              0,                                27,
                              0,                                [
                              0,
                              0                 ["n"]
                        ]                                       ],
                  ]                                             [
,                 [                                             ],
                        [461,                                   [
576, 0, 61, 70, 0, 0,                                               0,
0.2000000029802322,
0.5081967115402222, 0.5, 0, 0,                  "Default",
[]],                                                                2,
                        11,                                         1
                        23,                                     ]
                        [                               ]
                                          ,             [
      ["n"]                                                     [768,
                        ],                      576, 0, 66, 70, 0, 0,
                        [                       0.2000000029802322, 0.5, 0.5,
                        ],                      0, 0, []],
                        [                                       11,
                              0,                                28,
                                                                [
      "Default",
                              0,                ["a"]
                              1                                 ],
                        ]                                       [
                  ]                                             ],
,                 [                                             [
                        [561,                                       0,
576, 0, 66, 70, 0, 0,
0.2000000029802322, 0.5, 0.5,                   "Default",
0, 0, []],                                                          3,
                        11,                                         1
                        26,                                     ]
```

```
                ]                                       ]
,               [                       ,               [
                    [864,                                   [704,
576, 0, 59, 70, 0, 0,                   224, 0, 89, 143, 0, 0, 1, 0, 0,
0.2000000029802322,                     0, 0, []],
0.508474588394165, 0.5, 0, 0,                               10,
[]],                                                        179,
                    11,                                     [
                    29,
                    [                       [704],

    ["s"]                               [224],
                    ],                                          [0]
                    [                                       ],
                    ],                                      [
                    [                                       [
                        0,                                      0,
                                                                1
    "Default",                                              ]
                        4,                                  ],
                        1                                   [
                    ]                                           "g",
                ]                                               0,
,               [
                    [512,               "90pt Arial Rounded MT
352, 0, 89, 143, 0, 0, 1, 0, 0,     Bold",
0, 0, []],
                    10,                 "rgb(153,0,255)",
                    178,                                        0,
                    [                                           0,
                                                                0,
    [512],                                                      0,
                                                                0
    [352],                                                  ]
                        [0]                         ]
                    ],              ,               [
                    [                                       [512,
                    [               224, 0, 89, 143, 0, 0, 1, 0, 0,
                        0,          0, 0, []],
                        1                               10,
                    ]                                   180,
                    ],                                  [
                    [
                        "c",            [512],
                        0,
                                        [224],
    "90pt Arial Rounded MT                                      [0]
Bold",                                                      ],
                                                            [
    "rgb(153,0,255)",                                       [
                        0,                                      0,
                        0,                                      1
                        0,                              ]
                        0,                              ],
                        0                               [
                    ]                                           "h",
```

```
                        0,
                                                [352],
     "90pt Arial Rounded MT                                          [0]
Bold",                                                        ],
                                                              [
     "rgb(153,0,255)",                                        [
                        0,                                          0,
                        0,                                          1
                        0,                                    ]
                        0,                                    ],
                        0                                     [
                    ]                                               "o",
                ]                                                   0,
,               [
                    [416,               "90pt Arial Rounded MT
224, 0, 89, 143, 0, 0, 1, 0, 0,      Bold",
0, 0, []],
                    10,                 "rgb(153,0,255)",
                    181,                                            0,
                    [                                               0,
                                                                    0,
     [416],                                                         0,
                                                                    0
     [224],                                                   ]
                        [0]                           ]
                    ],              ,                 [
                    [                                        [-543,
                    [               373, 0, 1062, 768, 0, 0, 1,
                        0,          0.5, 0.5, 0, 0, []],
                        1                                    13,
                    ]                                        25,
                    ],                                       [
                    [                                        ],
                        "u",                                 [
                        0,                                   [
                                                             ]
     "90pt Arial Rounded MT                                  ],
Bold",                                                       [
                                                                    0,
     "rgb(153,0,255)",
                        0,              "Default",
                        0,                                          0,
                        0,                                          1
                        0,                                   ]
                        0                             ]
                    ]               ,                 [
                ]                                            [301, 35,
,               [                   0, 165, 83, 0, 0, 1, 0, 0, 0,
                    [800,           0, []],
352, 0, 89, 143, 0, 0, 1, 0, 0,                              36,
0, 0, []],                                                   102,
                    10,                                      [
                    182,                                     ],
                    [                                        [
                                                             ],
     [800],                                                  [
```

"hallo",
0,
"48pt Arial Rounded MT Bold",
"rgb(255,0,0)",
0,
0,
0,
0,
0
]
]
,
[
[474, 34, 0, 613, 190, 0, 0, 1, 0, 0, 0, 0, []],
37,
103,
[
],
[
],
[
"nama",
0,
"italic 48pt Arial Rounded MT Bold",
"rgb(51,51,255)",
0,
0,
0,
0,
0
]
]
,
[
[1204, 80, 0, 103.9230499267578, 103.9230499267578, 0, 0, 1, 0.5020580291748047, 0.5020580291748047, 0, 0, []],
45,
104,
[
],
[
],
[
0,
"Default",
0,
1
]
]
],
[
]
]
],
[
]
]
,
[
"GameMangga",
1280,
768,
false,
"eventGameMangga",
7141668855806651,
[
[
"Layer 0",
0,
2643346869779356,
true,
[255, 255, 255],
false,
1,
1,
1,
false,
1,
0,
0,
[
[
[640.5, 384, 0, 1279, 768, 0, 0, 1, 0.5003909468650818, 0.5, 0, 0, []],
3,
30,
[
],
[
],
[
0,
"Default",
0,

```
                        1
                    ]                   "Default",
                ]                                           0,
,               [                                           1
                    [188,                               ]
420, 0, 348, 467, 0, 0, 1, 0.5,                     ]
0.5010706782341003, 0, 0, []],      ,               [
                    14,                                 [576,
                    31,             579, 0, 63, 67, 0, 0,
                    [               0.2000000029802322,
                    ],              0.5079365372657776,
                    [               0.5074626803398132, 0, 0, []],
                    ],                                  17,
                    [                                   39,
                        0,                              [

    "Default",                          ["a"]
                        0,                              ],
                        1                               [
                    ]                                   ],
                ]                                       [
,               [                                           0,
                    [752,
581, 0, 732, 114, 0,                    "Default",
3.141592741012573, 1, 0.5, 0.5,                             1,
0, 0, []],                                                  1
                    16,                                 ]
                    37,                             ]
                    [               ,               [
                    ],                                  [696,
                    [               580, 0, 59, 67, 0, 0,
                    ],              0.2000000029802322,
                    [               0.508474588394165,
                        0,          0.5074626803398132, 0, 0, []],
                                                        17,
    "Default",                                          40,
                        0,                              [
                        1
                    ]                   ["n"]
                ]                                       ],
,               [                                       [
                    [448,                               ],
575, 0, 95, 67, 0, 0,                                   [
0.2000000029802322,                                         0,
0.5052631497383118,
0.5074626803398132, 0, 0, []],          "Default",
                    17,                                     2,
                    38,                                     1
                    [                                   ]
                                                    ]
    ["m"]                           ,               [
                    ],                                  [810,
                    [               582, 0, 65, 92, 0, 0,
                    ],              0.2000000029802322,
                    [               0.5076923370361328, 0.5, 0, 0,
                        0,          []],
```

```
                        17,                                         5,
                        41,                                         1
                        [                                       ]
                                                            ]
      ["g"]                             ,                   [
                        ],                                      [512,
                        [               224, 0, 112, 149, 0, 0, 1, 0,
                        ],              0, 0, 0, []],
                        [                                       15,
                              0,                                14,
                                                                [
      "Default",
                              3,              [512],
                              1
                        ]                     [224],
                  ]                                                     [0]
,                 [                                             ],
                        [969,                                   [
581, 0, 65, 92, 0, 0,                                           [
0.2000000029802322,                                                   0,
0.5076923370361328, 0.5, 0, 0,                                        1
[]],                                                            ]
                        17,                                     ],
                        42,                                     [
                        [                                             "m",
                                                                      0,
      ["g"]
                        ],                    "90pt Arial Rounded MT
                        [               Bold",
                        ],
                        [                     "rgb(0,153,0)",
                              0,                                      0,
                                                                      0,
      "Default",                                                      0,
                              4,                                      0,
                              1                                       0
                        ]                                       ]
                  ]                                           ]
,                 [                     ,                     [
                        [1063,                                  [896,
576, 0, 63, 67, 0, 0,                   352, 0, 80, 149, 0, 0, 1, 0, 0,
0.2000000029802322,                     0, 0, []],
0.5079365372657776,                                             15,
0.5074626803398132, 0, 0, []],                                  32,
                        17,                                     [
                        43,
                        [                     [896],

      ["a"]                                   [352],
                        ],                                            [0]
                        [                                       ],
                        ],                                      [
                        [                                       [
                              0,                                      0,
                                                                      1
      "Default",                                              ]
```

```
                        ],                                      [
                        [
                              "n",            [704],
                              0,
                                              [352],
      "90pt Arial Rounded MT                                          [0]
Bold",                                                        ],
                                                              [
      "rgb(0,153,0)",                                         [
                              0,                                      0,
                              0,                                      1
                              0,                              ]
                              0,                              ],
                              0                               [
                        ]                                             "g",
                  ]                                                   0,
,                 [
                        [640,                 "90pt Arial Rounded MT
224, 0, 76, 149, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        15,                   "rgb(0,153,0)",
                        33,                                           0,
                        [                                             0,
                                                                      0,
      [640],                                                          0,
                                                                      0
      [224],                                                  ]
                              [0]                         ]
                        ],                ,                 [
                        [                                       [832,
                        [                 224, 0, 84, 149, 0, 0, 1, 0, 0,
                              0,          0, 0, []],
                              1                               15,
                        ]                                     35,
                        ],                                    [
                        [
                              "g",            [832],
                              0,
                                              [224],
      "90pt Arial Rounded MT                                          [0]
Bold",                                                        ],
                                                              [
      "rgb(0,153,0)",                                         [
                              0,                                      0,
                              0,                                      1
                              0,                              ]
                              0,                              ],
                              0                               [
                        ]                                             "a",
                  ]                                                   0,
,                 [
                        [704,                 "90pt Arial Rounded MT
352, 0, 75, 149, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        15,                   "rgb(0,153,0)",
                        34,                                           0,
```

```
                              0,                                      1
                              0,                                ]
                              0,                                ],
                              0                                 [
                        ]                                             "k",
                  ]                                                   0,
,                 [
                        [512,                 "90pt Arial Rounded MT
352, 0, 80, 149, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        15,                   "rgb(0,153,0)",
                        36,                                           0,
                        [                                             0,
                                                                      0,
      [512],                                                          0,
                                                                      0
      [352],                                                    ]
                              [0]                         ]
                        ],                ,                 [
                        [                                         [992,
                        [                 352, 0, 78, 141, 0, 0, 1, 0, 0,
                              0,          0, 0, []],
                              1                                 15,
                        ]                                       184,
                        ],                                      [
                        [
                              "a",                [992],
                              0,
                                                  [352],
      "90pt Arial Rounded MT                                          [0]
Bold",                                                            ],
                                                                  [
      "rgb(0,153,0)",                                             [
                              0,                                      0,
                              0,                                      1
                              0,                                ]
                              0,                                ],
                              0                                 [
                        ]                                             "i",
                  ]                                                   0,
,                 [
                        [416,                 "90pt Arial Rounded MT
224, 0, 78, 141, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        15,                   "rgb(0,153,0)",
                        183,                                          0,
                        [                                             0,
                                                                      0,
      [416],                                                          0,
                                                                      0
      [224],                                                    ]
                              [0]                         ]
                        ],                ,                 [
                        [                                         [608,
                        [                 352, 0, 78, 141, 0, 0, 1, 0, 0,
                              0,          0, 0, []],
```

15,
185, "rgb(0,153,0)",
[0,
0,
[608], 0,
0,
[352], 0
[0]]
],]
[, [
[[736,
0, 224, 0, 82, 142, 0, 0, 1, 0, 0,
1 0, 0, []],
] 15,
], 187,
[[
"z",
0, [736],

"90pt Arial Rounded MT [224],
Bold", [0]
], [
"rgb(0,153,0)", [
0, [
0, 0,
0, 1
0,]
0], [
] "b",
] 0,
, [
[416, "90pt Arial Rounded MT
352, 0, 78, 141, 0, 0, 1, 0, 0, Bold",
0, 0, []],
15,
186, "rgb(0,153,0)",
[0,
0,
[416], 0,
0,
[352], 0
[0]]
],]
[, [
[[800,
0, 352, 0, 78, 141, 0, 0, 1, 0, 0,
1 0, 0, []],
] 15,
], 188,
[[
"u",
0, [800],

"90pt Arial Rounded MT [352],
Bold", [0]
],]

```
                        [                                       [-572,
                        [               383, 0, 1062, 768, 0, 0, 1,
                              0,        0.5, 0.5, 0, 0, []],
                              1                                 19,
                        ]                                       45,
                        ],                                      [
                        [                                       ],
                              "x",                              [
                              0,                                [
                                                                ]
      "90pt Arial Rounded MT                                    ],
Bold",                                                          [
                                                                      0,
      "rgb(0,153,0)",
                              0,              "Default",
                              0,                                      0,
                              0,                                      1
                              0,                                ]
                              0                           ]
                        ]               ,                   [
                  ]                                             [295, 27,
,                 [                     0, 165, 83, 0, 0, 1, 0, 0, 0,
                        [928,           0, []],
224, 0, 85, 155, 0, 0, 1, 0, 0,                                36,
0, 0, []],                                                      105,
                        15,                                     [
                        189,                                    ],
                        [                                       [
                                                                ],
      [928],                                                    [

      [224],                                  "hallo",
                              [0]                                     0,
                        ],
                        [                     "48pt Arial Rounded MT
                        [               Bold",
                              0,
                              1               "rgb(255,0,0)",
                        ]                                             0,
                        ],                                            0,
                        [                                             0,
                              "p",                                    0,
                              0,                                      0
                                                                ]
      "90pt Arial Rounded MT                            ]
Bold",                                  ,                   [
                                                                [468, 26,
      "rgb(0,153,0)",                   0, 613, 190, 0, 0, 1, 0, 0, 0,
                              0,        0, []],
                              0,                                37,
                              0,                                106,
                              0,                                [
                              0                                 ],
                        ]                                       [
                  ]                                             ],
,                 [                                             [
```

"nama",
0,
"italic 48pt Arial Rounded MT Bold",
"rgb(51,51,255)",
0,
0,
0,
0,
0
]
]
,
[
[1198, 72, 0, 103.9230499267578, 103.9230499267578, 0, 0, 1, 0.5020580291748047, 0.5020580291748047, 0, 0, []],
45,
107,
[
],
[
],
[
0,
"Default",
0,
1
]
]
],
[
]
]
],
[
],
[]
]
,
[
"GameSemangka",
1280,
768,
false,
"eventSemangka",
8759755991037678,
[
[
"Layer 0",
0,
1689627031191662,
true,
[255, 255, 255],
false,
1,
1,
1,
false,
1,
0,
0,
[
[
[640.5, 384, 0, 1279, 768, 0, 0, 1, 0.5003909468650818, 0.5, 0, 0, []],
3,
60,
[
],
[
],
[
0,
"Default",
0,
1
]
]
,
[
[183, 416, 0, 348, 467, 0, 0, 1, 0.5, 0.5010706782341003, 0, 0, []],
25,
61,
[
],
[
],
[
0,
"Default",
0,
1
]
]
,
[
[821, 661, 0, 886.8099975585938, 110.8512496948242, 0, 0, 1, 0.5, 0.5, 0, 0, []],

26,
62,
[
],
[
],
[
0,
0,
1
]
]
,
[
[702,
181, 0, 77, 141, 0, 0, 1, 0, 0,
0, 0, []],
27,
63,
[
[704],
[32],
[0]
],
[
[
0,
1
]
],
[
"s",
0,
"90pt Arial Rounded MT
Bold",
"rgb(0,153,0)",
0,
0,
0,
0,
0
]
]
,
[
[478,
437, 0, 77, 141, 0, 0, 1, 0, 0,
0, 0, []],
27,
64,
[
[480],
[288],
[0]
],
[
[
0,
1
]
],
[
"e",
0,
"90pt Arial Rounded MT
Bold",
"rgb(0,153,0)",
0,
0,
0,
0,
0
]
]
,
[
[862,
437, 0, 82, 144, 0, 0, 1, 0, 0,
0, 0, []],
27,
65,
[
[864],
[288],
[0]
],
[
[
0,
1
]
],
[
"n",
0,
"90pt Arial Rounded MT
Bold",
"rgb(0,153,0)",
0,
0,

```
                              0,                                  ]
                              0,                                  ],
                              0                                   [
                        ]                                               "k",
                  ]                                                     0,
,                 [
                        [382,                 "90pt Arial Rounded MT
181, 0, 110, 140, 0, 0, 1, 0,         Bold",
0, 0, 0, []],
                        27,                   "rgb(0,153,0)",
                        66,                                             0,
                        [                                               0,
                                                                        0,
      [384],                                                            0,
                                                                        0
      [32],                                                       ]
                              [0]                         ]
                        ],                ,                 [
                        [                                           [382,
                        [                 437, 0, 80, 143, 0, 0, 1, 0, 0,
                              0,          0, 0, []],
                              1                                   27,
                        ]                                         68,
                        ],                                        [
                        [
                              "m",                [384],
                              0,
                                                  [288],
      "90pt Arial Rounded MT                                            [0]
Bold",                                                            ],
                                                                  [
      "rgb(0,153,0)",                                             [
                              0,                                        0,
                              0,                                        1
                              0,                                  ]
                              0,                                  ],
                              0                                   [
                        ]                                               "a",
                  ]                                                     0,
,                 [
                        [478,                 "90pt Arial Rounded MT
309, 0, 77, 140, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        27,                   "rgb(0,153,0)",
                        67,                                             0,
                        [                                               0,
                                                                        0,
      [480],                                                            0,
                                                                        0
      [160],                                                      ]
                              [0]                         ]
                        ],                ,                 [
                        [                                           [638,
                        [                 309, 0, 75, 140, 0, 0, 1, 0, 0,
                              0,          0, 0, []],
                              1                                   27,
```

```
                    69,                                         0,
                    [                                           0,
                                                                0,
    [640],                                                      0,
                                                                0
    [160],                                                  ]
                         [0]                            ]
                    ],              ,                   [
                    [                                       [434,
                    [               659, 0, 57, 67, 0, 0, 0,
                         0,         0.5087719559669495,
                         1          0.5074626803398132, 0, 0, []],
                    ]                                       28,
                    ],                                      71,
                    [                                       [
                         "g",
                         0,             ["s"]
                                                            ],
    "90pt Arial Rounded MT                                  [
Bold",                                                      ],
                                                            [
    "rgb(0,153,0)",                                             0,
                         0,
                         0,             "Default",
                         0,                                     0,
                         0,                                     1
                         0                                  ]
                    ]                                   ]
                ]                   ,                   [
,               [                                       [530,
                    [670,           659, 0, 63, 67, 0, 0, 0,
437, 0, 77, 140, 0, 0, 1, 0, 0,     0.5079365372657776,
0, 0, []],                          0.5074626803398132, 0, 0, []],
                    27,                                     28,
                    70,                                     72,
                    [                                       [

    [672],                              ["e"]
                                                            ],
    [288],                                                  [
                         [0]                                ],
                    ],                                      [
                    [                                           0,
                    [
                         0,             "Default",
                         1                                      1,
                    ]                                           1
                    ],                                      ]
                    [                                   ]
                         "a",       ,                   [
                         0,                                 [638,
                                    659, 0, 95, 67, 0, 0, 0,
    "90pt Arial Rounded MT          0.5052631497383118,
Bold",                              0.5074626803398132, 0, 0, []],
                                                            28,
    "rgb(0,153,0)",                                         73,
```

```
                    [                   ,                   [
                                                            [962,
      ["m"]                             663, 0, 65, 92, 0, 0, 0,
                    ],                  0.5076923370361328, 0.5, 0, 0,
                    [                   []],
                    ],                                      28,
                    [                                       76,
                          0,                                [

      "Default",                              ["g"]
                          2,                                ],
                          1                                 [
                    ]                                       ],
                ]                                           [
,               [                                                 0,
                    [745,
659, 0, 63, 67, 0, 0, 0,                      "Default",
0.5079365372657776,                                                 5,
0.5074626803398132, 0, 0, []],                                      1
                    28,                                     ]
                    74,                               ]
                    [                   ,                   [
                                                            [1070,
      ["a"]                             659, 0, 57, 92, 0, 0, 0,
                    ],                  0.5087719559669495, 0.5, 0, 0,
                    [                   []],
                    ],                                      28,
                    [                                       77,
                          0,                                [

      "Default",                              ["k"]
                          3,                                ],
                          1                                 [
                    ]                                       ],
                ]                                           [
,               [                                                 0,
                    [845,
661, 0, 59, 67, 0, 0, 0,                      "Default",
0.508474588394165,                                                  6,
0.5074626803398132, 0, 0, []],                                      1
                    28,                                     ]
                    75,                               ]
                    [                   ,                   [
                                                            [1184,
      ["n"]                             657, 0, 63, 67, 0, 0, 0,
                    ],                  0.5079365372657776,
                    [                   0.5074626803398132, 0, 0, []],
                    ],                                      28,
                    [                                       78,
                          0,                                [

      "Default",                              ["a"]
                          4,                                ],
                          1                                 [
                    ]                                       ],
                ]                                           [
```

```
                        0,                                      0,
                                                                1
      "Default",                                        ]
                        7,                              ],
                        1                               [
                    ]                                       "o",
                ]                                           0,
,               [
                    [382,               "90pt Arial Rounded MT
309, 0, 75, 140, 0, 0, 1, 0, 0,     Bold",
0, 0, []],
                    27,                 "rgb(0,153,0)",
                    197,                                    0,
                    [                                       0,
                                                            0,
      [384],                                                0,
                                                            0
      [160],                                            ]
                        [0]                     ]
                    ],              ,               [
                    [                                   [574,
                    [               437, 0, 82, 140, 0, 0, 1, 0, 0,
                        0,          0, 0, []],
                        1                               27,
                    ]                                   199,
                    ],                                  [
                    [
                        "d",        [576],
                        0,
                                    [288],
      "90pt Arial Rounded MT                                [0]
Bold",                                                  ],
                                                        [
      "rgb(0,153,0)",                                   [
                        0,                                  0,
                        0,                                  1
                        0,                              ]
                        0,                              ],
                        0                               [
                    ]                                       "h",
                ]                                           0,
,               [
                    [766,               "90pt Arial Rounded MT
437, 0, 82, 140, 0, 0, 1, 0, 0,     Bold",
0, 0, []],
                    27,                 "rgb(0,153,0)",
                    198,                                    0,
                    [                                       0,
                                                            0,
      [768],                                                0,
                                                            0
      [288],                                            ]
                        [0]                     ]
                    ],              ,               [
                    [
                    [
```

```
                        [574,
309, 0, 82, 140, 0, 0, 1, 0, 0,
0, 0, []],
                        27,
                        200,
                        [
      [576],
      [160],
                              [0]
                        ],
                        [
                        [
                              0,
                              1
                        ]
                        ],
                        [
                              "i",
                              0,
      "90pt Arial Rounded MT
Bold",
      "rgb(0,153,0)",
                              0,
                              0,
                              0,
                              0,
                              0
                        ]
                  ]
,                 [
                        [606,
181, 0, 82, 140, 0, 0, 1, 0, 0,
0, 0, []],
                        27,
                        201,
                        [
      [608],
      [32],
                              [0]
                        ],
                        [
                        [
                              0,
                              1
                        ]
                        ],
                        [
                              "z",
                              0,
      "90pt Arial Rounded MT
Bold",
      "rgb(0,153,0)",
                              0,
                              0,
                              0,
                              0,
                              0
                        ]
                  ]
,                 [
                        [510,
181, 0, 82, 140, 0, 0, 1, 0, 0,
0, 0, []],
                        27,
                        202,
                        [
      [512],
      [32],
                              [0]
                        ],
                        [
                        [
                              0,
                              1
                        ]
                        ],
                        [
                              "x",
                              0,
      "90pt Arial Rounded MT
Bold",
      "rgb(0,153,0)",
                              0,
                              0,
                              0,
                              0,
                              0
                        ]
                  ]
,                 [
                        [734,
309, 0, 112, 139, 0, 0, 1, 0,
0, 0, 0, []],
                        27,
                        203,
                        [
      [736],
```

[160],
[0]
],
[
[
0,
1
]
],
[
"w",
0,
"90pt Arial Rounded MT Bold",
"rgb(0,153,0)",
0,
0,
0,
0,
0
]
]
,
[
[-562, 376, 0, 1062, 768, 0, 0, 1, 0.5, 0.5, 0, 0, []],
29,
79,
[
],
[
[
]
],
[
0,
"Default",
0,
1
]
]
,
[
[297, 26, 0, 165, 83, 0, 0, 1, 0, 0, 0, 0, []],
36,
108,
[
],
[
],
[
"hallo",
0,
"48pt Arial Rounded MT Bold",
"rgb(255,0,0)",
0,
0,
0,
0,
0
]
]
,
[
[470, 25, 0, 613, 190, 0, 0, 1, 0, 0, 0, 0, []],
37,
109,
[
],
[
],
[
"nama",
0,
"italic 48pt Arial Rounded MT Bold",
"rgb(51,51,255)",
0,
0,
0,
0,
0
]
]
,
[
[1200, 71, 0, 103.9230499267578, 103.9230499267578, 0, 0, 1, 0.5020580291748047, 0.5020580291748047, 0, 0, []],
45,
110,
[
],
[
],
[
0,

```
                                                    ]
      "Default",                        ,           [
                        0,                                [1210,
                        1         58, 0, 97.34474945068359,
                  ]               97.34474945068359, 0, 0, 1,
            ]                     0.5, 0.5, 0, 0, []],
            ],                                          35,
            [                                           50,
      ]                                                 [
            ]                                           ],
            ],                                          [
            [                                           ],
            ],                                          [
            []                                                0,
      ]
,     [                                 "Default",
            "Help",                                           0,
            1280,                                             1
            768,                                        ]
            false,                              ]
            "eventHome",                        ],
            5025879684798269,                   [
            [                           ]
            [                                 ]
                  "Layer 0",                  ],
                  0,                          [
                  11889135600134,             ],
                  true,                       []
                  [255, 255,          ]
255],                               ,     [
                  false,                      "hlmAkhir",
                  1,                          1280,
                  1,                          768,
                  1,                          false,
                  false,                      "eventHalAkhir",
                  1,                          7607558849499734,
                  0,                          [
                  0,                          [
                  [                                   "Layer 0",
                  [                                   0,
                        [640,
384, 0, 1276, 768, 0, 0, 1,           2777078667877836,
0.5, 0.5, 0, 0, []],                                  true,
                        34,                           [255, 255,
                        49,               255],
                        [                             false,
                        ],                            1,
                        [                             1,
                        ],                            1,
                        [                             false,
                              0,                      1,
                                                      0,
      "Default",                                      0,
                              0,                      [
                              1                       [
                        ]
```

```
                        [640.5,                                         0,
384, 0, 1279, 768, 0, 0, 1,                                             0,
0.5003909468650818, 0.5, 0, 0,                                          0,
[]],                                                                    0,
                        0,                                              0
                        48,                                     ]
                        [                               ]
                        ],              ,               [
                        [                                       [332,
                        ],              403, 0, 192, 96, 0, 0, 1, 0, 0,
                        [               0, 0, []],
                              0,                                37,
                                                                57,
      "Default",                                                [
                              0,                                ],
                              1                                 [
                        ]                                       ],
                  ]                                             [
,                 [
                        [649,                 "nama",
570, 0, 1053, 368, 0, 0, 1,                                             0,
0.5004749894142151, 0.5, 0, 0,
[]],                                                  "48pt Arial Rounded MT
                        40,             Bold",
                        55,
                        [                     "rgb(0,0,204)",
                        ],                                              0,
                        [                                               0,
                        ],                                              0,
                        [                                               0,
                              0,                                        0
                                                                ]
      "Default",                                        ]
                              0,        ,               [
                              1                                 [147,
                        ]               579, 0, 473, 165, 0, 0, 1, 0,
                  ]                     0, 0, 0, []],
,                 [                                               41,
                        [160,                                           58,
406, 0, 160, 96, 0, 0, 1, 0, 0,                                         [
0, 0, []],                                                              ],
                        36,                                             [
                        56,                                             ],
                        [                                               [
                        ],
                        [                     "100",
                        ],                                              0,
                        [
                                              "90pt Arial Rounded MT
      "hallo",                        Bold",
                              0,
                                              "rgb(0,0,0)",
      "48pt Arial Rounded MT                                            1,
Bold",                                                                  0,
                                                                        0,
      "rgb(255,0,0)",                                                   0,
```

```
                        0
                    ]
                ]
                [
                    [1108,
453, 0, 97.03607940673828,
97.03607940673828, 0, 0, 1,
0.5, 0.5, 0, 0, []],
                    35,
                    59,
                    [
                    ],
                    [
                    ],
                    [
                        0,
    "Default",
                        0,
                        1
                    ]
                ]
                [
                    [159,
474, 0, 592, 118, 0, 0, 1, 0,
0, 0, 0, []],
                    42,
                    80,
                    [
                    ],
                    [
                    ],
                    [
    "nilai kamu ",
                        0,
    "72pt Arial Rounded MT
Bold",
    "rgb(255,153,0)",
                        0,
                        0,
                        0,
                        0,
                        0
                    ]
                ]
                ],
                [
    ]
        ]
        ],
        [
        ],
        []
        ]
    ,   [
            "GameJeruk",
            1280,
            768,
            false,
            "eventGameJeruk",
            5713264964167947,
            [
            [
                "Layer 0",
                0,
        9875604913653917,
                true,
                [255, 255,
255],
                false,
                1,
                1,
                1,
                false,
                1,
                0,
                0,
                [
                [
                    [642,
387, 0, 1279, 768, 0, 0, 1,
0.5003909468650818, 0.5, 0, 0,
[]],
                    3,
                    47,
                    [
                    ],
                    [
                    ],
                    [
                        0,
    "Default",
                        0,
                        1
                    ]
                ]
                [
                    [197,
430, 0, 348, 467, 0, 0, 1, 0.5,
0.5010706782341003, 0, 0, []],
                    20,
                    83,
                    [
                    ],
                    [
                    ],
                    [
```

```
                        0,                                      0,
                                                                0
     "Default",                                         ]
                        0,                      ]
                        1         ,             [
                    ]                                   [609,
                ]                   163, 0, 68, 142, 0, 0, 1, 0, 0,
,               [                   0, 0, []],
                    [717,                               22,
536, 0, 599.0366821289063,                              86,
119.8073425292969, 0, 0, 1,                             [
0.5, 0.5, 0, 0, []],
                    21,                 [608],
                    84,
                    [                   [160],
                    ],                                      [0]
                    [                                   ],
                    ],                                  [
                    [                                   [
                        0,                                      0,
                                                                1
     "Default",                                         ]
                        0,                              ],
                        1                               [
                    ]                                       "r",
                ]                                           0,
,               [
                    [609,               "90pt Arial Rounded MT
291, 0, 54, 154, 0, 0, 1, 0, 0,     Bold",
0, 0, []],
                    22,                 "rgb(255,255,87)",
                    85,                                     0,
                    [                                       0,
                                                            0,
     [608],                                                 0,
                                                            0
     [228],                                             ]
                        [0]                     ]
                    ],              ,           [
                    [                                   [417,
                    [               163, 0, 82, 154, 0, 0, 1, 0, 0,
                        0,          0, 0, []],
                        1                               22,
                    ]                                   87,
                    ],                                  [
                    [
                        "j",            [416],
                        0,
                                        [160],
     "90pt Arial Rounded MT                                 [0]
Bold",                                                  ],
                                                        [
     "rgb(255,255,87)",                                 [
                        0,                                      0,
                        0,                                      1
                        0,                              ]
```

```
                        ],                                      [
                        [
                              "k",            [704],
                              0,
                                              [160],
      "90pt Arial Rounded MT                                          [0]
Bold",                                                        ],
                                                              [
      "rgb(255,255,87)",                                      [
                              0,                                      0,
                              0,                                      1
                              0,                              ]
                              0,                              ],
                              0                               [
                        ]                                             "e",
                  ]                                                   0,
,                 [
                        [897,                 "90pt Arial Rounded MT
163, 0, 84, 142, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        22,                   "rgb(255,255,87)",
                        88,                                           0,
                        [                                             0,
                                                                      0,
      [896],                                                          0,
                                                                      0
      [160],                                                  ]
                              [0]                       ]
                        ],                ,               [
                        [                                       [471,
                        [               538, 0, 34, 116, 0, 0, 0, 0.5,
                              0,        0.5, 0, 0, []],
                              1                               23,
                        ]                                     90,
                        ],                                    [
                        [
                              "u",            ["j"]
                              0,                              ],
                                                              [
      "90pt Arial Rounded MT                                  ],
Bold",                                                        [
                                                                      0,
      "rgb(255,255,87)",
                              0,              "Default",
                              0,                                      0,
                              0,                                      1
                              0,                              ]
                              0                         ]
                        ]                 ,               [
                  ]                                           [597,
,                 [                     536, 0, 63, 67, 0, 0, 0,
                        [705,             0.5079365372657776,
163, 0, 78, 144, 0, 0, 1, 0, 0,       0.5074626803398132, 0, 0, []],
0, 0, []],                                                    23,
                        22,                                   91,
                        89,                                   [
```

```
                                                                    [945,
      ["e"]                          532, 0, 57, 92, 0, 0, 0,
                        ],           0.5087719559669495, 0.5, 0, 0,
                        [            []],
                        ],                                          23,
                        [                                           94,
                              0,                                    [

      "Default",                         ["k"]
                              1,                                    ],
                              1                                     [
                        ]                                           ],
                  ]                                                 [
,                 [                                                       0,
                        [713,
538, 0, 45, 67, 0, 0, 0,                 "Default",
0.5111111402511597,                                                     4,
0.5074626803398132, 0, 0, []],                                          1
                        23,                                         ]
                        92,                                   ]
                        [            ,                        [
                                                                    [673,
      ["r"]                          291, 0, 78, 144, 0, 0, 1, 0, 0,
                        ],           0, 0, []],
                        [                                           22,
                        ],                                          95,
                        [                                           [
                              0,
                                           [672],
      "Default",
                              2,           [288],
                              1                                           [0]
                        ]                                           ],
                  ]                                                 [
,                 [                                                 [
                        [827,                                               0,
534, 0, 59, 67, 0, 0, 0,                                                    1
0.508474588394165,                                                  ]
0.5074626803398132, 0, 0, []],                                      ],
                        23,                                         [
                        93,                                                 "q",
                        [                                                   0,

      ["u"]                                "90pt Arial Rounded MT
                        ],           Bold",
                        [
                        ],                 "rgb(255,255,87)",
                        [                                                   0,
                              0,                                            0,
                                                                            0,
      "Default",                                                            0,
                              3,                                            0
                              1                                     ]
                        ]                                     ]
                  ]                  ,                        [
,                 [
```

```
                        [513,
163, 0, 78, 144, 0, 0, 1, 0, 0,
0, 0, []],
                        22,
                        96,
                        [
      [512],
      [160],
                              [0]
                        ],
                        [
                        [
                              0,
                              1
                        ]
                        ],
                        [
                              "n",
                              0,
      "90pt Arial Rounded MT
Bold",
      "rgb(255,255,87)",
                              0,
                              0,
                              0,
                              0,
                              0
                        ]
                  ]
,                 [
                        [513,
291, 0, 78, 144, 0, 0, 1, 0, 0,
0, 0, []],
                        22,
                        97,
                        [
      [512],
      [228],
                              [0]
                        ],
                        [
                        [
                              0,
                              1
                        ]
                        ],
                        [
                              "c",
                              0,
                        "90pt Arial Rounded MT
                  Bold",
                        "rgb(255,255,87)",
                                                  0,
                                                  0,
                                                  0,
                                                  0,
                                                  0
                                                  ]
                                            ]
                  ,                         [
                                                  [417,
                  291, 0, 78, 144, 0, 0, 1, 0, 0,
                  0, 0, []],
                                                  22,
                                                  98,
                                                  [
                        [416],

                        [288],
                                                  [0]
                                                  ],
                                                  [
                                                  [
                                                  0,
                                                  1
                                                  ]
                                                  ],
                                                  [
                                                  "f",
                                                  0,
                        "90pt Arial Rounded MT
                  Bold",
                        "rgb(255,255,87)",
                                                  0,
                                                  0,
                                                  0,
                                                  0,
                                                  0
                                                  ]
                                            ]
                  ,                         [
                                                  [801,
                  163, 0, 84, 144, 0, 0, 1, 0, 0,
                  0, 0, []],
                                                  22,
                                                  99,
                                                  [
                        [800],
```

```
                                                                    0
      [160],                                                      ]
                              [0]                         ]
                        ],              ,                 [
                        [                                       [801,
                        [               291, 0, 78, 144, 0, 0, 1, 0, 0,
                              0,        0, 0, []],
                              1                                 22,
                        ]                                       101,
                        ],                                      [
                        [
                              "b",            [800],
                              0,
                                              [288],
      "90pt Arial Rounded MT                                          [0]
Bold",                                                            ],
                                                                  [
      "rgb(255,255,87)",                                          [
                              0,                                        0,
                              0,                                        1
                              0,                                  ]
                              0,                                  ],
                              0                                   [
                        ]                                             "a",
                  ]                                                   0,
,                 [
                        [897,                 "90pt Arial Rounded MT
291, 0, 78, 144, 0, 0, 1, 0, 0,       Bold",
0, 0, []],
                        22,                   "rgb(255,255,87)",
                        100,                                          0,
                        [                                             0,
                                                                      0,
      [869],                                                          0,
                                                                      0
      [288],                                                      ]
                              [0]                         ]
                        ],              ,                 [
                        [                                       [301, 25,
                        [               0, 165, 83, 0, 0, 1, 0, 0, 0,
                              0,        0, []],
                              1                                 36,
                        ]                                       111,
                        ],                                      [
                        [                                       ],
                              "t",                              [
                              0,                                ],
                                                                [
      "90pt Arial Rounded MT
Bold",                                "hallo",
                                                                      0,
      "rgb(255,255,87)",
                              0,            "48pt Arial Rounded MT
                              0,        Bold",
                              0,
                              0,            "rgb(255,0,0)",
```

```
                                0,                          ]
                                0,                          ],
                                0,                          [
                                0,                          ],
                                0                           []
                            ]                           ]
                        ]                               ],
,                       [                               [
                            [474, 24,                   [
0, 613, 190, 0, 0, 1, 0, 0, 0,                              "eventGameApel",
0, []],                                                     [
                            37,                             [
                            112,                                    2,
                            [                                       "eventHome",
                            ],                                      false
                            [                               ]
                            ],                      ,       [
                            [                                       2,
                                                                    "eventCaller",
      "nama",                                                       false
                                0,                          ]
                                                        ,       [
      "italic 48pt Arial Rounded                                    1,
MT Bold",                                                           "Match",
                                                                    0,
      "rgb(51,51,255)",                                             0,
                                0,                      false,false,7142177961714187,fa
                                0,                      lse
                                0,                          ]
                                0,                      ,       [
                                0                                   0,
                            ]                                       [true, "apel"],
                        ]                                           false,
,                       [                                           null,
                            [1204,
70, 0, 103.9230499267578,                               4468415853756644,
103.9230499267578, 0, 0, 1,                                         [
0.5020580291748047,                                                 [
0.5020580291748047, 0, 0, []],                                          -1,
                            45,
                            113,                            cr.system_object.prototype
                            [                           .cnds.IsGroupActive,
                            ],                                          null,
                            [                                           0,
                            ],                                          false,
                            [                                           false,
                                0,                                      false,

      "Default",                                            4468415853756644,
                                0,                                      false
                                1                                       ,[
                            ]                                           [
                        ]                                                   1,
                        ],                                                  [
                        [
      ]                                                     2,
```

"apel" cr.plugins_.Sprite.prototy
] pe.cnds.IsOverlapping,
]
] null,
]
], 0,
[
] false,
,[
[false,
0,
null, false,
false,
null, 601035106757226,

7116307806532878, false
[
[,[
6,
[
cr.behaviors.DragnDrop.pro
totype.cnds.OnDrop, 4,

"DragDrop", 6
1,
]
false,
]
false,]
],
false, [
]
3299477150102218, ,[
[
false
] 0,
],
[null,
]
,[false,
[
0, null,

null, 5490820305094724,

false, [

null, [

9122827998029729, 7,
[
[
cr.plugins_.Sprite.prototy
7, pe.cnds.CompareInstanceVar,

```
            null,
            0,
            false,
            false,
            false,
            5681574542148793,
            false
            ,[
            [
                  10,
                  0
            ]
,
            [
                  8,
                  0
            ]
,
            [
                  7,
                  [
                        20,
                        6,
      cr.plugins_.Text.prototype
.exps.Text,
                        true,
                        null
                  ]
            ]
            ]
            ]
            ],
            [
            [
                  6,
      cr.plugins_.Text.prototype
.acts.SetVisible,
                  null,
                  2096100788858905,
                  false
                  ,[
                  [
                        3,
                        0
                  ]
                  ]
            ]
,
            [
                  7,
      cr.plugins_.Sprite.prototy
pe.acts.SetOpacity,
                  null,
                  9817866752101906,
                  false
                  ,[
                  [
                        0,
                        [
```

```
                                                                        ]
                    0,                  ,                               [
                    100                         0,
                ]                               null,
            ]                                   false,
            ]                                   null,
        ]                                       6866605560151687,
,
        [                                       [
            -1,                                 [
                                                    7,
        cr.system_object.prototype
.acts.AddVar,
                                                cr.plugins_.Sprite.prototy
            null,                       pe.cnds.CompareInstanceVar,
            8542494087055549,                       null,
            false                                   0,
            ,[                                      false,
            [                                       true,
                11,                                 false,
                "Match"                             6690256182076532,
            ]                                       false
,
            [                                       ,[
                7,                                  [
                [                                       10,
                    0,                                  0
                    1                               ]
                                            ,
                ]                                   [
            ]                                       8,
            ]                                       0
        ]                                       ]
                                            ,
        ]                                       [
```

```
                7,                                          null
                [                                           ,0
                    20,                             ]
                    6,                          ]
                                        ,
                                                [
    cr.plugins_.Text.prototype
.exps.Text,                                             0,
                    true,                           [
                    null                                    21,
                ]                                           6,
            ]                                               false,
            ]                                               null
        ]                                                   ,1
        ],                                              ]
        [                                           ]
        [                                           ]
            6,                              ]
                                            ]
        cr.plugins_.Text.prototype                                  ]
.acts.SetPos,                                                       ]
                                                            ]
            null,                       ,                   [
                                                                    0,
            9504178223513678,
                                            null,
            false
                                            false,
            ,[
                                            null,
            [
                                            2253651788833785,
                0,                                                  [
                                                                    [
                [
                                            7,
                    21,
                                            cr.plugins_.Sprite.prototy
                    6,              pe.cnds.IsOverlapping,
                    false,                  null,
```

```
      0,
      false,
      true,
      false,
      1832119985417854,
      false
      ,[
      [
            4,
            6
      ]
      ]
      ]
,     [
      6,
      cr.plugins_.Text.prototype
.cnds.IsBoolInstanceVarSet,
      null,
      0,
      false,
      false,
      false,
      9848670766712201,
      false
      ,[
      [
            10,
            2
      ]
      ]
      6,
      cr.plugins_.Text.prototype
.acts.SetPos,
      null,
      3342163136377851,
      false
      ,[
      [
            0,
            [
                  21,
                  6,
                  false,
                  null
                  ,0
            ]
      ]
,     [
            0,
            [
                  21,
                  6,
                  false,
                  null
                  ,1
```

```
            ]                                   null
                                                                    ]
      ]                                                         ]
                                                                ]
      ]                                                     ]
                        ]                                   ],
                        ]                                   [
                  ]                                         [
                  ]                                               -1,
            ]
            ]                                       cr.system_object.prototype
            ]                           .acts.Wait,
,           [                                               null,
            0,
            null,                       8686263823427513,
            false,                                          false
            null,                                           ,[
                                                            [
      4401958249095924,                                           0,
            [                                                   [
            [
                  -1,                   0,

      cr.system_object.prototype        3
.cnds.CompareVar,                                                   ]
                  null,                                         ]
                  0,                                            ]
                  false,                                    ]
                  false,          ,                         [
                  false,                                        8,

      3869812617745388,                 cr.plugins_.Sprite.prototy
                  false           pe.acts.SetPos,
                  ,[                                            null,
                  [
                        11,         2666573652548173,
                                                                false
      "Match"                                                   ,[
                  ]                                             [
,                 [                                                   0,
                        8,                                          [
                        0
                  ]                 0,
,                 [
                        7,          667
                        [                                           ]
                                                                ]
      20,                         ,                             [
                                                                    0,
      7,                                                            [

      cr.plugins_.Sprite.prototy        0,
pe.exps.Count,
                                        368
      false,                                                        ]
```

```
                    ]               ,                       [
                    ]                                           7,
                ]                                               [
,               [
                    -1,                 0,

     cr.system_object.prototype         20
.acts.Wait,                                                     ]
                    null,                               ]
                                                        ]
     4032803265028199,                              ]
                    false                           ]
                    ,[                      ]
                    [                       ]
                        0,              ]
                        [       ,       [
                                            "eventGameNanas",
     0,                                     [
                                            [
     3                                          2,
                        ]                       "eventCaller",
                    ]                           false
                    ]                       ]
                ]                   ,       [
,               [                           2,
                    -1,                     "eventHome",
                                            false
     cr.system_object.prototype             ]
.acts.GoToLayout,                   ,       [
                    null,                       1,
                                                "Match2",
     2318273468967281,                          0,
                    false                       0,
                    ,[              false,false,7917483802187251,fa
                    [               lse
                        6,                  ]
                                    ,       [
     "GameNanas"                                0,
                    ]                           [true,
                    ]               "nanas"],
                ]                               false,
,               [                               null,
                    -1,
                                        7545172699145373,
     cr.system_object.prototype                 [
.acts.AddVar,                                   [
                    null,                           -1,

     5054009556874013,                  cr.system_object.prototype
                    false           .cnds.IsGroupActive,
                    ,[                                  null,
                    [                                   0,
                        11,                             false,
                                                        false,
     "Score"                                            false,
                    ]
```

```
7545172699145373,                    null,
                 false
                 ,[                  9122827998029729,
                 [                                        [
                       1,                                 [
                       [
                                     11,
2,
                                     cr.plugins_.Sprite.prototy
"nanas"                        pe.cnds.IsOverlapping,
                       ]
                 ]                   null,
                 ]
            ]                        0,
            ],
            [                        false,
            ]
            ,[                       false,
            [
                 0,                  false,
                 null,
                 false,              601035106757226,
                 null,
                                     false
7116307806532878,
                 [                   ,[
                 [
                       10,           [

cr.behaviors.DragnDrop.pro                 4,
totype.cnds.OnDrop,
                                           10
"DragDrop",
                       1,            ]

false,                               ]
                                                         ]
false,                                                   ],
                                                         [
false,                                                   ]
                                                         ,[
3299477150102218,                                        [

false                                0,
                 ]
                 ],                  null,
                 [
                 ]                   false,
                 ,[
                 [                   null,
                       0,
                                     5490820305094724,
null,
                                     [
false,
```

```
      [                                                                null
            11,                                                  ]
                                                        ]
      cr.plugins_.Sprite.prototy
pe.cnds.CompareInstanceVar,                             ]
            null,                                 ]
            0,                                    ],
            false,                                [
            false,                                [
            false,                                      10,
            5681574542148793,
                                                  cr.plugins_.Text.prototype
            false                           .acts.SetVisible,
            ,[                                          null,
            [                                           2096100788858905,
                  10,                                   false
                  0                                     ,[
            ]                                           [
,
            [                                                  3,
                  8,                                           0
                  0                                     ]
            ]                                           ]
,
            [                                     ]
                                            ,
                  7,                              [
                  [                                     11,
                        20,
                                                  cr.plugins_.Sprite.prototy
                        10,                     pe.acts.SetOpacity,
                                                        null,
      cr.plugins_.Text.prototype
.exps.Text,                                             9817866752101906,
                        true,                           false
```

```
        ,[
        [
            0,
            [
                0,
                100
            ]
        ]
        ]
    ]
,
    [
        -1,

    cr.system_object.prototype
.acts.AddVar,
        null,
        8542494087055549,
        false
        ,[
        [
            11,
            "Match2"
        ]
,
        [
            7,
            [
                0,
                1
            ]
```

```
        ]
        ]
    ]
    ]
                                ]
,                               [
    0,
    null,
    false,
    null,
    6866605560151687,
    [
    [
        11,

    cr.plugins_.Sprite.prototy
pe.cnds.CompareInstanceVar,
        null,
        0,
        false,
        true,
        false,
        6690256182076532,
        false
        ,[
        [
            10,
            0
        ]
,
        [
```

```
                8,                                      [
                0                                               21,
            ]                                               10,
,
            [                                               false,
                7,                                          null
                [                                           ,0
                    20,                                 ]
                    10,                         ]
                                        ,
                                                [
      cr.plugins_.Text.prototype
.exps.Text,                                             0,
                    true,                               [
                    null                                        21,
                ]                                               10,
            ]                                                   false,
            ]                                                   null
      ]                                                         ,1
      ],                                                    ]
      [                                         ]
      [                                         ]
            10,                             ]
                                            ]
      cr.plugins_.Text.prototype                                    ]
.acts.SetPos,                                                       ]
                                                            ]
            null,                       ,                   [
                                                                  0,
            9504178223513678,
                                            null,
            false
                                            false,
            ,[
                                            null,
            [
                                            2253651788833785,
                0,                                                  [
```

```
                        [
                                                [
      11,                                                   10,
      cr.plugins_.Sprite.prototy                            2
pe.cnds.IsOverlapping,
      null,                                     ]
      0,                                        ]
                                                                        ]
      false,                                                            ],
                                                                        [
      true,                                                             [
      false,                                    10,
      1832119985417854,                         cr.plugins_.Text.prototype
                                        .acts.SetPos,
      false
                                                null,
      ,[
                                                3342163136377851,
      [
                                                false
            4,
                                                ,[
            10
                                                [
      ]
                                                      0,
      ]
                        ]                       [
,                       [
                                                            21,
      10,
                                                            10,
      cr.plugins_.Text.prototype
.cnds.IsBoolInstanceVarSet,                                 false,
      null,                                                 null
      0,                                                    ,0
      false,                                        ]
      false,                                    ]
                                        ,
      false,                                    [
      9848670766712201,                             0,
      false                                         [
      ,[                                                    21,
```

```
                10,                         20,

                false,                      11,

                null                        cr.plugins_.Sprite.prototy
                                          pe.exps.Count,
                ,1
                                            false,
            ]
                                            null
        ]                                                           ]
                                                            ]
        ]                                                   ]
                            ]                           ]
                            ]                           ],
                        ]                               [
                        ]                               [
                ]                                           -1,
                ]
            ]                               cr.system_object.prototype
,           [                             .acts.Wait,
                0,                                          null,
                null,
                false,                      4710162831462175,
                null,                                       false
                                                            ,[
        172228494949041,                                    [
                [                                                   0,
                [                                                   [
                        -1,
                                            0,
        cr.system_object.prototype
.cnds.CompareVar,                           3
                        null,                                       ]
                        0,                                  ]
                        false,                              ]
                        false,                      ]
                        false,          ,           [
                                                            13,
        3085676412805815,
                        false               cr.plugins_.Sprite.prototy
                        ,[                pe.acts.SetPos,
                        [                                   null,
                            11,
                                            9506208972704106,
        "Match2"                                            false
                        ]                                   ,[
,                       [                                   [
                            8,                                      0,
                            0                                       [
                        ]
,                       [                   0,
                            7,
                            [               667
                                                                    ]
```

```
                    ]
,                   [                   5129474400409499,
                          0,                                false
                          [                                 ,[
                                                            [
      0,                                                          11,

      368                                 "Score"
                          ]                                 ]
                    ]               ,                       [
                    ]                                           7,
                ]                                               [
,               [
                    -1,                 0,

      cr.system_object.prototype        20
.acts.Wait,                                                     ]
                    null,                                   ]
                                                            ]
      7426013968323202,                                 ]
                    false                           ]
                    ,[                        ]
                    [                         ]
                          0,            ]
                          [         ,   [
                                        "eventGameMangga",
      0,                                [
                                        [
      3                                     2,
                          ]                 "eventCaller",
                    ]                       false
                    ]                   ]
                ]                   ,   [
,               [                           2,
                    -1,                     "eventHome",
                                            false
      cr.system_object.prototype        ]
.acts.GoToLayout,                   ,   [
                    null,                   1,
                                            "Match3",
      462420049557336,                      0,
                    false                   0,
                    ,[              false,false,1646408545169134,fa
                    [               lse
                          6,            ]
                                    ,   [
      "GameMangga"                          0,
                    ]                       [true,
                    ]               "mangga"],
                ]                           false,
,               [                           null,
                    -1,
                                        5927845589078944,
      cr.system_object.prototype            [
.acts.AddVar,                               [
                    null,                       -1,
```

```
                                                                ]
      cr.system_object.prototype                                ,[
.cnds.IsGroupActive,                                            [
                        null,                                         0,
                        0,
                        false,              null,
                        false,
                        false,              false,

      5927845589078944,                     null,
                        false
                        ,[                  9122827998029729,
                        [                                             [
                              1,                                      [
                              [
                                            17,
      2,
                                            cr.plugins_.Sprite.prototy
      "mangga"                        pe.cnds.IsOverlapping,
                              ]
                        ]                   null,
                        ]
                  ]                         0,
                  ],
                  [                         false,
                  ]
                  ,[                        false,
                  [
                        0,                  false,
                        null,
                        false,              601035106757226,
                        null,
                                            false
      7116307806532878,
                        [                   ,[
                        [
                              15,           [

      cr.behaviors.DragnDrop.pro                  4,
totype.cnds.OnDrop,
                                                  15
      "DragDrop",
                              1,            ]

      false,                                ]
                                                                    ]
      false,                                                        ],
                                                                    [
      false,                                                        ]
                                                                    ,[
      3299477150102218,                                             [

      false                                 0,
                        ]
                        ],                  null,
                        [
```

false,
null,
5490820305094724,
[
[
17,
cr.plugins_.Sprite.prototype.cnds.CompareInstanceVar,
null,
0,
false,
false,
false,
5681574542148793,
false
,[
[
10,
0
]
,
[
8,
0
]
,
[
7,
[
20,
15,
cr.plugins_.Text.prototype.exps.Text,
true,
null
]
]
]
]
,
[
15,
cr.plugins_.Text.prototype.acts.SetVisible,
null,
2096100788858905,
false
,[
[
3,
0
]
]
]
,
[
17,

```
     cr.plugins_.Sprite.prototy
pe.acts.SetOpacity,
          null,
          9817866752101906,
          false
          ,[
          [
               0,
               [
                    0,
                    100
               ]
          ]
          ]
     ]
,
     [
          -1,
     cr.system_object.prototype
.acts.AddVar,
          null,
          8542494087055549,
          false
          ,[
          [
               11,
               "Match3"
          ]
,
          [
               7,
```

```
               [
                    0,
                    1
               ]
          ]
          ]
     ]
     ]
                         ]
,                        [
     0,
     null,
     false,
     null,
     6866605560151687,
     [
     [
          17,
     cr.plugins_.Sprite.prototy
pe.cnds.CompareInstanceVar,
          null,
          0,
          false,
          true,
          false,
          6690256182076532,
          false
          ,[
          [
```

```
                10,                             false
                0                               ,[
            ]                                   [
,
            [                                       0,
                8,                                  [
                0                                       21,
            ]                                           15,
,
            [                                           false,
                7,                                      null
                [                                       ,0
                    20,                             ]
                    15,                         ]
                                    ,
                                                [
      cr.plugins_.Text.prototype                    0,
.exps.Text,
                    true,                           [
                    null                                21,
                ]                                       15,
            ]                                           false,
            ]                                           null
      ]                                                 ,1
      ],                                            ]
      [                                         ]
      [                                         ]
            15,                             ]
                                            ]
      cr.plugins_.Text.prototype                            ]
.acts.SetPos,                                               ]
                                                        ]
            null,                   ,                   [
                                                            0,
            9504178223513678,
```

null,

false,

null,

2253651788833785,

17,

cr.plugins_.Sprite.prototype.cnds.IsOverlapping,

null,

0,

false,

true,

false,

1832119985417854,

false

,[

[

4,

15

]

]

,

10,

cr.plugins_.Text.prototype.cnds.IsBoolInstanceVarSet,

null,

0,

false,

false,

false,

9848670766712201,

false

,[

[

10,

2

]

]

15,

cr.plugins_.Text.prototype.acts.SetPos,

null,

3342163136377851,

false

,[

[

0,

[

21,

15,

false,

null

,0

]

]

]

],

[

[

]

```
,
      [
            0,
            [
                  21,
                  15,
                  false,
                  null
                  ,1
            ]
      ]
      ]
                        ]
                        ]
                  ]
                  ]
                  ]
                  ]
            ]
,           [
                  0,
                  null,
                  false,
                  null,
      7996782938068451,
                  [
                  [
                        -1,
      cr.system_object.prototype
.cnds.CompareVar,
                        null,
                        0,
                        false,
                        false,
                        false,
      4063447790162572,
                        false
                        ,[
                        [
                              11,
      "Match3"
                        ]
,                                   [
                                          8,
                                          0
                                    ]
,                                   [
                                          7,
                                          [
                                          20,
                                          17,
                                          cr.plugins_.Sprite.prototy
pe.exps.Count,
                                          false,
                                          null
                                          ]
                                    ]
                                    ]
                                    ]
                                    ],
                                    [
                                    [
                                          -1,
                                          cr.system_object.prototype
.acts.Wait,
                                          null,
                                          2827539725972915,
                                          false
                                          ,[
                                          [
                                                0,
                                                [
                                          0,
                                          3
                                          ]
                                          ]
                                          ]
                                    ]
                                    [
                                          19,
                                          cr.plugins_.Sprite.prototy
pe.acts.SetPos,
                                          null,
                                          4355150067997034,
                                          false
                                          ,[
                                          [
```

```
                        0,                                      7,
                        [                                       [

      0,                                  0,

      667                                 20
                        ]                                       ]
                  ]                                     ]
,                 [                                     ]
                        0,                        ]
                        [                         ]
                                                ]
      0,                                        ]
                                          ]
      368                         ,     [
                        ]                       "eventHome",
                  ]                             [
                  ]                             [
            ]                                         2,
,           [                                         "eventCaller",
                  -1,                               false
                                                ]
      cr.system_object.prototype      ,         [
.acts.Wait,                                       0,
                  null,                           null,
                                                  false,
      8867781953868606,                           null,
                  false
                  ,[                      1357394184518671,
                  [                                 [
                        0,                          [
                        [                                 2,

      0,                                  cr.plugins_.Touch.prototyp
                                    e.cnds.OnTouchObject,
      3                                                   null,
                        ]                                 1,
                  ]                                       false,
                  ]                                       false,
            ]                                             false,
,           [
                  -1,                     7602358214382441,
                                                          false
      cr.system_object.prototype                          ,[
.acts.AddVar,                                             [
                  null,                                         4,
                                                                44
      1626630180288728,                                   ]
                  false                                   ]
                  ,[                                ]
                  [                                 ],
                        11,                         [
                                                    [
      "Score"                                             44,
                  ]
,                 [
```

```
                                                                    ]
      cr.plugins_.Sprite.prototy                          ]
pe.acts.SetPos,                                           ],
                        null,                             [
                                                          [
      3653745315601777,                                         -1,
                        false
                        ,[                                cr.system_object.prototype
                        [                           .acts.GoToLayout,
                              0,                                    null,
                              [
                                                          5915123832830291,
      0,                                                            false
                                                                    ,[
      -760                                                          [
                              ]                                           6,
                        ]
,                       [                                 "GameApel"
                              0,                                    ]
                              [                                     ]
                                                                ]
      0,                                                        ]
                                                                ,[
      -888                                                      [
                              ]                                     0,
                        ]                                           null,
                        ]                                           false,
                  ]                                                 null,
                  ]
            ]                                             9679622105783494,
,           [                                                       [
                  0,                                                [
                  null,                                                   39,
                  false,
                  null,                                   cr.plugins_.TextBox.protot
                                                    ype.cnds.CompareInstanceVar,
      4520486084657536,
                  [                                       null,
                  [                                                       0,
                        2,
                                                          false,
      cr.plugins_.Touch.prototyp
e.cnds.OnTouchObject,                                     false,
                        null,
                        1,                                false,
                        false,
                        false,                            5309942039599919,
                        false,
                                                          false
      2358310887445634,                                                   ,[
                        false                                             [
                        ,[
                        [                                 10,
                              4,
                              1                           0
                        ]                                                 ]
```

```
,                           [
                                      cr.system_object.prototype
      8,                            .cnds.Else,

      0                               null,
                            ]                                 0,
,                           [
                                      false,
      7,
                                      false,
      [
                                      false,
            20,
                                      2919368959421816,
            39,
                                      false
                                                        ]
      cr.plugins_.TextBox.protot                        ],
ype.exps.Text,                                          [
                                                        [
            true,                                             -1,

            null                      cr.system_object.prototype
                                  .acts.AddVar,
      ]
                            ]         null,
                            ]
                        ]             4403463698485485,
                        ],
                        [             false
                        [                                     ,[
                            39,                               [

      cr.plugins_.TextBox.protot      11,
ype.acts.SetFocus,
                                      "username"
      null,                                                   ]
                                  ,                           [
      2342073906177248,
                                      7,
      false
                        ]             [
                        ]
                  ]                         20,
,                 [
                        0,                  39,
                        null,
                        false,
                        null,         cr.plugins_.TextBox.protot
                                  ype.exps.Text,
      2375774950031283,
                        [                   true,
                        [
                            -1,             null

                                      ]
```

```
                        ]                                   37,
                        ]
                    ]                       cr.plugins_.Text.prototype
,                   [                   .cnds.IsOnScreen,
                        -1,                                 null,
                                                            0,
    cr.system_object.prototype                              false,
.acts.AddVar,                                               false,
                                                            false,
    null,
                                            4313036054395373,
    1440681948092139,                                       false
                                                        ]
    false                                               ],
                        ,[                              [
                        [                               [
                                                            37,
    11,
                                            cr.plugins_.Text.prototype
    "player"                            .acts.SetText,
                        ]                                   null,
,                       [
                                            5070634954081294,
    7,                                                      false
                                                            ,[
    [                                                       [
                                                                7,
        20,                                                     [

        39,                                 23,

                                            "username"
    cr.plugins_.TextBox.protot                                  ]
ype.exps.Text,                                              ]
                                                            ]
        true,                                           ]
                                                        ]
        null                                        ]
                                        ,           [
    ]                                                   0,
                        ]                               null,
                        ]                               false,
                    ]                                   null,
                    ]
                ]                           4530293244059871,
                ]                                       [
        ]                                               [
,       [                                                   39,
                0,
                null,                       cr.plugins_.TextBox.protot
                false,                  ype.cnds.OnClicked,
                null,                                       null,
                                                            1,
    2074777175936829,                                       false,
                [                                           false,
                [                                           false,
```

```
                                                        0,
      6974414456725952,                                 0,
                        false           false,false,1362333675107603,fa
                  ]                     lse
                  ],                              ]
                  [                     ,         [
                  [                                 0,
                        39,                         [true,
                                        "Semangka"],
      cr.plugins_.TextBox.protot                    false,
ype.acts.SetCSSStyle,                                null,
                        null,
                                              2937217301603955,
      3446530985202442,                             [
                        false                       [
                        ,[                                -1,
                        [
                              1,            cr.system_object.prototype
                              [         .cnds.IsGroupActive,
                                                          null,
      2,                                                  0,
                                                          false,
      "font-size"                                         false,
                              ]                           false,
                        ]
,                       [                     2937217301603955,
                              1,                          false
                              [                           ,[
                                                          [
      2,                                                        1,
                                                                [
      "2em"
                              ]             2,
                        ]
                        ]                   "Semangka"
                  ]                                             ]
                  ]                                       ]
            ]                                             ]
            ]                                         ]
      ]                                               ],
,     [                                               [
            "eventSemangka",                          ]
            [                                         ,[
            [                                         [
                  2,                                      0,
                  "eventCaller",                          null,
                  false                                   false,
            ]                                             null,
,           [
                  2,                      7116307806532878,
                  "eventHome",                            [
                  false                                   [
            ]                                                 27,
,           [
                  1,                      cr.behaviors.DragnDrop.pro
                  "Match5",         totype.cnds.OnDrop,
```

"DragDrop",
1,
false,
false,
false,
3299477150102218,
false
]
],
[
]
,[
[
0,
null,
false,
null,
9122827998029729,
[
[
28,
cr.plugins_.Sprite.prototype.cnds.IsOverlapping,
null,
0,
false,
false,
false,
601035106757226,
false
,[
[
4,
27
]
]
]
]
,[
[
0,
null,
5490820305094724,
[
[
28,
cr.plugins_.Sprite.prototype.cnds.CompareInstanceVar,
null,
0,
false,
false,
false,
5681574542148793,
false
,[
[
10,
0
]

,
[
8,
0
]
,
[
7,
[
20,
27,
cr.plugins_.Text.prototype
.exps.Text,
true,
null
]
]
]
]
],
[
[
27,
cr.plugins_.Text.prototype
.acts.SetVisible,
null,
2096100788858905,
false
,[
[
3,
0
]
]
]
,
[
28,
cr.plugins_.Sprite.prototy
pe.acts.SetOpacity,
null,
9817866752101906,
false
,[
[
0,
[
0,
100
]
]
]
]
,
[
-1,
cr.system_object.prototype
.acts.AddVar,
null,
8542494087055549,

```
            false
            ,[
            [
                  11,
                  "Match5"
            ]
,
            [
                  7,
                  [
                        0,
                        1
                  ]
            ]
            ]
      ]
      ]
,
      0,
      null,
      false,
      null,
      6866605560151687,
      [
      [
            28,

      cr.plugins_.Sprite.prototy
pe.cnds.CompareInstanceVar,
            null,
            0,
            false,
            true,
            false,
            6690256182076532,
            false
            ,[
            [
                  10,
                  0
            ]
            [
                  8,
                  0
            ]
            [
                  7,
                  [
                        20,
                        27,
                  cr.plugins_.Text.prototype
.exps.Text,
                        true,
                        null
                  ]
            ]
            ]
      ]
      ]
,
      ]
```

```
      ],                                                  ]

      [                                          ]

      [                                          ]

            27,                              ]

                                             ]
      cr.plugins_.Text.prototype                                  ]
.acts.SetPos,                                                     ]
                                                            ]
            null,                     ,                     [
                                                                  0,
            9504178223513678,
                                             null,
            false
                                             false,
            ,[
                                             null,
            [
                                             2253651788833785,
                  0,                                              [
                                                                  [
                  [
                                             28,
                        21,
                                             cr.plugins_.Sprite.prototy
                        27,            pe.cnds.IsOverlapping,

                        false,               null,

                        null                 0,

                        ,0                   false,

                  ]                          true,

            ]                                false,
,
            [                                1832119985417854,

                  0,                         false

                  [                          ,[

                        21,                  [

                        27,                        4,

                        false,                     27

                        null                 ]

                        ,1                   ]
```

```
                                ]
,                               [                                21,

      27,                                                        27,

      cr.plugins_.Text.prototype                                 false,
.cnds.IsBoolInstanceVarSet,
                                                                 null
      null,
                                                                 ,0
      0,
                                                        ]
      false,
                                                  ]
      false,                            ,
                                                  [
      false,
                                                        0,
      9848670766712201,
                                                        [
      false
                                                                 21,
      ,[
                                                                 27,
      [
                                                                 false,
            10,
                                                                 null
            2
                                                                 ,1
      ]
                                                        ]
      ]
                                ]               ]
                                ],
                                [               ]
                                [                                        ]
                                                                         ]
      27,                                                              ]
                                                                       ]
      cr.plugins_.Text.prototype                                 ]
.acts.SetPos,                                                    ]
                                                        ]
      null,                             ,             [
                                                                 0,
      3342163136377851,                                          null,
                                                                 false,
      false                                                      null,

      ,[                                          3798550380066826,
                                                                 [
      [                                                          [
                                                                       -1,
            0,
                                                  cr.system_object.prototype
            [                           .cnds.CompareVar,
```

null,]
0,]
false,]
false, , [
false, 29,
5167791079330566, cr.plugins_.Sprite.prototy
false pe.acts.SetPos,
,[null,
[
11, 2970722245826924,
false
"Match5" ,[
] [
, [0,
8, [
0
] 0,
, [
7, 667
[ ]
]
20, , [
0,
28, [
cr.plugins_.Sprite.prototy 0,
pe.exps.Count,
368
false,]
]
null]
]]
] , [
] -1,
]
], cr.system_object.prototype
[.acts.Wait,
[null,
-1,
8256289504717846,
cr.system_object.prototype false
.acts.Wait, ,[
null, [
0,
3058763495181827, [
false
,[0,
[
0, 3
[ ]
]
0,]
]
3 , [
] -1,

```
                                                        [
      cr.system_object.prototype                                2,
.acts.GoToLayout,
                        null,                 cr.plugins_.Touch.prototyp
                                        e.cnds.OnTouchObject,
      1221460994204566,                                         null,
                        false                                   1,
                        ,[                                      false,
                        [                                       false,
                              6,                                false,

      "Home"                                  5871041168198688,
                        ]                                       false
                        ]                                       ,[
                  ]                                             [
,                 [                                                   4,
                        -1,                                           30
                                                                ]
      cr.system_object.prototype                                ]
.acts.AddVar,                                           ]
                        null,                           ],
                                                        [
      7850060908234018,                                 [
                        false                                   -1,
                        ,[
                        [                                 cr.system_object.prototype
                              11,         .acts.GoToLayout,
                                                                null,
      "Score"
                        ]                                 685191220566931,
,                       [                                       false
                              7,                                ,[
                              [                                 [
                                                                    6,
      0,
                                                        "GameApel"
      20                                                        ]
                              ]                                 ]
                        ]                               ]
                        ]                               ]
                  ]                                   ]
                  ]                         ,         [
            ]                                           0,
            ]                                           null,
      ]                                                 false,
,     [                                                 null,
            "eventGameSelect",
            [                                           1596619682318049,
            [                                           [
                  0,                                    [
                  null,                                       2,
                  false,
                  null,                           cr.plugins_.Touch.prototyp
                                        e.cnds.OnTouchObject,
      170732902232754,                                          null,
                  [                                             1,
```

```
                    false,
                    false,
                    false,

    3678498044538979,
                    false
                    ,[
                    [
                        4,
                        31
                    ]
                    ]
                ]
                ],
                [
                ]
            ]
,           [
                0,
                null,
                false,
                null,

    8245036851077273,
                [
                [
                    2,

    cr.plugins_.Touch.prototyp
e.cnds.OnTouchObject,
                    null,
                    1,
                    false,
                    false,
                    false,

    7556726251385684,
                    false
                    ,[
                    [
                        4,
                        33
                    ]
                    ]
                ]
                ],
                [
                [
                    -1,

    cr.system_object.prototype
.acts.GoToLayout,
                    null,

    4874394133897088,
                    false
                    ,[
                    [
                        6,
                    "Help"
                    ]
                    ]
                ]
                ]
            ]
,           [
                "eventCaller",
                [
                [
                    1,
                    "player",
                    1,
                    "",
false,false,1909706221303689,fa
lse
                ]
,               [
                    1,
                    "username",
                    1,
                    "",
false,false,1249434712286548,fa
lse
                ]
,               [
                    1,
                    "Score",
                    0,
                    0,
false,false,5458875965822442,fa
lse
                ]
,               [
                    0,
                    null,
                    false,
                    null,

                7095284434636507,
                    [
                    [
                        2,

                cr.plugins_.Touch.prototyp
e.cnds.OnTouchObject,
                        null,
                        1,
                        false,
                        false,
```

false,
6624632999473662,
false
,[
[
4,
35
]
]
]
],
[
[
-1,
cr.system_object.prototype
.acts.GoToLayout,
null,
7936594410205981,
false
,[
[
6,
"Home"
]
]
]
,
[
-1,
cr.system_object.prototype
.acts.ResetGlobals,
null,
3793735886757943,
false
]
]
]
,
[
0,
null,
false,
null,
5405564719572315,
[
[
2,
cr.plugins_.Touch.prototyp
e.cnds.OnTouchObject,
null,
1,
false,
false,
false,
728207645100678,
false
,[
[
4,
33
]
]
]
],
[
[
44,
cr.plugins_.Sprite.prototy
pe.acts.SetPos,
null,
3233536399742597,
false
,[
[
0,
[
0,
640
]
]
,
[
0,
[
0,
348
]
]
]
]
]
,
[
0,
null,
false,
null,
3087568394680876,
[

```
                [                                                       [
                      2,
                                                  0,
     cr.plugins_.Touch.prototyp
e.cnds.OnTouchObject,                             10
                      null,                                             ]
                      1,                                        ]
                      false,                                    ]
                      false,                              ]
                      false,                              ]
                                                    ]
     5189445441388846,                              ]
                      false                         ]
                      ,[                  ,     [
                      [                             "eventHalAkhir",
                            4,                      [
                            45                      [
                      ]                                   2,
                      ]                                   "eventHome",
                ]                                         false
                ],                                  ]
                [                         ,         [
                [                                   0,
                      -1,                           null,
                                                    false,
     cr.system_object.prototype                     null,
.acts.GoToLayout,
                      null,                         1528176464185622,
                                                          [
     7969075932142844,                                    [
                      false                                     41,
                      ,[
                      [                             cr.plugins_.Text.prototype
                            6,                .cnds.IsOnScreen,
                                                            null,
     "hlmAkhir"                                             0,
                      ]                                     false,
                      ]                                     false,
                ]                                           false,
,               [
                      -1,                           8177520651667001,
                                                            false
     cr.system_object.prototype                          ]
.acts.AddVar,                                            ],
                      null,                               [
                                                          [
     1939395602657794,                                      41,
                      false
                      ,[                            cr.plugins_.Text.prototype
                      [                     .acts.SetText,
                            11,                             null,

     "Score"                                        3928873766409208,
                      ]                                     false
,                     [                                     ,[
                            7,                              [
```

```
                        7,                              false
                        [                               ,[
                                                        [
      23,                                                     1,
                                                              [
      "Score"
                        ]               2,
                  ]
                  ]                     "Jeruk"
            ]                                                 ]
            ]                                           ]
      ]                                                 ]
      ]                                           ]
,     [                                           ],
            "eventGameJeruk",                     [
            [                                     ]
            [                                     ,[
                  2,                              [
                  "eventCaller",                        0,
                  false                                 null,
            ]                                           false,
,           [                                           null,
                  2,                      7116307806532878,
                  "eventHome",                          [
                  false                                 [
            ]                                                 22,
,           [
                  1,                            cr.behaviors.DragnDrop.pro
                  "Match4",           totype.cnds.OnDrop,
                  0,
                  0,                            "DragDrop",
false,false,1126079736781616,fa                         1,
lse
            ]                                   false,
,           [
                  0,                            false,
                  [true,
"Jeruk"],                                       false,
                  false,
                  null,                         3299477150102218,

      7440653661019765,                         false
                  [                                     ]
                  [                                     ],
                        -1,                             [
                                                        ]
      cr.system_object.prototype                        ,[
.cnds.IsGroupActive,                                    [
                        null,                                 0,
                        0,
                        false,                  null,
                        false,
                        false,                  false,

      7440653661019765,                         null,
```

```
      9122827998029729,
      23,
      cr.plugins_.Sprite.prototype.cnds.IsOverlapping,
      null,
      0,
      false,
      false,
      false,
      601035106757226,
      false
      ,[
      [
            4,
            22
      ]
      ]
                        ]
                        ],
                        [
                        ]
                        ,[
                        [
                              23,
                              [
                              [
                                    cr.plugins_.Sprite.prototype.cnds.CompareInstanceVar,
                              null,
                              0,
                              false,
                              false,
                              false,
                              5681574542148793,
                              false
                              ,[
                              [
                                    10,
                                    0
                              ]
                              [
                                    8,
                                    0
                              ]
                              [
                              [
                                    7,
                                    [
                                          20,
                                          22,
                                    cr.plugins_.Text.prototype.exps.Text,
                                          true,
                                          null
      5490820305094724,
      [
      [
```

```
                ]
            ]
            ]
        ]
        ],
        [
        [
            22,
        cr.plugins_.Text.prototype.acts.SetVisible,
            null,
            2096100788858905,
            false
            ,[
            [
                3,
                0
            ]
            ]
        ]
        [
            23,
        cr.plugins_.Sprite.prototype.acts.SetOpacity,
            null,
            9817866752101906,
            false
            ,[
                [
                    0,
                    [
                        0,
                        100
                    ]
                ]
                ]
            ]
            [
                -1,
            cr.system_object.prototype.acts.AddVar,
                null,
                8542494087055549,
                false
                ,[
                [
                    11,
                    "Match4"
                ]
                ,
                [
                    7,
                    [
                        0,
                        1
                    ]
                ]
```

```
            ]
        ]
        ]
,
        0,
        null,
        false,
        null,
        6866605560151687,
        [
        [
            23,
        cr.plugins_.Sprite.prototy
pe.cnds.CompareInstanceVar,
            null,
            0,
            false,
            true,
            false,
            6690256182076532,
            false
            ,[
            [
                10,
                0
            ]
,
            [
                8,
                0
            ]
            [
                7,
                [
                    20,
                    22,
                cr.plugins_.Text.prototype
.exps.Text,
                    true,
                    null
                ]
            ]
            ]
        ]
        ],
        [
        [
            22,
            cr.plugins_.Text.prototype
.acts.SetPos,
            null,
            9504178223513678,
            false
            ,[
            [
                0,
                [
```

```
                    21,                 23,
                    22,                 cr.plugins_.Sprite.prototy
                                      pe.cnds.IsOverlapping,
                    false,
                                        null,
                    null
                                        0,
                    ,0
                                        false,
                ]
                                        true,
            ]
,                                       false,
            [
                                        1832119985417854,
                0,
                                        false
                [
                                        ,[
                    21,
                                        [
                    22,
                                            4,
                    false,
                                            22
                    null
                                        ]
                    ,1
                                        ]
                ]                                           ]
                                      ,                     [
            ]
                                        22,
            ]
                                        cr.plugins_.Text.prototype
        ]                             .cnds.IsBoolInstanceVarSet,

        ]                               null,
                        ]
                        ]               0,
                    ]
,                   [                   false,
                        0,
                                        false,
        null,
                                        false,
        false,
                                        9848670766712201,
        null,
                                        false
        2253651788833785,
                        [               ,[
                        [
                                        [
```

```
            10,                                         false,

            2                                           null

      ]                                                 ,1

      ]                                           ]
                              ]
                              ],              ]
                              [
                              [               ]
                                                                    ]
      15,                                                           ]
                                                                ]
      cr.plugins_.Text.prototype                                ]
.acts.SetPos,                                           ]
                                                        ]
      null,                                         ]
                                        ,           [
      3342163136377851,                                 0,
                                                        null,
      false                                             false,
                                                        null,
      ,[
                                            3310976227814638,
      [                                                 [
                                                        [
            0,                                                  -1,

            [                               cr.system_object.prototype
                                        .cnds.CompareVar,
                  21,                                           null,
                                                                0,
                  15,                                           false,
                                                                false,
                  false,                                        false,

                  null                      415137643572535,
                                                                false
                  ,0                                            ,[
                                                                [
            ]                                                       11,

      ]                                         "Match4"
,                                                               ]
      [                                 ,                       [
                                                                      8,
            0,                                                        0
                                                                ]
            [                           ,                       [
                                                                      7,
                  21,                                                 [

                  15,                       20,
```

```
                                                                    0,
      23,                                                           [

      cr.plugins_.Sprite.prototy            0,
pe.exps.Count,
                                            368
      false,                                                        ]
                                                                  ]
      null                                                        ]
                        ]                                       ]
                  ]               ,                             [
                  ]                                               -1,
            ]
            ],                          cr.system_object.prototype
            [                     .acts.Wait,
            [                                                     null,
                  -1,
                                            9010200391957553,
      cr.system_object.prototype                                  false
.acts.Wait,                                                       ,[
                  null,                                           [
                                                                      0,
      3790198969092679,                                             [
                  false
                  ,[                        0,
                  [
                        0,                  3
                        [                                         ]
                                                                  ]
      0,                                                          ]
                                                                ]
      3                           ,                             [
                        ]                                         -1,
                  ]
                  ]                         cr.system_object.prototype
            ]                     .acts.GoToLayout,
,           [                                                     null,
                  24,
                                            7570133771385381,
      cr.plugins_.Sprite.prototy                                  false
pe.acts.SetPos,                                                   ,[
                  null,                                           [
                                                                      6,
      1543586857451031,
                  false                     "GameSemangka"
                  ,[                                              ]
                  [                                               ]
                        0,                                      ]
                        [           ,                           [
                                                                  -1,
      0,
                                            cr.system_object.prototype
      667                           .acts.AddVar,
                        ]                                         null,
                  ]
,                 [                         6933487650976236,
```

```
                        false
                        ,[
                        [
                              11,

      "Score"
                        ]
,                       [
                              7,
                              [

      0,

      20
                              ]
                        ]
                        ]
                  ]
                  ]
            ]
            ]
      ]
      ],
      "media/",
      false,
      1280,
      768,
      4,
      true,
      true,
      true,
      "1.0.0.0",
      true,
      false,
      0,
      0,
      204,
      false,
      true,
      [
      ]
];};
```