

BASIC OBJECTS

**circle** (center, size, color)  
**cone** (center, size, color)  
**cube** (center, size, color)  
**cylinder** (center, size, color)  
**line** (center/from, to, color)  
**point** (center, size, color)  
**polygon** (count, center, size, color)  
**prism** (count, center, size, color)  
**pyramid** (count, center, size, color)  
**sphere** (center, size, color)  
**square** (center, size, color)

BASIC PROPERTIES

**.center** = [.x, .y, .z]  
**.color** = "colorname" / 0xFFFFFF / [r,g,b]<sup>0..1</sup>  
          = **rgb** (r,g,b)<sup>255</sup> / **hsl** (h<sup>360</sup>,s<sup>100</sup>,l<sup>100</sup>)  
**.count** = count / [count, count]  
**.image** = drawing / "filename" / **image** ("filename")  
**.images** = .count / [count, count]  
**.size** = width / [.width, .height, .depth]  
**.spin** = spinH / [.spinH, .spinV, .spinT, .spinS]  
**.visible** / **.hidden** = true / yes / false / no  
**.wireframe** = true / yes / false / no

SUICA

**background** (color)  
**xyz** (size, color)  
**demo** (distance, altitude, speed)  
**orbit** (distance, altitude, speed)  
**trackball** (distance, altitude)  
**lookAt** (from, to, up)  
**perspective** (near, far, fov)  
**orthographic** (near, far)  
**fullWindow** ()  
**fullScreen** ()  
**stereo** (distance)  
**anaglyph** (distance)  
**vr** ()  
**capture** (filename, time, fps, format, skipframes)  
**SUICA\_VERSION**, **SUICA\_DATE**

MISC

**its**  
**obj.clone**  
**obj.style** ({name: value, ...})  
**allObjects** ()  
**findPosition** (event)  
**findObject** (event, interactive)  
**findObject** (event, [object, ...])  
**findObjects** (event, interactive)  
**findObjects** (event, [object, ...])  
**objectPosition** (local)  
**screenPosition** (local, global)  
**radians** (degrees)  
**degrees** (radians)  
**lerp** (first, second, k)  
**clamp** (value, min, max)  
**random** (from, to)  
**random** ([value,...])  
**randomIn** (object)  
**randomOn** (object)

ADVANCED OBJECTS

**construct** (expression, size, color)  
**convex** (src<sup>2</sup>, size, color)  
**group** (object, object, ...)  
    **.add** (object, object, ...)  
**model** (filename, center, size)  
**model.save** (filename, [object, object, ...])  
**surface** (center, curve<sup>3</sup>, count, size, color)  
**text3d** (text, font<sup>1</sup>, center, size, color)  
**tube** (center, curve<sup>2</sup>, radius, count, size, color)  
**extrude** (shape, center, size, color)  
    **.radius** **.offset** **.count**<sup>1,2</sup>  
**spline** (src<sup>2</sup>, closed, interpolating)  
    (src<sup>5</sup>, param, param)  
**spplane** (src<sup>3</sup>, closed<sup>1,2</sup>, interpolating<sup>1,2</sup>)  
    (src<sup>4</sup>, param, param)

ADVANCED PROPERTIES

**.closed** = bool<sup>1</sup> / [bool, bool]<sup>2</sup>  
**.curve**<sup>2</sup> = [point, ...] / spline / f(u)  
          <sup>3</sup> = [[point, ...], ...] / spline / f(u,v)  
**.expression** = "string"                   A+B, A-B, A\*B, (...)  
**.font<sup>1</sup>** = "fontname.json"  
**.interpolating** = bool<sup>1</sup> / [bool, bool]<sup>2</sup>  
**.src** = [point, ... point]<sup>2</sup> / f(u)<sup>5</sup>  
      = [[point, ...], ... [point, ...]]<sup>3</sup> / f(u,v)<sup>4</sup>  
**.vertices**  
**.threejs** = THREE.Mesh  
    **.material** = THREE.Material  
    **.geometry** = THREE.BufferGeometry  
**.intersectData**  
**.randomIn**  
**.randomOn**

EVENTS

**onPointerEnter**, **onPointerLeave**, **onPointerMove**,  
**onPointerDown**, **onPointerUp**, **onClick**, **onTime**, **onLoad**  
  
obj.**addEventListener** (eventName, eventHandler)  
obj.**removeEventListener** (eventName)  
obj.**eventName** = eventHandler  
  
function pointerEventHandler (event) { ... }  
function timeEventHandler (time, dTime) { ... }  
function loadEventHandler (object) { ... }  
  
**proactive** ()

DRAWINGS & SHAPES

**drawing** (width, height, color)  
**shape** (count)  
**moveTo** (x, y, x, y, ...)  
**lineTo** (x, y, x, y, ...)  
**curveTo** (m<sub>x</sub>, m<sub>y</sub>, x, y)  
**arc** (x, y, radius, from, to, cw)  
**stroke** (color, width, closed)  
**fill** (color)  
**fillText** (x, y, text, color, font)  
                    "bold 20px Courier"  
**clear** (color)

LMS

**scorm**  
**.api** **.score**, **.studentName**, **.getValue** (value) **.setValue** (name, value), **.derandomize** (seed)

shapes

BASIC OBJECTS

<circle center size color ...>  
<cone center size color ...>  
<cube center size color ...>  
<cylinder center size color ...>  
<line center/from to color ...>  
<point center size color ...>  
<polygon count center size color ...>  
<prism count center size color ...>  
<pyramid count center size color ...>  
<sphere center size color ...>  
<square center size color ...>  
    <... id spin image images wireframe>

ADVANCED OBJECTS

<clone src<sup>1</sup> center size color ...>  
<construct expression center size color ...>  
<convex src<sup>2</sup> size color ...>  
<group center size color ...> ... </group>  
<model filename center size ...>  
<surface center curve<sup>3</sup> count size color ...>  
<text3d text font<sup>1</sup> center size color ...>  
<tube center curve<sup>2</sup> radius count size color ...>  
<extrude shape center size color radius offset count<sup>1,2</sup>>  
<splane src<sup>2,5</sup> closed<sup>1</sup> interpolating<sup>1</sup>>  
<splane src<sup>3,4</sup> closed<sup>1,2</sup> interpolating<sup>1,2</sup>>  
    <... id spin image images wireframe>

BASIC PROPERTIES

center = "x, y, z"  
color = "colorname" / "0xFFFFFF", "r,g,b"  
    = "rgb (r,g,b)"<sup>255</sup> / "hsl (h<sup>360</sup>,s<sup>100</sup>,l<sup>100</sup>)"  
count = "count" / "count, count"  
id = "string"  
image = "drawing" / "filename"  
images = "count" / "count, count"  
size = "width" / "width, height, depth"  
spin = "spinH" / "spinH, spinV, spinT, spinS"  
visible / hidden = "bool"  
wireframe = "bool"

ADVANCED PROPERTIES

closed = "bool" / "bool, bool"<sup>2</sup>  
curve<sup>2</sup> = "point; ..." / "spline" / "func(u)"  
    <sup>3</sup> = "point; ... | ..." / "splane" / "func(u,v)"  
expression = "string"      A+B, A-B, A\*B, (...)  
font<sup>1</sup> = "fontname.json"  
interpolating = "bool" / "bool, bool"<sup>2</sup>  
src = "id" / "point; ..." / "func(u)"<sup>5</sup>  
    = "point; ... | ..." / "func(u,v)"<sup>4</sup>  
  
interpolating vs approximating  
closed vs open, cw vs ccw

SUICA

<suica width height background orientation proactive  
    perspective orthographic fullWindow fullScreen stereo  
    anaglyph vr> ... </suica>  
  
<background color>  
<xyz size color>  
<demo distance altitude speed>  
<orbit id distance altitude speed>  
<trackball id distance altitude>  
<lookAt from to up>  
<perspective near far fovy>  
<orthographic near far>  
<fullWindow>  
<fullScreen>  
<stereo distance>  
<anaglyph distance>  
<vr>  
<capture filename time fps format skipframes>

EVENTS

onPointerEnter, onPointerLeave, onPointerMove,  
onPointerDown, onPointerUp, onClick, onTime, onLoad  
  
<tag ... eventName="eventHandler">  
  
<proactive>

DRAWINGS & SHAPES

<drawing size color>  
<shape count>  
<moveTo point>  
<lineTo point>  
<curveTo m point>  
<arc point radius from to cw>  
<stroke color width closed>  
<fill color>  
<fillText point text color font>  
<clear color>  
  
point="x, y" or x="x" y="y"  
font="bold 20px Courier"

shapes