Media

#WWDC14

WebGL

Creating Interactive Content with WebGL

Session 509

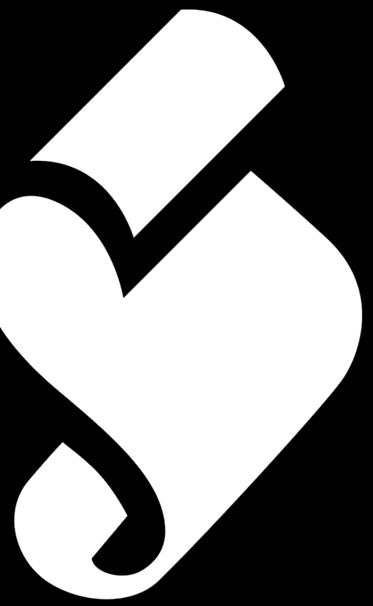
Dean Jackson and Brady Eidson

WebKit Engineers

© 2014 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.


WebGL


OpenGL|ES
TM



JavaScript

MONTAGE STUDIO

2012 Pint

Share

Body Color — Custom $1500

Frame Color — Body Color $1500

Interior — Design Black $0

- [ ] Dashboard Gauges $120
- [ ] Heated Seats $240
- [ ] Luggage Compartment Cover $99
- [ ] Ambient light for interior $190
- [ ] Center console storage $30

Price

Base	$14890
Configured	$17890

MONTAGE STUDIO

2012 Pint

Share

Body Color — Custom $1500

Frame Color — Body Color $1500

Interior — Design Black $0

- [ ] Dashboard Gauges $120
- [ ] Heated Seats $240
- [ ] Luggage Compartment Cover $99
- [ ] Ambient light for interior $190
- [ ] Center console storage $30

Price

Base	$14890
Configured	$17890

Eco Score Average

$ 628,000

EcoHomes™

Welcome

Create a beautiful, high quality, environmentally conscious home using this interactive brochure. Get started by choosing the finishes, options, and appliances best suited to meet your needs.

Photographic reproductions may not be representative of the full range of color, texture, and grain variations which can occur in the product itself.

Crafted with **Montage Studio**

Eco Score Average

$ 628,000

EcoHomes™

Welcome

Create a beautiful, high quality, environmentally conscious home using this interactive brochure. Get started by choosing the finishes, options, and appliances best suited to meet your needs.

Photographic reproductions may not be representative of the full range of color, texture, and grain variations which can occur in the product itself.

Crafted with **Montage Studio**

iSAT Interactive Satellite Viewer

NASA

CANADA

NORTH AMERICA

UNITED STATES

MEXICO

CUBA

Caribbean Sea

VENEZUELA

COLOMBIA

ECUADOR

PERU

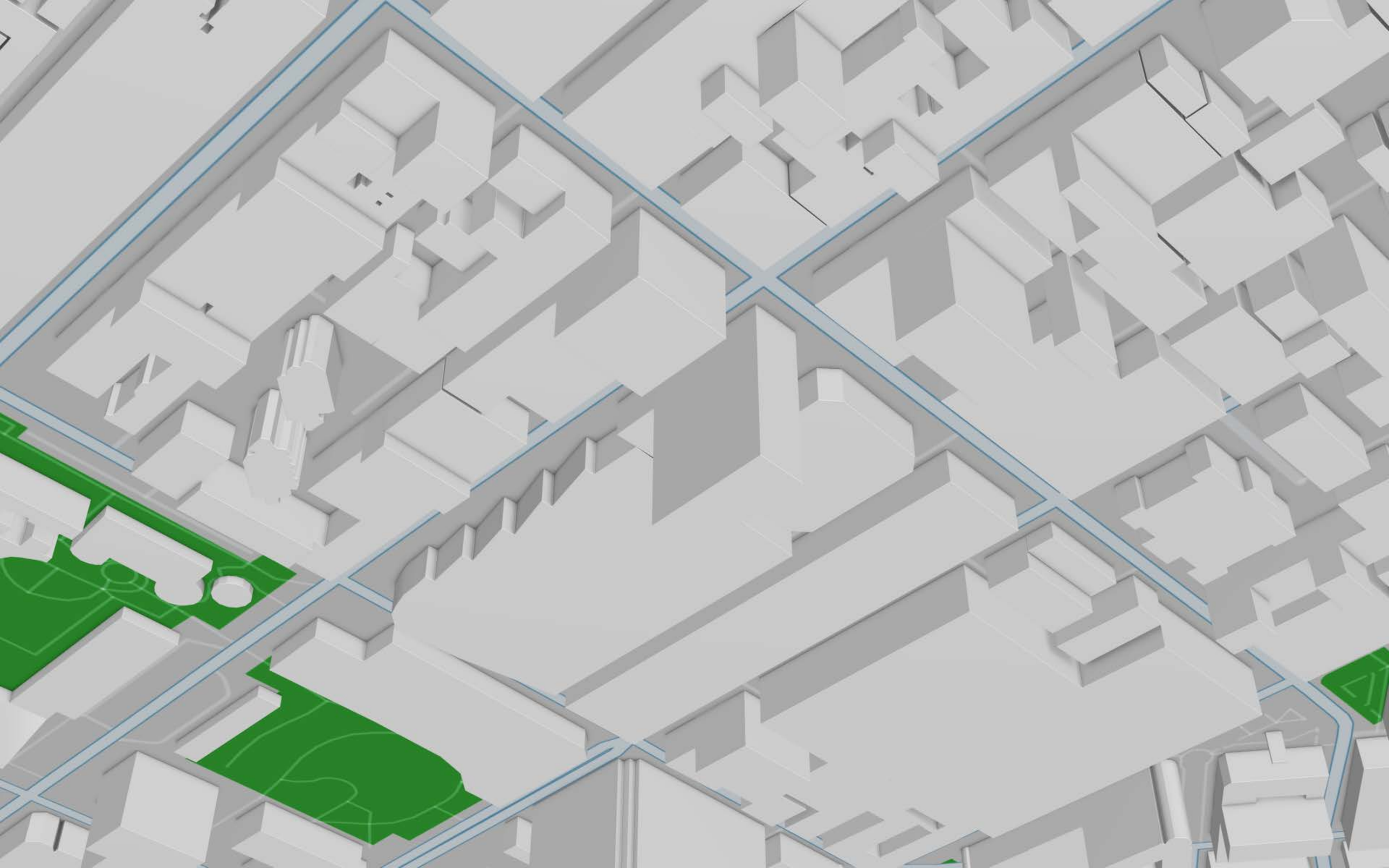


Image: Santa Monica 1

Effect: None

Image: Santa Monica 1

Effect: None


emotion


emotion

AngryBots by Unity

AngryBots by Unity

Swooop by PlayCanvas

Swooop by PlayCanvas

What You Will Learn

What You Will Learn

Setting up WebGL in your page

What You Will Learn

Setting up WebGL in your page

How to do basic drawing

What You Will Learn

Setting up WebGL in your page

How to do basic drawing

Advanced rendering and animation

What You Will Learn

Setting up WebGL in your page

How to do basic drawing

Advanced rendering and animation

Relationship to other HTML features

UNREAL
ENGINE

unity

PLAYCANVAS

three.js

MontageJS

Copperlicht

Babylon JS

Turbulenz

Cesium

Goo Engine

Motivation

Motivation

Powerful graphics in Web content

Motivation

Powerful graphics in Web content

Interoperability through Open Standard

```
  BEGIN { print "Hello, world!" }
```

```
                                                                    (princ "Hello, world!")
```

```
                                        display dialog "Hello, world!"
```

```
                                                                    #import <Foundation/Foundation.h>
                                                                    int main(void)
                                                                    {
                                                                        NSLog(@"Hello, world!\n");
                                                                        return 0;
                                                                    }
```

```
                                    program HelloWorld;
                                    begin
#include <stdio.h>                    WriteLn('Hello, world!');
int main()                          end.
{
    printf("Hello, world!\n");
    return 0;
}
```

```
                                                                    package main
                                                                    import "fmt"
                                                                    func main() {
                                                                       fmt.Println("Hello, world!")
                                                                    }
```

```
                            print "Hello, world!"
```

```
                                                                                PRINT "Hello, world!"
```

```
       DISPLAY 'Hello, world!'.
       STOP RUN.
```

```
                                                                    puts "Hello, world!"
```

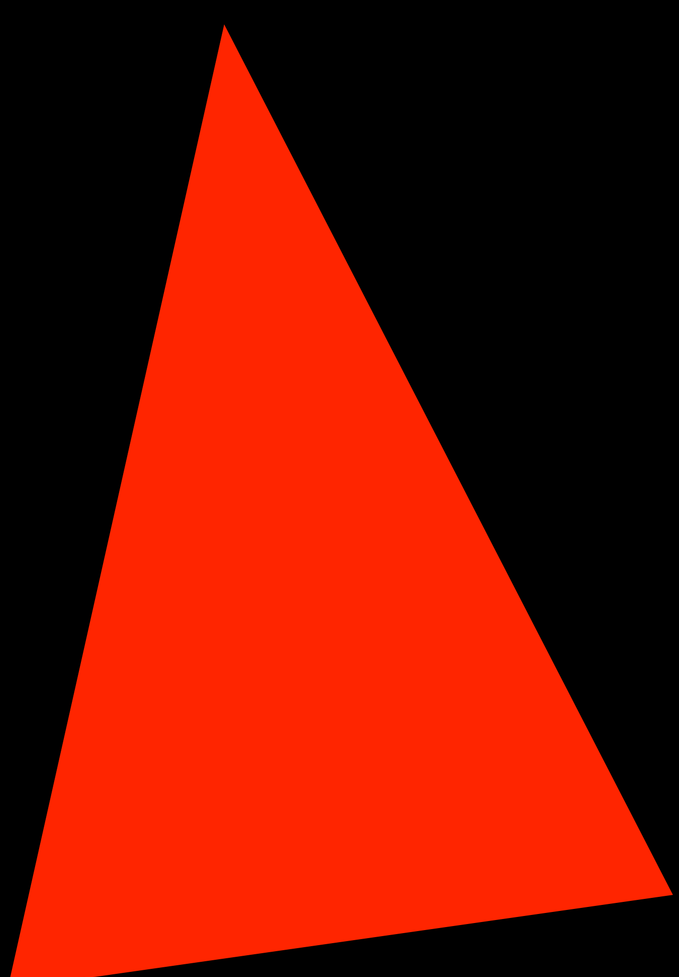
```
                                                                                %!PS
                                                                                /Courier 72 selectfont
                                                                                20 20 moveto
                                                                                (Hello World!) show
                                                                                showpage
```

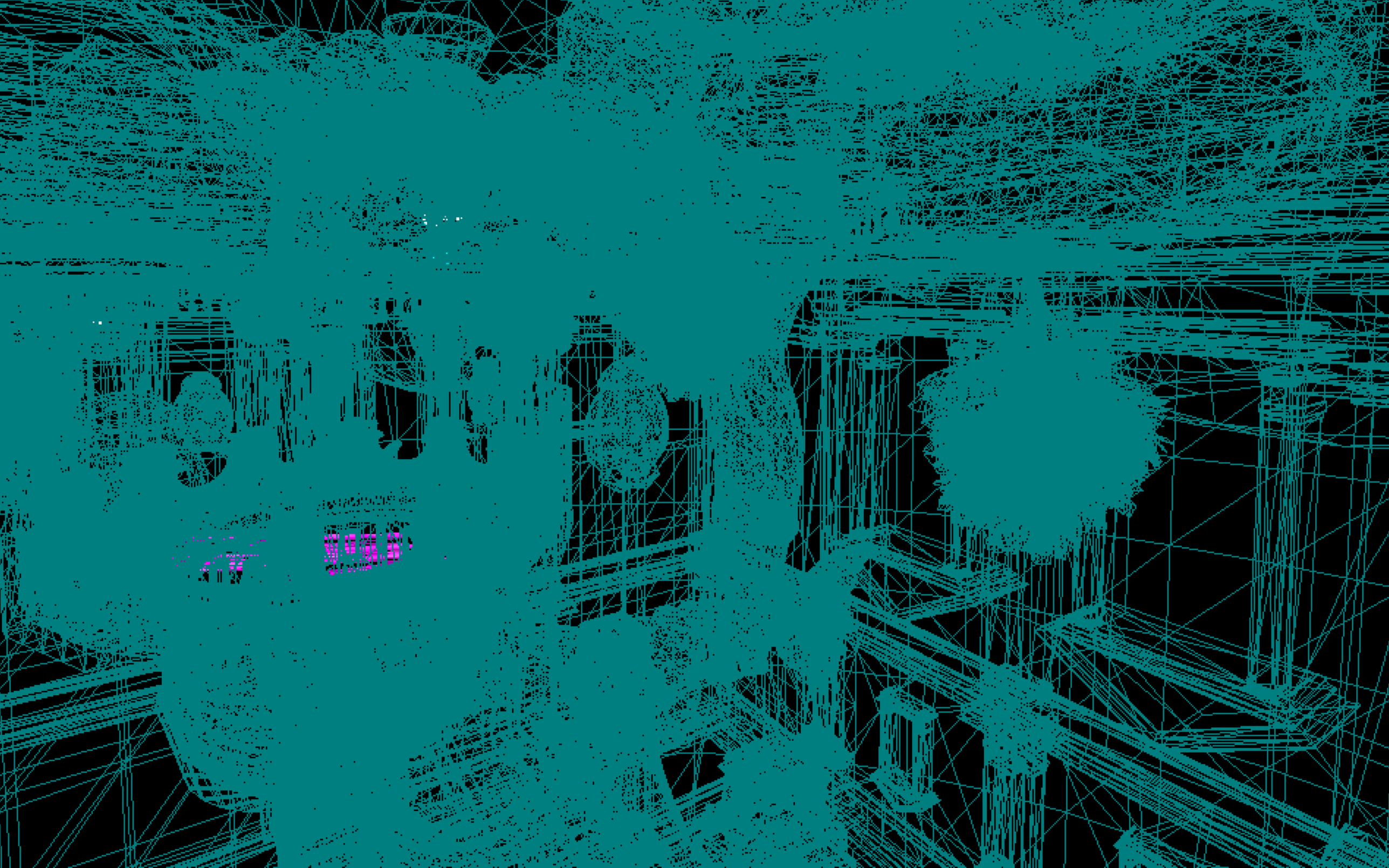
```
                          public class HelloWorld {
                             public static void main(String[] args) {
                                 System.out.println("Hello, world!");
                             }
                          }
```

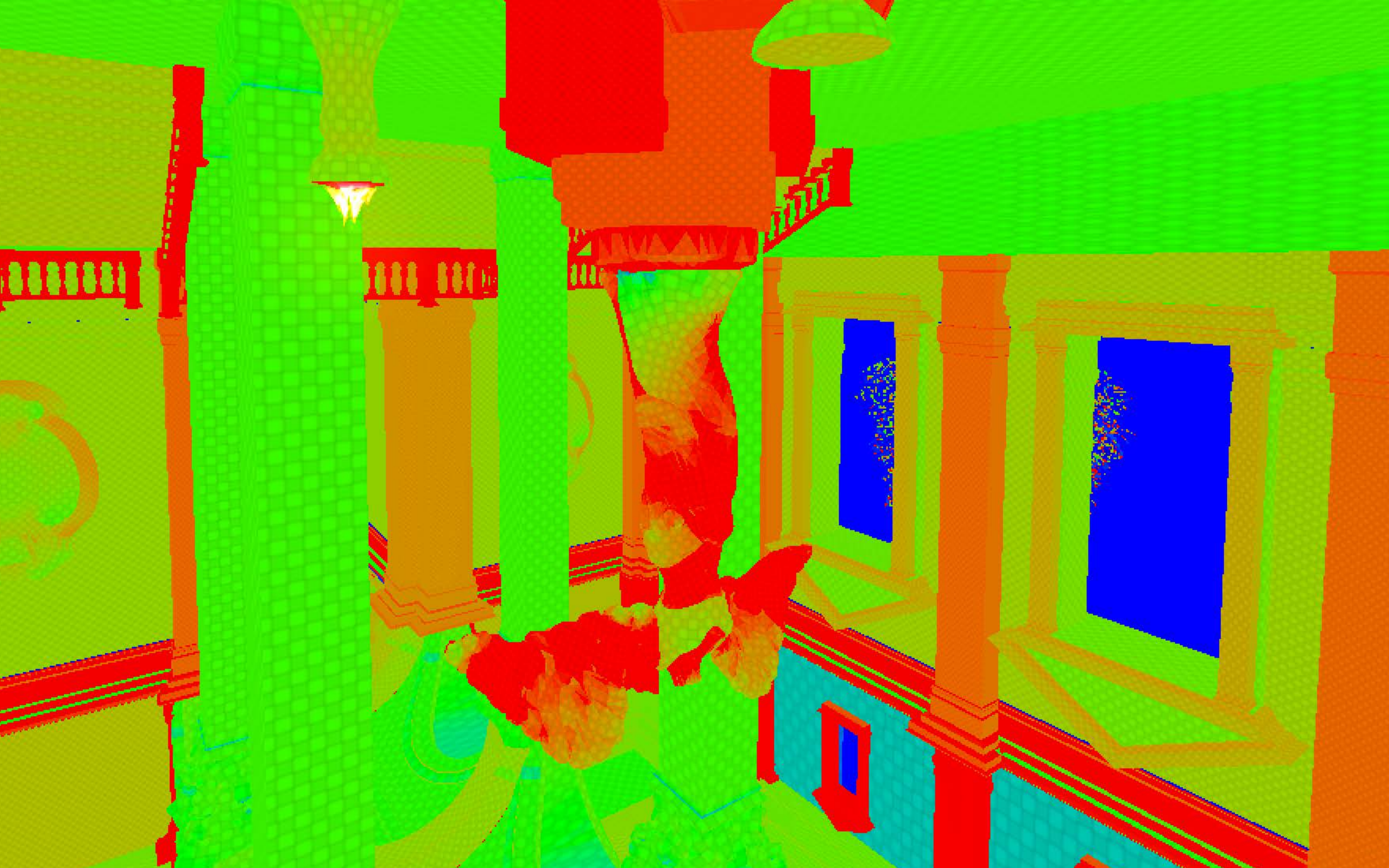
```
HAI
CAN HAS STDIO?
VISIBLE "HAI WORLD!"
KTHXBYE
```

```
                                                                    #include <iostream>
                                                                    int main()
                                                                    {
                                                                        std::cout << "Hello, world!" << std::endl;
                                                                        return 0;
                                                                    }
```

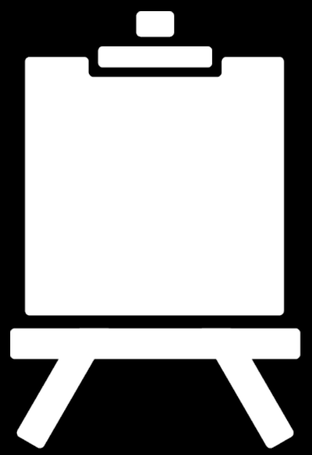
```
                                <?php echo 'Hello, world!'; ?>
```

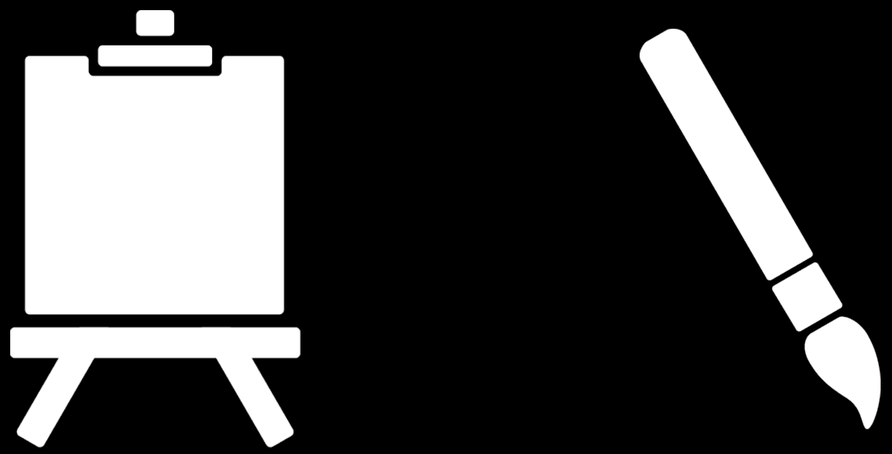


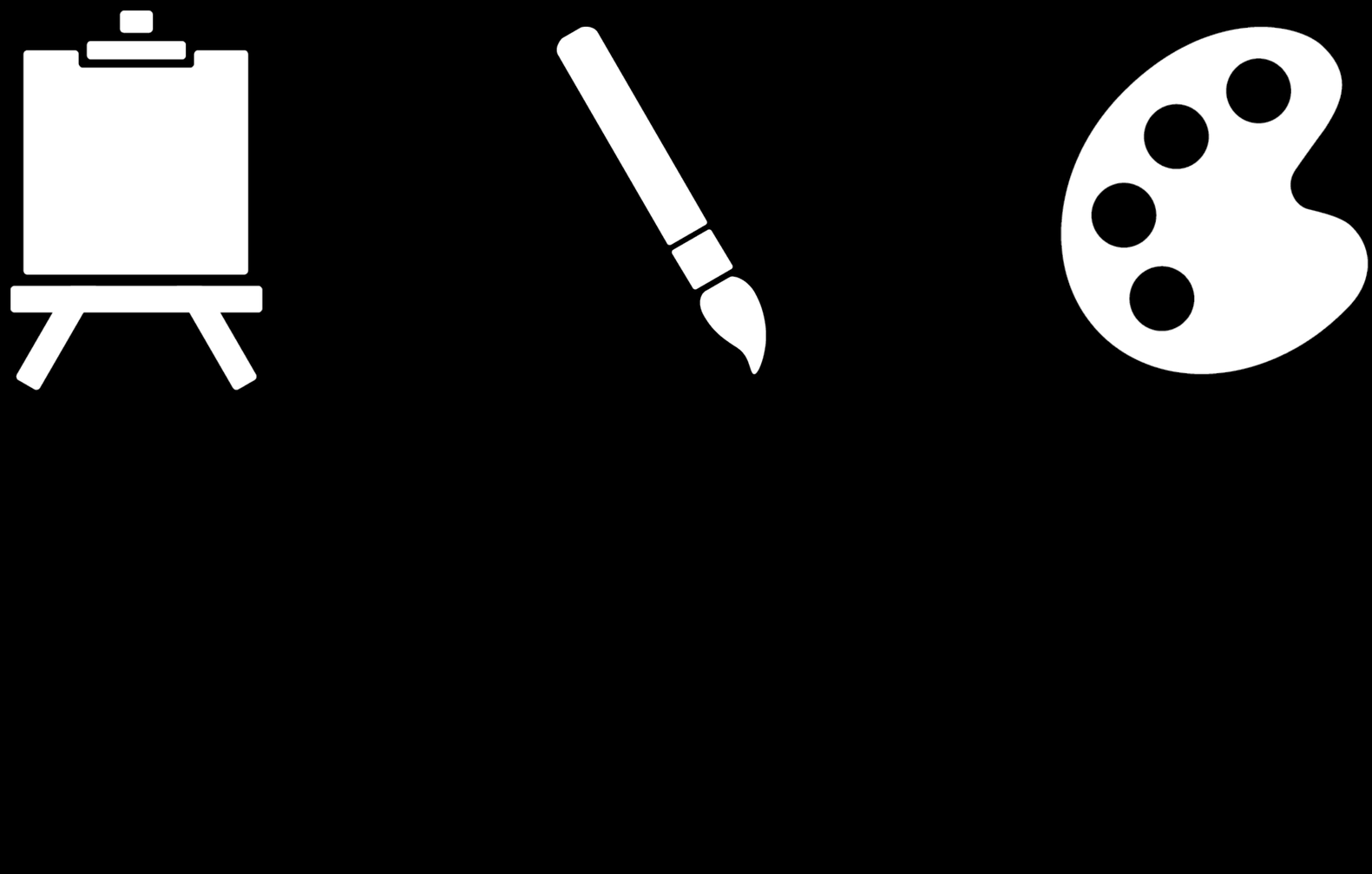


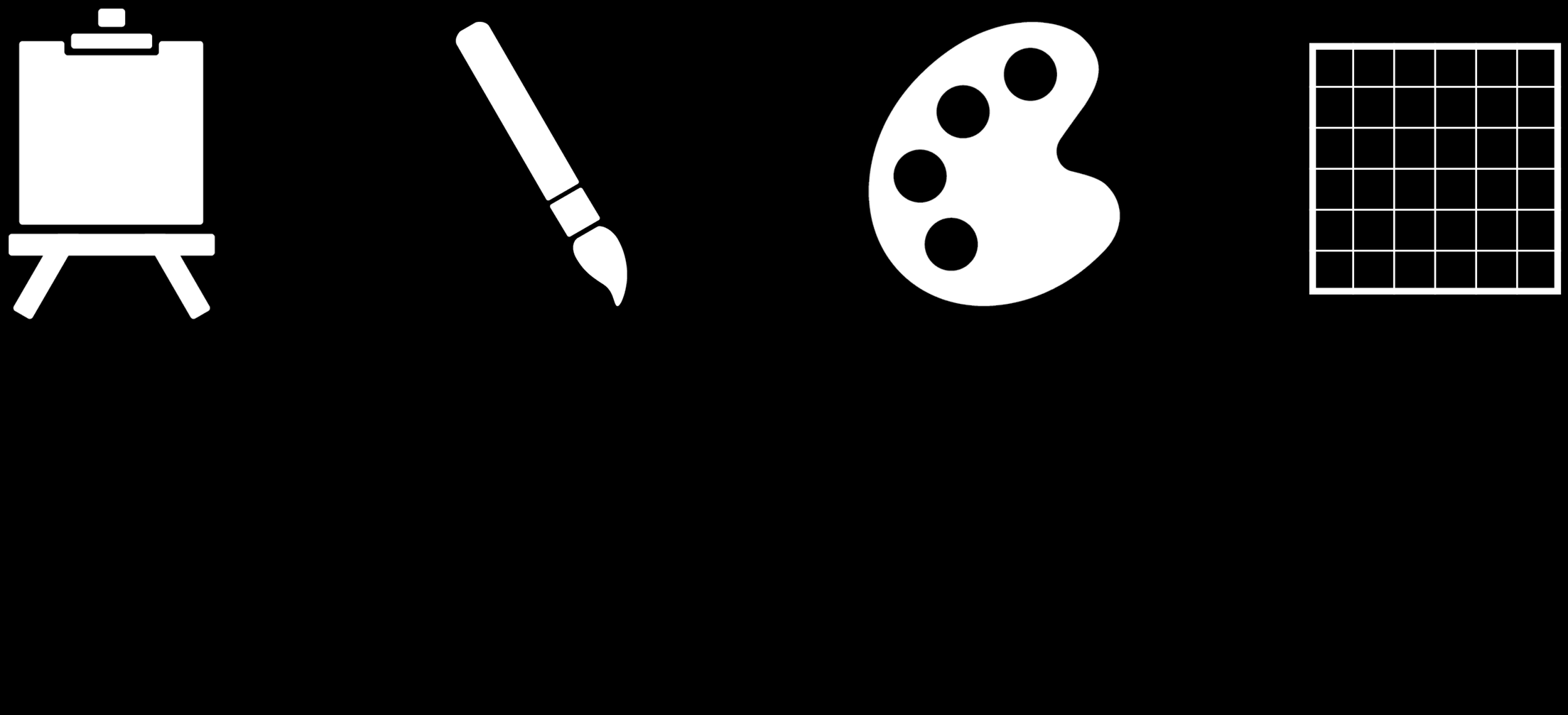


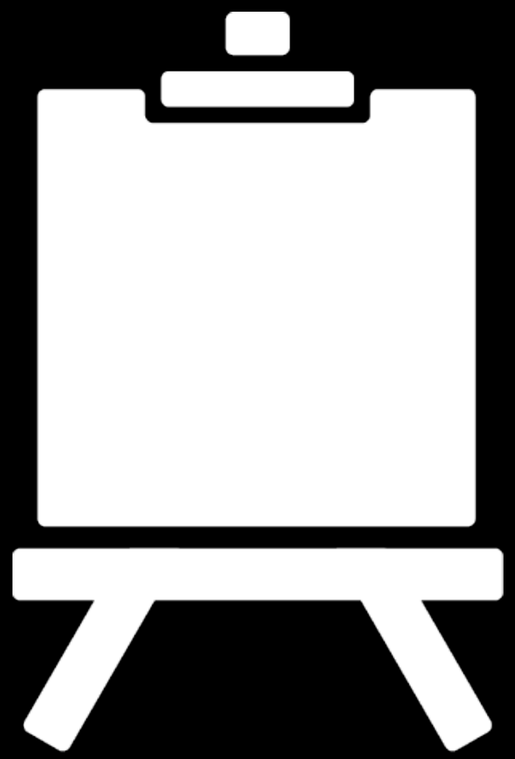
Creating, Configuring, and Drawing











Somewhere to Draw

Canvas

Somewhere to Draw

Canvas

<canvas> element

Or create one via JavaScript:

```
var canvas = document.createElement("canvas");
```

Somewhere to Draw

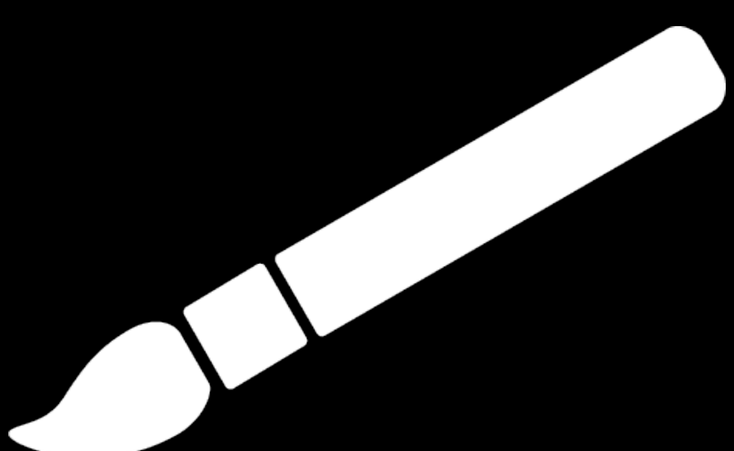
Canvas

```
var canvas = document.querySelector("canvas");
```

Somewhere to Draw

Canvas

```
var canvas = document.querySelector("canvas");

canvas.width = 600 * window.devicePixelRatio;
canvas.height = 400 * window.devicePixelRatio;
```



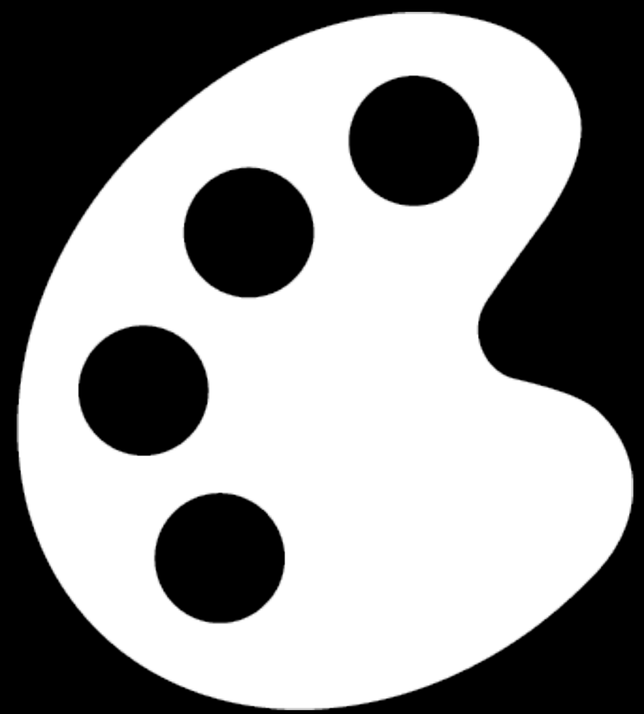
Something to Draw With

Creating the WebGLRenderingContext

Something to Draw With

Creating the WebGLRenderingContext

```
var gl = canvas.getContext("webgl");
```



Configuring the System

Creating and setting up the resources

Configuring the System

Creating and setting up the resources

Before we can do the actual draw operation:

Configuring the System

Creating and setting up the resources

Before we can do the actual draw operation:

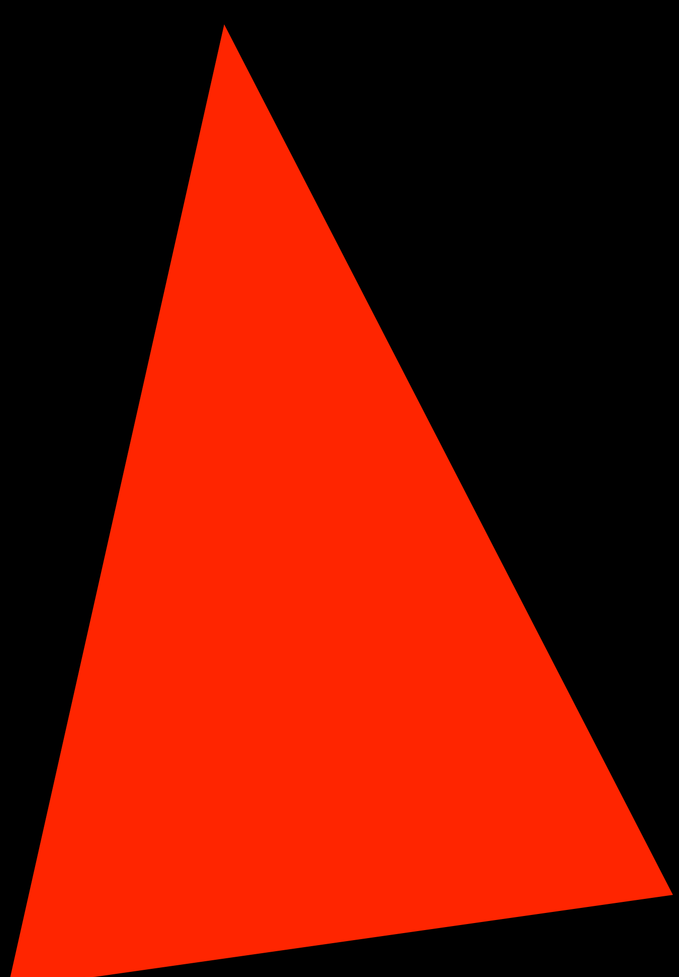
- Create some buffers

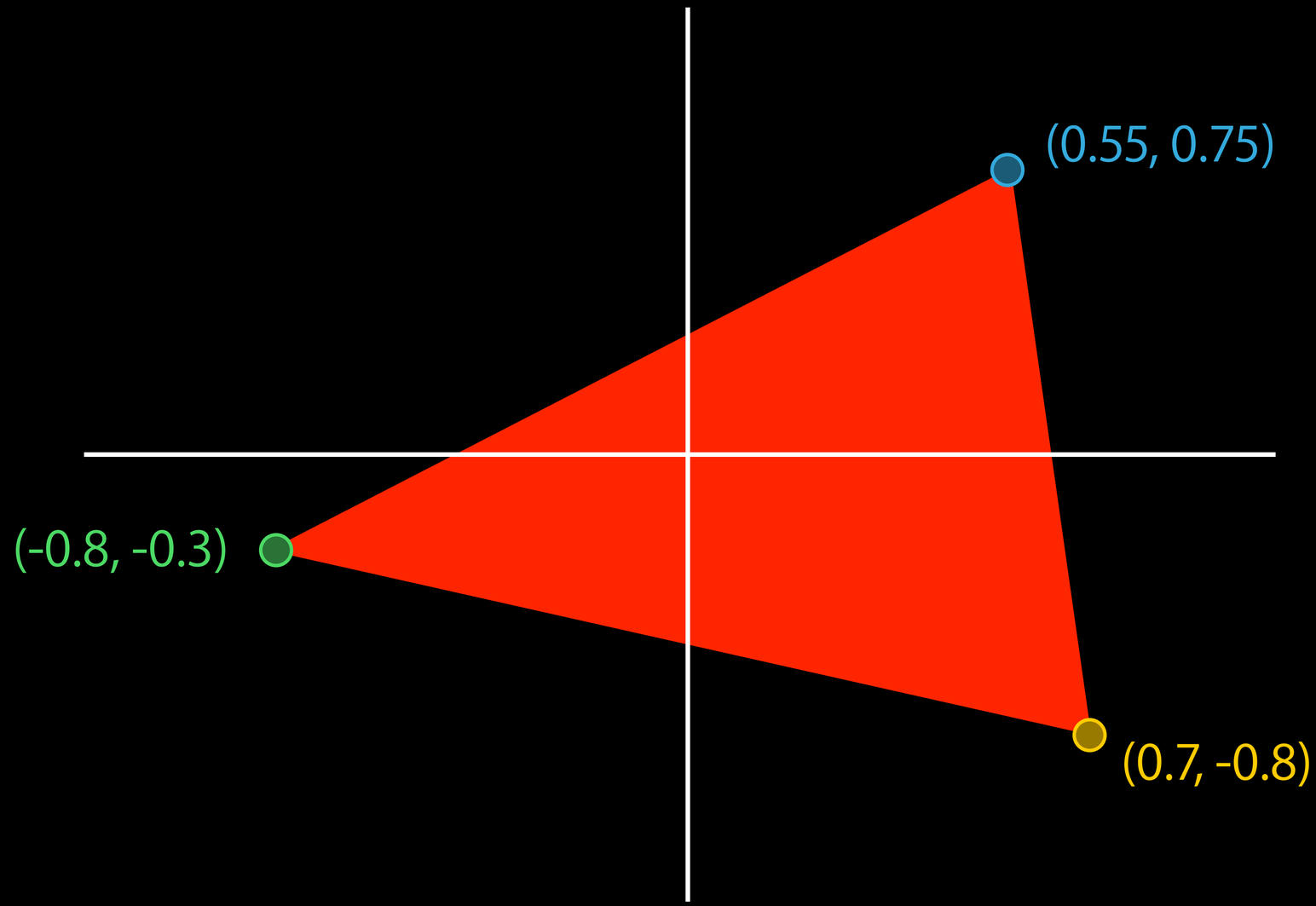
Configuring the System

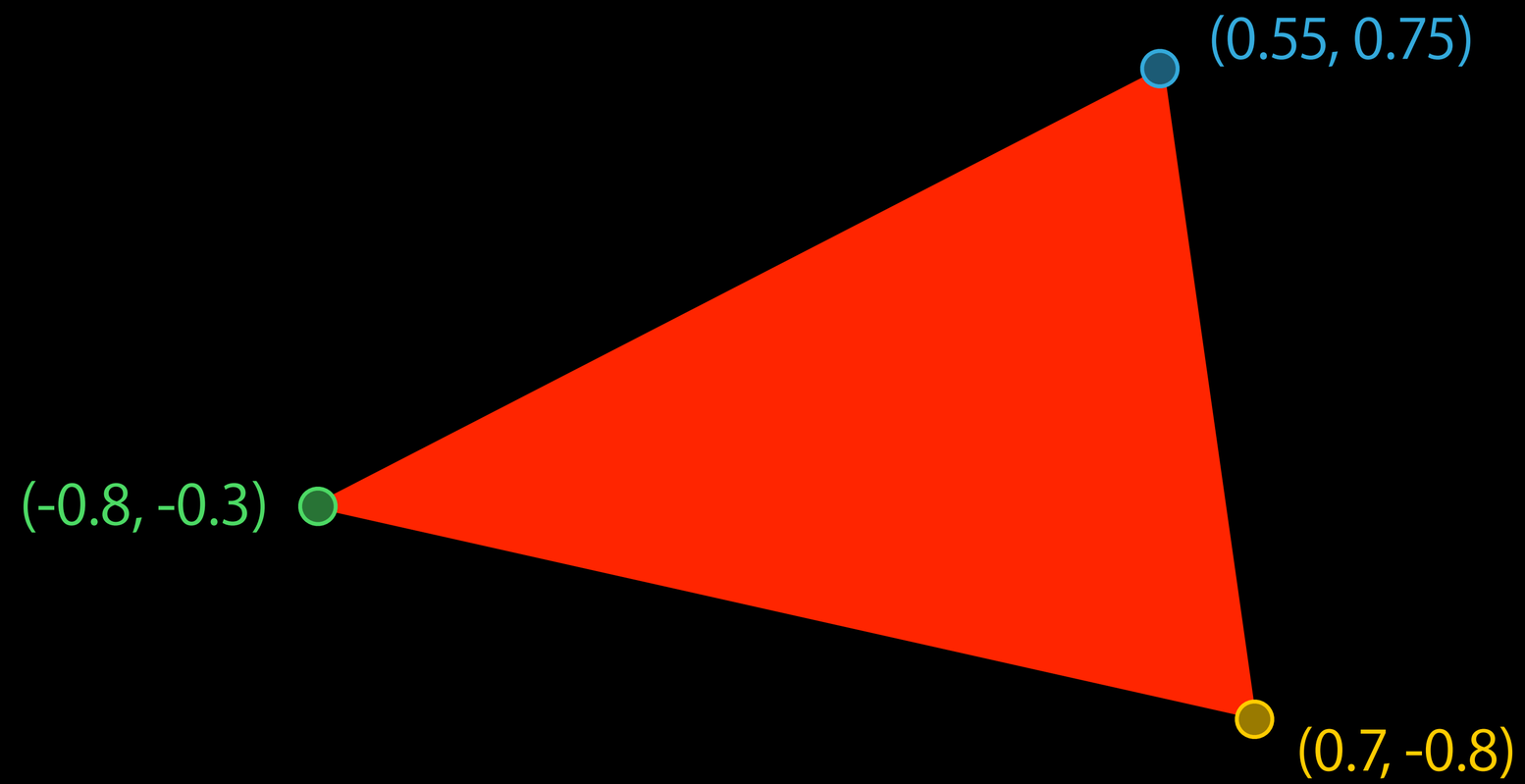
Creating and setting up the resources

Before we can do the actual draw operation:

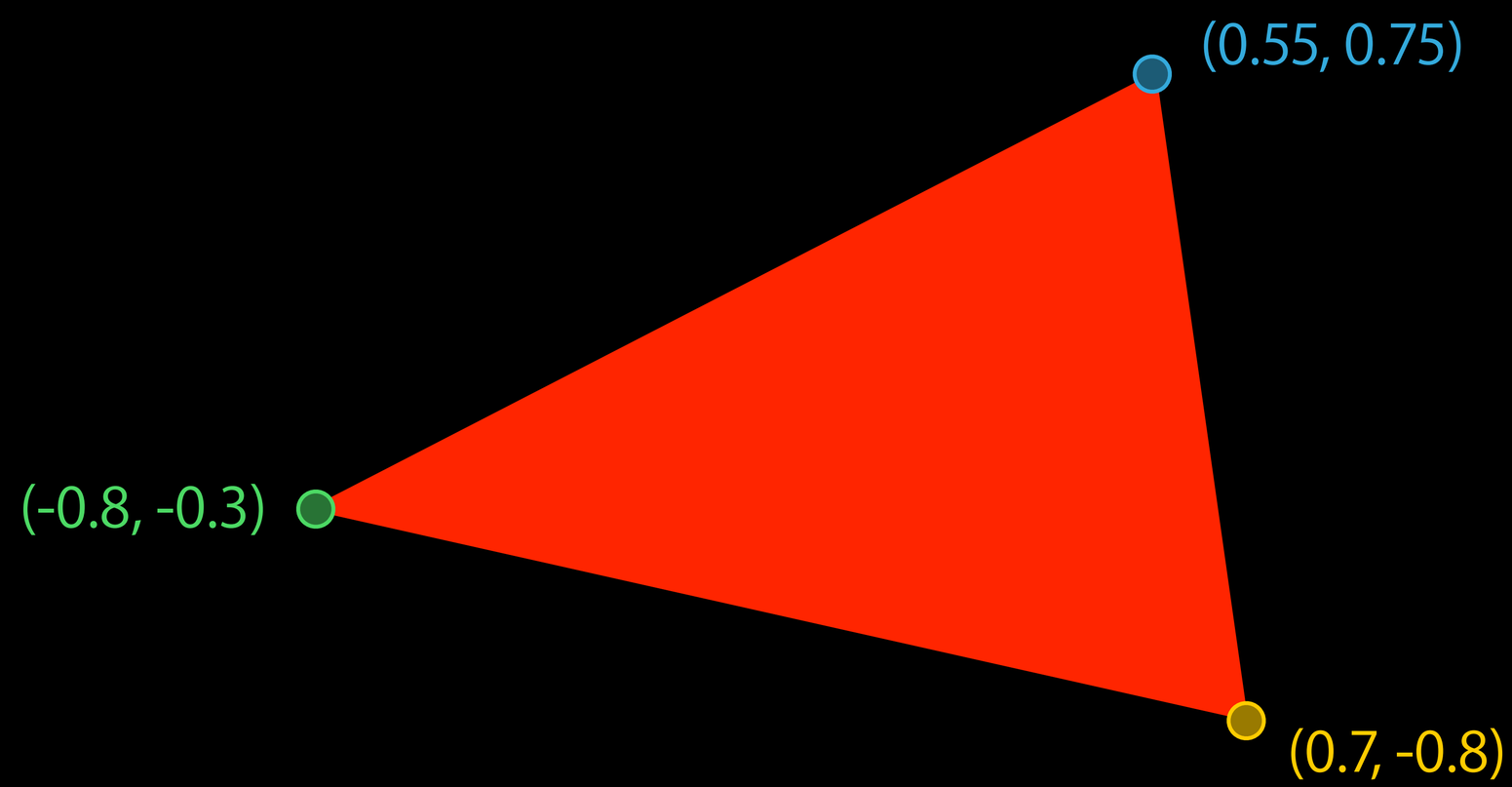
- Create some buffers
- Create a program to use while drawing



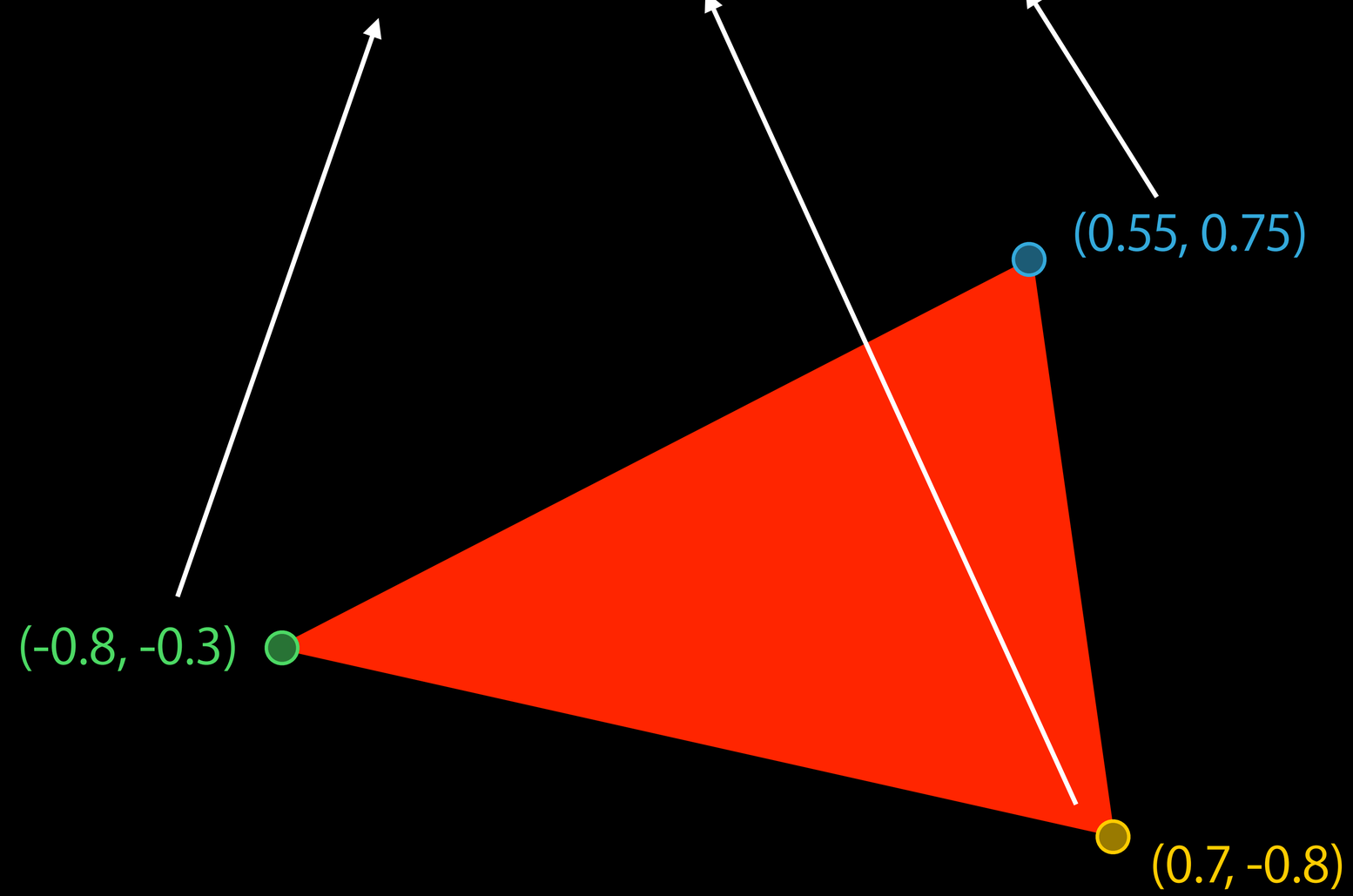

(0.55, 0.75)
(-0.8, -0.3)
(0.7, -0.8)


(0.55, 0.75)
(-0.8, -0.3)
(0.7, -0.8)

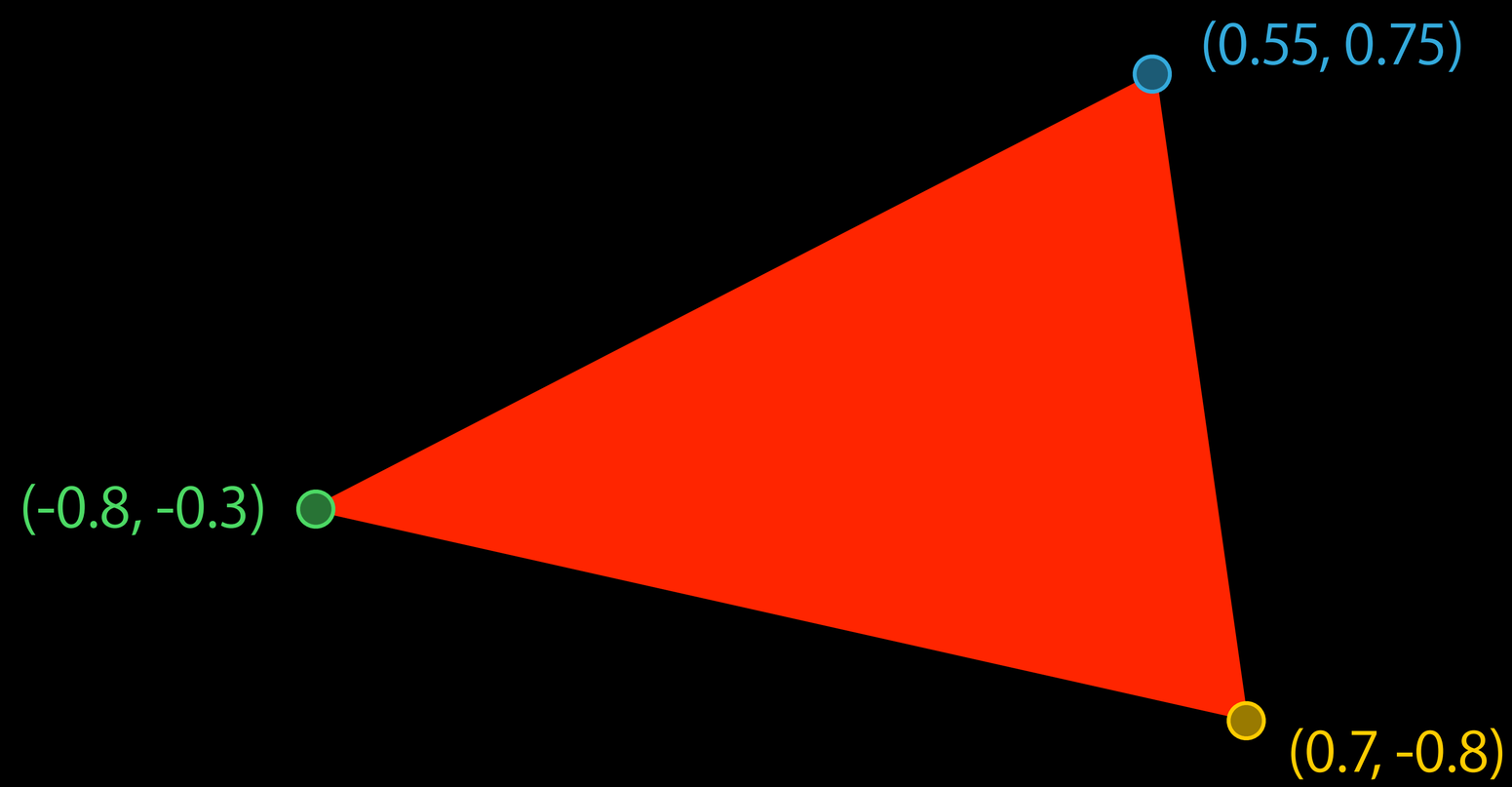
-0.8	-0.3	0.7	-0.8	0.55	0.75



-0.8	-0.3	0.7	-0.8	0.55	0.75



-0.8	-0.3	0.7	-0.8	0.55	0.75



```
var vertices = new Float32Array([
   -0.8, -0.3,
   0.7, -0.8,
   0.55, 0.75
]);
```

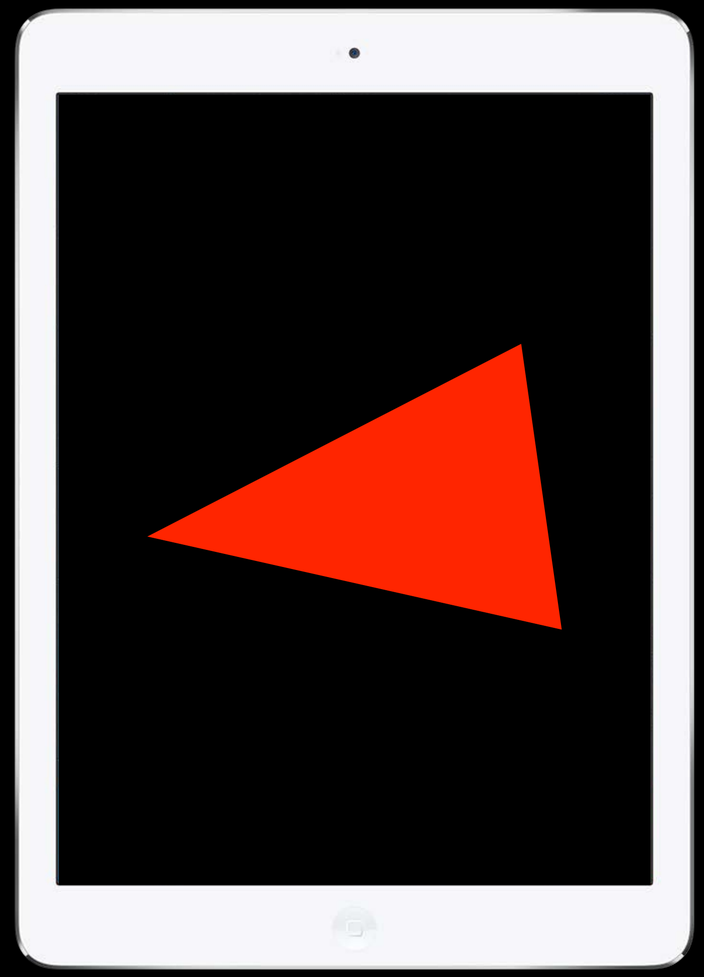
```
var vertices = new Float32Array([
   -0.8, -0.3,
   0.7, -0.8,
   0.55, 0.75
]);
```

```
var vertices = new Float32Array([
   -0.8, -0.3,
   0.7, -0.8,
   0.55, 0.75
]);
```

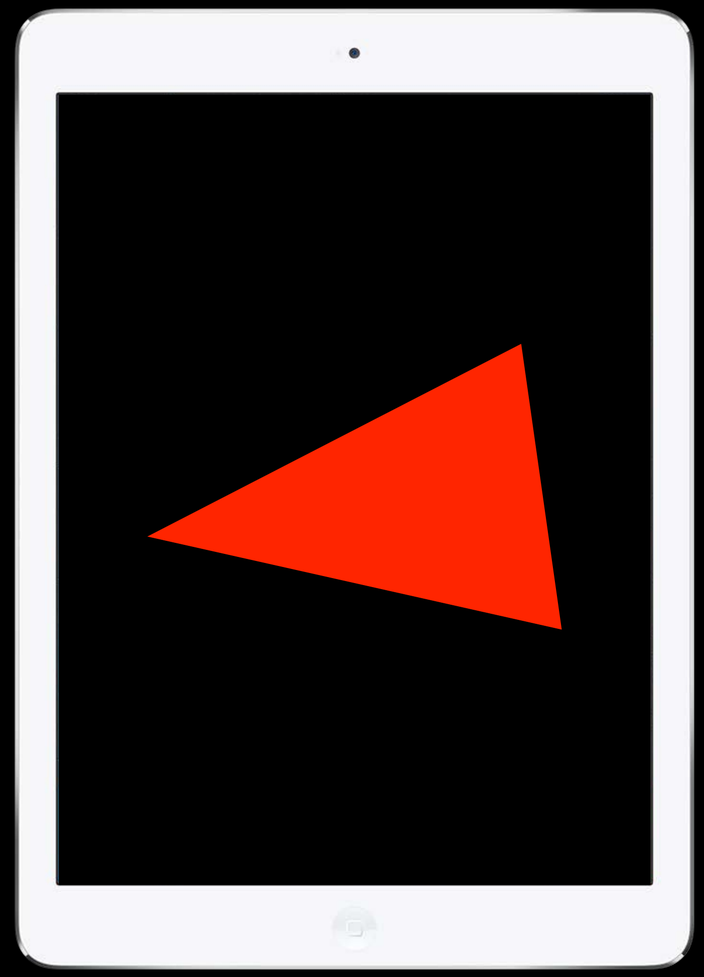
```
var vertices = new Float32Array([
   -0.8, -0.3,
   0.7, -0.8,
   0.55, 0.75
]);

var triangleBuffer = gl.createBuffer();
```

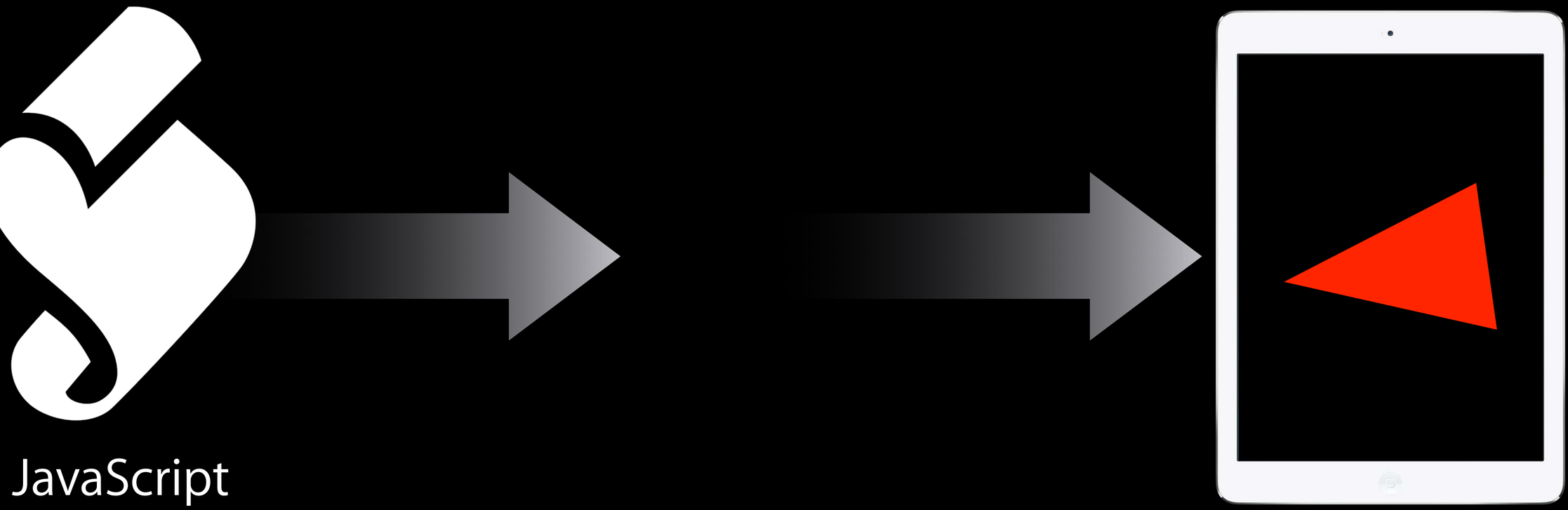
```
var vertices = new Float32Array([
   -0.8, -0.3,
   0.7, -0.8,
   0.55, 0.75
]);

var triangleBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, triangleBuffer);
gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);
```

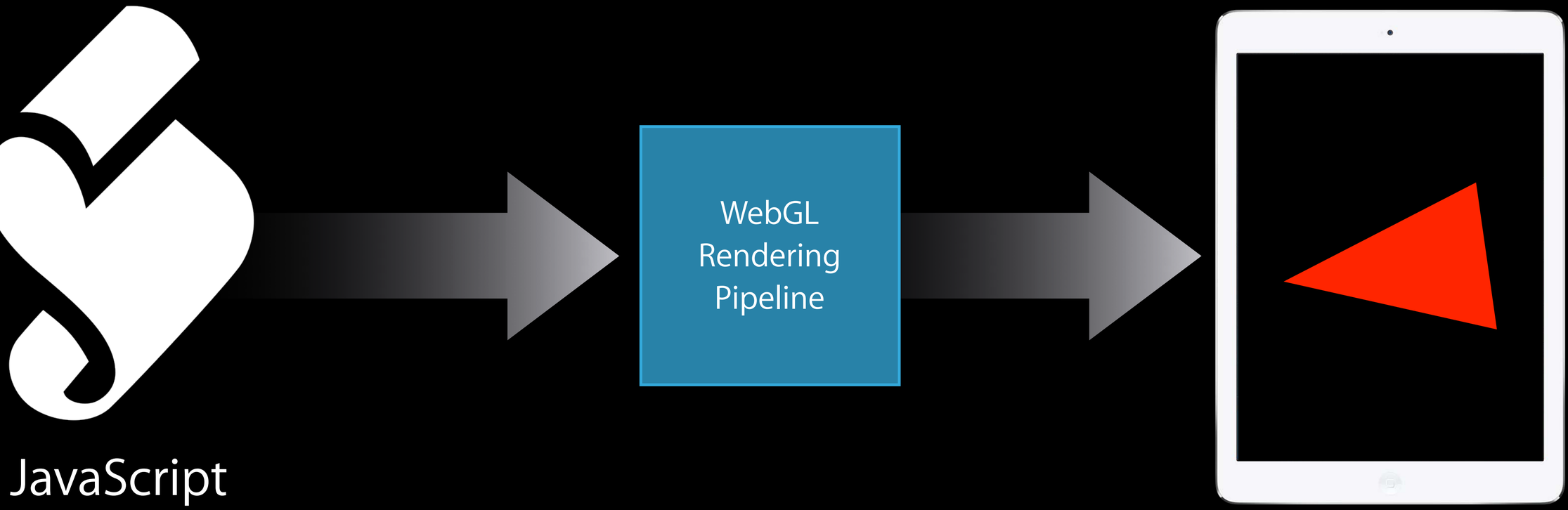
JavaScript



JavaScript

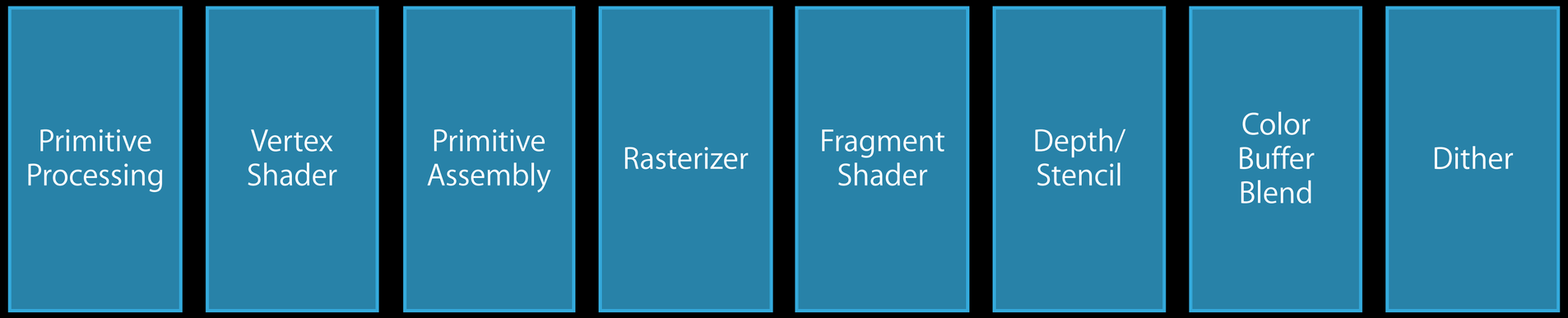



JavaScript


WebGL
Rendering
Pipeline
JavaScript

WebGL
Rendering
Pipeline

WebGL
Rendering
Pipeline

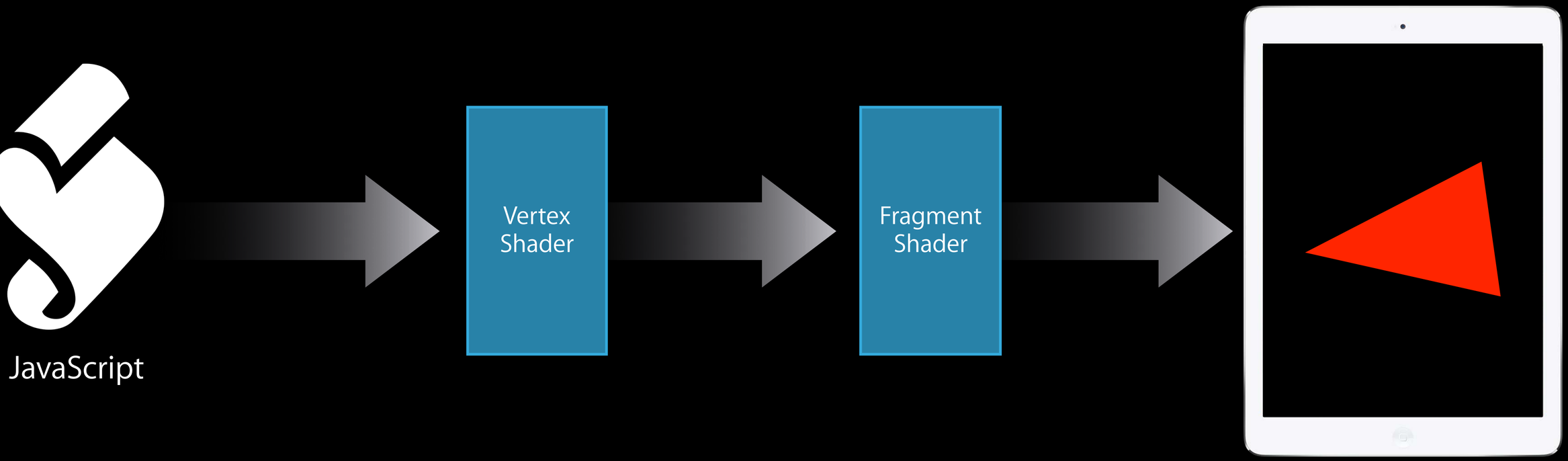

Primitive Processing
Vertex Shader
Primitive Assembly
Rasterizer
Fragment Shader
Depth/ Stencil
Color Buffer Blend
Dither

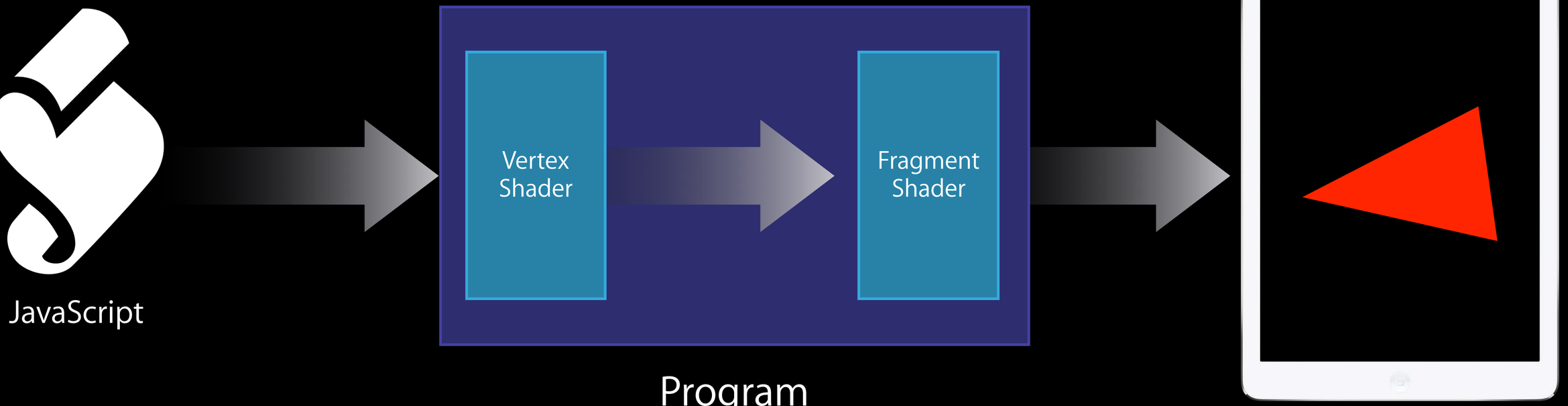

Primitive Processing
Vertex Shader
Primitive Assembly
Rasterizer
Fragment Shader
Depth/ Stencil
Color Buffer Blend
Dither

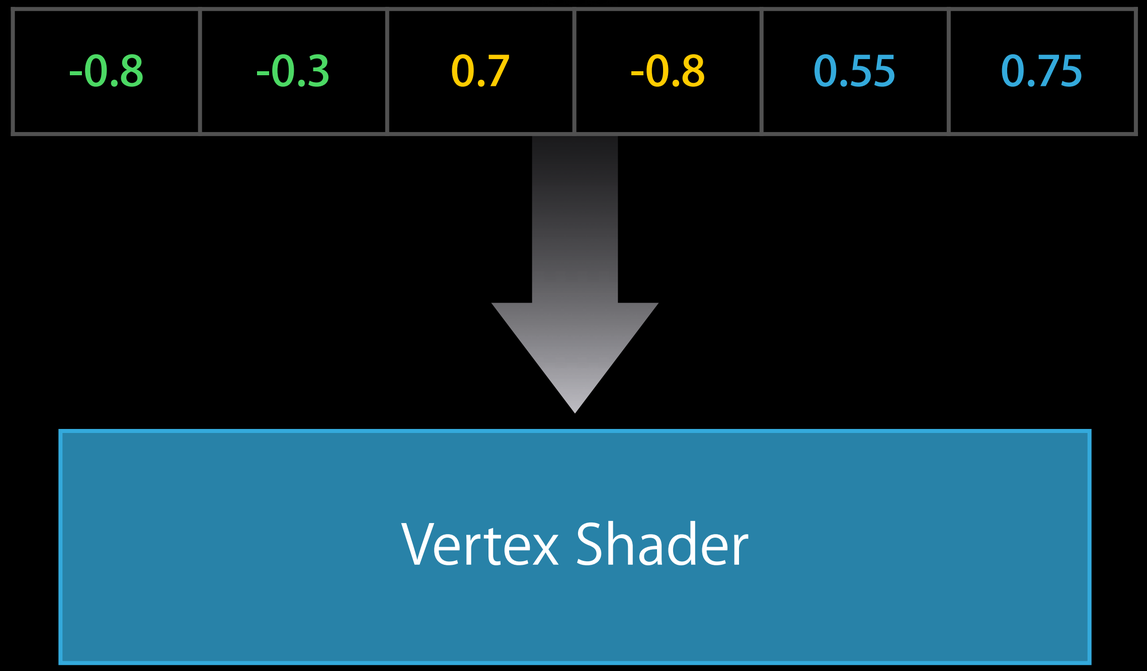
JavaScript

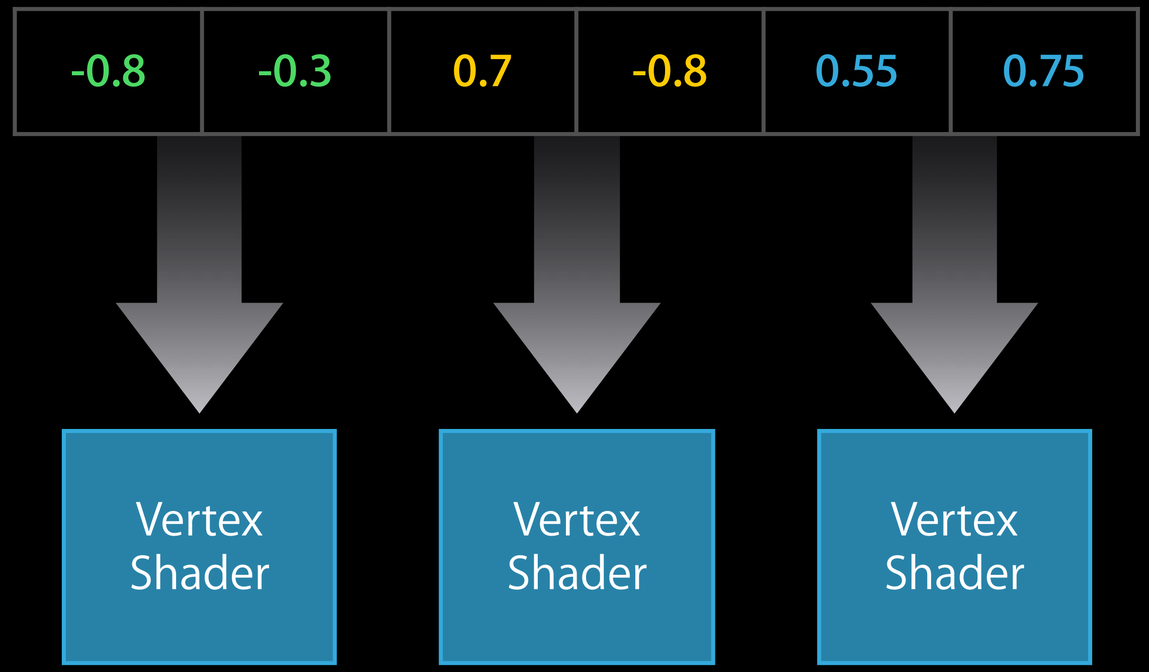
Vertex Shader

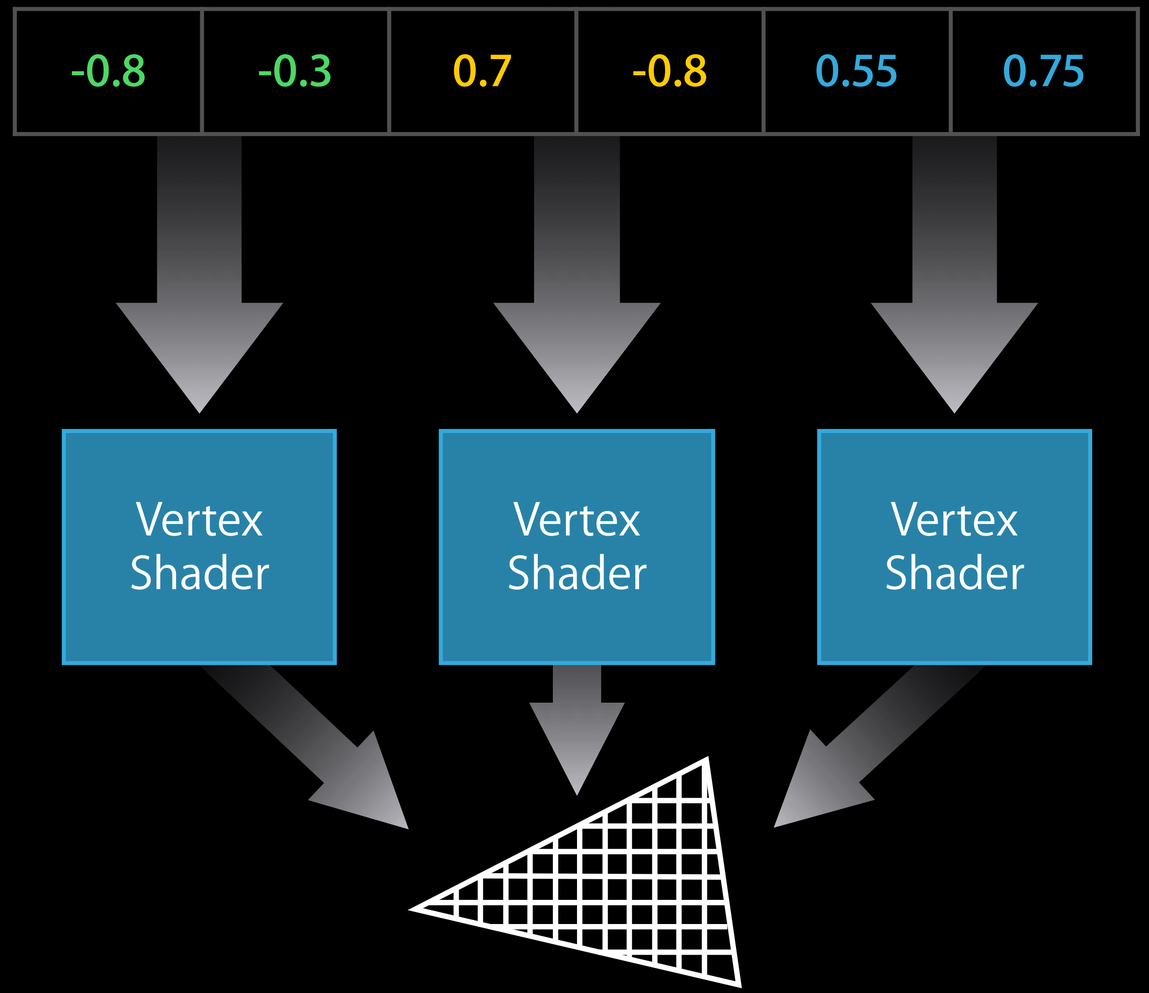
Fragment Shader

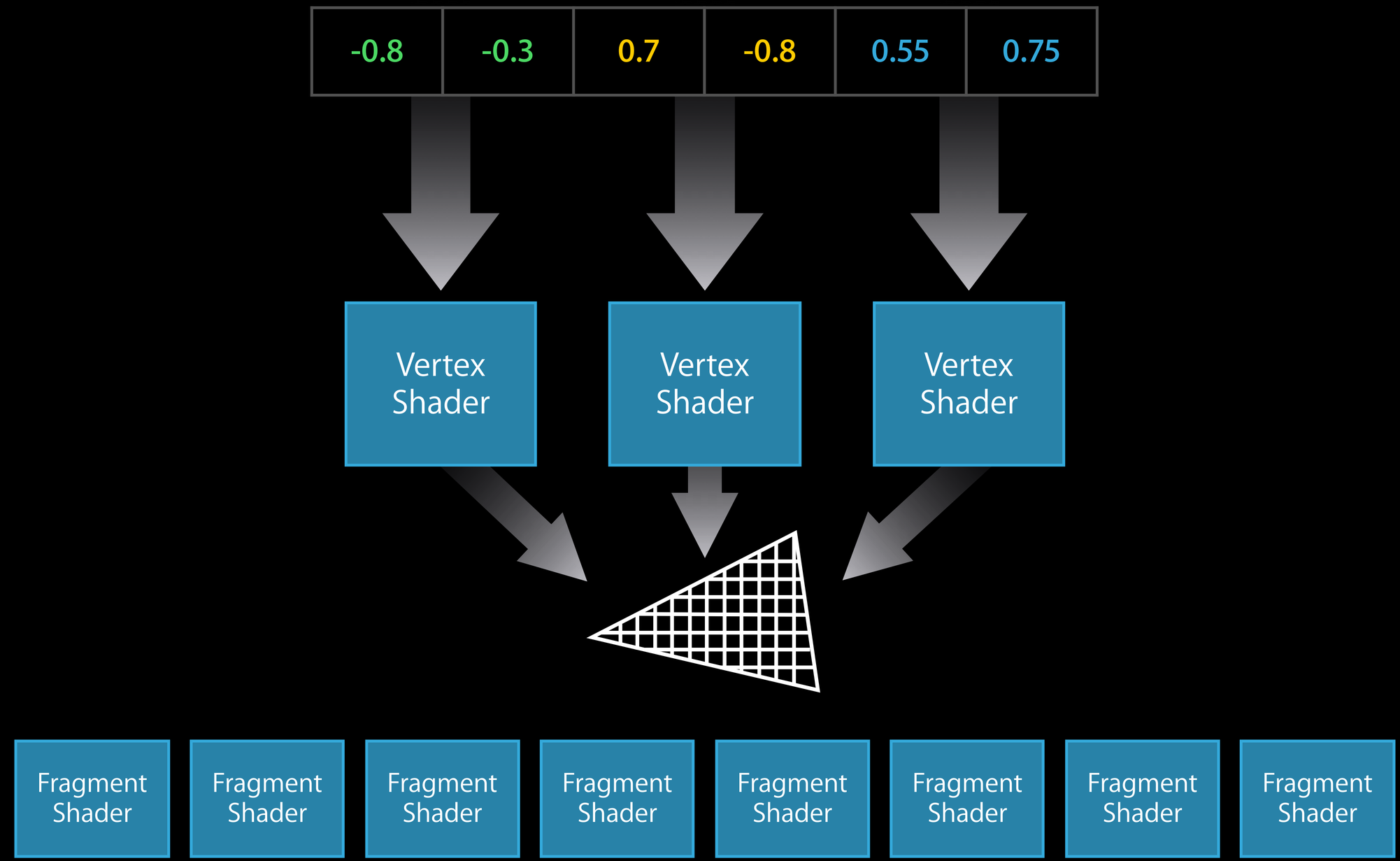

JavaScript
Vertex Shader
Fragment Shader


Vertex
Shader
Fragment
Shader
JavaScript
Program


-0.8
-0.3
0.7
-0.8
0.55
0.75
Vertex Shader


-0.8
-0.3
0.7
-0.8
0.55
0.75
Vertex Shader
Vertex Shader
Vertex Shader


-0.8
-0.3
0.7
-0.8
0.55
0.75
Vertex Shader
Vertex Shader
Vertex Shader


-0.8
-0.3
0.7
-0.8
0.55
0.75
Vertex Shader
Vertex Shader
Vertex Shader
Fragment Shader
Fragment Shader
Fragment Shader
Fragment Shader
Fragment Shader
Fragment Shader
Fragment Shader
Fragment Shader

```
var vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertexShader, "... source code ...");
gl.compileShader(vertexShader);
```

```
var vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertexShader, "... source code ...");
gl.compileShader(vertexShader);
```

```
var vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertexShader, "... source code ...");
gl.compileShader(vertexShader);
```

```
var vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertexShader, "... source code ...");
gl.compileShader(vertexShader);

var fragmentShader = gl.createShader(gl.FRAGMENT_SHADER);
gl.shaderSource(fragmentShader, "... source code ...");
gl.compileShader(fragmentShader);
```

```
var vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertexShader, "... source code ...");
gl.compileShader(vertexShader);

var fragmentShader = gl.createShader(gl.FRAGMENT_SHADER);
gl.shaderSource(fragmentShader, "... source code ...");
gl.compileShader(fragmentShader);
```

```
var program = gl.createProgram();
```

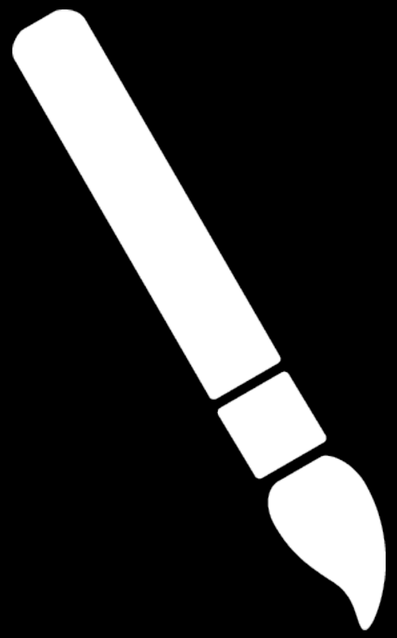
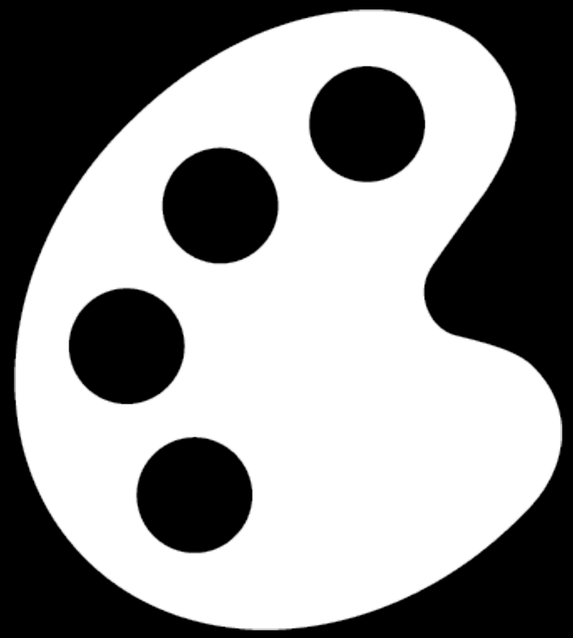
```
var vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertexShader, "... source code ...");
gl.compileShader(vertexShader);

var fragmentShader = gl.createShader(gl.FRAGMENT_SHADER);
gl.shaderSource(fragmentShader, "... source code ...");
gl.compileShader(fragmentShader);
```

```
var program = gl.createProgram();
gl.attachShader(program, vertexShader);
gl.attachShader(program, fragmentShader);
```

```
var vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertexShader, "... source code ...");
gl.compileShader(vertexShader);

var fragmentShader = gl.createShader(gl.FRAGMENT_SHADER);
gl.shaderSource(fragmentShader, "... source code ...");
gl.compileShader(fragmentShader);
```

```
var program = gl.createProgram();
gl.attachShader(program, vertexShader);
gl.attachShader(program, fragmentShader);
gl.linkProgram(program);
```

```
var vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertexShader, "... source code ...");
gl.compileShader(vertexShader);

var fragmentShader = gl.createShader(gl.FRAGMENT_SHADER);
gl.shaderSource(fragmentShader, "... source code ...");
gl.compileShader(fragmentShader);

var program = gl.createProgram();
gl.attachShader(program, vertexShader);
gl.attachShader(program, fragmentShader);
gl.linkProgram(program);

gl.useProgram(program);
```

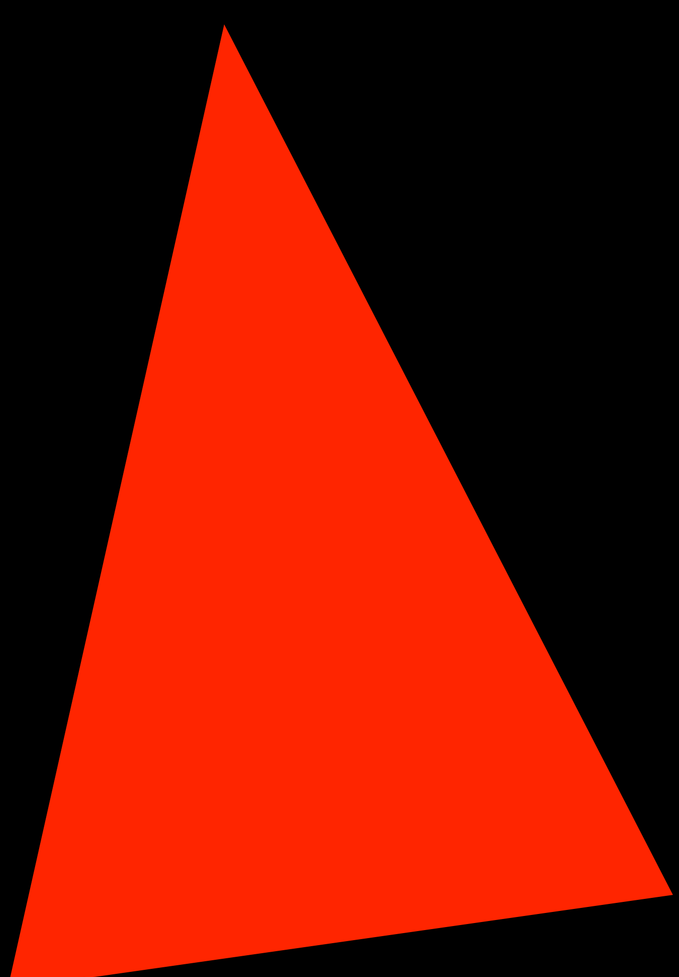




```
var positionAttribute = gl.getAttribLocation(program, "aPosition");
gl.enableVertexAttribArray(positionAttribute);
```

```
var positionAttribute = gl.getAttribLocation(program, "aPosition");
gl.enableVertexAttribArray(positionAttribute);

gl.bindBuffer(gl.ARRAY_BUFFER, triangleBuffer);
gl.vertexAttribPointer(positionAttribute, 2, gl.FLOAT, false, 0, 0);
```

```
var positionAttribute = gl.getAttribLocation(program, "aPosition");
gl.enableVertexAttribArray(positionAttribute);

gl.bindBuffer(gl.ARRAY_BUFFER, triangleBuffer);
gl.vertexAttribPointer(positionAttribute, 2, gl.FLOAT, false, 0, 0);
```

```
var positionAttribute = gl.getAttribLocation(program, "aPosition");
gl.enableVertexAttribArray(positionAttribute);

gl.bindBuffer(gl.ARRAY_BUFFER, triangleBuffer);
gl.vertexAttribPointer(positionAttribute, 2, gl.FLOAT, false, 0, 0);

gl.drawArrays(gl.TRIANGLES, 0, 3);
```



```
<!DOCTYPE html>
<head>
<title>WebGL Triangle</title>
<style>
canvas {
    width: 600px;
    height: 400px;
}
</style>
</head>
<script id="vertexShaderSource" type="text/glsl">
// Simple vertex shader. Simply returns the position it was provided.
attribute vec4 position;
void main() {
  gl_Position = position;
}
</script>
<script id="fragmentShaderSource" type="text/glsl">
// Extremely simple fragment shader. Draws every pixel red.
#ifdef GL_ES
precision mediump float;
#endif

void main() {
  gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
```

```
    gl.bindBuffer(gl.ARRAY_BUFFER, triangleBuffer);
    gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);

    // ---- DRAWING ----

    // Clear to black.
    gl.clearColor(0, 0, 0, 1);
    gl.clear(gl.COLOR_BUFFER_BIT);

    // Bind the vertex attributes for the draw operation. We first make
    // sure we're talking to the correct buffer, then we're going to
    // associate that buffer with the vertex attribute from the program.
    // Our buffer is an array of (x,y) points, so 2 floating point
values
    // per vertex.
    gl.bindBuffer(gl.ARRAY_BUFFER, triangleBuffer);
    gl.vertexAttribPointer(positionAttribute, 2, gl.FLOAT, false, 0, 0);

    // The actual draw call. We know there are 3 points in the buffer
    // (a triangle).
    gl.drawArrays(gl.TRIANGLES, 0, 3);
}

window.addEventListener("load", drawTriangle, false);
</script>
<body>
```

Shaders

GLSL

GLSL

C-like language designed for parallel graphics

GLSL

C-like language designed for parallel graphics

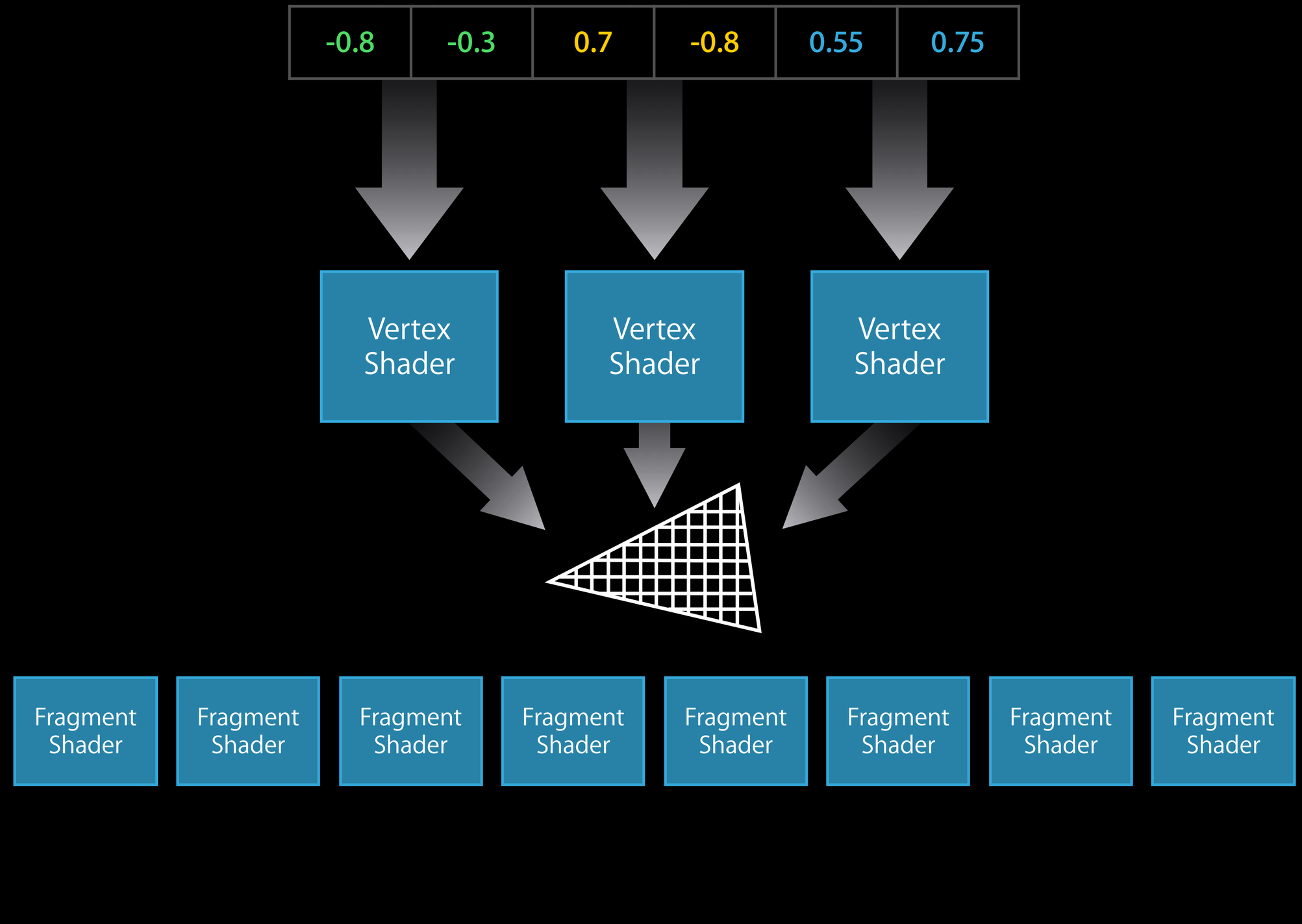
Primitives: Vectors, matrices

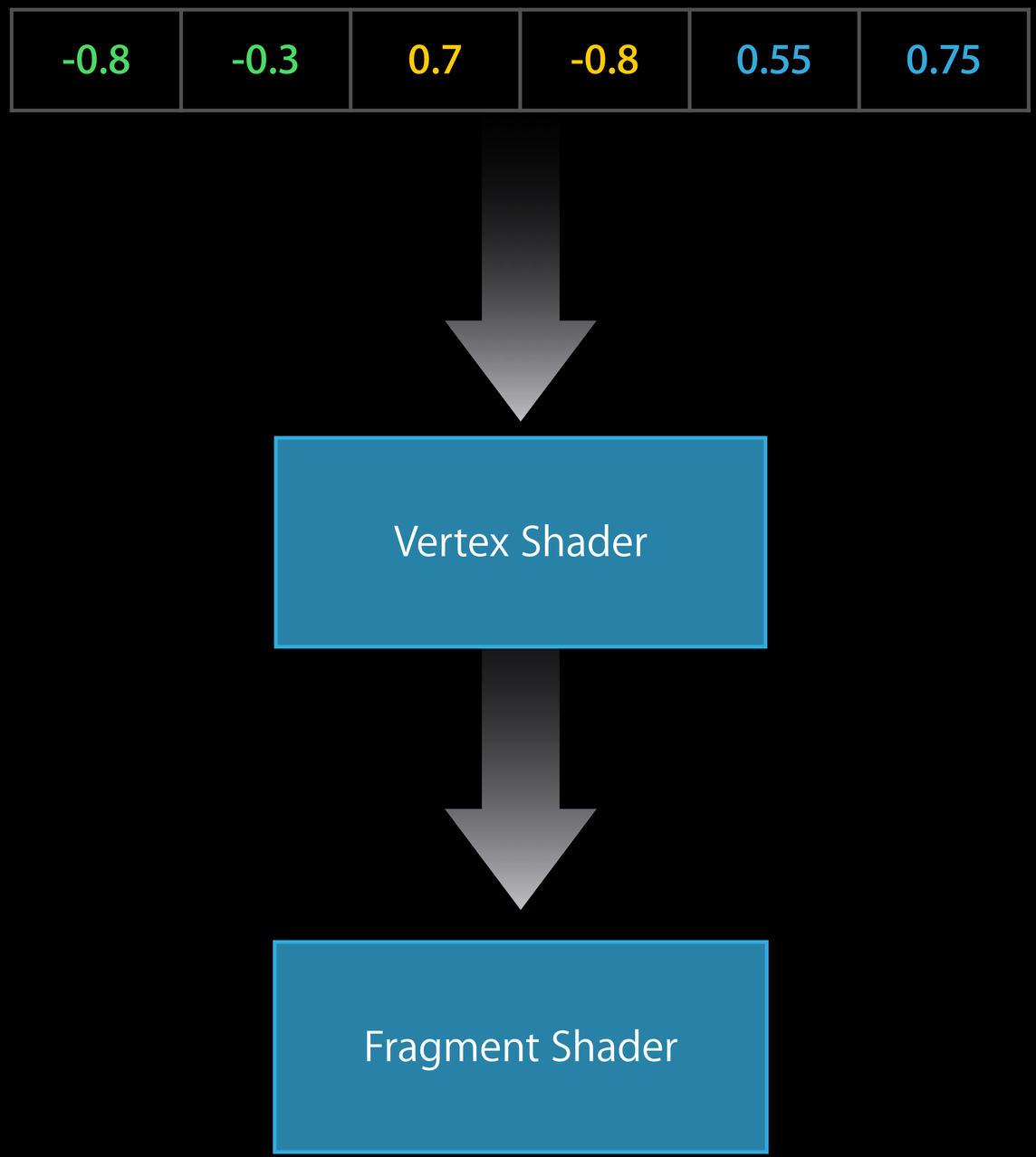
GLSL

C-like language designed for parallel graphics

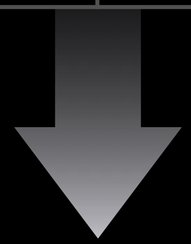
Primitives: Vectors, matrices

Built-in functions: Trigonometry, vector math, smoothing, and clamping

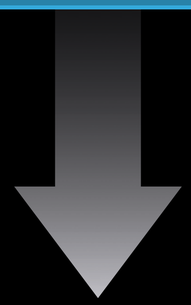

-0.8
-0.3
0.7
-0.8
0.55
0.75
Vertex Shader
Vertex Shader
Vertex Shader
Fragment Shader
Fragment Shader
Fragment Shader
Fragment Shader
Fragment Shader
Fragment Shader
Fragment Shader
Fragment Shader


-0.8
-0.3
0.7
-0.8
0.55
0.75
Vertex Shader
Fragment Shader

-0.8	-0.3	0.7	-0.8	0.55	0.75
1	100	2	987	3	46

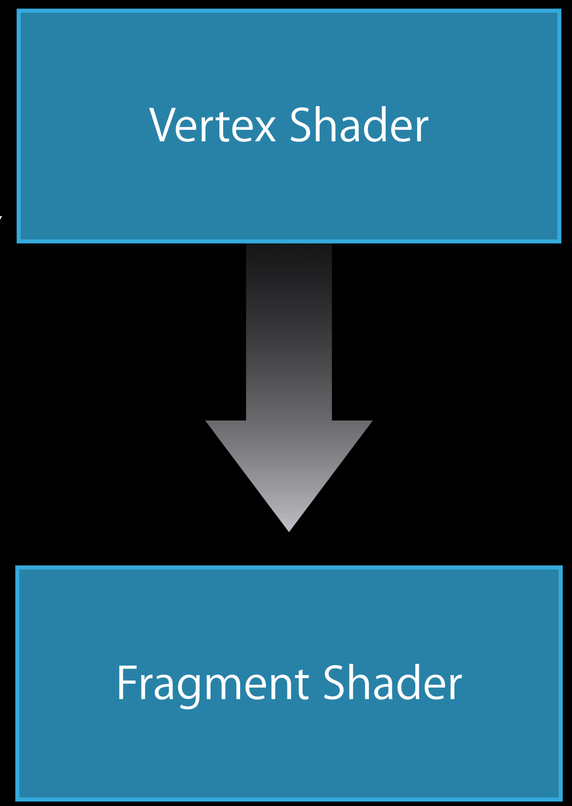


Vertex Shader



Fragment Shader

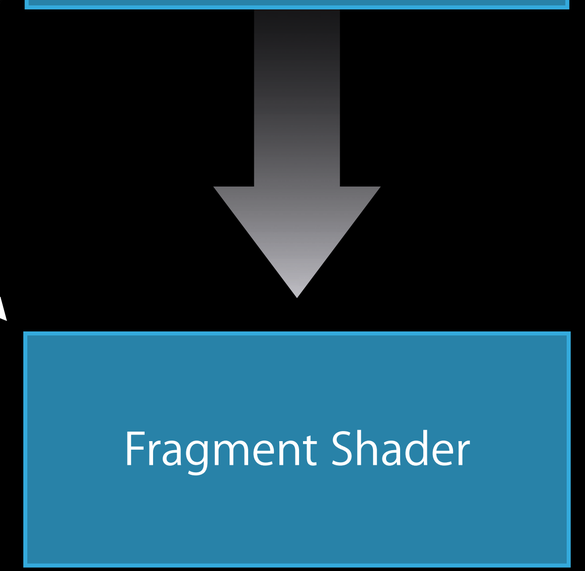
-0.8	-0.3	0.7	-0.8	0.55	0.75
1	100	2	987	3	46



Global Constants
(uniforms)

-0.8	-0.3	0.7	-0.8	0.55	0.75
1	100	2	987	3	46

Global Constants
(uniforms)

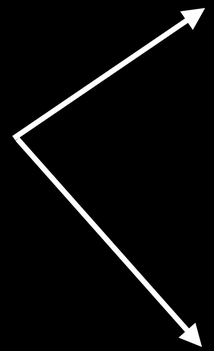


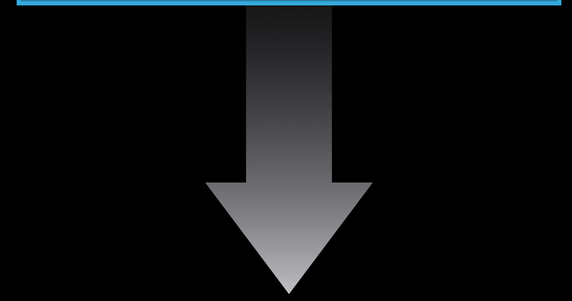
-0.8	-0.3	0.7	-0.8	0.55	0.75
1	100	2	987	3	46

Vertex Shader

`gl_Position`

`Global Constants`
`(uniforms)`





Fragment Shader

`gl_FragColor`

Vertex Shader Source Code

```
attribute vec4 aPosition;

void main() {
  gl_Position = aPosition;
}
```

Vertex Shader Source Code

```
attribute vec4 aPosition;

void main() {
  gl_Position = aPosition;
}
```

Vertex Shader Source Code

```
attribute vec4 aPosition;

void main() {
  gl_Position = aPosition;
}
```

Fragment Shader Source Code

```
precision mediump float;

void main() {
  gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```

Fragment Shader Source Code

```
precision mediump float;

void main() {
  gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```

Fragment Shader Source Code

```
precision mediump float;

void main() {
  gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```

Fragment Shader Source Code

```
precision mediump float;

void main() {
  gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```

red green blue alpha

Demo

Live Shader Editor

Shaders

C-like programs that run on the GPU

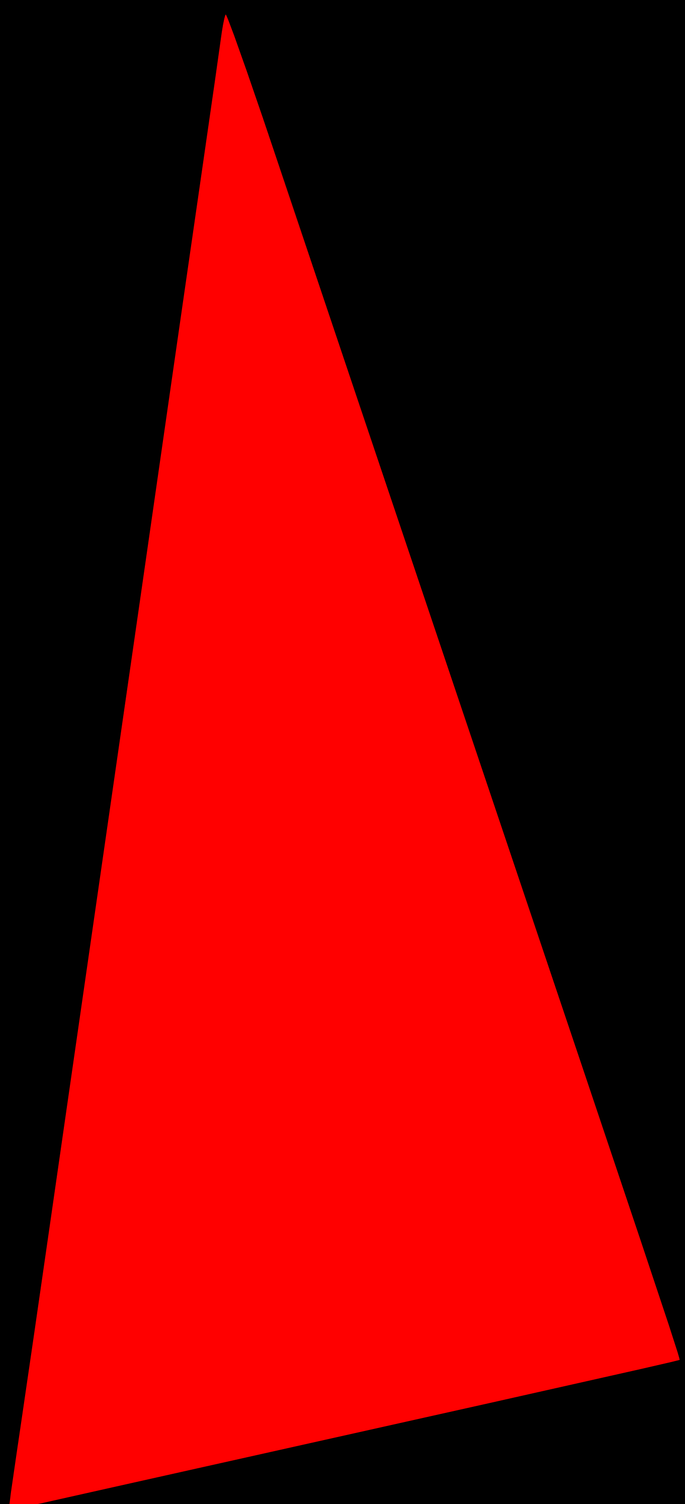
Control over vertex positions and pixel colors

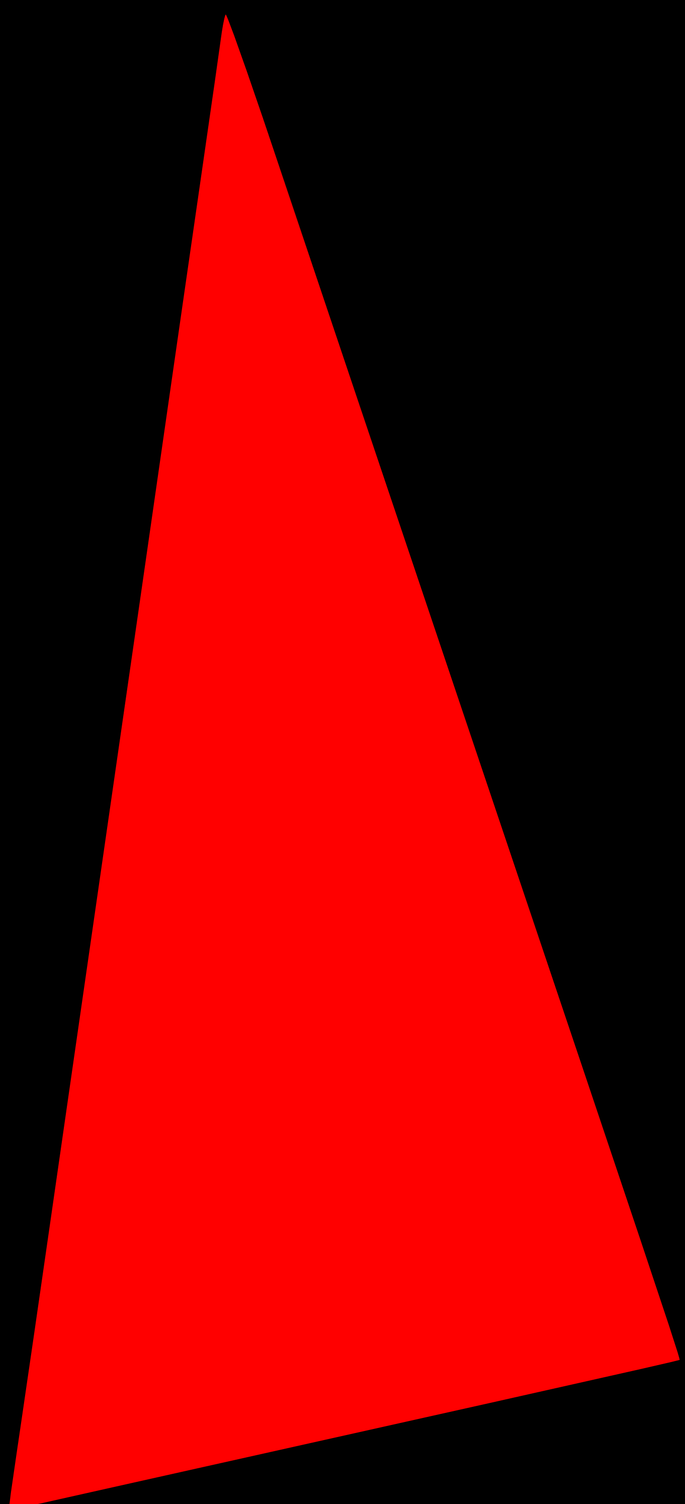
Extremely powerful

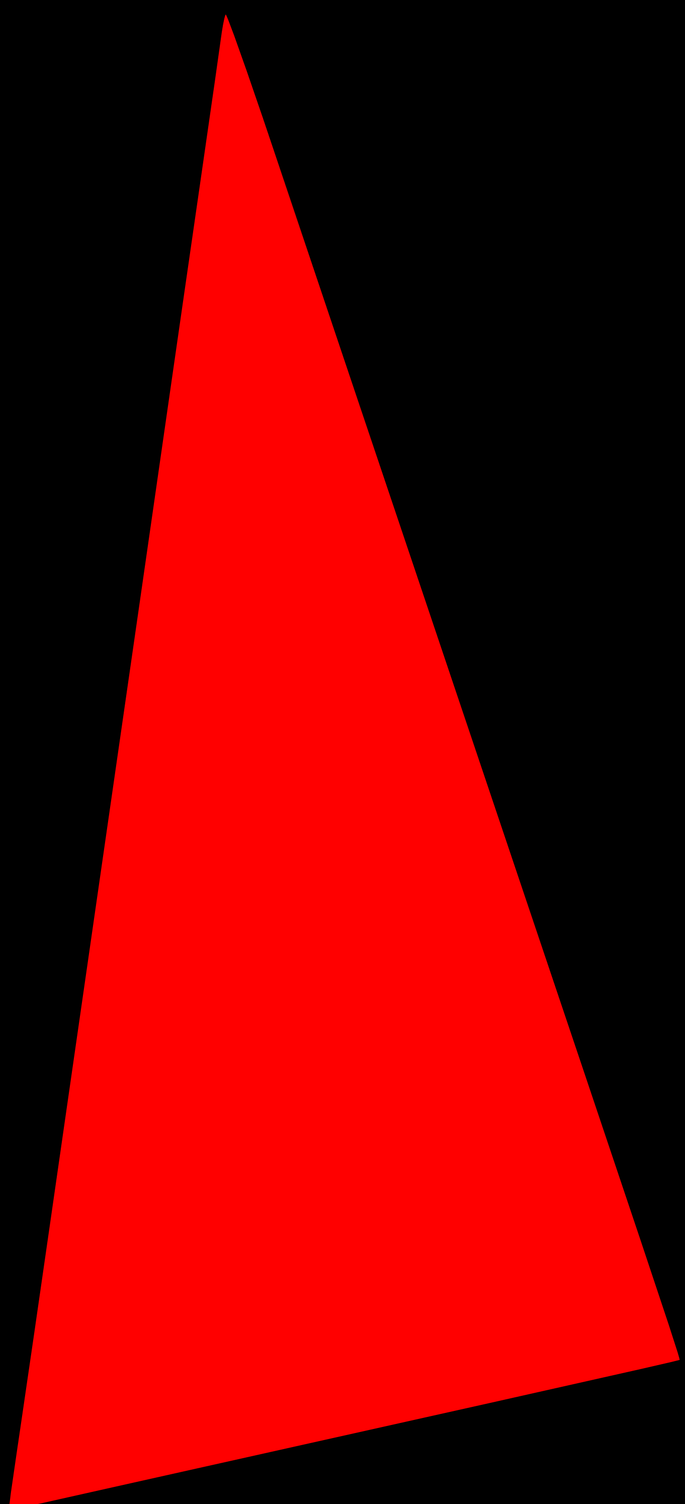
Advanced Rendering

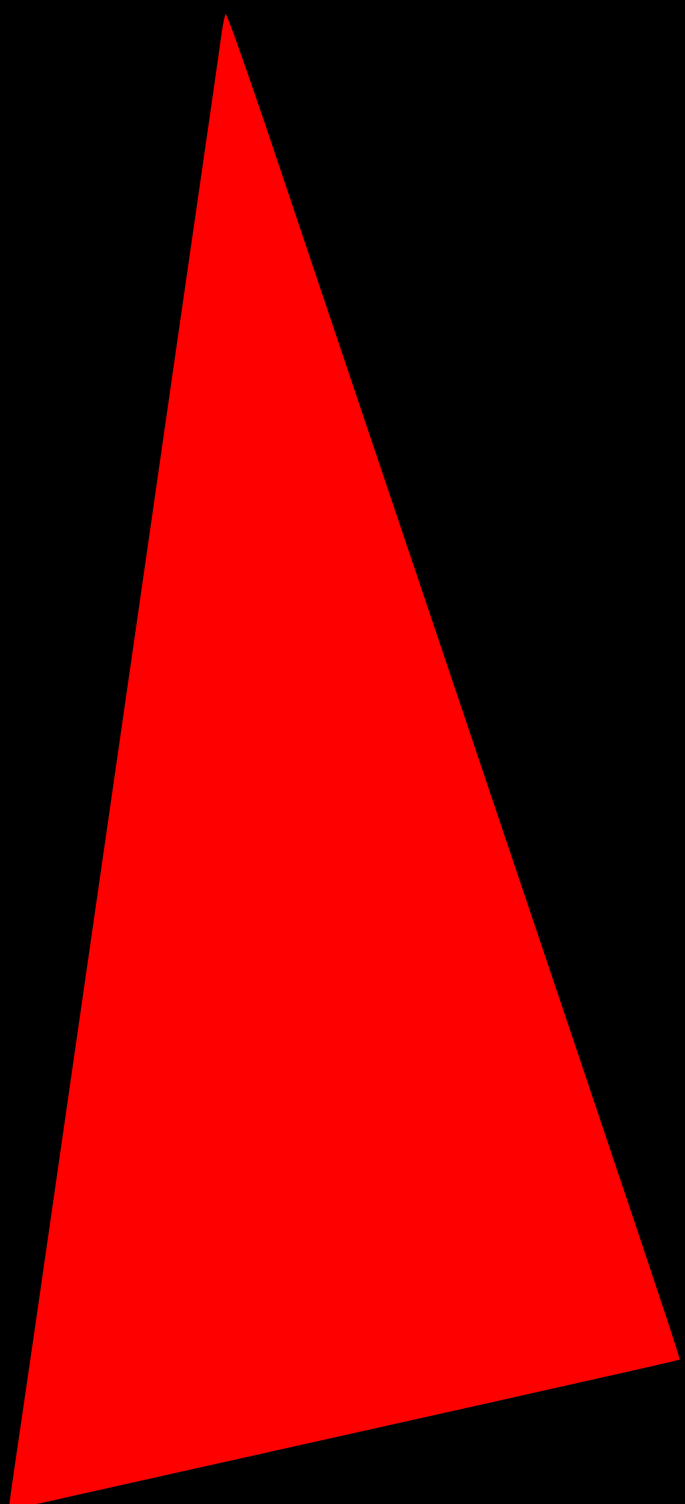
Merging WebGL with the rest of HTML

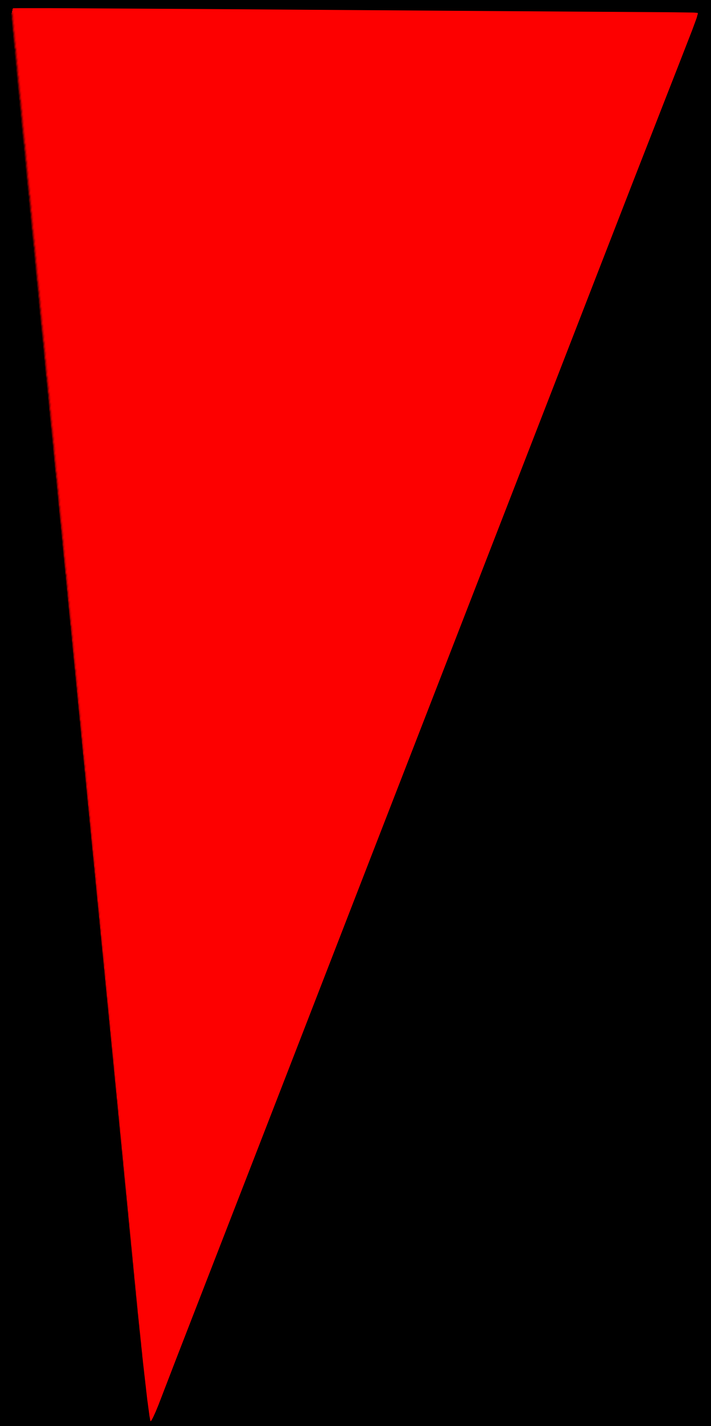
Brady Eidson
WebKit Engineer

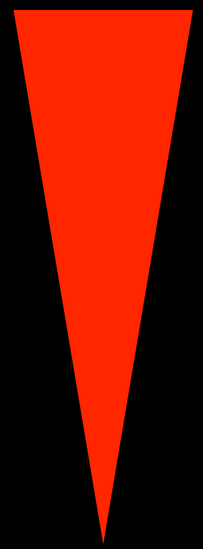


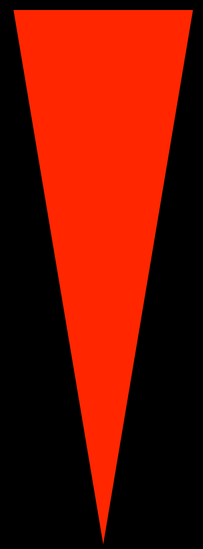


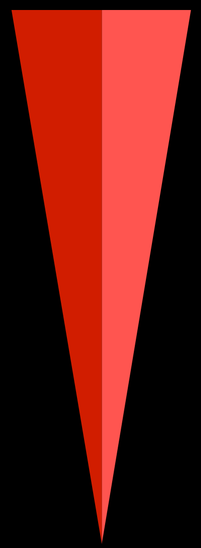


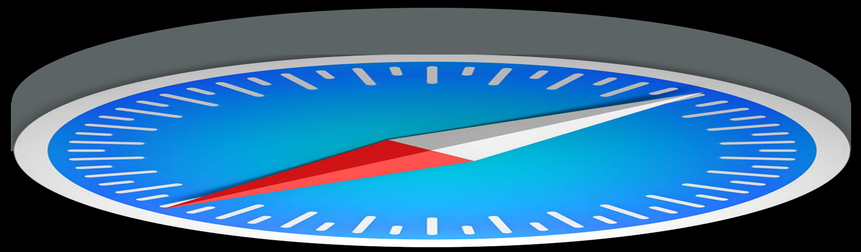


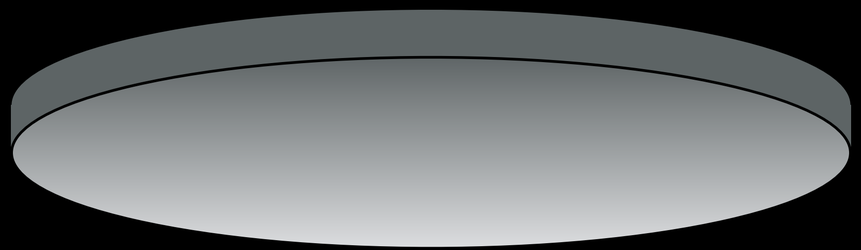


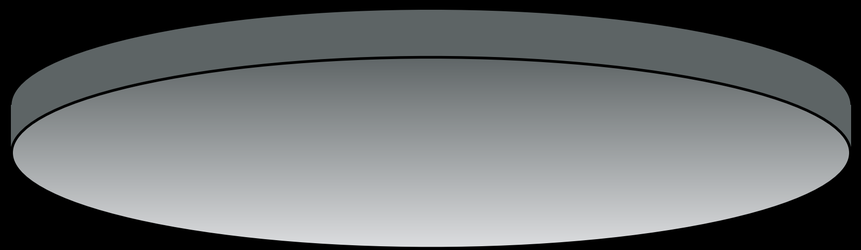


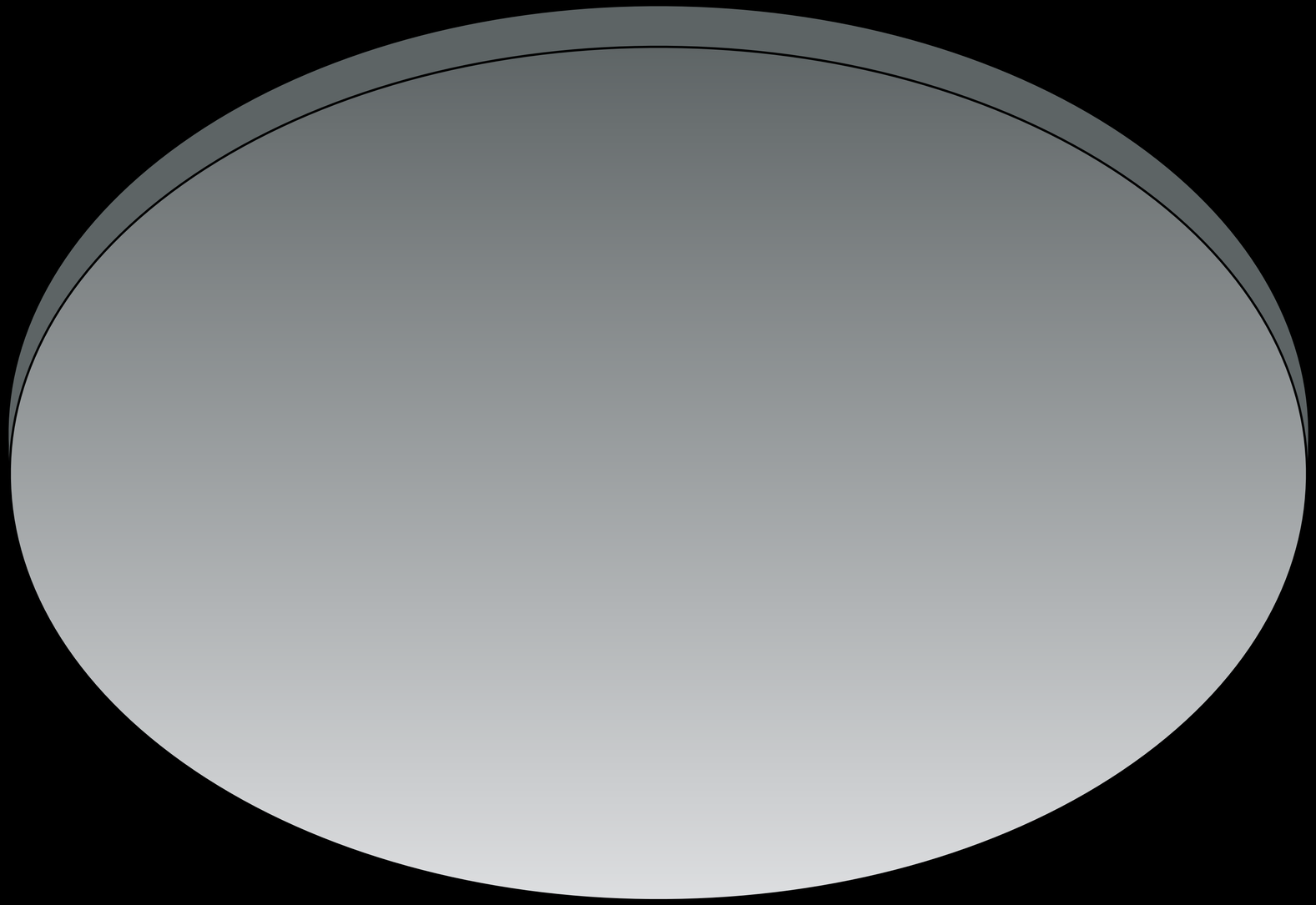


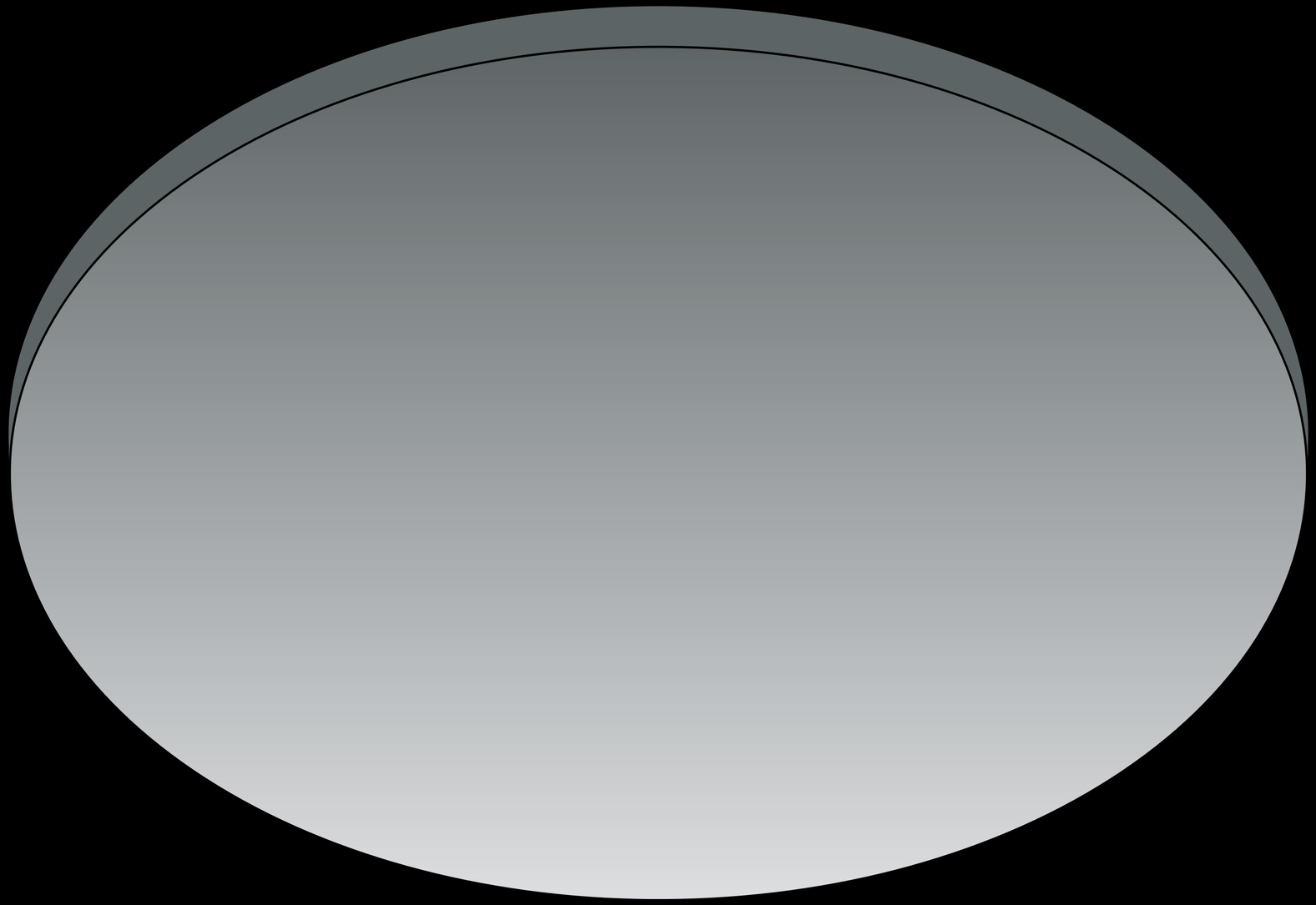


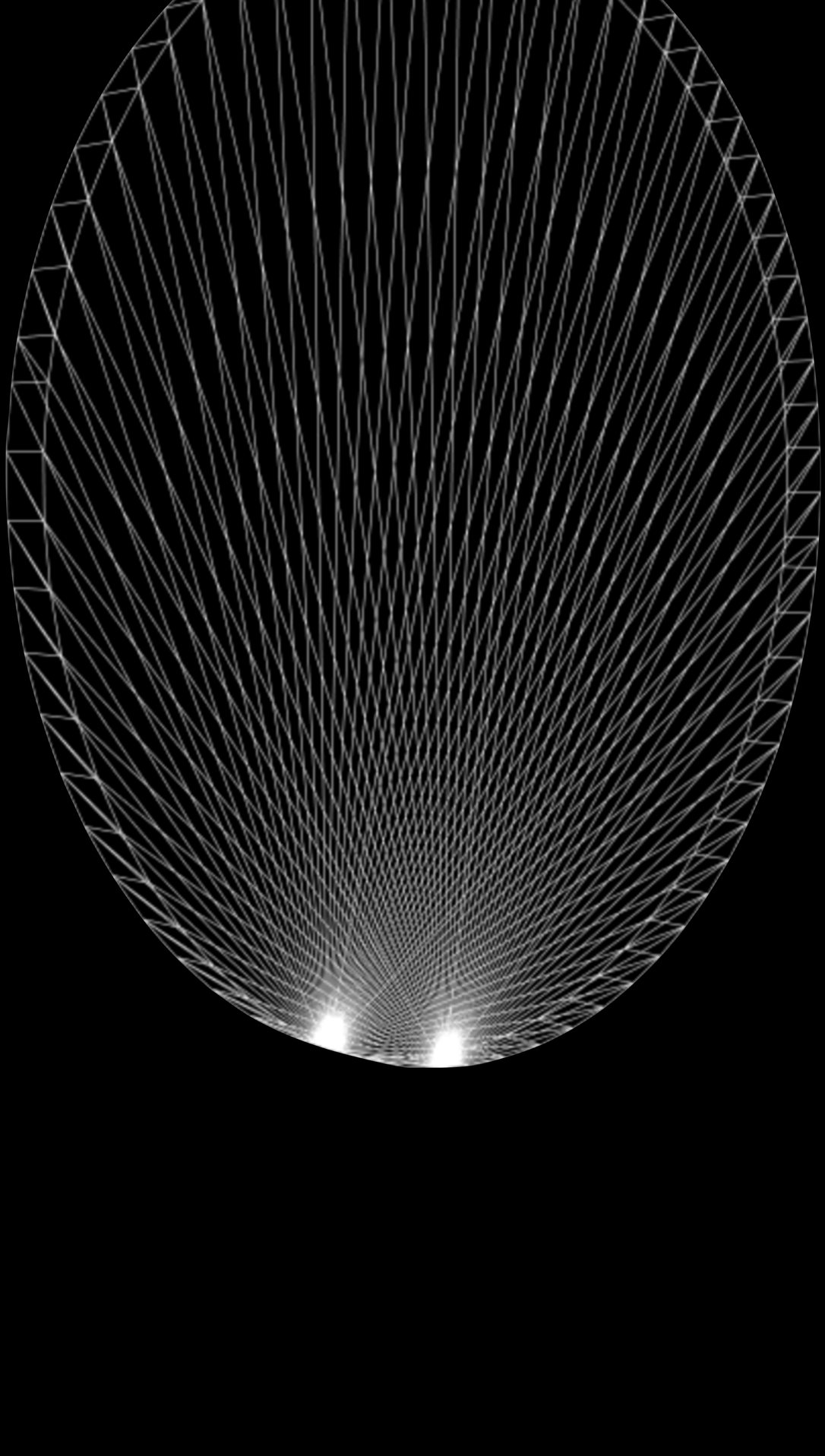












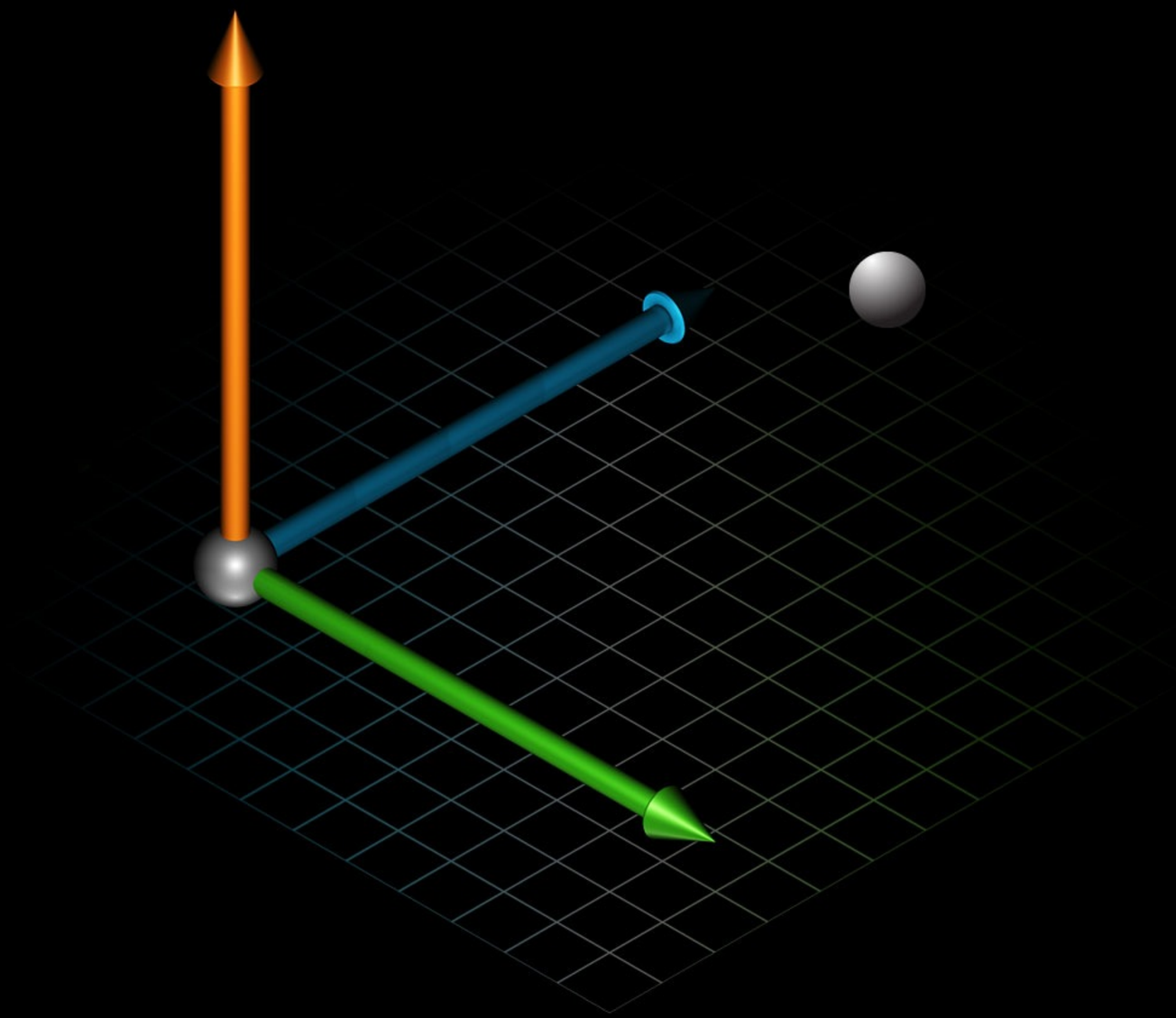
```
var vertices = new Float32Array([
   -0.8, -0.3,
   0.7, -0.8,
   0.55, 0.75
]);

var triangleBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, triangleBuffer);
gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);
```

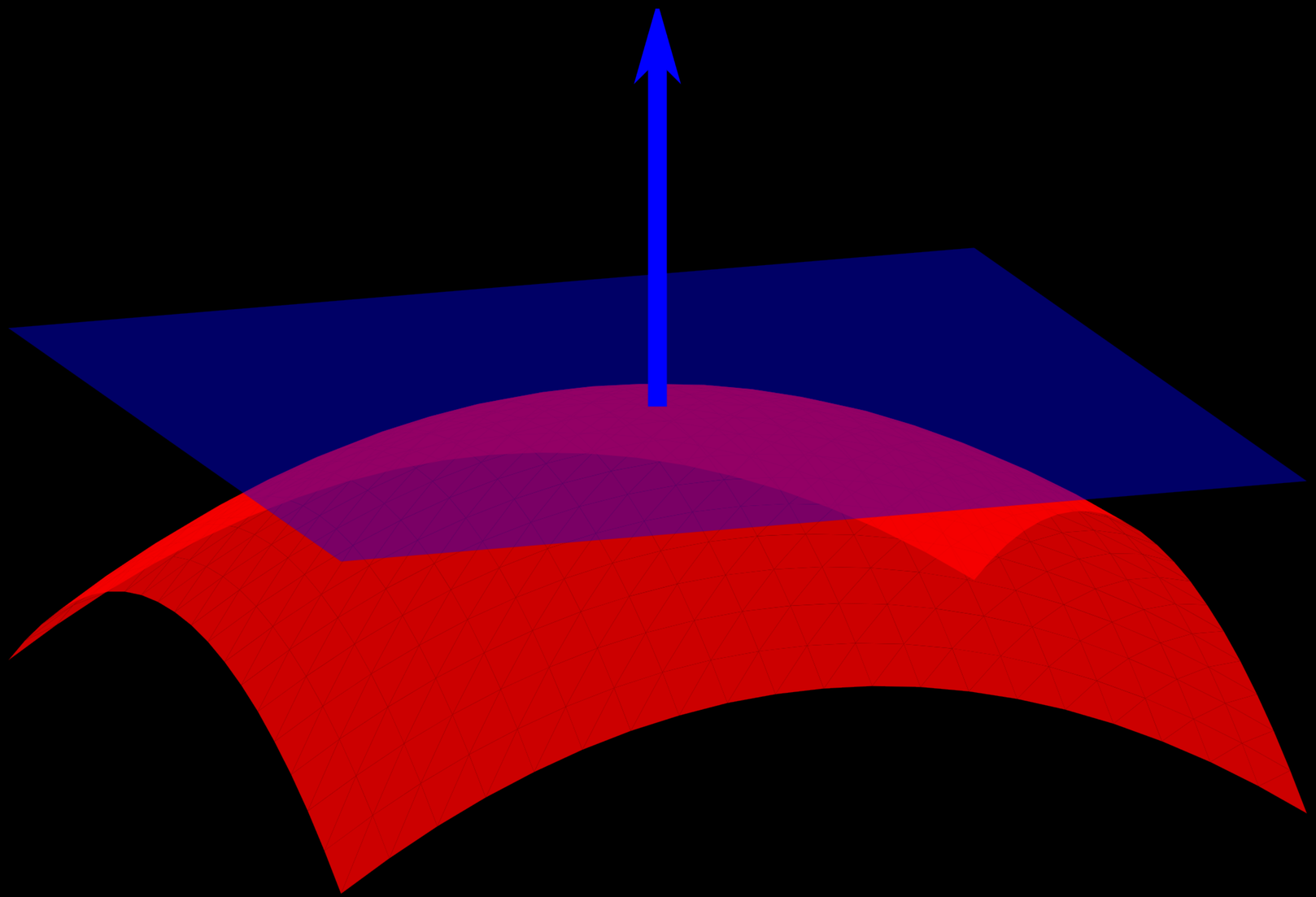
```
var vertices = new Float32Array([
   ...
]);

var discBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, discBuffer);
gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);
```

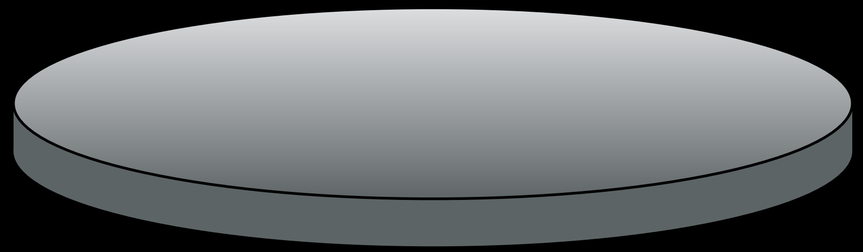
```
var vertices = new Float32Array([
   -0.020478, 28.7138, 0.149324,
   -0.020478, 70.3138, 0.149324,
   17.830223, 28.7138, 0.149324,
   17.830223, 28.7138, 0.149324,
   17.830223, 70.3138, 0.149324,
   17.830223, 70.3138, 0.149324,
   15.438682, 28.7138, -8.776026,
   15.438682, 28.7138, -8.776026,
   15.438682, 70.3138, -8.776026,
   15.438682, 70.3138, -8.776026,
   8.904872, 28.7138, -15.309836,
   8.904872, 28.7138, -15.309836,
   8.904872, 70.3138, -15.309836,
   8.904872, 70.3138, -15.309836,
   -0.020478, 28.7138, -17.701376,
   -0.020478, 28.7138, -17.701376,
   -0.020478, 70.3138, -17.701376,
   -0.020478, 70.3138, -17.701376,
   -8.945828, 28.7138, -15.309836,
```

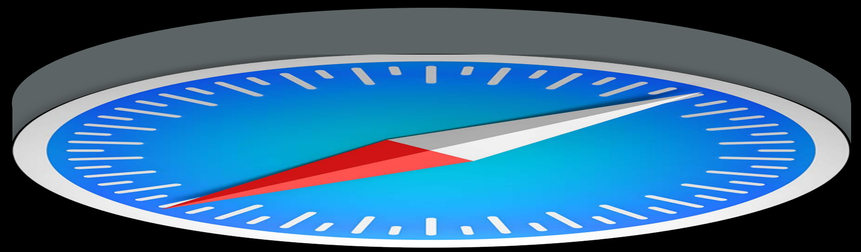
Any Data You Want

Any Data You Want for Any Point You Want









```
var vertices = new Float32Array([
   -0.020478, 28.7138, 0.149324,
   -0.020478, 70.3138, 0.149324,
   17.830223, 28.7138, 0.149324,
   17.830223, 28.7138, 0.149324,
   17.830223, 70.3138, 0.149324,
   17.830223, 70.3138, 0.149324,
   15.438682, 28.7138, -8.776026,
   15.438682, 28.7138, -8.776026,
   15.438682, 70.3138, -8.776026,
   15.438682, 70.3138, -8.776026,
  …
]);

var geometryBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, geometryBuffer);
gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);
```

```
var textureCoords = new Float32Array([
   0.498234, 0.530450
   0.498234, 0.600143
   0.512222, 0.530450
   0.512222, 0.530450
   0.512222, 0.600143
   0.512222, 0.600143
   0.509996, 0.530450
   0.509996, 0.530450
   0.509996, 0.600143
   0.509996, 0.600143
  …
]);

var textureBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, textureBuffer);
gl.bufferData(gl.ARRAY_BUFFER, textureCoords, gl.STATIC_DRAW);
```

Configuring Our Shaders

```
var positionAttribute = gl.getAttribLocation(program, "aPosition");
gl.enableVertexAttribArray(positionAttribute);

gl.bindBuffer(gl.ARRAY_BUFFER, positionBuffer);
gl.vertexAttribPointer(positionAttribute, 3, gl.FLOAT, false, 0, 0);
```

Configuring Our Shaders

```
var positionAttribute = gl.getAttribLocation(program, "aPosition");
gl.enableVertexAttribArray(positionAttribute);
gl.bindBuffer(gl.ARRAY_BUFFER, positionBuffer);
gl.vertexAttribPointer(positionAttribute, 3, gl.FLOAT, false, 0, 0);
```

```
var textureAttribute = gl.getAttribLocation(program, "aTextureCoord");
gl.enableVertexAttribArray(textureAttribute);
```

Configuring Our Shaders

```
var positionAttribute = gl.getAttribLocation(program, "aPosition");
gl.enableVertexAttribArray(positionAttribute);
gl.bindBuffer(gl.ARRAY_BUFFER, positionBuffer);
gl.vertexAttribPointer(positionAttribute, 3, gl.FLOAT, false, 0, 0);
```

```
var textureAttribute = gl.getAttribLocation(program, "aTextureCoord");
gl.enableVertexAttribArray(textureAttribute);
gl.bindBuffer(gl.ARRAY_BUFFER, textureBuffer);
gl.vertexAttribPointer(textureAttribute, 2, gl.FLOAT, false, 0, 0);
```

Vertex Shader Source Code

```
attribute vec4 aPosition;

void main() {
  gl_Position = aPosition;
}
```

Vertex Shader Source Code

```
attribute vec4 aPosition;
attribute vec2 aTextureCoord;

void main() {
  gl_Position = aPosition;
}
```

Vertex Shader Source Code

```
attribute vec4 aPosition;
attribute vec2 aTextureCoord;

varying vec2 vTextureCoord;

void main() {
  gl_Position = aPosition;
}
```

Vertex Shader Source Code

```
attribute vec4 aPosition;
attribute vec2 aTextureCoord;

varying vec2 vTextureCoord;

void main() {
  gl_Position = aPosition;
  vTextureCoord = aTextureCoord;
}
```

Fragment Shader Source Code

```
precision mediump float;

void main() {
  gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```

Fragment Shader Source Code

```
precision mediump float;

varying vec2 vTextureCoord;

void main() {
    gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```

Configuring Our Shaders

```
var positionAttribute = gl.getAttribLocation(program, "aPosition");
gl.enableVertexAttribArray(positionAttribute);
gl.bindBuffer(gl.ARRAY_BUFFER, positionBuffer);
gl.vertexAttribPointer(positionAttribute, 3, gl.FLOAT, false, 0, 0);
```

```
var textureAttribute = gl.getAttribLocation(program, "aTextureCoord");
gl.enableVertexAttribArray(textureAttribute);
gl.bindBuffer(gl.ARRAY_BUFFER, textureBuffer);
gl.vertexAttribPointer(textureAttribute, 2, gl.FLOAT, false, 0, 0);
```

Configuring Our Shaders

```
var positionAttribute = gl.getAttribLocation(program, "aPosition");
gl.enableVertexAttribArray(positionAttribute);
gl.bindBuffer(gl.ARRAY_BUFFER, positionBuffer);
gl.vertexAttribPointer(positionAttribute, 3, gl.FLOAT, false, 0, 0);

var textureAttribute = gl.getAttribLocation(program, "aTextureCoord");
gl.enableVertexAttribArray(textureAttribute);
gl.bindBuffer(gl.ARRAY_BUFFER, textureBuffer);
gl.vertexAttribPointer(textureAttribute, 2, gl.FLOAT, false, 0, 0);

var samplerUniform = gl.getUniformLocation(program, "sampler");
```

Fragment Shader Source Code

```
precision mediump float;

varying vec2 vTextureCoord;

void main() {
    gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```

Fragment Shader Source Code

```
precision mediump float;

varying vec2 vTextureCoord;

uniform sampler2D sampler;
```

```
void main() {
  gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```

Fragment Shader Source Code

```
precision mediump float;

varying vec2 vTextureCoord;

uniform sampler2D sampler;
```

```
void main() {
  gl_FragColor = texture2D(sampler, vTextureCoord);
}
```

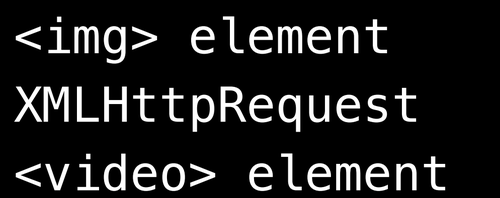
Texture Sources

Texture Sources

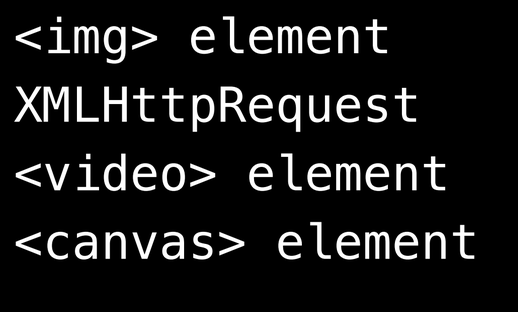
`<img>` element

Texture Sources

Texture Sources



Texture Sources



```
<img id=“TextureImage” src=“FlatCompass.png”>
```

```
<img id="TextureImage" src="FlatCompass.png">

<script type="text/javascript">
var texture = gl.createTexture();
</script>
```

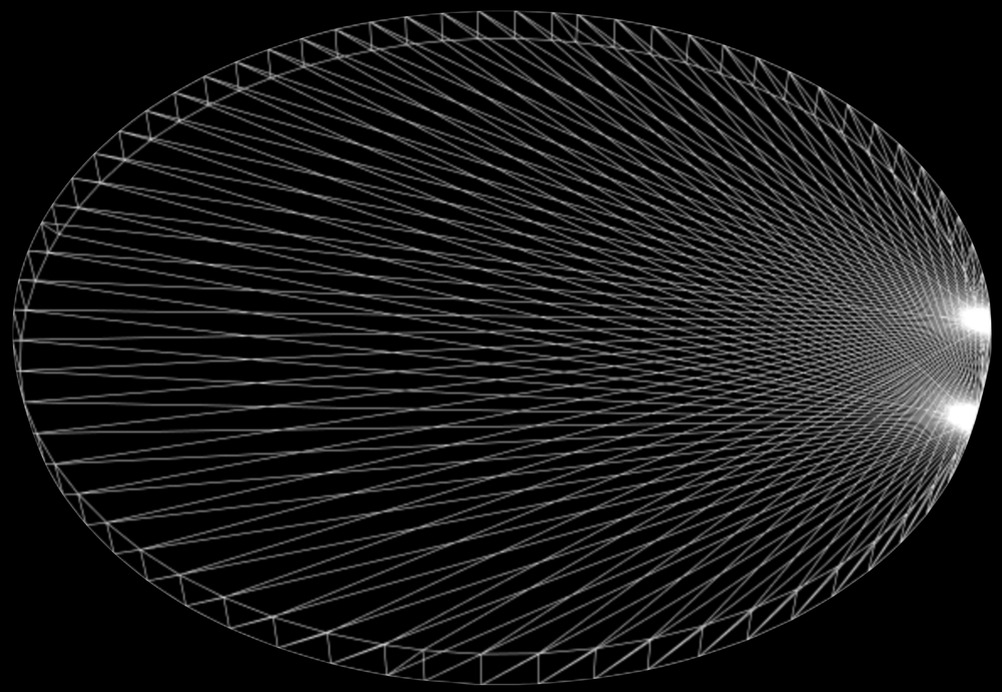
```
<img id=“TextureImage” src=“FlatCompass.png”>

<script type=“text/javascript”>
var texture = gl.createTexture();
gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, texture);
</script>
```

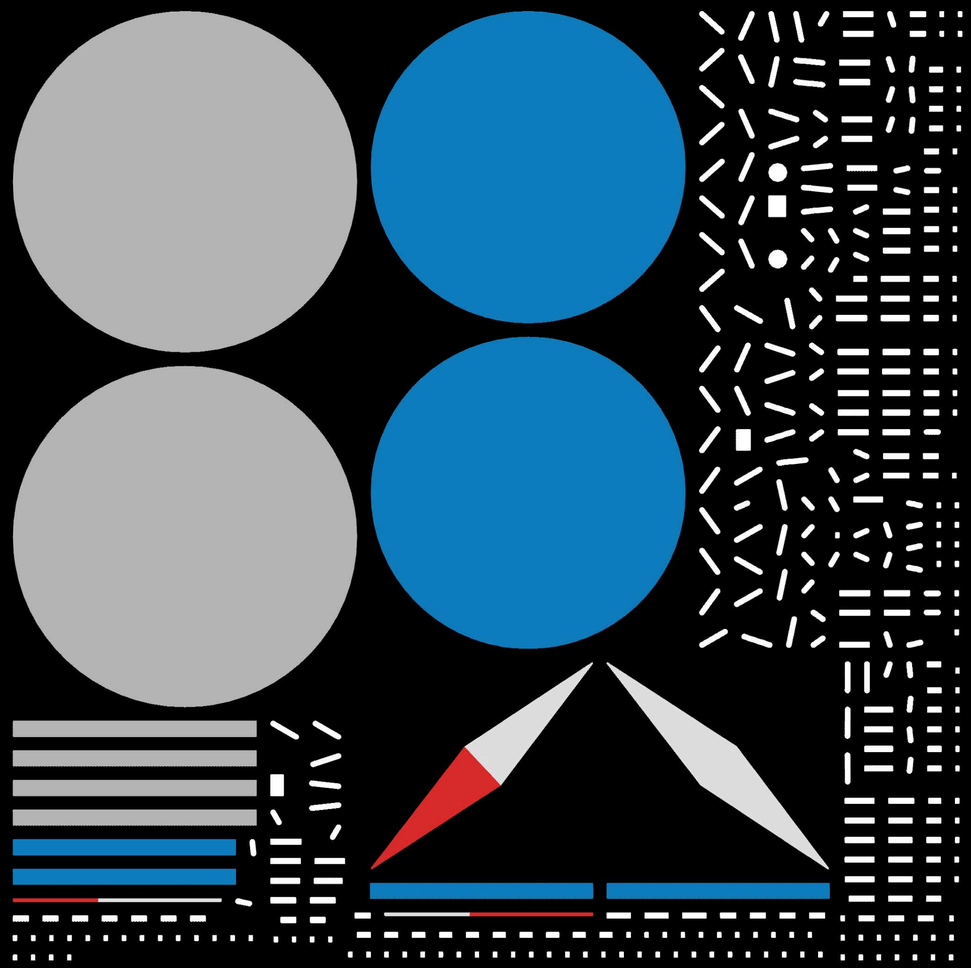
```
<img id=“TextureImage” src=“FlatCompass.png”>

<script type=“text/javascript”>
var texture = gl.createTexture();
gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, texture);

var image = document.getElementById(“TextureImage”);
</script>
```

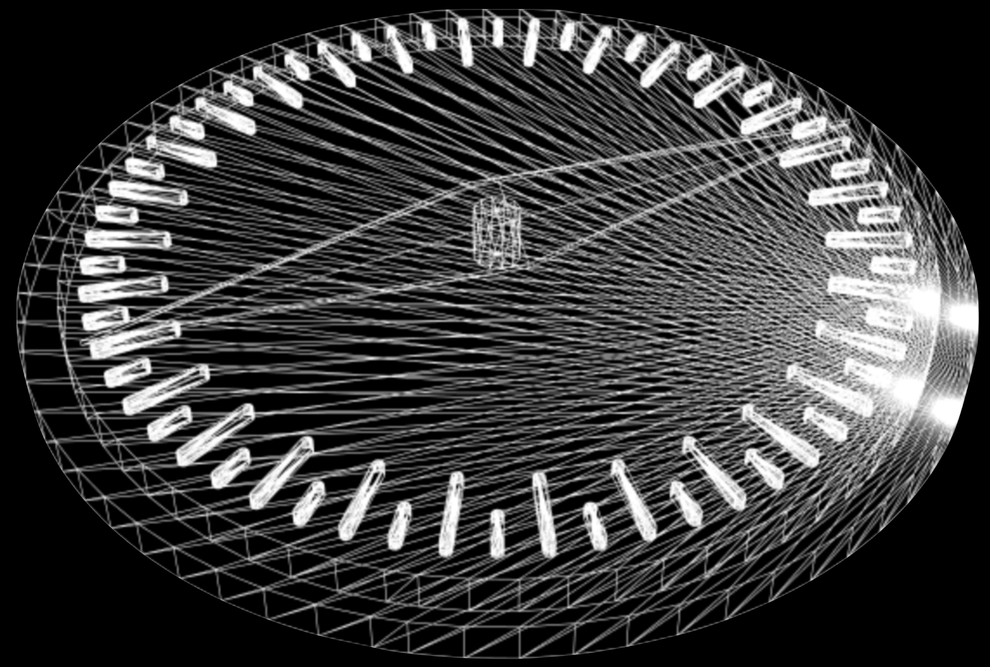
```
<img id=“TextureImage” src=“FlatCompass.png”>

<script type=“text/javascript”>
var texture = gl.createTexture();
gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, texture);

var image = document.getElementById(“TextureImage”);

gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);
</script>
```

```
<img id=“TextureImage” src=“FlatCompass.png”>

<script type=“text/javascript”>
var texture = gl.createTexture();
gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, texture);

var image = document.getElementById(“TextureImage”);

gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);
</script>
```

```
<img id=“TextureImage” src=“FlatCompass.png”>

<script type=“text/javascript”>
var texture = gl.createTexture();
gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, texture);

var image = document.getElementById(“TextureImage”);

gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);
gl.uniform1i(samplerUniform, 0);
</script>
```







Demo

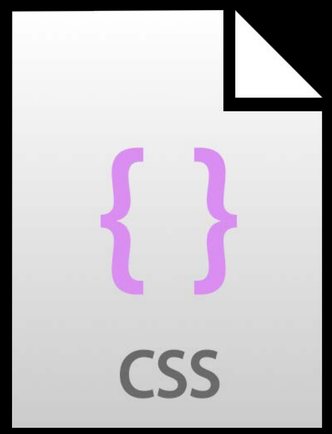
Building a compass

The Web Platform

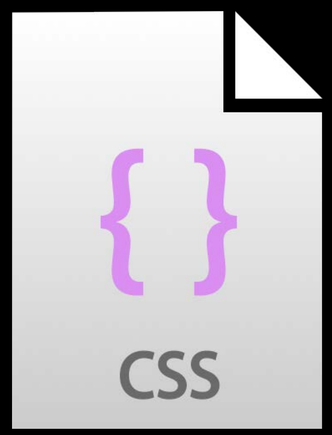
The Web Platform

The Web Platform

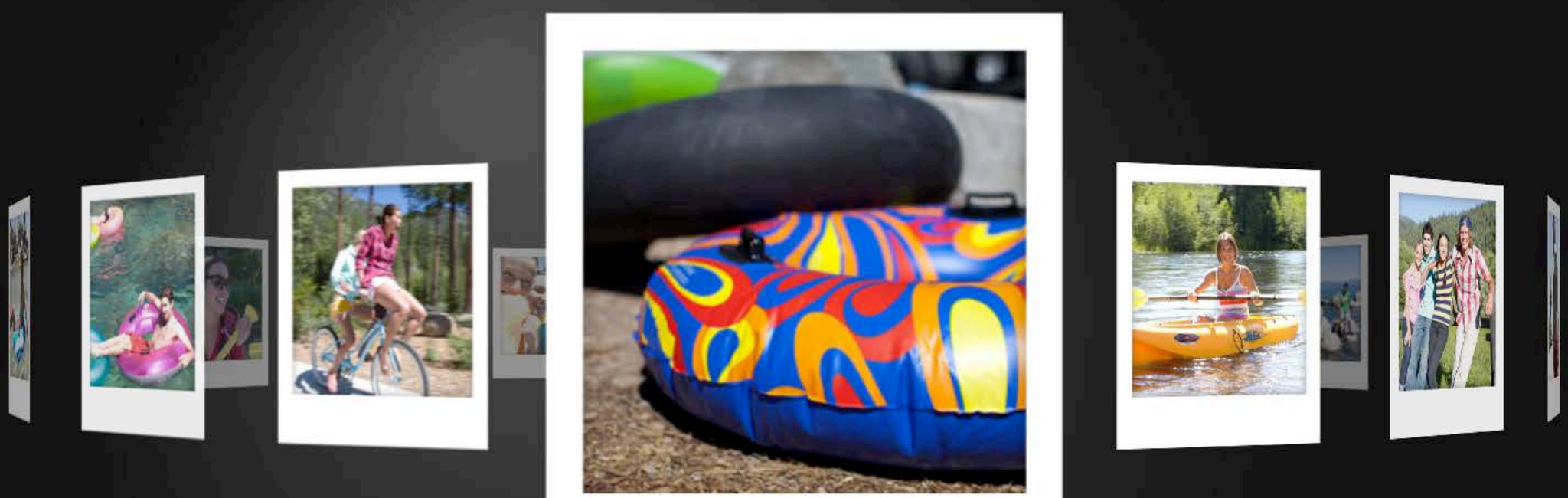
The Web Platform



The Web Platform

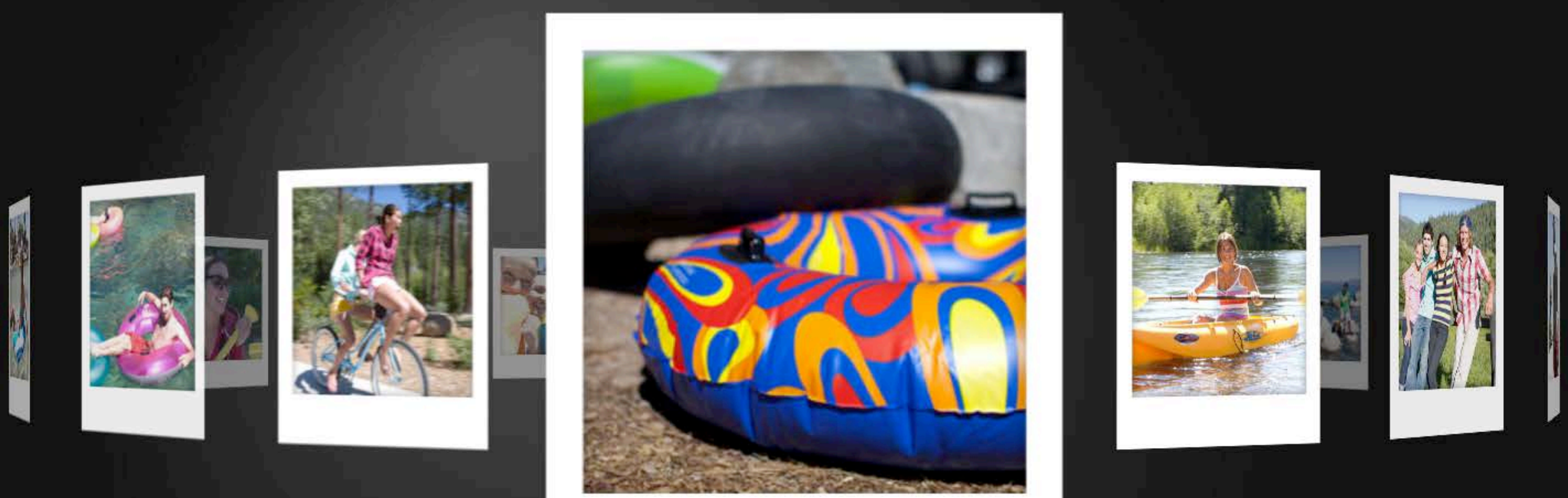


Gallery



◀ Prev Next ▶

Gallery



◀ Prev Next ▶

Chunk Five

Size

Color

Leading

Tracking

Transparency

Rotation

Shadow ON OFF

Hello.

Chunk Five

Size

Color

Leading

Tracking

Transparency

Rotation

Shadow ON OFF

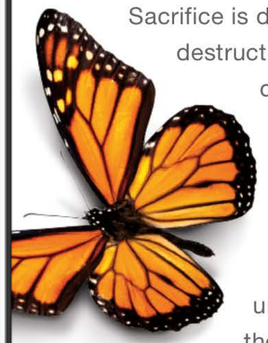
Hello.

iPad 9:41 AM 100%

Acknowledge the Risk

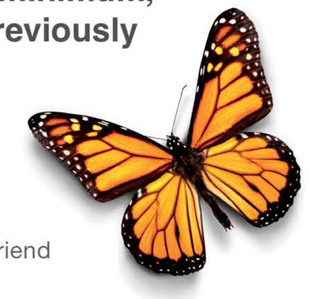
People experience a sense of fear when they embark on a journey that involves change, because change has an element of the unknown. That is what makes it so frightening.

Change is about accepting the new and abandoning the old. New societies cannot rise unless old societies fall. If new technology emerges, it renders the old technology obsolete. In the presentation space, accepting something new often means sacrificing something held dear.



Sacrifice is defined as the surrender or destruction of something prized or desirable for the sake of something considered as having a higher or more pressing claim. Often, your audience can't change without making a sacrifice. You need to make them understand that without sacrifice, there can be no reward.

To adopt your perspective, the audience has to, at a minimum, abandon what they previously held as true.



Changing their minds is like asking them to forsake an old friend who has stood by them for a long time. Losing an old friend is painful.

Even something seemingly trivial—like a forfeit of their time—might require them to risk something. Working late might mean missing volleyball practice or the chance to tuck their kids into bed at night. Be cognizant of the sacrifice the audience will make when you ask them to do something, because you're asking them to give up a small—but still irretrievable—slice of their lives.

When to Draw

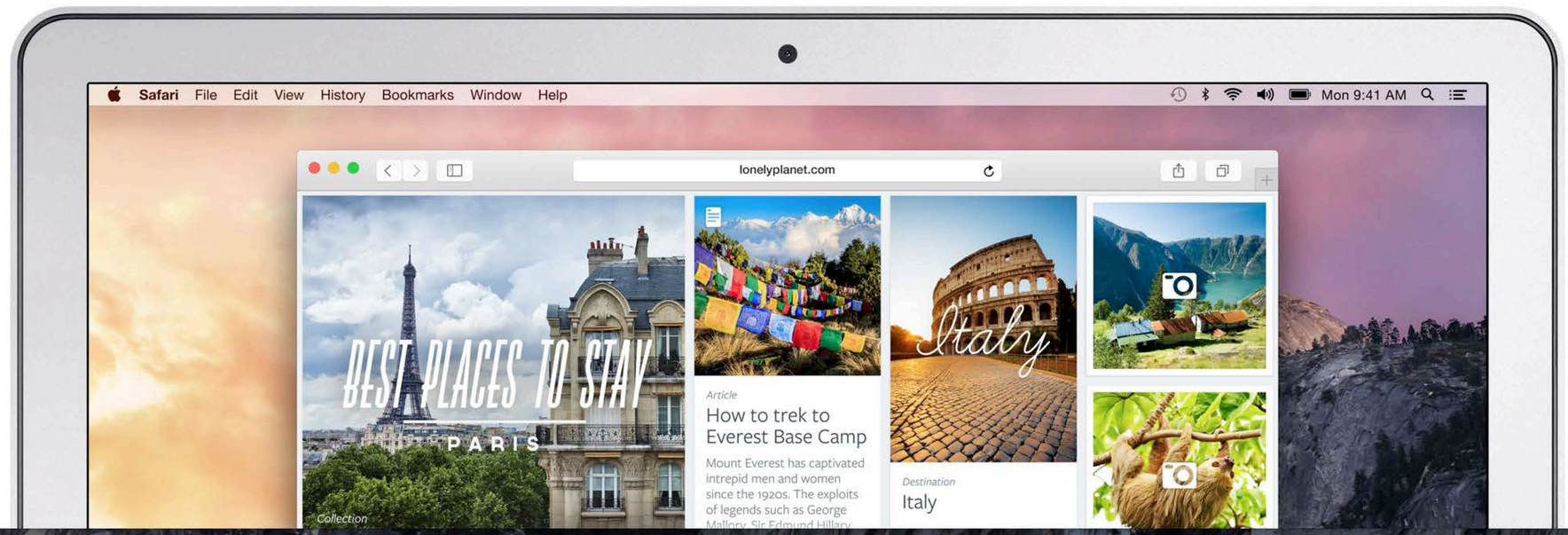
Safari File Edit View History Bookmarks Window Help Mon 9:32 PM

apple.com

Safari

The smartest way to surf just got smarter.

In OS X Yosemite, a streamlined toolbar in Safari puts your most important controls at your fingertips and gives you more room for what you're viewing. Safari also gives you new ways to access your favorite sites, manage your tabs, and have more control over your privacy. With an improved Nitro JavaScript engine and support for the latest web standards, it's the fastest, most advanced way to browse the web.



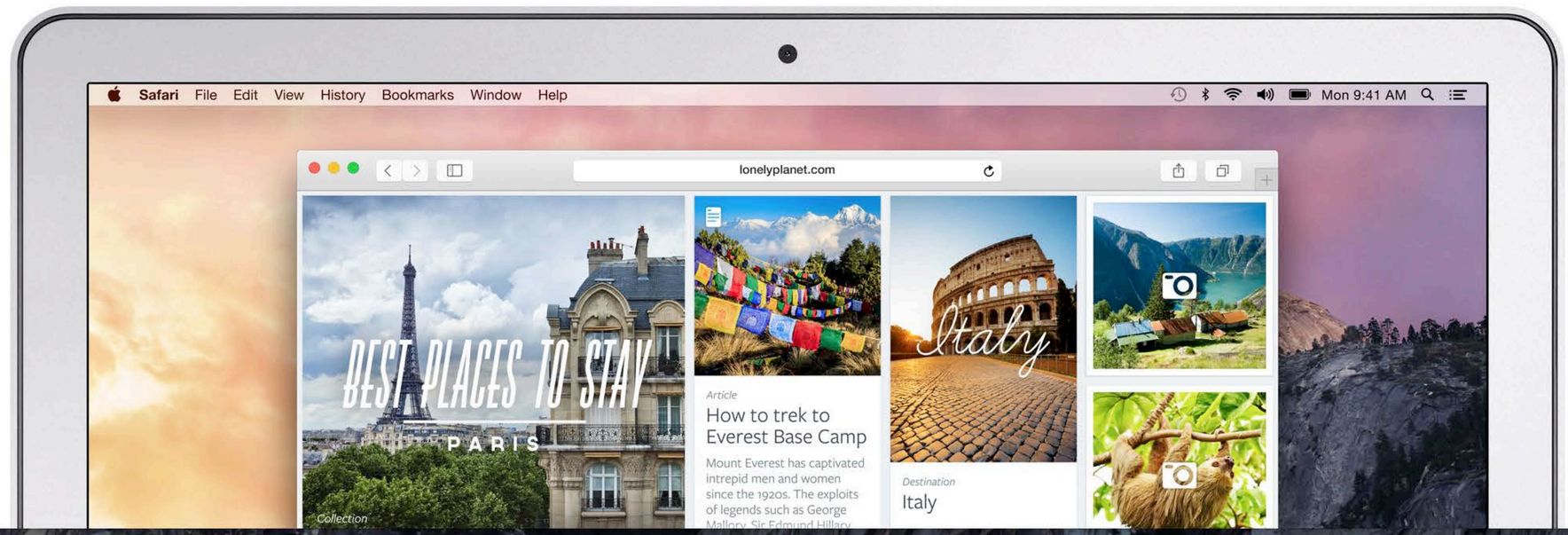
Safari File Edit View History Bookmarks Window Help Mon 9:32 PM

apple.com

Safari

The smartest way to surf just got smarter.

In OS X Yosemite, a streamlined toolbar in Safari puts your most important controls at your fingertips and gives you more room for what you're viewing. Safari also gives you new ways to access your favorite sites, manage your tabs, and have more control over your privacy. With an improved Nitro JavaScript engine and support for the latest web standards, it's the fastest, most advanced way to browse the web.



Do Not Use Timers!

requestAnimationFrame()

requestAnimationFrame()

1. Call “requestAnimationFrame()” with a callback

requestAnimationFrame()

1. Call “requestAnimationFrame()” with a callback

```
var drawingCallback = function() {
  …
}
```

requestAnimationFrame()

1. Call “requestAnimationFrame()” with a callback

```
var drawingCallback = function() {
  …
}

requestAnimationFrame(drawingCallback);
```

requestAnimationFrame()

2 Draw your scene

```
var drawingCallback = function() {
  …
}

requestAnimationFrame(drawingCallback);
```

requestAnimationFrame()

2 Draw your scene

```
var drawingCallback = function() {
  updatePhysics();
  drawCompass();
  drawTerrain();
}

requestAnimationFrame(drawingCallback);
```

requestAnimationFrame()

3 Request the next callback

```
var drawingCallback = function() {
  updatePhysics();
  drawCompass();
  drawTerrain();
}

requestAnimationFrame(drawingCallback);
```

requestAnimationFrame()

3 Request the next callback

```
var drawingCallback = function() {
  updatePhysics();
  drawCompass();
  drawTerrain();
  …
  requestAnimationFrame(drawingCallback);
}

requestAnimationFrame(drawingCallback);
```

Demo

Grand finale

Wrapping Up

Wrapping Up

Rich, powerful, fast graphics

Wrapping Up

Rich, powerful, fast graphics

WebGL available in Safari on OS X Yosemite and iOS 8

Wrapping Up

Rich, powerful, fast graphics

WebGL available in Safari on OS X Yosemite and iOS 8

Also available in WKWebView

More Information

Evangelism
evangelism@apple.com

Safari for Developers
http://developer.apple.com/safari/

WebKit
http://webkit.org/

Developer Technical Support
http://developer.apple.com/contact

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Introducing the Modern WebKit API	Nob Hill	Tuesday 2:00PM
Web Inspector and Modern JavaScript	Russian Hill	Thursday 10:15AM

Labs

• Safari and WebKit Lab	Media Lab B	Wednesday 4:30PM
• Safari and WebKit Lab	Media Lab B	Thursday 2:00PM

WWDC14