7 - WebGL

https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial

https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/By_example/

https://webglfundamentals.org

http://stack.gl/

https://www.khronos.org/files/webgl/webgl-reference-card-1_0.pdf

WebGL1 vs WebGL2 vs WebGLNext vs … GLSL versions …

- WebGL1 ~= OpenGL ES2, WebGL2 ~= OpenGL ES3
- https://webgl2fundamentals.org/webgl/lessons/webgl1-to-webgl2.html

- https://www.khronos.org/webgl/
- https://www.khronos.org/files/webgl/webgl-reference-card-1_0.pdf
- https://www.khronos.org/files/webgl20-reference-guide.pdf

WebGL: Rasterizing Primitives

- vs CSS3 3D Transforms: positioning rectangles

Shaders in GLSL (GL Shader Language)

- Vertex shader + fragment shader

Walk through example

https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/By_example/Hello_GLSL

function getRenderingContext()

```
var canvas = document.querySelector("canvas");
canvas.width = canvas.clientWidth;
canvas.height = canvas.clientHeight;
var gl = canvas.getContext("webgl") || canvas.getContext("experimental-webgl");
if (!gl) {
  var paragraph = document.querySelector("p");
  paragraph.innerHTML = "Failed to get WebGL context.";
  return null;
}
gl.viewport(0, 0, gl.drawingBufferWidth, gl.drawingBufferHeight);
gl.clearColor(0.0, 0.0, 0.0, 1.0);
gl.clear(gl.COLOR_BUFFER_BIT);
return gl;
```

Shaders

```
<script type="x-shader/x-vertex" id="vertex-shader">
#version 100

void main() {
   gl_Position = vec4(0.0, 0.0, 0.0, 1.0);
   gl_PointSize = 64.0;
}

</script>
```

```
<script type="x-shader/x-fragment" id="fragment-shader">
#version 100

void main() {
  gl_FragColor = vec4(0.18, 0.54, 0.34, 1.0);
}

</script>
```

Built-In Inputs, Outputs, and Constants [7]

Shader programs use Special Variables to communicate with fixed-function parts of the pipeline. Output Special Variables may be read back after writing. Input Special Variables are read-only. All Special Variables have global scope.

Vertex Shader Special Variables [7.1]

Outputs:

Variable	Description	Units or coordinate system
highp vec4 gl_Position;	transformed vertex position	clip coordinates
mediump float gl_PointSize;	transformed point size (point rasterization only)	pixels

Fragment Shader Special Variables [7.2]

Fragment shaders may write to **gl_FragColor** or to one or more elements of **gl_FragData[]**, but not both.
The size of the **gl_FragData** array is given by the built-in constant **gl_MaxDrawBuffers**.

Inputs:

Variable	Description	Units or coordinate system
mediump vec4 gl_FragCoord;	fragment position within frame buffer	window coordinates
bool gl_FrontFacing;	fragment belongs to a front-facing primitive	Boolean
mediump vec2 gl_PointCoord;	fragment position within a point (point rasterization only)	0.0 to 1.0 for each component

Outputs:

Variable	Description	Units or coordinate system
mediump vec4 gl_FragColor;	fragment color	RGBA color
mediump vec4 gl_FragData[*n*]	fragment color for color attachment *n*	RGBA color

```
var source = document.querySelector("#vertex-shader").innerHTML;
var vertexShader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vertexShader,source);
gl.compileShader(vertexShader);

source = document.querySelector("#fragment-shader").innerHTML
var fragmentShader = gl.createShader(gl.FRAGMENT_SHADER);
gl.shaderSource(fragmentShader,source);
gl.compileShader(fragmentShader);
```

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


gl.detachShader(program, vertexShader);
gl.detachShader(program, fragmentShader);
gl.deleteShader(vertexShader);
gl.deleteShader(fragmentShader);
if (!gl.getProgramParameter(program, gl.LINK_STATUS)) { // error … }
```

```
gl.enableVertexAttribArray(0);
buffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, buffer);

// void gl.vertexAttribPointer(index, size, type, normalized, stride, offset);
gl.vertexAttribPointer(0, 1, gl.FLOAT, false, 0, 0);

gl.useProgram(program);

// void gl.drawArrays(mode, first, count);
gl.drawArrays(gl.POINTS, 0, 1);
```

Bit more complex

https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/By_example/Hello_vertex_attributes

```
<script type="x-shader/x-vertex" id="vertex-shader">
#version 100
precision highp float;

attribute float position;

void main() {
  gl_Position = vec4(position, 0.0, 0.0, 1.0);
  gl_PointSize = 64.0;
}

</script>
```

```
<script type="x-shader/x-fragment" id="fragment-shader">
#version 100
precision mediump float;

void main() {
  gl_FragColor = vec4(0.18, 0.54, 0.34, 1.0);
}

</script>
```

```
gl.enableVertexAttribArray(0);
buffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, buffer);

// void gl.vertexAttribPointer(index, size, type, normalized, stride, offset);
gl.vertexAttribPointer(0, 1, gl.FLOAT, false, 0, 0);

gl.useProgram(program);

// void gl.drawArrays(mode, first, count);
gl.drawArrays(gl.POINTS, 0, 1);
```

```
gl.enableVertexAttribArray(0);
buffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, buffer);
gl.bufferData(gl.ARRAY_BUFFER, new Float32Array([0.0]), gl.STATIC_DRAW);
// void gl.vertexAttribPointer(index, size, type, normalized, stride, offset);
gl.vertexAttribPointer(0, 1, gl.FLOAT, false, 0, 0);

gl.useProgram(program);

// void gl.drawArrays(mode, first, count);
gl.drawArrays(gl.POINTS, 0, 1);
```

On Mouse Click …

```
var clickXrelativToCanvas = evt.pageX - evt.target.offsetLeft;
var clickXinWebGLCoords = 2.0 *
                (clickXrelativToCanvas- gl.drawingBufferWidth/2) /
                gl.drawingBufferWidth;
gl.bufferData(gl.ARRAY_BUFFER,
                 new Float32Array([clickXinWebGLCoords]),
                 gl.STATIC_DRAW);
gl.drawArrays(gl.POINTS, 0, 1);
```

A bit more …

https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Adding_2D_content_to_a_WebGL_context

https://webglfundamentals.org/webgl/lessons/webgl-how-it-works.html