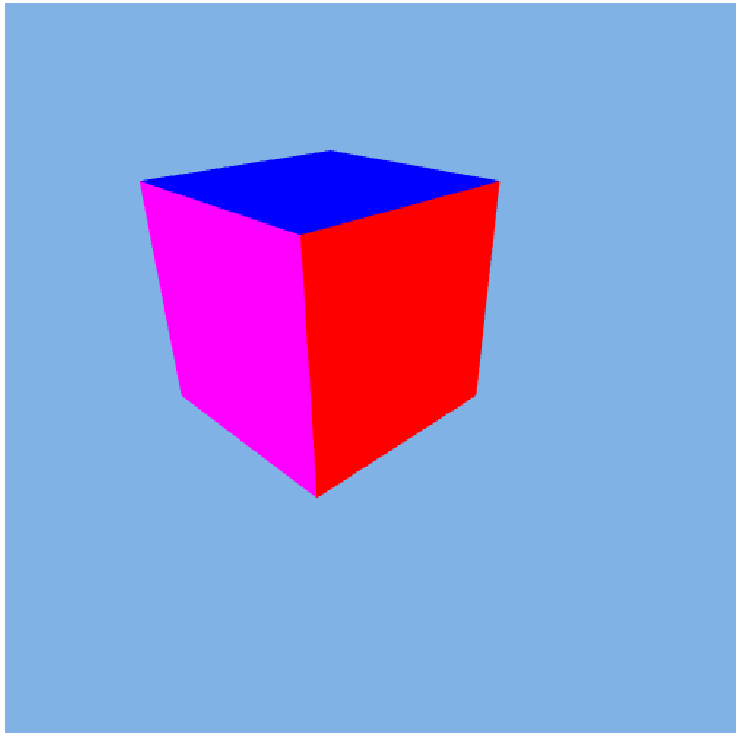
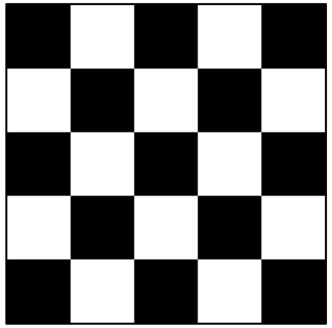
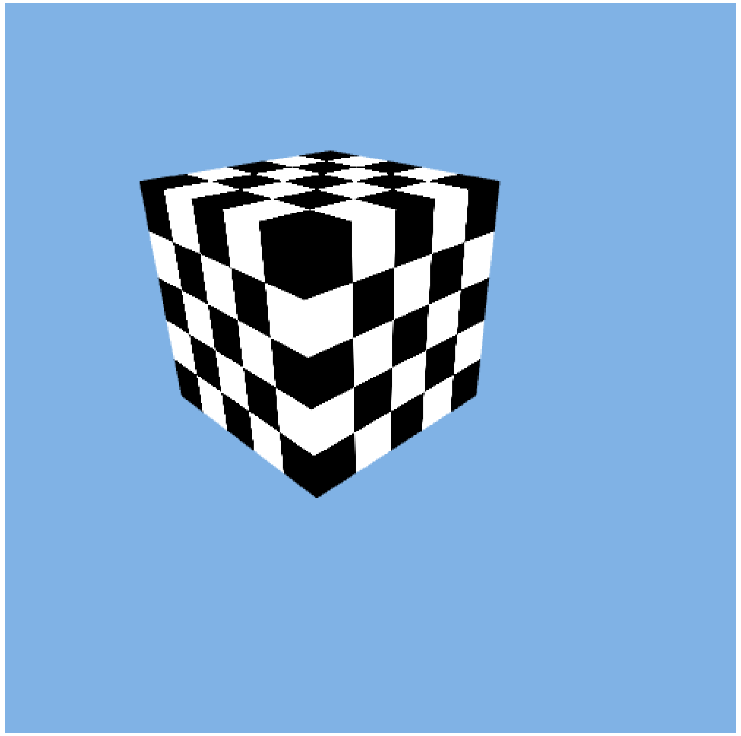
CSE4204
LAB-5 : texture mapping, animation

Mohammad Imrul Jubair

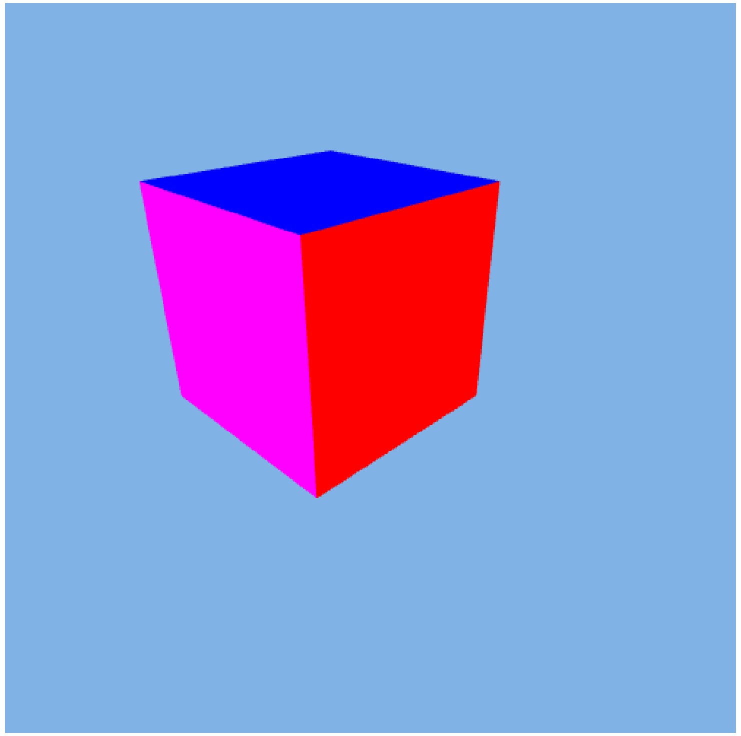
Texture Mapping

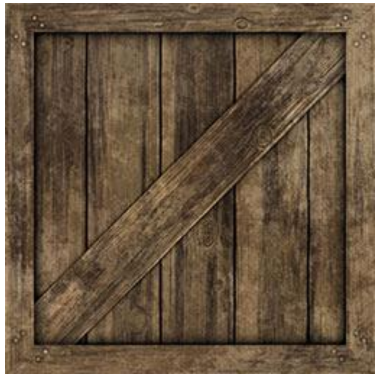


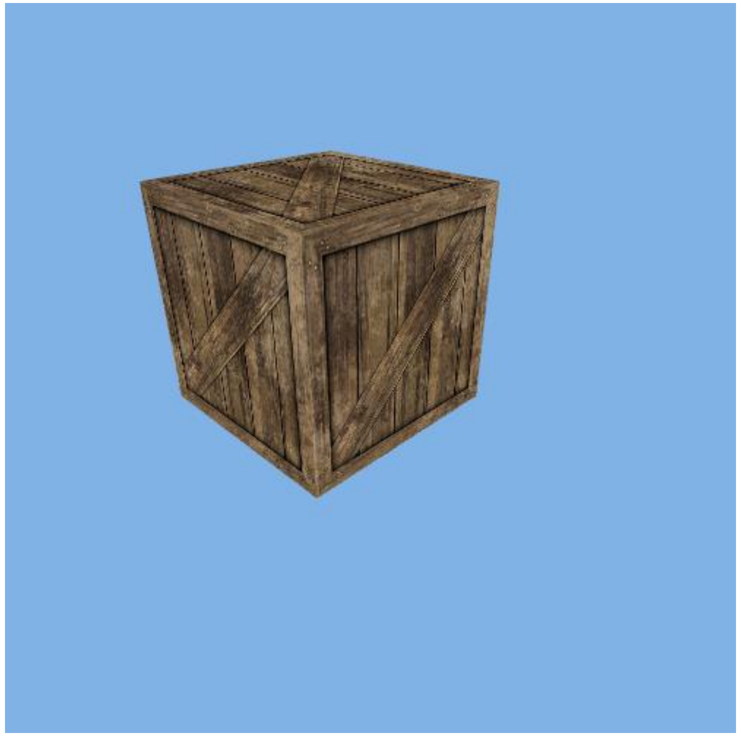




Texture Mapping

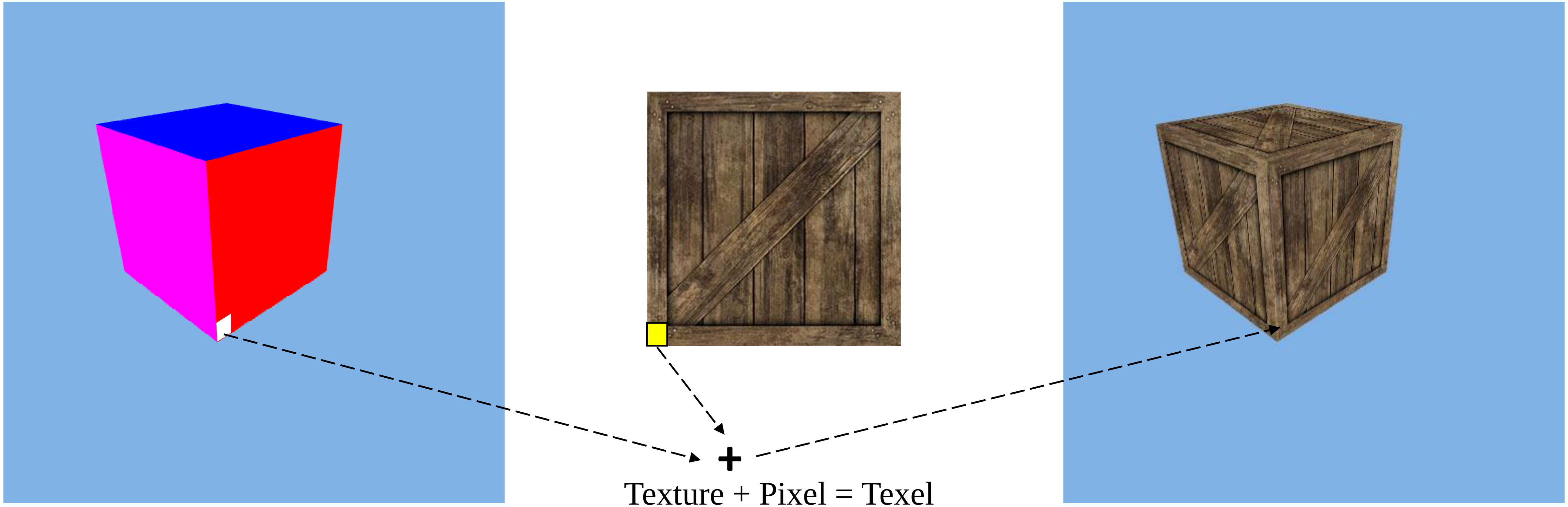






Texture Mapping

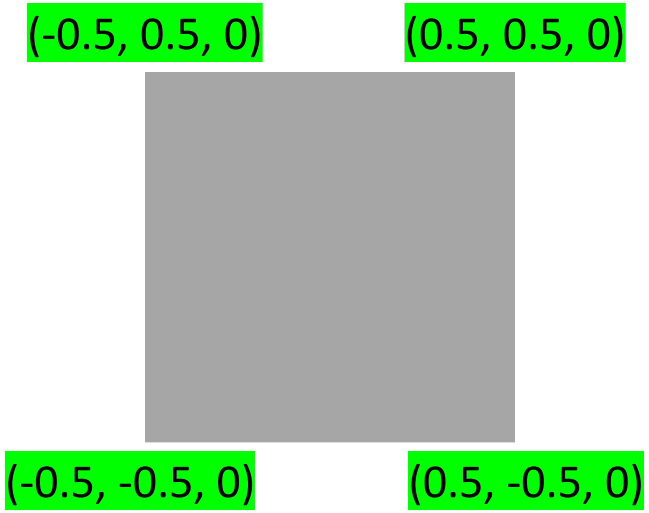
Each point on the surface has to correspond to a point in the texture.

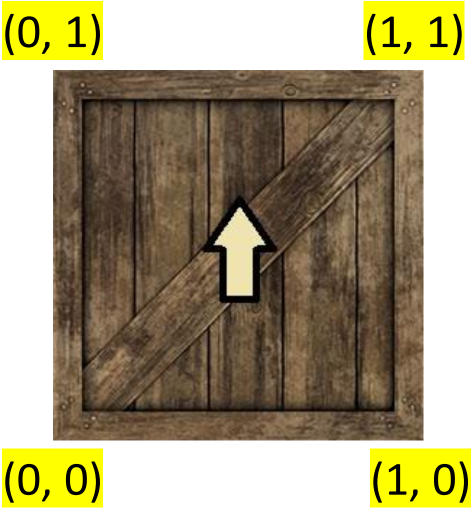


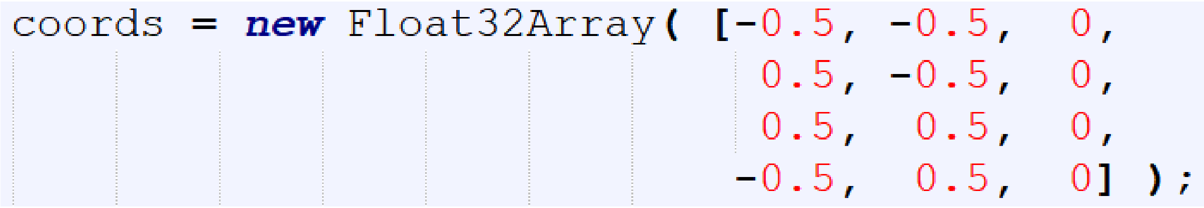
Texture + Pixel = Texel

Source: http://math.hws.edu/graphicsbook/c4/s3.html

Texture Lookup

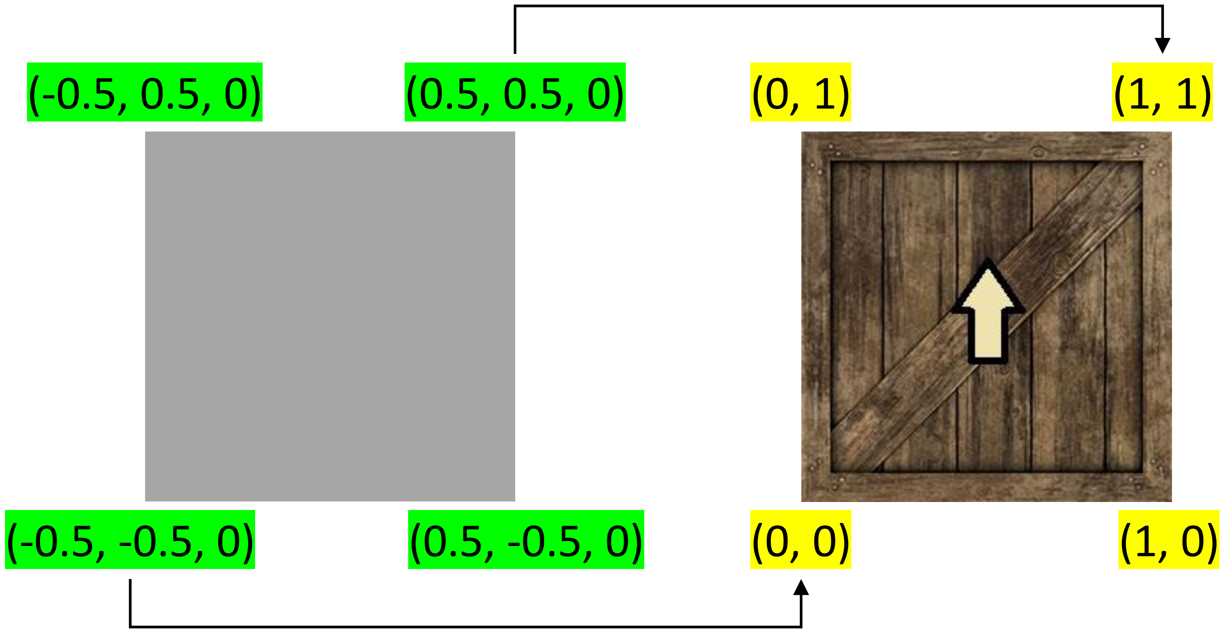






```
coords = new Float32Array( [-0.5, -0.5,  0,
                             0.5, -0.5,  0,
                             0.5,  0.5,  0,
                            -0.5,  0.5,  0] );
```

Texture Lookup



(-0.5, 0.5, 0)

(0.5, 0.5, 0)

(0, 1)

(1, 1)

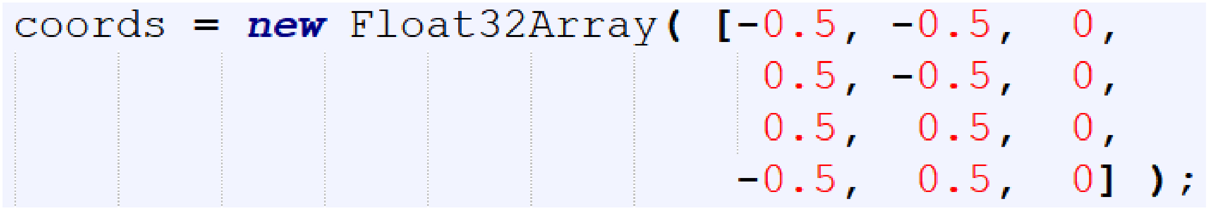
(-0.5, -0.5, 0)

(0.5, -0.5, 0)

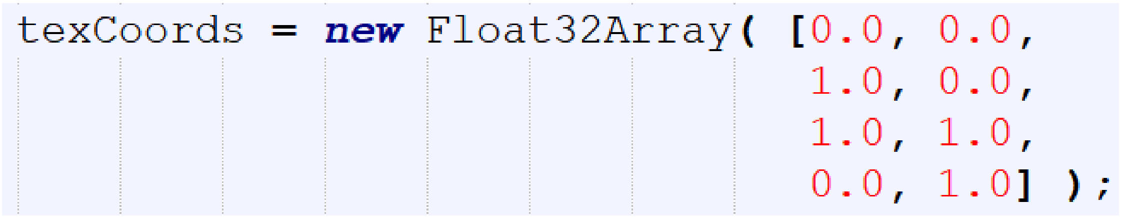
(0, 0)

(1, 0)

XY	UV
(-0.5, -0.5, 0)	(0, 0)
(0.5, -0.5, 0)	(1, 0)
(0.5, 0.5, 0)	(1, 1)
(-0.5, 0.5, 0)	(0, 1)

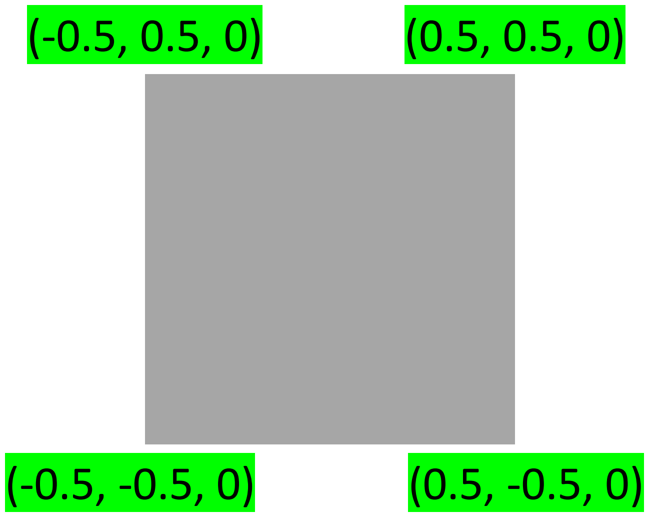


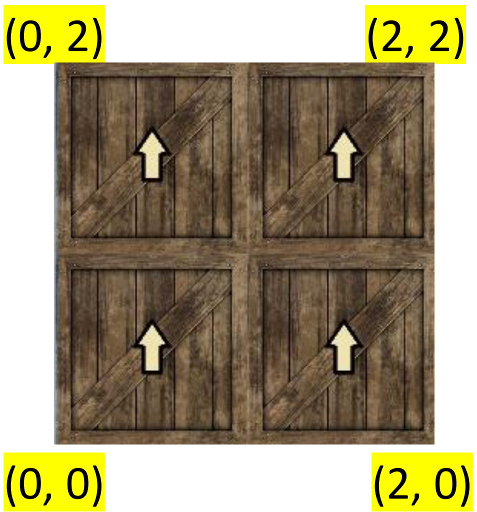
```
coords = new Float32Array( [-0.5, -0.5,  0,
                             0.5, -0.5,  0,
                             0.5,  0.5,  0,
                            -0.5,  0.5,  0] );
```



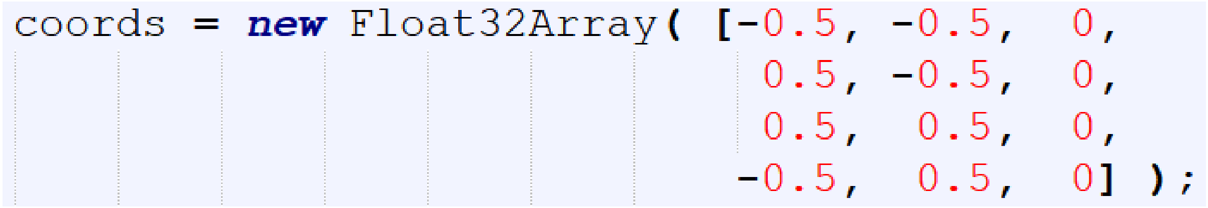
```
texCoords = new Float32Array( [0.0, 0.0,
                               1.0, 0.0,
                               1.0, 1.0,
                               0.0, 1.0] );
```

Texture Lookup

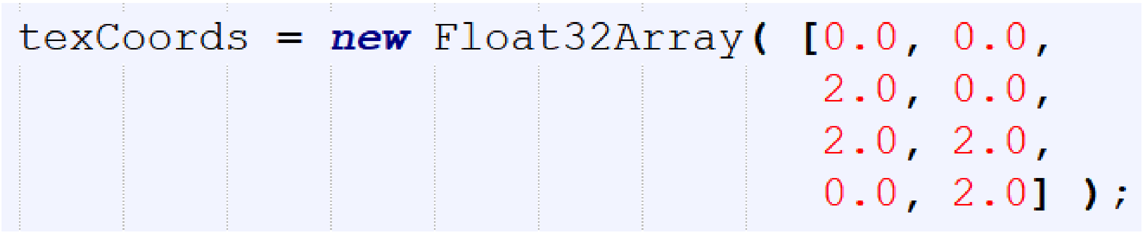




XY	UV
(-0.5, -0.5, 0)	(0, 0)
(0.5, -0.5, 0)	(2, 0)
(0.5, 0.5, 0)	(2, 2)
(-0.5, 0.5, 0)	(0, 2)

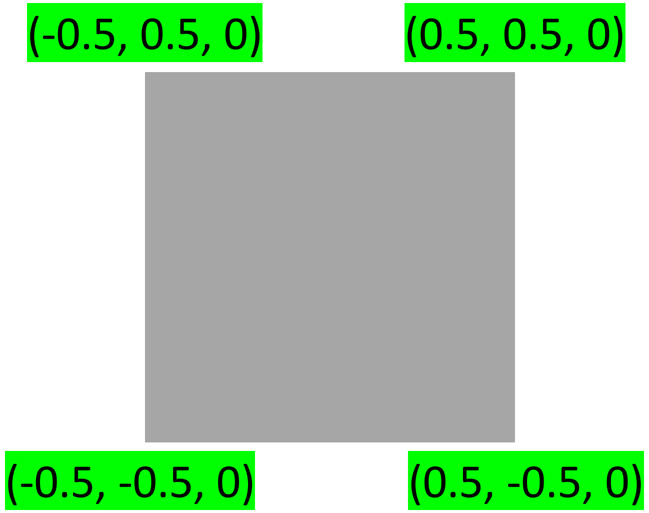


```
coords = new Float32Array( [-0.5, -0.5,  0,
                             0.5, -0.5,  0,
                             0.5,  0.5,  0,
                            -0.5,  0.5,  0] );
```



```
texCoords = new Float32Array( [0.0, 0.0,
                               2.0, 0.0,
                               2.0, 2.0,
                               0.0, 2.0] );
```

Texture Lookup

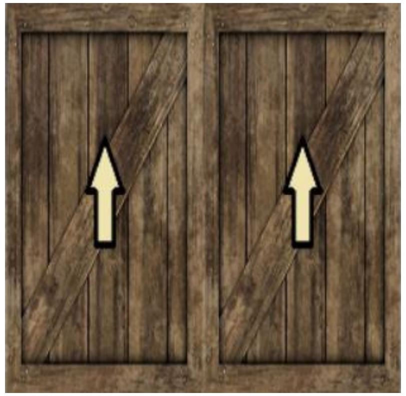


(-0.5, 0.5, 0)

(0.5, 0.5, 0)

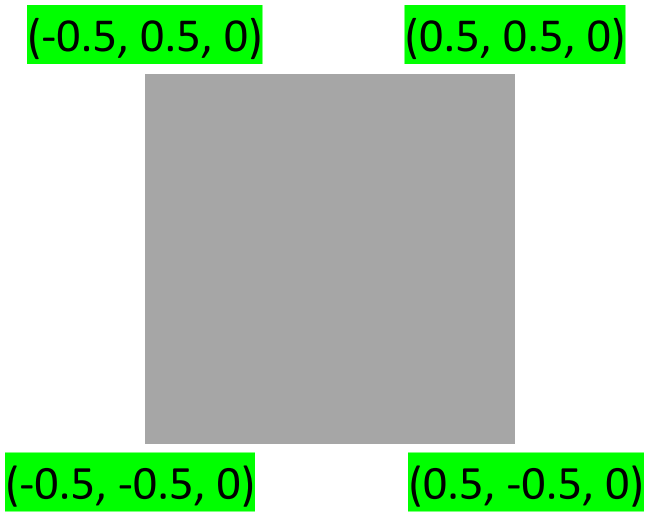
(-0.5, -0.5, 0)

(0.5, -0.5, 0)



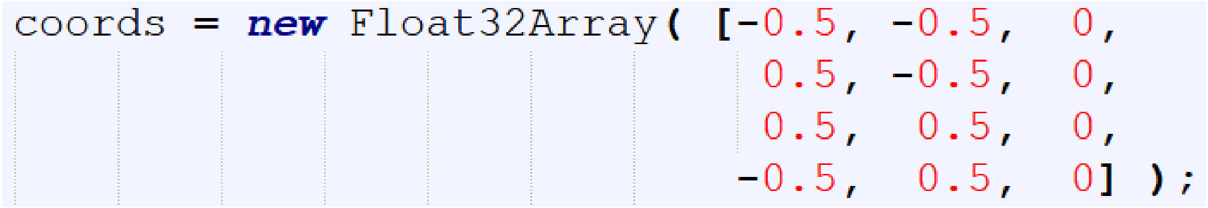
XY	UV
(-0.5, -0.5, 0)	(0, 0)
(0.5, -0.5, 0)	(,)
(0.5, 0.5, 0)	(,)
(-0.5, 0.5, 0)	(,)

Texture Lookup

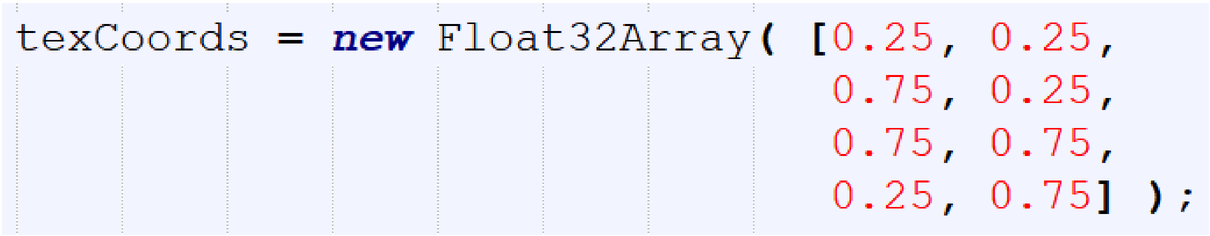


?

XY	UV
(-0.5, -0.5, 0)	(0.25, 0.25)
(0.5, -0.5, 0)	(0.75, 0.25)
(0.5, 0.5, 0)	(0.75, 0.75)
(-0.5, 0.5, 0)	(0.25, 0.75)

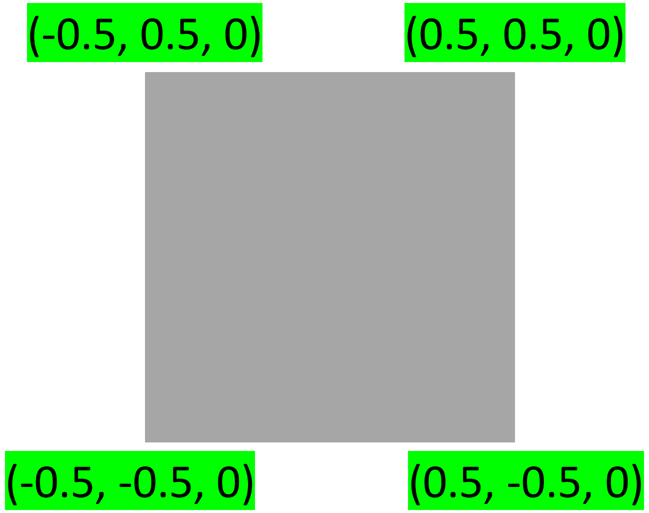


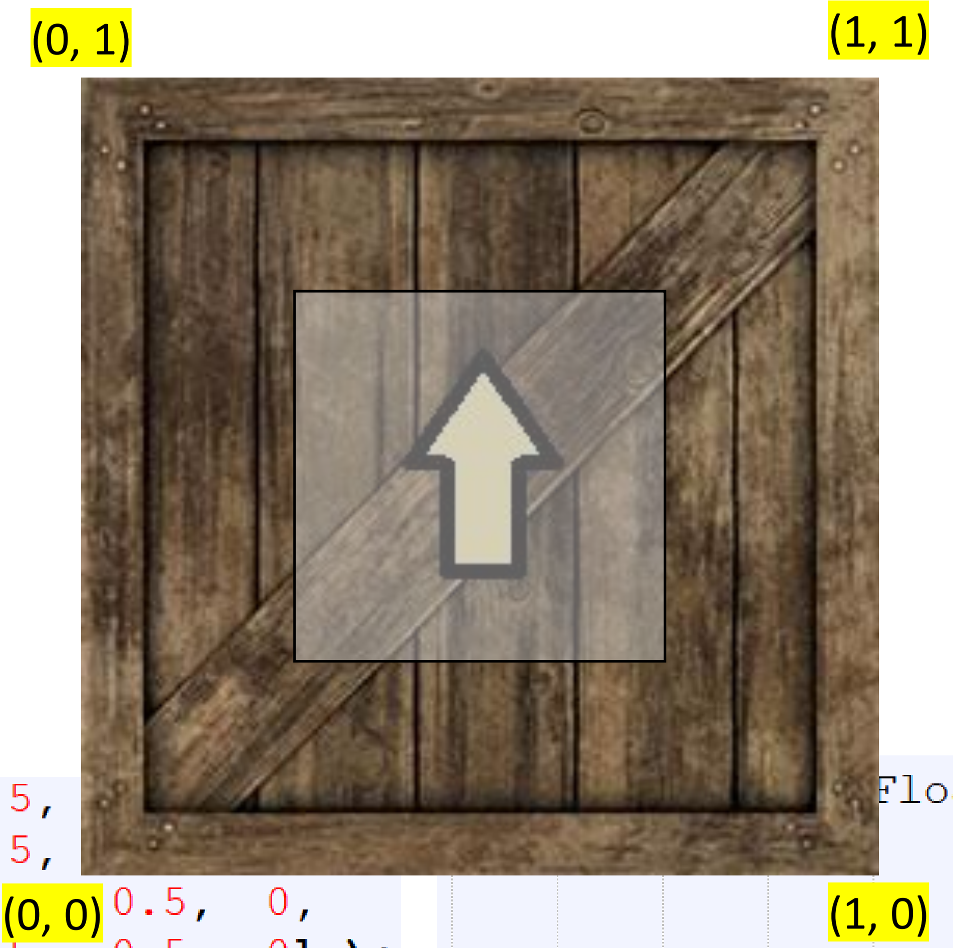
```
coords = new Float32Array( [-0.5, -0.5,  0,
                             0.5, -0.5,  0,
                             0.5,  0.5,  0,
                            -0.5,  0.5,  0] );
```



```
texCoords = new Float32Array( [0.25, 0.25,
                               0.75, 0.25,
                               0.75, 0.75,
                               0.25, 0.75] );
```

Texture Lookup



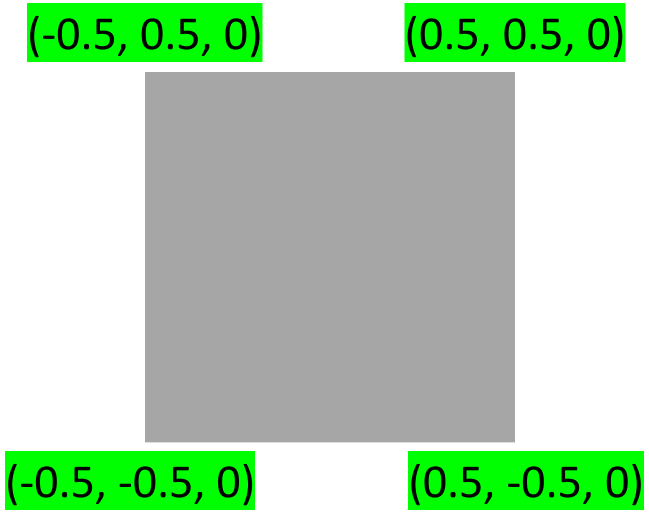


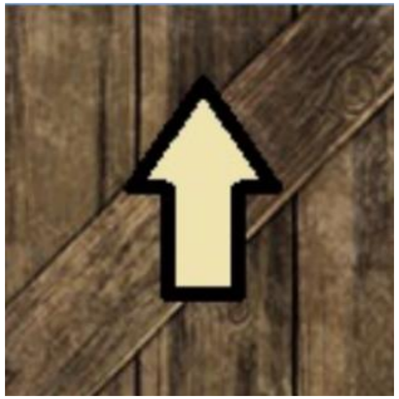
XY	UV
(-0.5, -0.5, 0)	(0.25, 0.25)
(0.5, -0.5, 0)	(0.75, 0.25)
(0.5, 0.5, 0)	(0.75, 0.75)
(-0.5, 0.5, 0)	(0.25, 0.75)

```
coords = new Float32Array( [-0.5,
                             0.5,
                             0.      0.5,   0,
                            -0.5,    0.5,   0] );
```

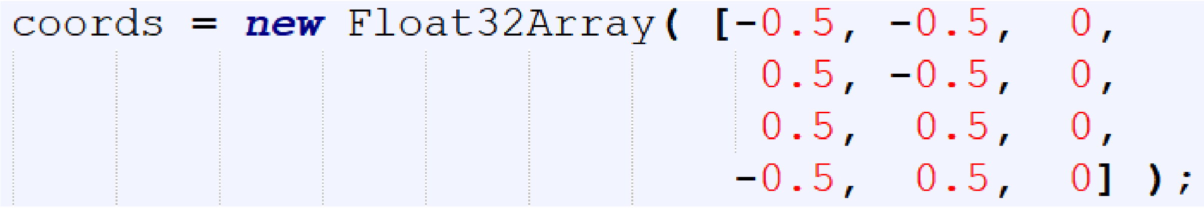
```
Float32Array( [0.25, 0.25,
               0.75, 0.25,
               0.75, 0.75,
               0.25, 0.75] );
```

Texture Lookup

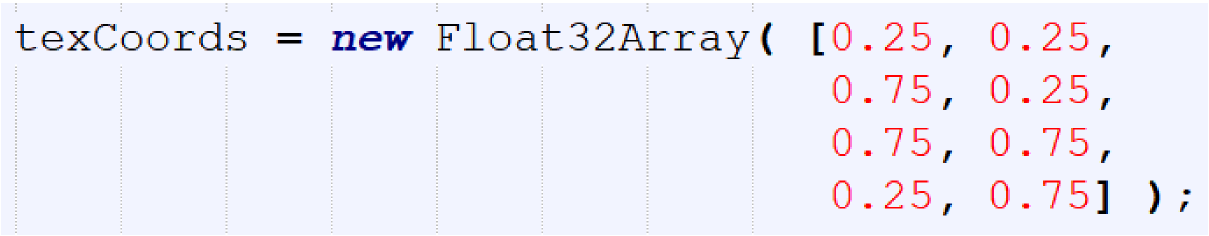




XY	UV
(-0.5, -0.5, 0)	(0.25, 0.25)
(0.5, -0.5, 0)	(0.75, 0.25)
(0.5, 0.5, 0)	(0.75, 0.75)
(-0.5, 0.5, 0)	(0.25, 0.75)



```
coords = new Float32Array( [-0.5, -0.5,  0,
                              0.5, -0.5,  0,
                              0.5,  0.5,  0,
                             -0.5,  0.5,  0] );
```



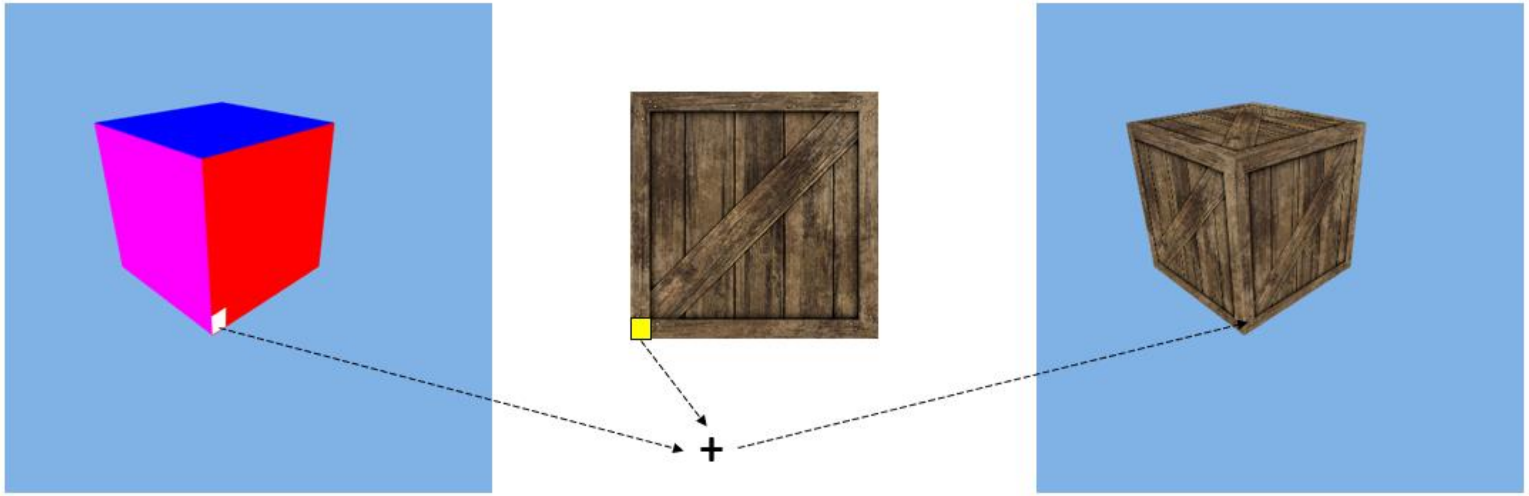
```
texCoords = new Float32Array( [0.25, 0.25,
                               0.75, 0.25,
                               0.75, 0.75,
                               0.25, 0.75] );
```

Texture Object

- Texture object is a data structure that contains the color data for an image texture

Sampling and TU

- Sampling is the process of computing a color from an image texture and texture coordinates
- A texture unit (TU) is a hardware component in a GPU that does sampling.
- There are multiple Tus.



Workflow

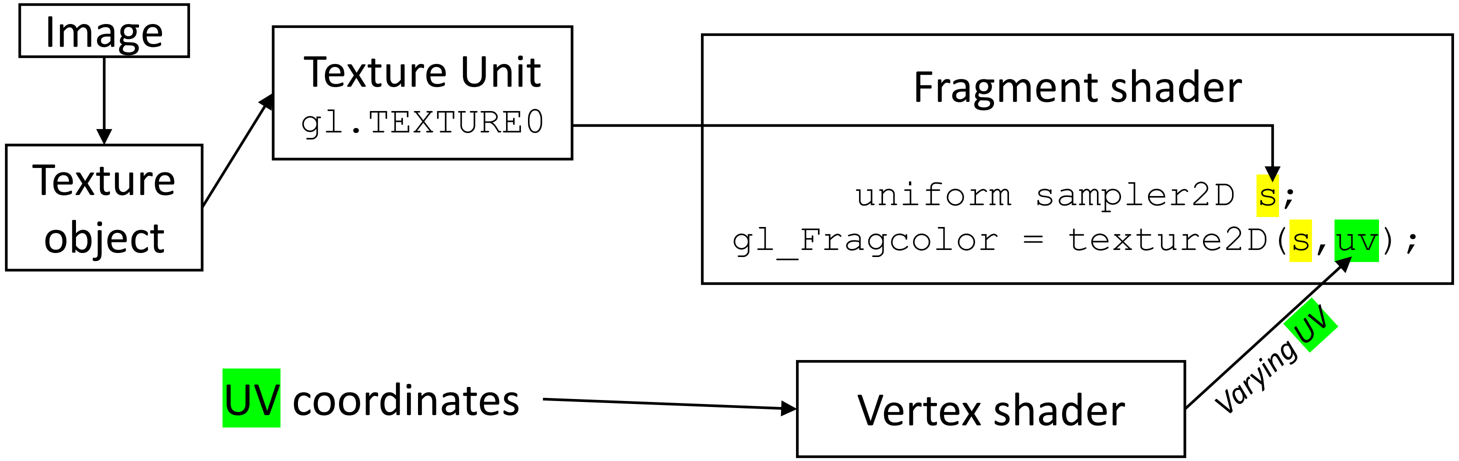
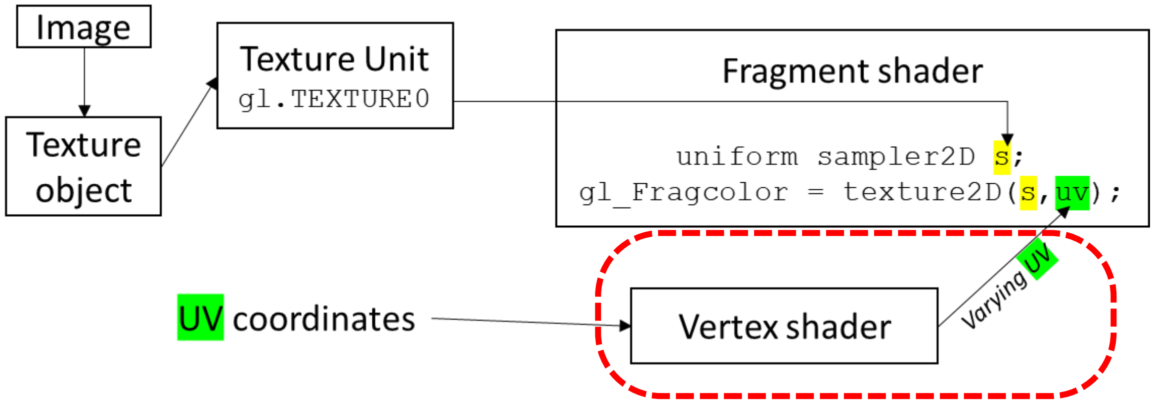
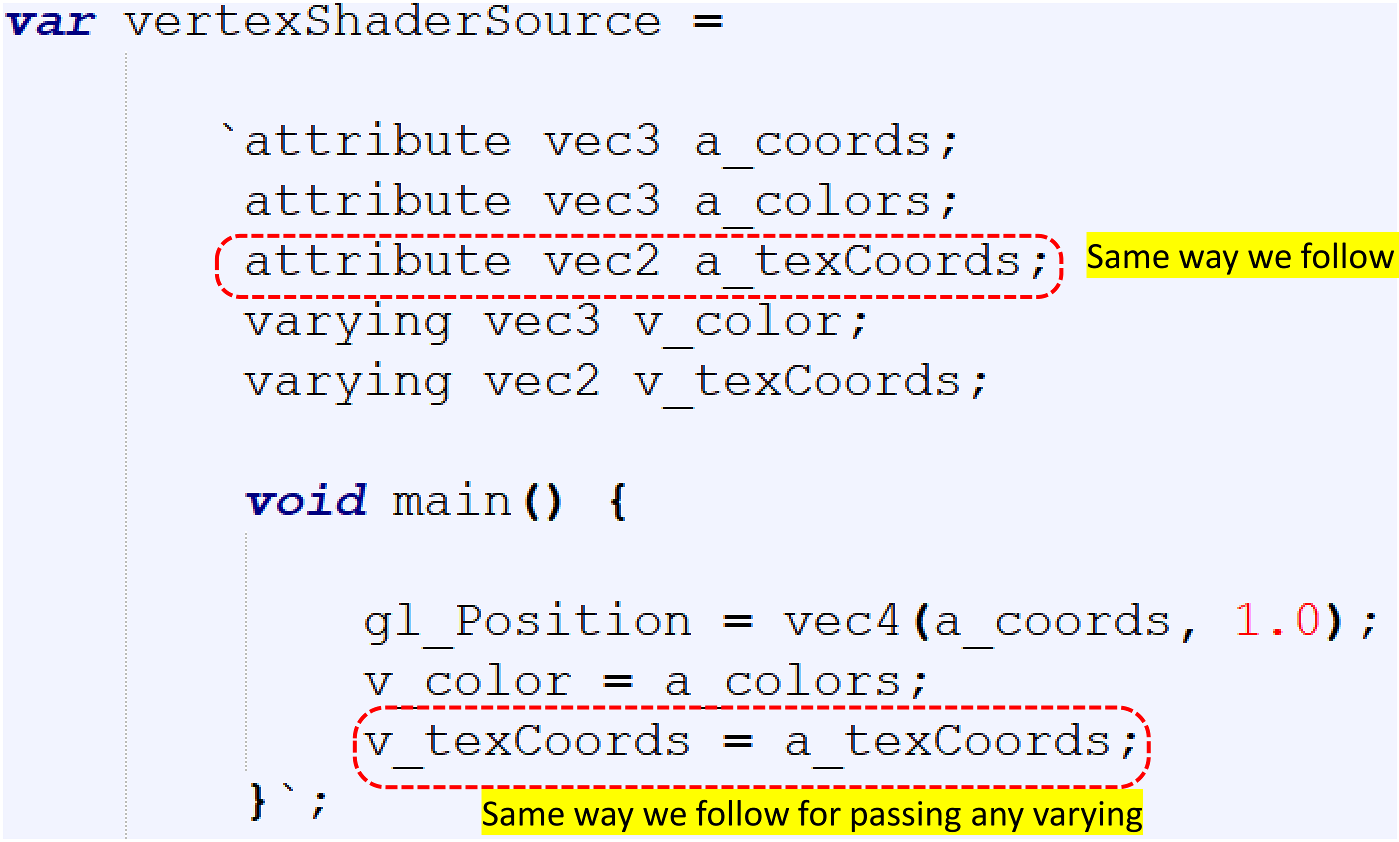


Image
Texture object
Texture Unit
gl.TEXTURE0
Fragment shader
uniform sampler2D s;
gl_Fragcolor = texture2D(s,uv);
UV coordinates
Vertex shader
Varying UV

Passing Texture Coordinates --> V.S --> F.S



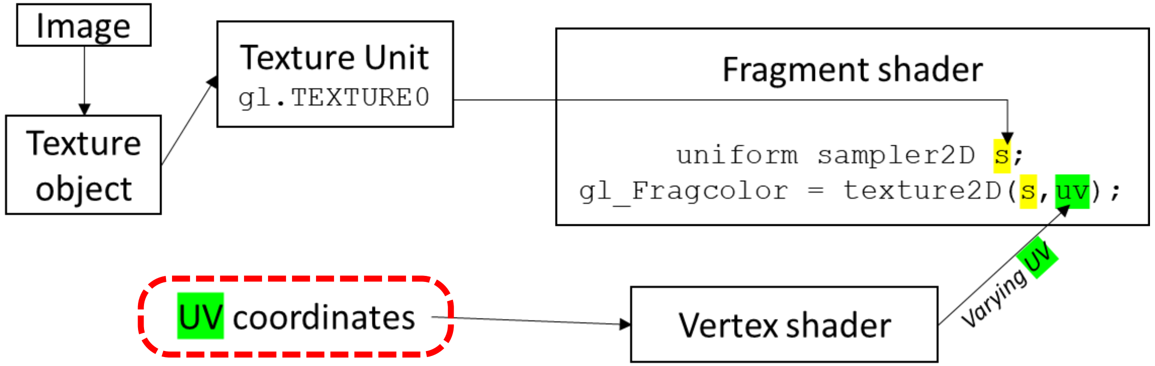


```
var vertexShaderSource =

        `attribute vec3 a_coords;
         attribute vec3 a_colors;
         attribute vec2 a_texCoords;
         varying vec3 v_color;
         varying vec2 v_texCoords;

         void main() {

             gl_Position = vec4(a_coords, 1.0);
             v_color = a_colors;
             v_texCoords = a_texCoords;
         }`;
```

Same way we follow to receive att. inside V.S

Same way we follow for passing any varying

Defining Texture Lookup



Image

Texture object

Texture Unit
gl.TEXTURE0

Fragment shader

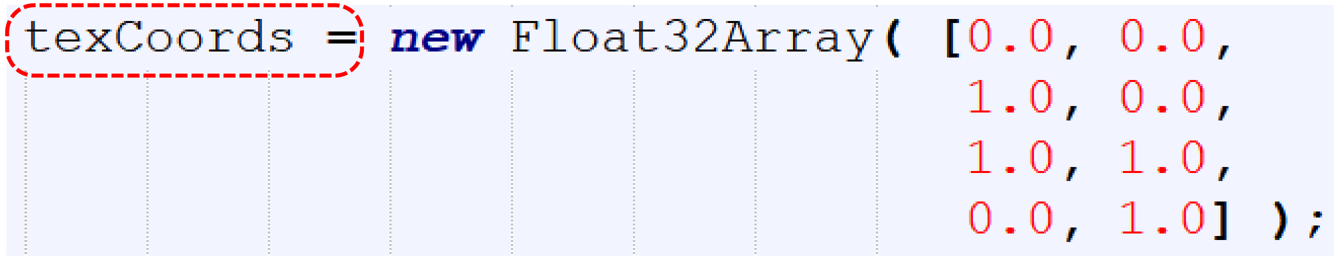
```
uniform sampler2D s;
gl_Fragcolor = texture2D(s,uv);
```

UV coordinates

Vertex shader

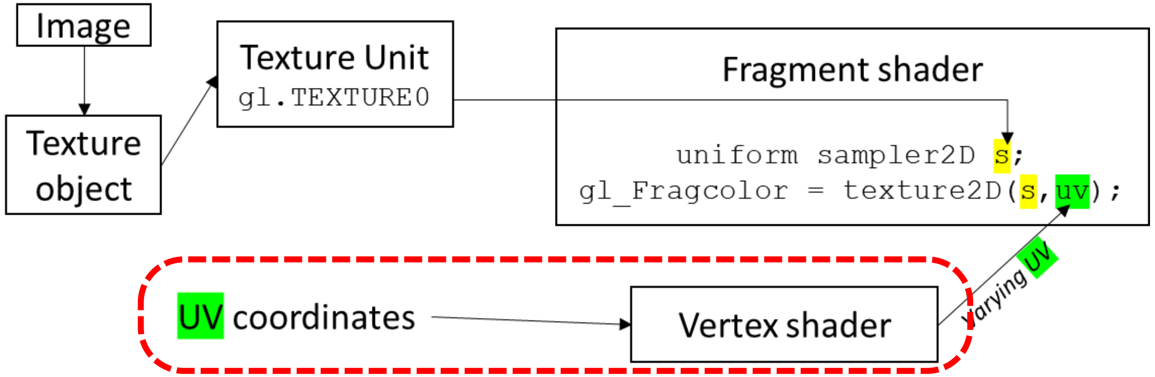
Varying UV

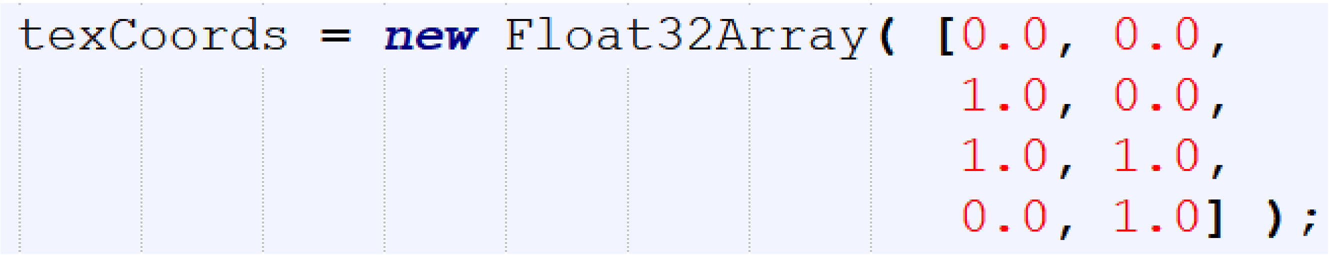
UV coordinate



```
texCoords = new Float32Array( [0.0, 0.0,
                               1.0, 0.0,
                               1.0, 1.0,
                               0.0, 1.0] );
```

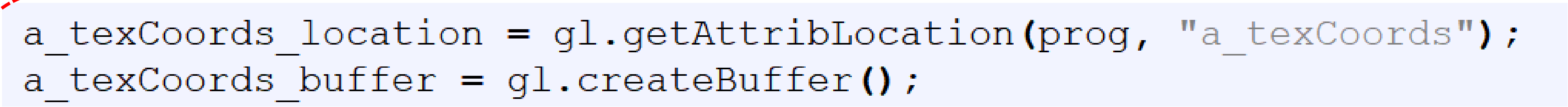
Defining Texture Lookup



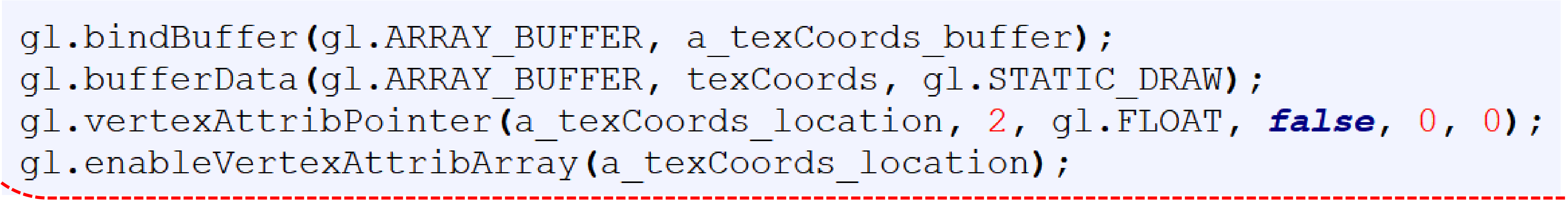


```
texCoords = new Float32Array( [0.0, 0.0,
                                1.0, 0.0,
                                1.0, 1.0,
                                0.0, 1.0] );
```

Same way we follow for passing any attribute data to V.S

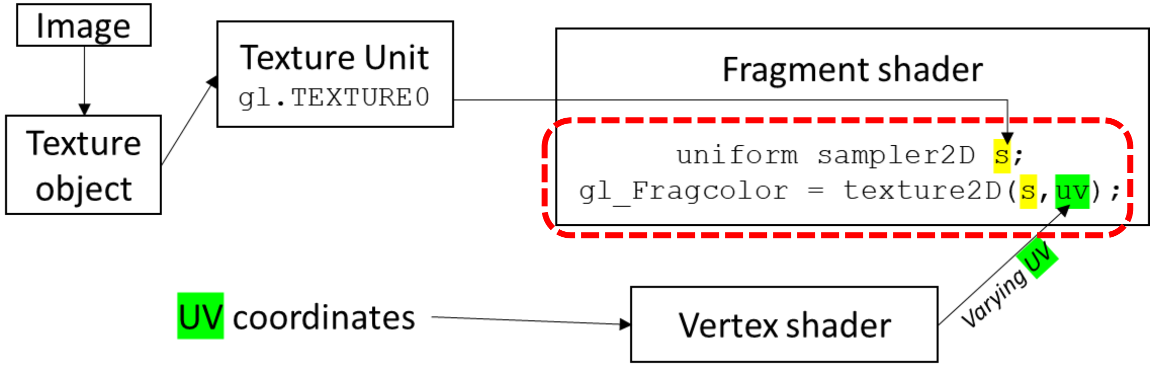


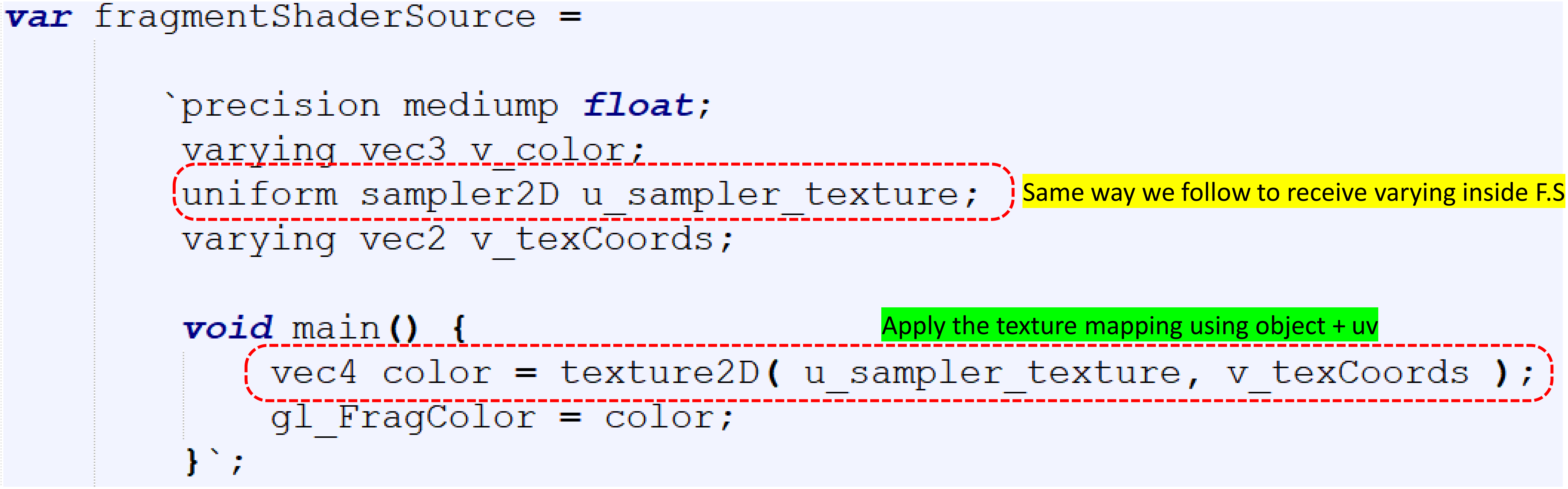
```
a_texCoords_location = gl.getAttribLocation(prog, "a_texCoords");
a_texCoords_buffer = gl.createBuffer();
```



```
gl.bindBuffer(gl.ARRAY_BUFFER, a_texCoords_buffer);
gl.bufferData(gl.ARRAY_BUFFER, texCoords, gl.STATIC_DRAW);
gl.vertexAttribPointer(a_texCoords_location, 2, gl.FLOAT, false, 0, 0);
gl.enableVertexAttribArray(a_texCoords_location);
```

Sampler in F.S



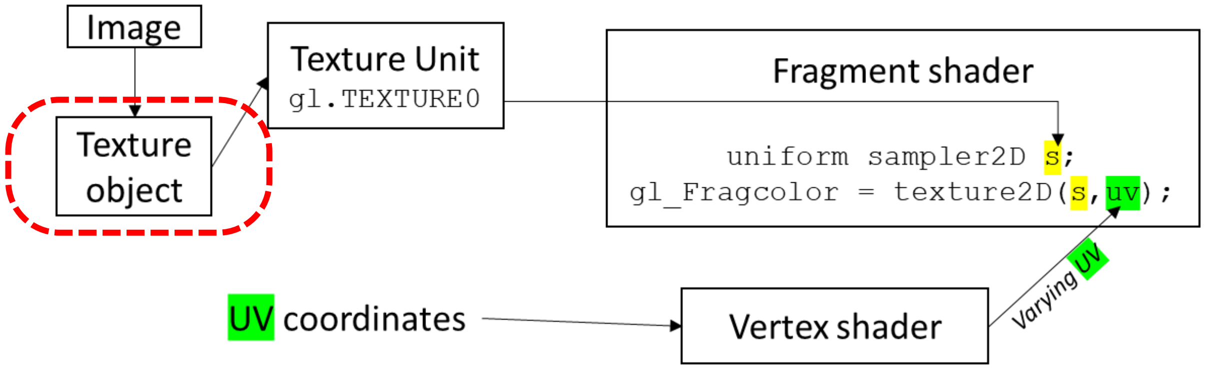


```
var fragmentShaderSource =

        `precision mediump float;
        varying vec3 v_color;
        uniform sampler2D u_sampler_texture;
        varying vec2 v_texCoords;

        void main() {
            vec4 color = texture2D( u_sampler_texture, v_texCoords );
            gl_FragColor = color;
        }`;
```

Same way we follow to receive varying inside F.S

Apply the texture mapping using object + uv

Texture object --> TU



```
uniform sampler2D s;
gl_Fragcolor = texture2D(s,uv);
```

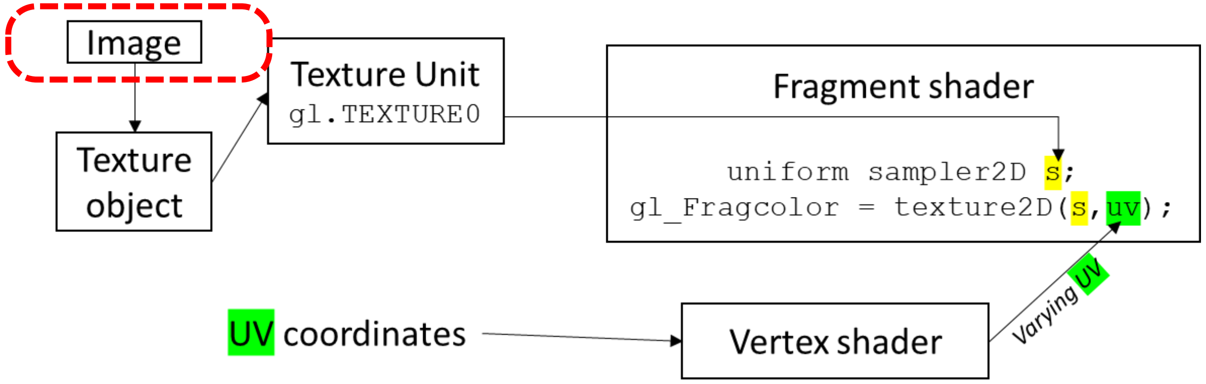
```
textureObject = gl.createTexture();
```

Allocates some memory for the texture object.

```
<img id="doorimage" src="crate2.jpg" width="0" height="0"></img>
```

```
loadTexture(textureObject, "doorimage");
```

Loading the Image



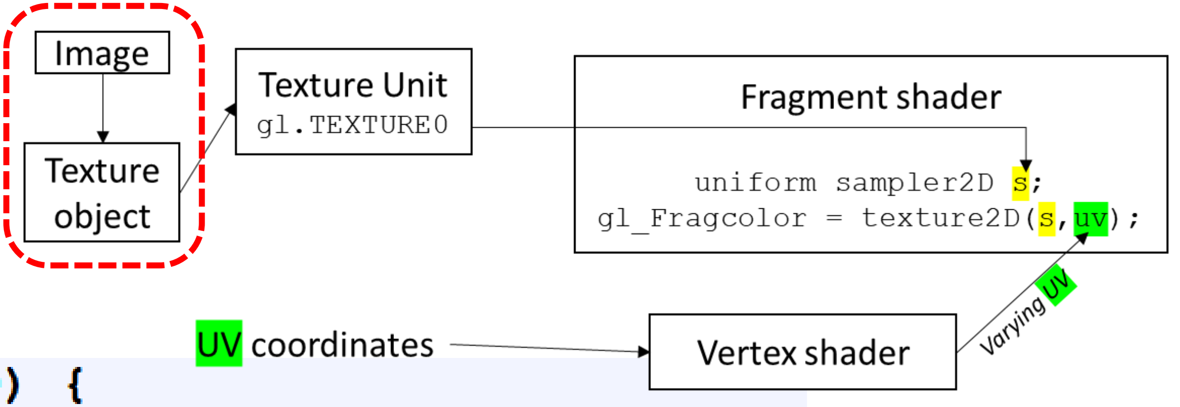
```
textureObject = gl.createTexture();
```

Loading the image

```
<img id="doorimage" src="crate2.jpg" width="0" height="0"></img>
loadTexture(textureObject, "doorimage");
```

Image → Texture Object

Image --> Texture Obj

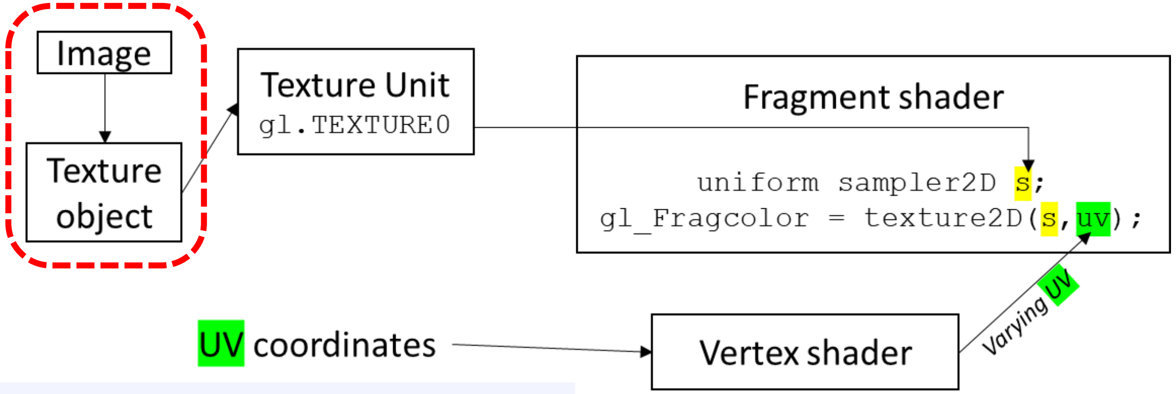


```
function loadTexture(textureObject, imageID) {

    gl.bindTexture(gl.TEXTURE_2D, textureObject); Using the memory

    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR);set texture parameters
                                               minification filter
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER, gl.LINEAR);set texture parameters
                                               magnification filter
    gl.pixelStorei(gl.UNPACK_FLIP_Y_WEBGL, 1);

    gl.texImage2D(gl.TEXTURE_2D, ordinary
                  level 0,
    texture object format gl.RGBA,
    original image format gl.RGBA,
      bytes for the color gl.UNSIGNED_BYTE,
                 source document.getElementById(imageID));
}
```

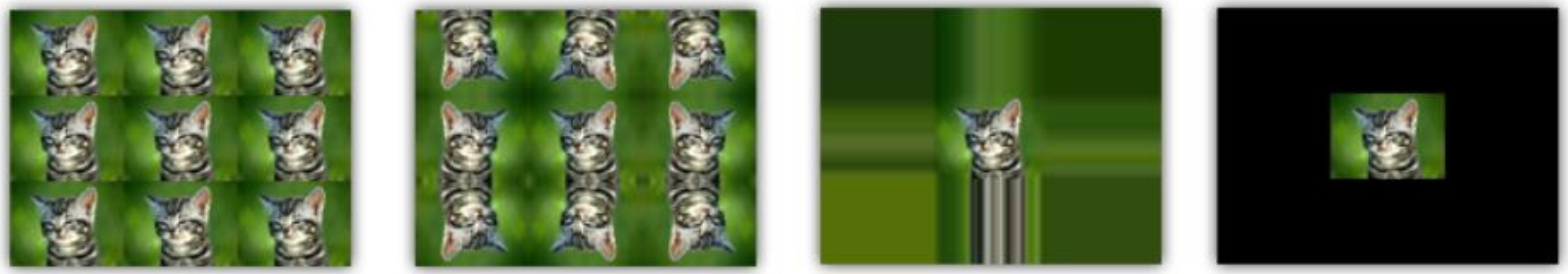
Learn more: https://webglfundamentals.org/webgl/lessons/webgl-3d-textures.html

More parameters

[self study]



```
function loadTexture(textureObject, imageID) {
    gl.bindTexture(gl.TEXTURE_2D, textureObject);

    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_MIN_FILTER, gl.LINEAR);
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER, gl.LINEAR);
```

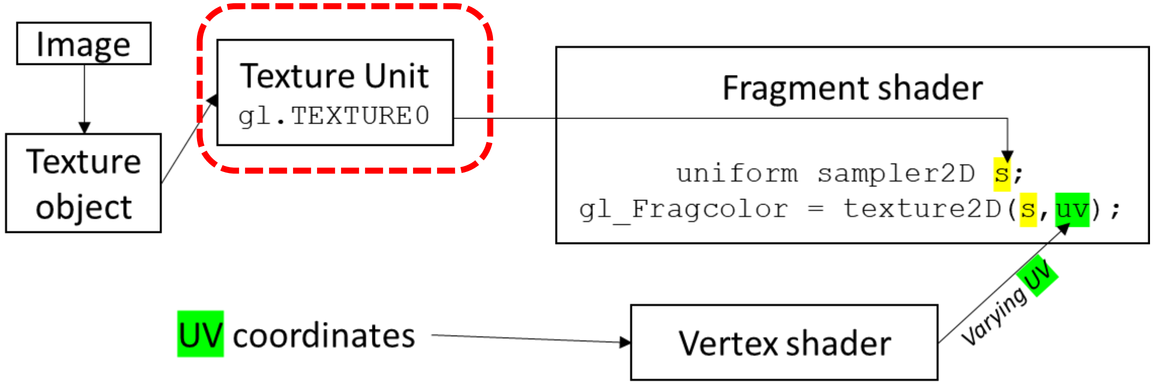


GL_REPEAT GL_MIRRORED_REPEAT GL_CLAMP_TO_EDGE GL_CLAMP_TO_BORDER

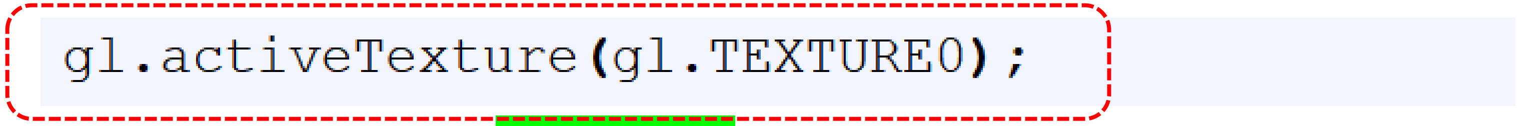
```
}
```

* https://gdbooks.gitbooks.io/legacyopengl/content/Chapter7/TexParams.html

Activating TU



```
uniform sampler2D s;
gl_Fragcolor = texture2D(s,uv);
```



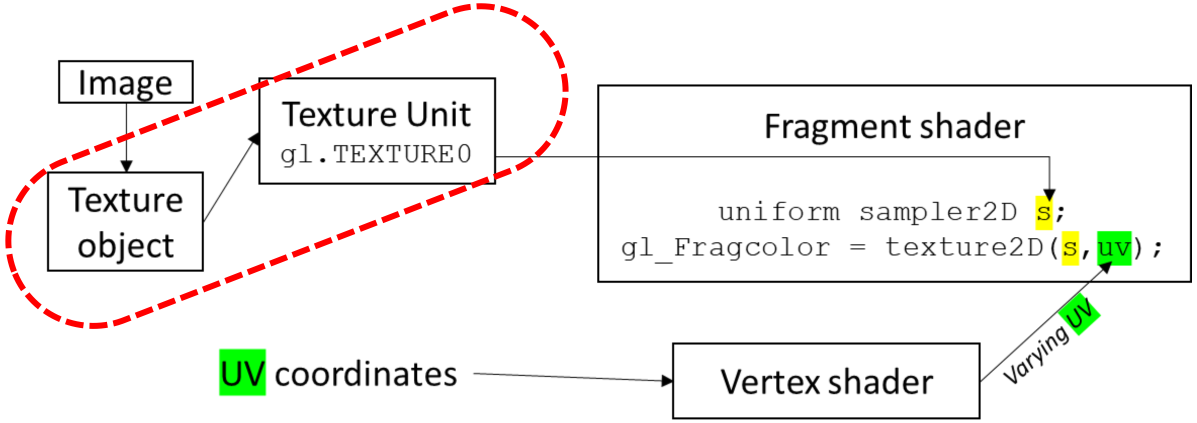
```
gl.activeTexture(gl.TEXTURE0);
```

Activating TU=0

You also need to tell a texture unit to use the texture object. Before you can do that, you need to make the texture unit "active," which is done by calling the function *gl.activeTexture*. The parameter is one of the constants *gl.TEXTURE0*, *gl.TEXTURE1*, *gl.TEXTURE2*, ..., which represent the available texture units.

Source: http://math.hws.edu/graphicsbook/c6/s4.html

Texture Obj --> TU

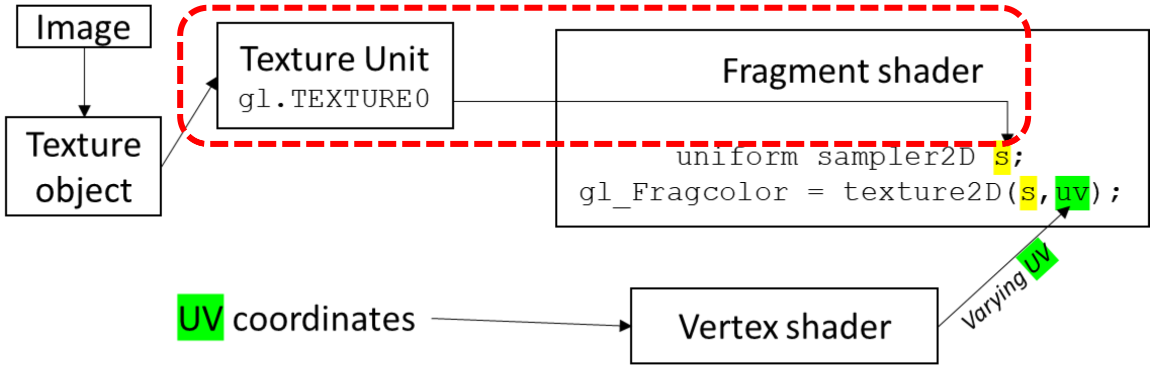


```
gl.activeTexture(gl.TEXTURE0);
```

```
gl.bindTexture(gl.TEXTURE_2D, textureObject);
```

Telling that TEXTURE0 will handle the texture object

Texture Obj --> TU --> Sampler in F.S



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

```
gl.bindTexture(gl.TEXTURE_2D, textureObject);
```

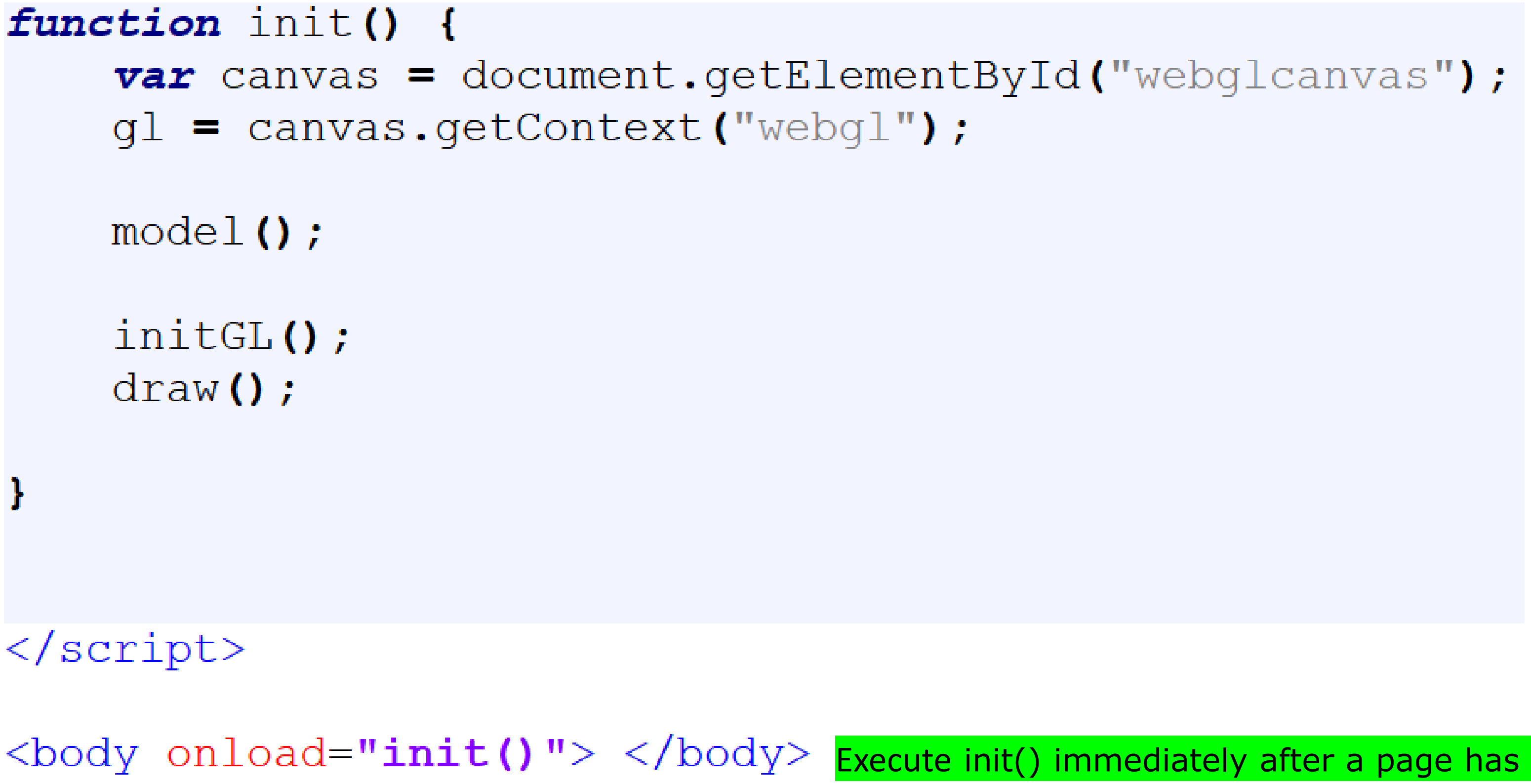
Association from GPU to CPU

```
u_sampler_texture_location = gl.getUniformLocation(prog, "u_sampler_texture");
```

```
gl.uniform1i(u_sampler_texture_location, 0);
```

TU0 → sampler of F.S

Calling *init()*



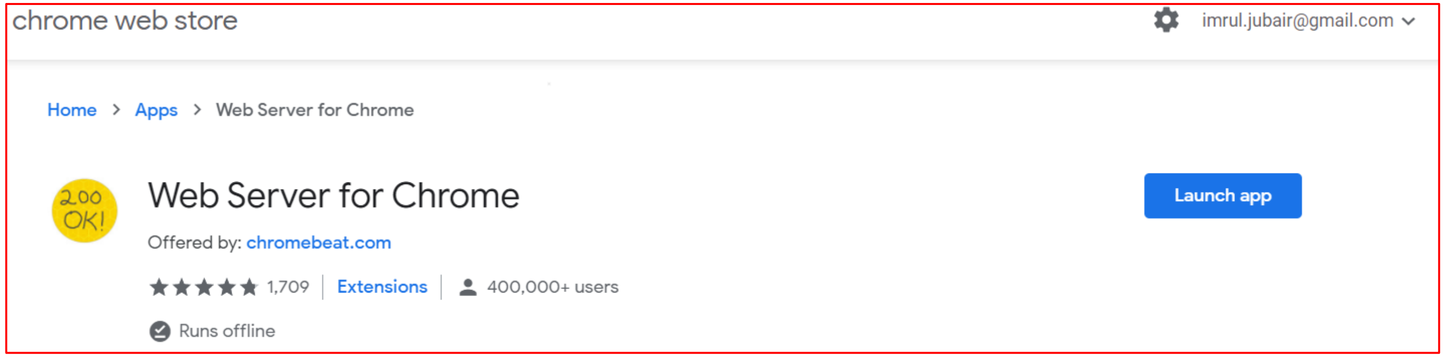
```
function init() {
    var canvas = document.getElementById("webglcanvas");
    gl = canvas.getContext("webgl");

    model();

    initGL();
    draw();

}


</script>

<body onload="init()"> </body>
```

Execute init() immediately after a page has been loaded

Get the materials

https://rb.gy/s2vhg7

We need a Web server



chrome web store — imrul.jubair@gmail.com

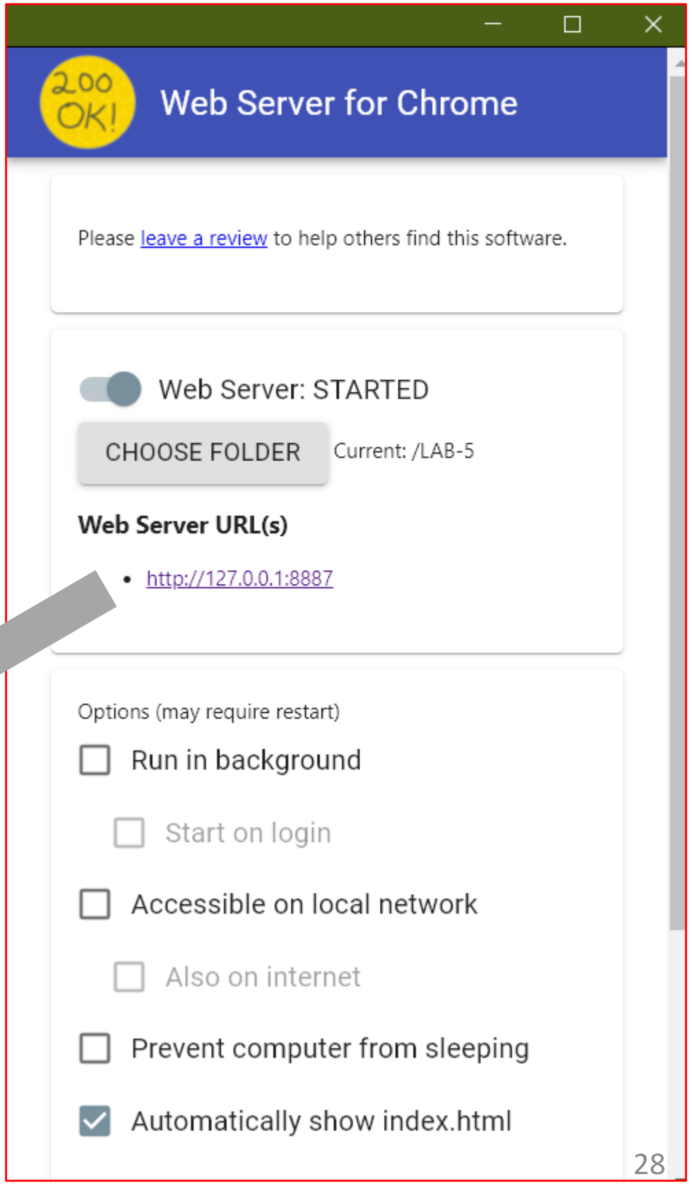
Home > Apps > Web Server for Chrome

Web Server for Chrome

Offered by: chromebeat.com

★★★★★ 1,709 | Extensions | 400,000+ users

Runs offline



Web Server for Chrome

Please leave a review to help others find this software.

Web Server: STARTED

CHOOSE FOLDER Current: /LAB-5

Web Server URL(s)

- http://127.0.0.1:8887

Options (may require restart)

☐ Run in background

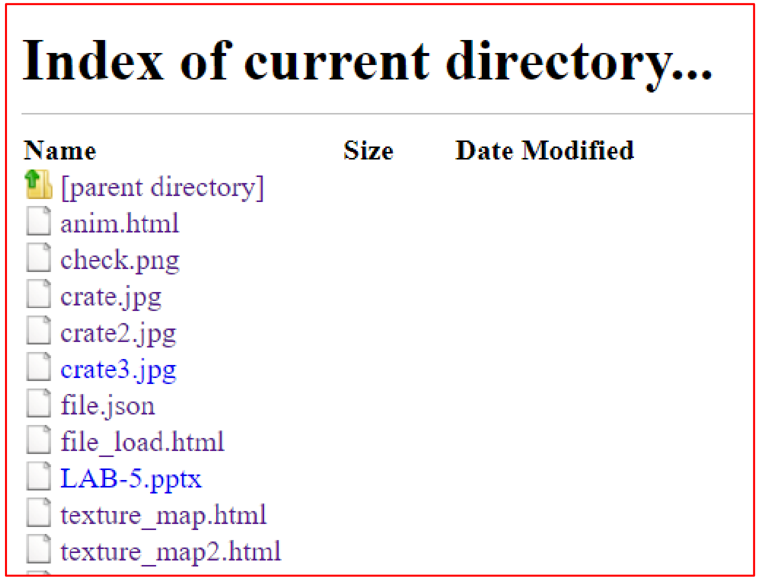
☐ Start on login

☐ Accessible on local network

☐ Also on internet

☐ Prevent computer from sleeping

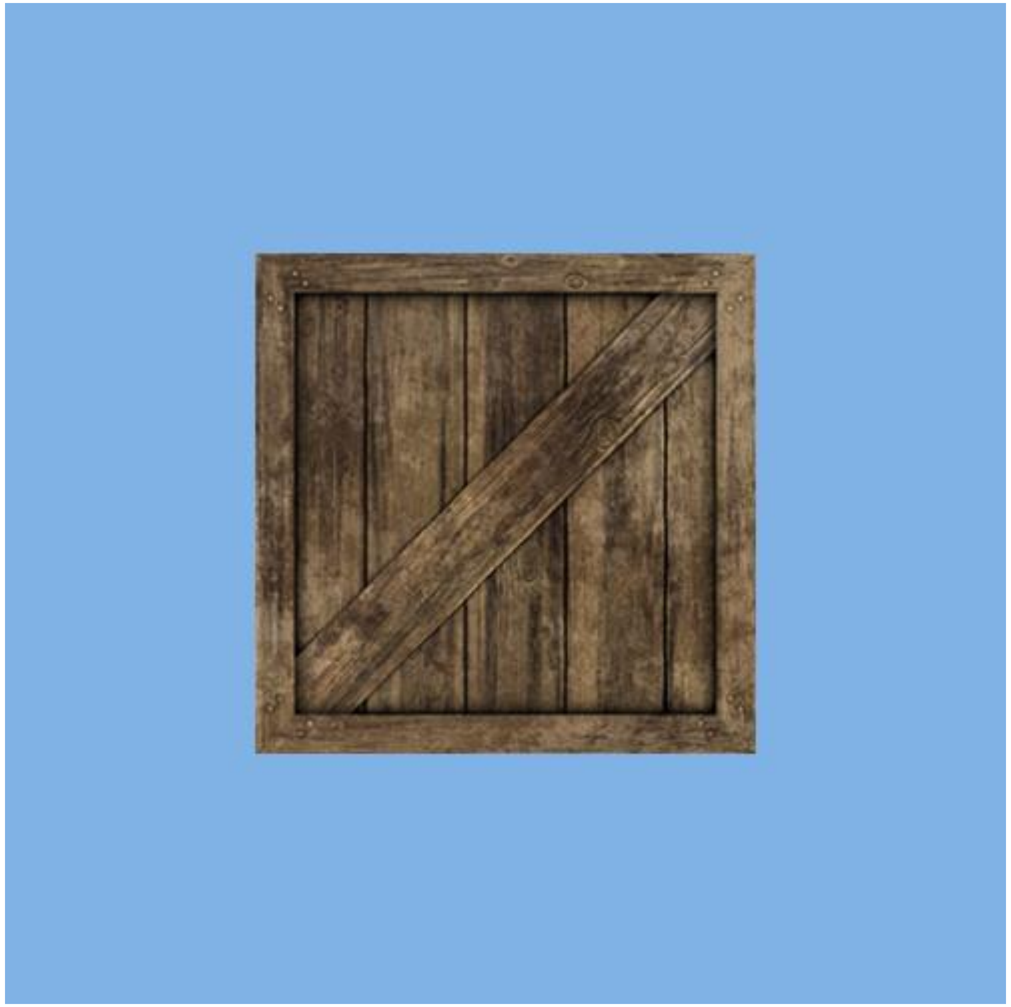
☑ Automatically show index.html



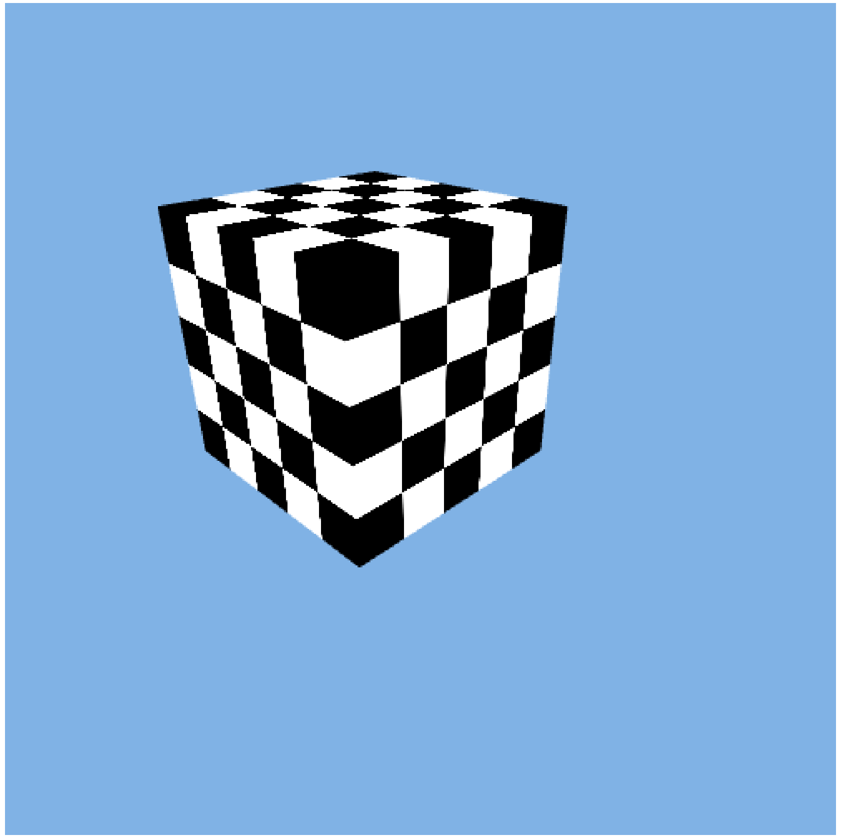
Index of current directory...

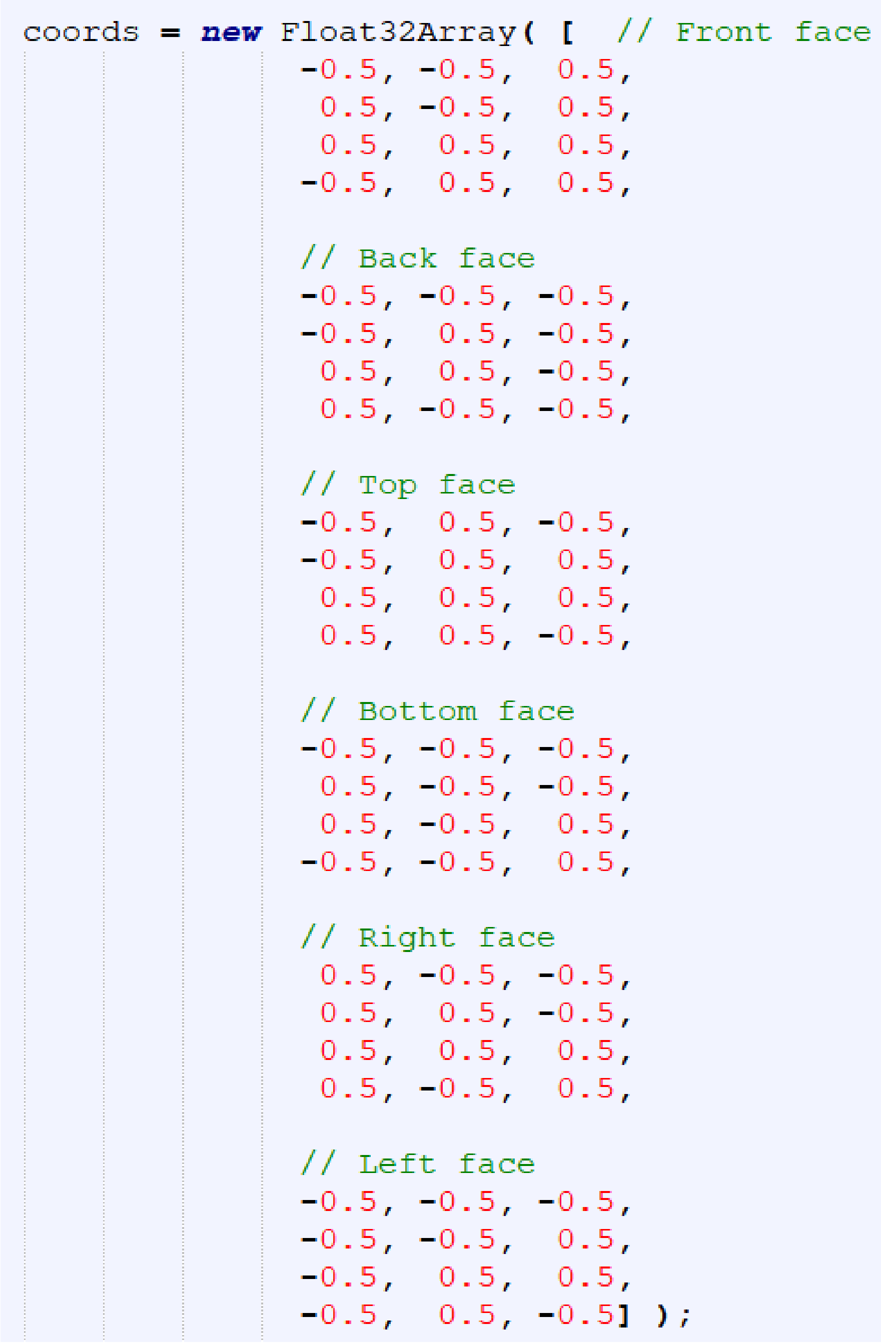
Name	Size	Date Modified
[parent directory]		
anim.html		
check.png		
crate.jpg		
crate2.jpg		
crate3.jpg		
file.json		
file_load.html		
LAB-5.pptx		
texture_map.html		
texture_map2.html		

Result (1)

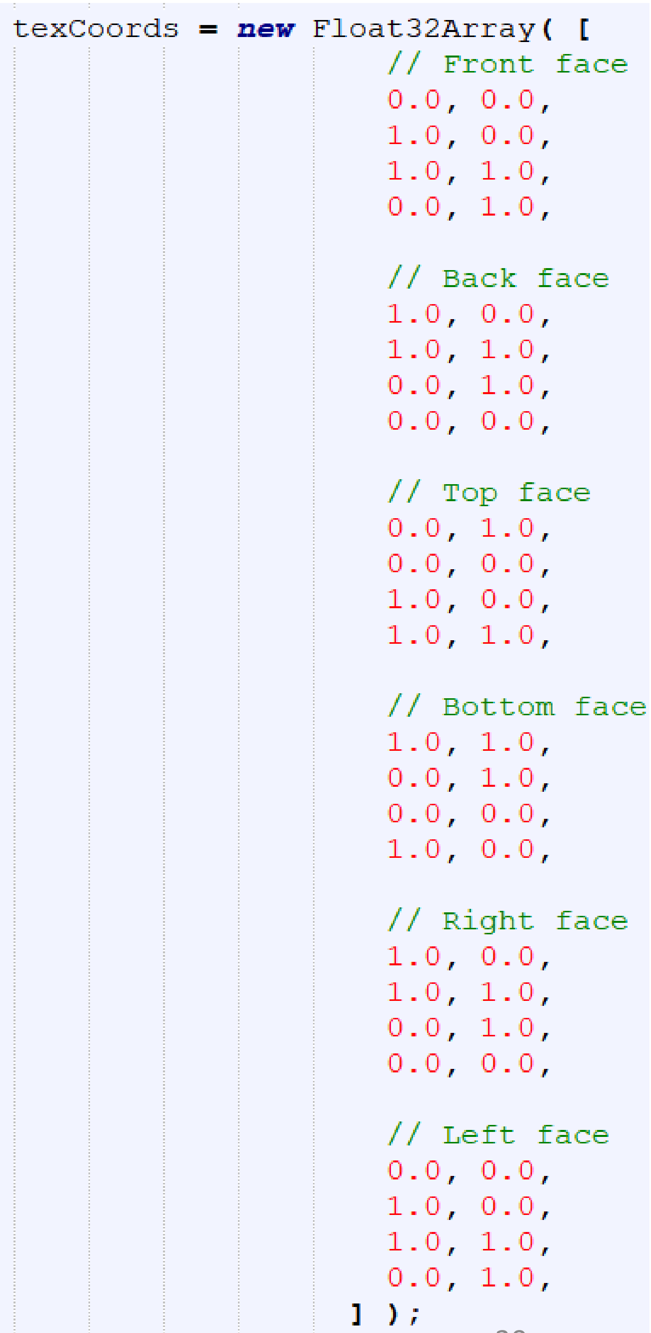


Result (2)



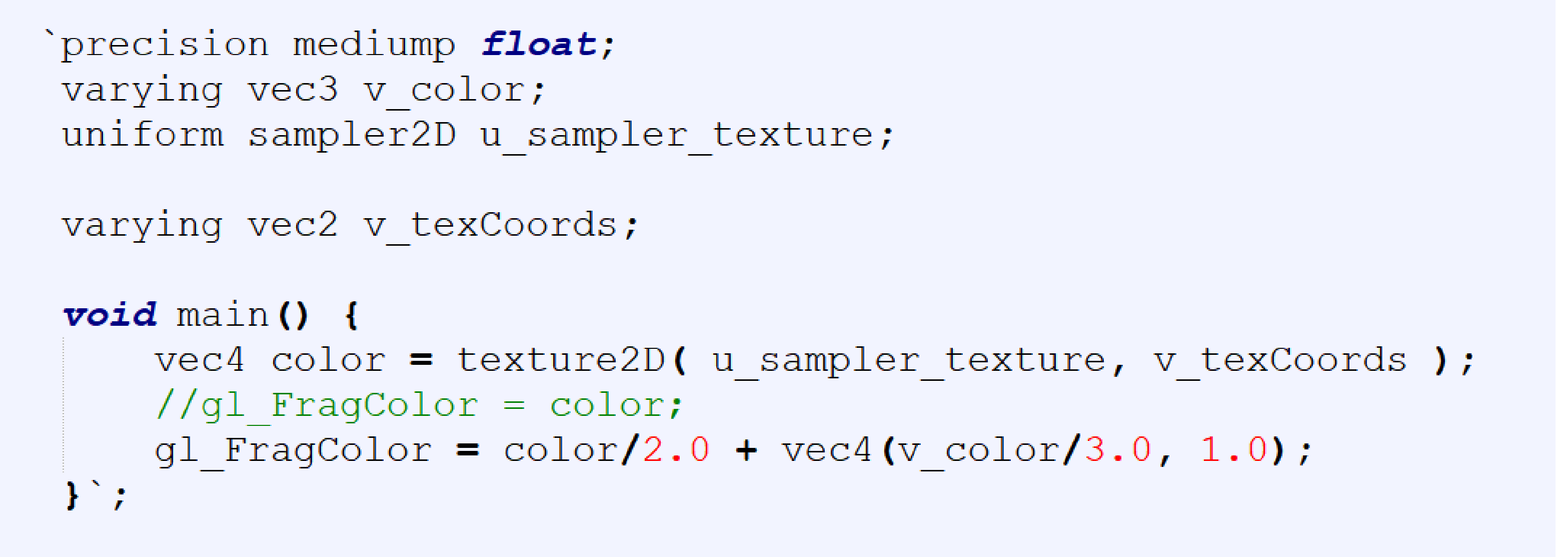


```
coords = new Float32Array( [  // Front face
                            -0.5, -0.5,  0.5,
                             0.5, -0.5,  0.5,
                             0.5,  0.5,  0.5,
                            -0.5,  0.5,  0.5,

                            // Back face
                            -0.5, -0.5, -0.5,
                            -0.5,  0.5, -0.5,
                             0.5,  0.5, -0.5,
                             0.5, -0.5, -0.5,

                            // Top face
                            -0.5,  0.5, -0.5,
                            -0.5,  0.5,  0.5,
                             0.5,  0.5,  0.5,
                             0.5,  0.5, -0.5,

                            // Bottom face
                            -0.5, -0.5, -0.5,
                             0.5, -0.5, -0.5,
                             0.5, -0.5,  0.5,
                            -0.5, -0.5,  0.5,

                            // Right face
                             0.5, -0.5, -0.5,
                             0.5,  0.5, -0.5,
                             0.5,  0.5,  0.5,
                             0.5, -0.5,  0.5,

                            // Left face
                            -0.5, -0.5, -0.5,
                            -0.5, -0.5,  0.5,
                            -0.5,  0.5,  0.5,
                            -0.5,  0.5, -0.5] );
```

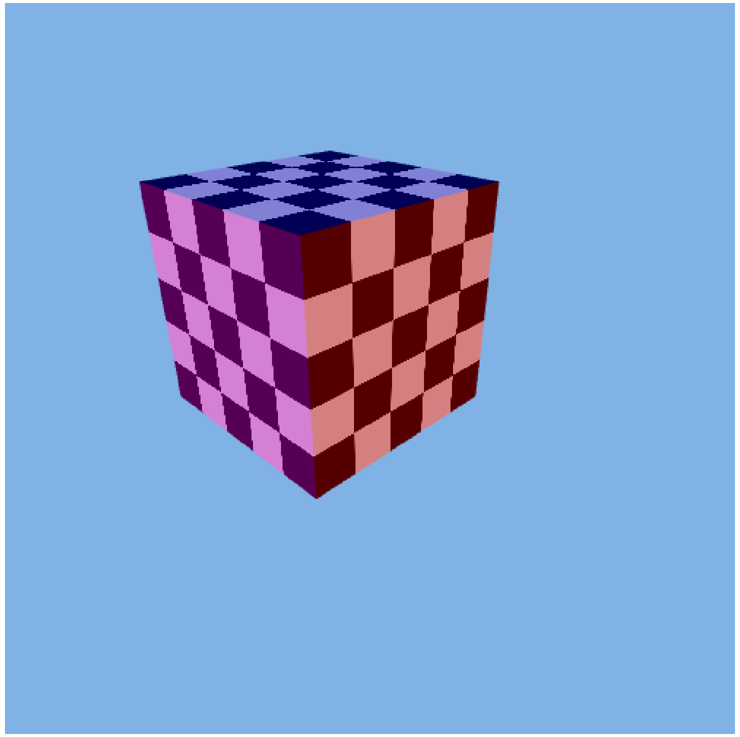


```
texCoords = new Float32Array( [
                    // Front face
                    0.0, 0.0,
                    1.0, 0.0,
                    1.0, 1.0,
                    0.0, 1.0,

                    // Back face
                    1.0, 0.0,
                    1.0, 1.0,
                    0.0, 1.0,
                    0.0, 0.0,

                    // Top face
                    0.0, 1.0,
                    0.0, 0.0,
                    1.0, 0.0,
                    1.0, 1.0,

                    // Bottom face
                    1.0, 1.0,
                    0.0, 1.0,
                    0.0, 0.0,
                    1.0, 0.0,

                    // Right face
                    1.0, 0.0,
                    1.0, 1.0,
                    0.0, 1.0,
                    0.0, 0.0,

                    // Left face
                    0.0, 0.0,
                    1.0, 0.0,
                    1.0, 1.0,
                    0.0, 1.0,
                ] );
```

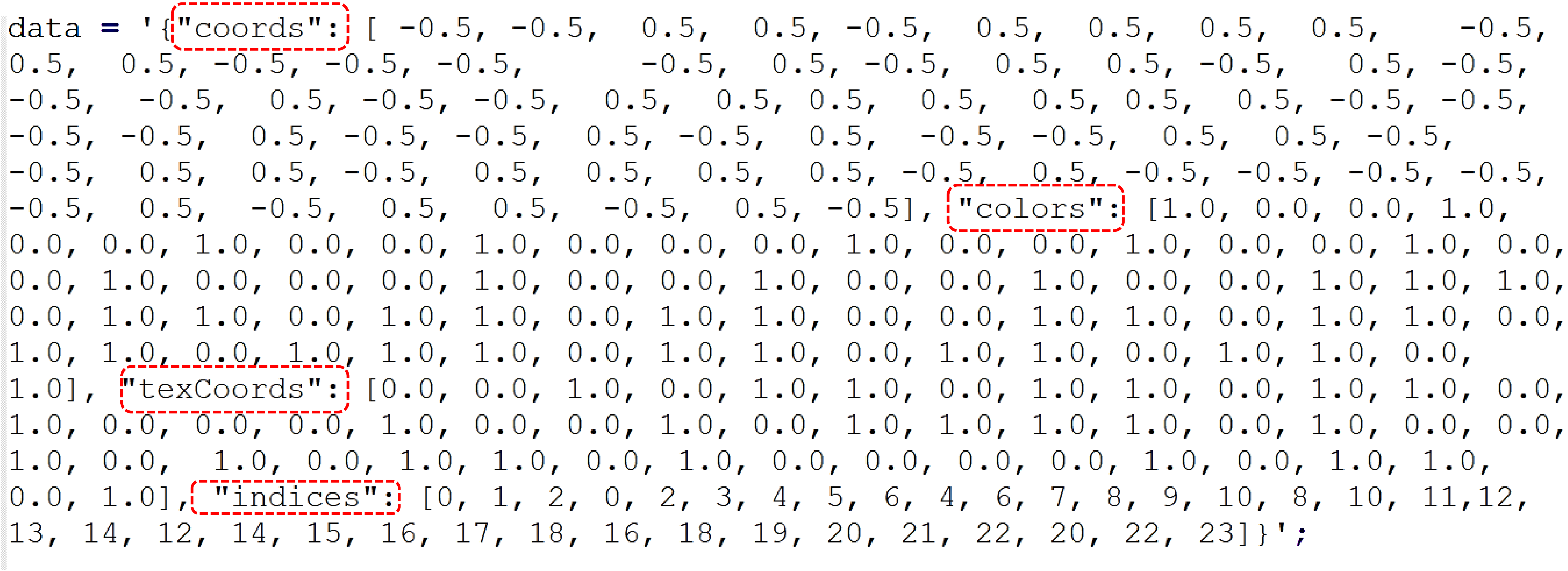
Combining Color + Texel



```
`precision mediump float;
 varying vec3 v_color;
 uniform sampler2D u_sampler_texture;

 varying vec2 v_texCoords;

 void main() {
     vec4 color = texture2D( u_sampler_texture, v_texCoords );
     //gl_FragColor = color;
     gl_FragColor = color/2.0 + vec4(v_color/3.0, 1.0);
 }`;
```



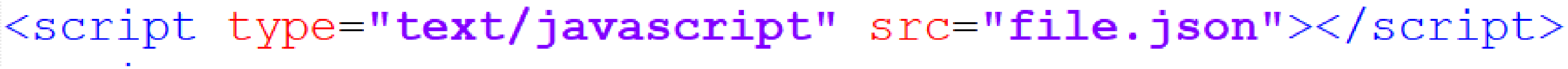
File Load (from JSON file)



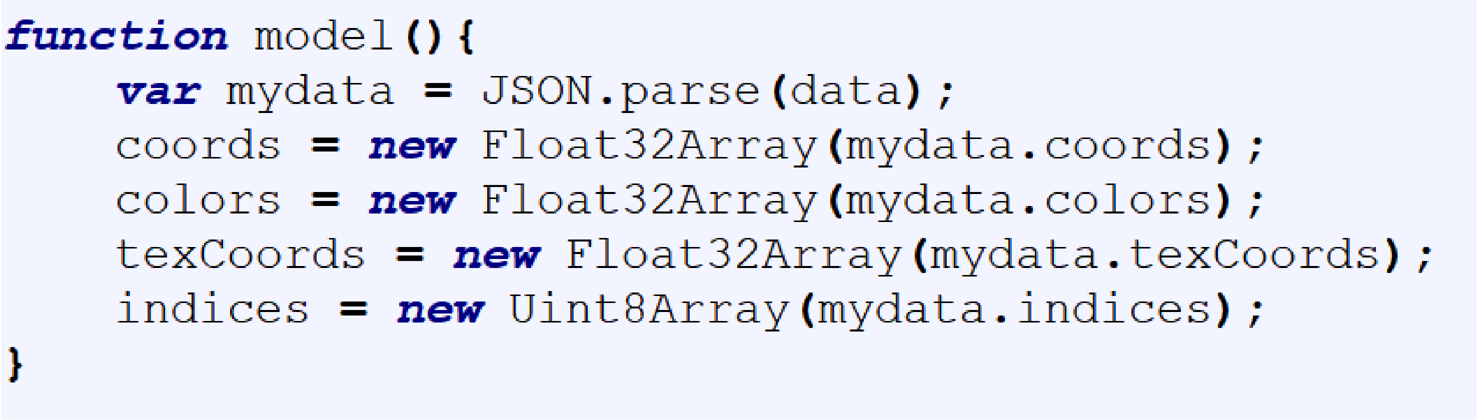
```
data = '{"coords": [ -0.5, -0.5,  0.5,  0.5, -0.5,  0.5,  0.5,  0.5,  0.5,    -0.5,
0.5,  0.5, -0.5, -0.5, -0.5,      -0.5,  0.5, -0.5,  0.5,  0.5, -0.5,   0.5, -0.5,
-0.5,  -0.5,  0.5, -0.5, -0.5,  0.5,  0.5, 0.5,  0.5,  0.5, 0.5,  0.5, -0.5, -0.5,
-0.5, -0.5,  0.5, -0.5, -0.5,  0.5, -0.5,  0.5,  -0.5, -0.5,  0.5,  0.5, -0.5,
-0.5,  0.5,  0.5, -0.5,  0.5,  0.5,  0.5,  0.5, -0.5,  0.5, -0.5, -0.5, -0.5, -0.5,
-0.5,  0.5,  -0.5,  0.5,  0.5,  -0.5,  0.5, -0.5], "colors": [1.0, 0.0, 0.0, 1.0,
0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0,
0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0, 1.0,
0.0, 1.0, 1.0, 0.0, 1.0, 1.0, 0.0, 1.0, 1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 1.0, 1.0, 0.0,
1.0, 1.0, 0.0, 1.0, 1.0, 1.0, 0.0, 1.0, 1.0, 0.0, 1.0, 1.0, 0.0, 1.0, 1.0, 0.0,
1.0], "texCoords": [0.0, 0.0, 1.0, 0.0, 1.0, 1.0, 0.0, 1.0, 1.0, 0.0, 1.0, 1.0, 0.0,
1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0,
1.0, 0.0,  1.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1.0, 1.0,
0.0, 1.0], "indices": [0, 1, 2, 0, 2, 3, 4, 5, 6, 4, 6, 7, 8, 9, 10, 8, 10, 11,12,
13, 14, 12, 14, 15, 16, 17, 18, 16, 18, 19, 20, 21, 22, 20, 22, 23]}';
```

file.json

File Load (from JSON file)

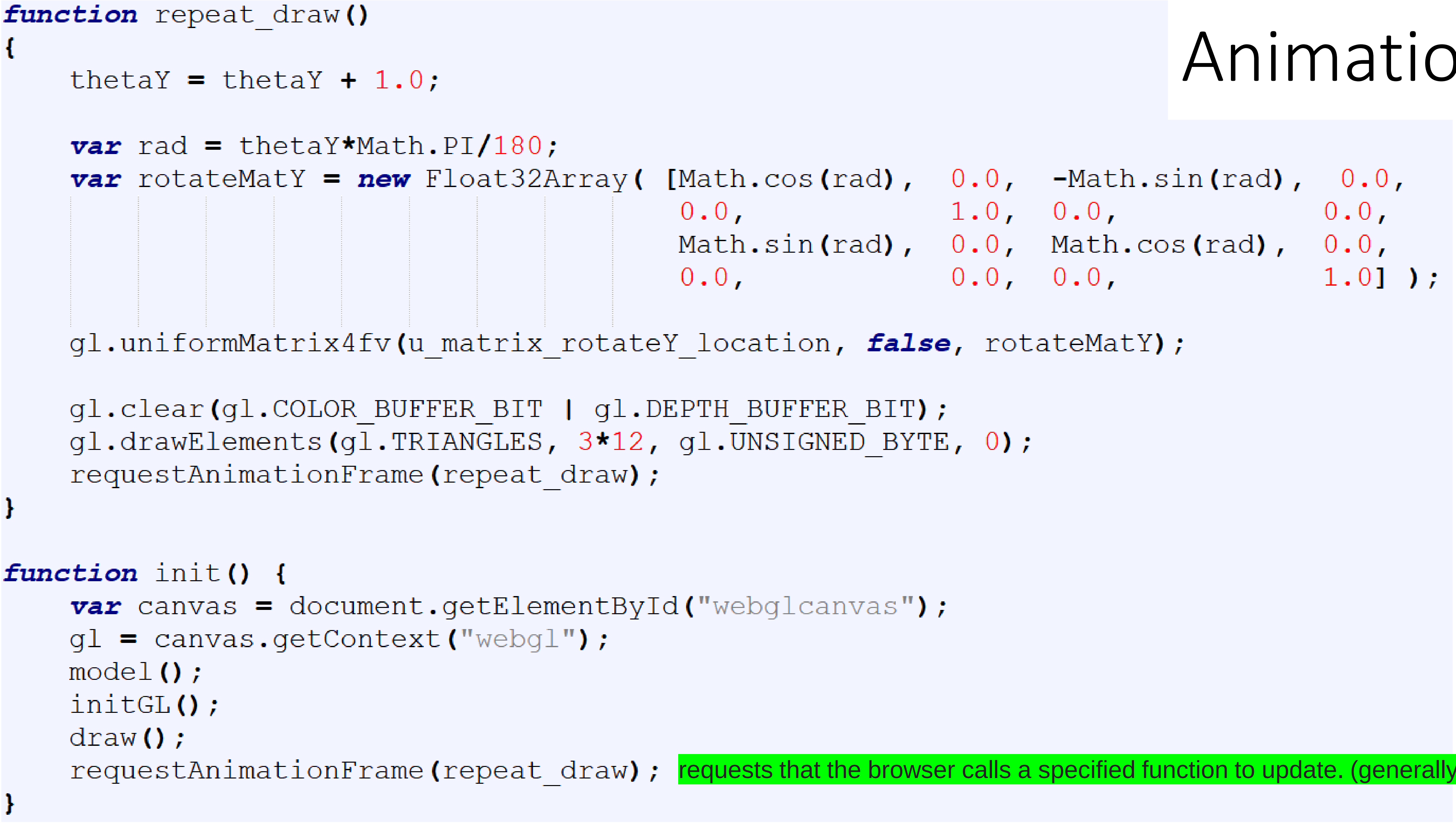


```
<script type="text/javascript" src="file.json"></script>
```



```
function model(){
    var mydata = JSON.parse(data);
    coords = new Float32Array(mydata.coords);
    colors = new Float32Array(mydata.colors);
    texCoords = new Float32Array(mydata.texCoords);
    indices = new Uint8Array(mydata.indices);
}
```

Animation



```
function repeat_draw()
{
    thetaY = thetaY + 1.0;

    var rad = thetaY*Math.PI/180;
    var rotateMatY = new Float32Array( [Math.cos(rad),  0.0,  -Math.sin(rad),  0.0,
                                        0.0,            1.0,  0.0,             0.0,
                                        Math.sin(rad),  0.0,  Math.cos(rad),   0.0,
                                        0.0,            0.0,  0.0,             1.0] );

    gl.uniformMatrix4fv(u_matrix_rotateY_location, false, rotateMatY);

    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
    gl.drawElements(gl.TRIANGLES, 3*12, gl.UNSIGNED_BYTE, 0);
    requestAnimationFrame(repeat_draw);
}

function init() {
    var canvas = document.getElementById("webglcanvas");
    gl = canvas.getContext("webgl");
    model();
    initGL();
    draw();
    requestAnimationFrame(repeat_draw);
}
```

requests that the browser calls a specified function to update. (generally 60FPS)

Source: https://developer.mozilla.org/en-US/docs/Web/API/window/requestAnimationFrame

Thank You