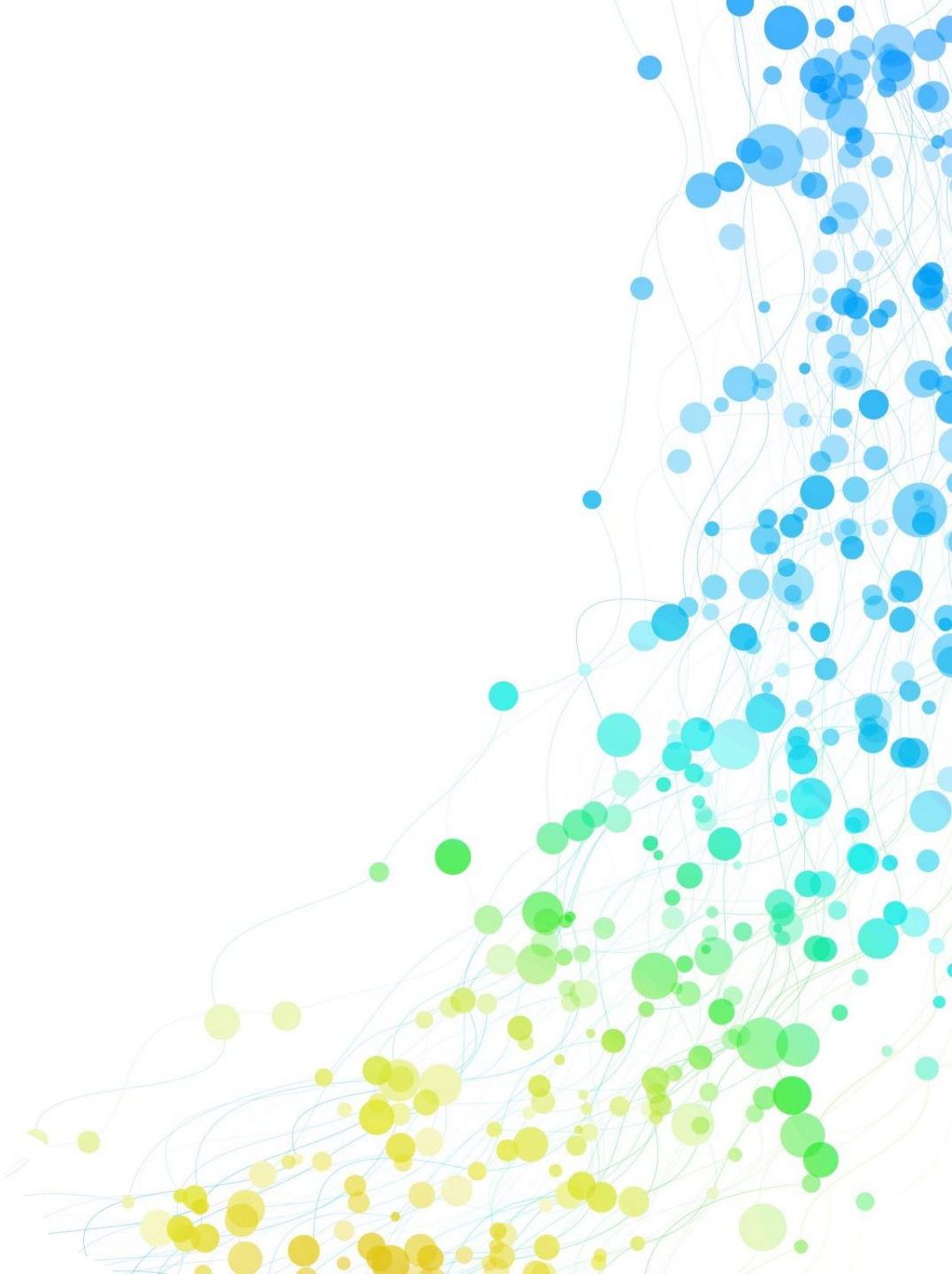
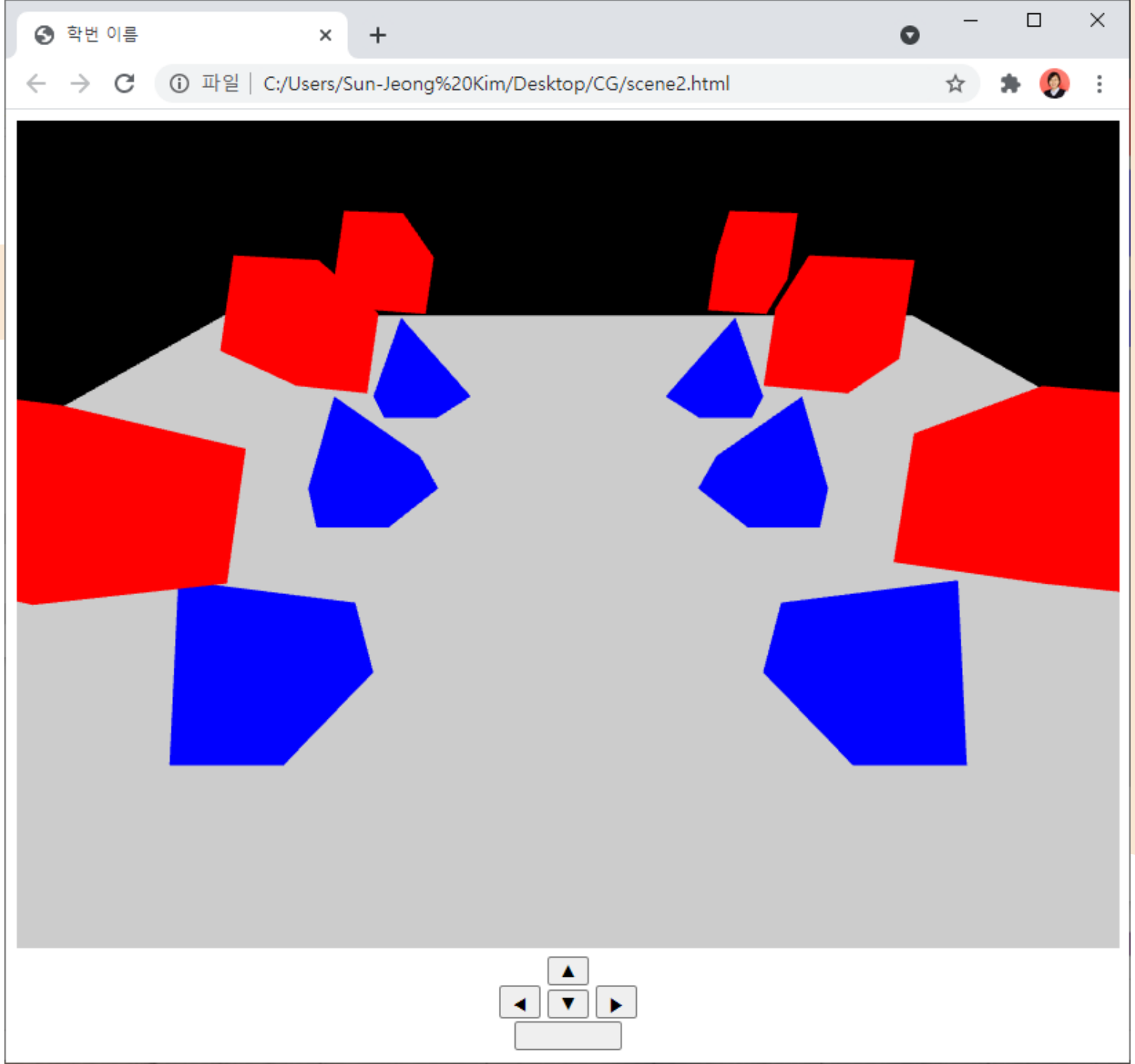
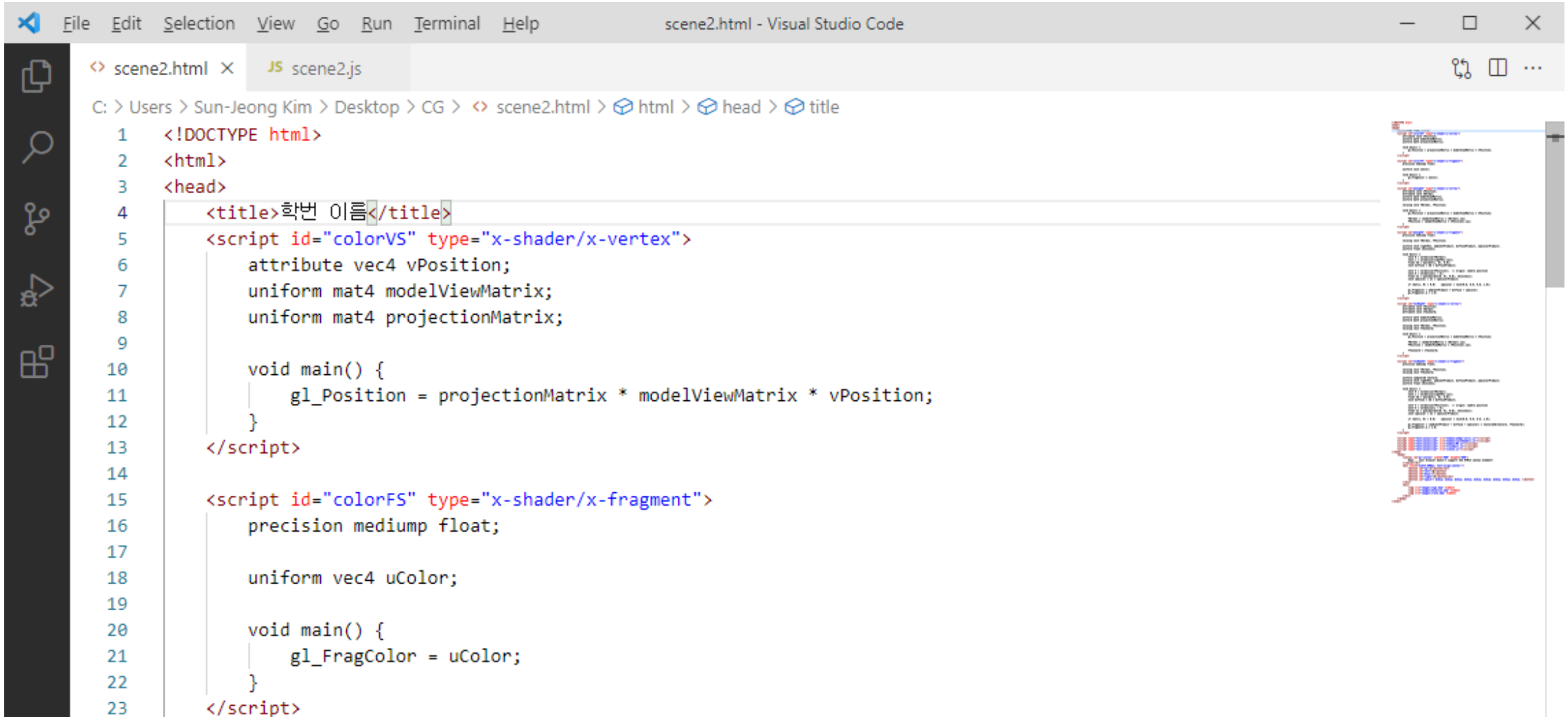
Building a Scene with Texture Mapping

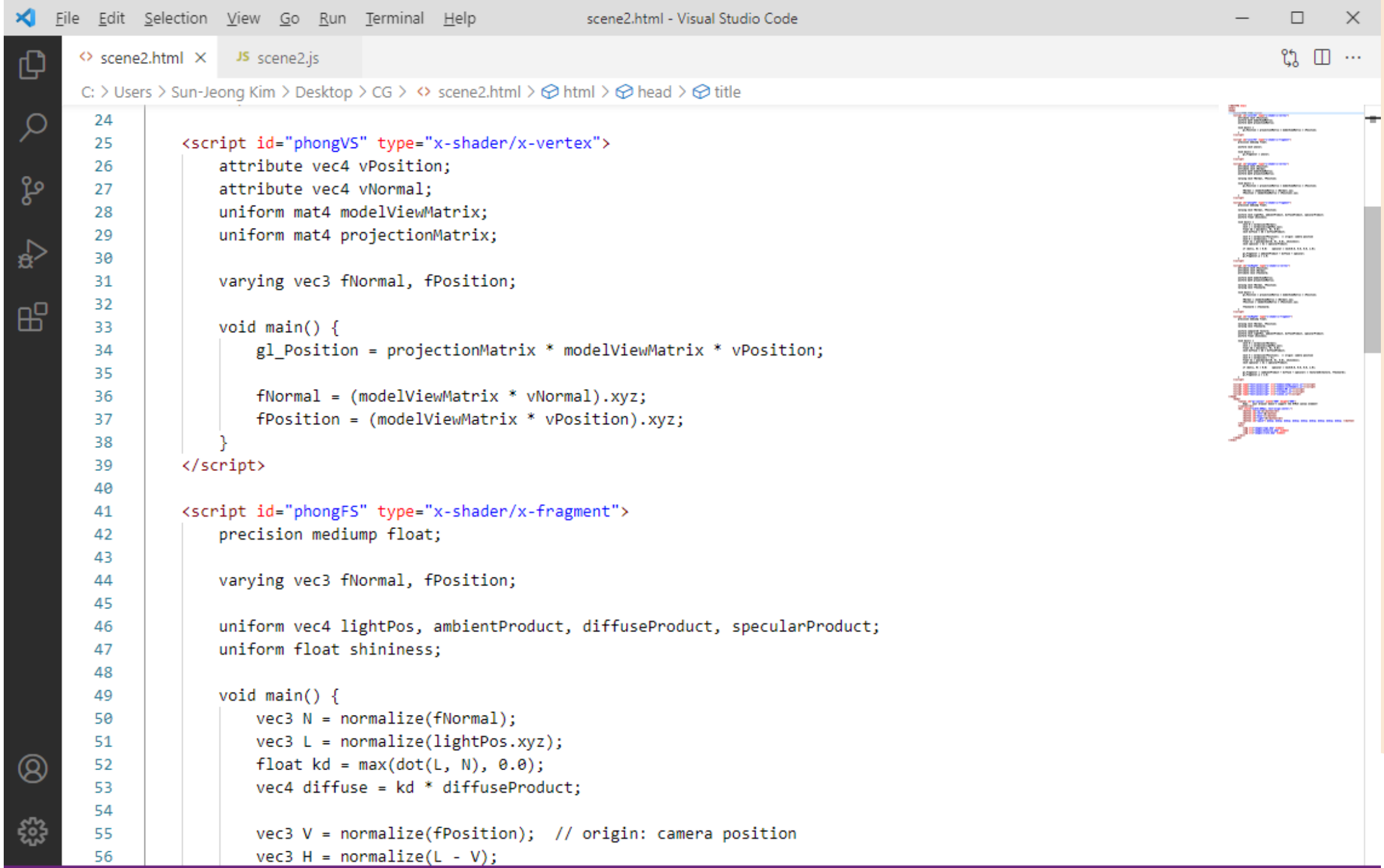
14TH WEEK, 2021

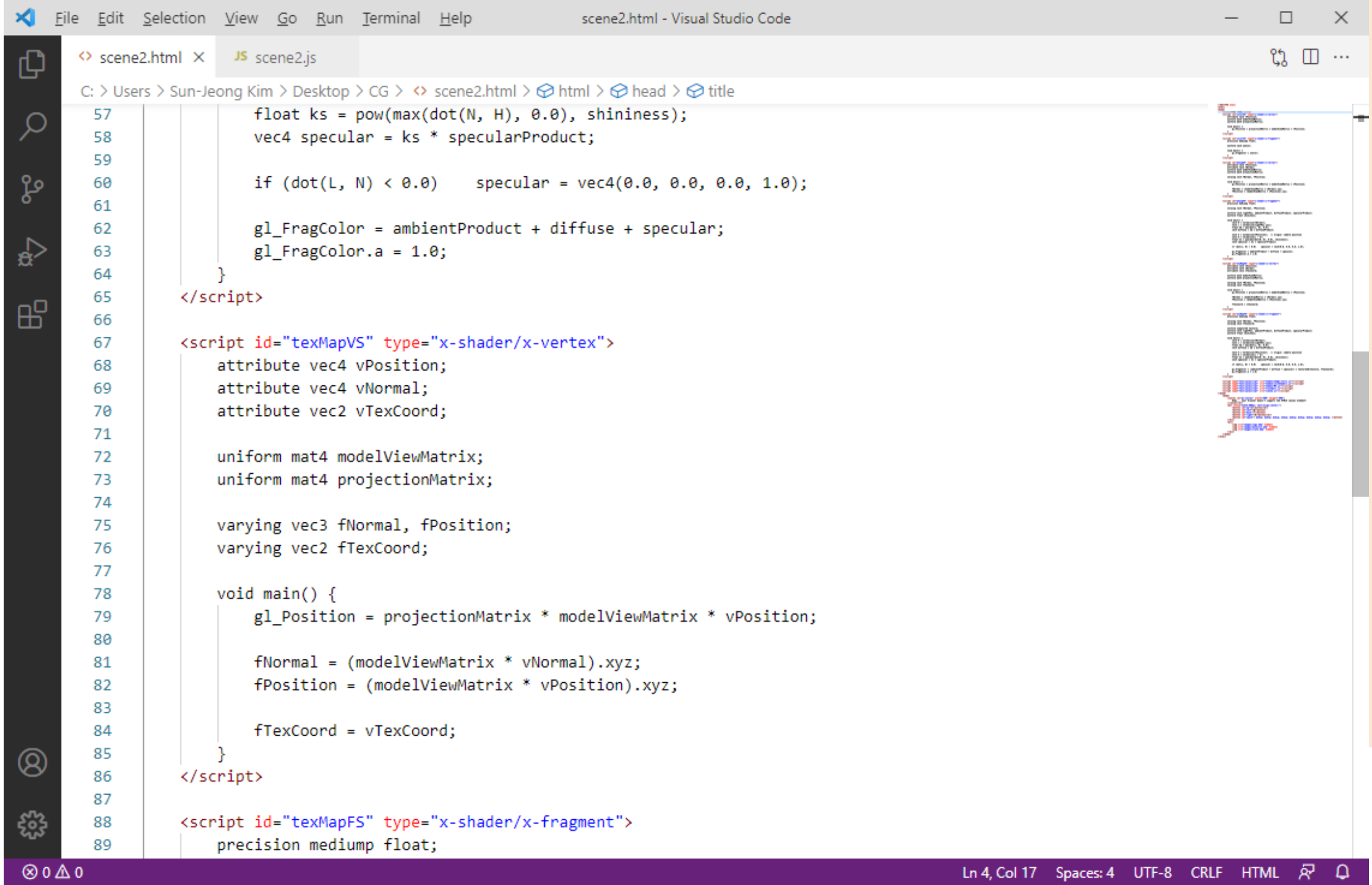


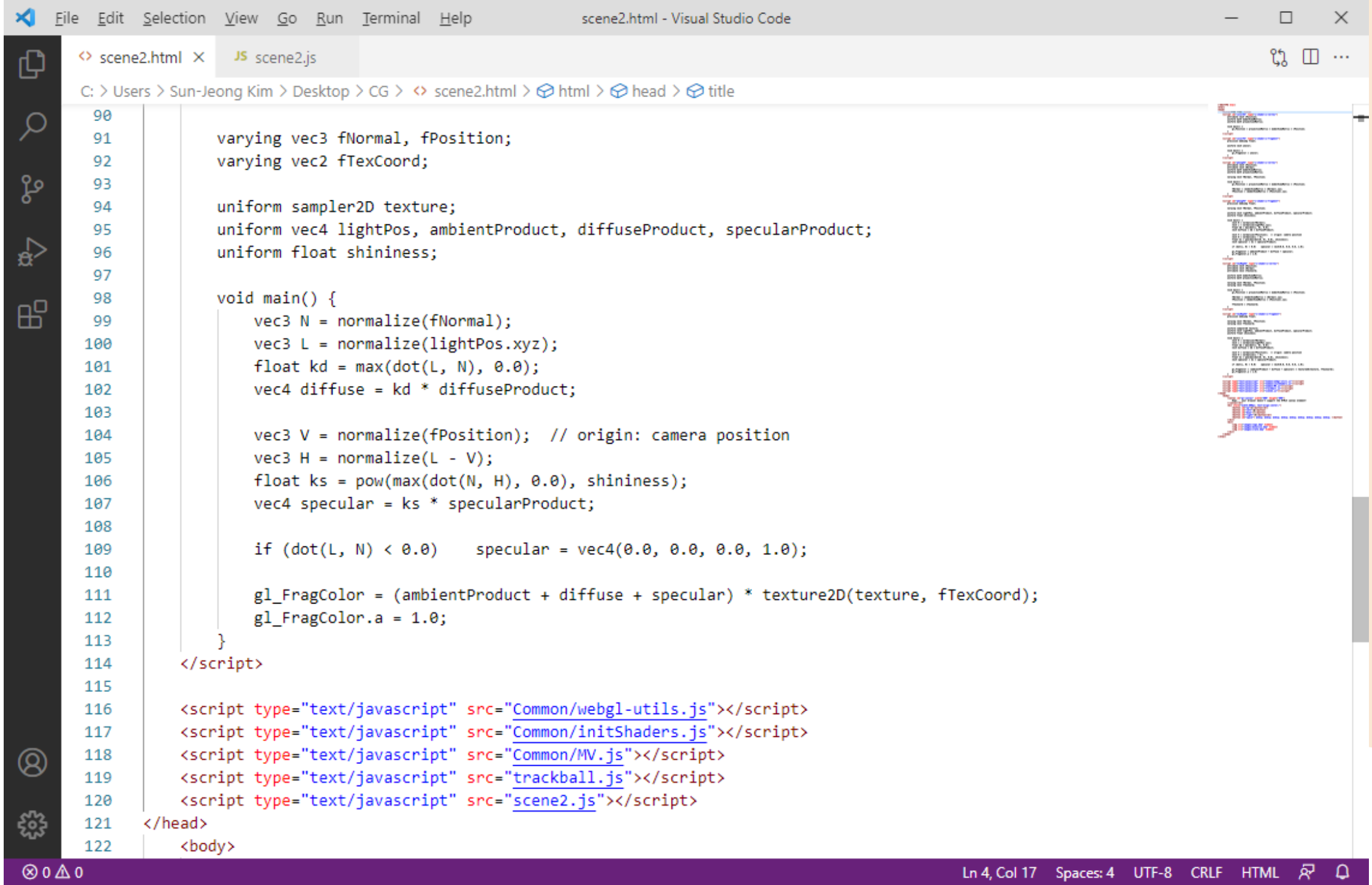

학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene2.html

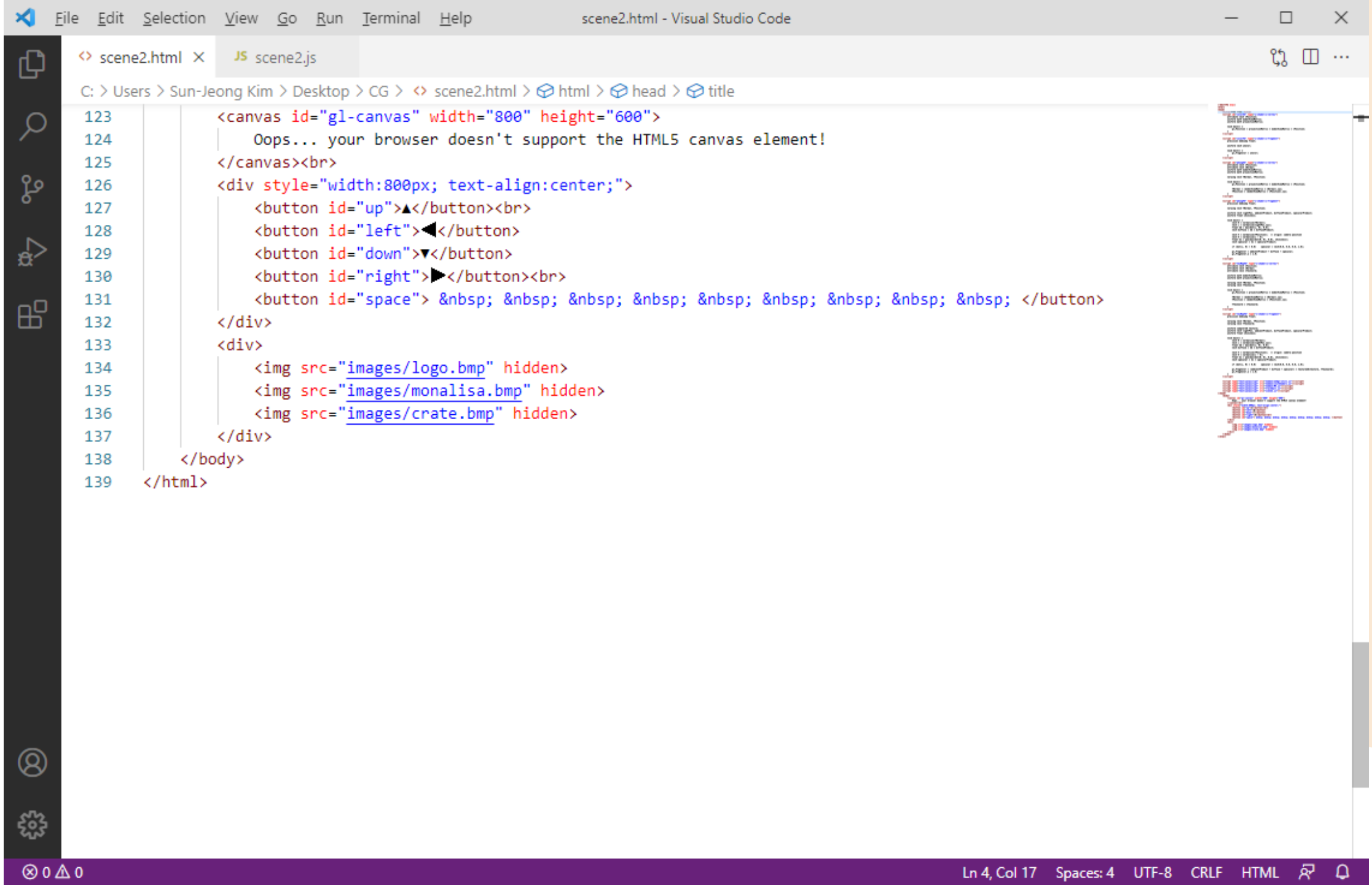
Multiple Shaders




File Edit Selection View Go Run Terminal Help
scene2.html - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.html > html > head > title
24
25 <script id="phongVS" type="x-shader/x-vertex">
26 attribute vec4 vPosition;
27 attribute vec4 vNormal;
28 uniform mat4 modelViewMatrix;
29 uniform mat4 projectionMatrix;
30
31 varying vec3 fNormal, fPosition;
32
33 void main() {
34 gl_Position = projectionMatrix * modelViewMatrix * vPosition;
35
36 fNormal = (modelViewMatrix * vNormal).xyz;
37 fPosition = (modelViewMatrix * vPosition).xyz;
38 }
39 </script>
40
41 <script id="phongFS" type="x-shader/x-fragment">
42 precision mediump float;
43
44 varying vec3 fNormal, fPosition;
45
46 uniform vec4 lightPos, ambientProduct, diffuseProduct, specularProduct;
47 uniform float shininess;
48
49 void main() {
50 vec3 N = normalize(fNormal);
51 vec3 L = normalize(lightPos.xyz);
52 float kd = max(dot(L, N), 0.0);
53 vec4 diffuse = kd * diffuseProduct;
54
55 vec3 V = normalize(fPosition); // origin: camera position
56 vec3 H = normalize(L - V);
⊗ 0 ⚠ 0
Ln 4, Col 17 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
scene2.html - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.html > html > head > title
57 float ks = pow(max(dot(N, H), 0.0), shininess);
58 vec4 specular = ks * specularProduct;
59
60 if (dot(L, N) < 0.0) specular = vec4(0.0, 0.0, 0.0, 1.0);
61
62 gl_FragColor = ambientProduct + diffuse + specular;
63 gl_FragColor.a = 1.0;
64 }
65 </script>
66
67 <script id="texMapVS" type="x-shader/x-vertex">
68 attribute vec4 vPosition;
69 attribute vec4 vNormal;
70 attribute vec2 vTexCoord;
71
72 uniform mat4 modelViewMatrix;
73 uniform mat4 projectionMatrix;
74
75 varying vec3 fNormal, fPosition;
76 varying vec2 fTexCoord;
77
78 void main() {
79 gl_Position = projectionMatrix * modelViewMatrix * vPosition;
80
81 fNormal = (modelViewMatrix * vNormal).xyz;
82 fPosition = (modelViewMatrix * vPosition).xyz;
83
84 fTexCoord = vTexCoord;
85 }
86 </script>
87
88 <script id="texMapFS" type="x-shader/x-fragment">
89 precision mediump float;
0 0
Ln 4, Col 17 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
scene2.html - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.html > html > head > title
varying vec3 fNormal, fPosition;
varying vec2 fTexCoord;
uniform sampler2D texture;
uniform vec4 lightPos, ambientProduct, diffuseProduct, specularProduct;
uniform float shininess;
void main() {
vec3 N = normalize(fNormal);
vec3 L = normalize(lightPos.xyz);
float kd = max(dot(L, N), 0.0);
vec4 diffuse = kd * diffuseProduct;
vec3 V = normalize(fPosition); // origin: camera position
vec3 H = normalize(L - V);
float ks = pow(max(dot(N, H), 0.0), shininess);
vec4 specular = ks * specularProduct;
if (dot(L, N) < 0.0) specular = vec4(0.0, 0.0, 0.0, 1.0);
gl_FragColor = (ambientProduct + diffuse + specular) * texture2D(texture, fTexCoord);
gl_FragColor.a = 1.0;
}
</script>
<script type="text/javascript" src="Common/webgl-utils.js"></script>
<script type="text/javascript" src="Common/initShaders.js"></script>
<script type="text/javascript" src="Common/MV.js"></script>
<script type="text/javascript" src="trackball.js"></script>
<script type="text/javascript" src="scene2.js"></script>
</head>
<body>
Ln 4, Col 17 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
scene2.html - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.html > html > head > title
123 <canvas id="gl-canvas" width="800" height="600">
124 Oops... your browser doesn't support the HTML5 canvas element!
125 </canvas>
126
127 <button id="up">▲</button>
128 <button id="left">◀</button>
129 <button id="down">▼</button>
130 <button id="right">▶</button>
131 <button id="space">   </button>
132
133
134 <img src="images/logo.bmp" hidden>
135 <img src="images/monalisa.bmp" hidden>
136 <img src="images/crate.bmp" hidden>
137
138 </body>
139 </html>
⊗ 0 ⚠ 0
Ln 4, Col 17 Spaces: 4 UTF-8 CRLF HTML

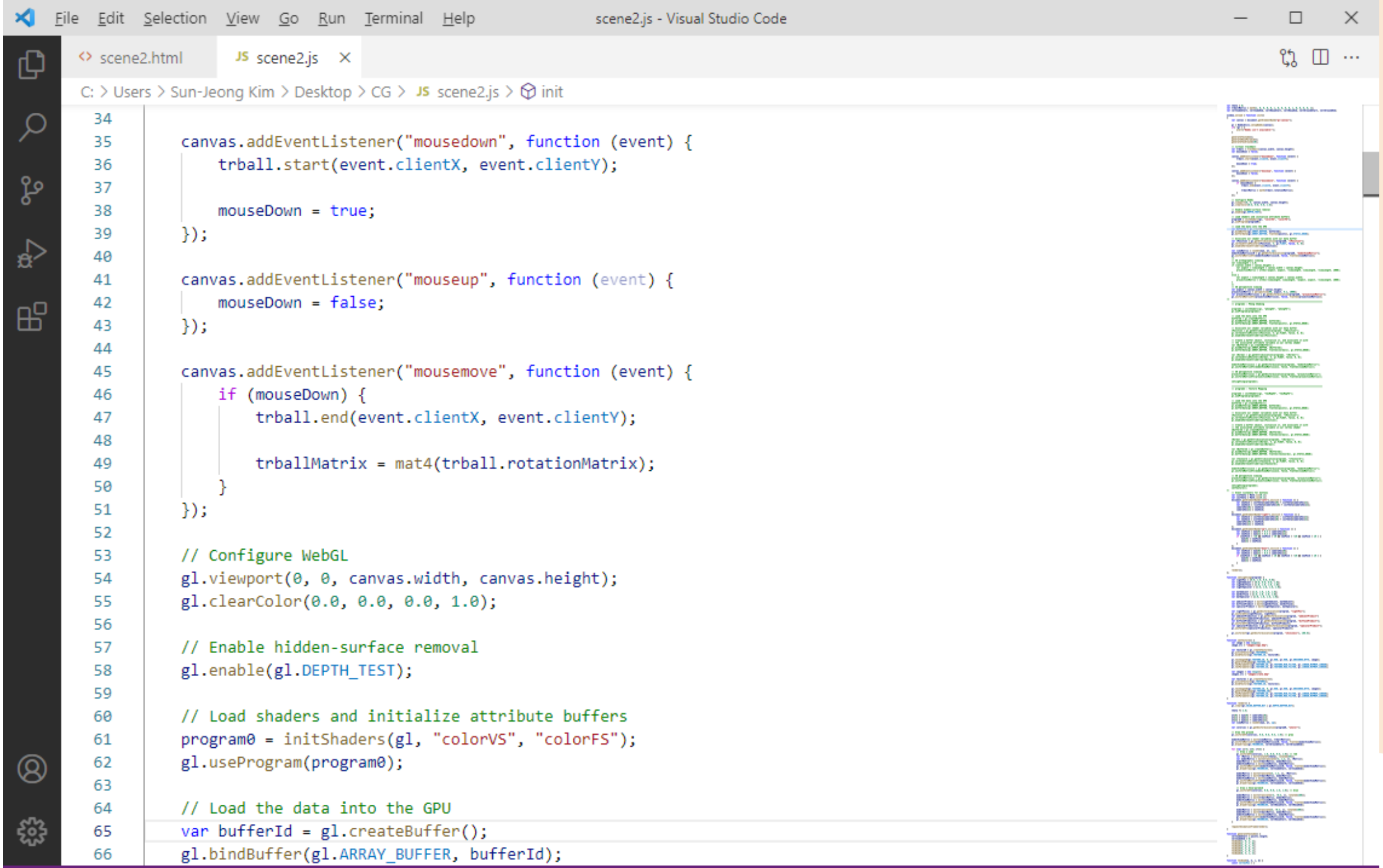
File Edit Selection View Go Run Terminal Help scene2.js - Visual Studio Code

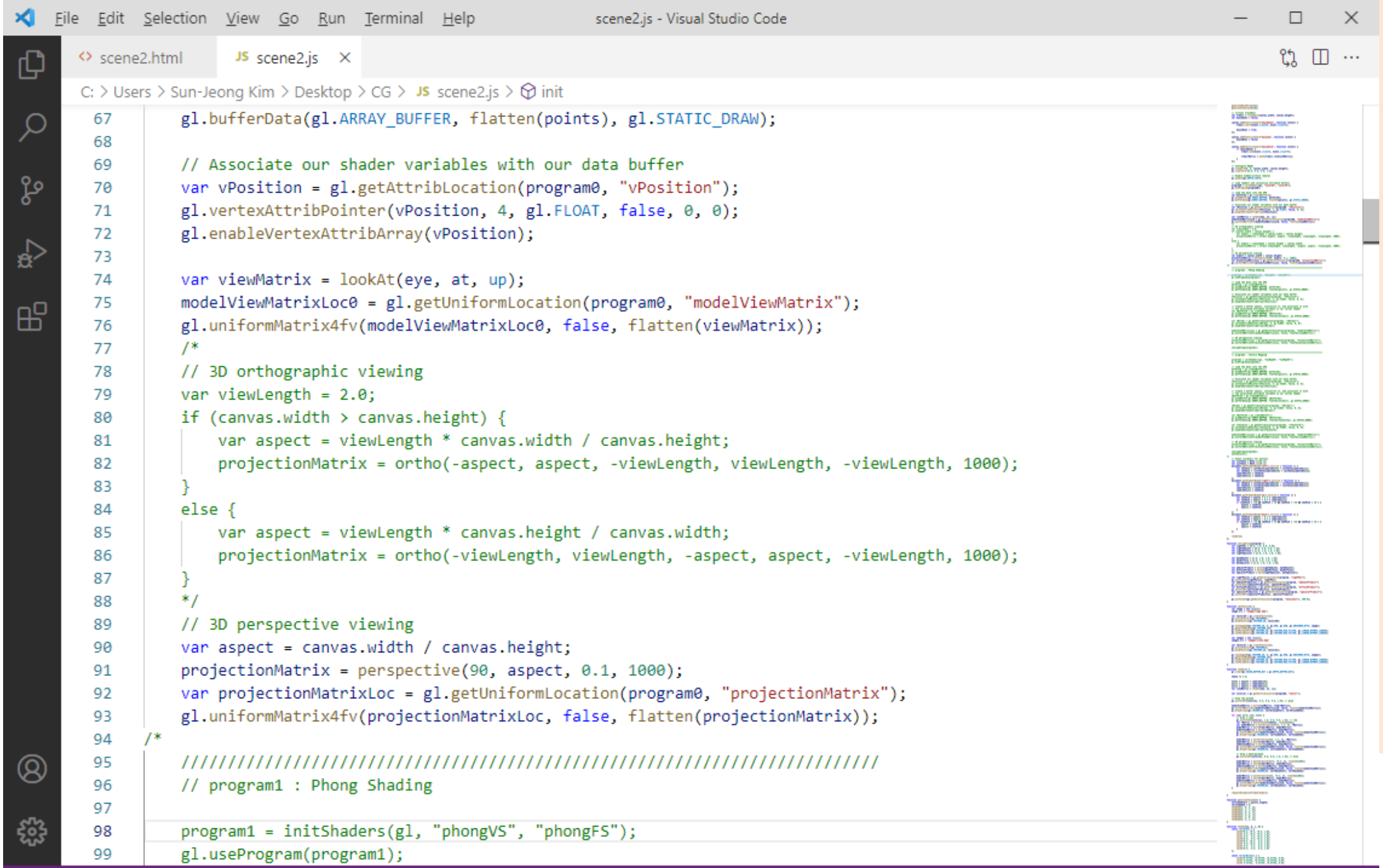
scene2.html scene2.js

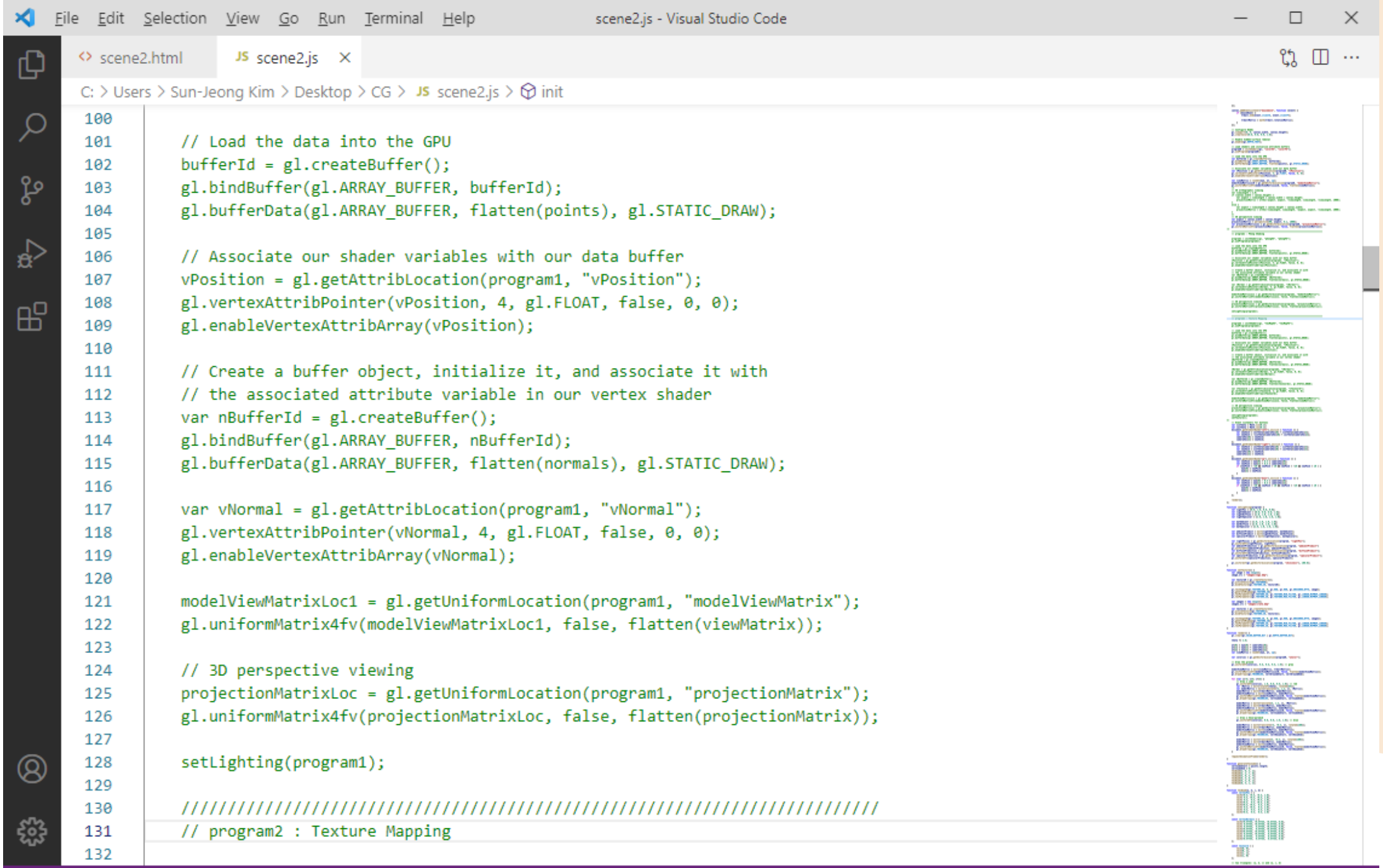
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > ...

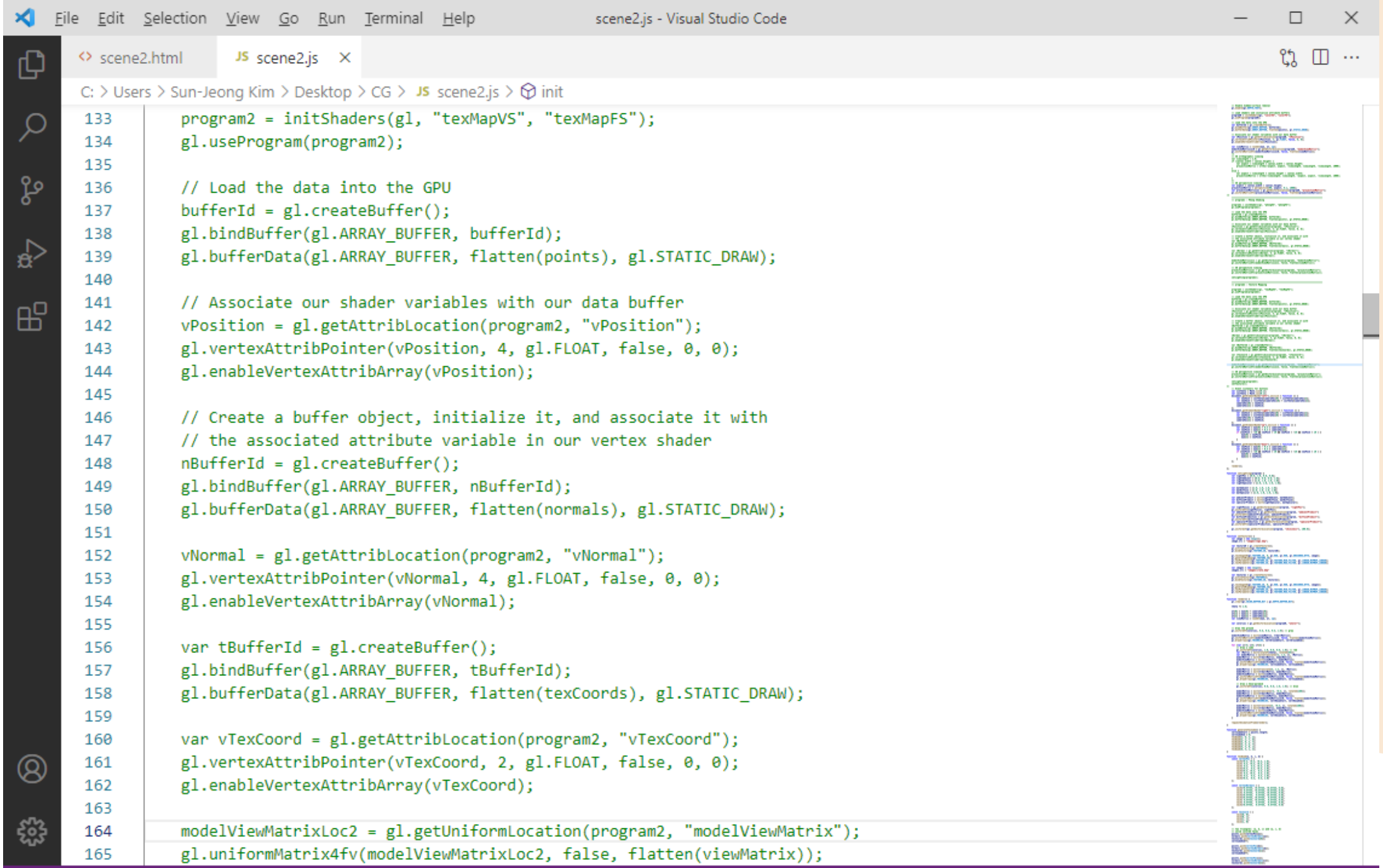
```
var gl;
var points = [];
var normals = [];
var texCoords = [];

var program0, program1, program2;     // [program1] Phong shading, [program2] Texture Mapping
var modelViewMatrixLoc0, modelViewMatrixLoc1, modelViewMatrixLoc2;

var eye = vec3(0, 3, 3);
var at = vec3(0, 0, 0);
const up = vec3(0, 1, 0);
var cameraVec = vec3(0, -0.7071, -0.7071); // 1.0/Math.sqrt(2.0)

var theta = 0;
var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);
var vertCubeStart, vertCubeEnd, vertHexaStart, vertHexaEnd, vertGroundStart, vertGroundEnd;

window.onload = function init()
{
    var canvas = document.getElementById("gl-canvas");

    gl = WebGLUtils.setupWebGL(canvas);
    if( !gl ) {
        alert("WebGL isn't available!");
    }

    generateTexCube();
    generateHexaPyramid();
    generateTexGround(10);

    // virtual trackball
    var trball = trackball(canvas.width, canvas.height);
    var mouseDown = false;
```

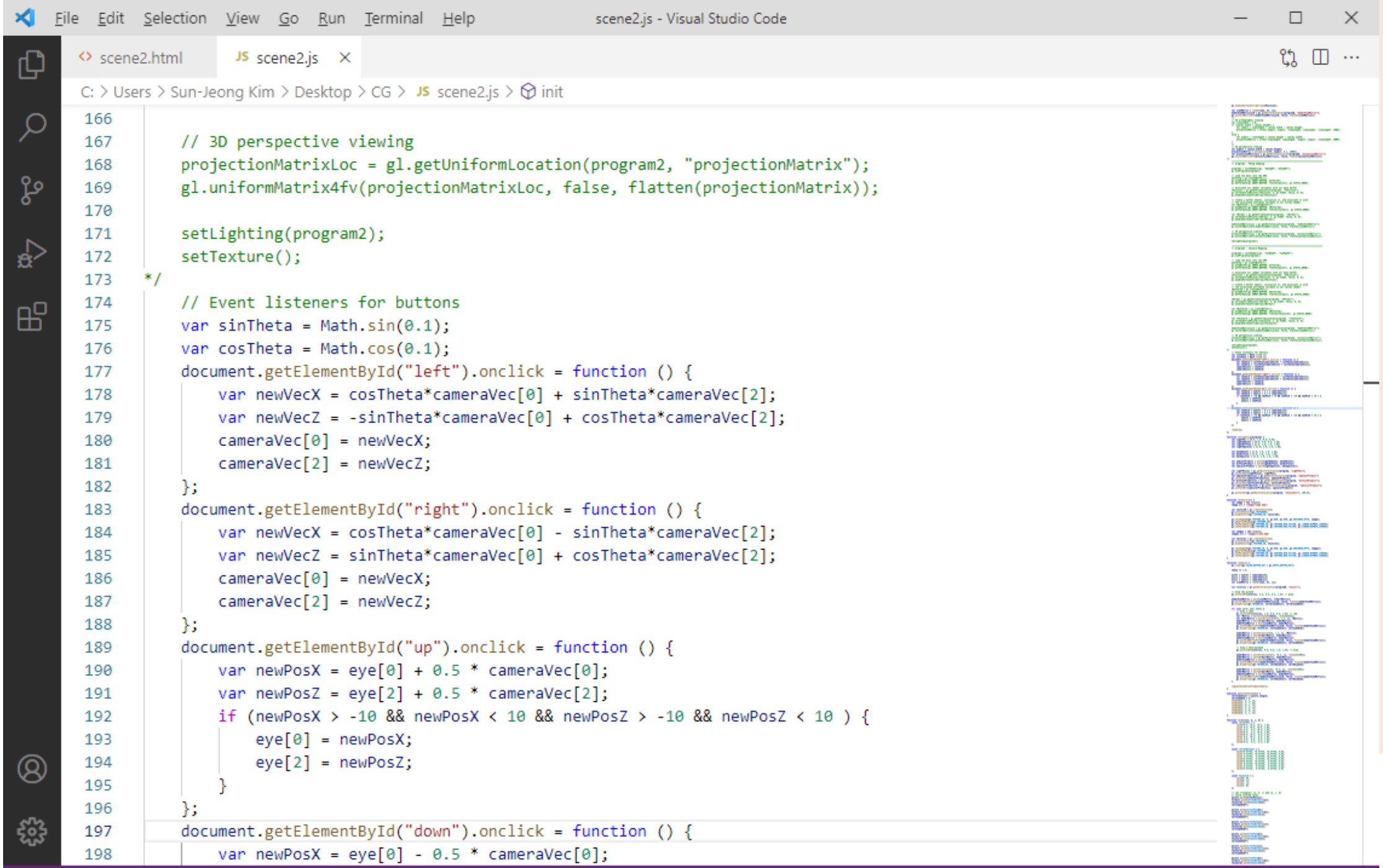
⊗ 0 ⚠ 0 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

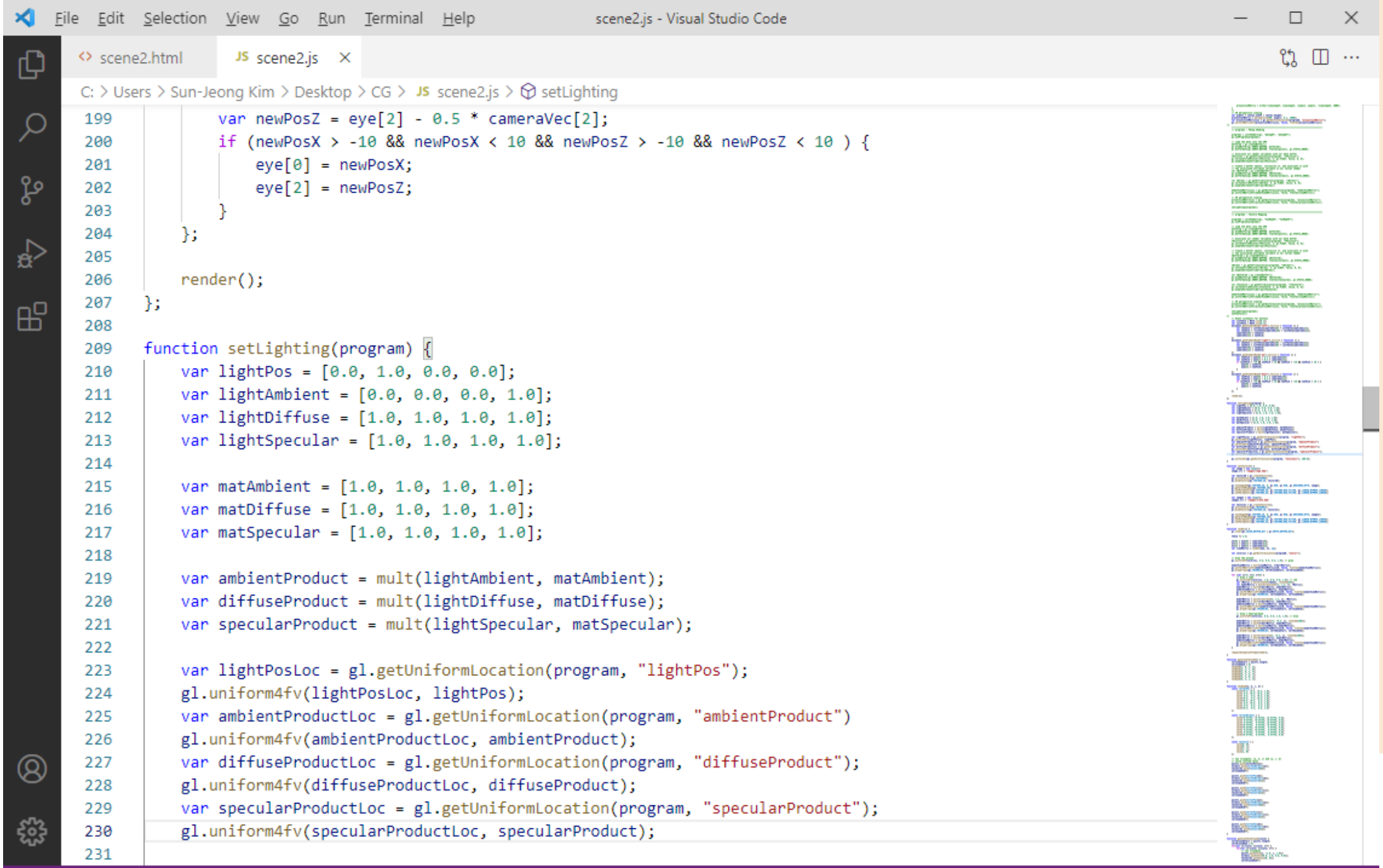

File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > init
34
35 canvas.addEventListener("mousedown", function (event) {
36 trball.start(event.clientX, event.clientY);
37
38 mouseDown = true;
39 });
40
41 canvas.addEventListener("mouseup", function (event) {
42 mouseDown = false;
43 });
44
45 canvas.addEventListener("mousemove", function (event) {
46 if (mouseDown) {
47 trball.end(event.clientX, event.clientY);
48
49 trballMatrix = mat4(trball.rotationMatrix);
50 }
51 });
52
53 // Configure WebGL
54 gl.viewport(0, 0, canvas.width, canvas.height);
55 gl.clearColor(0.0, 0.0, 0.0, 1.0);
56
57 // Enable hidden-surface removal
58 gl.enable(gl.DEPTH_TEST);
59
60 // Load shaders and initialize attribute buffers
61 program0 = initShaders(gl, "colorVS", "colorFS");
62 gl.useProgram(program0);
63
64 // Load the data into the GPU
65 var bufferId = gl.createBuffer();
66 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
⊗ 0 ⚠ 0
Ln 65, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

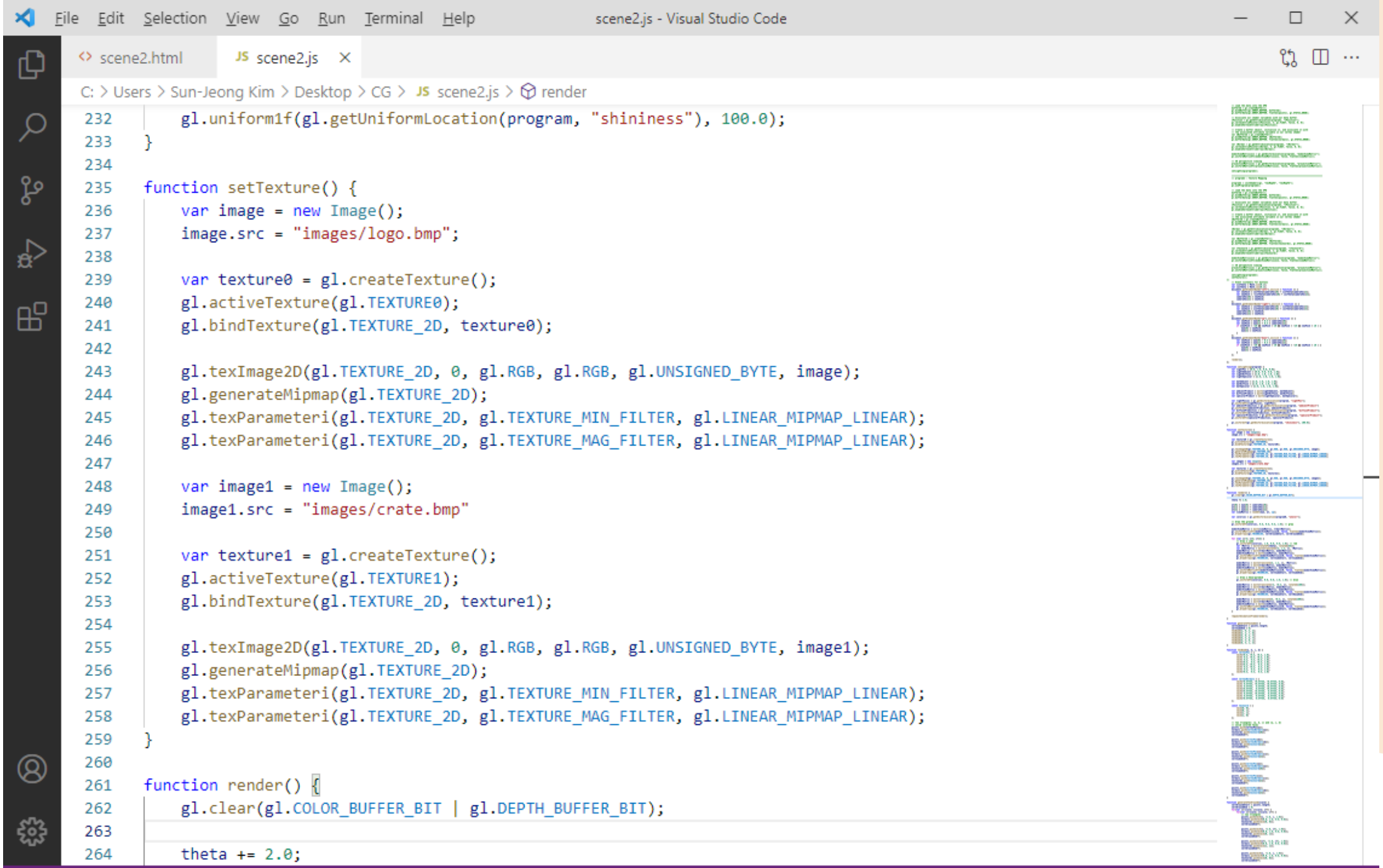

File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > init
gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
// Associate our shader variables with our data buffer
var vPosition = gl.getAttribLocation(program0, "vPosition");
gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
gl.enableVertexAttribArray(vPosition);
var viewMatrix = lookAt(eye, at, up);
modelViewMatrixLoc0 = gl.getUniformLocation(program0, "modelViewMatrix");
gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(viewMatrix));
/*
// 3D orthographic viewing
var viewLength = 2.0;
if (canvas.width > canvas.height) {
var aspect = viewLength * canvas.width / canvas.height;
projectionMatrix = ortho(-aspect, aspect, -viewLength, viewLength, -viewLength, 1000);
}
else {
var aspect = viewLength * canvas.height / canvas.width;
projectionMatrix = ortho(-viewLength, viewLength, -aspect, aspect, -viewLength, 1000);
}
*/
// 3D perspective viewing
var aspect = canvas.width / canvas.height;
projectionMatrix = perspective(90, aspect, 0.1, 1000);
var projectionMatrixLoc = gl.getUniformLocation(program0, "projectionMatrix");
gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projectionMatrix));
/*
//
// program1 : Phong Shading
program1 = initShaders(gl, "phongVS", "phongFS");
gl.useProgram(program1);
0 0
Ln 98, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > init
100
101 // Load the data into the GPU
102 bufferId = gl.createBuffer();
103 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
104 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
105
106 // Associate our shader variables with our data buffer
107 vPosition = gl.getAttribLocation(program1, "vPosition");
108 gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
109 gl.enableVertexAttribArray(vPosition);
110
111 // Create a buffer object, initialize it, and associate it with
112 // the associated attribute variable in our vertex shader
113 var nBufferId = gl.createBuffer();
114 gl.bindBuffer(gl.ARRAY_BUFFER, nBufferId);
115 gl.bufferData(gl.ARRAY_BUFFER, flatten(normals), gl.STATIC_DRAW);
116
117 var vNormal = gl.getAttribLocation(program1, "vNormal");
118 gl.vertexAttribPointer(vNormal, 4, gl.FLOAT, false, 0, 0);
119 gl.enableVertexAttribArray(vNormal);
120
121 modelViewMatrixLoc1 = gl.getUniformLocation(program1, "modelViewMatrix");
122 gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(viewMatrix));
123
124 // 3D perspective viewing
125 projectionMatrixLoc = gl.getUniformLocation(program1, "projectionMatrix");
126 gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projectionMatrix));
127
128 setLighting(program1);
129
130 //
131 // program2 : Texture Mapping
132
0 0
Ln 131, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > init
133 program2 = initShaders(gl, "texMapVS", "texMapFS");
134 gl.useProgram(program2);
135
136 // Load the data into the GPU
137 bufferId = gl.createBuffer();
138 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
139 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
140
141 // Associate our shader variables with our data buffer
142 vPosition = gl.getAttribLocation(program2, "vPosition");
143 gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
144 gl.enableVertexAttribArray(vPosition);
145
146 // Create a buffer object, initialize it, and associate it with
147 // the associated attribute variable in our vertex shader
148 nBufferId = gl.createBuffer();
149 gl.bindBuffer(gl.ARRAY_BUFFER, nBufferId);
150 gl.bufferData(gl.ARRAY_BUFFER, flatten(normals), gl.STATIC_DRAW);
151
152 vNormal = gl.getAttribLocation(program2, "vNormal");
153 gl.vertexAttribPointer(vNormal, 4, gl.FLOAT, false, 0, 0);
154 gl.enableVertexAttribArray(vNormal);
155
156 var tBufferId = gl.createBuffer();
157 gl.bindBuffer(gl.ARRAY_BUFFER, tBufferId);
158 gl.bufferData(gl.ARRAY_BUFFER, flatten(texCoords), gl.STATIC_DRAW);
159
160 var vTexCoord = gl.getAttribLocation(program2, "vTexCoord");
161 gl.vertexAttribPointer(vTexCoord, 2, gl.FLOAT, false, 0, 0);
162 gl.enableVertexAttribArray(vTexCoord);
163
164 modelViewMatrixLoc2 = gl.getUniformLocation(program2, "modelViewMatrix");
165 gl.uniformMatrix4fv(modelViewMatrixLoc2, false, flatten(viewMatrix));
⊗ 0 ⚠ 0
Ln 164, Col 1 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > init
166
167 // 3D perspective viewing
168 projectionMatrixLoc = gl.getUniformLocation(program2, "projectionMatrix");
169 gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projectionMatrix));
170
171 setLighting(program2);
172 setTexture();
173 */
174 // Event listeners for buttons
175 var sinTheta = Math.sin(0.1);
176 var cosTheta = Math.cos(0.1);
177 document.getElementById("left").onclick = function () {
178 var newVecX = cosTheta*cameraVec[0] + sinTheta*cameraVec[2];
179 var newVecZ = -sinTheta*cameraVec[0] + cosTheta*cameraVec[2];
180 cameraVec[0] = newVecX;
181 cameraVec[2] = newVecZ;
182 };
183 document.getElementById("right").onclick = function () {
184 var newVecX = cosTheta*cameraVec[0] - sinTheta*cameraVec[2];
185 var newVecZ = sinTheta*cameraVec[0] + cosTheta*cameraVec[2];
186 cameraVec[0] = newVecX;
187 cameraVec[2] = newVecZ;
188 };
189 document.getElementById("up").onclick = function () {
190 var newPosX = eye[0] + 0.5 * cameraVec[0];
191 var newPosZ = eye[2] + 0.5 * cameraVec[2];
192 if (newPosX > -10 && newPosX < 10 && newPosZ > -10 && newPosZ < 10) {
193 eye[0] = newPosX;
194 eye[2] = newPosZ;
195 }
196 };
197 document.getElementById("down").onclick = function () {
198 var newPosX = eye[0] - 0.5 * cameraVec[0];
⊗ 0 ⚠ 0
Ln 197, Col 1 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > setLighting
199 var newPosZ = eye[2] - 0.5 * cameraVec[2];
200 if (newPosX > -10 && newPosX < 10 && newPosZ > -10 && newPosZ < 10) {
201 eye[0] = newPosX;
202 eye[2] = newPosZ;
203 }
204 };
205
206 render();
207 };
208
209 function setLighting(program) {
210 var lightPos = [0.0, 1.0, 0.0, 0.0];
211 var lightAmbient = [0.0, 0.0, 0.0, 1.0];
212 var lightDiffuse = [1.0, 1.0, 1.0, 1.0];
213 var lightSpecular = [1.0, 1.0, 1.0, 1.0];
214
215 var matAmbient = [1.0, 1.0, 1.0, 1.0];
216 var matDiffuse = [1.0, 1.0, 1.0, 1.0];
217 var matSpecular = [1.0, 1.0, 1.0, 1.0];
218
219 var ambientProduct = mult(lightAmbient, matAmbient);
220 var diffuseProduct = mult(lightDiffuse, matDiffuse);
221 var specularProduct = mult(lightSpecular, matSpecular);
222
223 var lightPosLoc = gl.getUniformLocation(program, "lightPos");
224 gl.uniform4fv(lightPosLoc, lightPos);
225 var ambientProductLoc = gl.getUniformLocation(program, "ambientProduct")
226 gl.uniform4fv(ambientProductLoc, ambientProduct);
227 var diffuseProductLoc = gl.getUniformLocation(program, "diffuseProduct");
228 gl.uniform4fv(diffuseProductLoc, diffuseProduct);
229 var specularProductLoc = gl.getUniformLocation(program, "specularProduct");
230 gl.uniform4fv(specularProductLoc, specularProduct);
231
⊗ 0 ⚠ 0
Ln 230, Col 1 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > render
232 gl.uniform1f(gl.getUniformLocation(program, "shininess"), 100.0);
233 }
234
235 function setTexture() {
236 var image = new Image();
237 image.src = "images/logo.bmp";
238
239 var texture0 = gl.createTexture();
240 gl.activeTexture(gl.TEXTURE0);
241 gl.bindTexture(gl.TEXTURE_2D, texture0);
242
243 gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGB, gl.RGB, gl.UNSIGNED_BYTE, image);
244 gl.generateMipmap(gl.TEXTURE_2D);
245 gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR);
246 gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER, gl.LINEAR_MIPMAP_LINEAR);
247
248 var image1 = new Image();
249 image1.src = "images/crate.bmp"
250
251 var texture1 = gl.createTexture();
252 gl.activeTexture(gl.TEXTURE1);
253 gl.bindTexture(gl.TEXTURE_2D, texture1);
254
255 gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGB, gl.RGB, gl.UNSIGNED_BYTE, image1);
256 gl.generateMipmap(gl.TEXTURE_2D);
257 gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.LINEAR_MIPMAP_LINEAR);
258 gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER, gl.LINEAR_MIPMAP_LINEAR);
259 }
260
261 function render() {
262 gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
263
264 theta += 2.0;
⊗ 0 ⚠ 0
Ln 263, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene2.js - Visual Studio Code

scene2.html scene2.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > render

```
265
266     at[0] = eye[0] + cameraVec[0];
267     at[1] = eye[1] + cameraVec[1];
268     at[2] = eye[2] + cameraVec[2];
269     var viewMatrix = lookAt(eye, at, up);
270
271     var colorLoc = gl.getUniformLocation(program0, "uColor");
272
273     // draw the ground
274     gl.uniform4f(colorLoc, 0.8, 0.8, 0.8, 1.0); // gray
275
276     modelViewMatrix = mult(viewMatrix, trballMatrix);
277     gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
278     gl.drawArrays(gl.TRIANGLES, vertGroundStart, vertGroundEnd);
279
280     for (var z=-5; z<5; z+=3) {
281         // draw a cube
282         gl.uniform4f(colorLoc, 1.0, 0.0, 0.0, 1.0); // red
283
284         var rMatrix = mult(rotateY(theta), rotateZ(45));
285         var modelMatrix = mult(translate(-3, 1.3, z), rMatrix);
286         modelMatrix = mult(trballMatrix, modelMatrix);
287         modelViewMatrix = mult(viewMatrix, modelMatrix);
288         gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
289         gl.drawArrays(gl.TRIANGLES, vertCubeStart, vertCubeEnd);
290
291         modelMatrix = mult(translate(3, 1.3, z), rMatrix);
292         modelMatrix = mult(trballMatrix, modelMatrix);
293         modelViewMatrix = mult(viewMatrix, modelMatrix);
294         gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
295         gl.drawArrays(gl.TRIANGLES, vertCubeStart, vertCubeEnd);
296
297         // draw a hexa-pyramid
```

⊗ 0 ⚠ 0 Ln 296, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

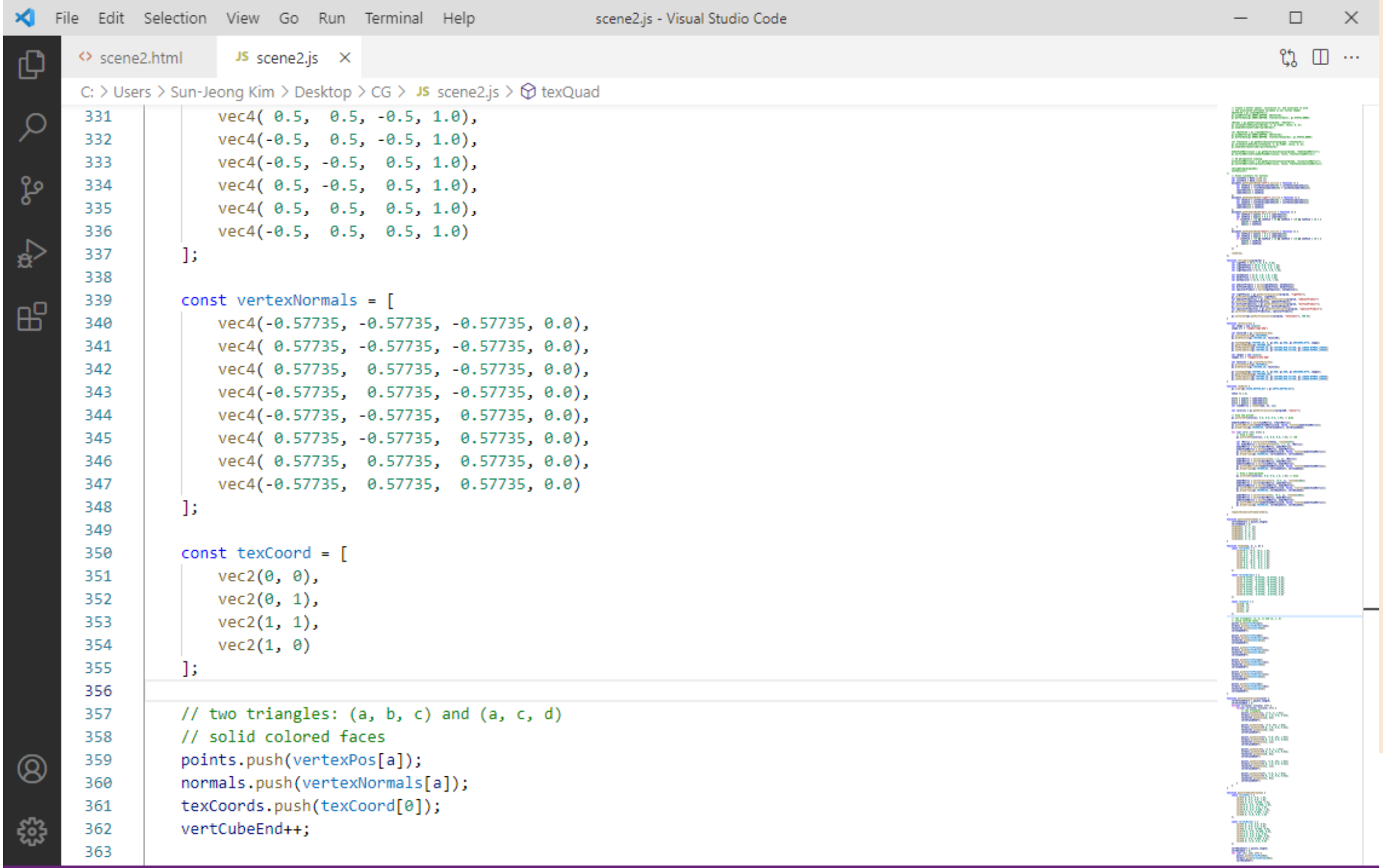
File Edit Selection View Go Run Terminal Help scene2.js - Visual Studio Code

scene2.html scene2.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > ...

```
298             gl.uniform4f(colorLoc, 0.0, 0.0, 1.0, 1.0); // blue
299
300             modelMatrix = mult(translate(-3, -0.5, z), rotateZ(180));
301             modelMatrix = mult(trballMatrix, modelMatrix);
302             modelViewMatrix = mult(viewMatrix, modelMatrix);
303             gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
304             gl.drawArrays(gl.TRIANGLES, vertHexaStart, vertHexaEnd);
305
306             modelMatrix = mult(translate(3, -0.5, z), rotateZ(180));
307             modelMatrix = mult(trballMatrix, modelMatrix);
308             modelViewMatrix = mult(viewMatrix, modelMatrix);
309             gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
310             gl.drawArrays(gl.TRIANGLES, vertHexaStart, vertHexaEnd);
311         }
312
313         requestAnimationFrame(render);
314     }
315
316     function generateTexCube() {
317         vertCubeStart = points.length;
318         vertCubeEnd = 0;
319         texQuad(1, 0, 3, 2);
320         texQuad(2, 3, 7, 6);
321         texQuad(3, 0, 4, 7);
322         texQuad(4, 5, 6, 7);
323         texQuad(5, 4, 0, 1);
324         texQuad(6, 5, 1, 2);
325     }
326
327     function texQuad(a, b, c, d) {
328         const vertexPos = [
329             vec4(-0.5, -0.5, -0.5, 1.0),
330             vec4( 0.5, -0.5, -0.5, 1.0),
```

⊗ 0 ⚠ 0 Ln 326, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

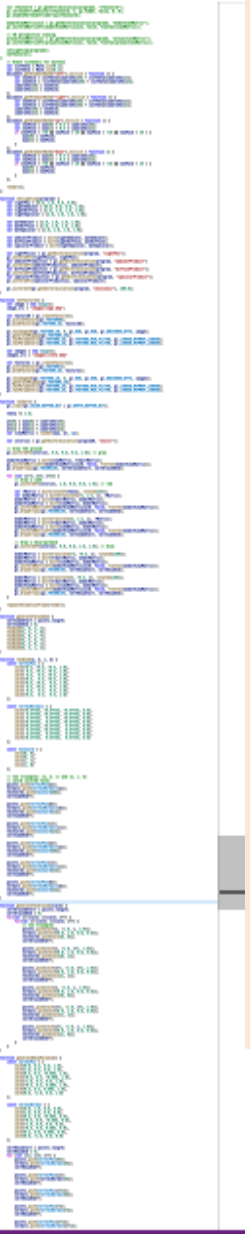

File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > texQuad
vec4(0.5, 0.5, -0.5, 1.0),
vec4(-0.5, 0.5, -0.5, 1.0),
vec4(-0.5, -0.5, 0.5, 1.0),
vec4(0.5, -0.5, 0.5, 1.0),
vec4(0.5, 0.5, 0.5, 1.0),
vec4(-0.5, 0.5, 0.5, 1.0)
];
const vertexNormals = [
vec4(-0.57735, -0.57735, -0.57735, 0.0),
vec4(0.57735, -0.57735, -0.57735, 0.0),
vec4(0.57735, 0.57735, -0.57735, 0.0),
vec4(-0.57735, 0.57735, -0.57735, 0.0),
vec4(-0.57735, -0.57735, 0.57735, 0.0),
vec4(0.57735, -0.57735, 0.57735, 0.0),
vec4(0.57735, 0.57735, 0.57735, 0.0),
vec4(-0.57735, 0.57735, 0.57735, 0.0)
];
const texCoord = [
vec2(0, 0),
vec2(0, 1),
vec2(1, 1),
vec2(1, 0)
];
// two triangles: (a, b, c) and (a, c, d)
// solid colored faces
points.push(vertexPos[a]);
normals.push(vertexNormals[a]);
texCoords.push(texCoord[0]);
vertCubeEnd++;
0 0
Ln 356, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene2.js - Visual Studio Code

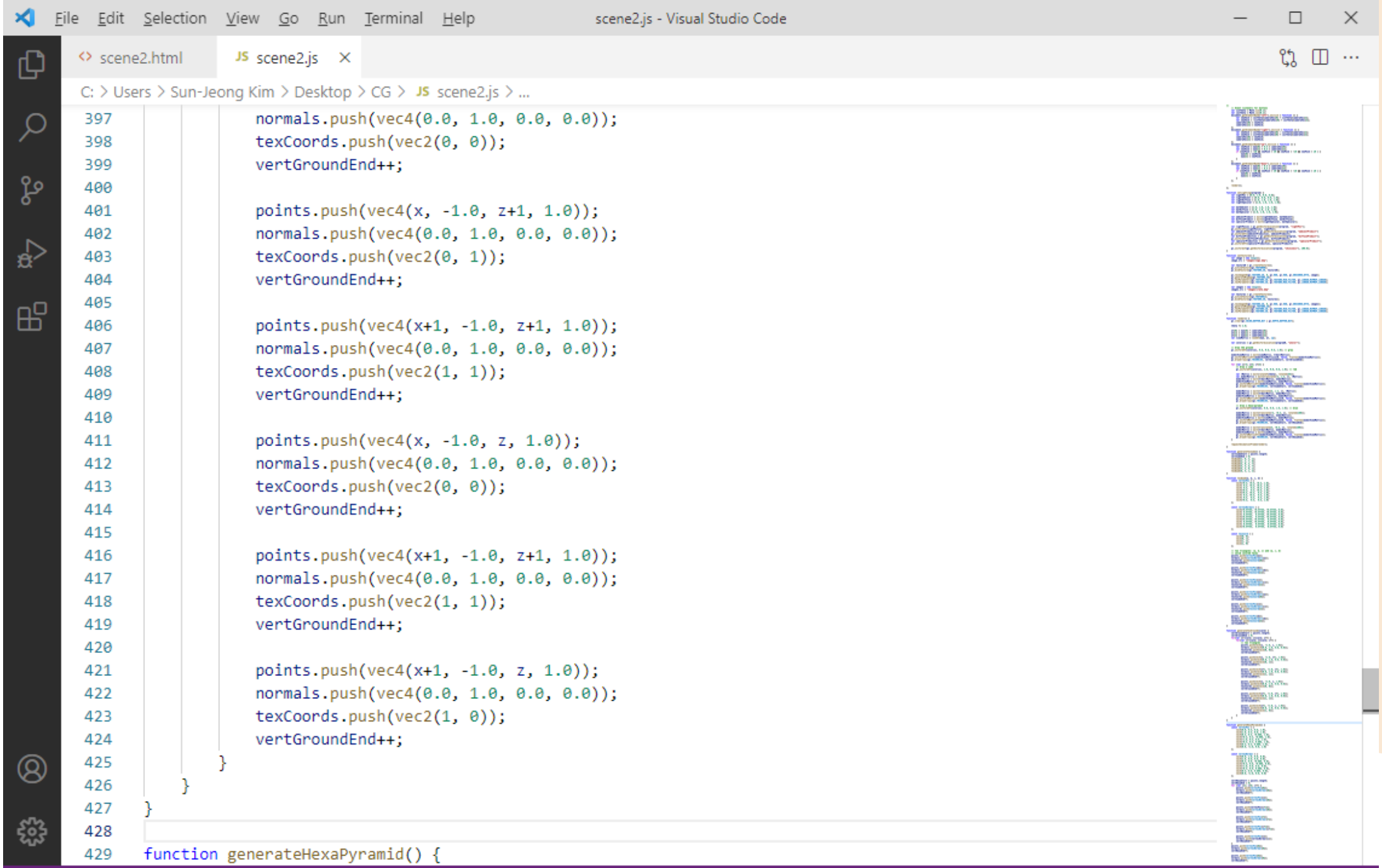
scene2.html scene2.js

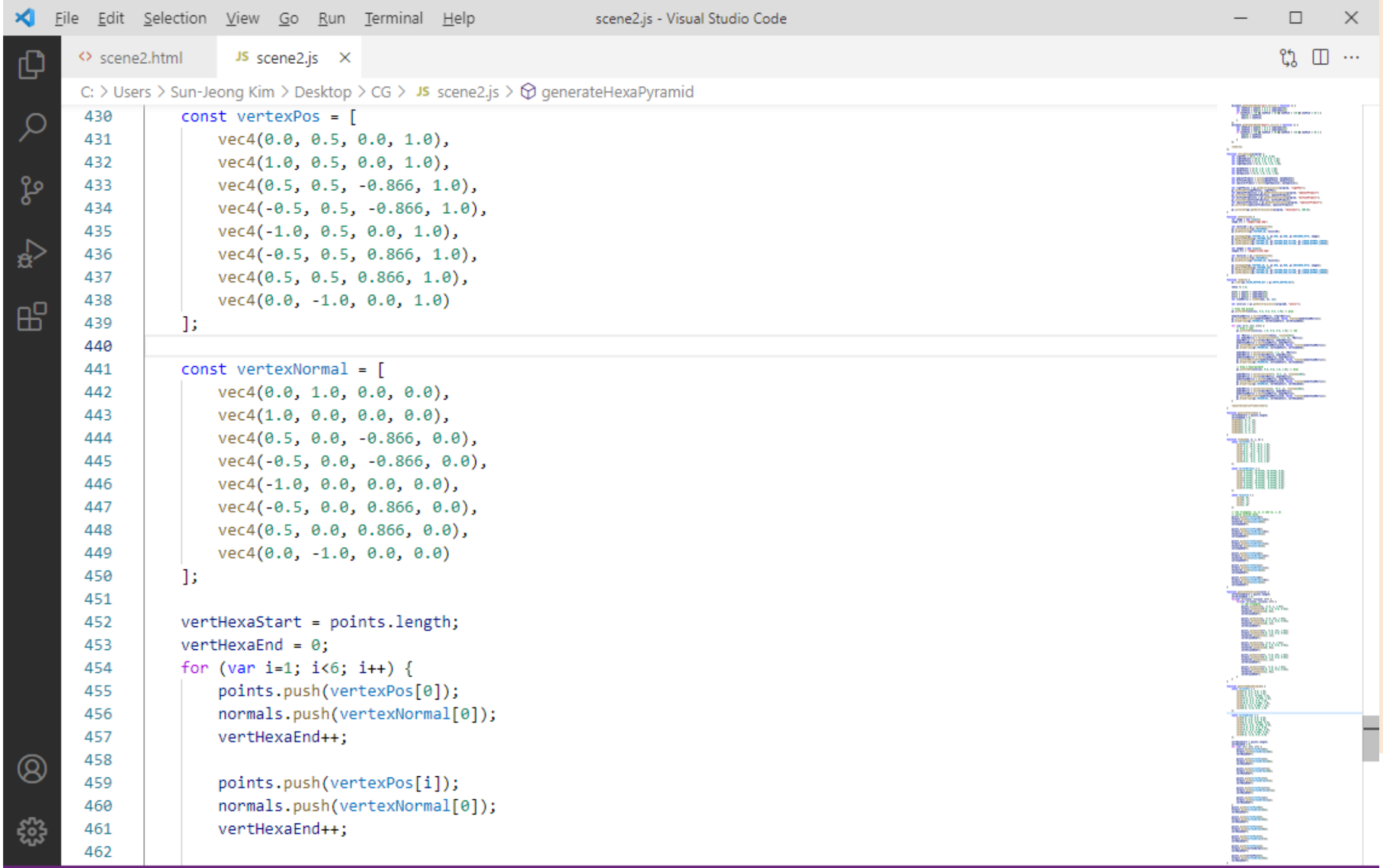
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > ...

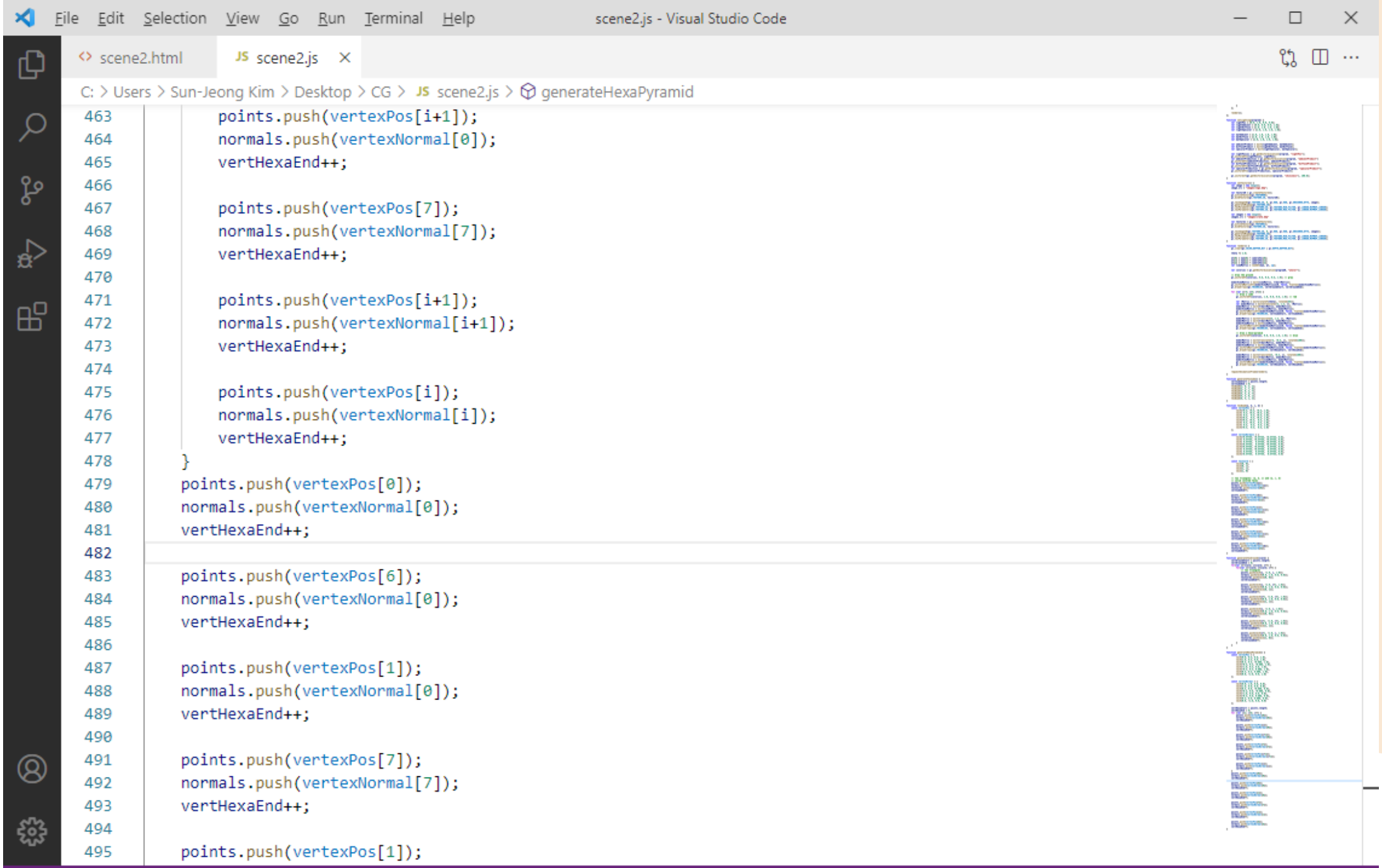
```
364         points.push(vertexPos[b]);
365         normals.push(vertexNormals[b]);
366         texCoords.push(texCoord[1]);
367         vertCubeEnd++;
368
369         points.push(vertexPos[c]);
370         normals.push(vertexNormals[c]);
371         texCoords.push(texCoord[2]);
372         vertCubeEnd++;
373
374         points.push(vertexPos[a]);
375         normals.push(vertexNormals[a]);
376         texCoords.push(texCoord[0]);
377         vertCubeEnd++;
378
379         points.push(vertexPos[c]);
380         normals.push(vertexNormals[c]);
381         texCoords.push(texCoord[2]);
382         vertCubeEnd++;
383
384         points.push(vertexPos[d]);
385         normals.push(vertexNormals[d]);
386         texCoords.push(texCoord[3]);
387         vertCubeEnd++;
388     }
389
390 function generateTexGround(scale) {
391     vertGroundStart = points.length;
392     vertGroundEnd = 0;
393     for(var x=-scale; x<scale; x++) {
394         for(var z=-scale; z<scale; z++) {
395             // two triangles
396             points.push(vec4(x, -1.0, z, 1.0));
```

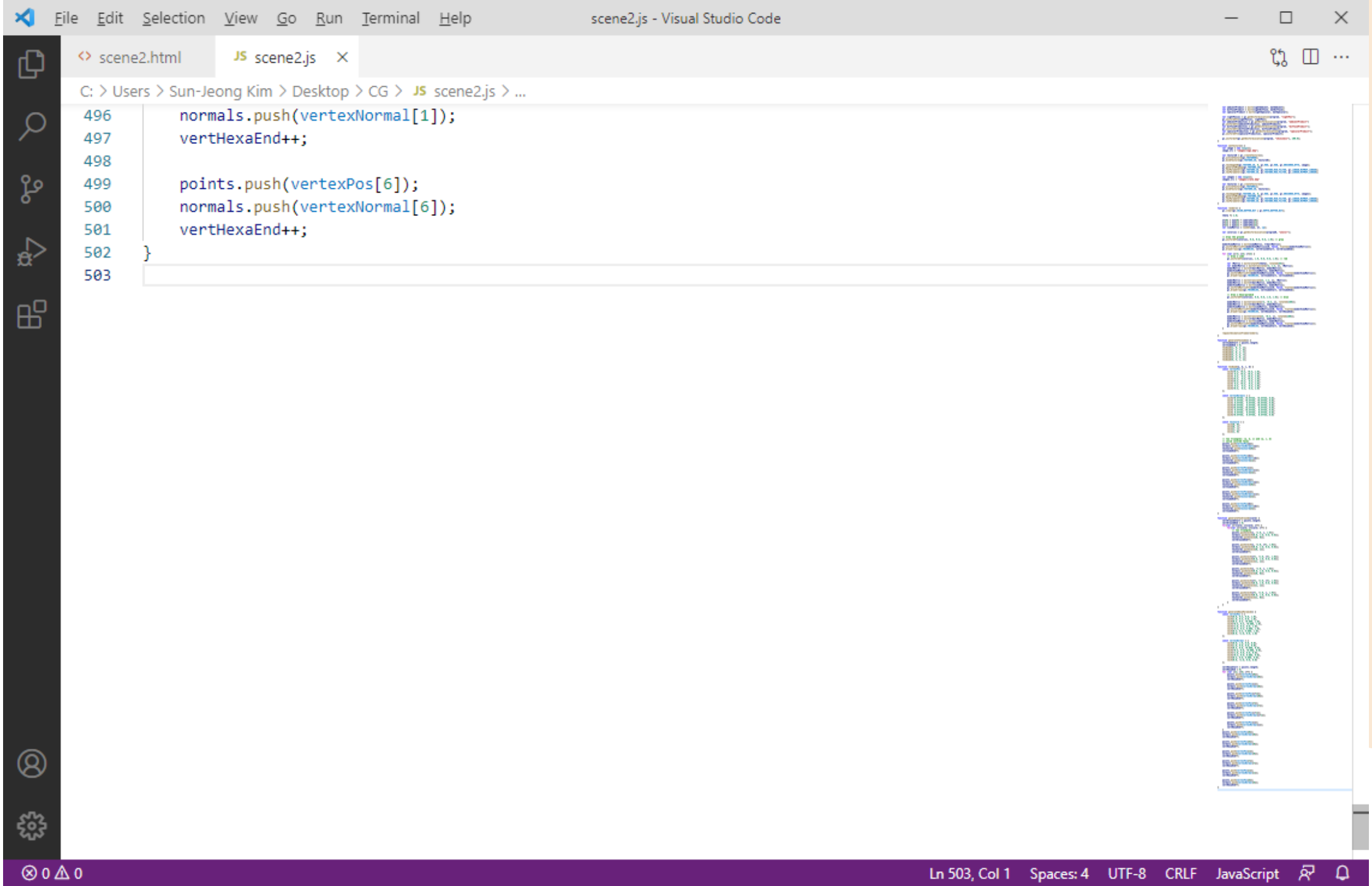


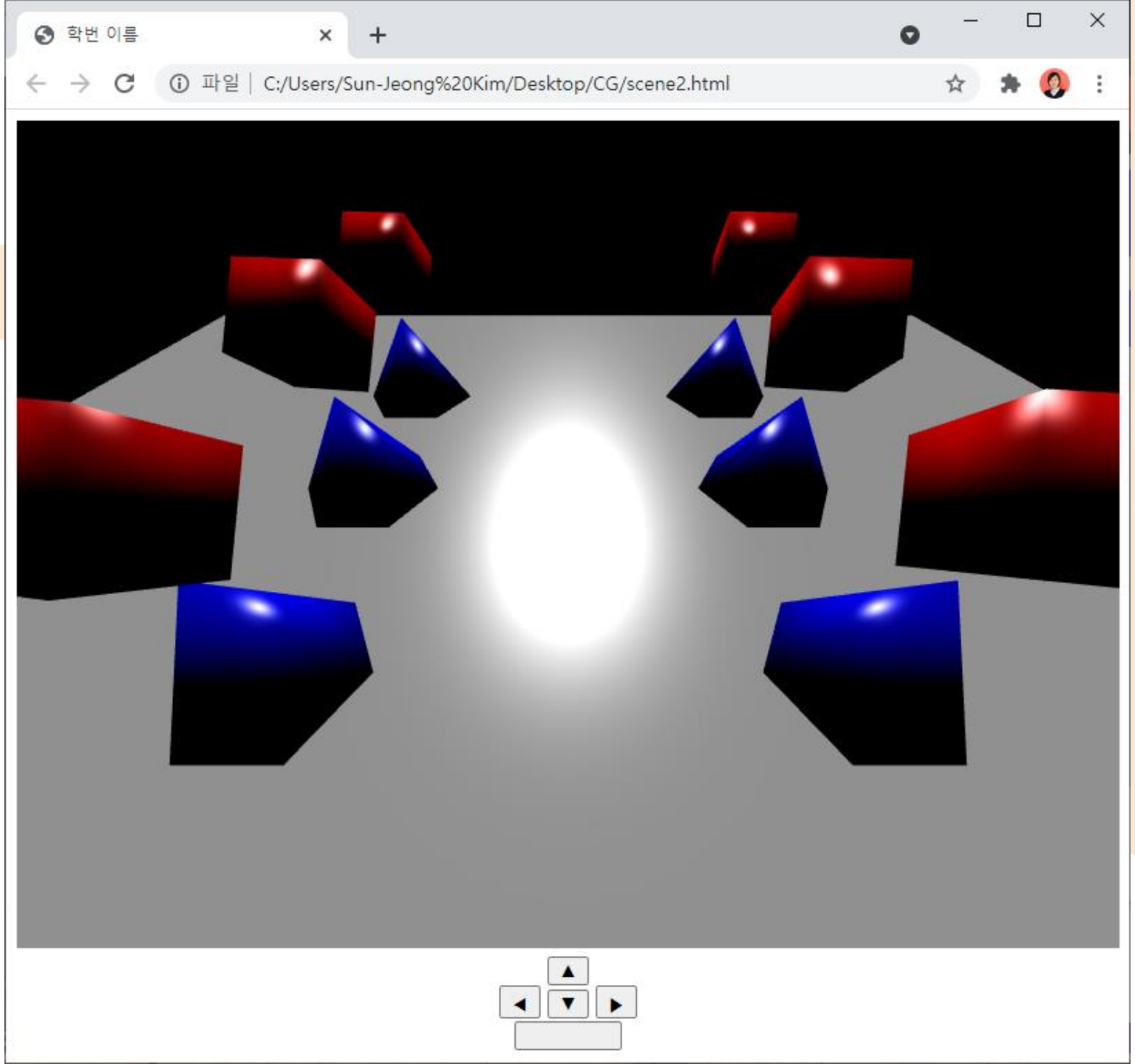
⊗ 0 ⚠ 0 Ln 389, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

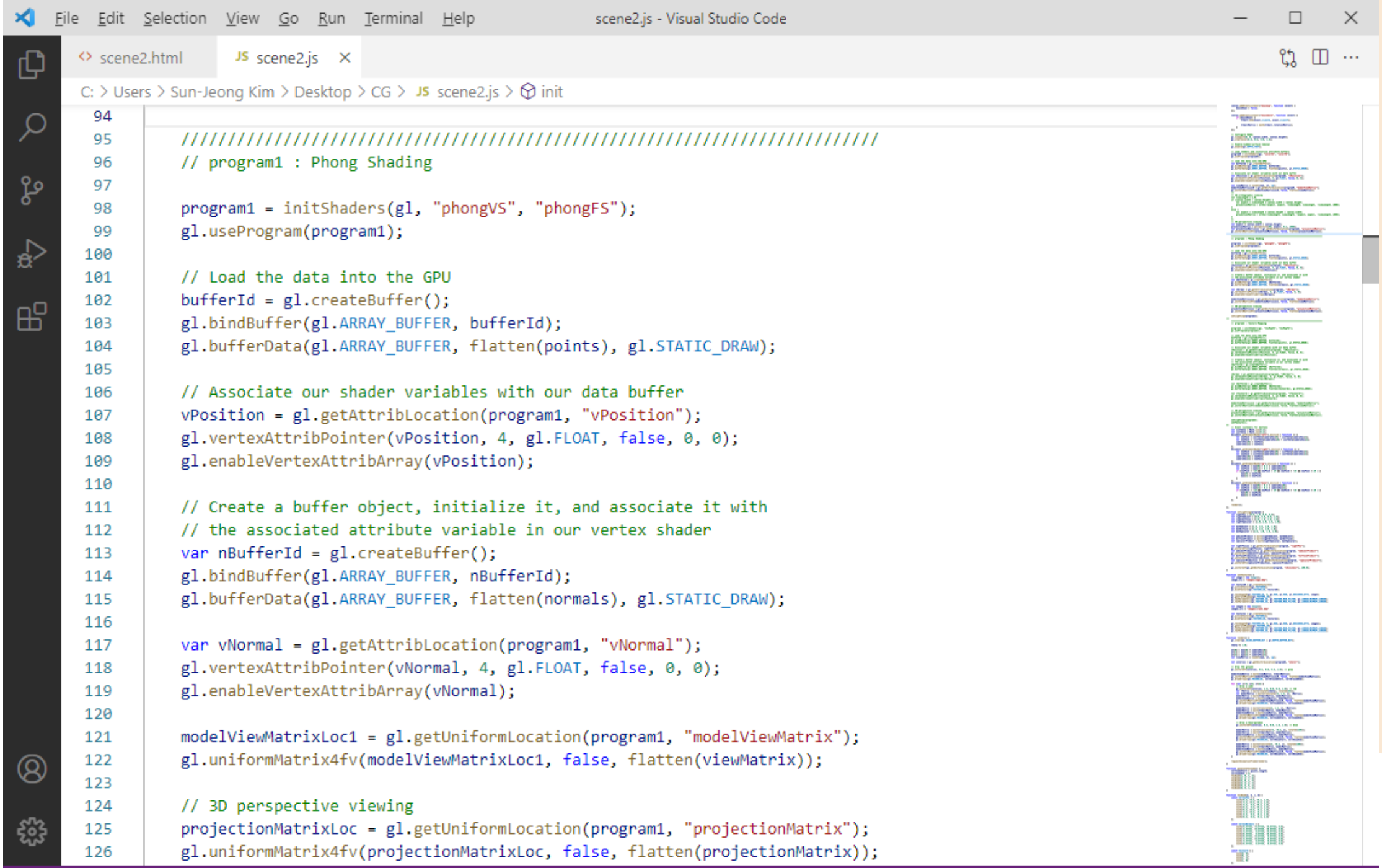

File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > ...
397 normals.push(vec4(0.0, 1.0, 0.0, 0.0));
398 texCoords.push(vec2(0, 0));
399 vertGroundEnd++;
400
401 points.push(vec4(x, -1.0, z+1, 1.0));
402 normals.push(vec4(0.0, 1.0, 0.0, 0.0));
403 texCoords.push(vec2(0, 1));
404 vertGroundEnd++;
405
406 points.push(vec4(x+1, -1.0, z+1, 1.0));
407 normals.push(vec4(0.0, 1.0, 0.0, 0.0));
408 texCoords.push(vec2(1, 1));
409 vertGroundEnd++;
410
411 points.push(vec4(x, -1.0, z, 1.0));
412 normals.push(vec4(0.0, 1.0, 0.0, 0.0));
413 texCoords.push(vec2(0, 0));
414 vertGroundEnd++;
415
416 points.push(vec4(x+1, -1.0, z+1, 1.0));
417 normals.push(vec4(0.0, 1.0, 0.0, 0.0));
418 texCoords.push(vec2(1, 1));
419 vertGroundEnd++;
420
421 points.push(vec4(x+1, -1.0, z, 1.0));
422 normals.push(vec4(0.0, 1.0, 0.0, 0.0));
423 texCoords.push(vec2(1, 0));
424 vertGroundEnd++;
425 }
426 }
427 }
428
429 function generateHexaPyramid() {
0 0
Ln 428, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

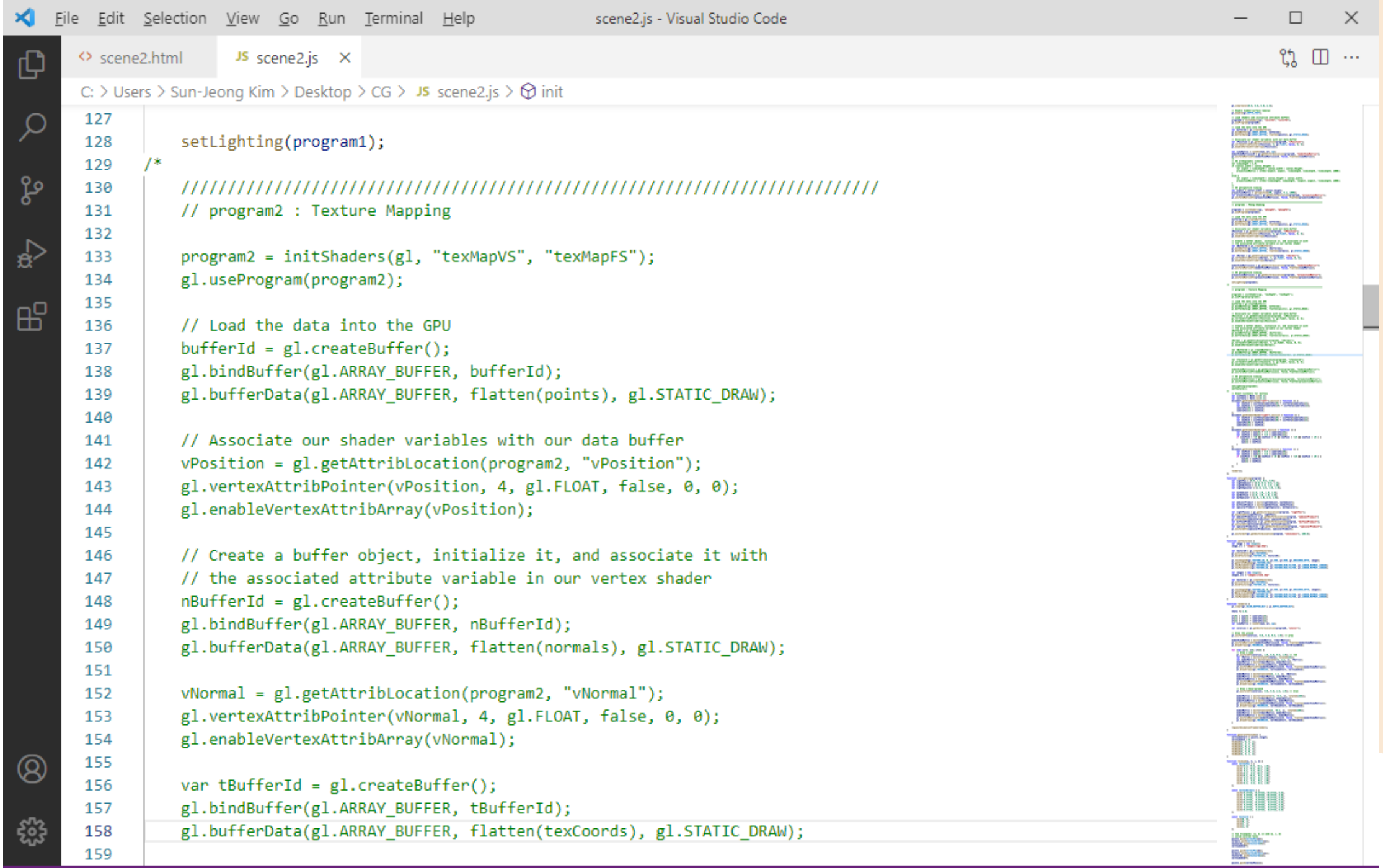

File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > generateHexaPyramid
430 const vertexPos = [
431 vec4(0.0, 0.5, 0.0, 1.0),
432 vec4(1.0, 0.5, 0.0, 1.0),
433 vec4(0.5, 0.5, -0.866, 1.0),
434 vec4(-0.5, 0.5, -0.866, 1.0),
435 vec4(-1.0, 0.5, 0.0, 1.0),
436 vec4(-0.5, 0.5, 0.866, 1.0),
437 vec4(0.5, 0.5, 0.866, 1.0),
438 vec4(0.0, -1.0, 0.0, 1.0)
439];
440
441 const vertexNormal = [
442 vec4(0.0, 1.0, 0.0, 0.0),
443 vec4(1.0, 0.0, 0.0, 0.0),
444 vec4(0.5, 0.0, -0.866, 0.0),
445 vec4(-0.5, 0.0, -0.866, 0.0),
446 vec4(-1.0, 0.0, 0.0, 0.0),
447 vec4(-0.5, 0.0, 0.866, 0.0),
448 vec4(0.5, 0.0, 0.866, 0.0),
449 vec4(0.0, -1.0, 0.0, 0.0)
450];
451
452 vertHexaStart = points.length;
453 vertHexaEnd = 0;
454 for (var i=1; i<6; i++) {
455 points.push(vertexPos[0]);
456 normals.push(vertexNormal[0]);
457 vertHexaEnd++;
458
459 points.push(vertexPos[i]);
460 normals.push(vertexNormal[0]);
461 vertHexaEnd++;
462
⊗ 0 ⚠ 0
Ln 440, Col 1 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > generateHexaPyramid
463 points.push(vertexPos[i+1]);
464 normals.push(vertexNormal[0]);
465 vertHexaEnd++;
466
467 points.push(vertexPos[7]);
468 normals.push(vertexNormal[7]);
469 vertHexaEnd++;
470
471 points.push(vertexPos[i+1]);
472 normals.push(vertexNormal[i+1]);
473 vertHexaEnd++;
474
475 points.push(vertexPos[i]);
476 normals.push(vertexNormal[i]);
477 vertHexaEnd++;
478 }
479 points.push(vertexPos[0]);
480 normals.push(vertexNormal[0]);
481 vertHexaEnd++;
482
483 points.push(vertexPos[6]);
484 normals.push(vertexNormal[0]);
485 vertHexaEnd++;
486
487 points.push(vertexPos[1]);
488 normals.push(vertexNormal[0]);
489 vertHexaEnd++;
490
491 points.push(vertexPos[7]);
492 normals.push(vertexNormal[7]);
493 vertHexaEnd++;
494
495 points.push(vertexPos[1]);
0 0
Ln 482, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > ...
496 normals.push(vertexNormal[1]);
497 vertHexaEnd++;
498
499 points.push(vertexPos[6]);
500 normals.push(vertexNormal[6]);
501 vertHexaEnd++;
502 }
503
0 0
Ln 503, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene2.html


File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > init
94
95 //
96 // program1 : Phong Shading
97
98 program1 = initShaders(gl, "phongVS", "phongFS");
99 gl.useProgram(program1);
100
101 // Load the data into the GPU
102 bufferId = gl.createBuffer();
103 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
104 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
105
106 // Associate our shader variables with our data buffer
107 vPosition = gl.getAttribLocation(program1, "vPosition");
108 gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
109 gl.enableVertexAttribArray(vPosition);
110
111 // Create a buffer object, initialize it, and associate it with
112 // the associated attribute variable in our vertex shader
113 var nBufferId = gl.createBuffer();
114 gl.bindBuffer(gl.ARRAY_BUFFER, nBufferId);
115 gl.bufferData(gl.ARRAY_BUFFER, flatten(normals), gl.STATIC_DRAW);
116
117 var vNormal = gl.getAttribLocation(program1, "vNormal");
118 gl.vertexAttribPointer(vNormal, 4, gl.FLOAT, false, 0, 0);
119 gl.enableVertexAttribArray(vNormal);
120
121 modelViewMatrixLoc1 = gl.getUniformLocation(program1, "modelViewMatrix");
122 gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(viewMatrix));
123
124 // 3D perspective viewing
125 projectionMatrixLoc = gl.getUniformLocation(program1, "projectionMatrix");
126 gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projectionMatrix));
⊗ 0 ⚠ 0
Ln 94, Col 1 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > init
127
128 setLighting(program1);
129 /*
130 //
131 // program2 : Texture Mapping
132
133 program2 = initShaders(gl, "texMapVS", "texMapFS");
134 gl.useProgram(program2);
135
136 // Load the data into the GPU
137 bufferId = gl.createBuffer();
138 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
139 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
140
141 // Associate our shader variables with our data buffer
142 vPosition = gl.getAttribLocation(program2, "vPosition");
143 gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
144 gl.enableVertexAttribArray(vPosition);
145
146 // Create a buffer object, initialize it, and associate it with
147 // the associated attribute variable in our vertex shader
148 nBufferId = gl.createBuffer();
149 gl.bindBuffer(gl.ARRAY_BUFFER, nBufferId);
150 gl.bufferData(gl.ARRAY_BUFFER, flatten(normals), gl.STATIC_DRAW);
151
152 vNormal = gl.getAttribLocation(program2, "vNormal");
153 gl.vertexAttribPointer(vNormal, 4, gl.FLOAT, false, 0, 0);
154 gl.enableVertexAttribArray(vNormal);
155
156 var tBufferId = gl.createBuffer();
157 gl.bindBuffer(gl.ARRAY_BUFFER, tBufferId);
158 gl.bufferData(gl.ARRAY_BUFFER, flatten(texCoords), gl.STATIC_DRAW);
159
⊗ 0 ⚠ 0
Ln 158, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

File Edit Selection View Go Run Terminal Help scene2.js - Visual Studio Code

scene2.html scene2.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > render

```
270
271     //var colorLoc = gl.getUniformLocation(program0, "uColor");
272     var diffuseProductLoc = gl.getUniformLocation(program1, "diffuseProduct");
273
274     // draw the ground
275     //gl.uniform4f(colorLoc, 0.8, 0.8, 0.8, 1.0); // gray
276     gl.uniform4f(diffuseProductLoc, 0.8, 0.8, 0.8, 1.0);
277
278     modelViewMatrix = mult(viewMatrix, trballMatrix);
279     //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
280     gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
281     gl.drawArrays(gl.TRIANGLES, vertGroundStart, vertGroundEnd);
282
283     for (var z=-5; z<5; z+=3) {
284         // draw a cube
285         //gl.uniform4f(colorLoc, 1.0, 0.0, 0.0, 1.0); // red
286         gl.uniform4f(diffuseProductLoc, 1.0, 0.0, 0.0, 1.0);
287
288         var rMatrix = mult(rotateY(theta), rotateZ(45));
289         var modelMatrix = mult(translate(-3, 1.3, z), rMatrix);
290         modelMatrix = mult(trballMatrix, modelMatrix);
291         modelViewMatrix = mult(viewMatrix, modelMatrix);
292         //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
293         gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
294         gl.drawArrays(gl.TRIANGLES, vertCubeStart, vertCubeEnd);
295
296         modelMatrix = mult(translate(3, 1.3, z), rMatrix);
297         modelMatrix = mult(trballMatrix, modelMatrix);
298         modelViewMatrix = mult(viewMatrix, modelMatrix);
299         //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
300         gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
301         gl.drawArrays(gl.TRIANGLES, vertCubeStart, vertCubeEnd);
302
```

⊗ 0 ⚠ 0 Ln 272, Col 79 Spaces: 4 UTF-8 CRLF JavaScript

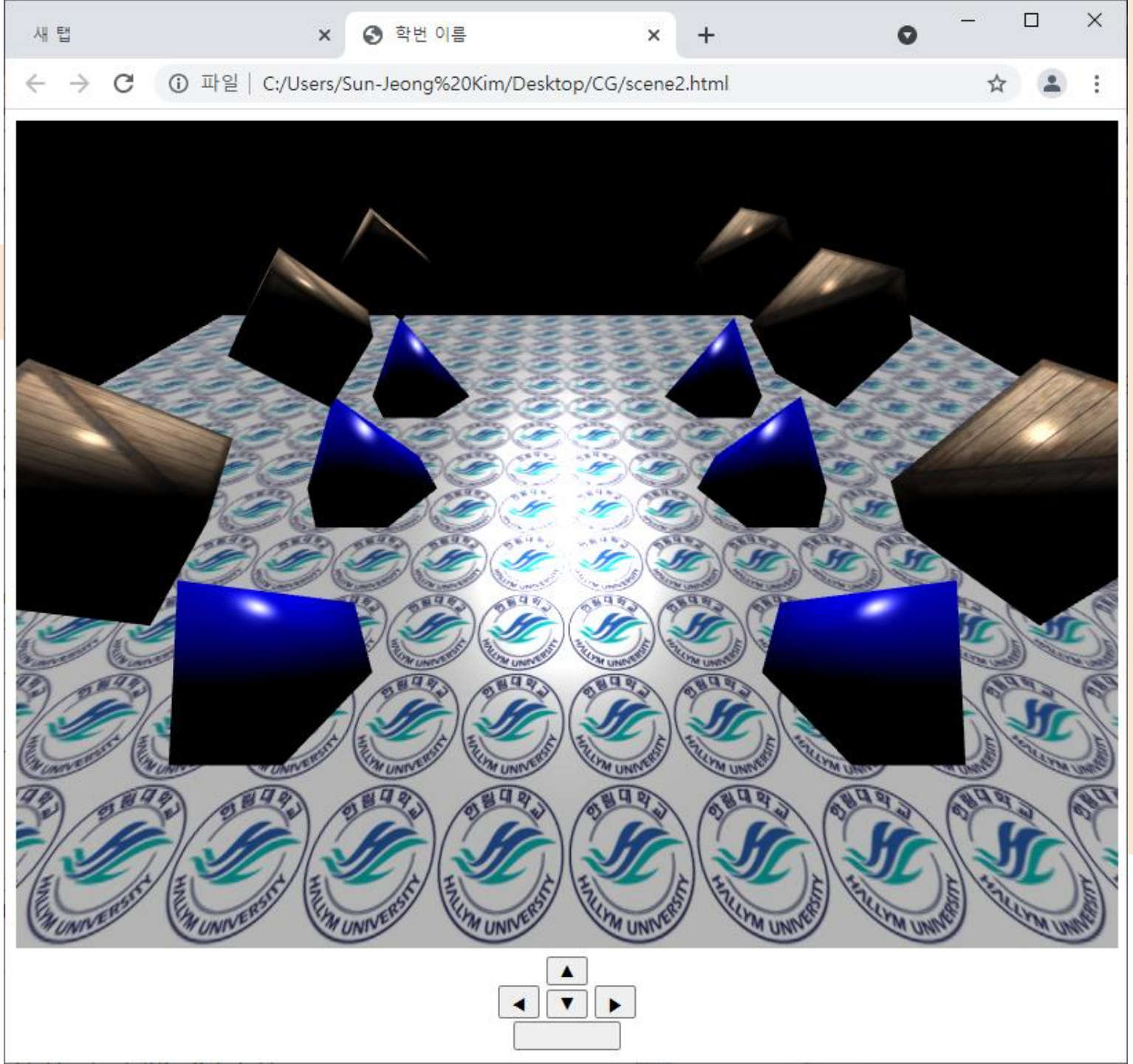
File Edit Selection View Go Run Terminal Help scene2.js - Visual Studio Code

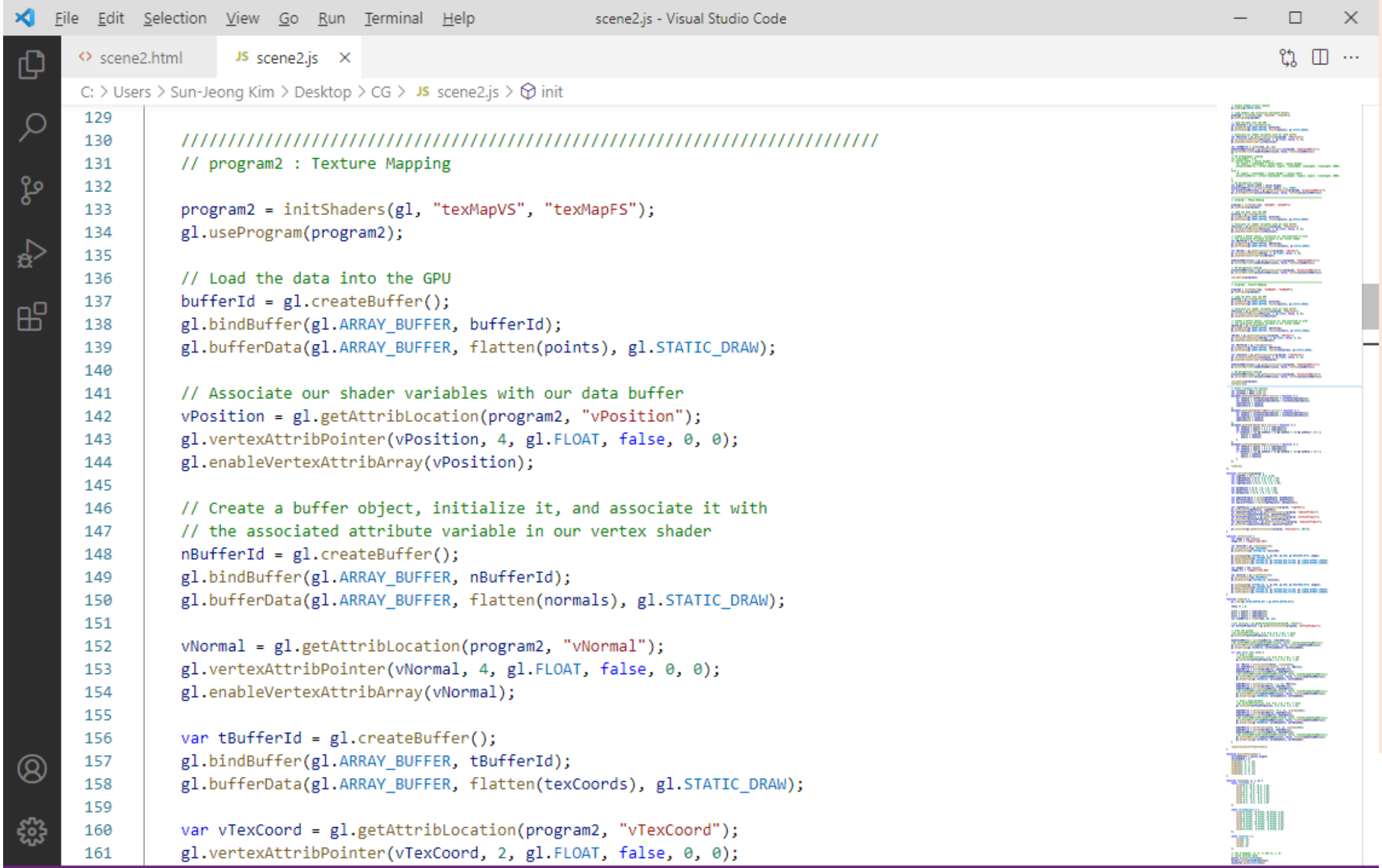
scene2.html scene2.js

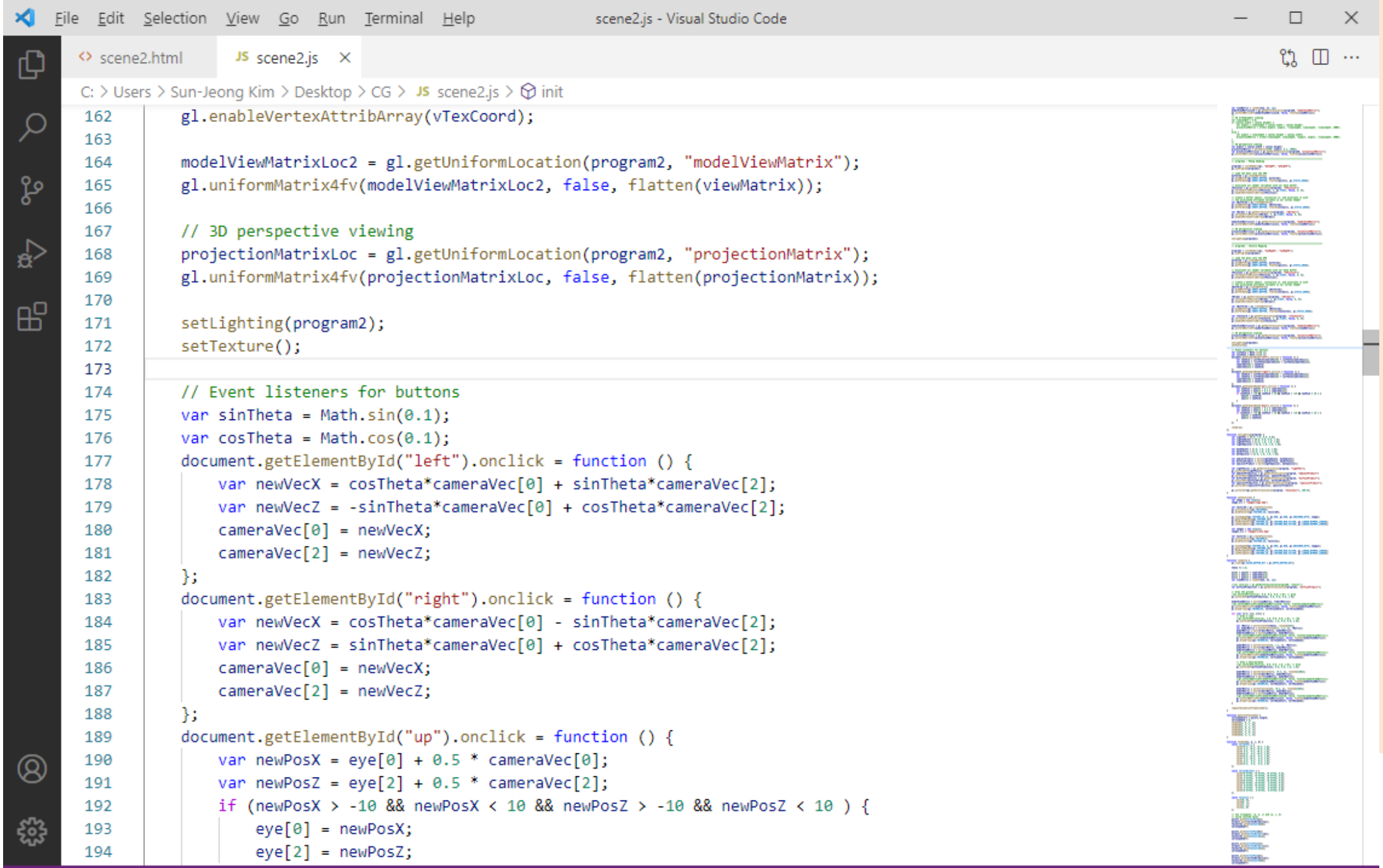
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > render

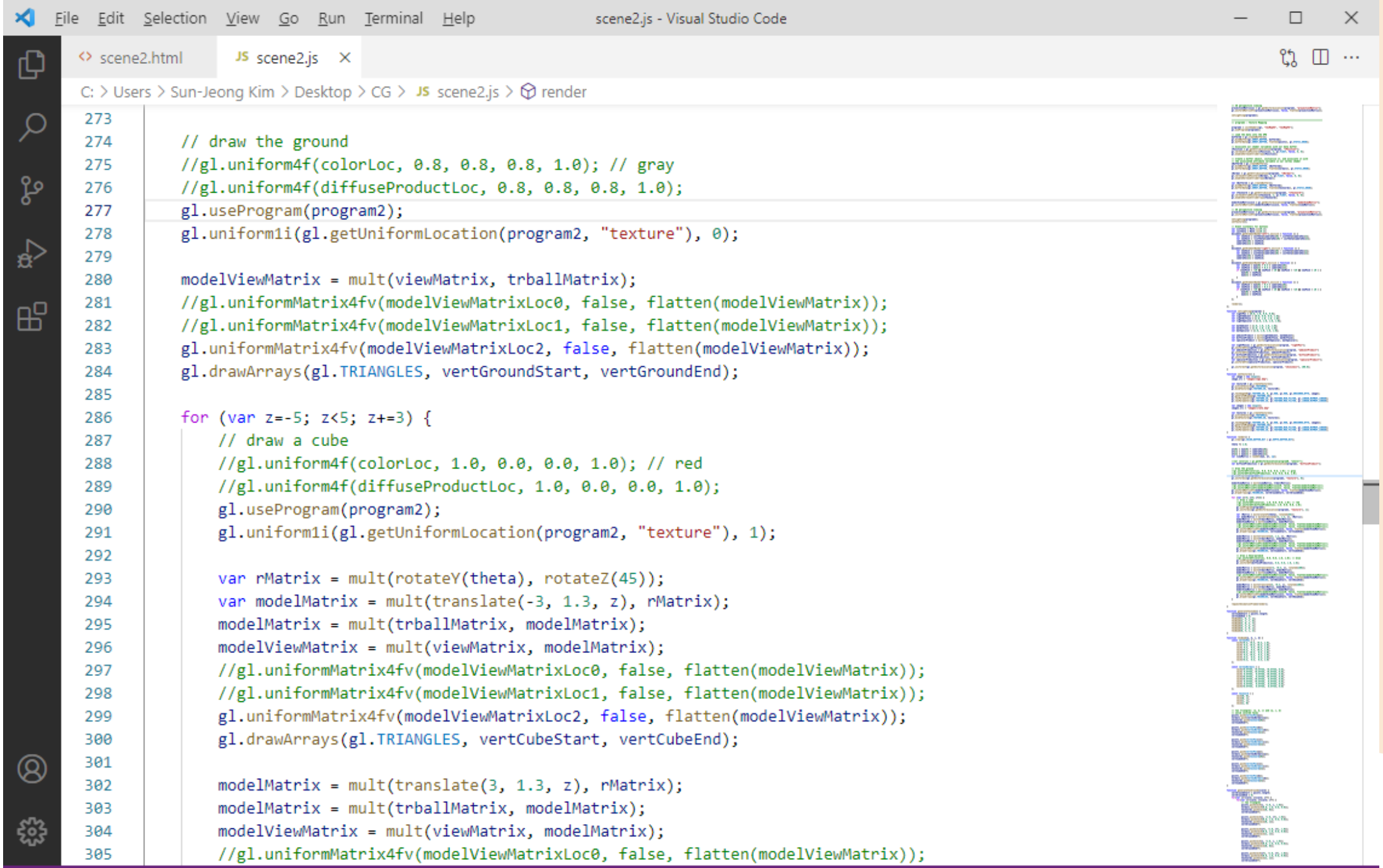
```
303         // draw a hexa-pyramid
304         //gl.uniform4f(colorLoc, 0.0, 0.0, 1.0, 1.0); // blue
305         gl.uniform4f(diffuseProductLoc, 0.0, 0.0, 1.0, 1.0);
306
307         modelMatrix = mult(translate(-3, -0.5, z), rotateZ(180));
308         modelMatrix = mult(trballMatrix, modelMatrix);
309         modelViewMatrix = mult(viewMatrix, modelMatrix);
310         //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
311         gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
312         gl.drawArrays(gl.TRIANGLES, vertHexaStart, vertHexaEnd);
313
314         modelMatrix = mult(translate(3, -0.5, z), rotateZ(180));
315         modelMatrix = mult(trballMatrix, modelMatrix);
316         modelViewMatrix = mult(viewMatrix, modelMatrix);
317         //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
318         gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
319         gl.drawArrays(gl.TRIANGLES, vertHexaStart, vertHexaEnd);
320     }
321
322     requestAnimationFrame(render);
323 }
324
325 function generateTexCube() {
326     vertCubeStart = points.length;
327     vertCubeEnd = 0;
328     texQuad(1, 0, 3, 2);
329     texQuad(2, 3, 7, 6);
330     texQuad(3, 0, 4, 7);
331     texQuad(4, 5, 6, 7);
332     texQuad(5, 4, 0, 1);
333     texQuad(6, 5, 1, 2);
334 }
335
```

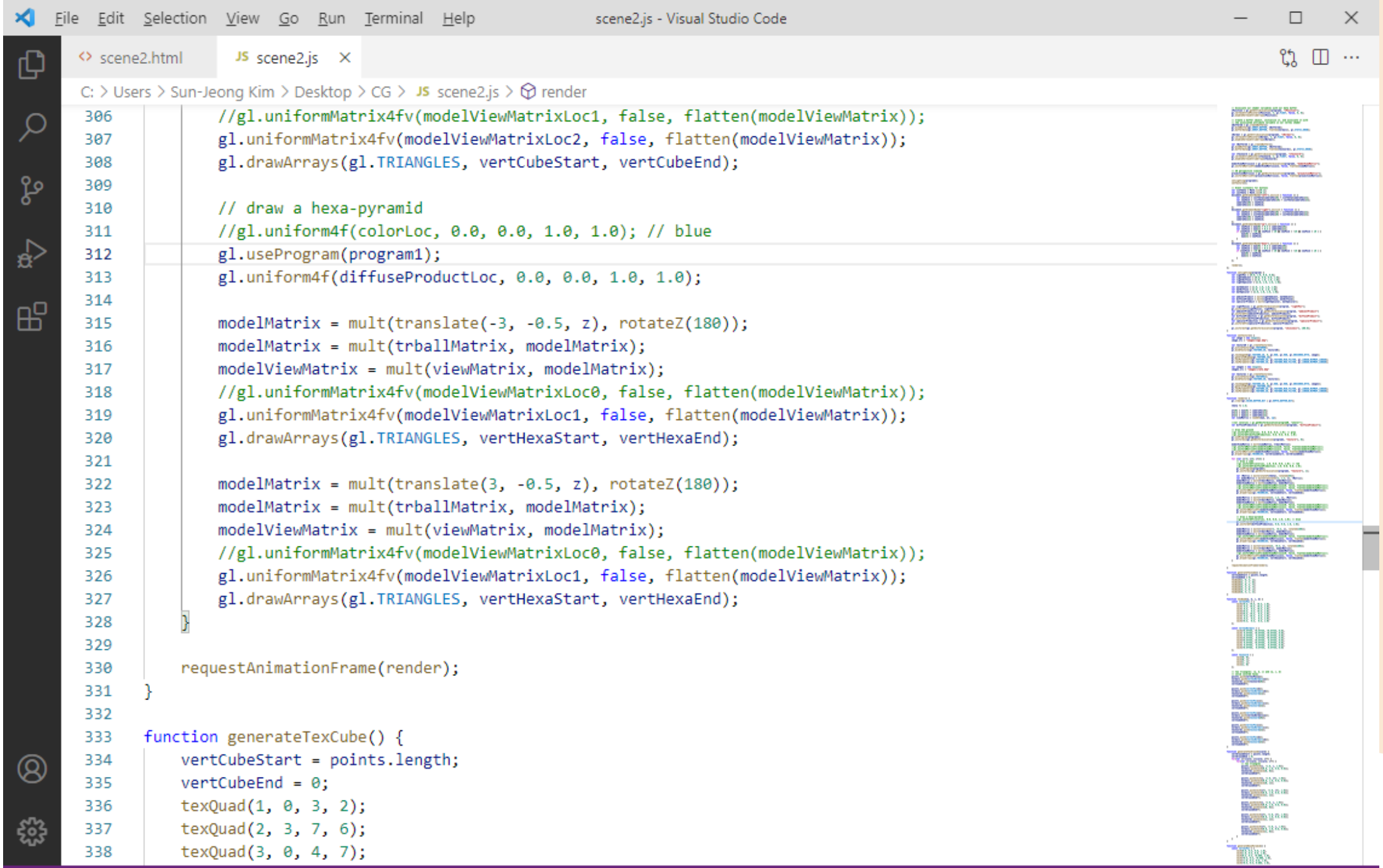
⊗ 0 ⚠ 0 Ln 318, Col 83 Spaces: 4 UTF-8 CRLF JavaScript

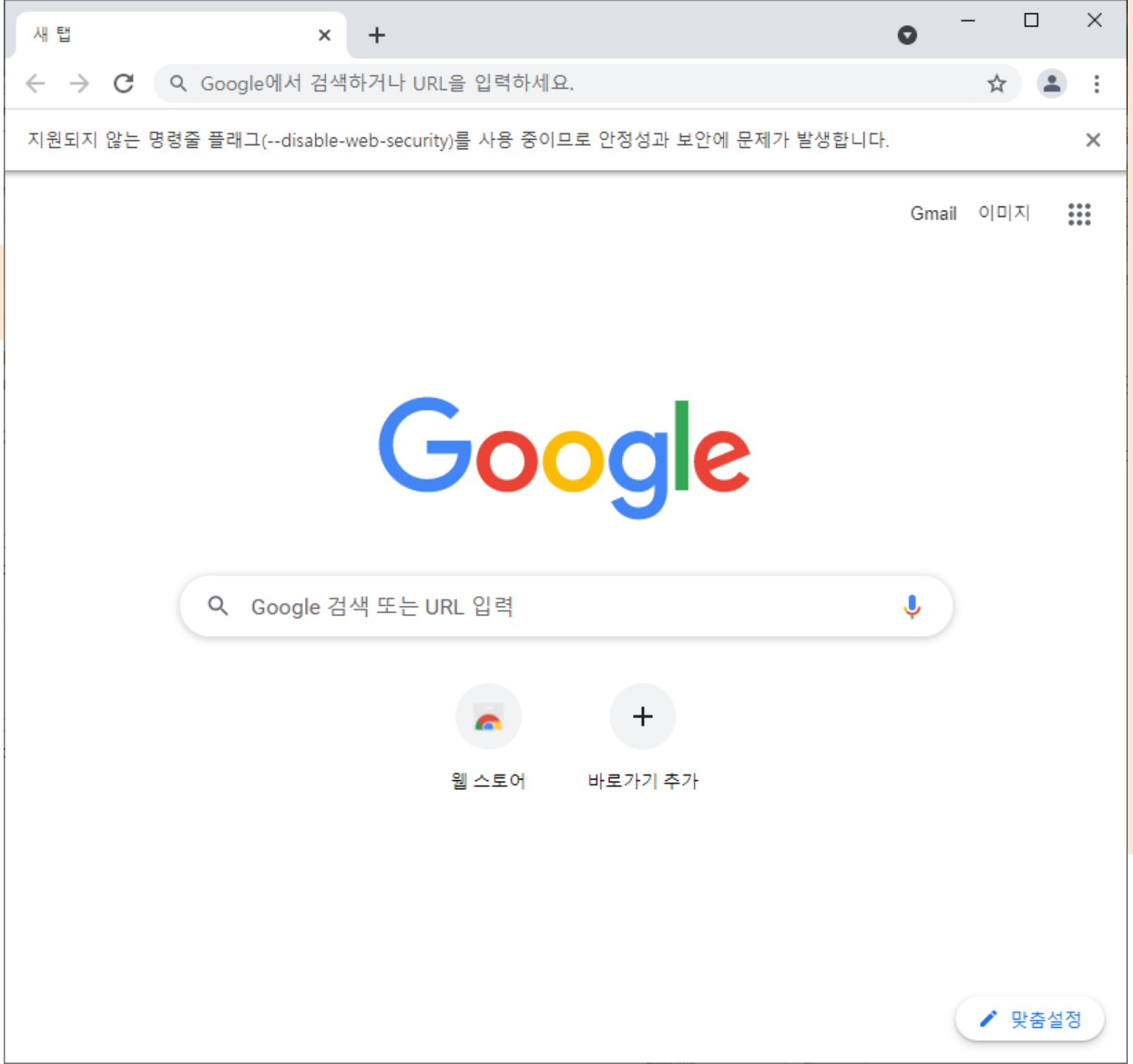

새 탭
학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene2.html
HALLYM UNIVERSITY


File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > init
129
130 //
131 // program2 : Texture Mapping
132
133 program2 = initShaders(gl, "texMapVS", "texMapFS");
134 gl.useProgram(program2);
135
136 // Load the data into the GPU
137 bufferId = gl.createBuffer();
138 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
139 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
140
141 // Associate our shader variables with our data buffer
142 vPosition = gl.getAttribLocation(program2, "vPosition");
143 gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
144 gl.enableVertexAttribArray(vPosition);
145
146 // Create a buffer object, initialize it, and associate it with
147 // the associated attribute variable in our vertex shader
148 nBufferId = gl.createBuffer();
149 gl.bindBuffer(gl.ARRAY_BUFFER, nBufferId);
150 gl.bufferData(gl.ARRAY_BUFFER, flatten(normals), gl.STATIC_DRAW);
151
152 vNormal = gl.getAttribLocation(program2, "vNormal");
153 gl.vertexAttribPointer(vNormal, 4, gl.FLOAT, false, 0, 0);
154 gl.enableVertexAttribArray(vNormal);
155
156 var tBufferId = gl.createBuffer();
157 gl.bindBuffer(gl.ARRAY_BUFFER, tBufferId);
158 gl.bufferData(gl.ARRAY_BUFFER, flatten(texCoords), gl.STATIC_DRAW);
159
160 var vTexCoord = gl.getAttribLocation(program2, "vTexCoord");
161 gl.vertexAttribPointer(vTexCoord, 2, gl.FLOAT, false, 0, 0);
⊗ 0 ⚠ 0
Ln 173, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

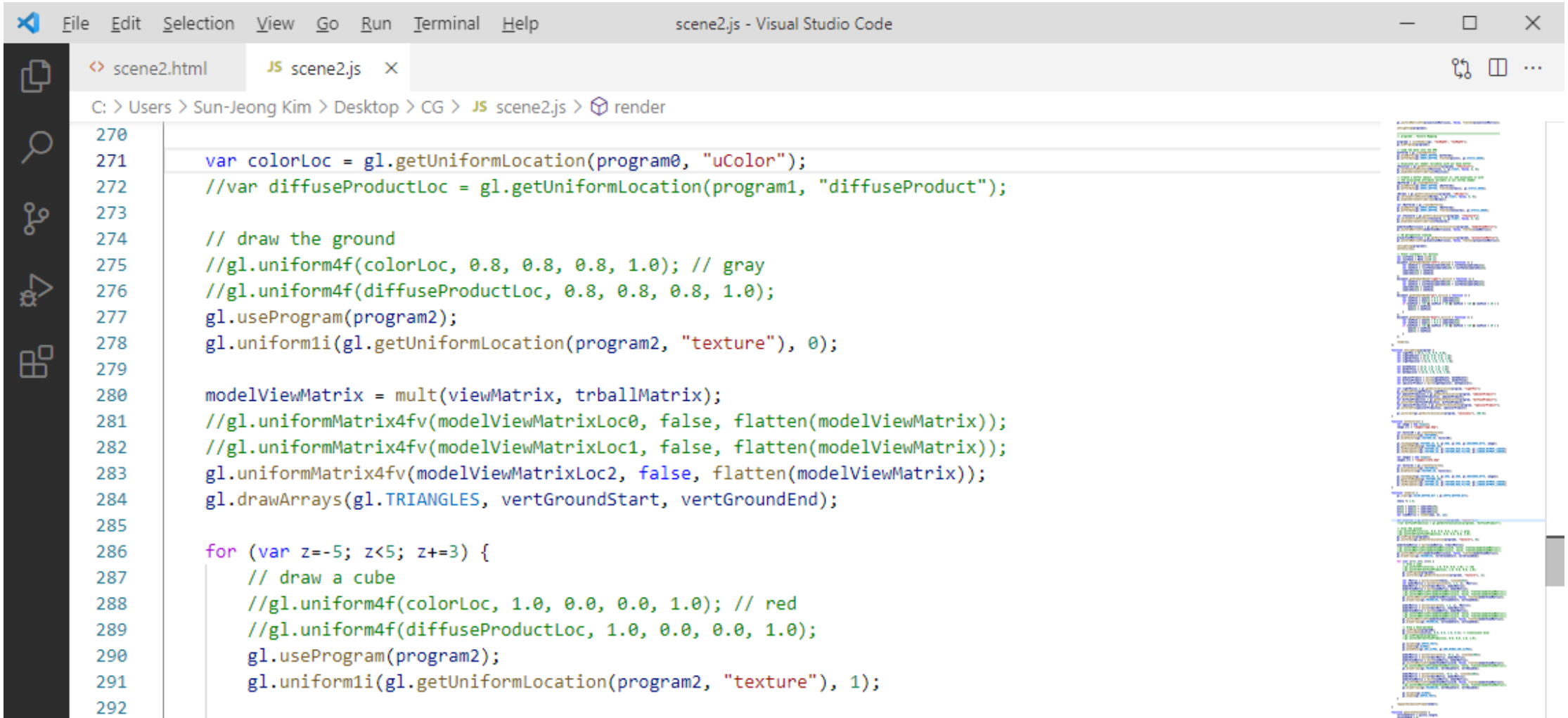

File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > init
gl.enableVertexAttribArray(vTexCoord);
modelViewMatrixLoc2 = gl.getUniformLocation(program2, "modelViewMatrix");
gl.uniformMatrix4fv(modelViewMatrixLoc2, false, flatten(viewMatrix));
// 3D perspective viewing
projectionMatrixLoc = gl.getUniformLocation(program2, "projectionMatrix");
gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projectionMatrix));
setLighting(program2);
setTexture();
// Event listeners for buttons
var sinTheta = Math.sin(0.1);
var cosTheta = Math.cos(0.1);
document.getElementById("left").onclick = function () {
var newVecX = cosTheta*cameraVec[0] + sinTheta*cameraVec[2];
var newVecZ = -sinTheta*cameraVec[0] + cosTheta*cameraVec[2];
cameraVec[0] = newVecX;
cameraVec[2] = newVecZ;
};
document.getElementById("right").onclick = function () {
var newVecX = cosTheta*cameraVec[0] - sinTheta*cameraVec[2];
var newVecZ = sinTheta*cameraVec[0] + cosTheta*cameraVec[2];
cameraVec[0] = newVecX;
cameraVec[2] = newVecZ;
};
document.getElementById("up").onclick = function () {
var newPosX = eye[0] + 0.5 * cameraVec[0];
var newPosZ = eye[2] + 0.5 * cameraVec[2];
if (newPosX > -10 && newPosX < 10 && newPosZ > -10 && newPosZ < 10) {
eye[0] = newPosX;
eye[2] = newPosZ;
0 0
Ln 173, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript

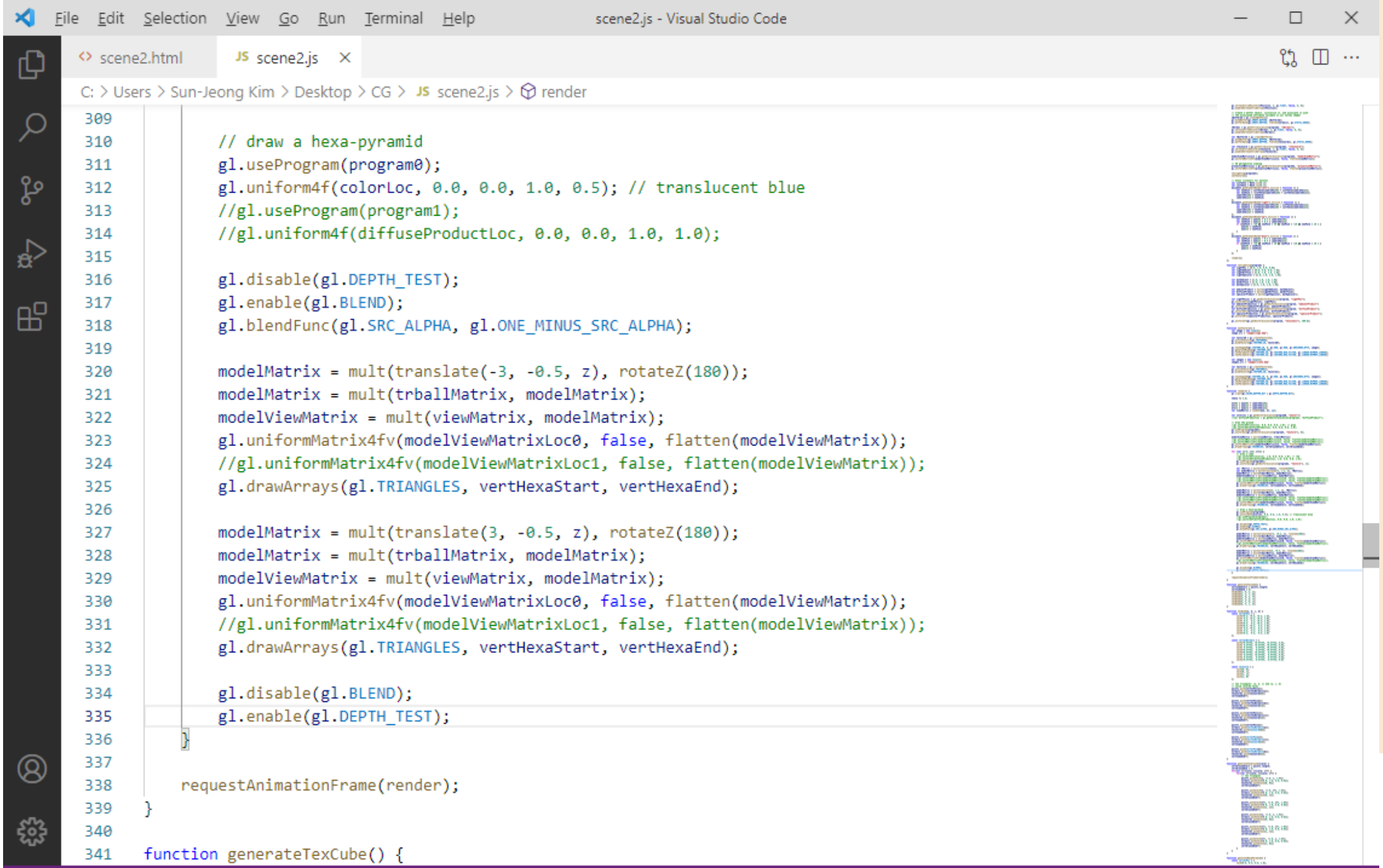

File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > render
273
274 // draw the ground
275 //gl.uniform4f(colorLoc, 0.8, 0.8, 0.8, 1.0); // gray
276 //gl.uniform4f(diffuseProductLoc, 0.8, 0.8, 0.8, 1.0);
277 gl.useProgram(program2);
278 gl.uniform1i(gl.getUniformLocation(program2, "texture"), 0);
279
280 modelViewMatrix = mult(viewMatrix, trballMatrix);
281 //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
282 //gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
283 gl.uniformMatrix4fv(modelViewMatrixLoc2, false, flatten(modelViewMatrix));
284 gl.drawArrays(gl.TRIANGLES, vertGroundStart, vertGroundEnd);
285
286 for (var z=-5; z<5; z+=3) {
287 // draw a cube
288 //gl.uniform4f(colorLoc, 1.0, 0.0, 0.0, 1.0); // red
289 //gl.uniform4f(diffuseProductLoc, 1.0, 0.0, 0.0, 1.0);
290 gl.useProgram(program2);
291 gl.uniform1i(gl.getUniformLocation(program2, "texture"), 1);
292
293 var rMatrix = mult(rotateY(theta), rotateZ(45));
294 var modelMatrix = mult(translate(-3, 1.3, z), rMatrix);
295 modelMatrix = mult(trballMatrix, modelMatrix);
296 modelViewMatrix = mult(viewMatrix, modelMatrix);
297 //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
298 //gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
299 gl.uniformMatrix4fv(modelViewMatrixLoc2, false, flatten(modelViewMatrix));
300 gl.drawArrays(gl.TRIANGLES, vertCubeStart, vertCubeEnd);
301
302 modelMatrix = mult(translate(3, 1.3, z), rMatrix);
303 modelMatrix = mult(trballMatrix, modelMatrix);
304 modelViewMatrix = mult(viewMatrix, modelMatrix);
305 //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
⊗ 0 ⚠ 0
Ln 277, Col 29 Spaces: 4 UTF-8 CRLF JavaScript

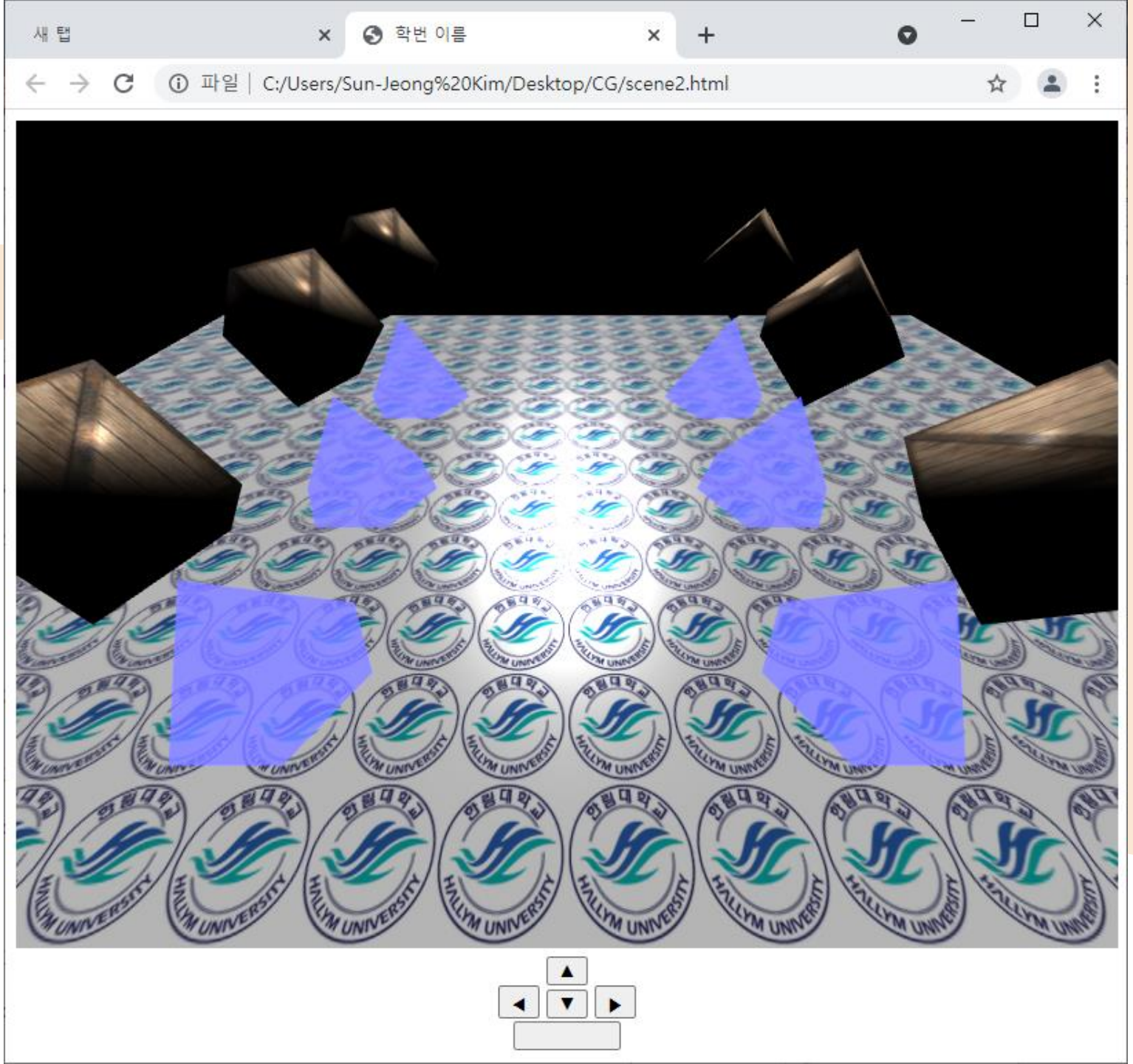

File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > render
306 //gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
307 gl.uniformMatrix4fv(modelViewMatrixLoc2, false, flatten(modelViewMatrix));
308 gl.drawArrays(gl.TRIANGLES, vertCubeStart, vertCubeEnd);
309
310 // draw a hexa-pyramid
311 //gl.uniform4f(colorLoc, 0.0, 0.0, 1.0, 1.0); // blue
312 gl.useProgram(program1);
313 gl.uniform4f(diffuseProductLoc, 0.0, 0.0, 1.0, 1.0);
314
315 modelMatrix = mult(translate(-3, -0.5, z), rotateZ(180));
316 modelMatrix = mult(trballMatrix, modelMatrix);
317 modelViewMatrix = mult(viewMatrix, modelMatrix);
318 //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
319 gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
320 gl.drawArrays(gl.TRIANGLES, vertHexaStart, vertHexaEnd);
321
322 modelMatrix = mult(translate(3, -0.5, z), rotateZ(180));
323 modelMatrix = mult(trballMatrix, modelMatrix);
324 modelViewMatrix = mult(viewMatrix, modelMatrix);
325 //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
326 gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
327 gl.drawArrays(gl.TRIANGLES, vertHexaStart, vertHexaEnd);
328 }
329
330 requestAnimationFrame(render);
331 }
332
333 function generateTexCube() {
334 vertCubeStart = points.length;
335 vertCubeEnd = 0;
336 texQuad(1, 0, 3, 2);
337 texQuad(2, 3, 7, 6);
338 texQuad(3, 0, 4, 7);
⊗ 0 ⚠ 0
Ln 312, Col 33 Spaces: 4 UTF-8 CRLF JavaScript


새 탭
Google에서 검색하거나 URL을 입력하세요.
지원되지 않는 명령줄 플래그(--disable-web-security)를 사용 중이므로 안정성과 보안에 문제가 발생합니다.
Gmail
이미지
Google
Google 검색 또는 URL 입력
웹 스토어
바로가기 추가
맞춤설정

Alpha Blending

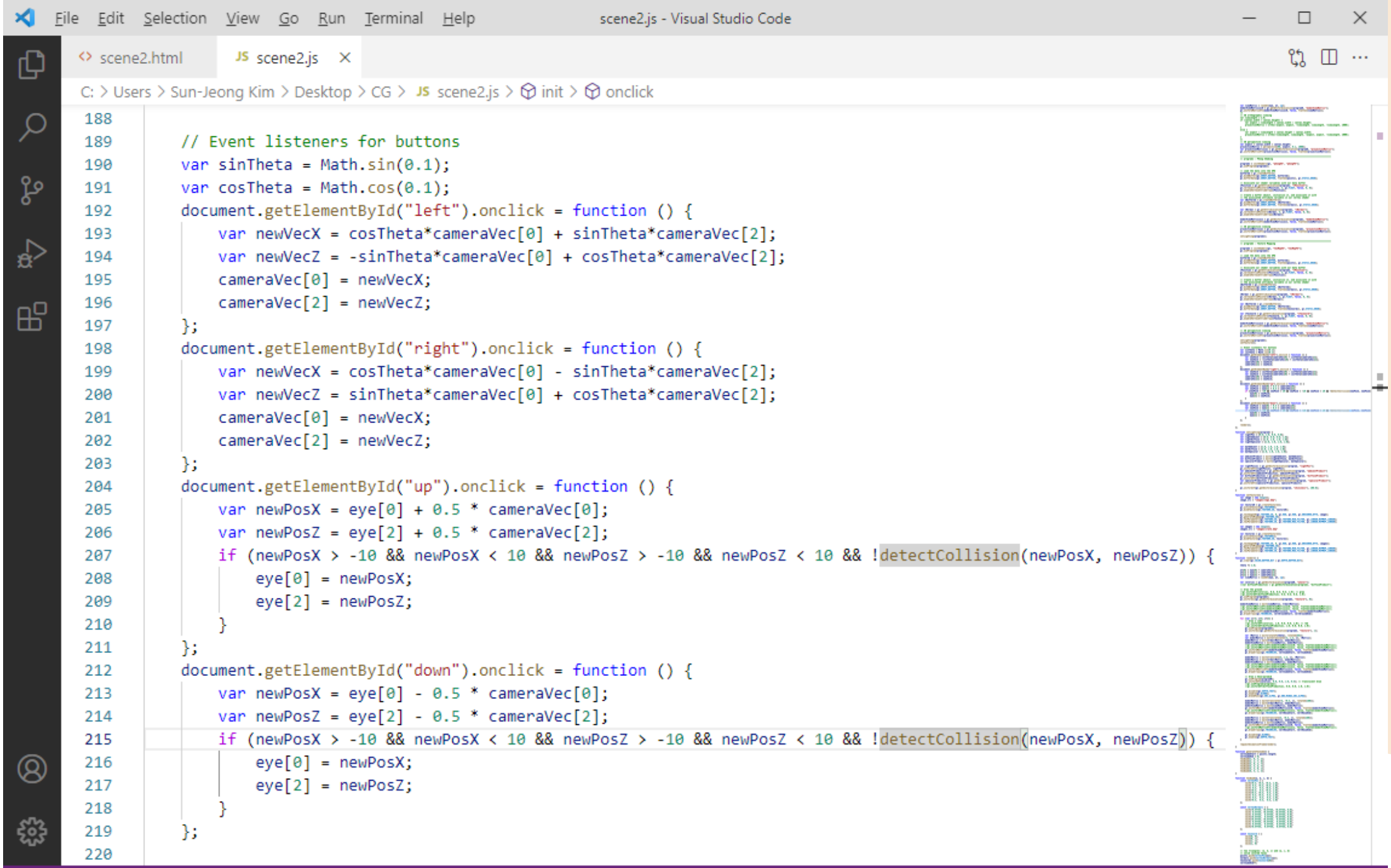


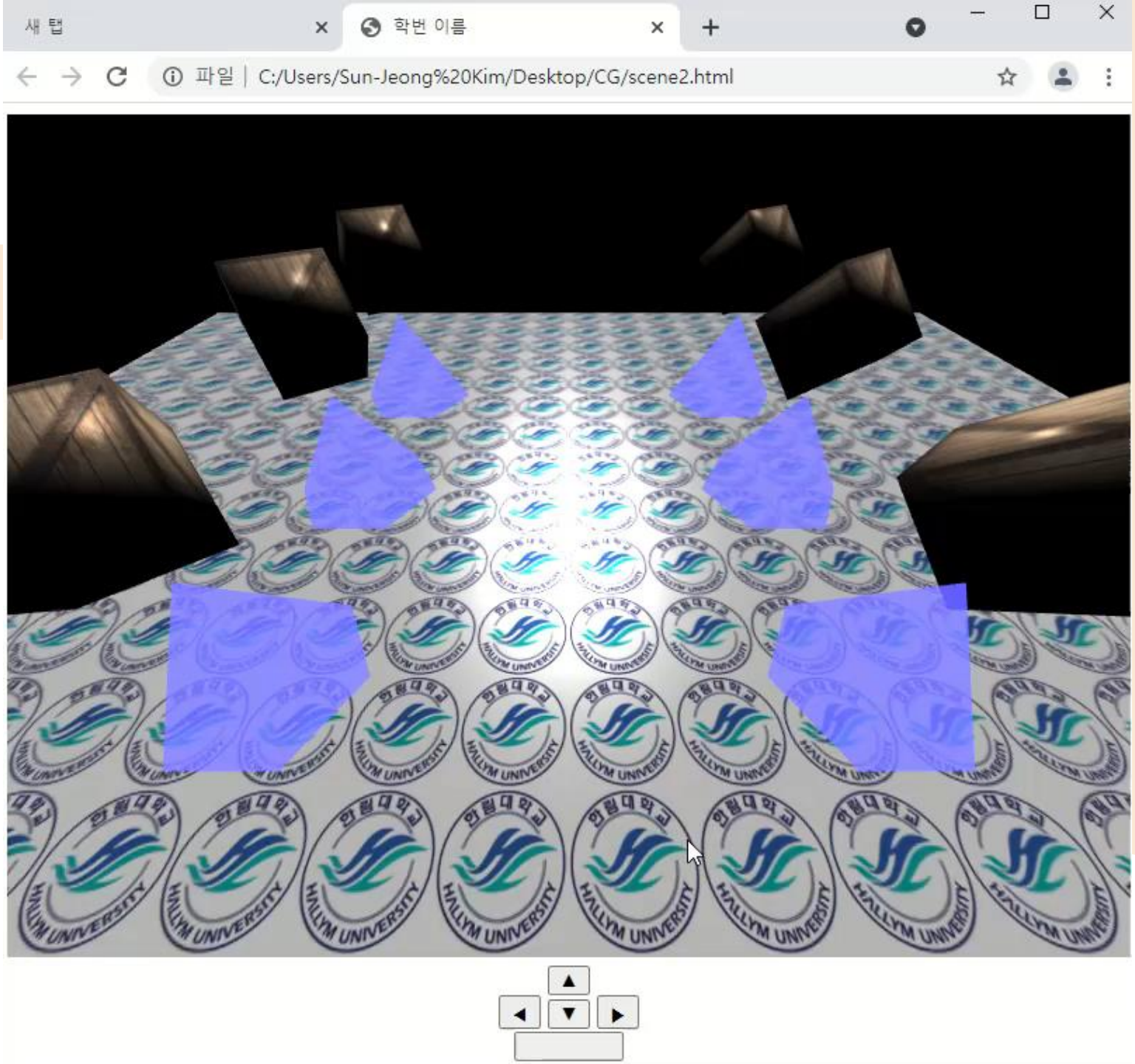

File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > render
309
310 // draw a hexa-pyramid
311 gl.useProgram(program0);
312 gl.uniform4f(colorLoc, 0.0, 0.0, 1.0, 0.5); // translucent blue
313 //gl.useProgram(program1);
314 //gl.uniform4f(diffuseProductLoc, 0.0, 0.0, 1.0, 1.0);
315
316 gl.disable(gl.DEPTH_TEST);
317 gl.enable(gl.BLEND);
318 gl.blendFunc(gl.SRC_ALPHA, gl.ONE_MINUS_SRC_ALPHA);
319
320 modelMatrix = mult(translate(-3, -0.5, z), rotateZ(180));
321 modelMatrix = mult(trballMatrix, modelMatrix);
322 modelViewMatrix = mult(viewMatrix, modelMatrix);
323 gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
324 //gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
325 gl.drawArrays(gl.TRIANGLES, vertHexaStart, vertHexaEnd);
326
327 modelMatrix = mult(translate(3, -0.5, z), rotateZ(180));
328 modelMatrix = mult(trballMatrix, modelMatrix);
329 modelViewMatrix = mult(viewMatrix, modelMatrix);
330 gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
331 //gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
332 gl.drawArrays(gl.TRIANGLES, vertHexaStart, vertHexaEnd);
333
334 gl.disable(gl.BLEND);
335 gl.enable(gl.DEPTH_TEST);
336 }
337
338 requestAnimationFrame(render);
339 }
340
341 function generateTexCube() {
⊗ 0 ⚠ 0
Ln 335, Col 34 Spaces: 4 UTF-8 CRLF JavaScript


새 탭
학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene2.html
HALLYM UNIVERSITY

Collision Detection

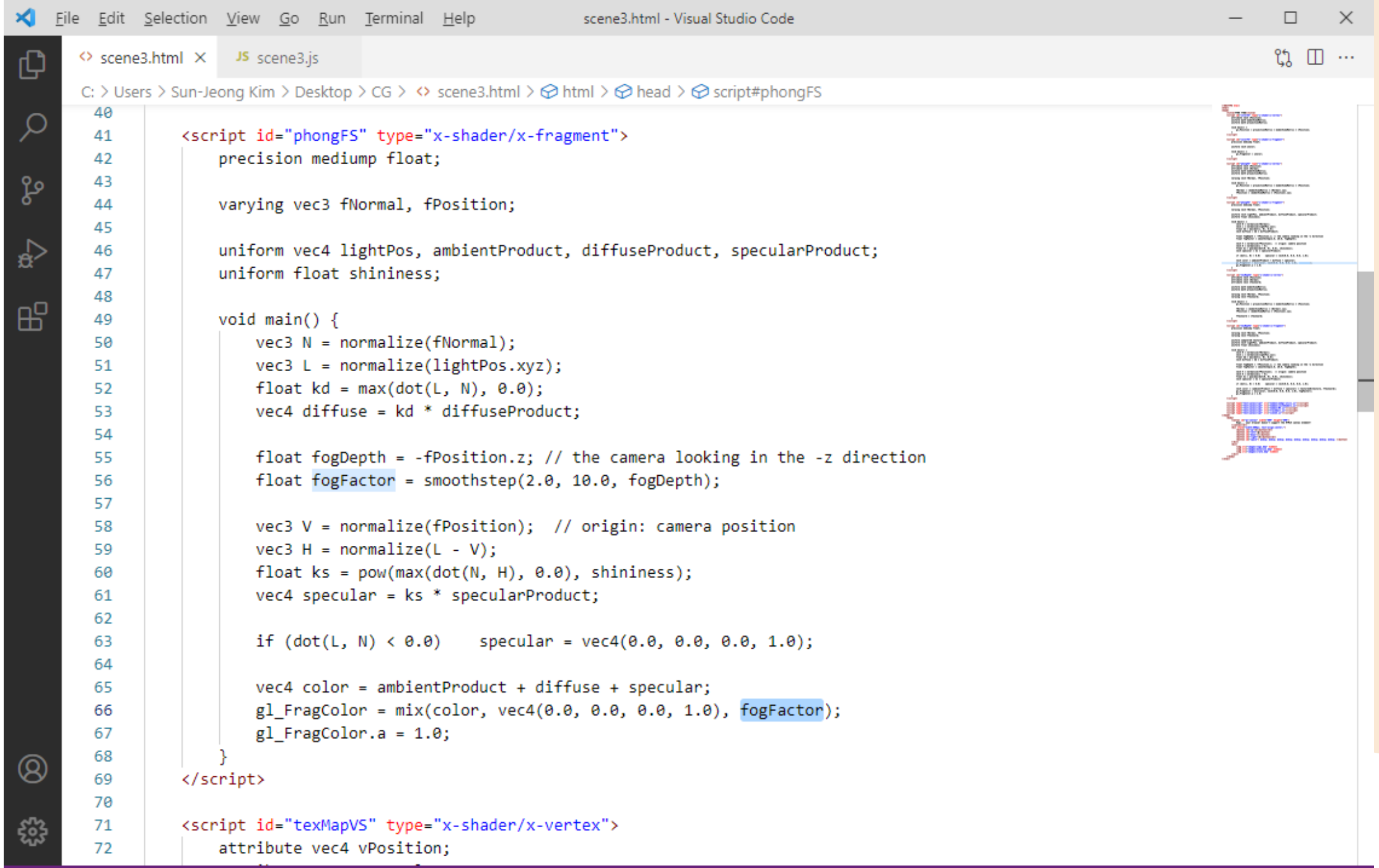
```
8
9   var eye = vec3(0, 3, 3
10      );
11  var at = vec3(0, 0, 0);
12  const up = vec3(0, 1, 0);
13  var cameraVec = vec3(0, -0.7071, -0.7071); // 1.0/Math.sqrt(2.0)
14
15  var theta = 0;
16  var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);
17  var vertCubeStart, vertCubeEnd, vertHexaStart, vertHexaEnd, vertGroundStart, vertGroundEnd;
18
19  var posObjects = [
20      vec2(-3, -5),    vec2(3, -5),
21      vec2(-3, -2),    vec2(3, -2),
22      vec2(-3, 1),     vec2(3, 1),
23      vec2(-3, 4),     vec2(3, 4),
24  ];
25  function detectCollision(newPosX, newPosZ) {
26      for(var index=0; index<posObjects.length; index++) {
27          if( Math.abs(newPosX - posObjects[index][0]) < 1.0 && Math.abs(newPosZ - posObjects[index][1]) < 1.0 )
28              return true;
29      }
30      return false;
31  };
32
33  window.onload = function init()
```

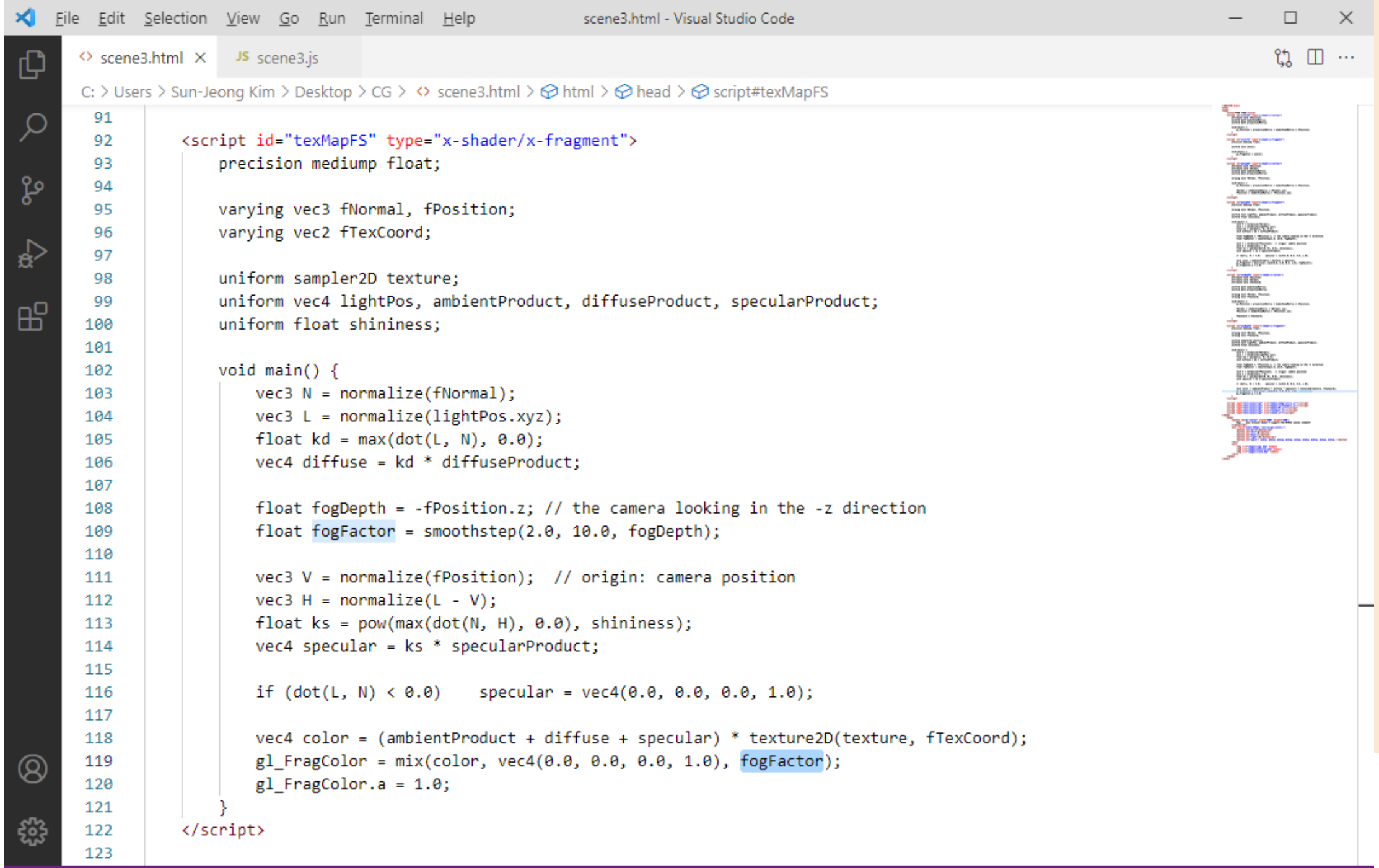

File Edit Selection View Go Run Terminal Help
scene2.js - Visual Studio Code
scene2.html
scene2.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene2.js > init > onclick
188
189 // Event listeners for buttons
190 var sinTheta = Math.sin(0.1);
191 var cosTheta = Math.cos(0.1);
192 document.getElementById("left").onclick = function () {
193 var newVecX = cosTheta*cameraVec[0] + sinTheta*cameraVec[2];
194 var newVecZ = -sinTheta*cameraVec[0] + cosTheta*cameraVec[2];
195 cameraVec[0] = newVecX;
196 cameraVec[2] = newVecZ;
197 };
198 document.getElementById("right").onclick = function () {
199 var newVecX = cosTheta*cameraVec[0] - sinTheta*cameraVec[2];
200 var newVecZ = sinTheta*cameraVec[0] + cosTheta*cameraVec[2];
201 cameraVec[0] = newVecX;
202 cameraVec[2] = newVecZ;
203 };
204 document.getElementById("up").onclick = function () {
205 var newPosX = eye[0] + 0.5 * cameraVec[0];
206 var newPosZ = eye[2] + 0.5 * cameraVec[2];
207 if (newPosX > -10 && newPosX < 10 && newPosZ > -10 && newPosZ < 10 && !detectCollision(newPosX, newPosZ)) {
208 eye[0] = newPosX;
209 eye[2] = newPosZ;
210 }
211 };
212 document.getElementById("down").onclick = function () {
213 var newPosX = eye[0] - 0.5 * cameraVec[0];
214 var newPosZ = eye[2] - 0.5 * cameraVec[2];
215 if (newPosX > -10 && newPosX < 10 && newPosZ > -10 && newPosZ < 10 && !detectCollision(newPosX, newPosZ)) {
216 eye[0] = newPosX;
217 eye[2] = newPosZ;
218 }
219 };
220
⊗ 0 ⚠ 0
Ln 215, Col 95 Spaces: 4 UTF-8 CRLF JavaScript


새 탭
학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene2.html

연습 문제

- 카메라 위치와 Vertex Position 사이의 거리를 Blending Factor로 하여 Fog 색상을 Blending 하는 shader를 작성하시오.


File Edit Selection View Go Run Terminal Help
scene3.html - Visual Studio Code
scene3.html
scene3.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene3.html > html > head > script#phongFS
40
41 <script id="phongFS" type="x-shader/x-fragment">
42 precision mediump float;
43
44 varying vec3 fNormal, fPosition;
45
46 uniform vec4 lightPos, ambientProduct, diffuseProduct, specularProduct;
47 uniform float shininess;
48
49 void main() {
50 vec3 N = normalize(fNormal);
51 vec3 L = normalize(lightPos.xyz);
52 float kd = max(dot(L, N), 0.0);
53 vec4 diffuse = kd * diffuseProduct;
54
55 float fogDepth = -fPosition.z; // the camera looking in the -z direction
56 float fogFactor = smoothstep(2.0, 10.0, fogDepth);
57
58 vec3 V = normalize(fPosition); // origin: camera position
59 vec3 H = normalize(L - V);
60 float ks = pow(max(dot(N, H), 0.0), shininess);
61 vec4 specular = ks * specularProduct;
62
63 if (dot(L, N) < 0.0) specular = vec4(0.0, 0.0, 0.0, 1.0);
64
65 vec4 color = ambientProduct + diffuse + specular;
66 gl_FragColor = mix(color, vec4(0.0, 0.0, 0.0, 1.0), fogFactor);
67 gl_FragColor.a = 1.0;
68 }
69 </script>
70
71 <script id="texMapVS" type="x-shader/x-vertex">
72 attribute vec4 vPosition;
⊗ 0 ⚠ 0
Ln 66, Col 74 (9 selected) Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
scene3.html - Visual Studio Code
scene3.html
scene3.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene3.html > html > head > script#texMapFS
91
92 <script id="texMapFS" type="x-shader/x-fragment">
93 precision mediump float;
94
95 varying vec3 fNormal, fPosition;
96 varying vec2 fTexCoord;
97
98 uniform sampler2D texture;
99 uniform vec4 lightPos, ambientProduct, diffuseProduct, specularProduct;
100 uniform float shininess;
101
102 void main() {
103 vec3 N = normalize(fNormal);
104 vec3 L = normalize(lightPos.xyz);
105 float kd = max(dot(L, N), 0.0);
106 vec4 diffuse = kd * diffuseProduct;
107
108 float fogDepth = -fPosition.z; // the camera looking in the -z direction
109 float fogFactor = smoothstep(2.0, 10.0, fogDepth);
110
111 vec3 V = normalize(fPosition); // origin: camera position
112 vec3 H = normalize(L - V);
113 float ks = pow(max(dot(N, H), 0.0), shininess);
114 vec4 specular = ks * specularProduct;
115
116 if (dot(L, N) < 0.0) specular = vec4(0.0, 0.0, 0.0, 1.0);
117
118 vec4 color = (ambientProduct + diffuse + specular) * texture2D(texture, fTexCoord);
119 gl_FragColor = mix(color, vec4(0.0, 0.0, 0.0, 1.0), fogFactor);
120 gl_FragColor.a = 1.0;
121 }
122 </script>
123
⊗ 0 ⚠ 0
Ln 119, Col 74 (9 selected) Spaces: 4 UTF-8 CRLF HTML

File Edit Selection View Go Run Terminal Help scene3.js - Visual Studio Code

scene3.html scene3.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene3.js > render > diffuseProductLoc

```
285
286     //var colorLoc = gl.getUniformLocation(program0, "uColor");
287     var diffuseProductLoc = gl.getUniformLocation(program1, "diffuseProduct");
288
289     // draw the ground
290     //gl.uniform4f(colorLoc, 0.8, 0.8, 0.8, 1.0); // gray
291     //gl.uniform4f(diffuseProductLoc, 0.8, 0.8, 0.8, 1.0);
292     gl.useProgram(program2);
293     gl.uniform1i(gl.getUniformLocation(program2, "texture"), 0);
294
295     modelViewMatrix = mult(viewMatrix, trballMatrix);
296     //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
297     //gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
298     gl.uniformMatrix4fv(modelViewMatrixLoc2, false, flatten(modelViewMatrix));
299     gl.drawArrays(gl.TRIANGLES, vertGroundStart, vertGroundEnd);
300
301     for (var z=-5; z<5; z+=3) {
302         // draw a cube
303         //gl.uniform4f(colorLoc, 1.0, 0.0, 0.0, 1.0); // red
304         //gl.uniform4f(diffuseProductLoc, 1.0, 0.0, 0.0, 1.0);
305         gl.useProgram(program2);
306         gl.uniform1i(gl.getUniformLocation(program2, "texture"), 1);
307
308         var rMatrix = mult(rotateY(theta), rotateZ(45));
309         var modelMatrix = mult(translate(-3, 1.3, z), rMatrix);
310         modelMatrix = mult(trballMatrix, modelMatrix);
311         modelViewMatrix = mult(viewMatrix, modelMatrix);
312         //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
313         //gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
314         gl.uniformMatrix4fv(modelViewMatrixLoc2, false, flatten(modelViewMatrix));
315         gl.drawArrays(gl.TRIANGLES, vertCubeStart, vertCubeEnd);
316
317         modelMatrix = mult(translate(3, 1.3, z), rMatrix);
```

⊗ 0 ⚠ 0 Ln 287, Col 59 Spaces: 4 UTF-8 CRLF JavaScript

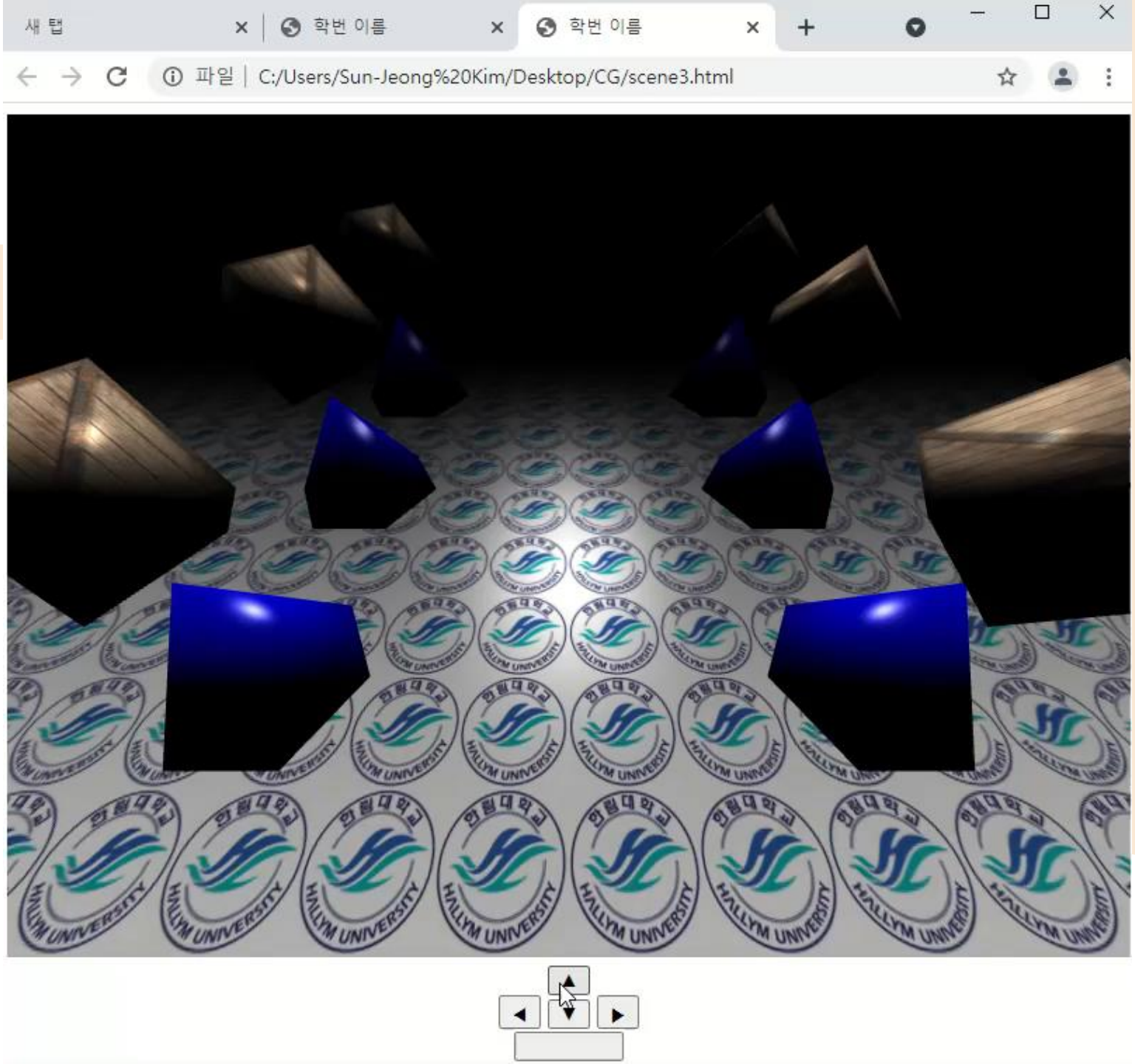
File Edit Selection View Go Run Terminal Help scene3.js - Visual Studio Code

scene3.html scene3.js

C: > Users > Sun-Jeong Kim > Desktop > CG > scene3.js > render

```
324
325         // draw a hexa-pyramid
326         //gl.useProgram(program0);
327         //gl.uniform4f(colorLoc, 0.0, 0.0, 1.0, 0.5); // translucent blue
328         gl.useProgram(program1);
329         gl.uniform4f(diffuseProductLoc, 0.0, 0.0, 1.0, 1.0);
330
331         //gl.disable(gl.DEPTH_TEST);
332         //gl.enable(gl.BLEND);
333         //gl.blendFunc(gl.SRC_ALPHA, gl.ONE_MINUS_SRC_ALPHA);
334
335         modelMatrix = mult(translate(-3, -0.5, z), rotateZ(180));
336         modelMatrix = mult(trballMatrix, modelMatrix);
337         modelViewMatrix = mult(viewMatrix, modelMatrix);
338         //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
339         gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
340         gl.drawArrays(gl.TRIANGLES, vertHexaStart, vertHexaEnd);
341
342         modelMatrix = mult(translate(3, -0.5, z), rotateZ(180));
343         modelMatrix = mult(trballMatrix, modelMatrix);
344         modelViewMatrix = mult(viewMatrix, modelMatrix);
345         //gl.uniformMatrix4fv(modelViewMatrixLoc0, false, flatten(modelViewMatrix));
346         gl.uniformMatrix4fv(modelViewMatrixLoc1, false, flatten(modelViewMatrix));
347         gl.drawArrays(gl.TRIANGLES, vertHexaStart, vertHexaEnd);
348
349         //gl.disable(gl.BLEND);
350         //gl.enable(gl.DEPTH_TEST);
351     }
352
353     requestAnimationFrame(render);
354 }
355
356 function generateTexCube() {
```

0 ⚠ 0 Ln 328, Col 23 Spaces: 4 UTF-8 CRLF JavaScript


새 탭
학번 이름
학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene3.html
HALLYM UNIVERSITY