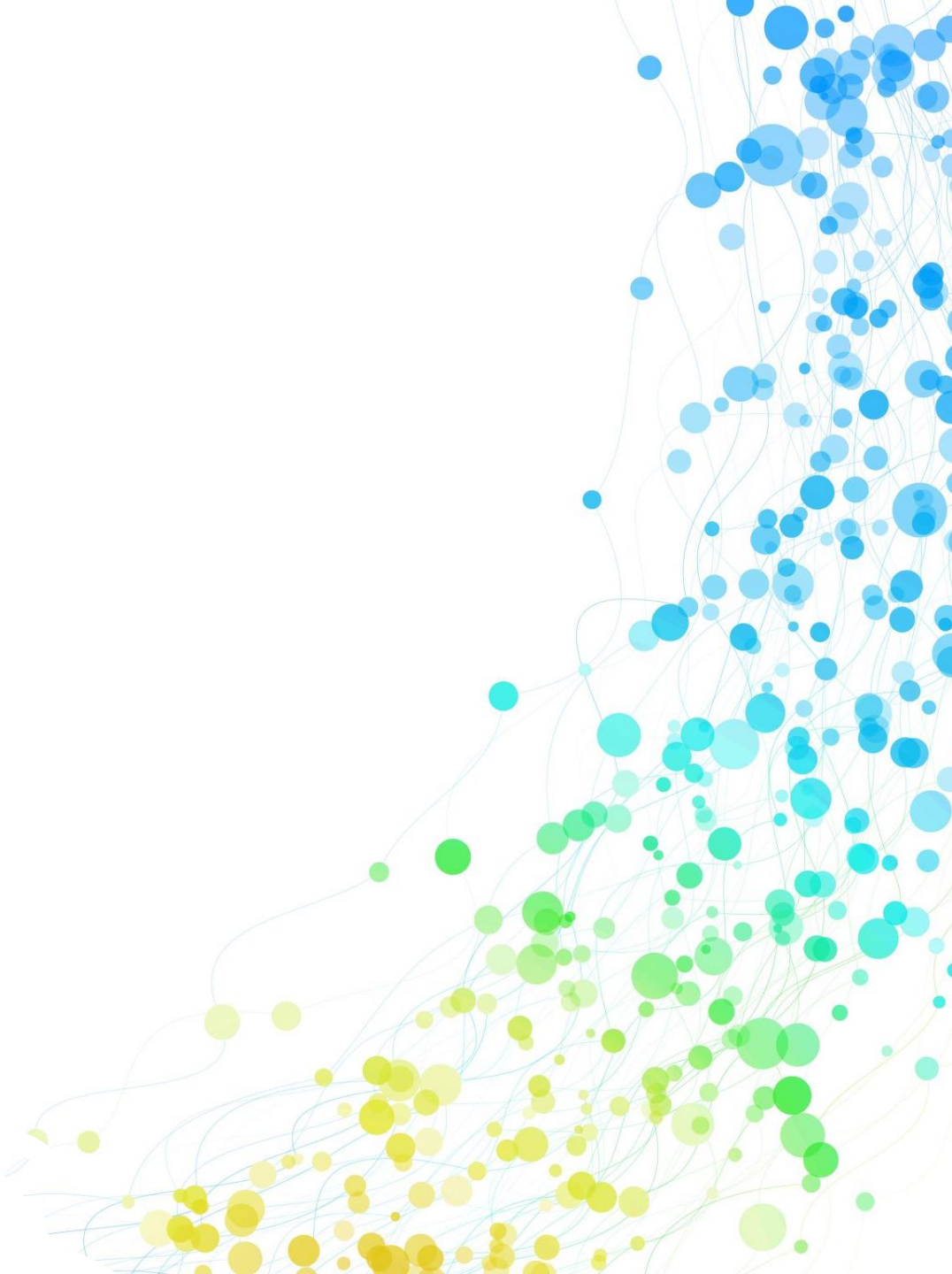
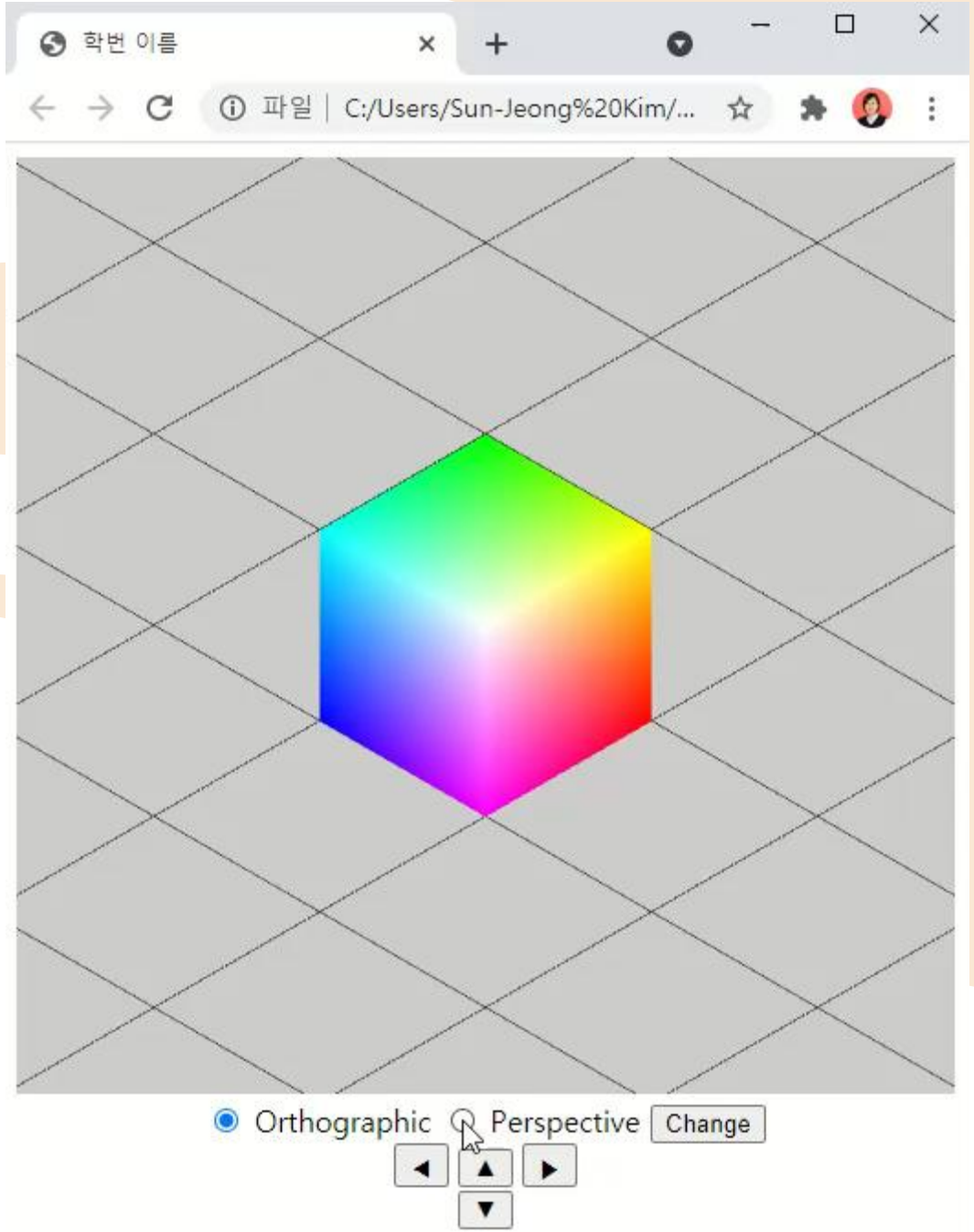
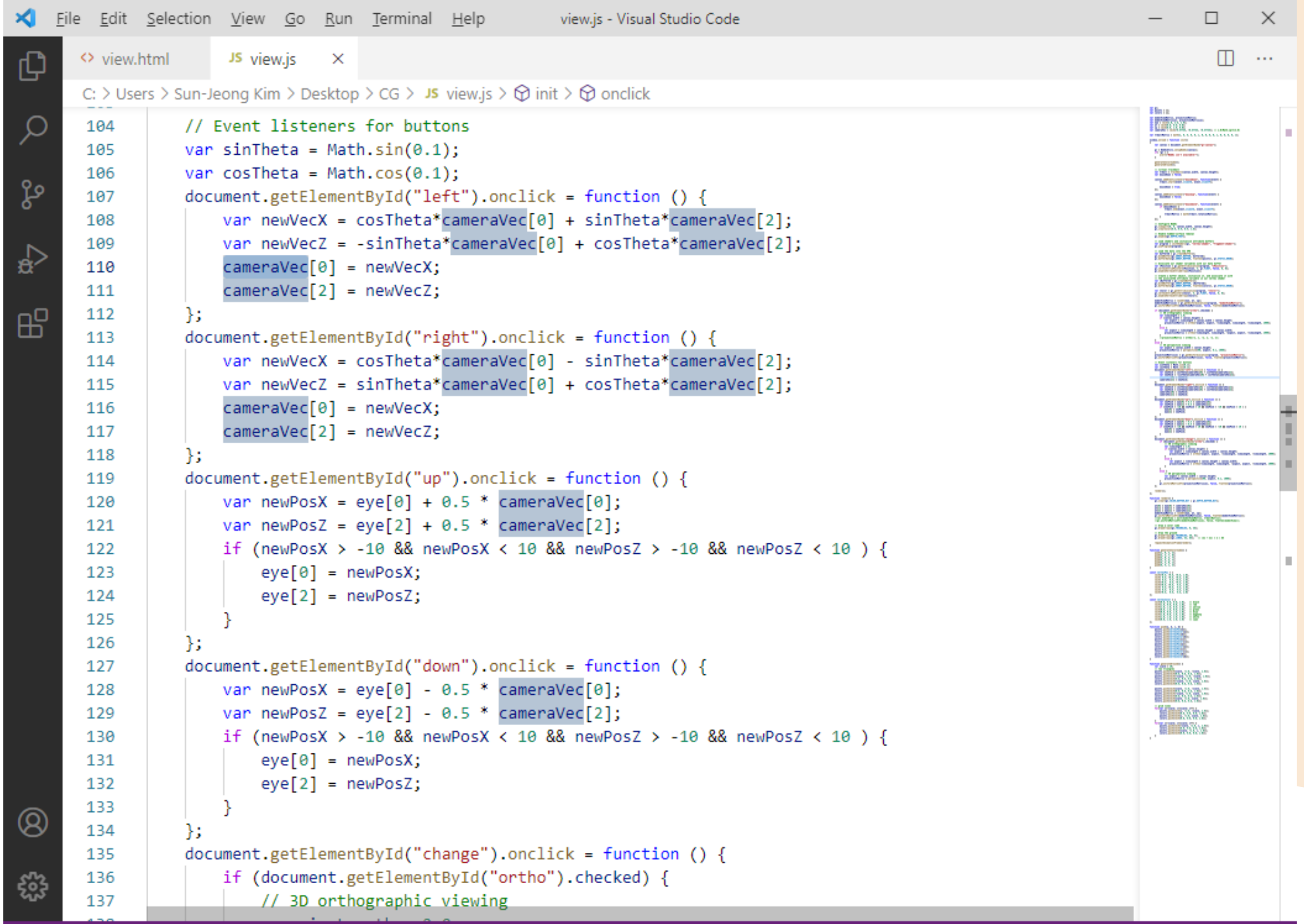
Building a Scene

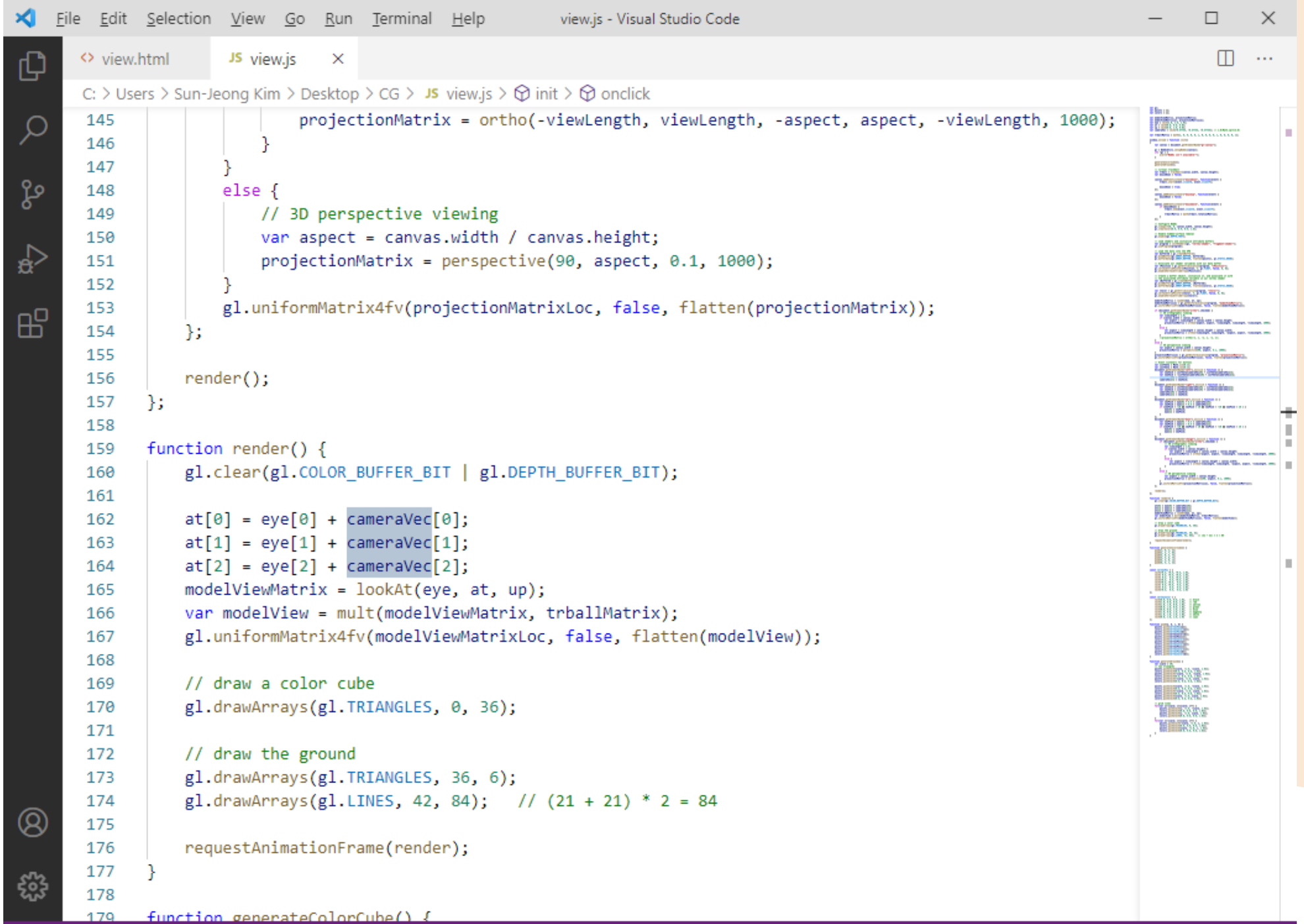
10TH WEEK, 2021

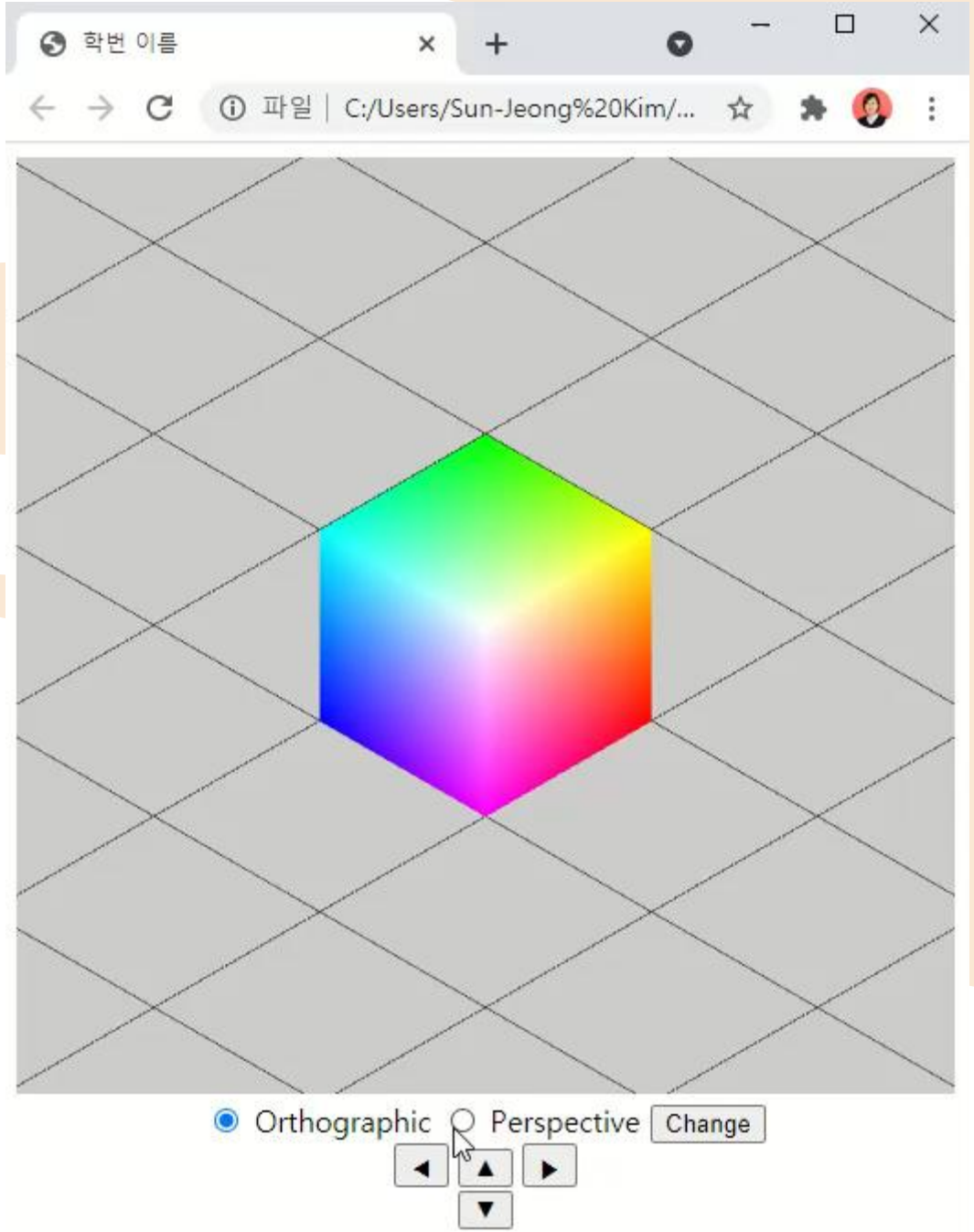



학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/...
Orthographic
Perspective
Change

Walking Through

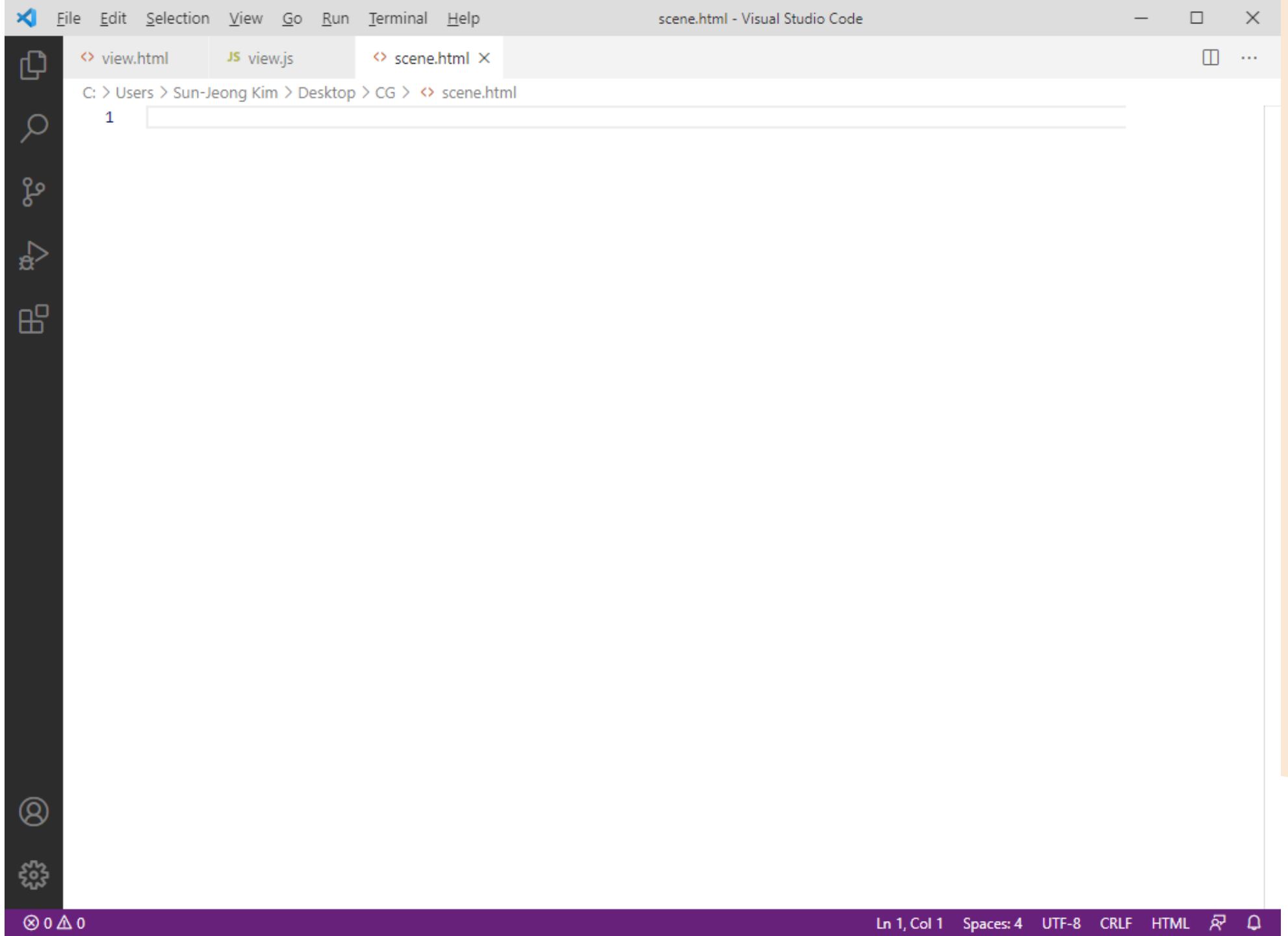

File Edit Selection View Go Run Terminal Help
view.js - Visual Studio Code
view.html
view.js
C: > Users > Sun-Jeong Kim > Desktop > CG > view.js > init > onclick
104 // Event listeners for buttons
105 var sinTheta = Math.sin(0.1);
106 var cosTheta = Math.cos(0.1);
107 document.getElementById("left").onclick = function () {
108 var newVecX = cosTheta*cameraVec[0] + sinTheta*cameraVec[2];
109 var newVecZ = -sinTheta*cameraVec[0] + cosTheta*cameraVec[2];
110 cameraVec[0] = newVecX;
111 cameraVec[2] = newVecZ;
112 };
113 document.getElementById("right").onclick = function () {
114 var newVecX = cosTheta*cameraVec[0] - sinTheta*cameraVec[2];
115 var newVecZ = sinTheta*cameraVec[0] + cosTheta*cameraVec[2];
116 cameraVec[0] = newVecX;
117 cameraVec[2] = newVecZ;
118 };
119 document.getElementById("up").onclick = function () {
120 var newPosX = eye[0] + 0.5 * cameraVec[0];
121 var newPosZ = eye[2] + 0.5 * cameraVec[2];
122 if (newPosX > -10 && newPosX < 10 && newPosZ > -10 && newPosZ < 10) {
123 eye[0] = newPosX;
124 eye[2] = newPosZ;
125 }
126 };
127 document.getElementById("down").onclick = function () {
128 var newPosX = eye[0] - 0.5 * cameraVec[0];
129 var newPosZ = eye[2] - 0.5 * cameraVec[2];
130 if (newPosX > -10 && newPosX < 10 && newPosZ > -10 && newPosZ < 10) {
131 eye[0] = newPosX;
132 eye[2] = newPosZ;
133 }
134 };
135 document.getElementById("change").onclick = function () {
136 if (document.getElementById("ortho").checked) {
137 // 3D orthographic viewing
0 0
Ln 110, Col 18 (9 selected) Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
view.js - Visual Studio Code
view.html
view.js
C: > Users > Sun-Jeong Kim > Desktop > CG > view.js > init > onclick
projectionMatrix = ortho(-viewLength, viewLength, -aspect, aspect, -viewLength, 1000);
}
}
else {
// 3D perspective viewing
var aspect = canvas.width / canvas.height;
projectionMatrix = perspective(90, aspect, 0.1, 1000);
}
gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projectionMatrix));
};
render();
};
function render() {
gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
at[0] = eye[0] + cameraVec[0];
at[1] = eye[1] + cameraVec[1];
at[2] = eye[2] + cameraVec[2];
modelViewMatrix = lookAt(eye, at, up);
var modelView = mult(modelViewMatrix, trballMatrix);
gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelView));
// draw a color cube
gl.drawArrays(gl.TRIANGLES, 0, 36);
// draw the ground
gl.drawArrays(gl.TRIANGLES, 36, 6);
gl.drawArrays(gl.LINES, 42, 84);   // (21 + 21) * 2 = 84
requestAnimationFrame(render);
}
function generateColorCube() {
0 0
Ln 110, Col 18 (9 selected)
Spaces: 4
UTF-8
CRLF
JavaScript


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/...
Orthographic
Perspective
Change

Building a Scene

- Ground 위에 오브젝트들(정육면체, 육각뿔)을 배치
- 오브젝트들은 각자 local 축을 중심으로 회전을 하고 있음
- 중앙에 새로운 오브젝트 생성 후 입력에 따라 카메라와 함께 이동


File Edit Selection View Go Run Terminal Help
scene.html - Visual Studio Code
view.html
view.js
scene.html
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.html
1
0 0
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
HTML

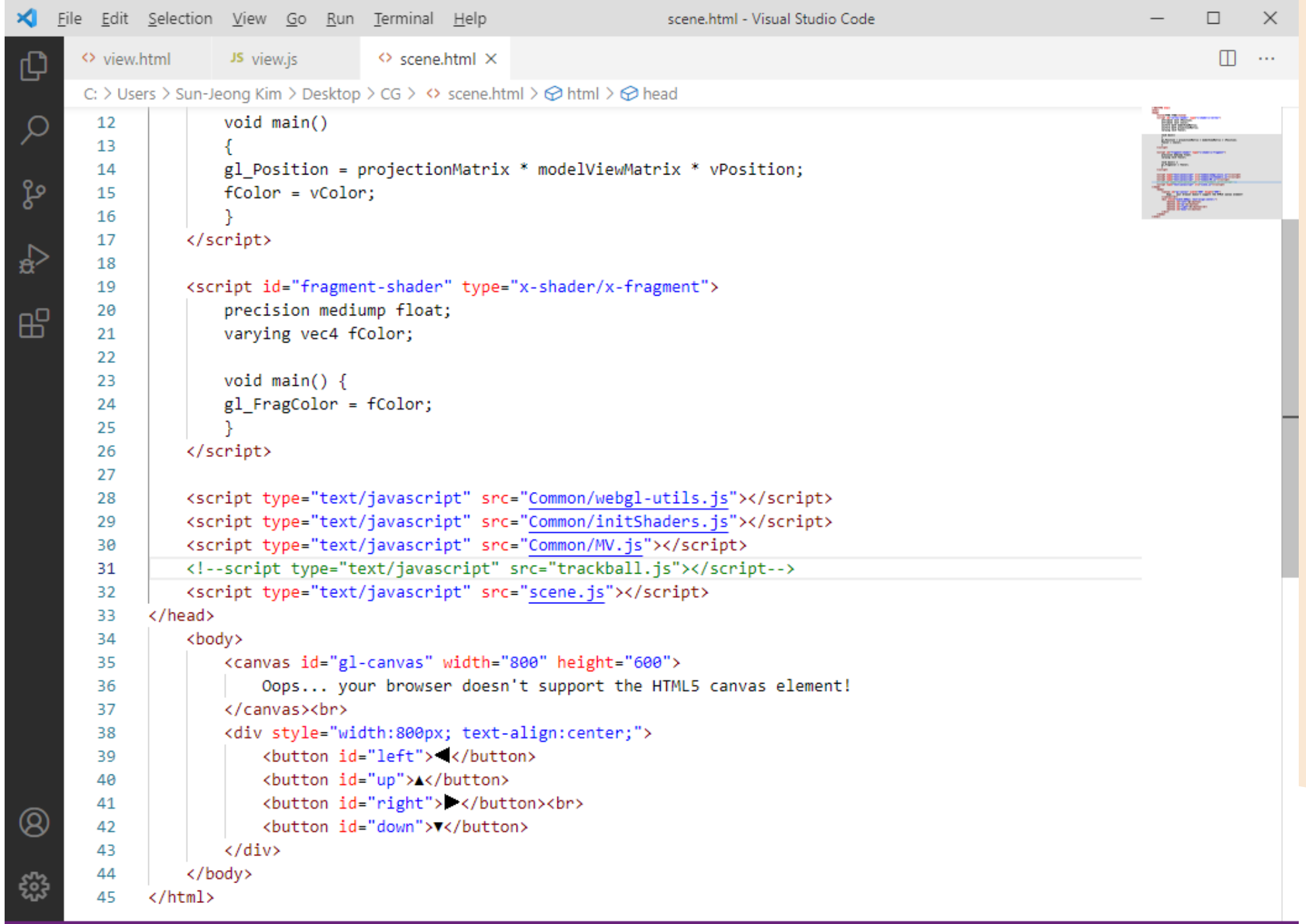
File Edit Selection View Go Run Terminal Help scene.html - Visual Studio Code

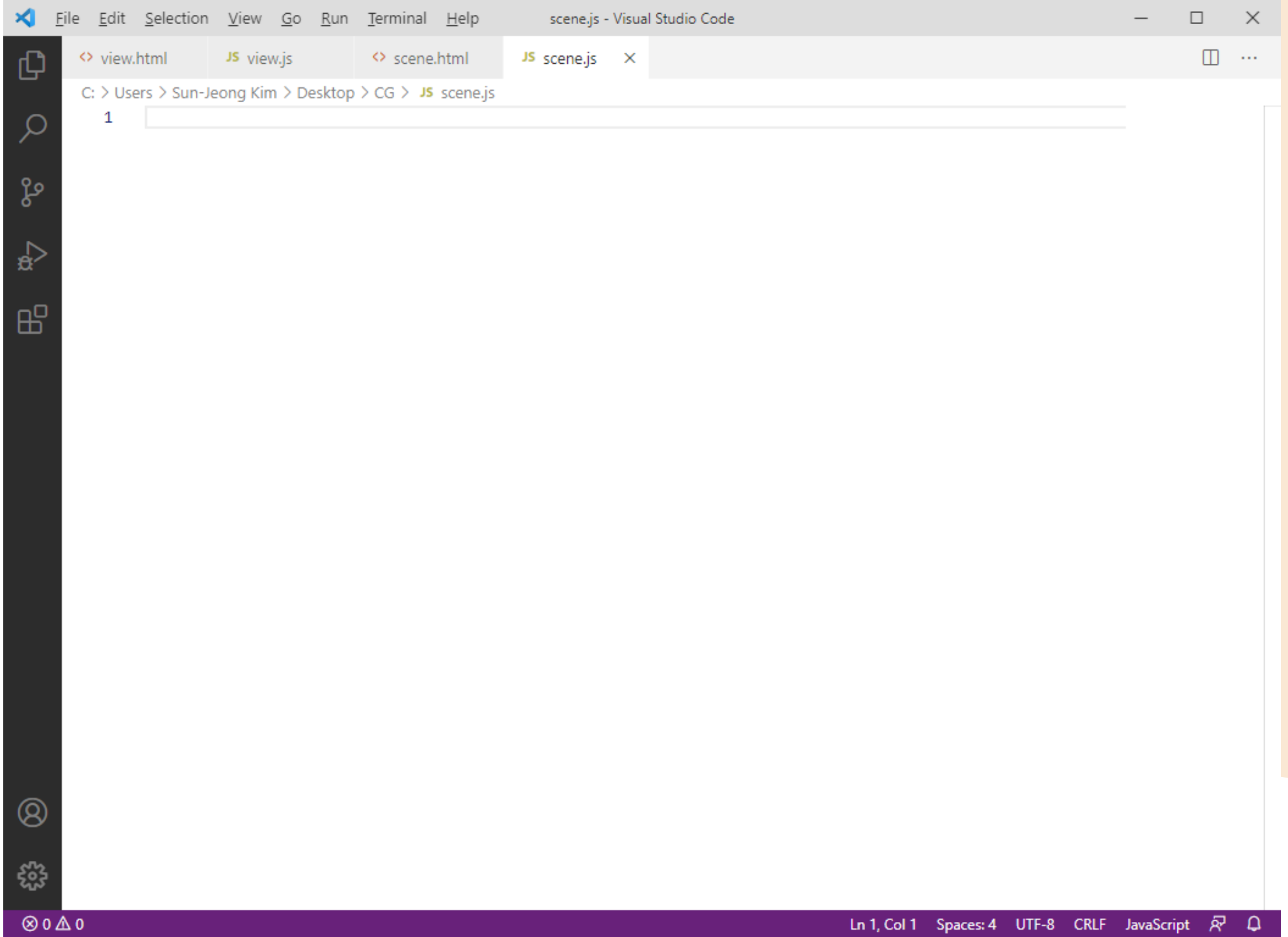
view.html view.js scene.html

C: > Users > Sun-Jeong Kim > Desktop > CG > scene.html > html > head

```
<!DOCTYPE html>
<html>
<head>
    <title>학번 이름</title>
    <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;
        attribute vec4 vColor;
        uniform mat4 modelViewMatrix;
        uniform mat4 projectionMatrix;
        varying vec4 fColor;

        void main()
        {
        gl_Position = projectionMatrix * modelViewMatrix * vPosition;
        fColor = vColor;
        }
    </script>

    <script id="fragment-shader" type="x-shader/x-fragment">
        precision mediump float;
        varying vec4 fColor;

        void main() {
        gl_FragColor = fColor;
        }
    </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="scene.js"></script>
</head>
    <body>
        <canvas id="gl-canvas" width="800" height="600">
```

⊗ 0 ⚠ 0 Ln 31, Col 68 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
scene.html - Visual Studio Code
view.html
view.js
scene.html
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.html > html > head
12 void main()
13 {
14 gl_Position = projectionMatrix * modelViewMatrix * vPosition;
15 fColor = vColor;
16 }
17 </script>
18
19 <script id="fragment-shader" type="x-shader/x-fragment">
20 precision mediump float;
21 varying vec4 fColor;
22
23 void main() {
24 gl_FragColor = fColor;
25 }
26 </script>
27
28 <script type="text/javascript" src="Common/webgl-utils.js"></script>
29 <script type="text/javascript" src="Common/initShaders.js"></script>
30 <script type="text/javascript" src="Common/MV.js"></script>
31 <!--script type="text/javascript" src="trackball.js"></script-->
32 <script type="text/javascript" src="scene.js"></script>
33 </head>
34 <body>
35 <canvas id="gl-canvas" width="800" height="600">
36 Oops... your browser doesn't support the HTML5 canvas element!
37 </canvas>
38
39 <button id="left">◀</button>
40 <button id="up">▲</button>
41 <button id="right">▶</button>
42 <button id="down">▼</button>
43
44 </body>
45 </html>
Ln 31, Col 68 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js
1
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript

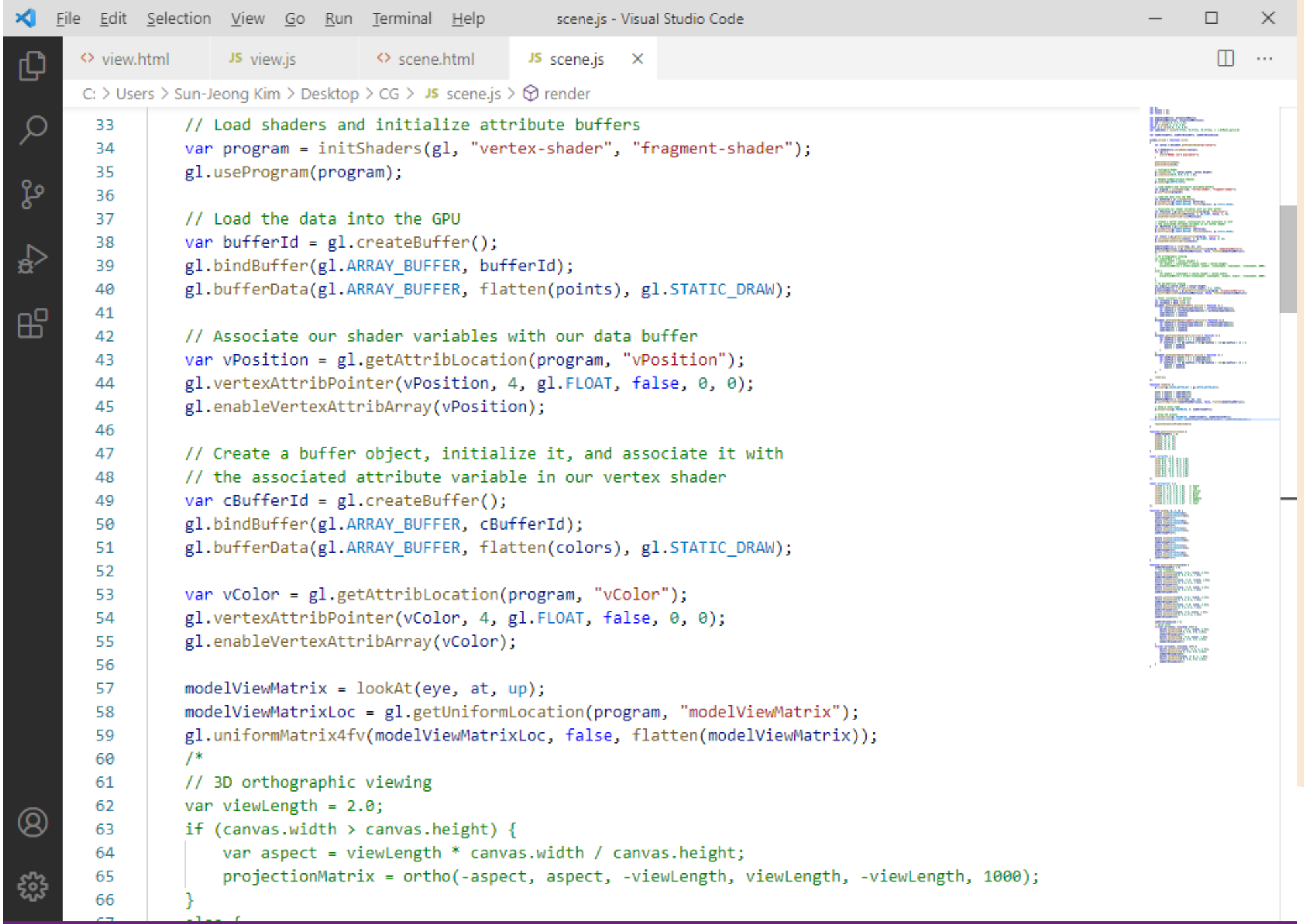
File Edit Selection View Go Run Terminal Help scene.js - Visual Studio Code

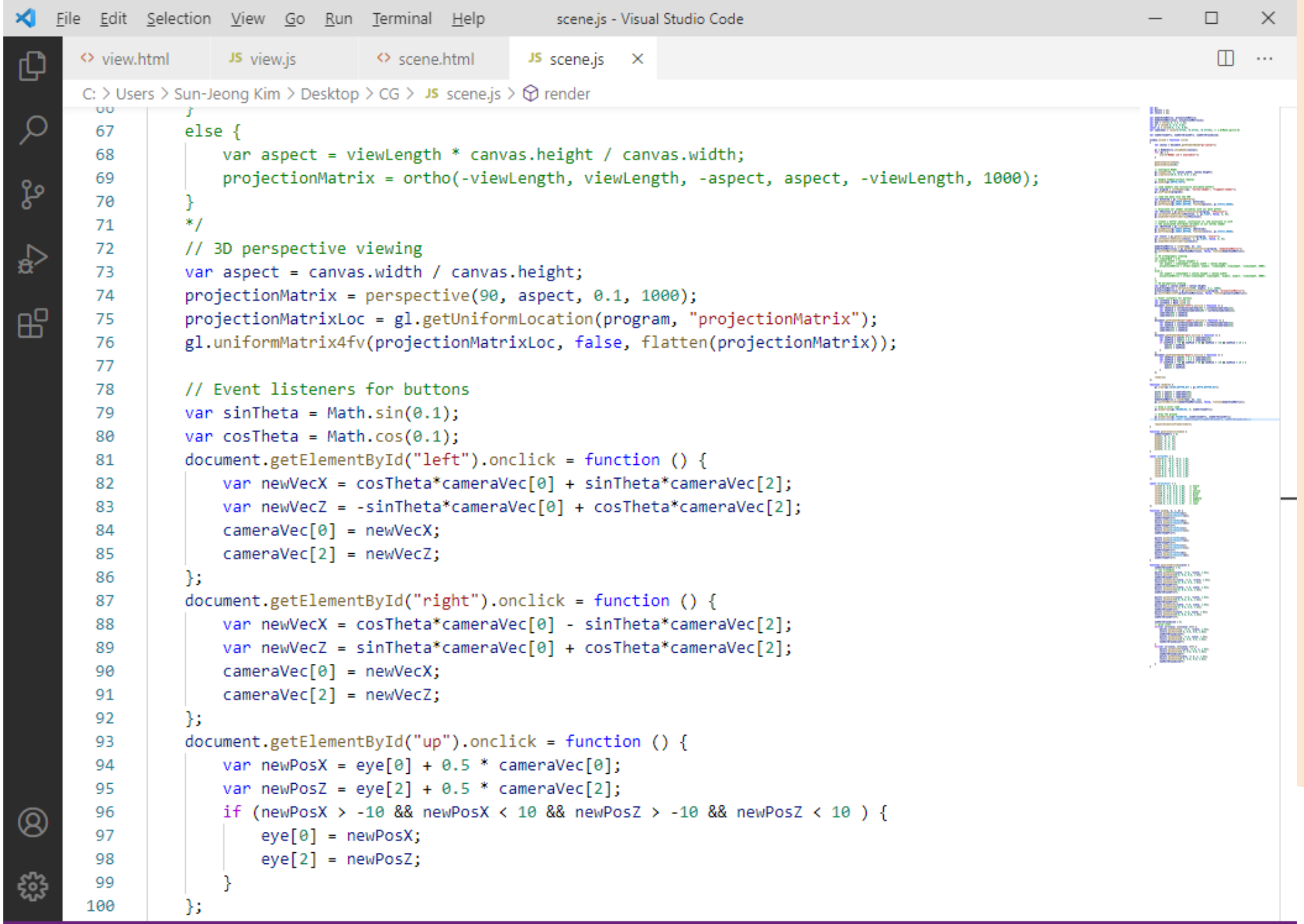
view.html | view.js | scene.html | scene.js

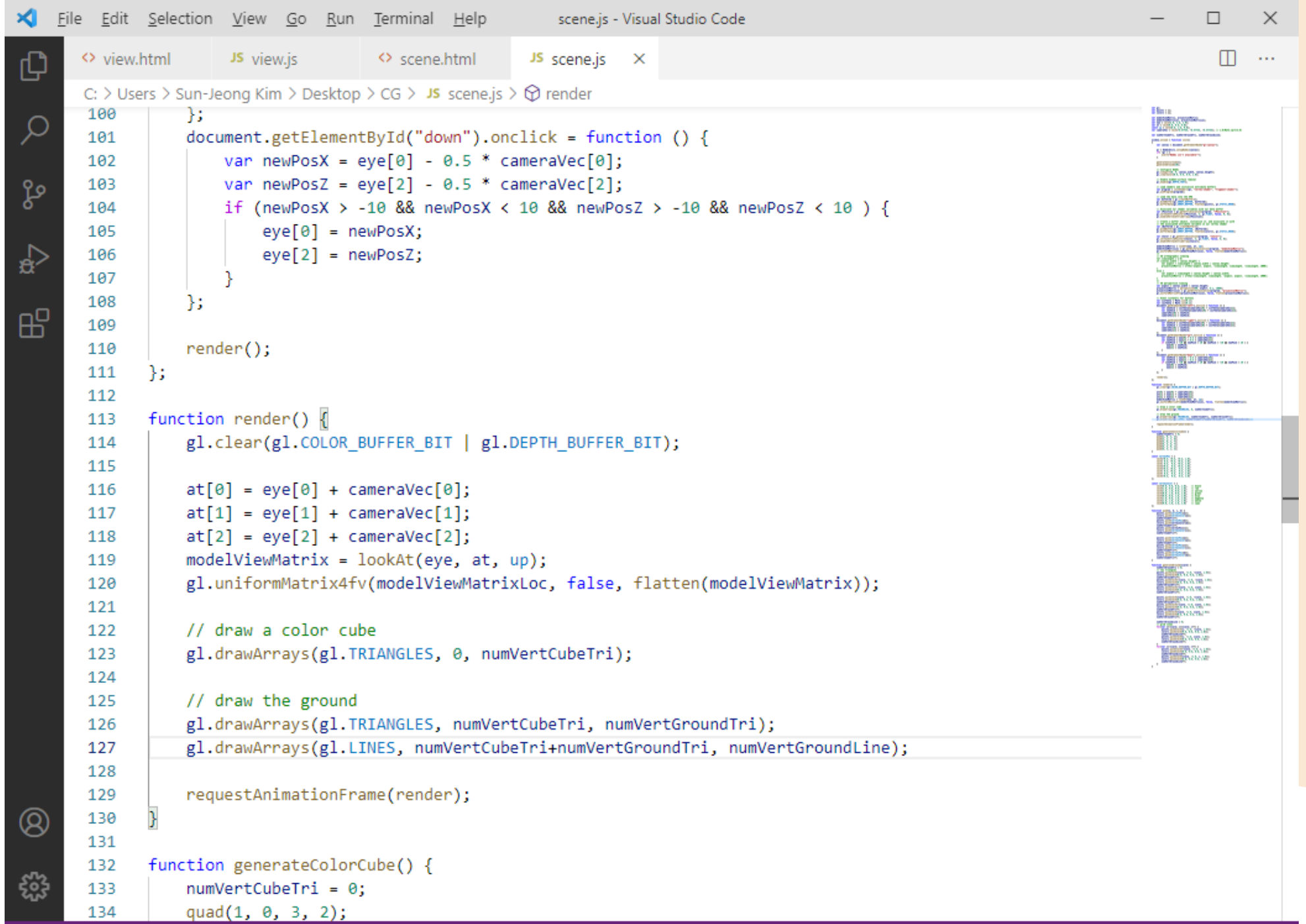
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > render

```
var gl;
var points = [];
var colors = [];

var modelViewMatrix, projectionMatrix;
var modelViewMatrixLoc, projectionMatrixLoc;
var eye = vec3(2.0, 2.0, 2.0);
var at = vec3(0.0, 0.0, 0.0);
const up = vec3(0.0, 1.0, 0.0);
var cameraVec = vec3(-0.57735, -0.57735, -0.57735); // 1.0/Math.sqrt(3.0)

var numVertCubeTri, numVertGroundTri, numVertGroundLine;

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

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

    generateColorCube();
    generateGround(10);

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.9, 0.9, 0.9, 1.0);

    // Enable hidden-surface removal
    gl.enable(gl.DEPTH_TEST);

    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);
```

⊗ 0 ⚠ 0 Ln 127, Col 81 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > render
33 // Load shaders and initialize attribute buffers
34 var program = initShaders(gl, "vertex-shader", "fragment-shader");
35 gl.useProgram(program);
36
37 // Load the data into the GPU
38 var bufferId = gl.createBuffer();
39 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
40 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
41
42 // Associate our shader variables with our data buffer
43 var vPosition = gl.getAttribLocation(program, "vPosition");
44 gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
45 gl.enableVertexAttribArray(vPosition);
46
47 // Create a buffer object, initialize it, and associate it with
48 // the associated attribute variable in our vertex shader
49 var cBufferId = gl.createBuffer();
50 gl.bindBuffer(gl.ARRAY_BUFFER, cBufferId);
51 gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
52
53 var vColor = gl.getAttribLocation(program, "vColor");
54 gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
55 gl.enableVertexAttribArray(vColor);
56
57 modelViewMatrix = lookAt(eye, at, up);
58 modelViewMatrixLoc = gl.getUniformLocation(program, "modelViewMatrix");
59 gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
60 /*
61 // 3D orthographic viewing
62 var viewLength = 2.0;
63 if (canvas.width > canvas.height) {
64 var aspect = viewLength * canvas.width / canvas.height;
65 projectionMatrix = ortho(-aspect, aspect, -viewLength, viewLength, -viewLength, 1000);
66 }
⊗ 0 ⚠ 0
Ln 127, Col 81 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > render
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);
projectionMatrixLoc = gl.getUniformLocation(program, "projectionMatrix");
gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projectionMatrix));
// 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 127, Col 81
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > render
100 };
101 document.getElementById("down").onclick = function () {
102 var newPosX = eye[0] - 0.5 * cameraVec[0];
103 var newPosZ = eye[2] - 0.5 * cameraVec[2];
104 if (newPosX > -10 && newPosX < 10 && newPosZ > -10 && newPosZ < 10) {
105 eye[0] = newPosX;
106 eye[2] = newPosZ;
107 }
108 };
109
110 render();
111 };
112
113 function render() {
114 gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
115
116 at[0] = eye[0] + cameraVec[0];
117 at[1] = eye[1] + cameraVec[1];
118 at[2] = eye[2] + cameraVec[2];
119 modelViewMatrix = lookAt(eye, at, up);
120 gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
121
122 // draw a color cube
123 gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
124
125 // draw the ground
126 gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertGroundTri);
127 gl.drawArrays(gl.LINES, numVertCubeTri+numVertGroundTri, numVertGroundLine);
128
129 requestAnimationFrame(render);
130 }
131
132 function generateColorCube() {
133 numVertCubeTri = 0;
134 quad(1, 0, 3, 2);
⊗ 0 ⚠ 0
Ln 127, Col 81 Spaces: 4 UTF-8 CRLF JavaScript

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

view.html | view.js | scene.html | scene.js

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

```
132  function generateColorCube() {
133      numVertCubeTri = 0;
134      quad(1, 0, 3, 2);
135      quad(2, 3, 7, 6);
136      quad(3, 0, 4, 7);
137      quad(4, 5, 6, 7);
138      quad(5, 4, 0, 1);
139      quad(6, 5, 1, 2);
140  }
141
142  function quad(a, b, c, d) {
143      const vertexPos = [
144          vec4(-0.5, -0.5, -0.5, 1.0),
145          vec4( 0.5, -0.5, -0.5, 1.0),
146          vec4( 0.5,  0.5, -0.5, 1.0),
147          vec4(-0.5,  0.5, -0.5, 1.0),
148          vec4(-0.5, -0.5,  0.5, 1.0),
149          vec4( 0.5, -0.5,  0.5, 1.0),
150          vec4( 0.5,  0.5,  0.5, 1.0),
151          vec4(-0.5,  0.5,  0.5, 1.0)
152      ];
153
154      const vertexColor = [
155          vec4(0.0, 0.0, 0.0, 1.0),   // black
156          vec4(1.0, 0.0, 0.0, 1.0),   // red
157          vec4(1.0, 1.0, 0.0, 1.0),   // yellow
158          vec4(0.0, 1.0, 0.0, 1.0),   // green
159          vec4(0.0, 0.0, 1.0, 1.0),   // blue
160          vec4(1.0, 0.0, 1.0, 1.0),   // magenta
161          vec4(1.0, 1.0, 1.0, 1.0),   // white
162          vec4(0.0, 1.0, 1.0, 1.0)    // cyan
163      ];
164
165      points.push(vertexPos[a]);
166      colors.push(vertexColor[a]);
```

⊗ 0 ⚠ 0 Ln 127, Col 81 Spaces: 4 UTF-8 CRLF JavaScript

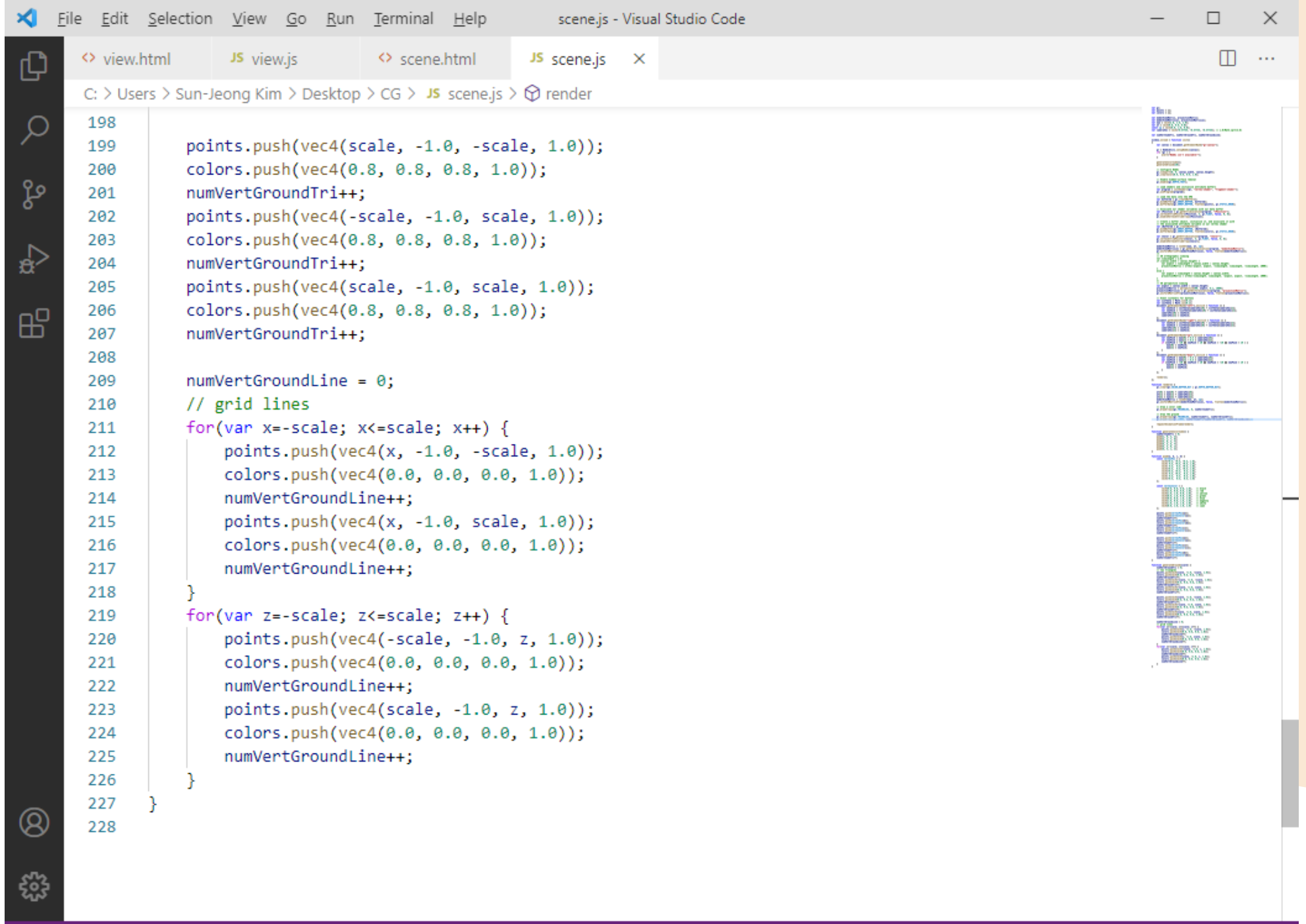
File Edit Selection View Go Run Terminal Help scene.js - Visual Studio Code

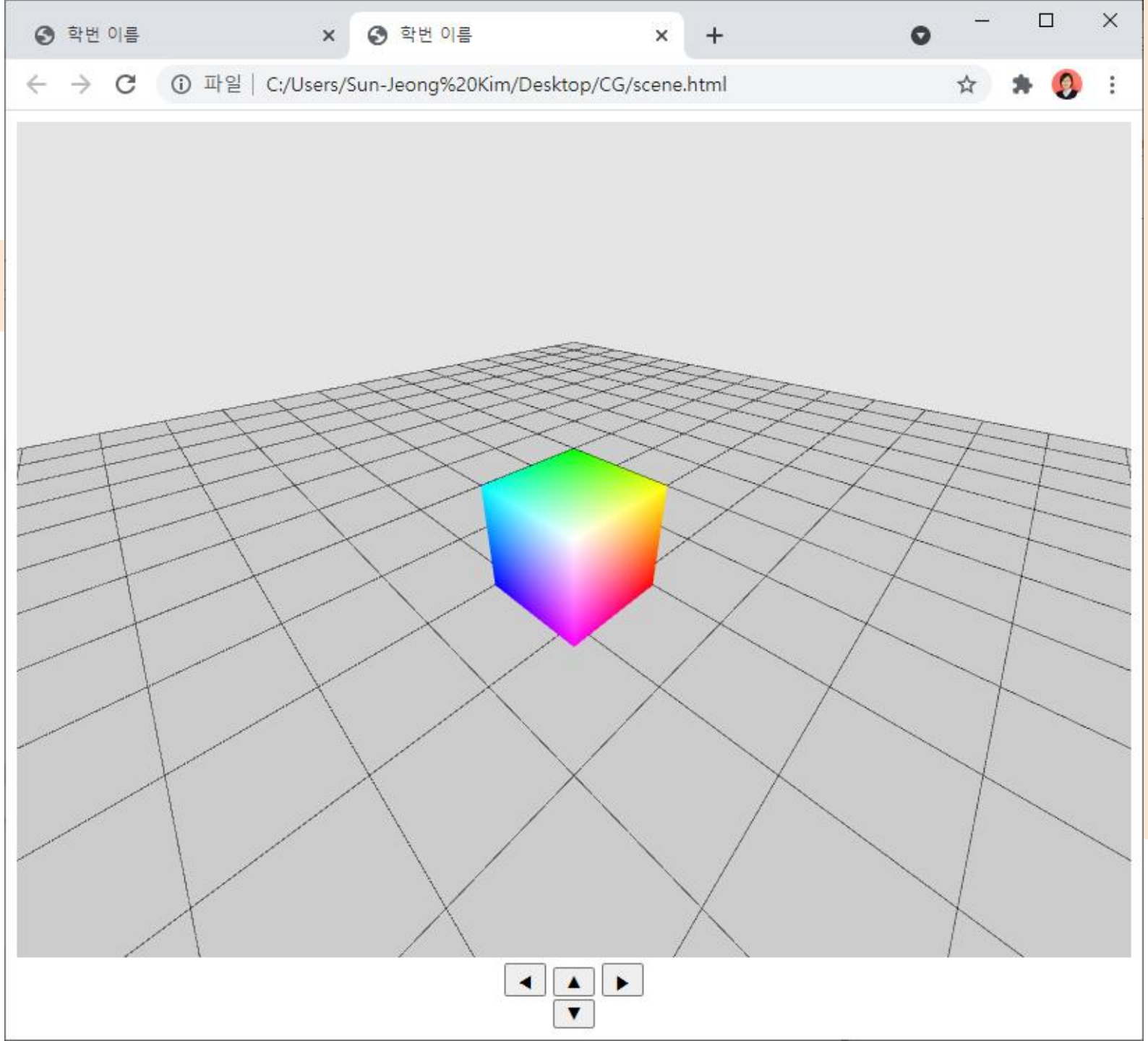
view.html | view.js | scene.html | scene.js

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

```
164
165     points.push(vertexPos[a]);
166     colors.push(vertexColor[a]);
167     numVertCubeTri++;
168     points.push(vertexPos[b]);
169     colors.push(vertexColor[b]);
170     numVertCubeTri++;
171     points.push(vertexPos[c]);
172     colors.push(vertexColor[c]);
173     numVertCubeTri++;
174
175     points.push(vertexPos[a]);
176     colors.push(vertexColor[a]);
177     numVertCubeTri++;
178     points.push(vertexPos[c]);
179     colors.push(vertexColor[c]);
180     numVertCubeTri++;
181     points.push(vertexPos[d]);
182     colors.push(vertexColor[d]);
183     numVertCubeTri++;
184 }
185
186 function generateGround(scale) {
187     numVertGroundTri = 0;
188     // two triangles
189     points.push(vec4(scale, -1.0, -scale, 1.0));
190     colors.push(vec4(0.8, 0.8, 0.8, 1.0));
191     numVertGroundTri++;
192     points.push(vec4(-scale, -1.0, -scale, 1.0));
193     colors.push(vec4(0.8, 0.8, 0.8, 1.0));
194     numVertGroundTri++;
195     points.push(vec4(-scale, -1.0, scale, 1.0));
196     colors.push(vec4(0.8, 0.8, 0.8, 1.0));
197     numVertGroundTri++;
198
```

⊗ 0 ⚠ 0 Ln 127, Col 81 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > render
198
199 points.push(vec4(scale, -1.0, -scale, 1.0));
200 colors.push(vec4(0.8, 0.8, 0.8, 1.0));
201 numVertGroundTri++;
202 points.push(vec4(-scale, -1.0, scale, 1.0));
203 colors.push(vec4(0.8, 0.8, 0.8, 1.0));
204 numVertGroundTri++;
205 points.push(vec4(scale, -1.0, scale, 1.0));
206 colors.push(vec4(0.8, 0.8, 0.8, 1.0));
207 numVertGroundTri++;
208
209 numVertGroundLine = 0;
210 // grid lines
211 for(var x=-scale; x<=scale; x++) {
212 points.push(vec4(x, -1.0, -scale, 1.0));
213 colors.push(vec4(0.0, 0.0, 0.0, 1.0));
214 numVertGroundLine++;
215 points.push(vec4(x, -1.0, scale, 1.0));
216 colors.push(vec4(0.0, 0.0, 0.0, 1.0));
217 numVertGroundLine++;
218 }
219 for(var z=-scale; z<=scale; z++) {
220 points.push(vec4(-scale, -1.0, z, 1.0));
221 colors.push(vec4(0.0, 0.0, 0.0, 1.0));
222 numVertGroundLine++;
223 points.push(vec4(scale, -1.0, z, 1.0));
224 colors.push(vec4(0.0, 0.0, 0.0, 1.0));
225 numVertGroundLine++;
226 }
227 }
228
⊗ 0 ⚠ 0
Ln 127, Col 81 Spaces: 4 UTF-8 CRLF JavaScript


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

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

view.html | view.js | scene.html | scene.js

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

```
var gl;
var points = [];
var colors = [];

var modelViewMatrix, projectionMatrix;
var modelViewMatrixLoc, projectionMatrixLoc;
var eye = vec3(0.0, 2.0, 2.0);
var at = vec3(0.0, 0.0, 0.0);
const up = vec3(0.0, 1.0, 0.0);
var cameraVec = vec3(0, -0.7071, -0.7071); // 1.0/Math.sqrt(2.0)

var numVertCubeTri, numVertGroundTri, numVertGroundLine;

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

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

    generateColorCube();
    generateGround(10);

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.9, 0.9, 0.9, 1.0);

    // Enable hidden-surface removal
    gl.enable(gl.DEPTH_TEST);

    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);
```

⊗ 0 ⚠ 0 Ln 133, Col 61 Spaces: 4 UTF-8 CRLF JavaScript

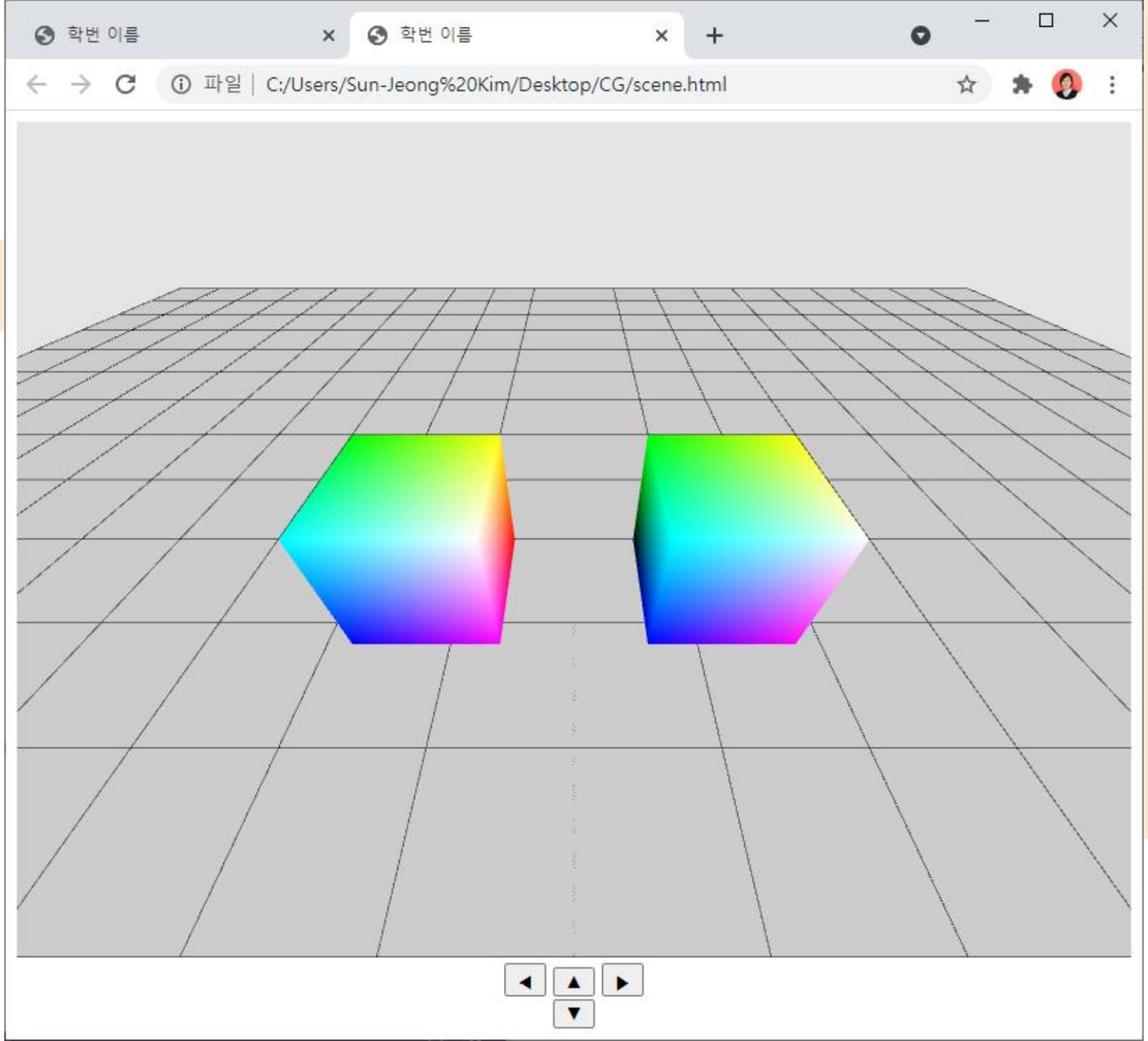
File Edit Selection View Go Run Terminal Help scene.js - Visual Studio Code

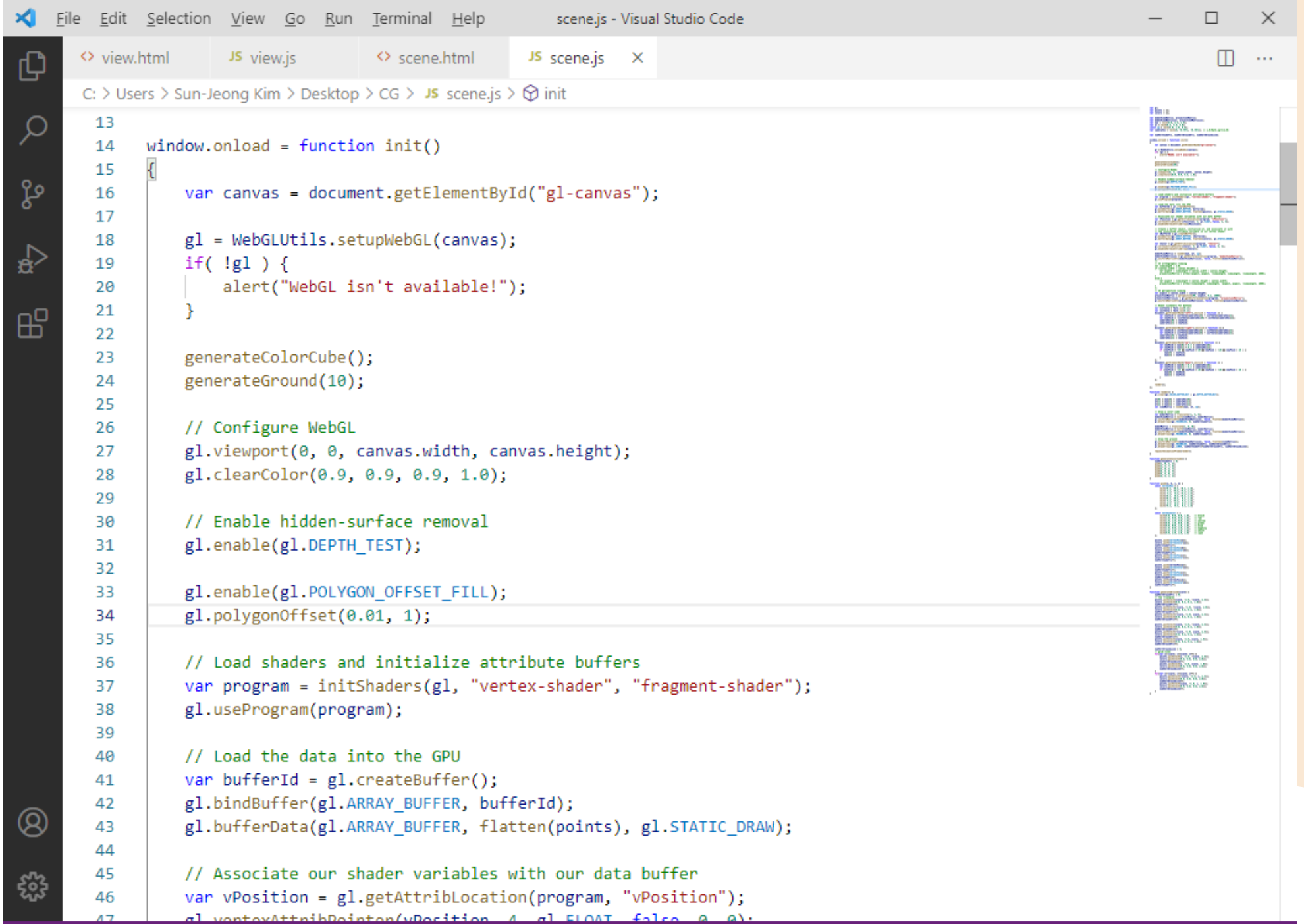
view.html | view.js | scene.html | scene.js

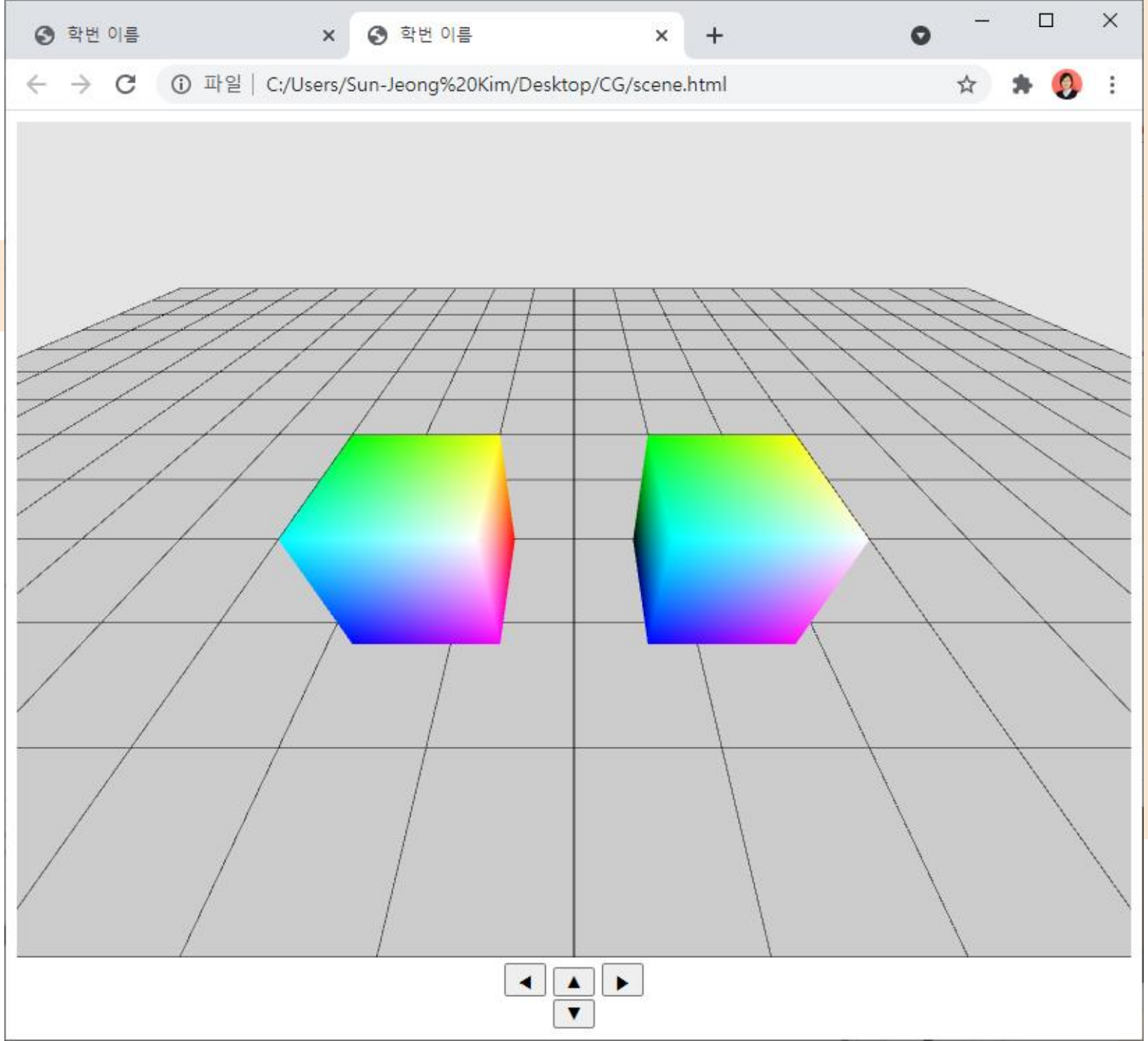
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > render

```
109
110     render();
111 };
112
113 function render() {
114     gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
115
116     at[0] = eye[0] + cameraVec[0];
117     at[1] = eye[1] + cameraVec[1];
118     at[2] = eye[2] + cameraVec[2];
119     var viewMatrix = lookAt(eye, at, up);
120
121     // draw a color cube
122     var modelMatrix = translate(-1, 0, 0);
123     modelViewMatrix = mult(viewMatrix, modelMatrix);
124     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
125     gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
126
127     modelMatrix = translate(1, 0, 0);
128     modelViewMatrix = mult(viewMatrix, modelMatrix);
129     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
130     gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
131
132     // draw the ground
133     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
134     gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertGroundTri);
135     gl.drawArrays(gl.LINES, numVertCubeTri+numVertGroundTri, numVertGroundLine);
136
137     requestAnimationFrame(render);
138 }
139
140 function generateColorCube() {
141     numVertCubeTri = 0;
142     quad(1, 0, 3, 2);
143     quad(2, 3, 7, 6);
```

⊗ 0 ⚠ 0 Ln 133, Col 61 Spaces: 4 UTF-8 CRLF JavaScript


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


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > init
13
14 window.onload = function init()
15 {
16 var canvas = document.getElementById("gl-canvas");
17
18 gl = WebGLUtils.setupWebGL(canvas);
19 if(!gl) {
20 alert("WebGL isn't available!");
21 }
22
23 generateColorCube();
24 generateGround(10);
25
26 // Configure WebGL
27 gl.viewport(0, 0, canvas.width, canvas.height);
28 gl.clearColor(0.9, 0.9, 0.9, 1.0);
29
30 // Enable hidden-surface removal
31 gl.enable(gl.DEPTH_TEST);
32
33 gl.enable(gl.POLYGON_OFFSET_FILL);
34 gl.polygonOffset(0.01, 1);
35
36 // Load shaders and initialize attribute buffers
37 var program = initShaders(gl, "vertex-shader", "fragment-shader");
38 gl.useProgram(program);
39
40 // Load the data into the GPU
41 var bufferId = gl.createBuffer();
42 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
43 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
44
45 // Associate our shader variables with our data buffer
46 var vPosition = gl.getAttribLocation(program, "vPosition");
⊗ 0 ⚠ 0
Ln 34, Col 31 Spaces: 4 UTF-8 CRLF JavaScript


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

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

view.html | view.js | scene.html | scene.js

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

```
1   var gl;
2   var points = [];
3   var colors = [];
4
5   var modelViewMatrix, projectionMatrix;
6   var modelViewMatrixLoc, projectionMatrixLoc;
7   var eye = vec3(0.0, 2.0, 2.0);
8   var at = vec3(0.0, 0.0, 0.0);
9   const up = vec3(0.0, 1.0, 0.0);
10  var cameraVec = vec3(0, -0.7071, -0.7071); // 1.0/Math.sqrt(2.0)
11
12  var numVertCubeTri, numVertGroundTri, numVertGroundLine;
13  var theta = 0;
14
15  window.onload = function init()
16  {
17      var canvas = document.getElementById("gl-canvas");
18
19      gl = WebGLUtils.setupWebGL(canvas);
20      if( !gl ) {
21          alert("WebGL isn't available!");
22      }
23
24      generateColorCube();
25      generateGround(10);
26
27      // Configure WebGL
28      gl.viewport(0, 0, canvas.width, canvas.height);
29      gl.clearColor(0.9, 0.9, 0.9, 1.0);
30
31      // Enable hidden-surface removal
32      gl.enable(gl.DEPTH_TEST);
33
34      gl.enable(gl.POLYGON_OFFSET_FILL);
35      gl.polygonOffset(0.01, 1);
```

⊗ 0 ⚠ 0 Ln 13, Col 10 (5 selected) Spaces: 4 UTF-8 CRLF JavaScript

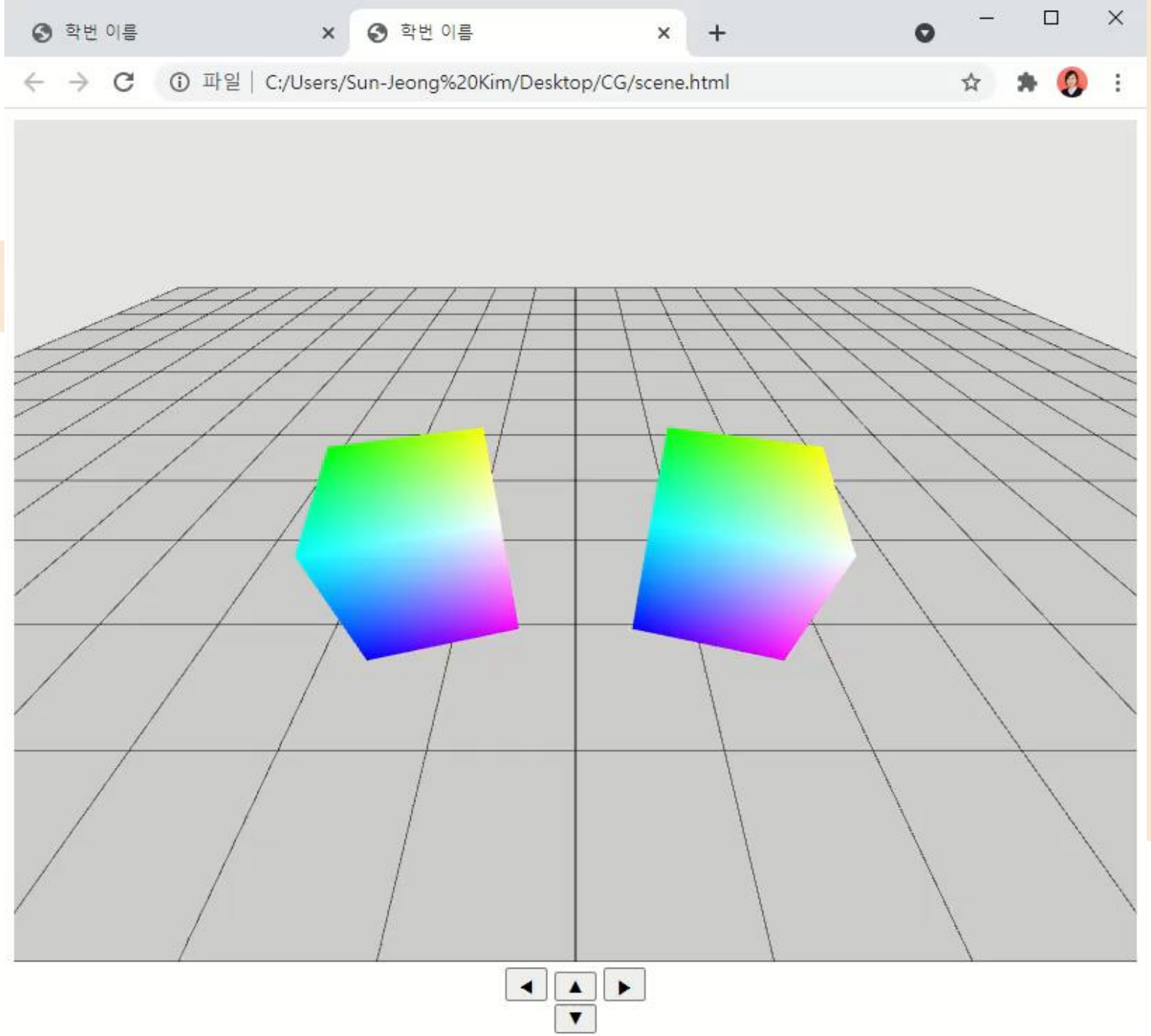
File Edit Selection View Go Run Terminal Help scene.js - Visual Studio Code

view.html | view.js | scene.html | scene.js

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

```
116
117 function render() {
118     gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
119
120     at[0] = eye[0] + cameraVec[0];
121     at[1] = eye[1] + cameraVec[1];
122     at[2] = eye[2] + cameraVec[2];
123     var viewMatrix = lookAt(eye, at, up);
124
125     theta += 2;
126
127     // draw a color cube
128     var modelMatrix = mult(translate(-1, 0, 0), rotate(theta, vec3(0, 1, 0)));
129     modelViewMatrix = mult(viewMatrix, modelMatrix);
130     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
131     gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
132
133     modelMatrix = mult(translate(1, 0, 0), rotate(-theta, vec3(0, 1, 0)));
134     modelViewMatrix = mult(viewMatrix, modelMatrix);
135     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
136     gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
137
138     // draw the ground
139     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
140     gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertGroundTri);
141     gl.drawArrays(gl.LINES, numVertCubeTri+numVertGroundTri, numVertGroundLine);
142
143     requestAnimationFrame(render);
144 }
145
146 function generateColorCube() {
147     numVertCubeTri = 0;
148     quad(1, 0, 3, 2);
149     quad(2, 3, 7, 6);
150     quad(3, 0, 4, 7);
```

⊗ 0 ⚠ 0 Ln 13, Col 10 (5 selected) Spaces: 4 UTF-8 CRLF JavaScript


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

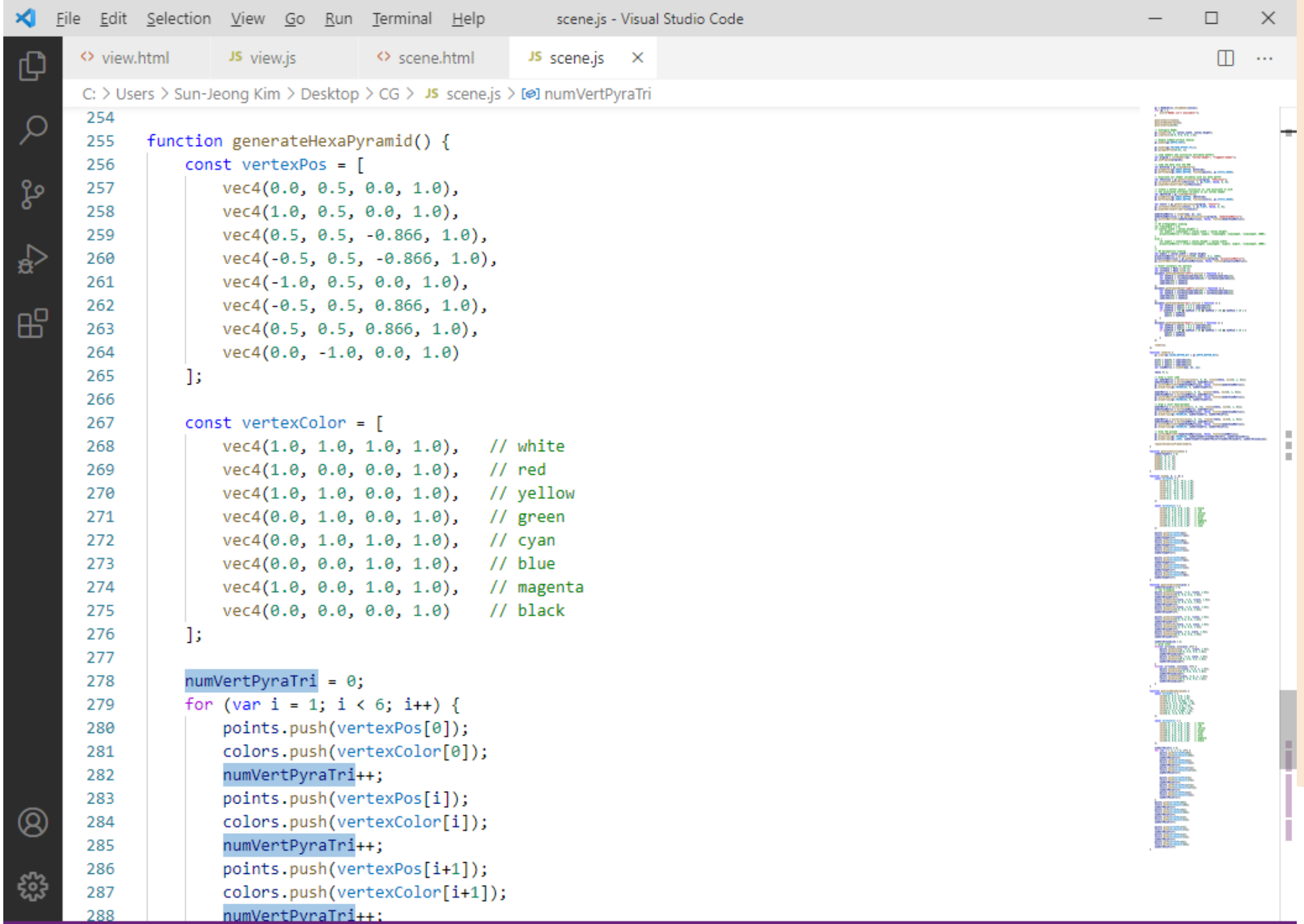
File Edit Selection View Go Run Terminal Help scene.js - Visual Studio Code

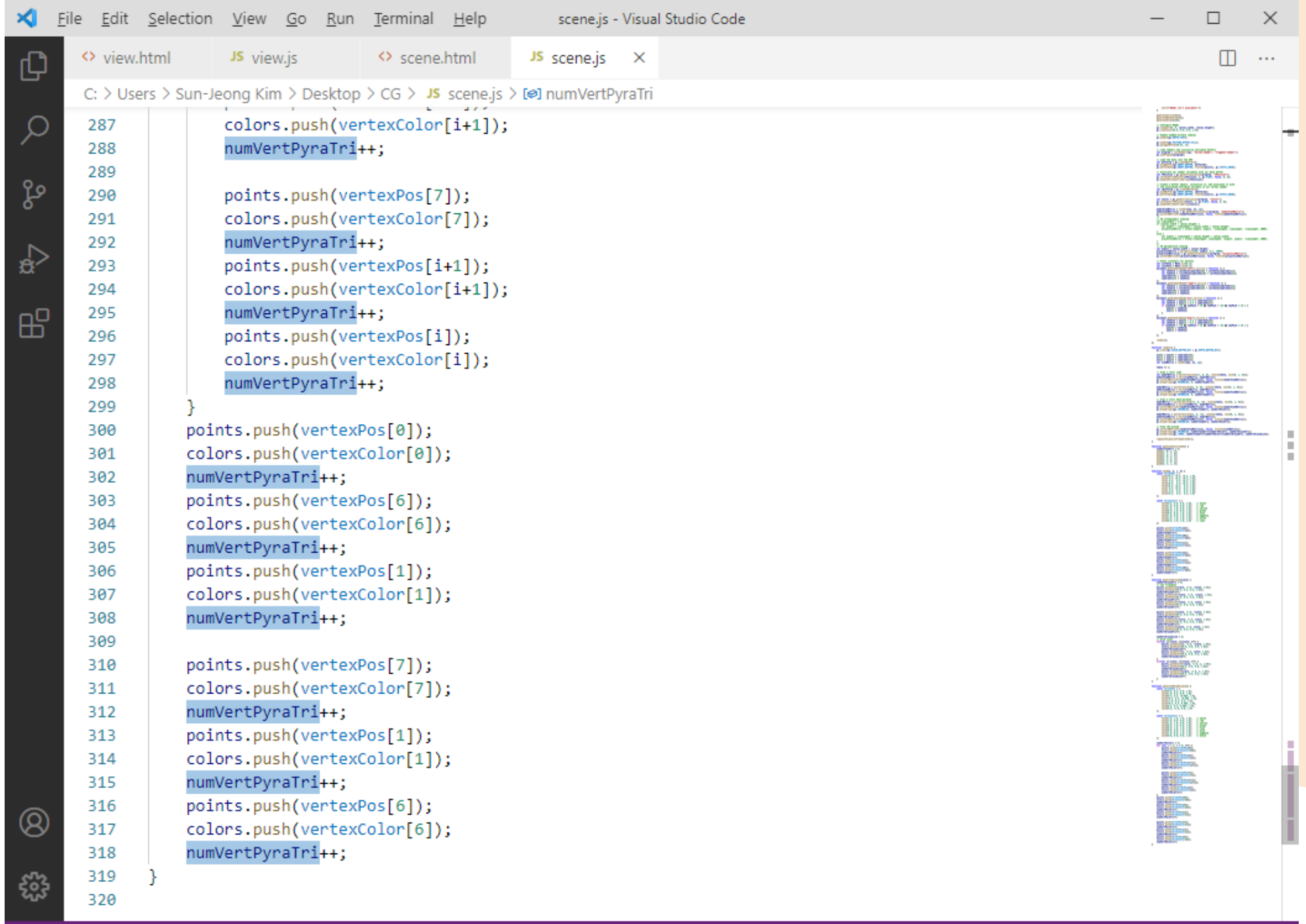
view.html | view.js | scene.html | scene.js

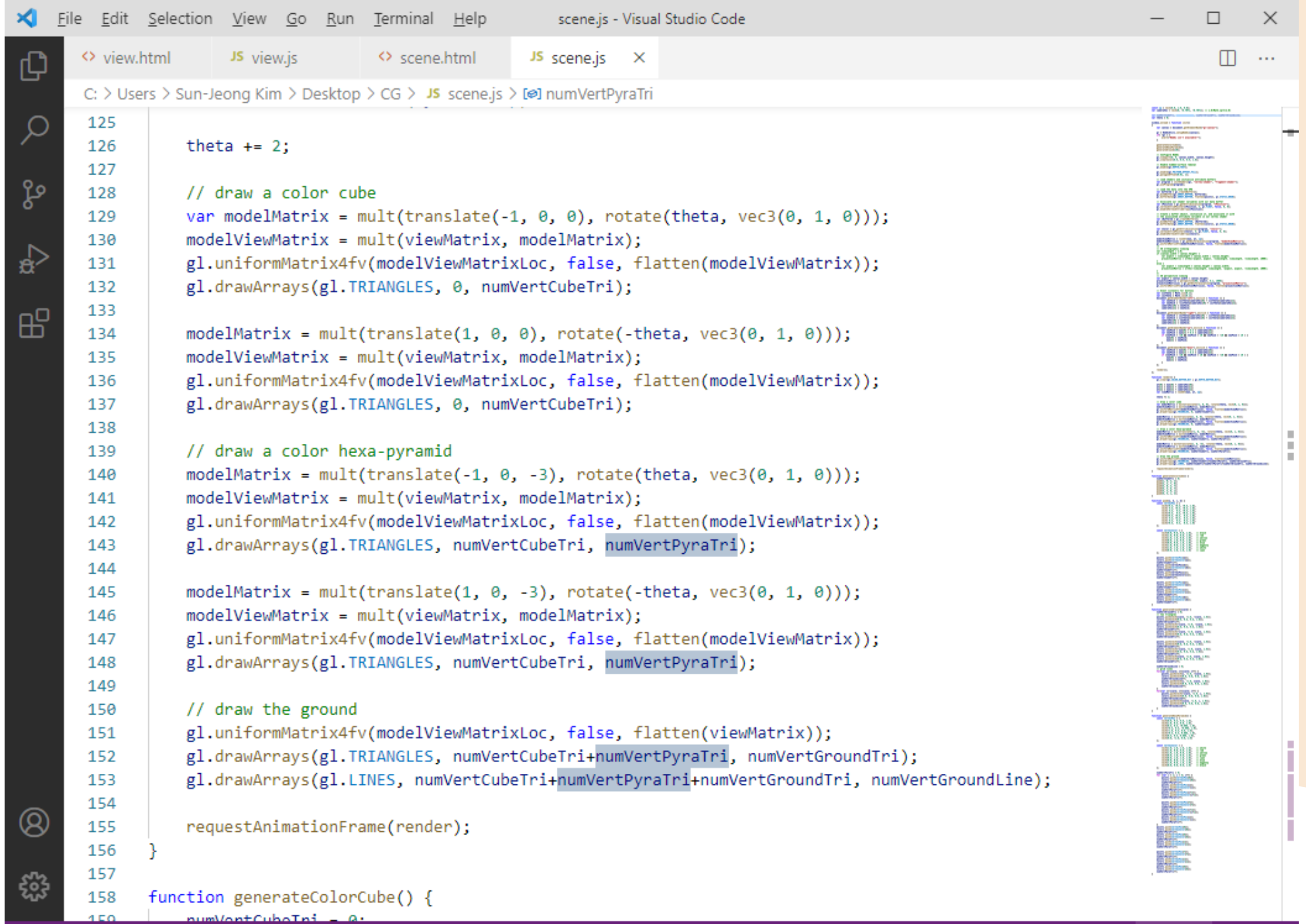
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > numVertPyraTri

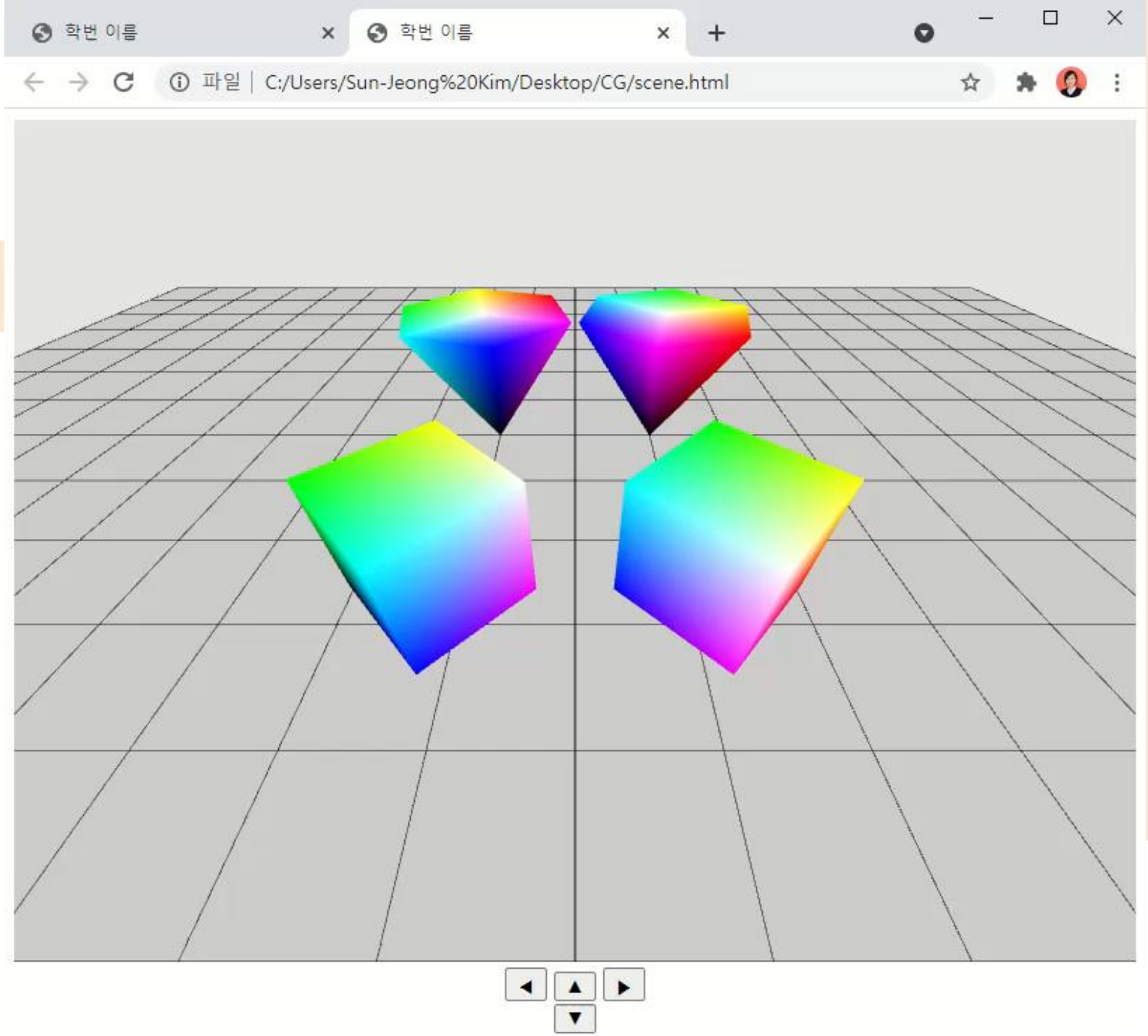
```
var gl;
var points = [];
var colors = [];

var modelViewMatrix, projectionMatrix;
var modelViewMatrixLoc, projectionMatrixLoc;
var eye = vec3(0.0, 2.0, 2.0);
var at = vec3(0.0, 0.0, 0.0);
const up = vec3(0.0, 1.0, 0.0);
var cameraVec = vec3(0, -0.7071, -0.7071); // 1.0/Math.sqrt(2.0)

var numVertCubeTri, numVertPyraTri, numVertGroundTri, numVertGroundLine;
var theta = 0;

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

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

    generateColorCube();
    generateHexaPyramid();
    generateGround(10);

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.9, 0.9, 0.9, 1.0);

    // Enable hidden-surface removal
    gl.enable(gl.DEPTH_TEST);

    gl.enable(gl.POLYGON_OFFSET_FILL);
```

⊗ 0 ⚠ 0 Ln 12, Col 35 (14 selected) Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > numVertPyraTri
254
255 function generateHexaPyramid() {
256 const vertexPos = [
257 vec4(0.0, 0.5, 0.0, 1.0),
258 vec4(1.0, 0.5, 0.0, 1.0),
259 vec4(0.5, 0.5, -0.866, 1.0),
260 vec4(-0.5, 0.5, -0.866, 1.0),
261 vec4(-1.0, 0.5, 0.0, 1.0),
262 vec4(-0.5, 0.5, 0.866, 1.0),
263 vec4(0.5, 0.5, 0.866, 1.0),
264 vec4(0.0, -1.0, 0.0, 1.0)
265];
266
267 const vertexColor = [
268 vec4(1.0, 1.0, 1.0, 1.0), // white
269 vec4(1.0, 0.0, 0.0, 1.0), // red
270 vec4(1.0, 1.0, 0.0, 1.0), // yellow
271 vec4(0.0, 1.0, 0.0, 1.0), // green
272 vec4(0.0, 1.0, 1.0, 1.0), // cyan
273 vec4(0.0, 0.0, 1.0, 1.0), // blue
274 vec4(1.0, 0.0, 1.0, 1.0), // magenta
275 vec4(0.0, 0.0, 0.0, 1.0) // black
276];
277
278 numVertPyraTri = 0;
279 for (var i = 1; i < 6; i++) {
280 points.push(vertexPos[0]);
281 colors.push(vertexColor[0]);
282 numVertPyraTri++;
283 points.push(vertexPos[i]);
284 colors.push(vertexColor[i]);
285 numVertPyraTri++;
286 points.push(vertexPos[i+1]);
287 colors.push(vertexColor[i+1]);
288 numVertPyraTri++;
⊗ 0 ⚠ 0
Ln 12, Col 35 (14 selected)
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > numVertPyraTri
colors.push(vertexColor[i+1]);
numVertPyraTri++;
points.push(vertexPos[7]);
colors.push(vertexColor[7]);
numVertPyraTri++;
points.push(vertexPos[i+1]);
colors.push(vertexColor[i+1]);
numVertPyraTri++;
points.push(vertexPos[i]);
colors.push(vertexColor[i]);
numVertPyraTri++;
}
points.push(vertexPos[0]);
colors.push(vertexColor[0]);
numVertPyraTri++;
points.push(vertexPos[6]);
colors.push(vertexColor[6]);
numVertPyraTri++;
points.push(vertexPos[1]);
colors.push(vertexColor[1]);
numVertPyraTri++;
points.push(vertexPos[7]);
colors.push(vertexColor[7]);
numVertPyraTri++;
points.push(vertexPos[1]);
colors.push(vertexColor[1]);
numVertPyraTri++;
points.push(vertexPos[6]);
colors.push(vertexColor[6]);
numVertPyraTri++;
}
0 0
Ln 12, Col 35 (14 selected)
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > numVertPyraTri
125
126 theta += 2;
127
128 // draw a color cube
129 var modelMatrix = mult(translate(-1, 0, 0), rotate(theta, vec3(0, 1, 0)));
130 modelViewMatrix = mult(viewMatrix, modelMatrix);
131 gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
132 gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
133
134 modelMatrix = mult(translate(1, 0, 0), rotate(-theta, vec3(0, 1, 0)));
135 modelViewMatrix = mult(viewMatrix, modelMatrix);
136 gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
137 gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
138
139 // draw a color hexa-pyramid
140 modelMatrix = mult(translate(-1, 0, -3), rotate(theta, vec3(0, 1, 0)));
141 modelViewMatrix = mult(viewMatrix, modelMatrix);
142 gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
143 gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);
144
145 modelMatrix = mult(translate(1, 0, -3), rotate(-theta, vec3(0, 1, 0)));
146 modelViewMatrix = mult(viewMatrix, modelMatrix);
147 gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
148 gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);
149
150 // draw the ground
151 gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
152 gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri, numVertGroundTri);
153 gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertGroundTri, numVertGroundLine);
154
155 requestAnimationFrame(render);
156 }
157
158 function generateColorCube() {
⊗ 0 ⚠ 0
Ln 12, Col 35 (14 selected)
Spaces: 4
UTF-8
CRLF
JavaScript


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

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

view.html | view.js | scene.html | scene.js

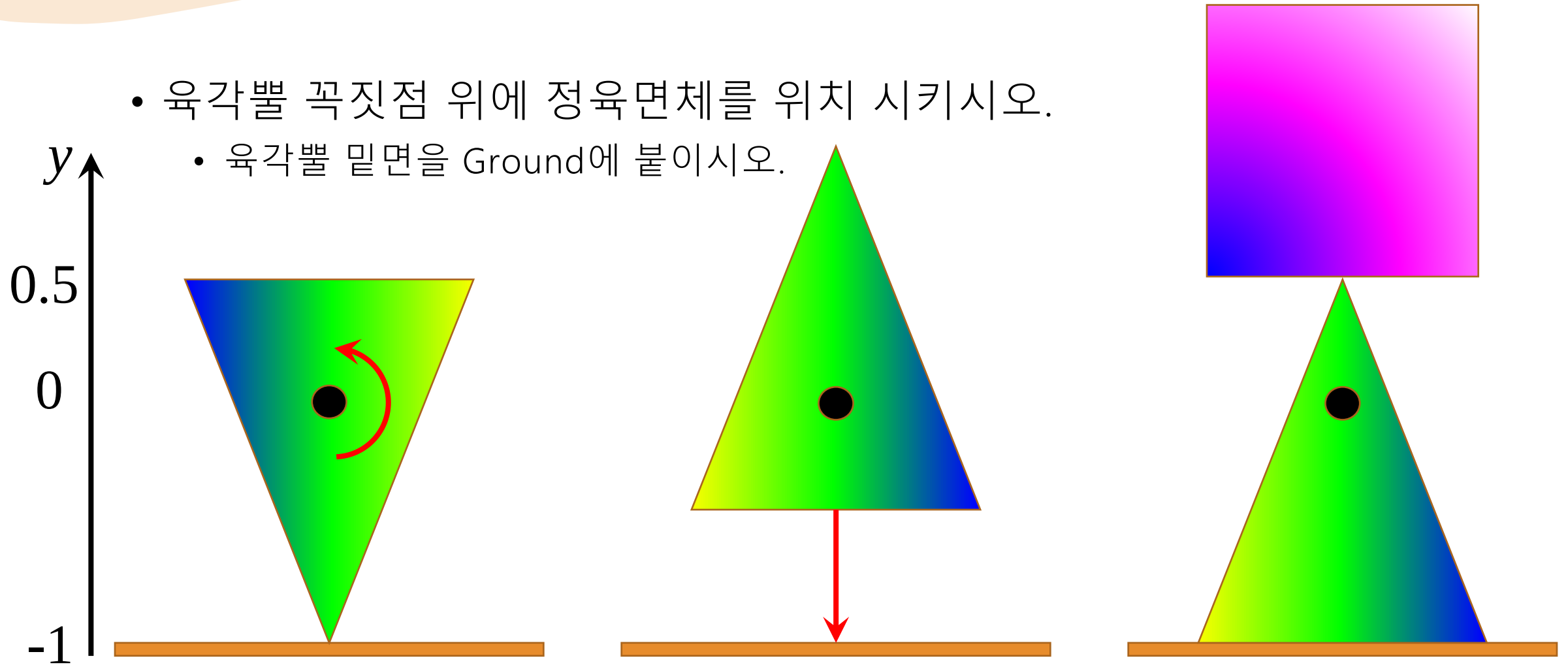
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > render

```
125
126     theta += 2;
127
128     // draw a color cube
129     var modelMatrix = mult(translate(-1, 0, 0), rotate(theta, vec3(0, 1, 0)));
130     modelViewMatrix = mult(viewMatrix, modelMatrix);
131     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
132     gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
133
134     modelMatrix = mult(translate(1, 0, 0), rotate(-theta, vec3(0, 1, 0)));
135     modelViewMatrix = mult(viewMatrix, modelMatrix);
136     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
137     gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
138
139     // draw a color hexa-pyramid
140     modelMatrix = mult(translate(-1, 0, -3), rotate(theta, vec3(0, 1, 0)));
141     modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
142     modelViewMatrix = mult(viewMatrix, modelMatrix);
143     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
144     gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);
145
146     modelMatrix = mult(translate(1, 0, -3), rotate(-theta, vec3(0, 1, 0)));
147     modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
148     modelViewMatrix = mult(viewMatrix, modelMatrix);
149     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
150     gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);
151
152     // draw the ground
153     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
154     gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri, numVertGroundTri);
155     gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertGroundTri, numVertGroundLine);
156
157     requestAnimationFrame(render);
158 }
```

⊗ 0 ⚠ 0 Ln 141, Col 45 Spaces: 4 UTF-8 CRLF JavaScript

연습 문제 (1)

- 육각뿔 꼭짓점 위에 정육면체를 위치 시키시오.
  - 육각뿔 밑면을 Ground에 붙이시오.



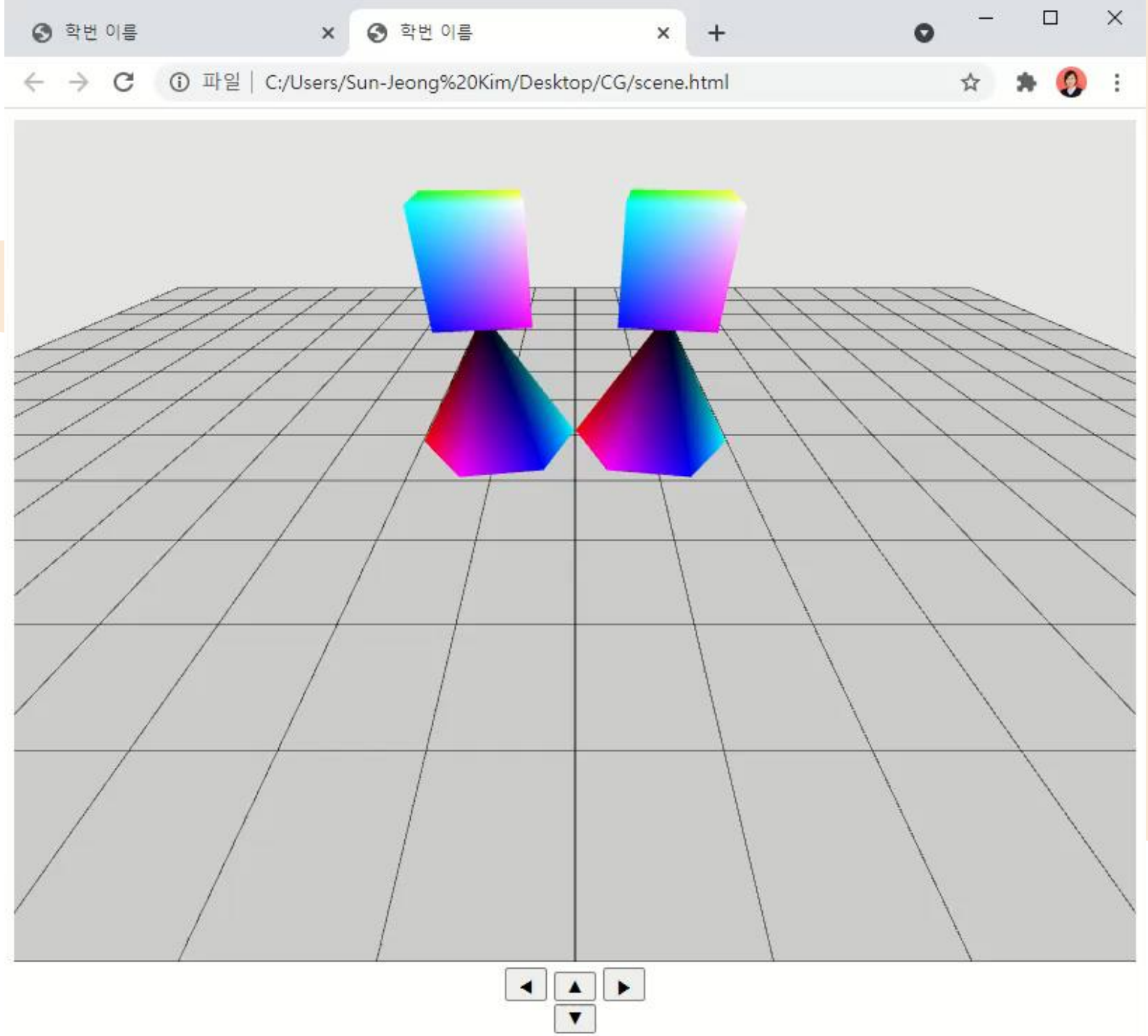
File Edit Selection View Go Run Terminal Help scene.js - Visual Studio Code

view.html | view.js | scene.html | scene.js

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

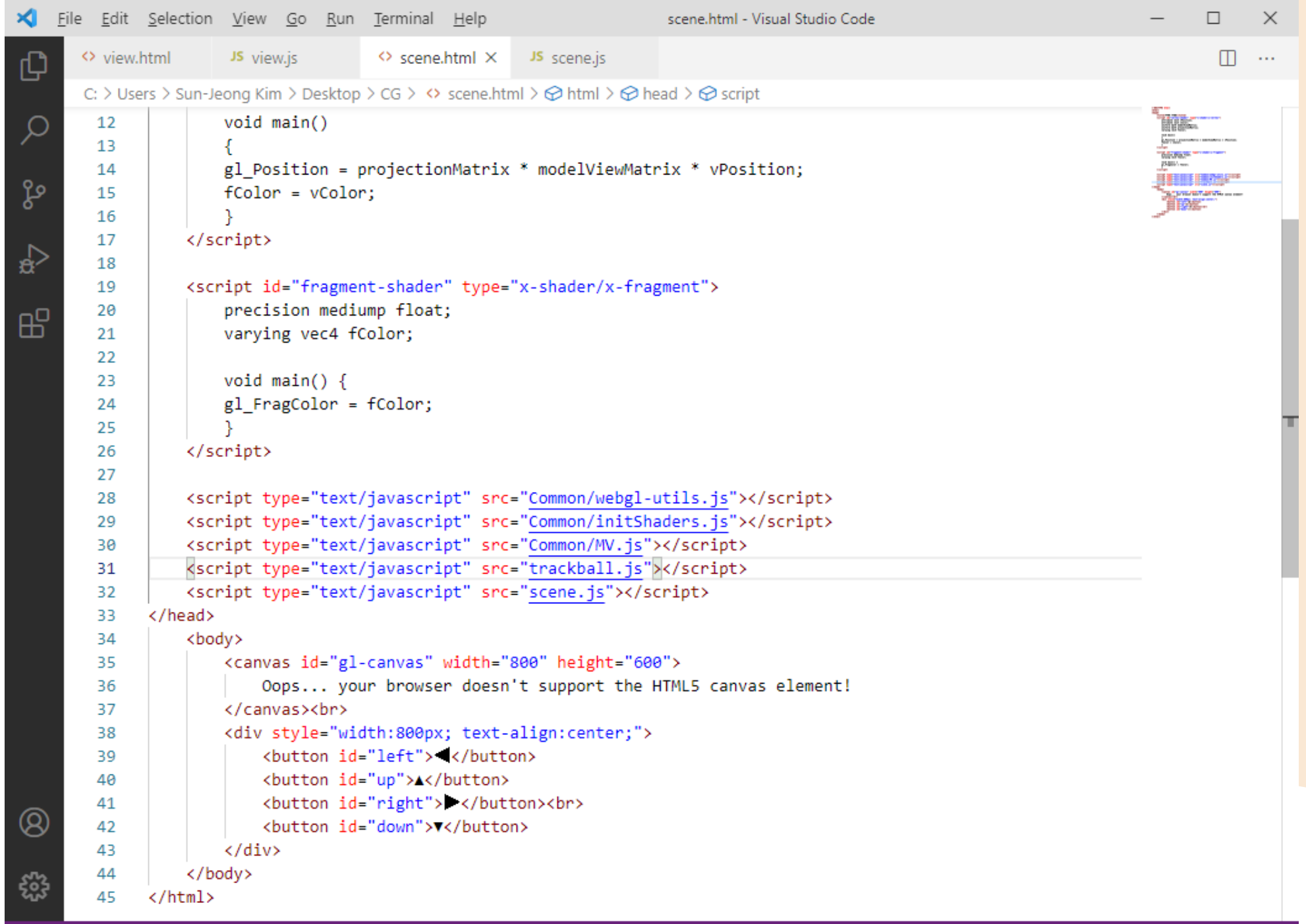
```
125
126     theta += 2;
127
128     // draw a color cube
129     var modelMatrix = mult(translate(-1, 1.0, -3), rotate(theta, vec3(0, 1, 0)));
130     modelViewMatrix = mult(viewMatrix, modelMatrix);
131     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
132     gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
133
134     modelMatrix = mult(translate(1, 1.0, -3), rotate(-theta, vec3(0, 1, 0)));
135     modelViewMatrix = mult(viewMatrix, modelMatrix);
136     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
137     gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
138
139     // draw a color hexa-pyramid
140     modelMatrix = mult(translate(-1, -0.5, -3), rotate(theta, vec3(0, 1, 0)));
141     modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
142     modelViewMatrix = mult(viewMatrix, modelMatrix);
143     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
144     gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);
145
146     modelMatrix = mult(translate(1, -0.5, -3), rotate(-theta, vec3(0, 1, 0)));
147     modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
148     modelViewMatrix = mult(viewMatrix, modelMatrix);
149     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
150     gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);
151
152     // draw the ground
153     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
154     gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri, numVertGroundTri);
155     gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertGroundTri, numVertGroundLine);
156
157     requestAnimationFrame(render);
158 }
159
```

⊗ 0 ⚠ 0 Ln 146, Col 41 Spaces: 4 UTF-8 CRLF JavaScript


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

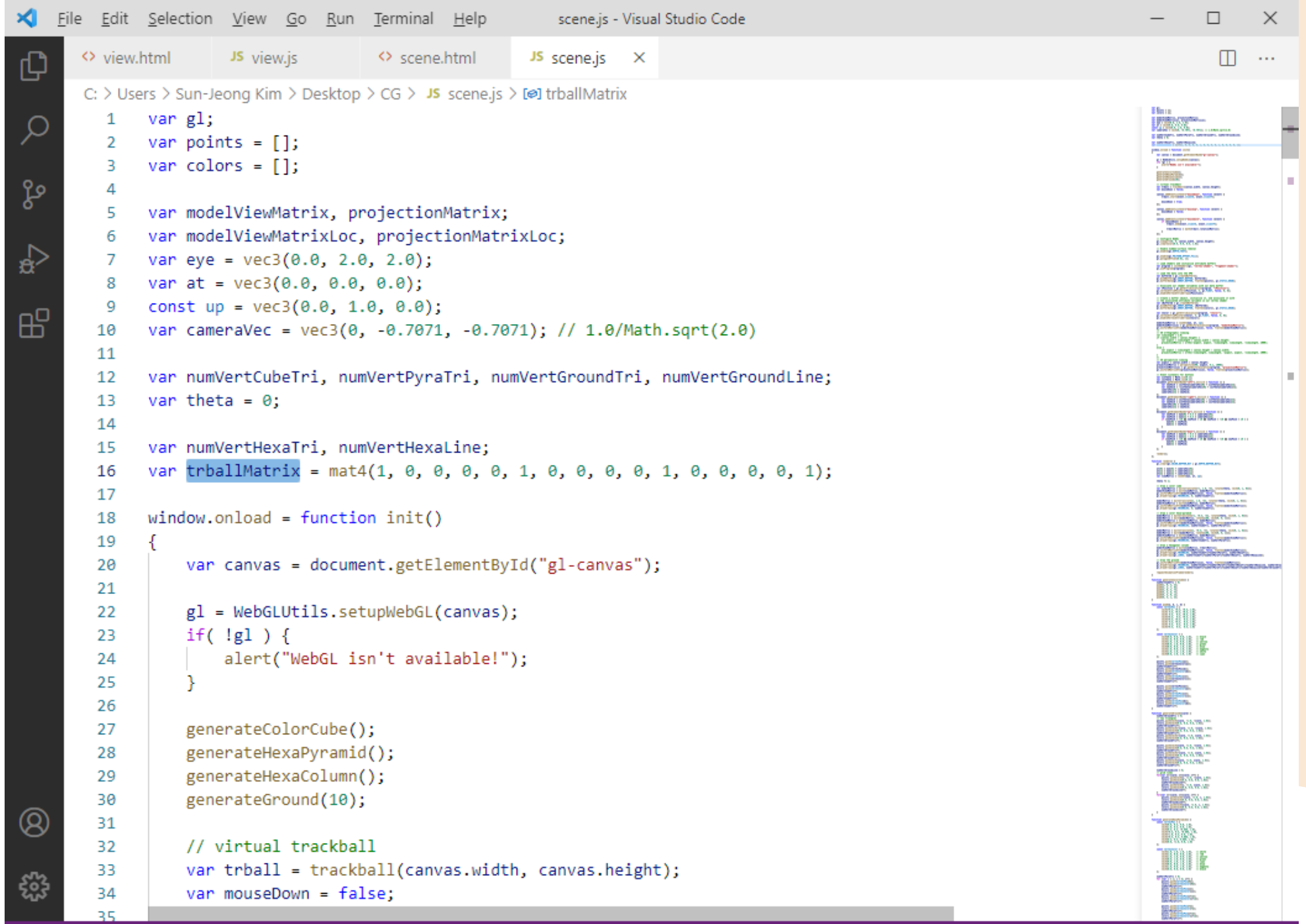
연습 문제 (2)

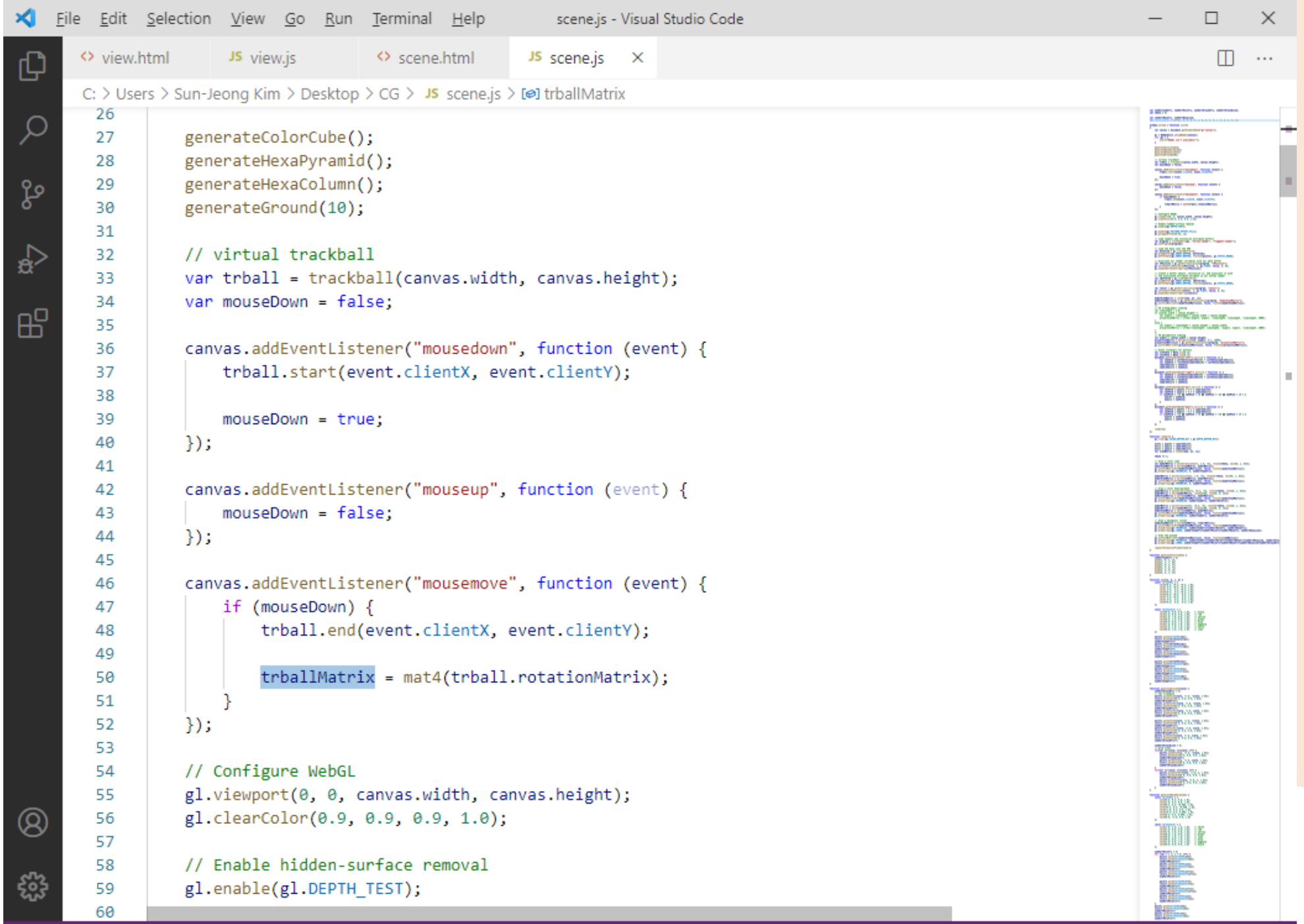
- 육각기둥을 모델링하여 화면 중앙에 위치 시키시오.
- 육각기둥은 단색으로 색칠하시오.
  - 선도 함께 그리시오.
- 육각기둥은 트랙볼로 회전 시키시오.


File Edit Selection View Go Run Terminal Help
scene.html - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.html > html > head > script
void main()
{
gl_Position = projectionMatrix * modelViewMatrix * vPosition;
fColor = vColor;
}
</script>
<script id="fragment-shader" type="x-shader/x-fragment">
precision mediump float;
varying vec4 fColor;
void main() {
gl_FragColor = fColor;
}
</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="scene.js"></script>
</head>
<body>
<canvas id="gl-canvas" width="800" height="600">
Oops... your browser doesn't support the HTML5 canvas element!
</canvas>

<button id="left">◀</button>
<button id="up">▲</button>
<button id="right">▶</button>
<button id="down">▼</button>

</body>
</html>
⊗ 0 ⚠ 0
Ln 31, Col 6
Spaces: 4
UTF-8
CRLF
HTML


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > trballMatrix
1 var gl;
2 var points = [];
3 var colors = [];
4
5 var modelViewMatrix, projectionMatrix;
6 var modelViewMatrixLoc, projectionMatrixLoc;
7 var eye = vec3(0.0, 2.0, 2.0);
8 var at = vec3(0.0, 0.0, 0.0);
9 const up = vec3(0.0, 1.0, 0.0);
10 var cameraVec = vec3(0, -0.7071, -0.7071); // 1.0/Math.sqrt(2.0)
11
12 var numVertCubeTri, numVertPyraTri, numVertGroundTri, numVertGroundLine;
13 var theta = 0;
14
15 var numVertHexaTri, numVertHexaLine;
16 var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);
17
18 window.onload = function init()
19 {
20 var canvas = document.getElementById("gl-canvas");
21
22 gl = WebGLUtils.setupWebGL(canvas);
23 if(!gl) {
24 alert("WebGL isn't available!");
25 }
26
27 generateColorCube();
28 generateHexaPyramid();
29 generateHexaColumn();
30 generateGround(10);
31
32 // virtual trackball
33 var trball = trackball(canvas.width, canvas.height);
34 var mouseDown = false;
35
⊗ 0 ⚠ 0
Ln 16, Col 17 (12 selected)
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > trballMatrix
generateColorCube();
generateHexaPyramid();
generateHexaColumn();
generateGround(10);
// virtual trackball
var trball = trackball(canvas.width, canvas.height);
var mouseDown = false;
canvas.addEventListener("mousedown", function (event) {
trball.start(event.clientX, event.clientY);
mouseDown = true;
});
canvas.addEventListener("mouseup", function (event) {
mouseDown = false;
});
canvas.addEventListener("mousemove", function (event) {
if (mouseDown) {
trball.end(event.clientX, event.clientY);
trballMatrix = mat4(trball.rotationMatrix);
}
});
// Configure WebGL
gl.viewport(0, 0, canvas.width, canvas.height);
gl.clearColor(0.9, 0.9, 0.9, 1.0);
// Enable hidden-surface removal
gl.enable(gl.DEPTH_TEST);
Ln 16, Col 17 (12 selected)
Spaces: 4
UTF-8
CRLF
JavaScript

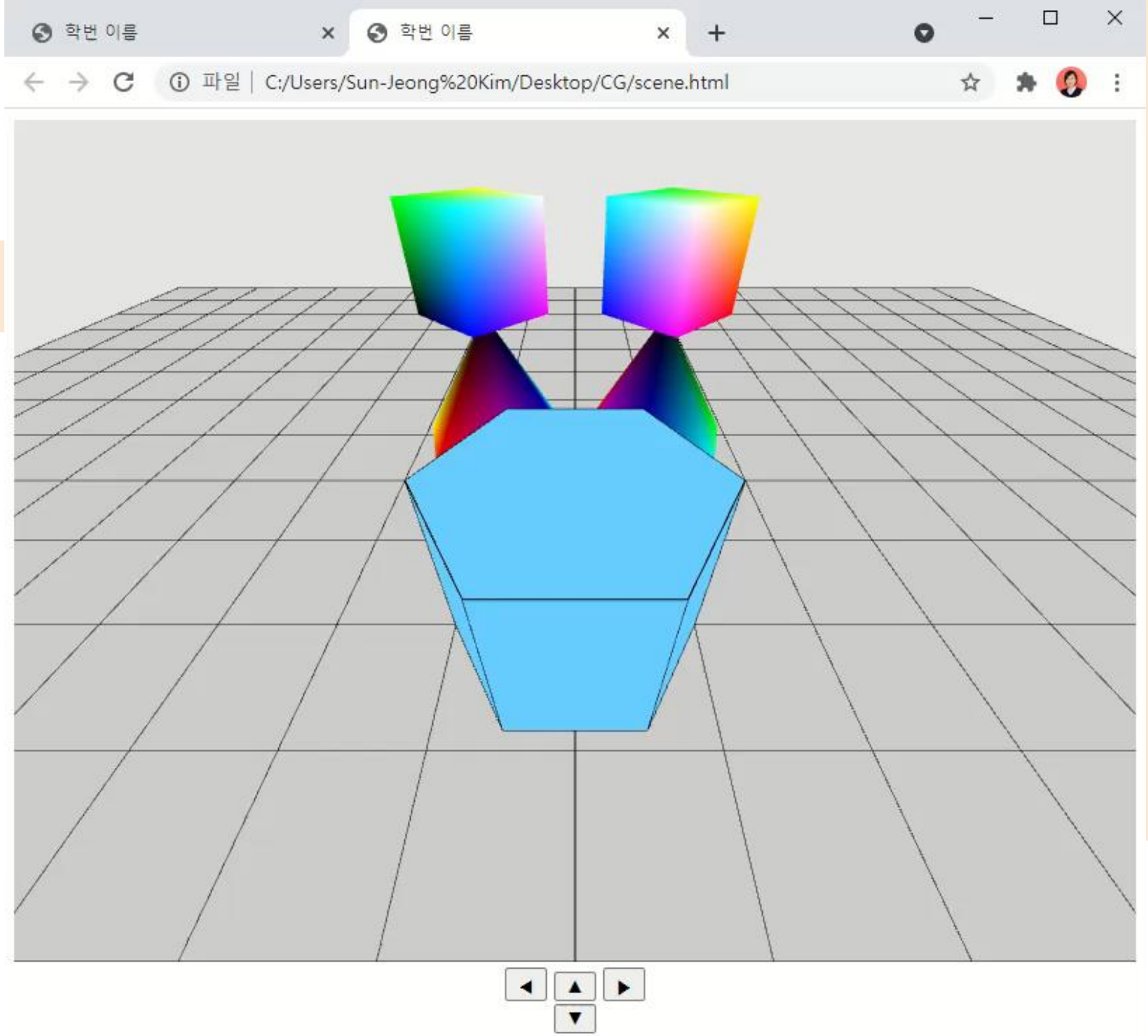
File Edit Selection View Go Run Terminal Help scene.js - Visual Studio Code

view.html | view.js | scene.html | scene.js

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

```
157     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
158     gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
159
160     modelMatrix = mult(translate(1, 1.0, -3), rotate(-theta, vec3(0, 1, 0)));
161     modelViewMatrix = mult(viewMatrix, modelMatrix);
162     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
163     gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);
164
165     // draw a color hexa-pyramid
166     modelMatrix = mult(translate(-1, -0.5, -3), rotate(theta, vec3(0, 1, 0)));
167     modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
168     modelViewMatrix = mult(viewMatrix, modelMatrix);
169     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
170     gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);
171
172     modelMatrix = mult(translate(1, -0.5, -3), rotate(-theta, vec3(0, 1, 0)));
173     modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
174     modelViewMatrix = mult(viewMatrix, modelMatrix);
175     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
176     gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);
177
178     // draw a hexagonal column
179     modelViewMatrix = mult(viewMatrix, trballMatrix);
180     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
181     gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri, numVertHexaTri);
182     gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertHexaTri, numVertHexaLine);
183
184     // draw the ground
185     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
186     gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri+numVertHexaTri+numVertHexaLine, numVertGroundTri);
187     gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertHexaTri+numVertHexaLine+numVertGroundTri, numVertGroundLine);
188
189     requestAnimationFrame(render);
190 }
191
```

⊗ 0 ⚠ 0 Ln 16, Col 17 (12 selected) Spaces: 4 UTF-8 CRLF JavaScript


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

수고하셨습니다