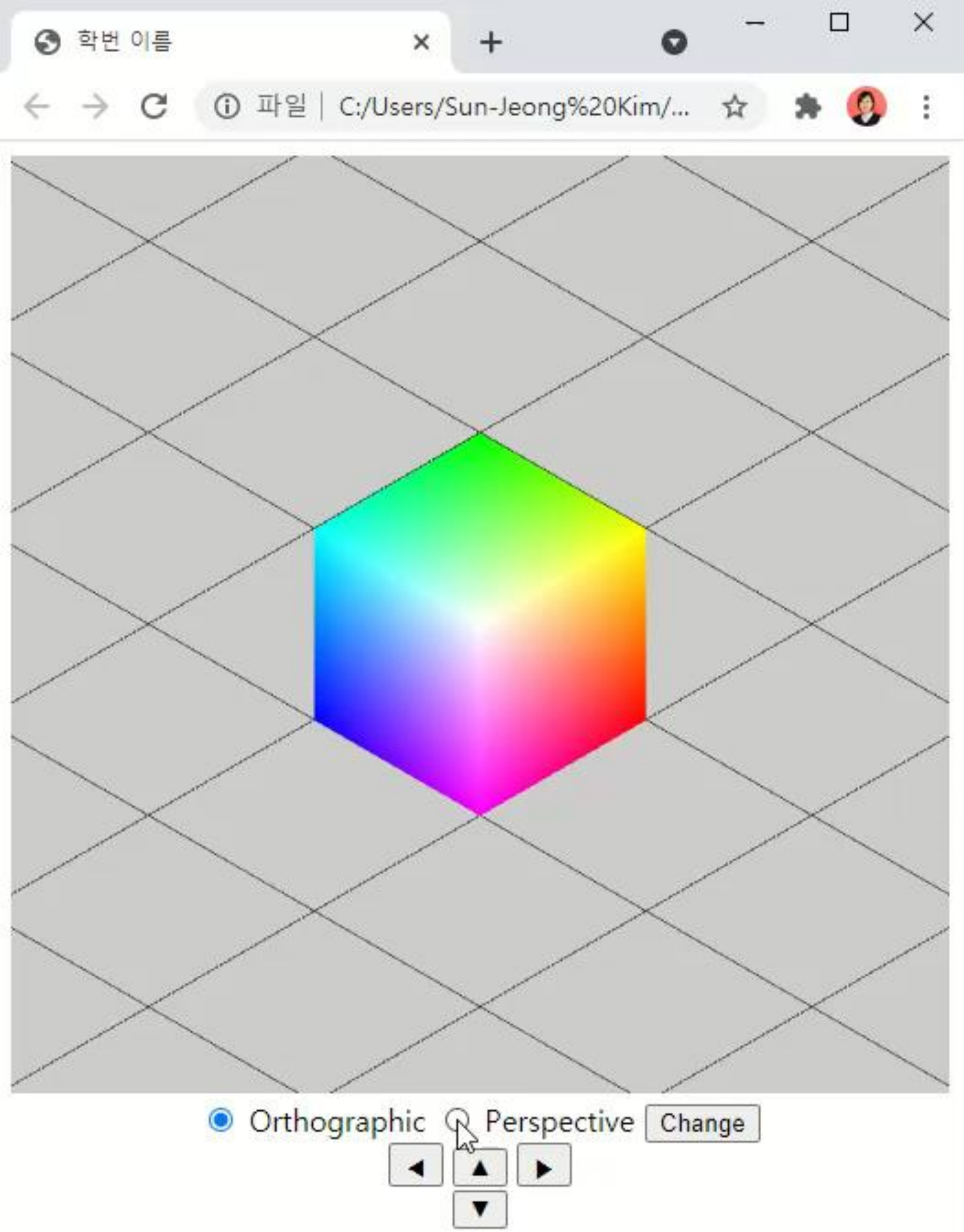
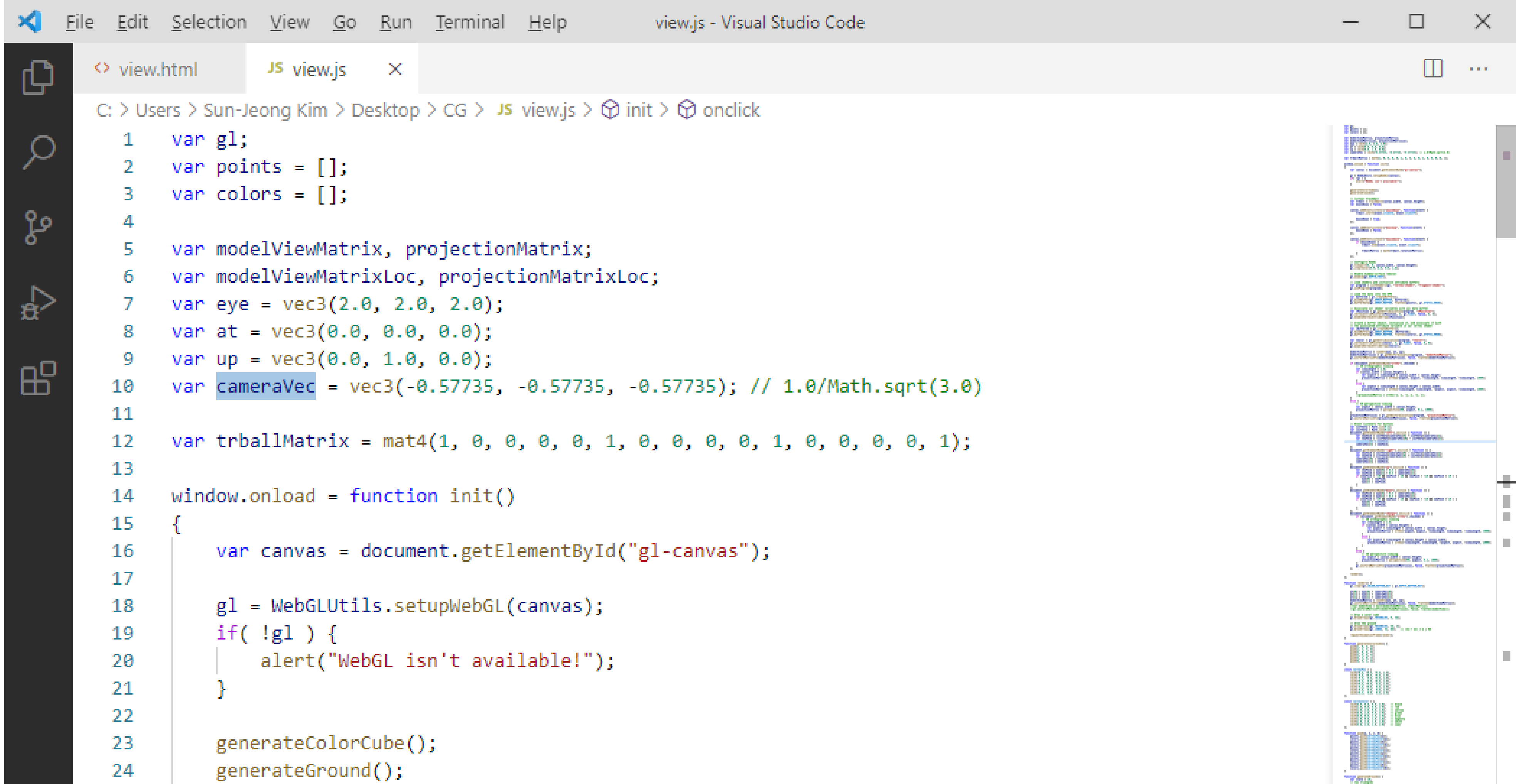
Building a Scene

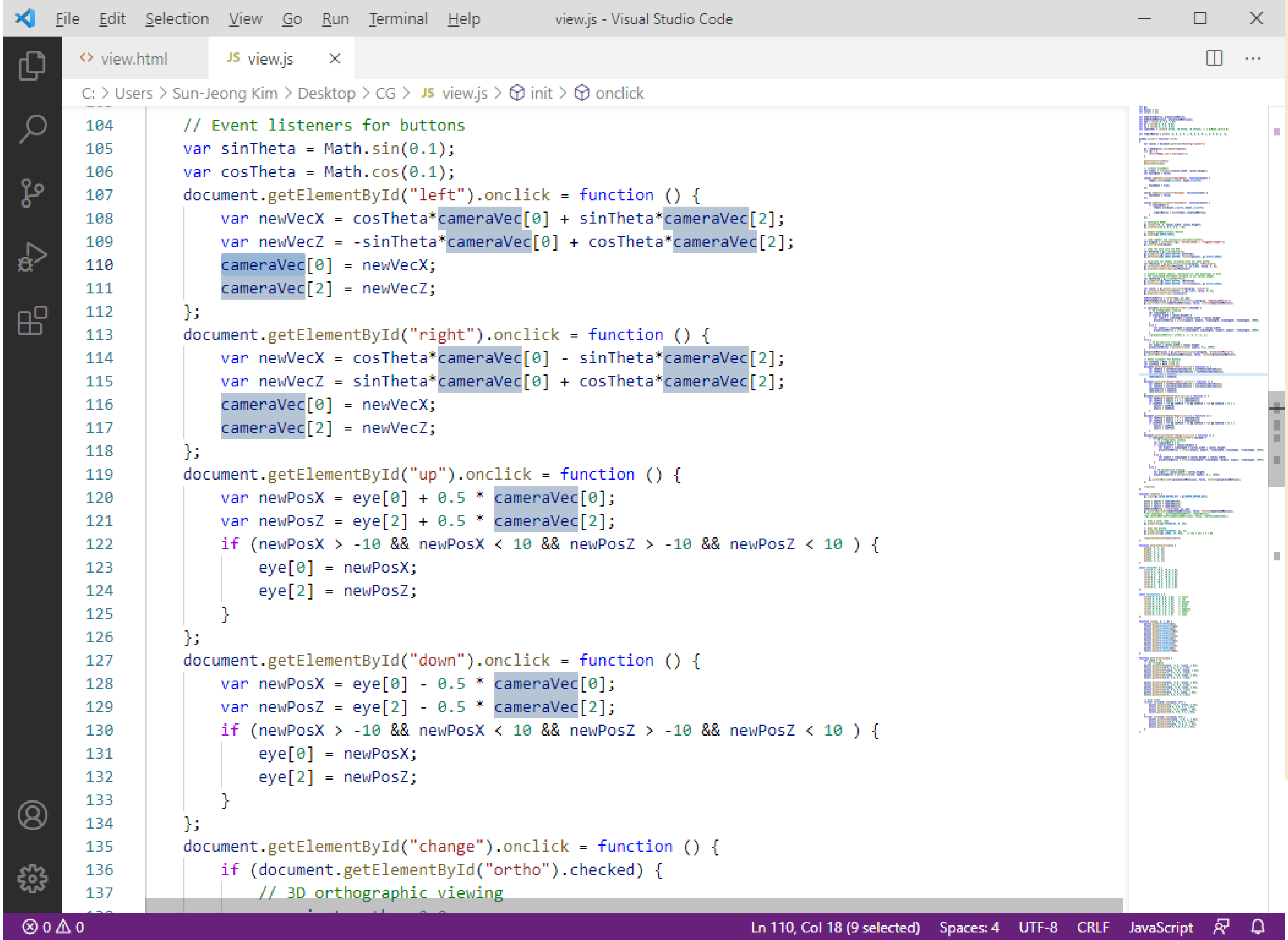
10TH WEEK, 2021


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/...
● Orthographic ○ Perspective
Change
◀
▲
▶
▼

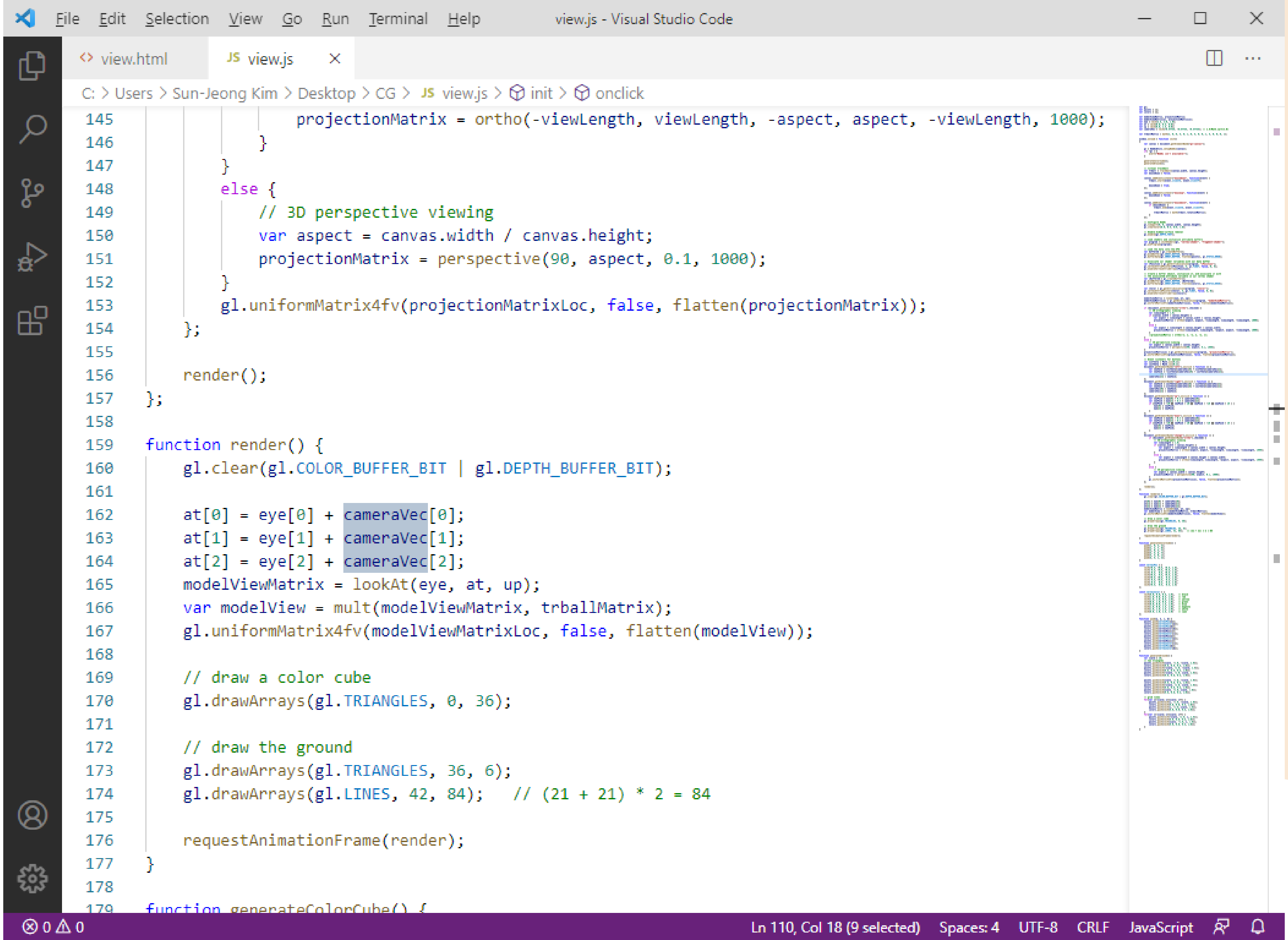
Walking Through


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

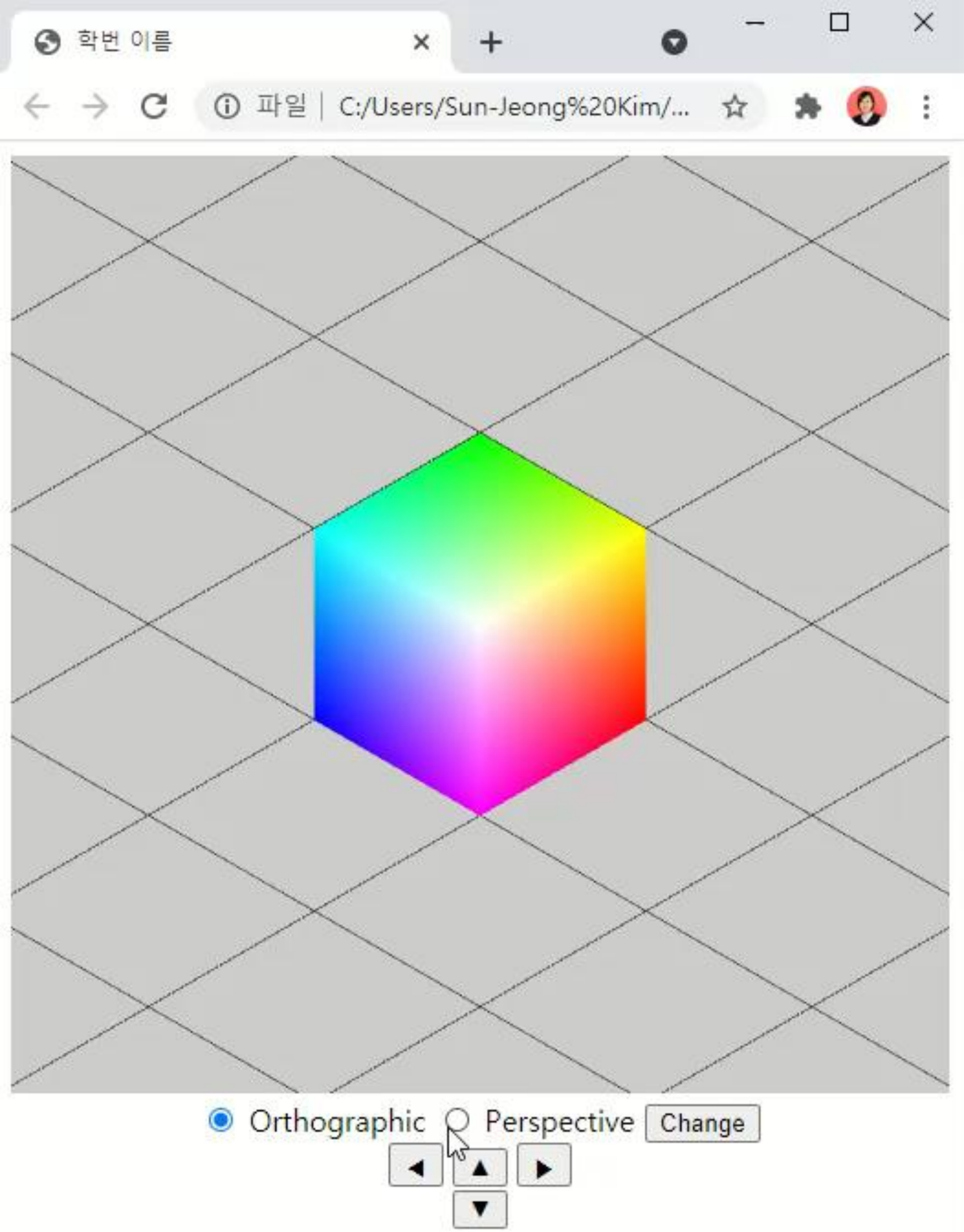
```javascript
var gl;
var points = [];
var colors = [];

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

var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);

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

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

    generateColorCube();
    generateGround();
```


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
⊗ 0 ⚠ 0
Ln 110, Col 18 (9 selected)
Spaces: 4
UTF-8
CRLF
JavaScript

```javascript
    // 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;
        }
    };
    document.getElementById("down").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;
        }
    };
    document.getElementById("change").onclick = function () {
        if (document.getElementById("ortho").checked) {
            // 3D orthographic viewing
```


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
⊗ 0 ⚠ 0
Ln 110, Col 18 (9 selected)
Spaces: 4
UTF-8
CRLF
JavaScript

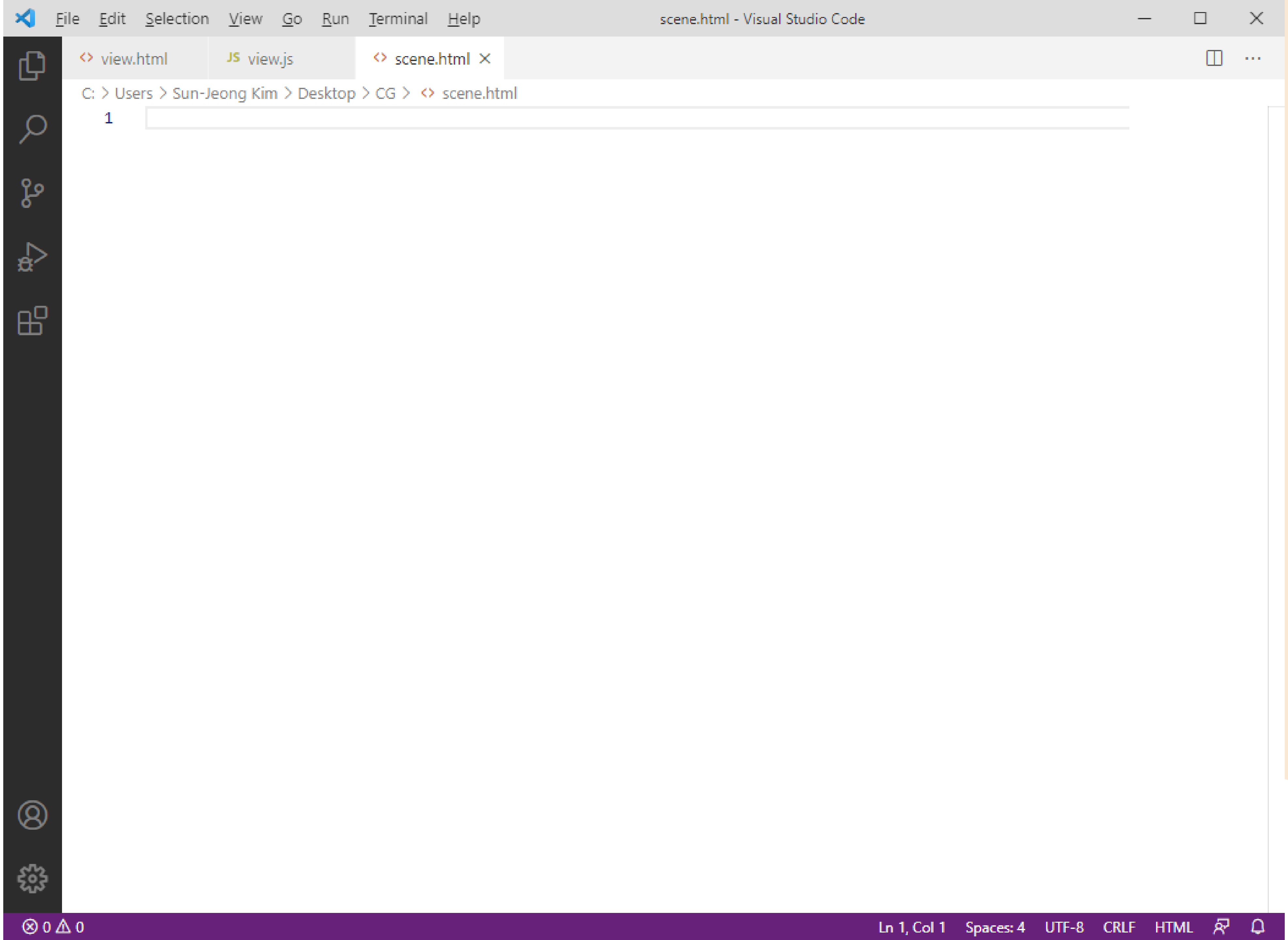
```javascript
                projectionMatrix = ortho(-viewLength, viewLength, -aspect, aspect, -viewLength, 1000);
            }
        }
        else {
            // 3D perspective viewing
            var aspect = canvas.width / canvas.height;
            projectionMatrix = perspective(90, aspect, 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() {
```

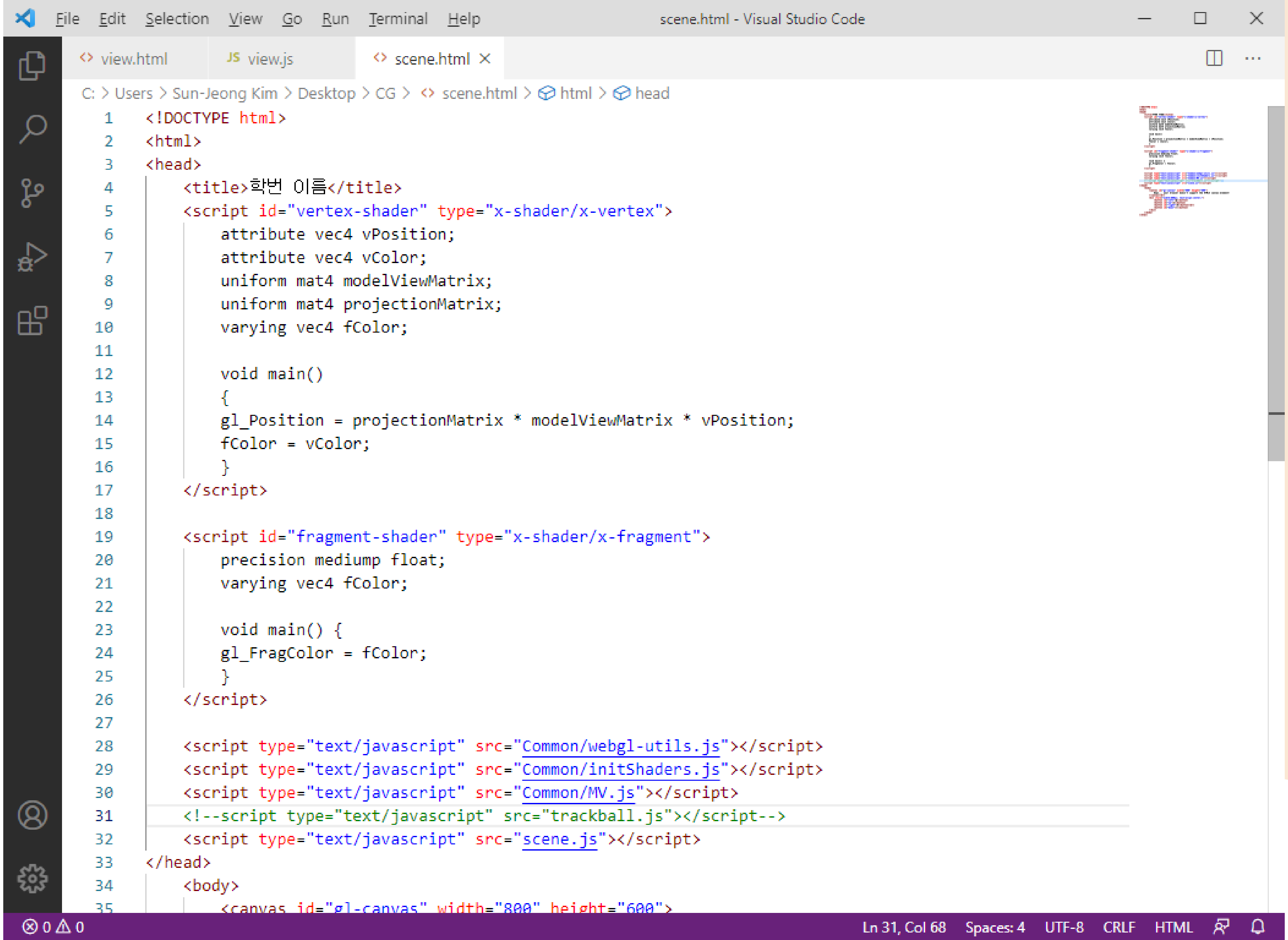


● Orthographic ○ Perspective Change

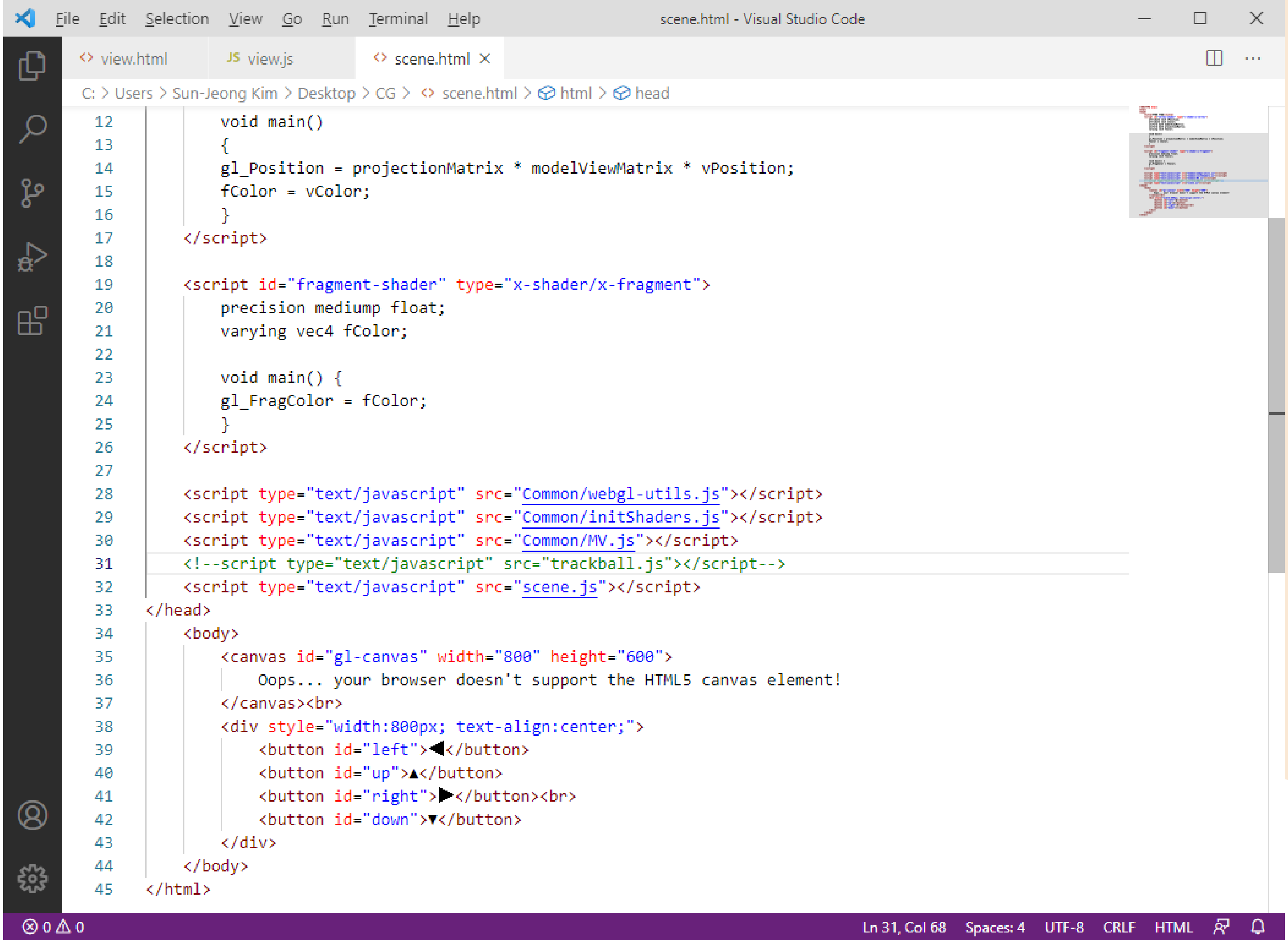
Building a Scene

- 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

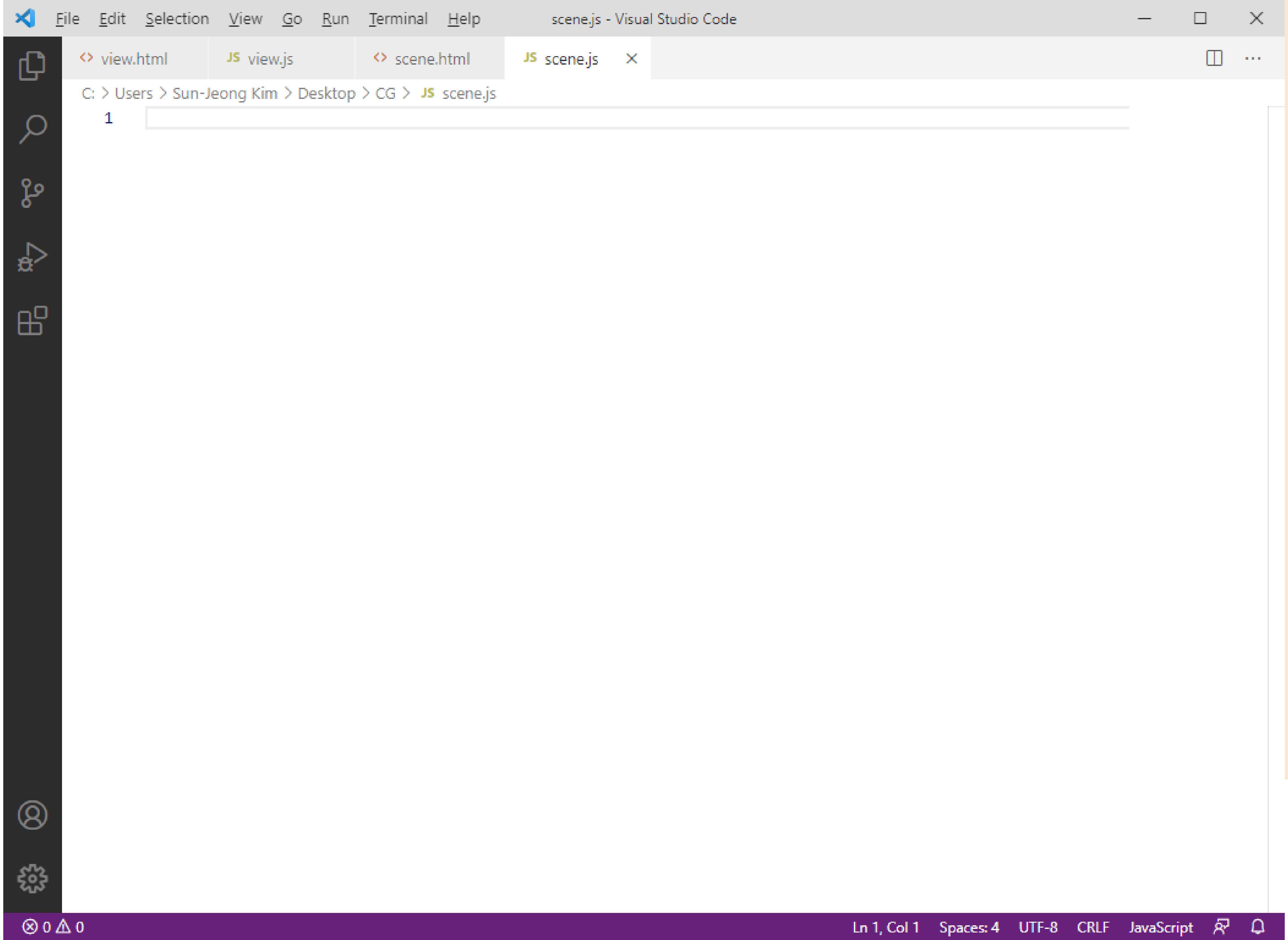

File
Edit
Selection
View
Go
Run
Terminal
Help
scene.html - Visual Studio Code
view.html
view.js
scene.html
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.html
1
⊗ 0 ⚠ 0
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
HTML

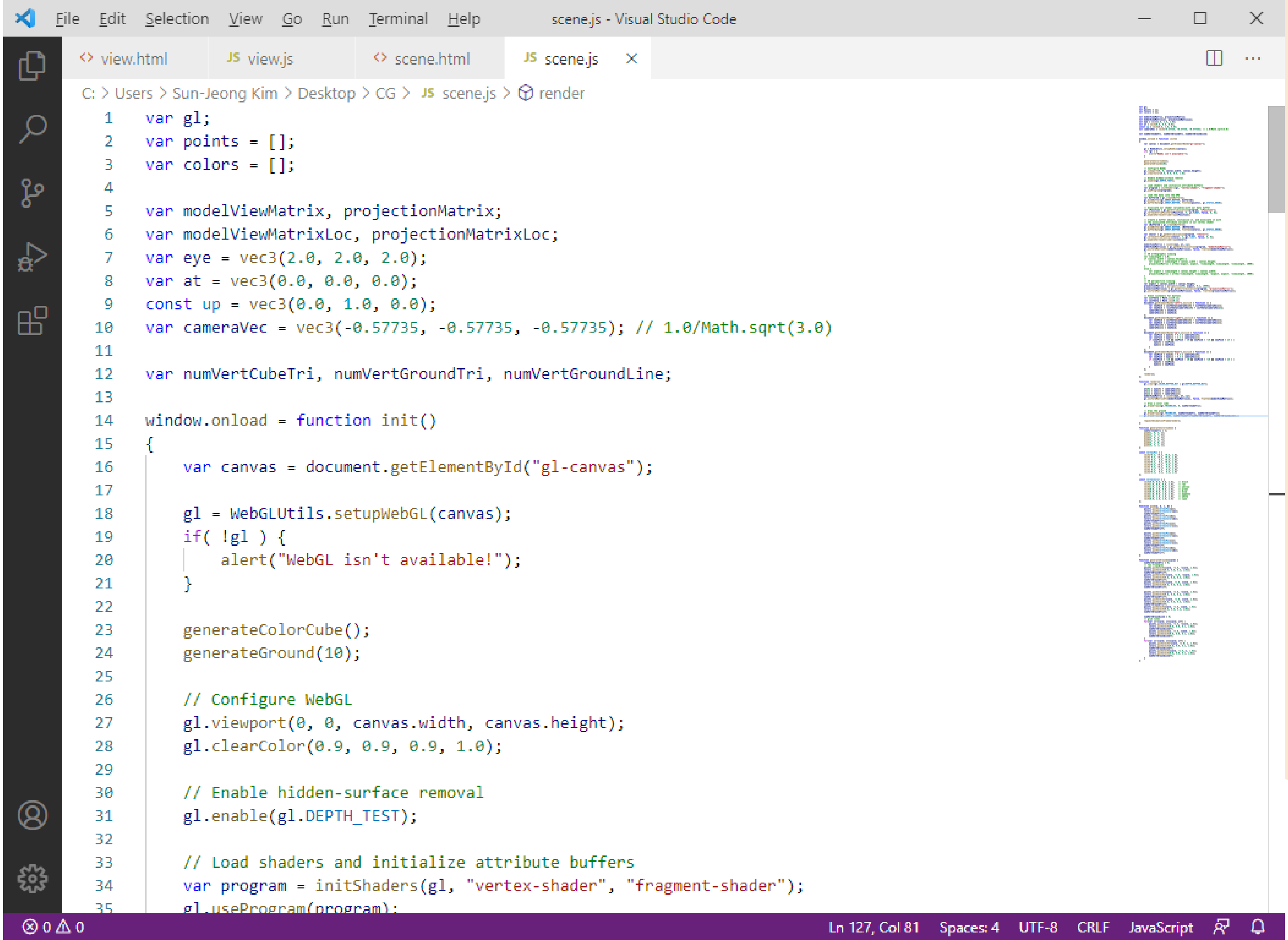


```html
<!DOCTYPE html>
<html>
<head>
    <title>학번 이름</title>
    <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;
        attribute vec4 vColor;
        uniform mat4 modelViewMatrix;
        uniform mat4 projectionMatrix;
        varying vec4 fColor;

        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">
```

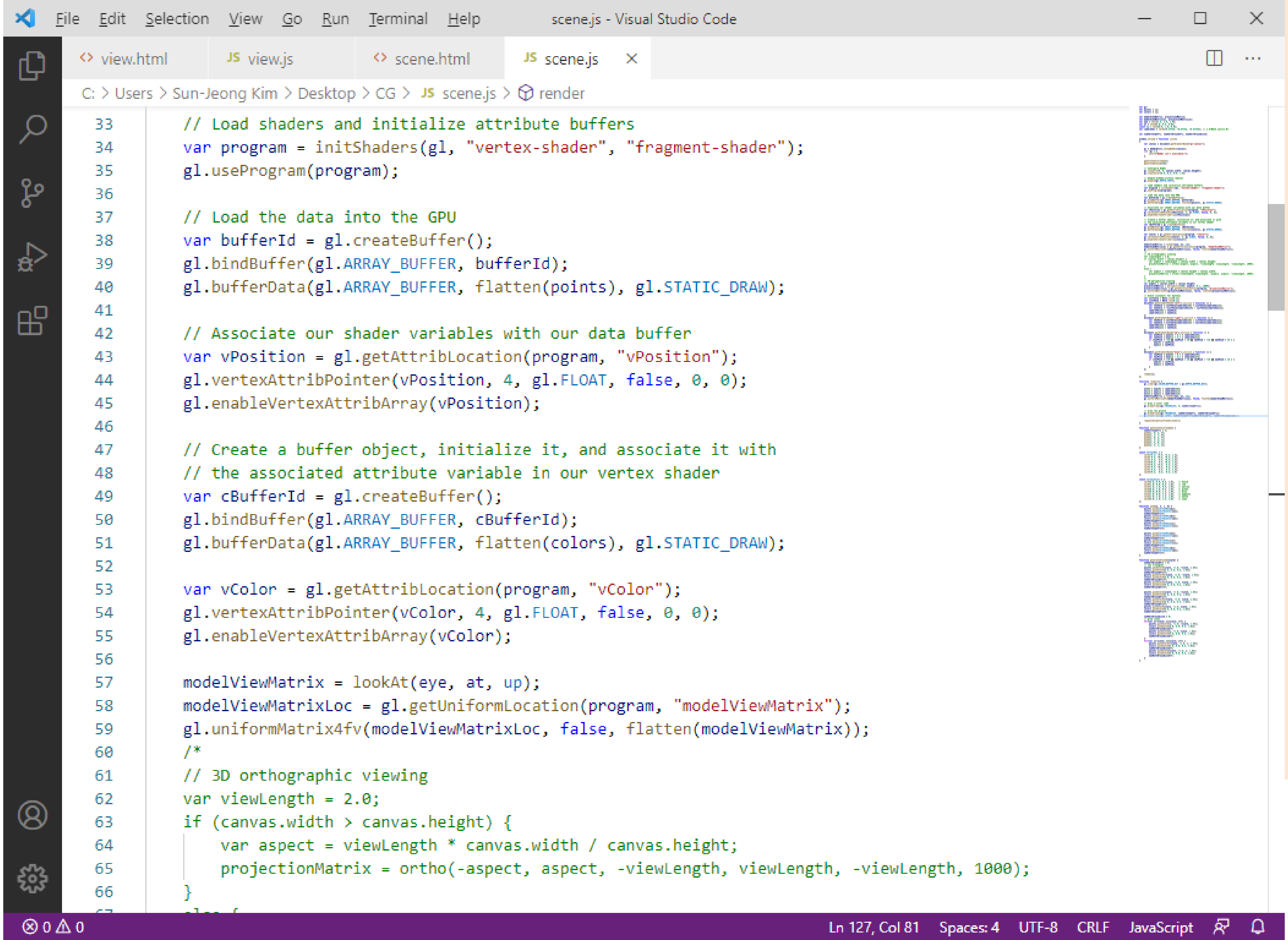


```
        void main()
        {
        gl_Position = projectionMatrix * modelViewMatrix * vPosition;
        fColor = vColor;
        }
    </script>

    <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><br>
        <div style="width:800px; text-align:center;">
            <button id="left">◀</button>
            <button id="up">▲</button>
            <button id="right">▶</button><br>
            <button id="down">▼</button>
        </div>
    </body>
</html>
```

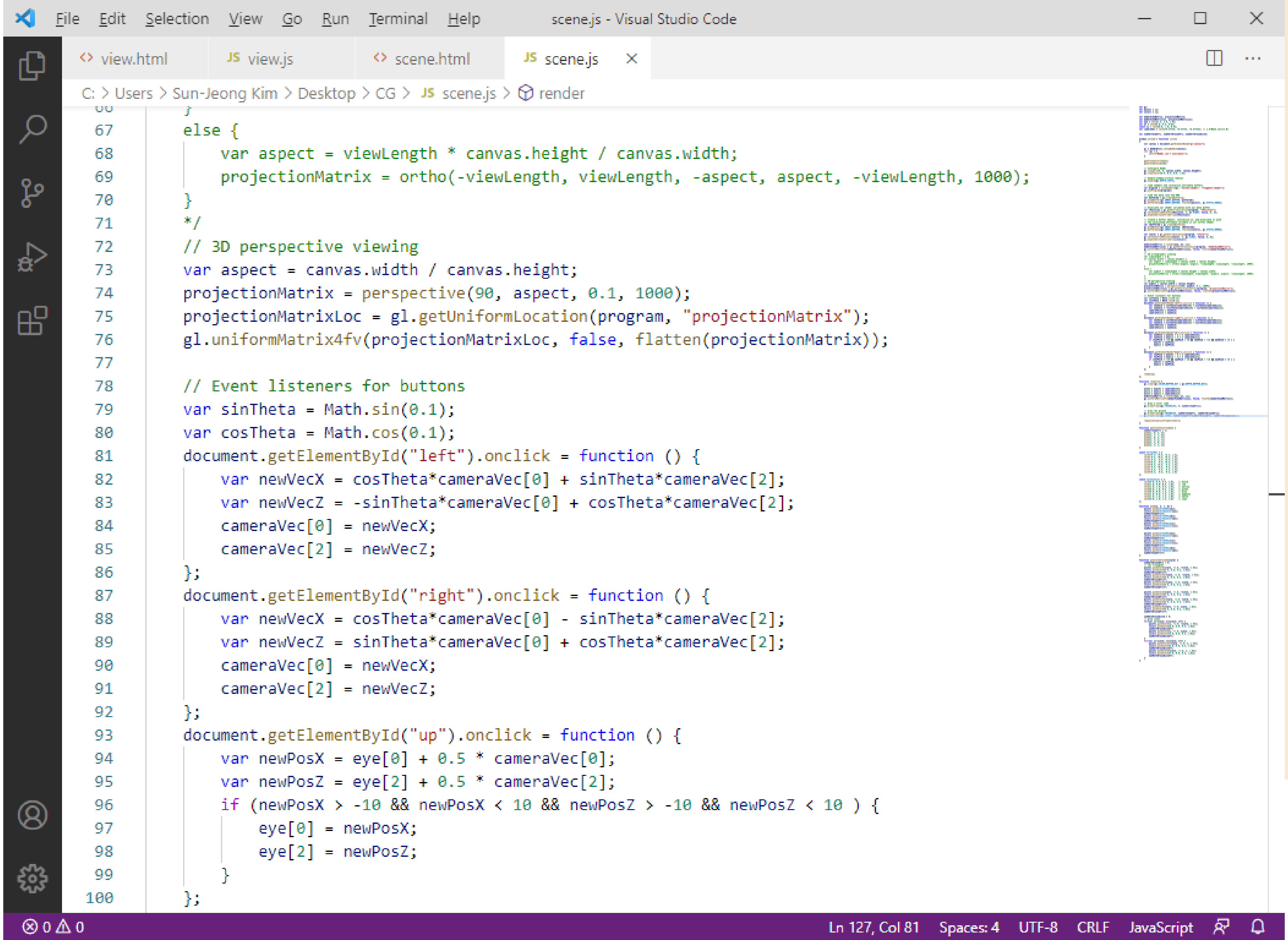

File
Edit
Selection
View
Go
Run
Terminal
Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js
⊗ 0 ⚠ 0
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript


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

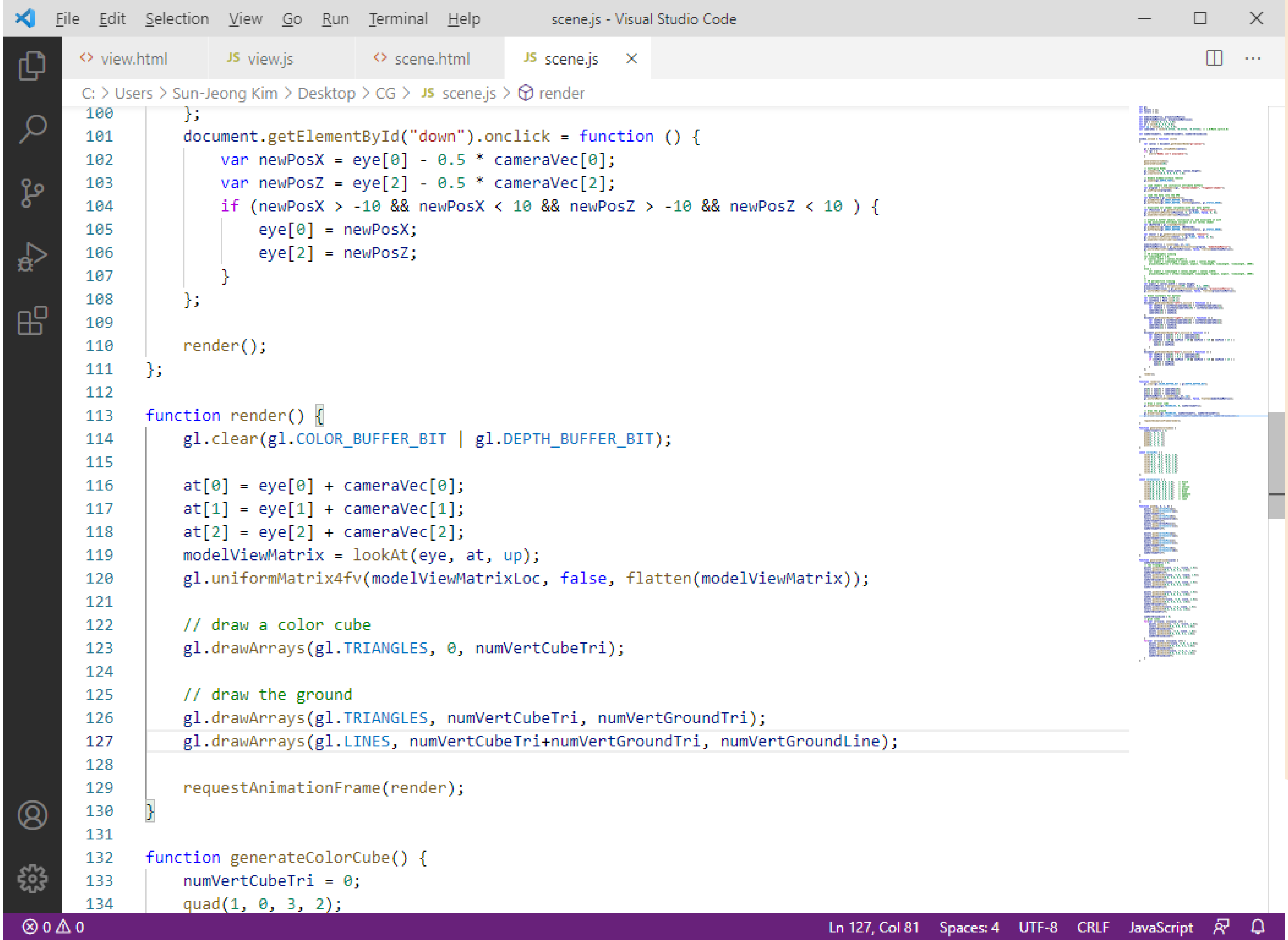
```javascript
var gl;
var points = [];
var colors = [];

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);
```



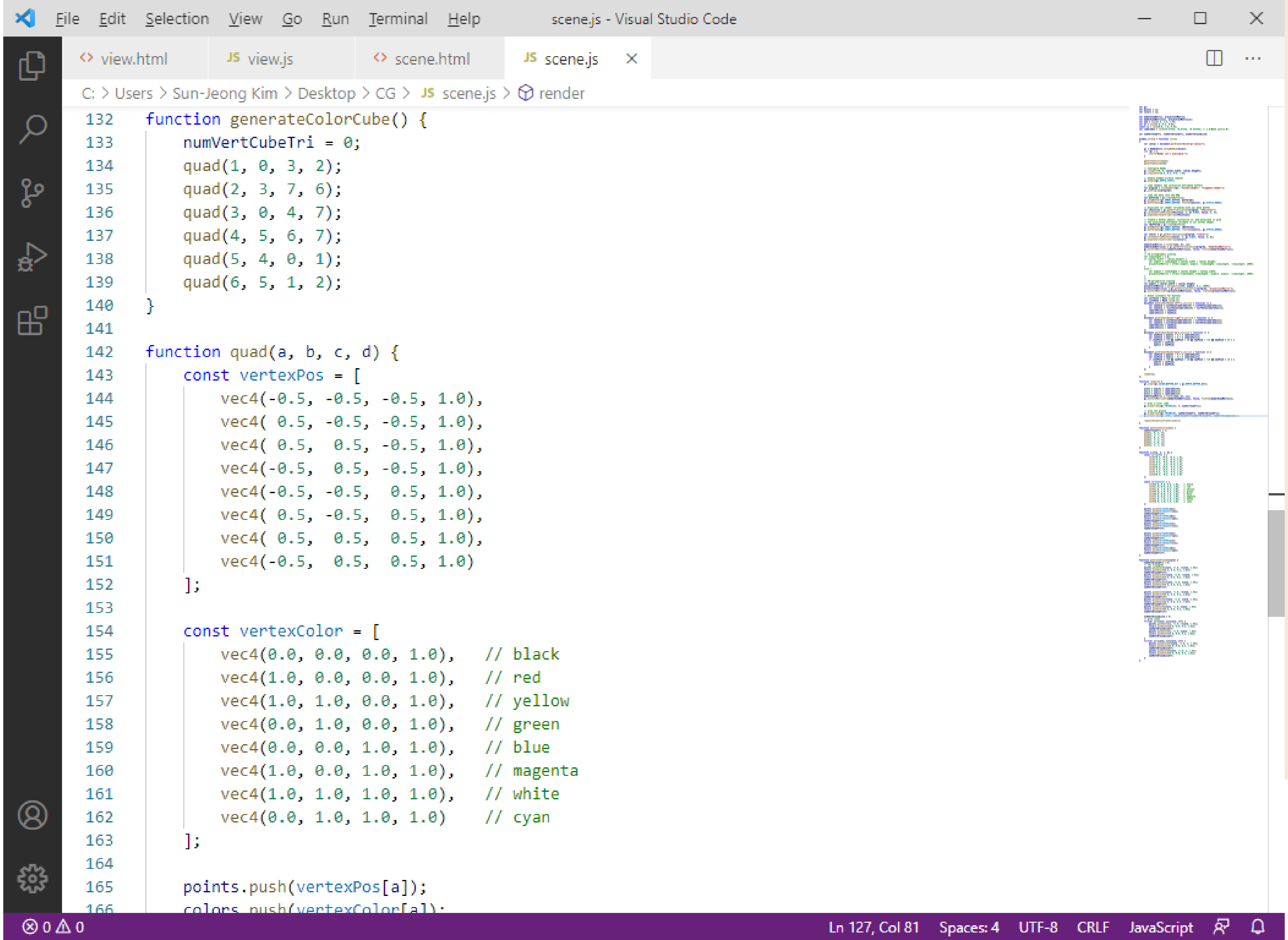
```javascript
    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    // Load the data into the GPU
    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);

    // Associate our shader variables with our data buffer
    var vPosition = gl.getAttribLocation(program, "vPosition");
    gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);

    // Create a buffer object, initialize it, and associate it with
    // the associated attribute variable in our vertex shader
    var cBufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, cBufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);

    var vColor = gl.getAttribLocation(program, "vColor");
    gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vColor);

    modelViewMatrix = lookAt(eye, at, up);
    modelViewMatrixLoc = gl.getUniformLocation(program, "modelViewMatrix");
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    /*
    // 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 {
```


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

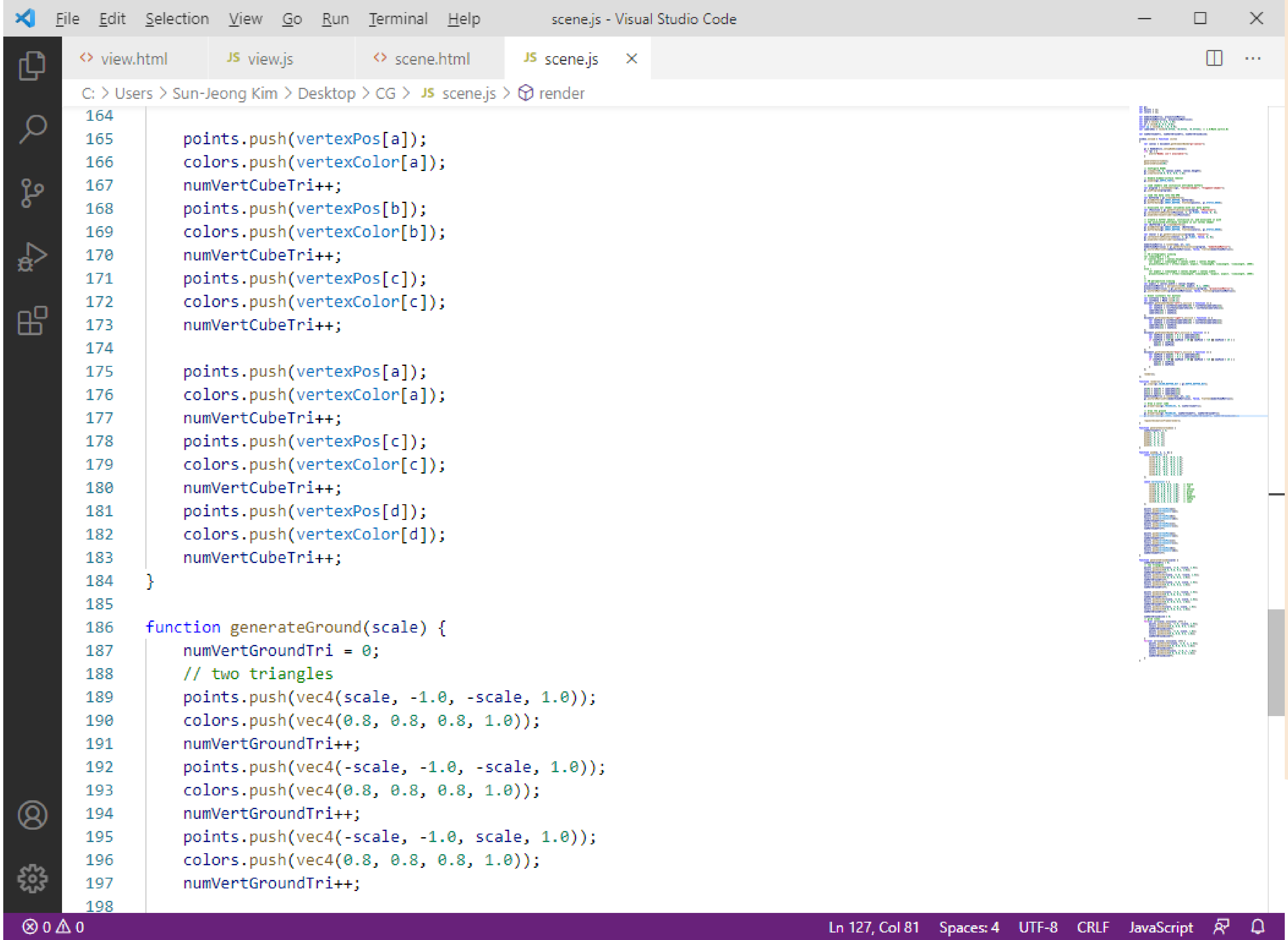
```
    }
    else {
        var aspect = viewLength * canvas.height / canvas.width;
        projectionMatrix = ortho(-viewLength, viewLength, -aspect, aspect, -viewLength, 1000);
    }
    */
    // 3D perspective viewing
    var aspect = canvas.width / canvas.height;
    projectionMatrix = perspective(90, 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;
        }
    };
```



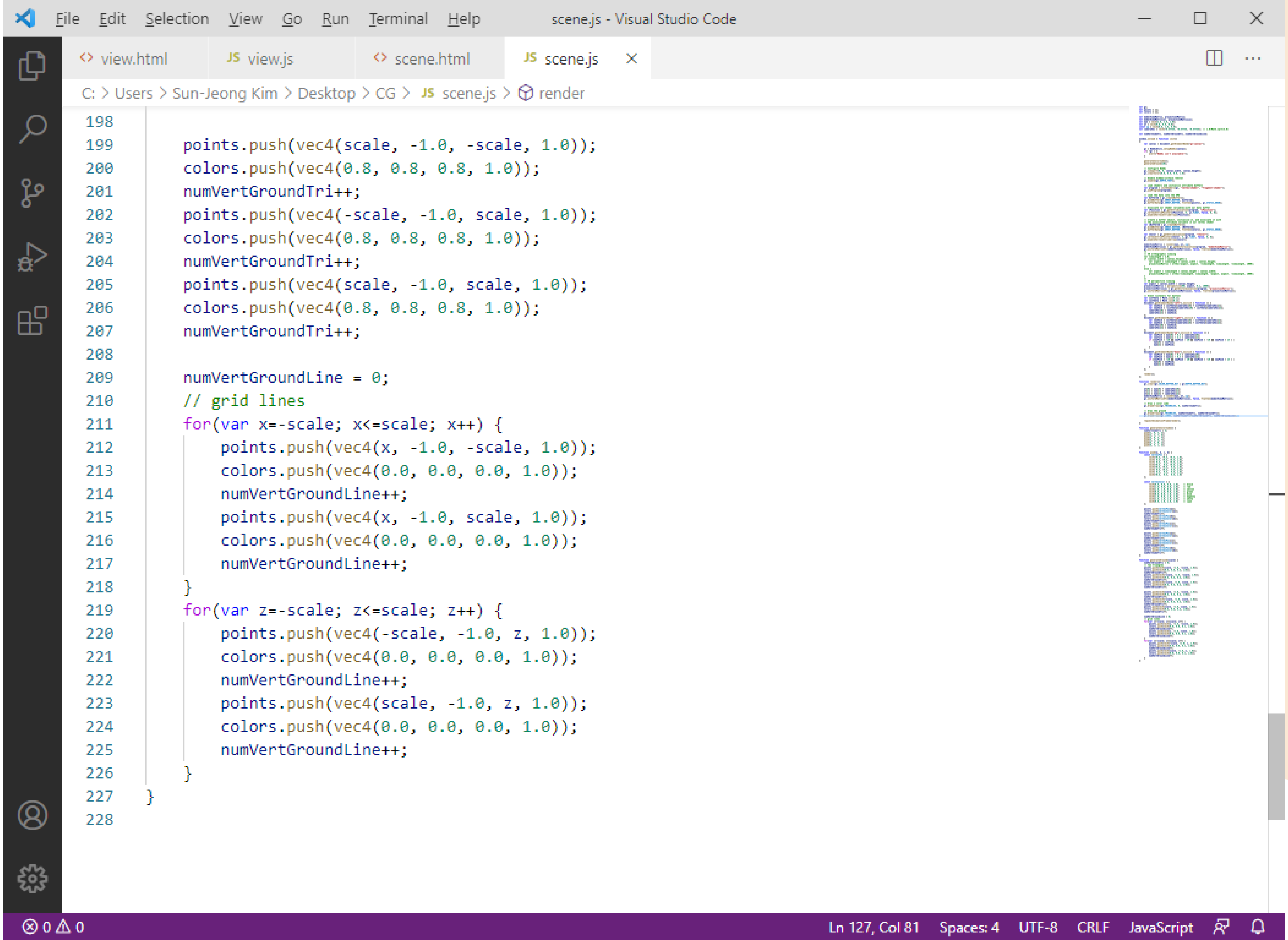
```javascript
    };
    document.getElementById("down").onclick = function () {
        var newPosX = eye[0] - 0.5 * cameraVec[0];
        var newPosZ = eye[2] - 0.5 * cameraVec[2];
        if (newPosX > -10 && newPosX < 10 && newPosZ > -10 && newPosZ < 10 ) {
            eye[0] = newPosX;
            eye[2] = newPosZ;
        }
    };

    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);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));

    // draw a color cube
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    // draw the ground
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertGroundTri);
    gl.drawArrays(gl.LINES, numVertCubeTri+numVertGroundTri, numVertGroundLine);

    requestAnimationFrame(render);
}

function generateColorCube() {
    numVertCubeTri = 0;
    quad(1, 0, 3, 2);
```


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

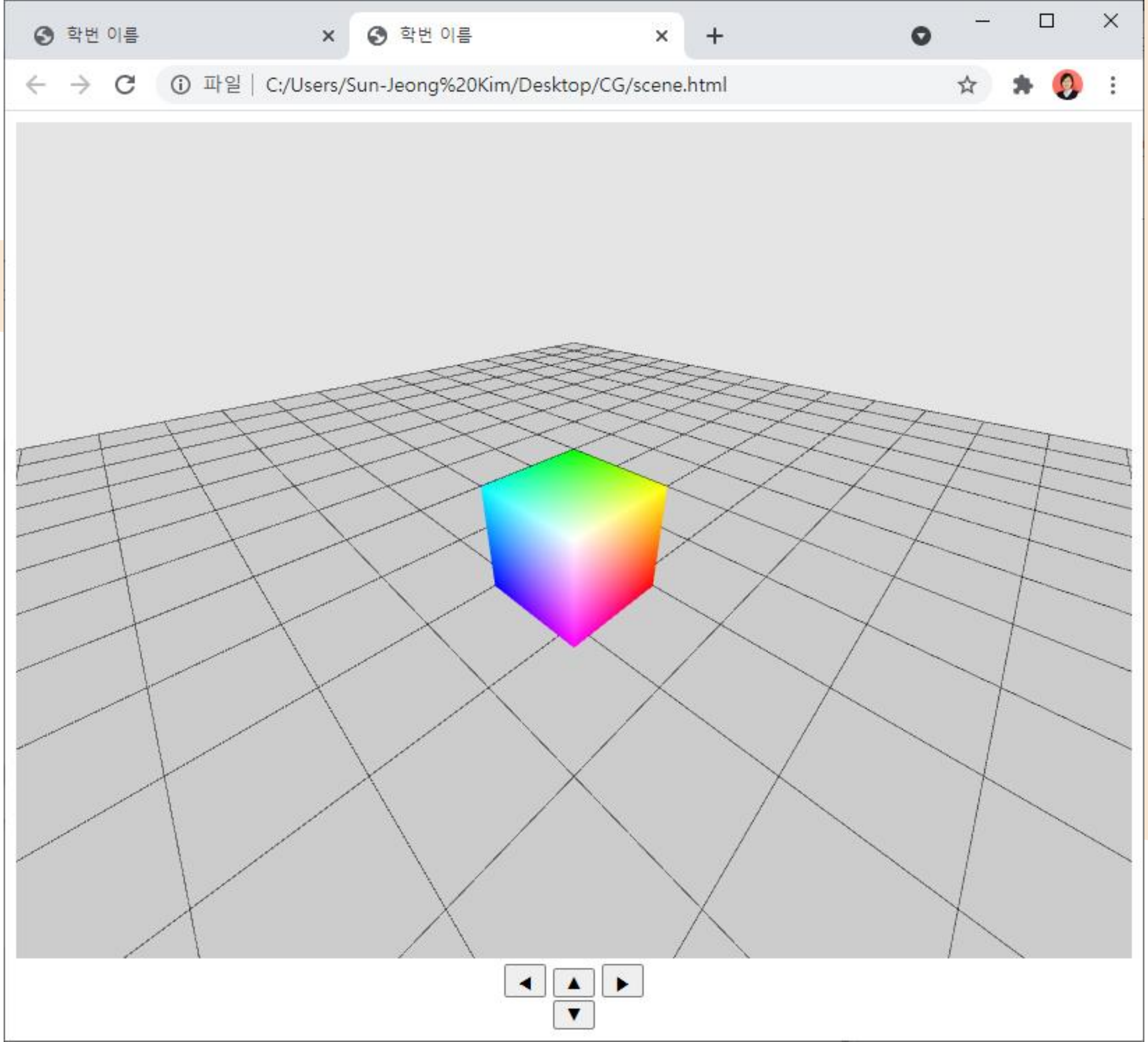
```
function generateColorCube() {
    numVertCubeTri = 0;
    quad(1, 0, 3, 2);
    quad(2, 3, 7, 6);
    quad(3, 0, 4, 7);
    quad(4, 5, 6, 7);
    quad(5, 4, 0, 1);
    quad(6, 5, 1, 2);
}

function quad(a, b, c, d) {
    const vertexPos = [
        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),
        vec4( 0.5,  0.5,  0.5, 1.0),
        vec4(-0.5,  0.5,  0.5, 1.0)
    ];

    const vertexColor = [
        vec4(0.0, 0.0, 0.0, 1.0),   // black
        vec4(1.0, 0.0, 0.0, 1.0),   // red
        vec4(1.0, 1.0, 0.0, 1.0),   // yellow
        vec4(0.0, 1.0, 0.0, 1.0),   // green
        vec4(0.0, 0.0, 1.0, 1.0),   // blue
        vec4(1.0, 0.0, 1.0, 1.0),   // magenta
        vec4(1.0, 1.0, 1.0, 1.0),   // white
        vec4(0.0, 1.0, 1.0, 1.0)    // cyan
    ];

    points.push(vertexPos[a]);
    colors.push(vertexColor[a]);
```

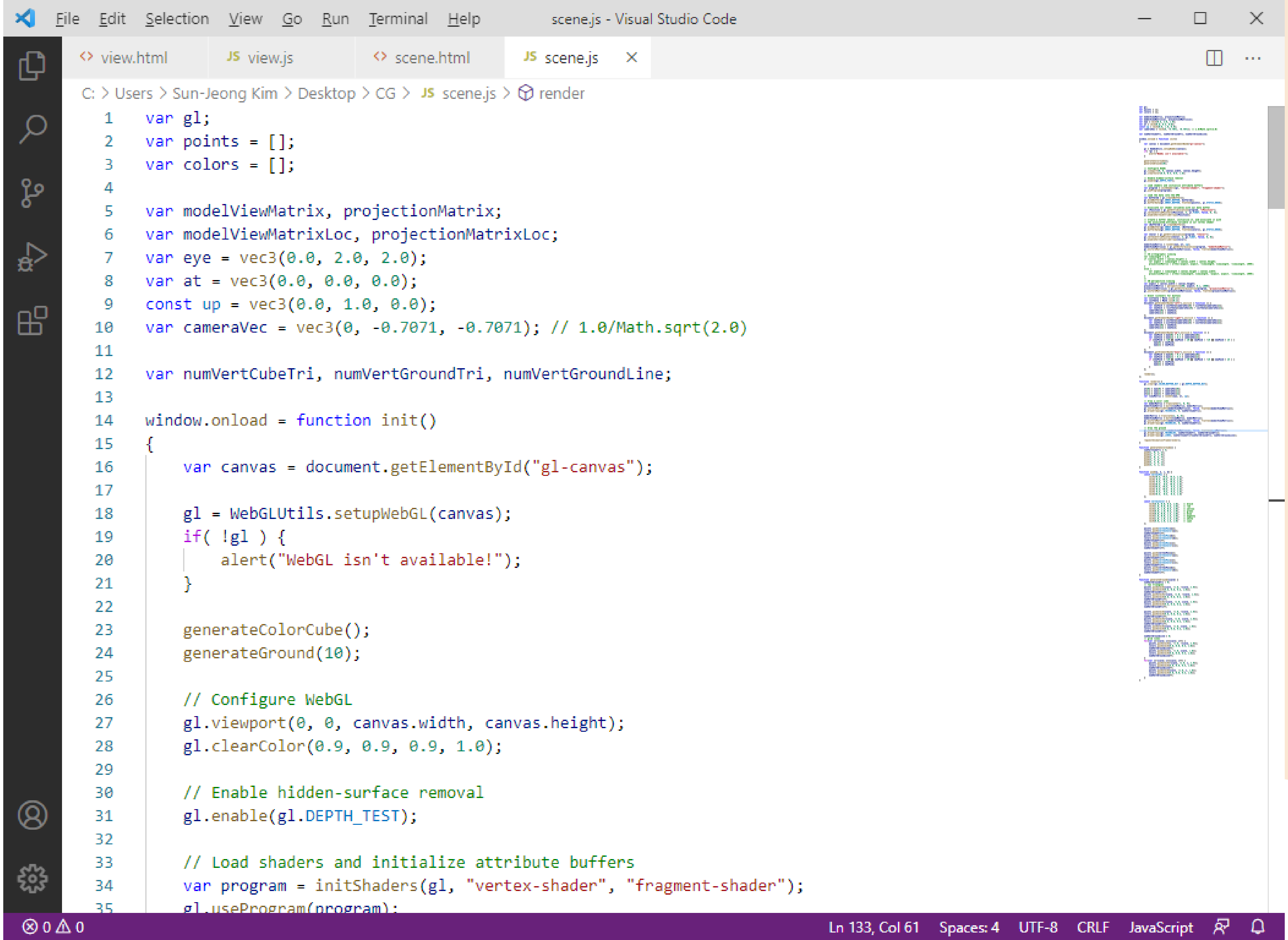


```javascript

    points.push(vertexPos[a]);
    colors.push(vertexColor[a]);
    numVertCubeTri++;
    points.push(vertexPos[b]);
    colors.push(vertexColor[b]);
    numVertCubeTri++;
    points.push(vertexPos[c]);
    colors.push(vertexColor[c]);
    numVertCubeTri++;

    points.push(vertexPos[a]);
    colors.push(vertexColor[a]);
    numVertCubeTri++;
    points.push(vertexPos[c]);
    colors.push(vertexColor[c]);
    numVertCubeTri++;
    points.push(vertexPos[d]);
    colors.push(vertexColor[d]);
    numVertCubeTri++;
}

function generateGround(scale) {
    numVertGroundTri = 0;
    // two triangles
    points.push(vec4(scale, -1.0, -scale, 1.0));
    colors.push(vec4(0.8, 0.8, 0.8, 1.0));
    numVertGroundTri++;
    points.push(vec4(-scale, -1.0, -scale, 1.0));
    colors.push(vec4(0.8, 0.8, 0.8, 1.0));
    numVertGroundTri++;
    points.push(vec4(-scale, -1.0, scale, 1.0));
    colors.push(vec4(0.8, 0.8, 0.8, 1.0));
    numVertGroundTri++;

```

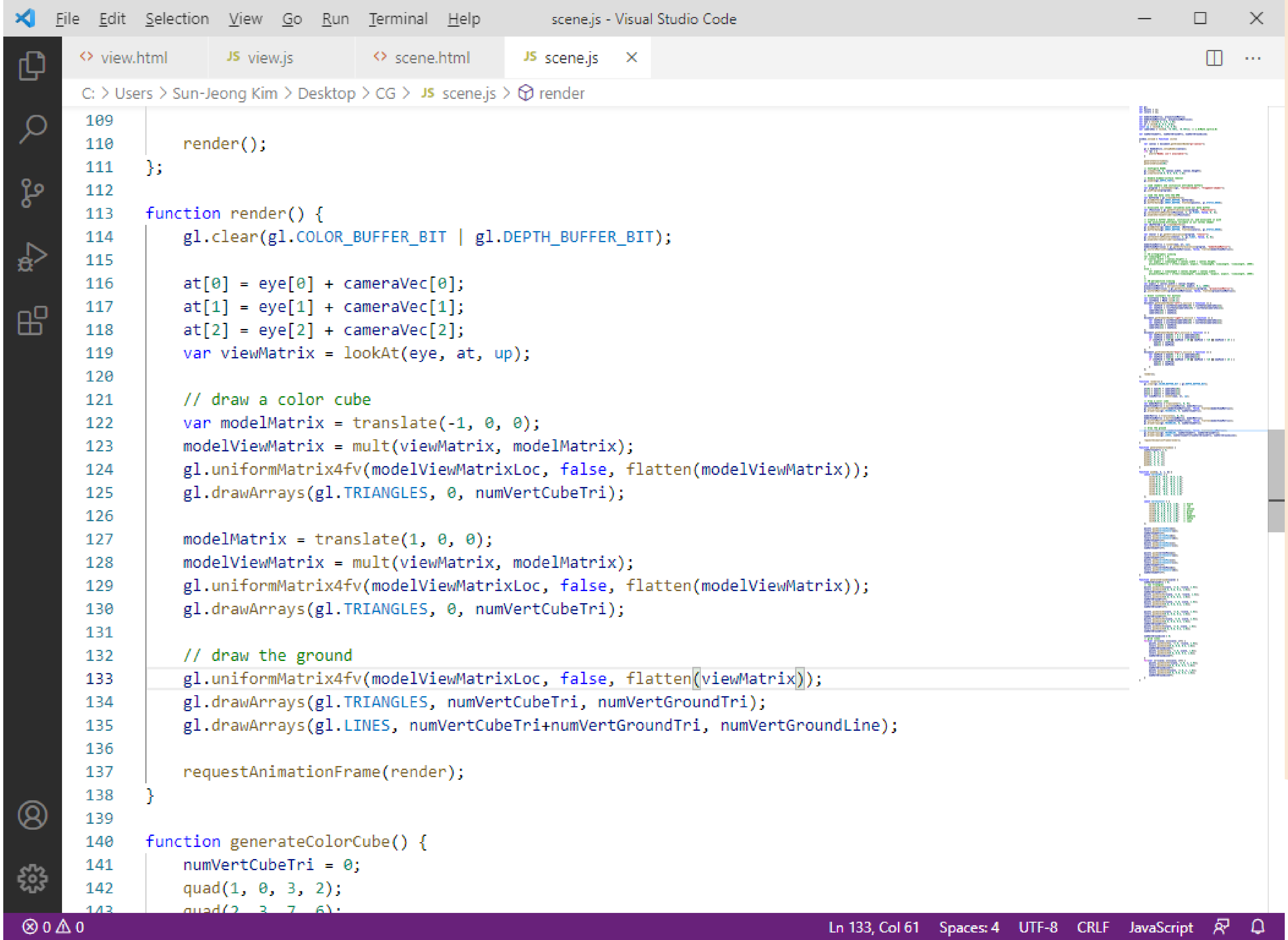

File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > render
Ln 127, Col 81
Spaces: 4
UTF-8
CRLF
JavaScript

```

    points.push(vec4(scale, -1.0, -scale, 1.0));
    colors.push(vec4(0.8, 0.8, 0.8, 1.0));
    numVertGroundTri++;
    points.push(vec4(-scale, -1.0, scale, 1.0));
    colors.push(vec4(0.8, 0.8, 0.8, 1.0));
    numVertGroundTri++;
    points.push(vec4(scale, -1.0, scale, 1.0));
    colors.push(vec4(0.8, 0.8, 0.8, 1.0));
    numVertGroundTri++;

    numVertGroundLine = 0;
    // grid lines
    for(var x=-scale; x<=scale; x++) {
        points.push(vec4(x, -1.0, -scale, 1.0));
        colors.push(vec4(0.0, 0.0, 0.0, 1.0));
        numVertGroundLine++;
        points.push(vec4(x, -1.0, scale, 1.0));
        colors.push(vec4(0.0, 0.0, 0.0, 1.0));
        numVertGroundLine++;
    }
    for(var z=-scale; z<=scale; z++) {
        points.push(vec4(-scale, -1.0, z, 1.0));
        colors.push(vec4(0.0, 0.0, 0.0, 1.0));
        numVertGroundLine++;
        points.push(vec4(scale, -1.0, z, 1.0));
        colors.push(vec4(0.0, 0.0, 0.0, 1.0));
        numVertGroundLine++;
    }
}
```

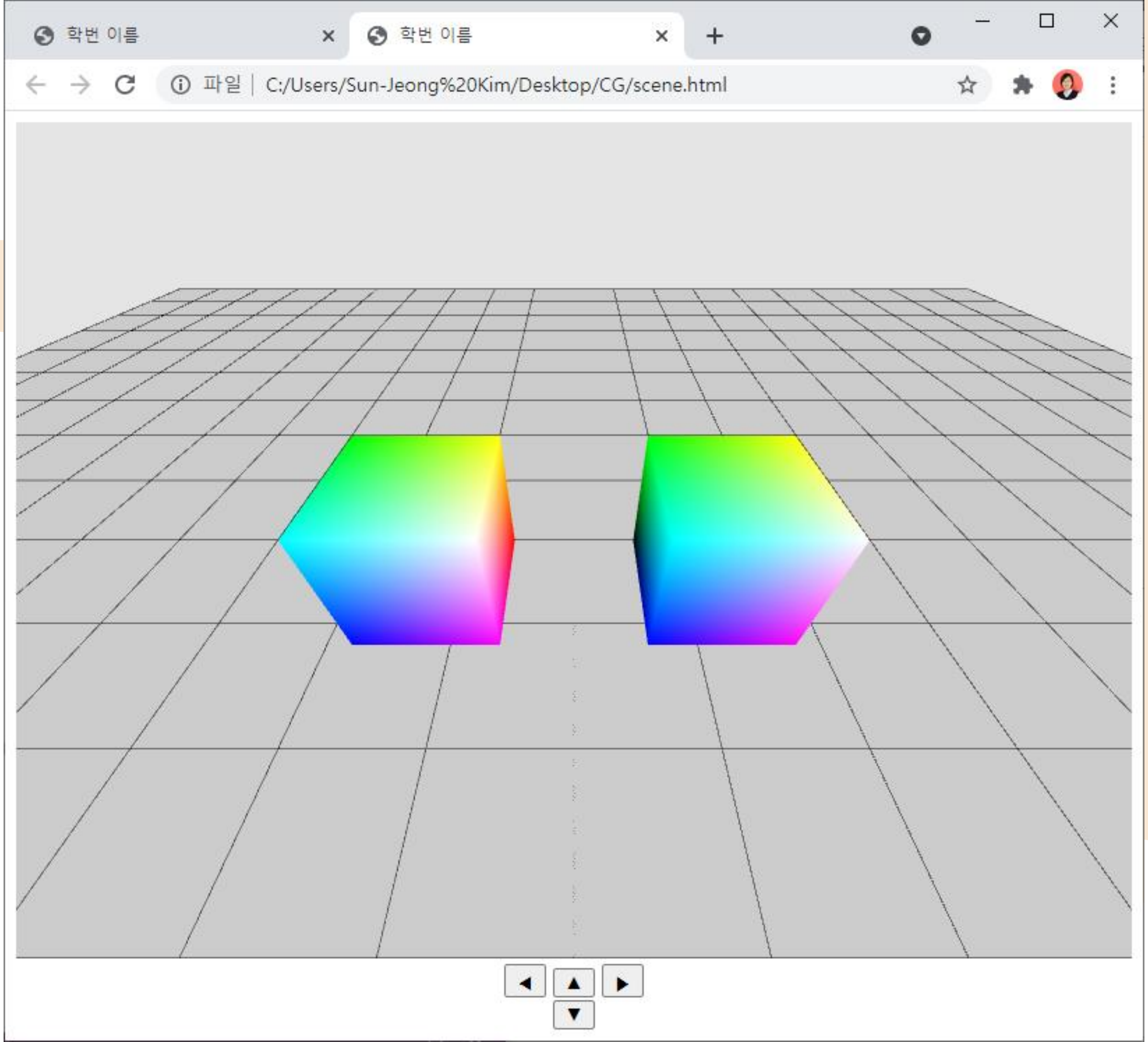

학번 이름
학번 이름
파일
C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene.html
◀
▲
▶
▼

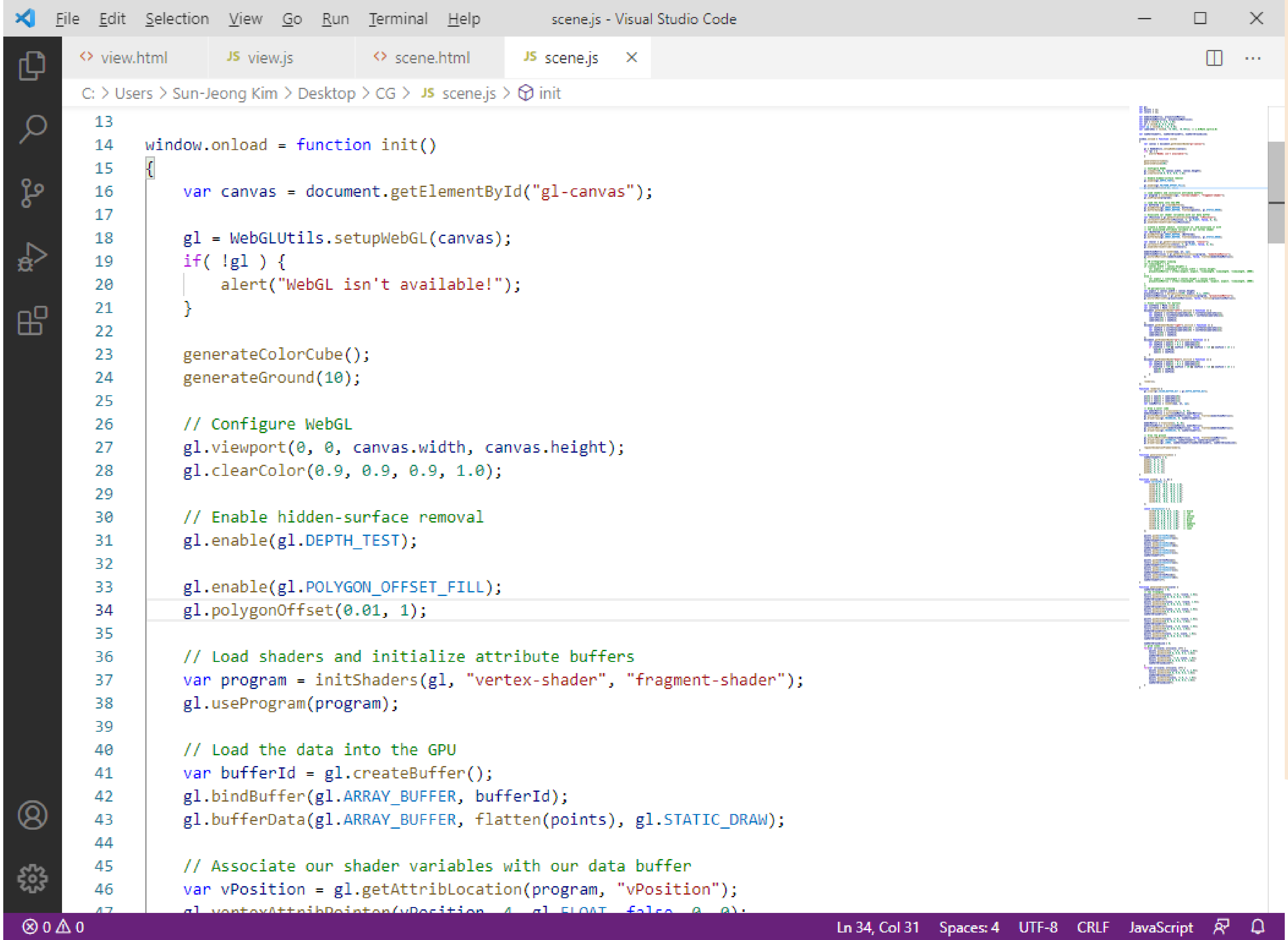


```javascript
var gl;
var points = [];
var colors = [];

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);
```

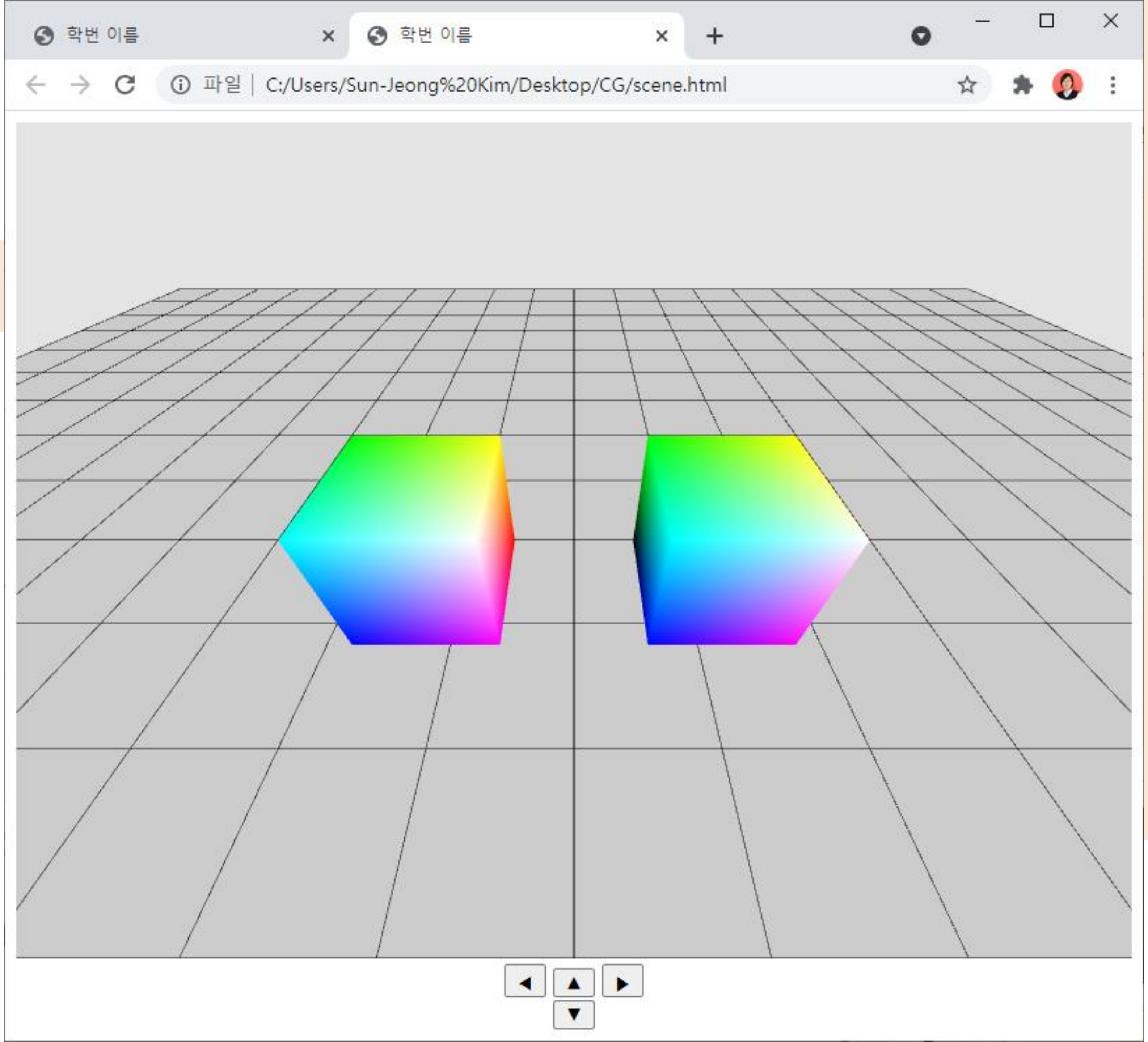

File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > render
⊗ 0 ⚠ 0
Ln 133, Col 61
Spaces: 4
UTF-8
CRLF
JavaScript

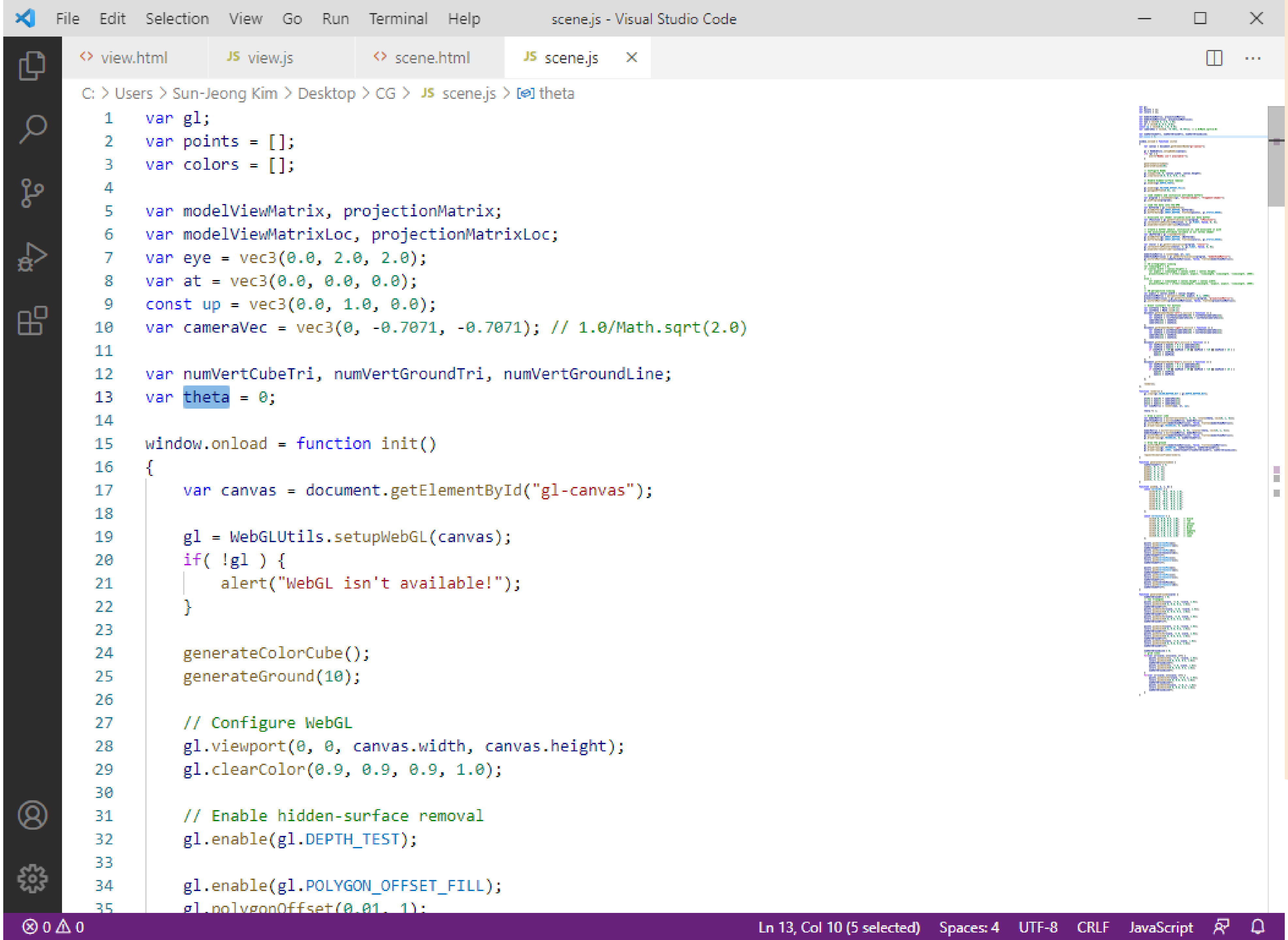
```

    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];
    var viewMatrix = lookAt(eye, at, up);

    // draw a color cube
    var modelMatrix = translate(-1, 0, 0);
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    modelMatrix = translate(1, 0, 0);
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    // draw the ground
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertGroundTri);
    gl.drawArrays(gl.LINES, numVertCubeTri+numVertGroundTri, numVertGroundLine);

    requestAnimationFrame(render);
}

function generateColorCube() {
    numVertCubeTri = 0;
    quad(1, 0, 3, 2);
    quad(2, 3, 7, 6);
```


학번 이름
학번 이름
파일
C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene.html
◀
▲
▶
▼

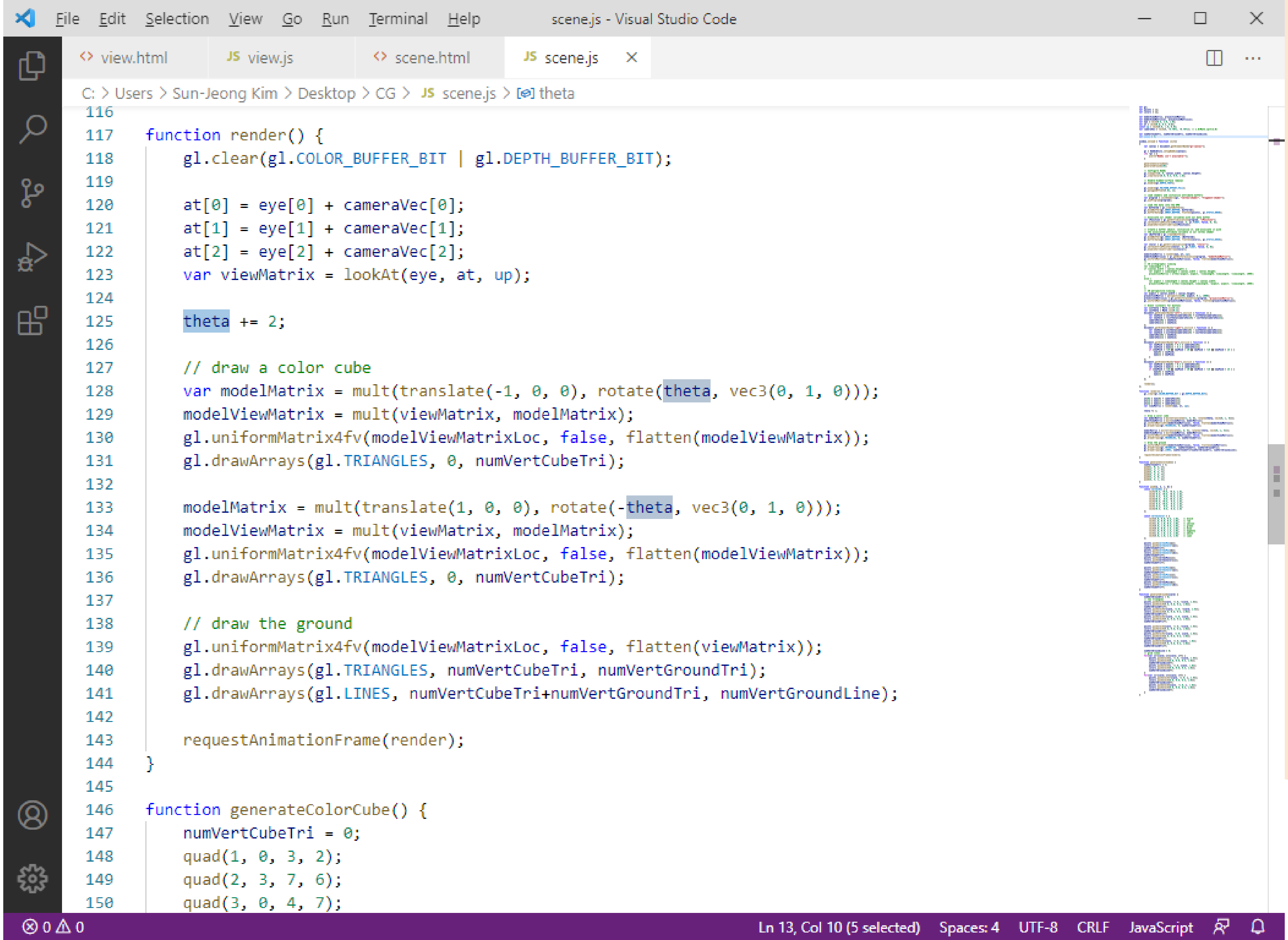

File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > init
⊗ 0 ⚠ 0
Ln 34, Col 31 Spaces: 4 UTF-8 CRLF JavaScript

```

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);

    gl.enable(gl.POLYGON_OFFSET_FILL);
    gl.polygonOffset(0.01, 1);

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

    // Load the data into the GPU
    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);

    // Associate our shader variables with our data buffer
    var vPosition = gl.getAttribLocation(program, "vPosition");
    gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
```


학번 이름
학번 이름
파일
C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene.html
◀
▲
▶
▼

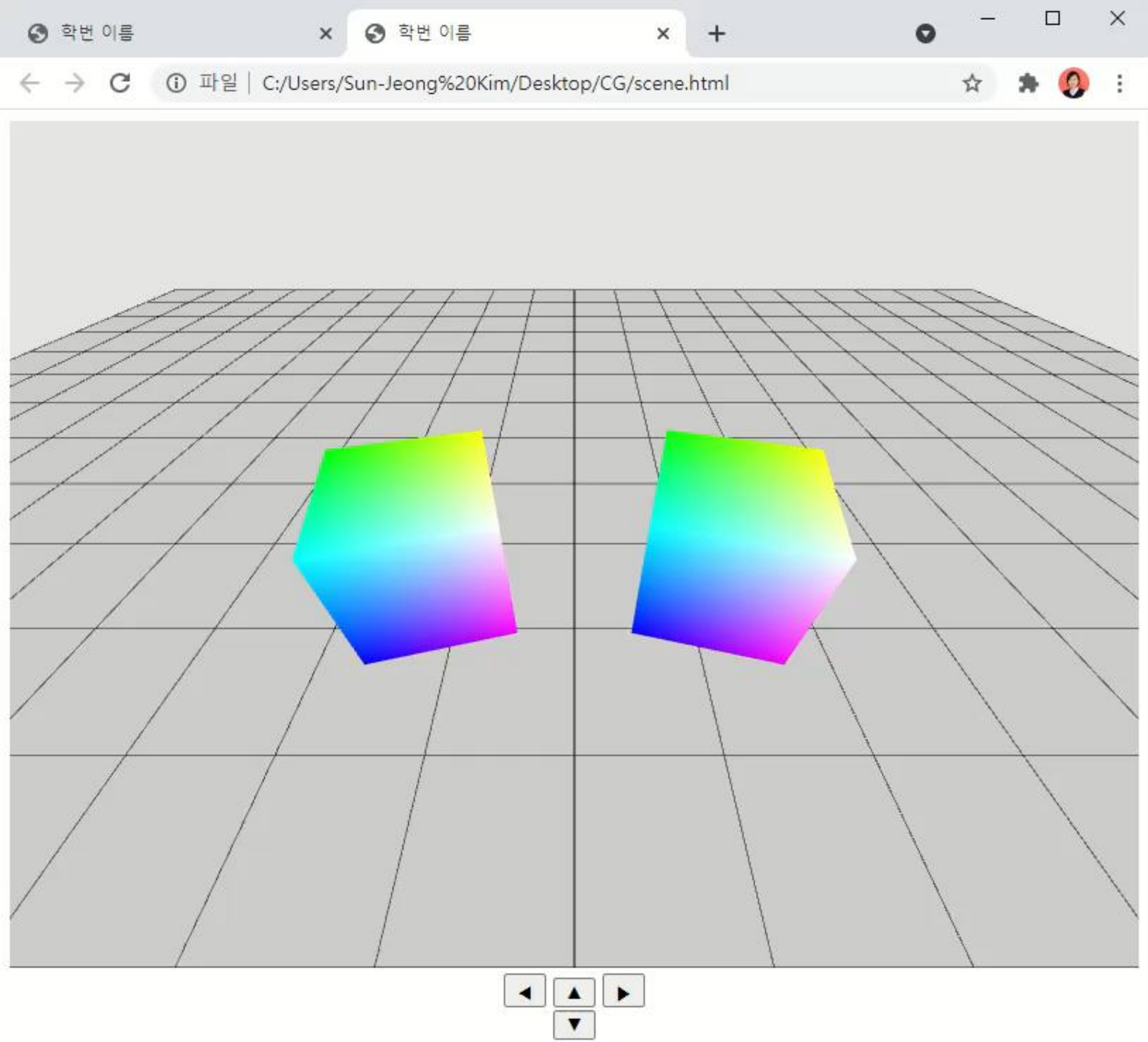

File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > theta
⊗ 0 ⚠ 0
Ln 13, Col 10 (5 selected)
Spaces: 4
UTF-8
CRLF
JavaScript

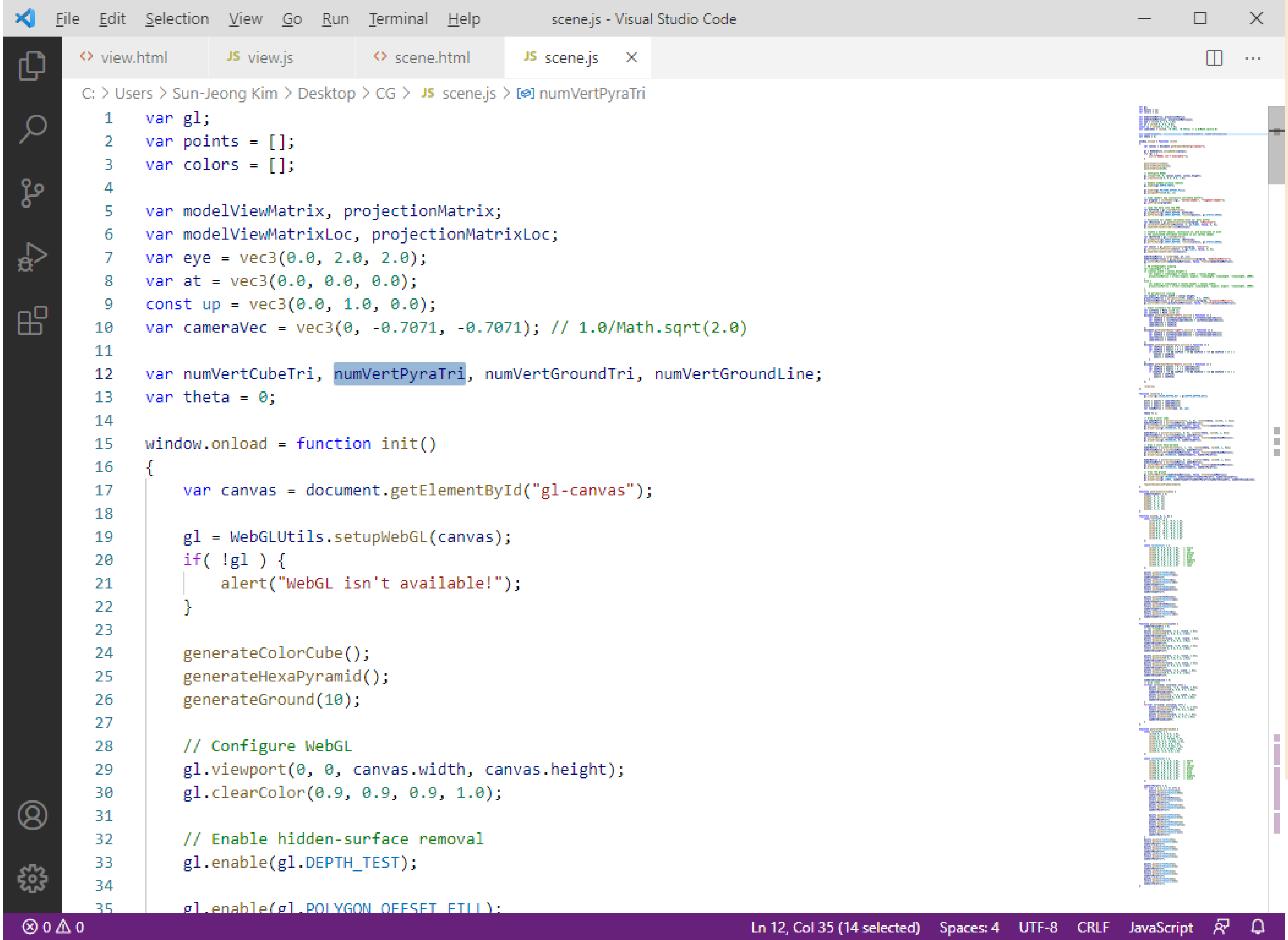
```
var gl;
var points = [];
var colors = [];

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;
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();
    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);
    gl.polygonOffset(0.01, 1);
```



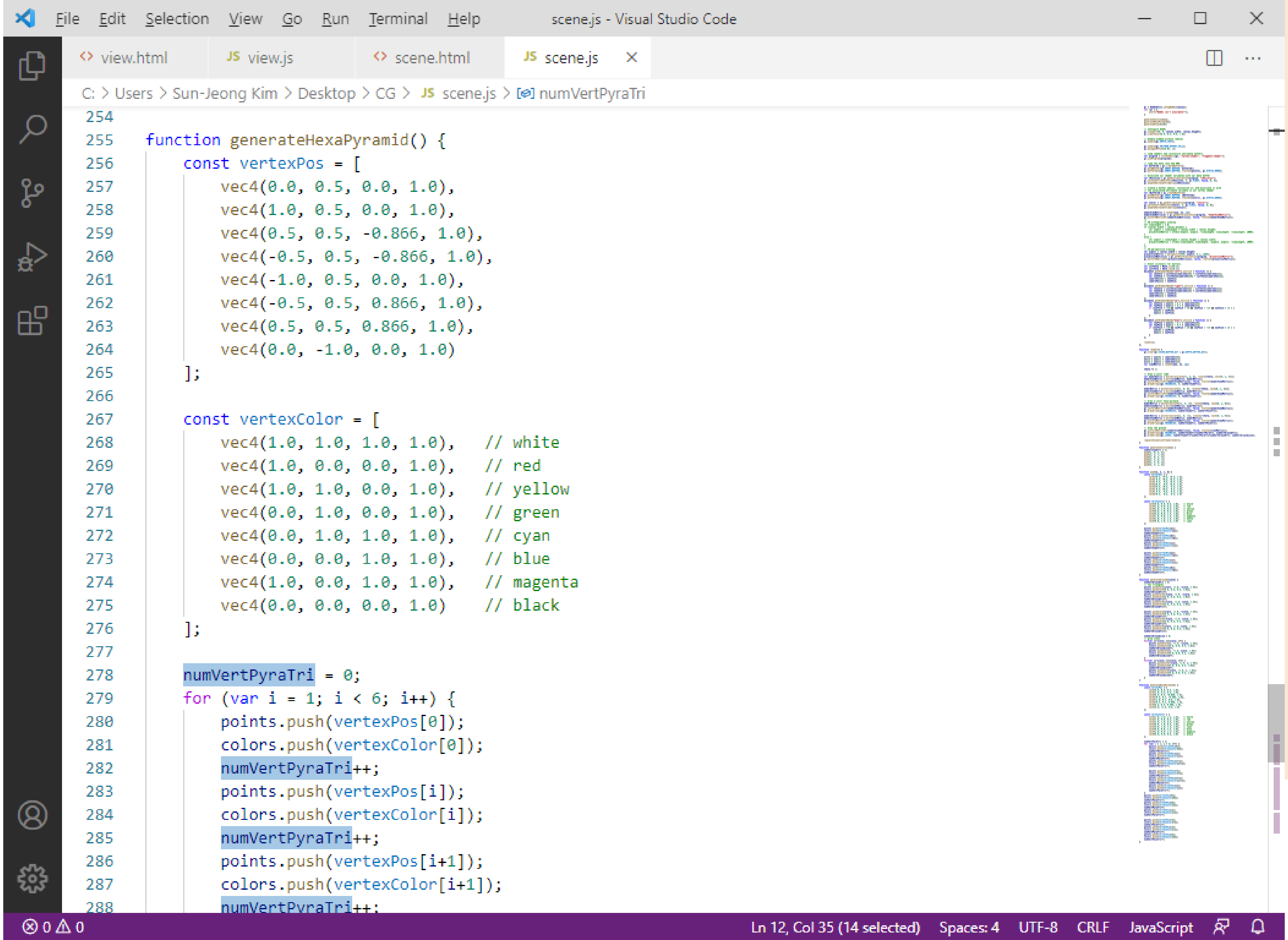
File Edit Selection View Go Run Terminal Help scene.js - Visual Studio Code
view.html view.js scene.html scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > theta
⊗ 0 ⚠ 0 Ln 13, Col 10 (5 selected) Spaces: 4 UTF-8 CRLF JavaScript

```javascript

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];
    var viewMatrix = lookAt(eye, at, up);

    theta += 2;

    // draw a color cube
    var modelMatrix = mult(translate(-1, 0, 0), rotate(theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    modelMatrix = mult(translate(1, 0, 0), rotate(-theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    // draw the ground
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertGroundTri);
    gl.drawArrays(gl.LINES, numVertCubeTri+numVertGroundTri, numVertGroundLine);

    requestAnimationFrame(render);
}

function generateColorCube() {
    numVertCubeTri = 0;
    quad(1, 0, 3, 2);
    quad(2, 3, 7, 6);
    quad(3, 0, 4, 7);
```

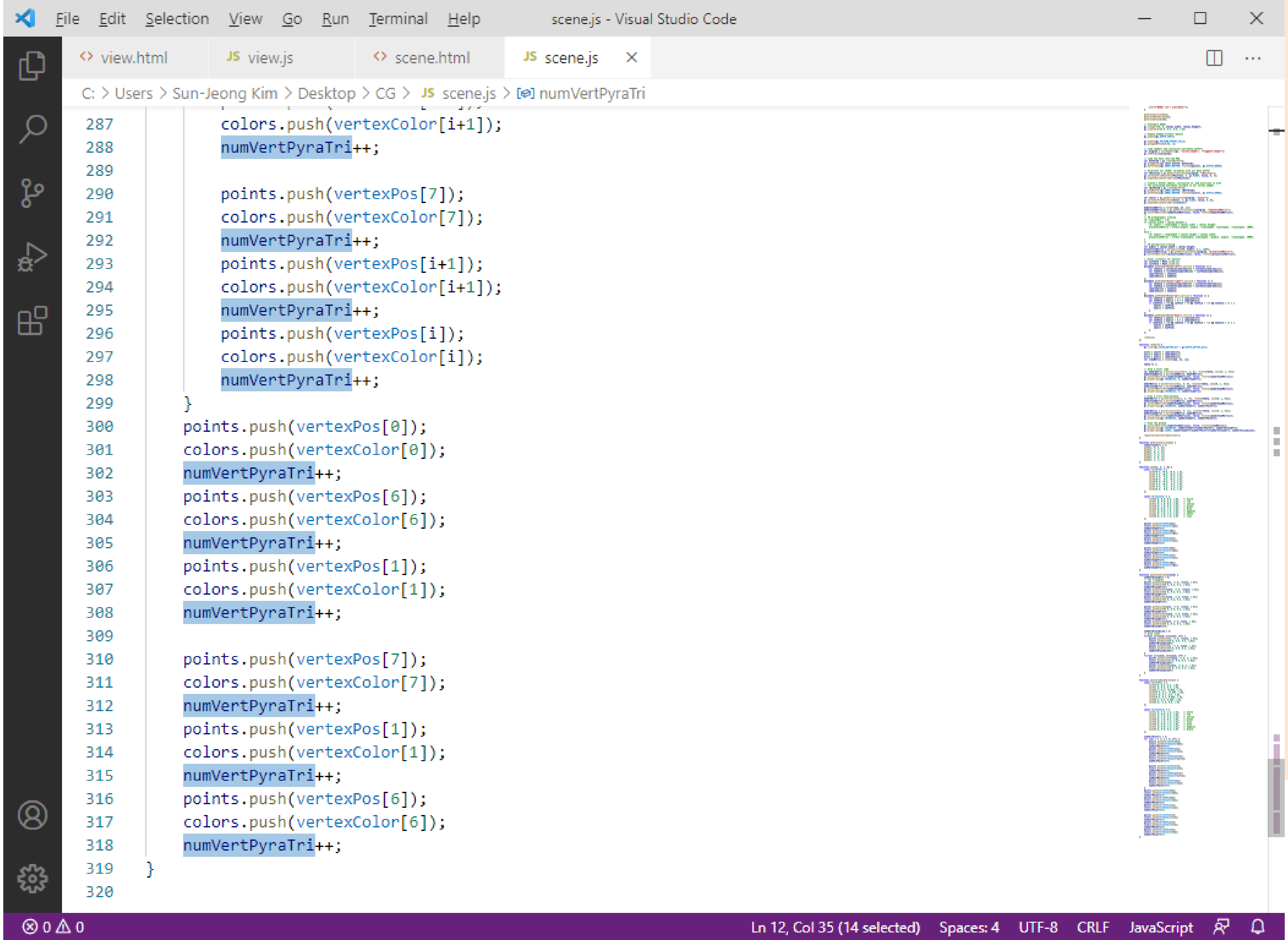

학번 이름
학번 이름
파일
C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene.html
◀
▲
▶
▼


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

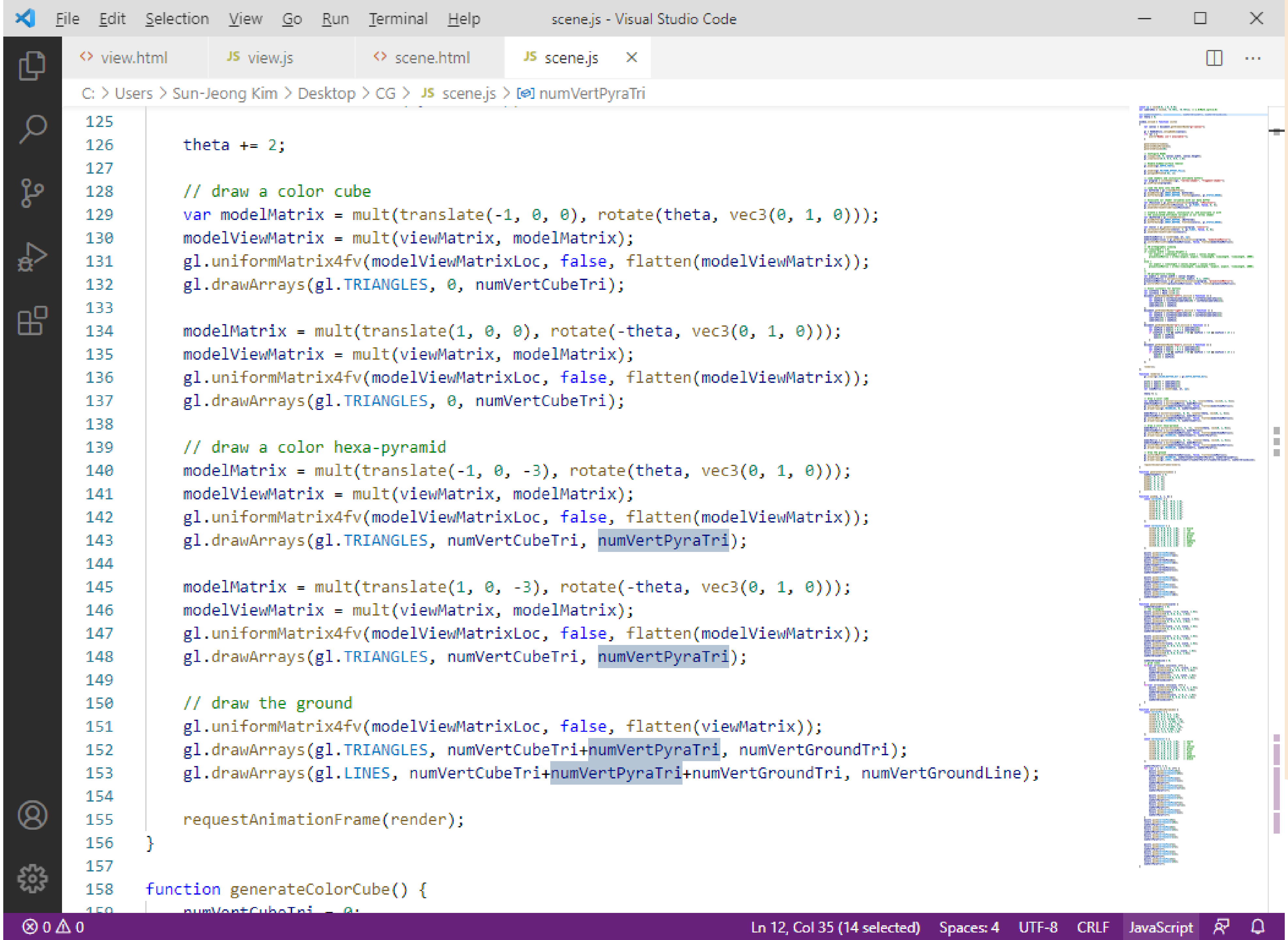
```javascript
var gl;
var points = [];
var colors = [];

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);
```


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

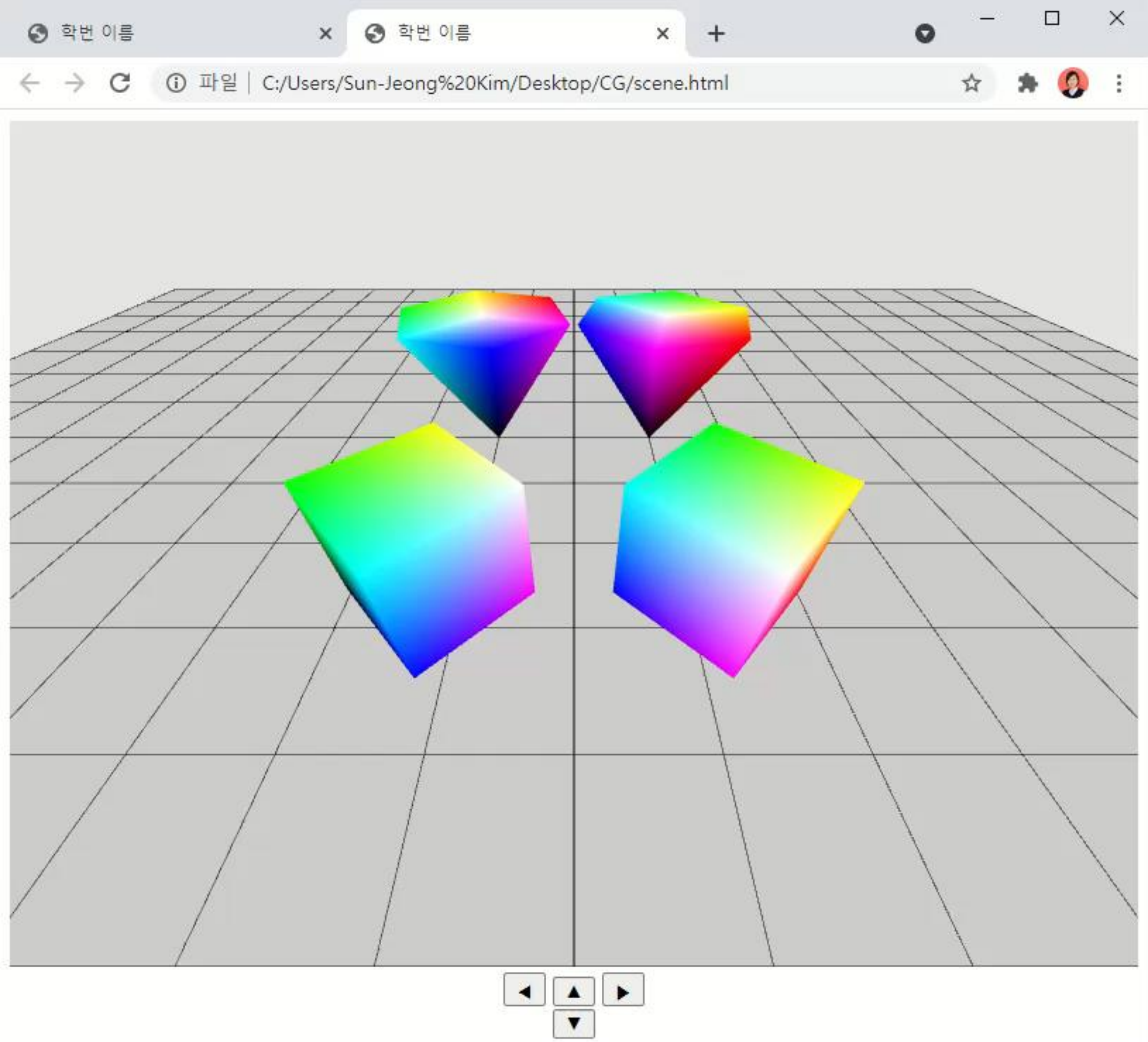
```javascript

function generateHexaPyramid() {
    const vertexPos = [
        vec4(0.0, 0.5, 0.0, 1.0),
        vec4(1.0, 0.5, 0.0, 1.0),
        vec4(0.5, 0.5, -0.866, 1.0),
        vec4(-0.5, 0.5, -0.866, 1.0),
        vec4(-1.0, 0.5, 0.0, 1.0),
        vec4(-0.5, 0.5, 0.866, 1.0),
        vec4(0.5, 0.5, 0.866, 1.0),
        vec4(0.0, -1.0, 0.0, 1.0)
    ];

    const vertexColor = [
        vec4(1.0, 1.0, 1.0, 1.0),   // white
        vec4(1.0, 0.0, 0.0, 1.0),   // red
        vec4(1.0, 1.0, 0.0, 1.0),   // yellow
        vec4(0.0, 1.0, 0.0, 1.0),   // green
        vec4(0.0, 1.0, 1.0, 1.0),   // cyan
        vec4(0.0, 0.0, 1.0, 1.0),   // blue
        vec4(1.0, 0.0, 1.0, 1.0),   // magenta
        vec4(0.0, 0.0, 0.0, 1.0)    // black
    ];

    numVertPyraTri = 0;
    for (var i = 1; i < 6; i++) {
        points.push(vertexPos[0]);
        colors.push(vertexColor[0]);
        numVertPyraTri++;
        points.push(vertexPos[i]);
        colors.push(vertexColor[i]);
        numVertPyraTri++;
        points.push(vertexPos[i+1]);
        colors.push(vertexColor[i+1]);
        numVertPyraTri++;
```

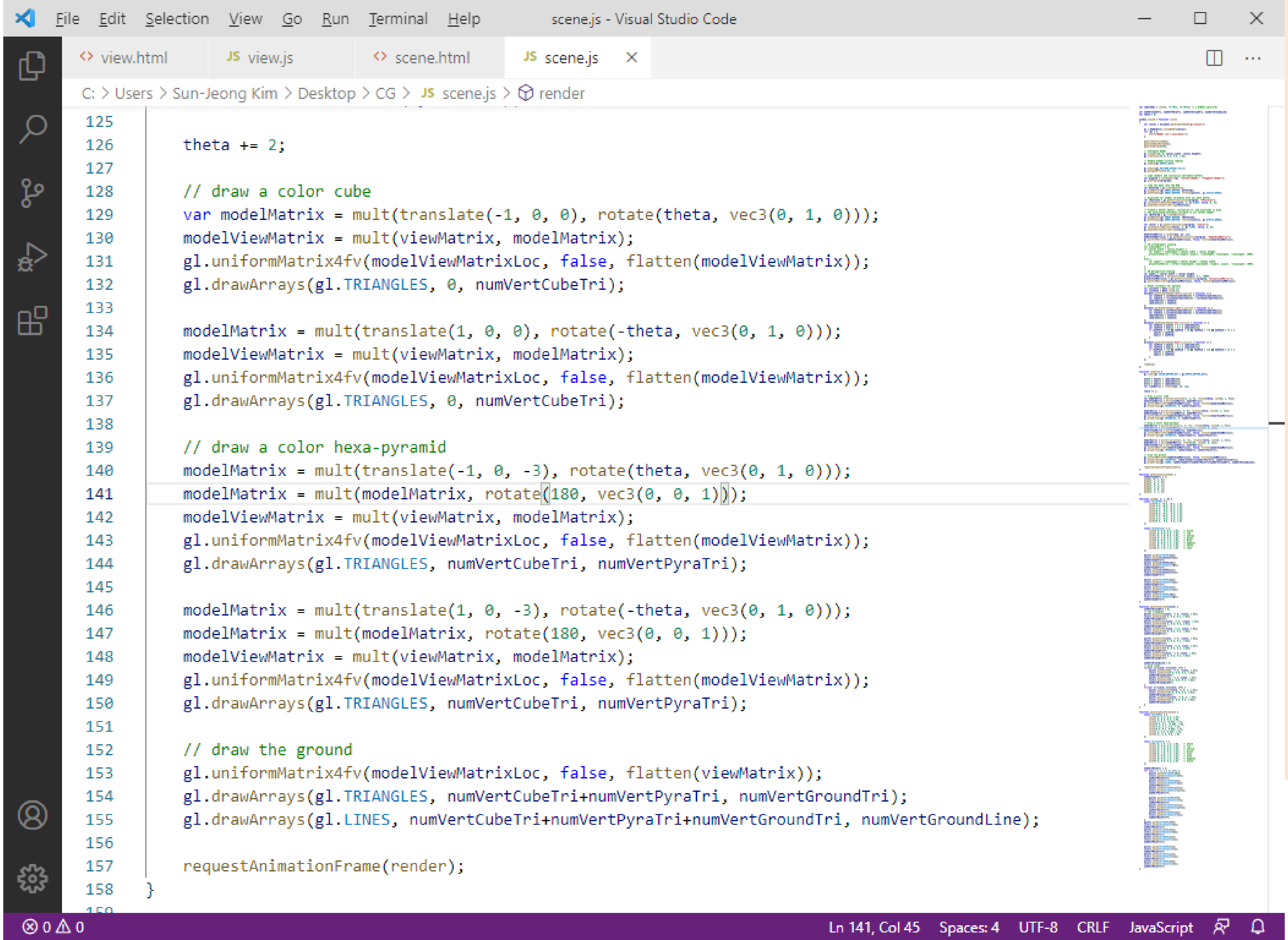

File
Edit
Selection
View
Go
Run
Terminal
Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > numVertPyraTri
⊗ 0 ⚠ 0
Ln 12, Col 35 (14 selected)
Spaces: 4
UTF-8
CRLF
JavaScript

```
            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++;
    }
```


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

```javascript

    theta += 2;

    // draw a color cube
    var modelMatrix = mult(translate(-1, 0, 0), rotate(theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    modelMatrix = mult(translate(1, 0, 0), rotate(-theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    // draw a color hexa-pyramid
    modelMatrix = mult(translate(-1, 0, -3), rotate(theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);

    modelMatrix = mult(translate(1, 0, -3), rotate(-theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);

    // draw the ground
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri, numVertGroundTri);
    gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertGroundTri, numVertGroundLine);

    requestAnimationFrame(render);
}

function generateColorCube() {
    numVertCubeTri = 0;
```

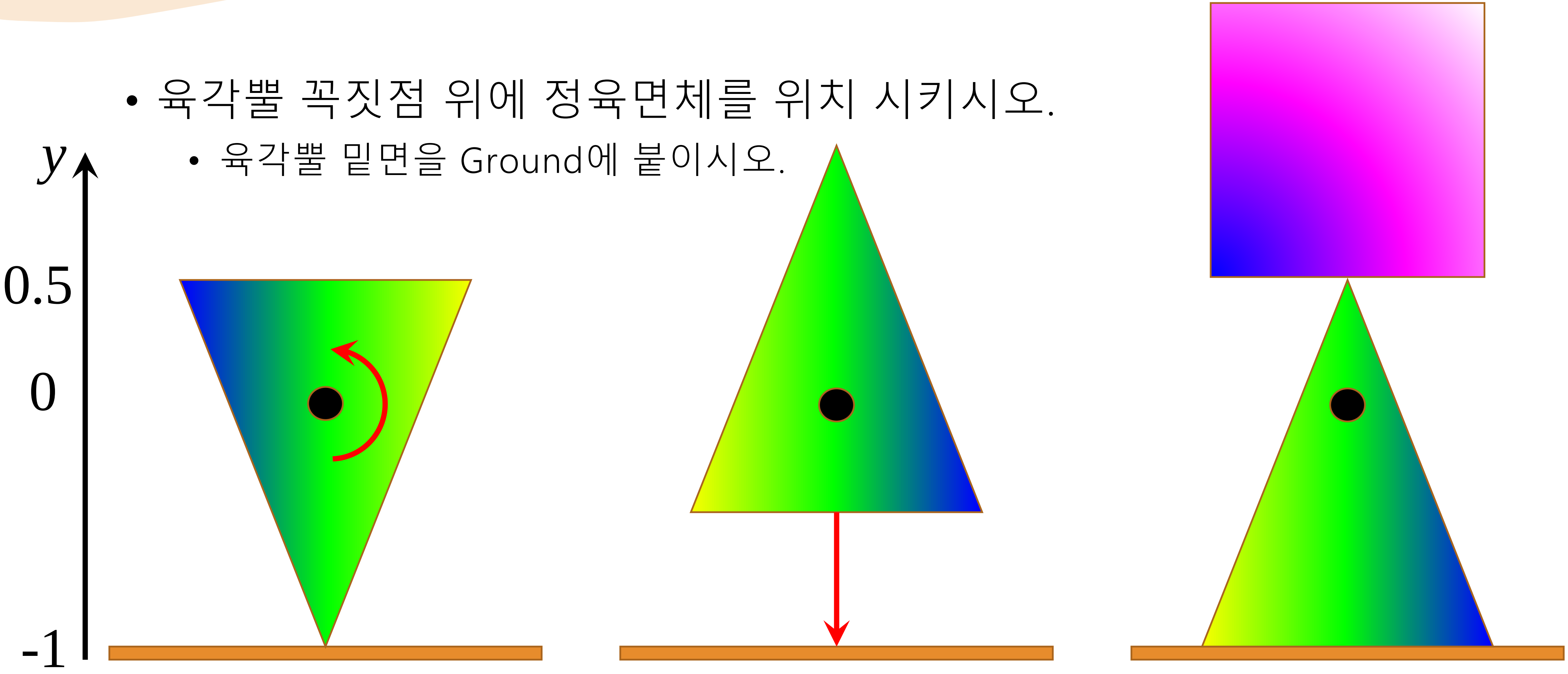

학번 이름
학번 이름
파일 | C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene.html
◀ ▲ ▶
▼

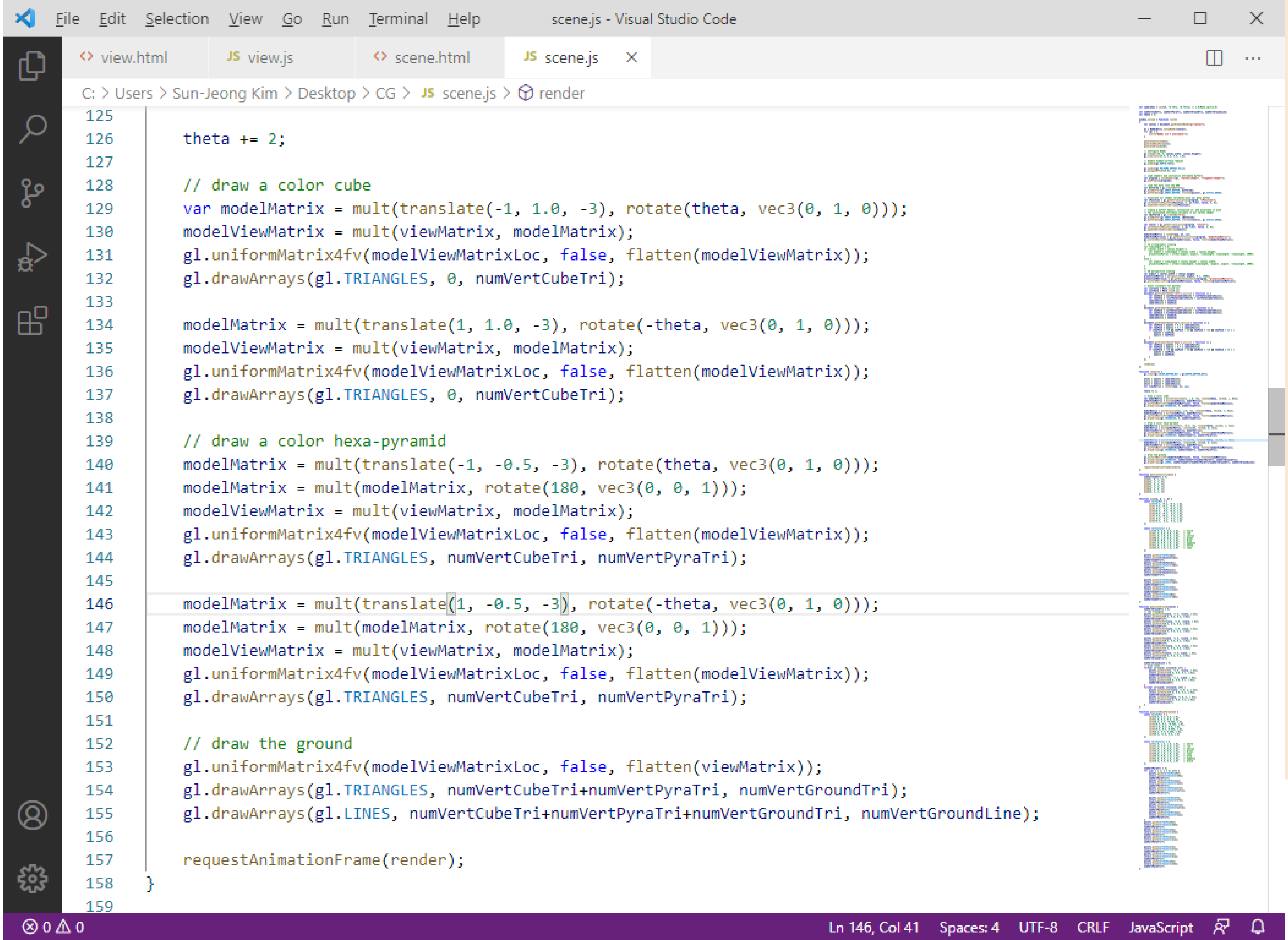


```
    theta += 2;

    // draw a color cube
    var modelMatrix = mult(translate(-1, 0, 0), rotate(theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    modelMatrix = mult(translate(1, 0, 0), rotate(-theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    // draw a color hexa-pyramid
    modelMatrix = mult(translate(-1, 0, -3), rotate(theta, vec3(0, 1, 0)));
    modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);

    modelMatrix = mult(translate(1, 0, -3), rotate(-theta, vec3(0, 1, 0)));
    modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);

    // draw the ground
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri, numVertGroundTri);
    gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertGroundTri, numVertGroundLine);

    requestAnimationFrame(render);
}
```

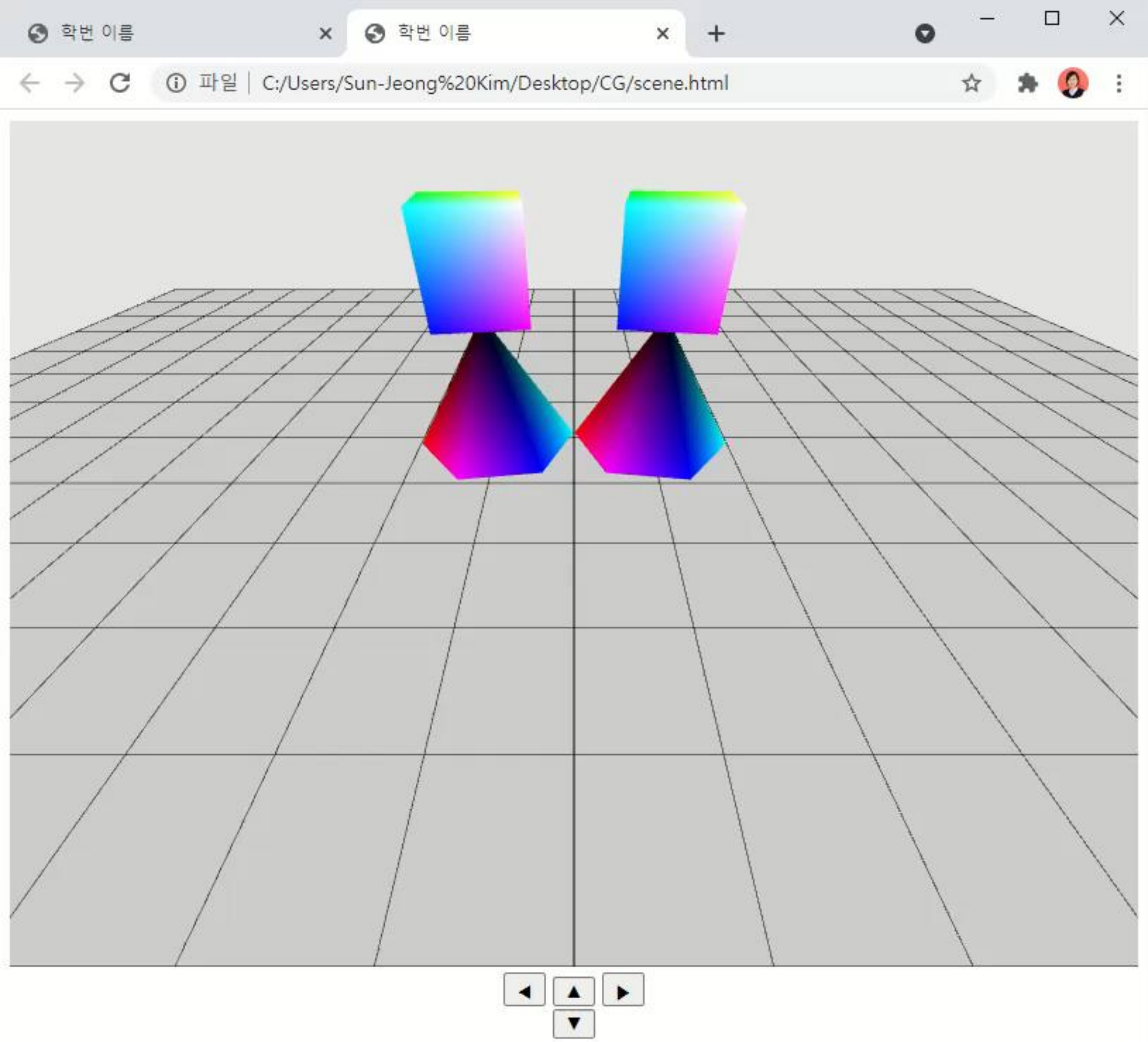
연습 문제 (1)

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





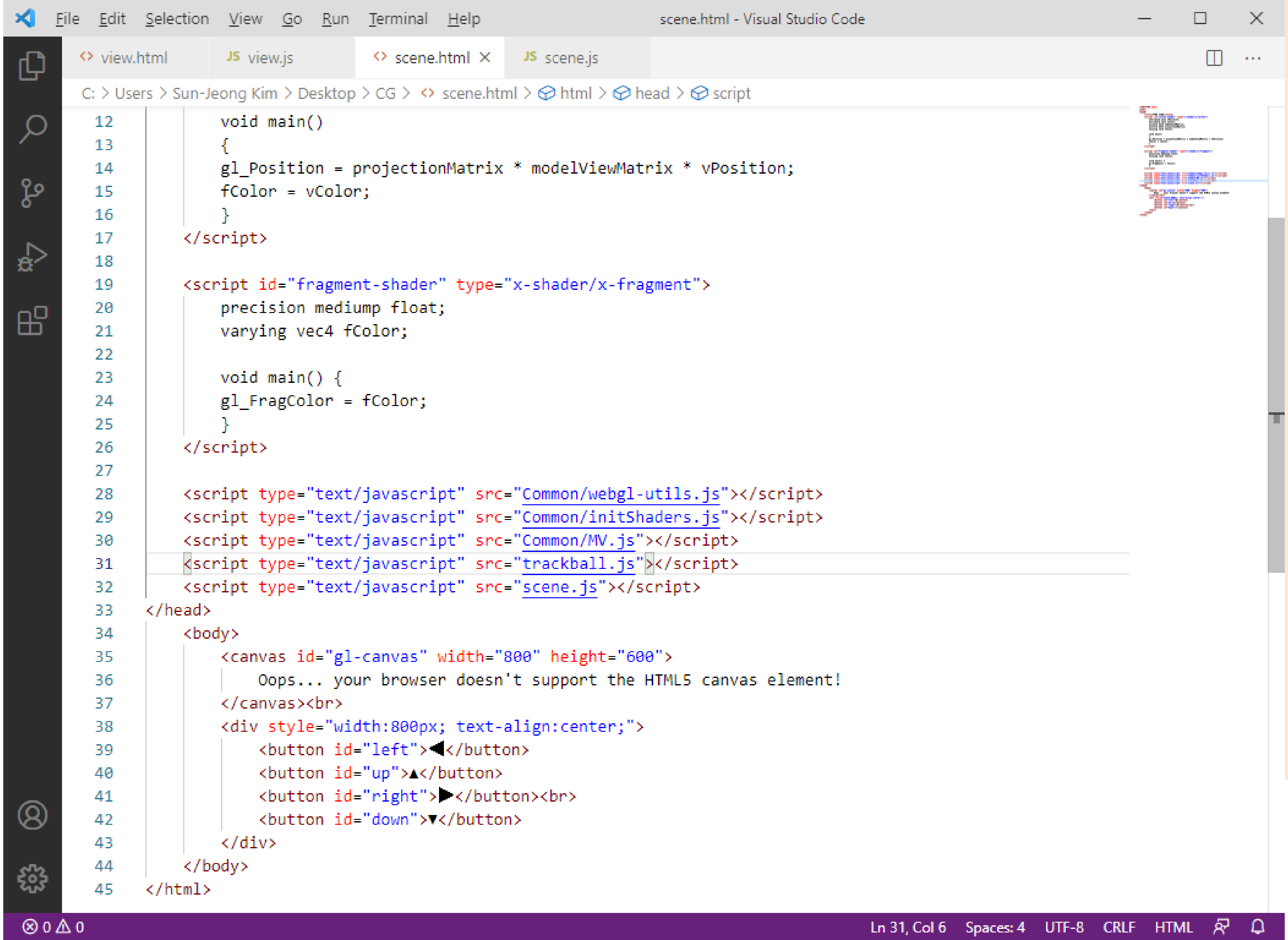
```

    theta += 2;

    // draw a color cube
    var modelMatrix = mult(translate(-1, 1.0, -3), rotate(theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    modelMatrix = mult(translate(1, 1.0, -3), rotate(-theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    // draw a color hexa-pyramid
    modelMatrix = mult(translate(-1, -0.5, -3), rotate(theta, vec3(0, 1, 0)));
    modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);

    modelMatrix = mult(translate(1, -0.5, -3), rotate(-theta, vec3(0, 1, 0)));
    modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);

    // draw the ground
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri, numVertGroundTri);
    gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertGroundTri, numVertGroundLine);

    requestAnimationFrame(render);
}

```



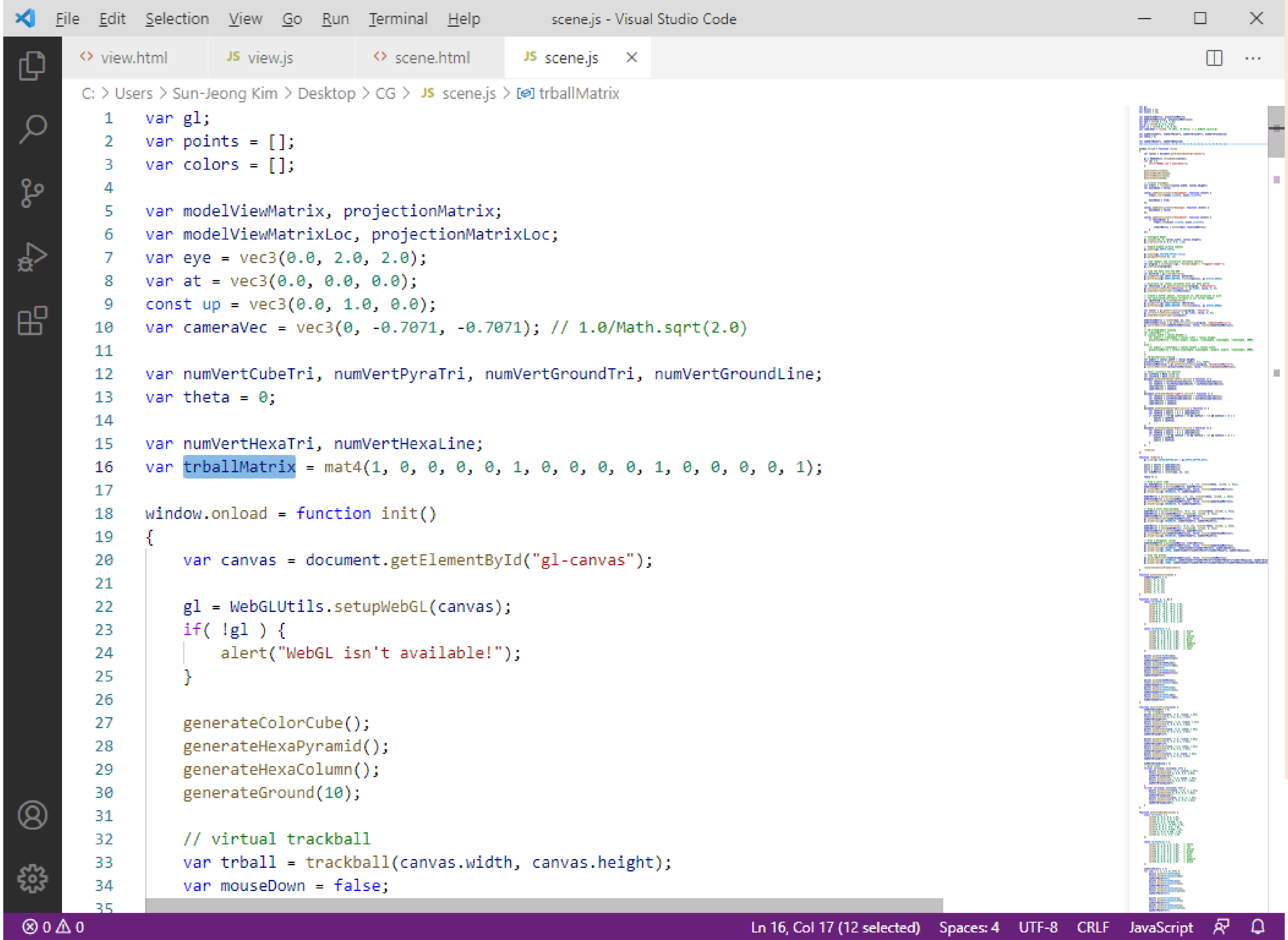
C:/Users/Sun-Jeong%20Kim/Desktop/CG/scene.html

연습 문제 (2)

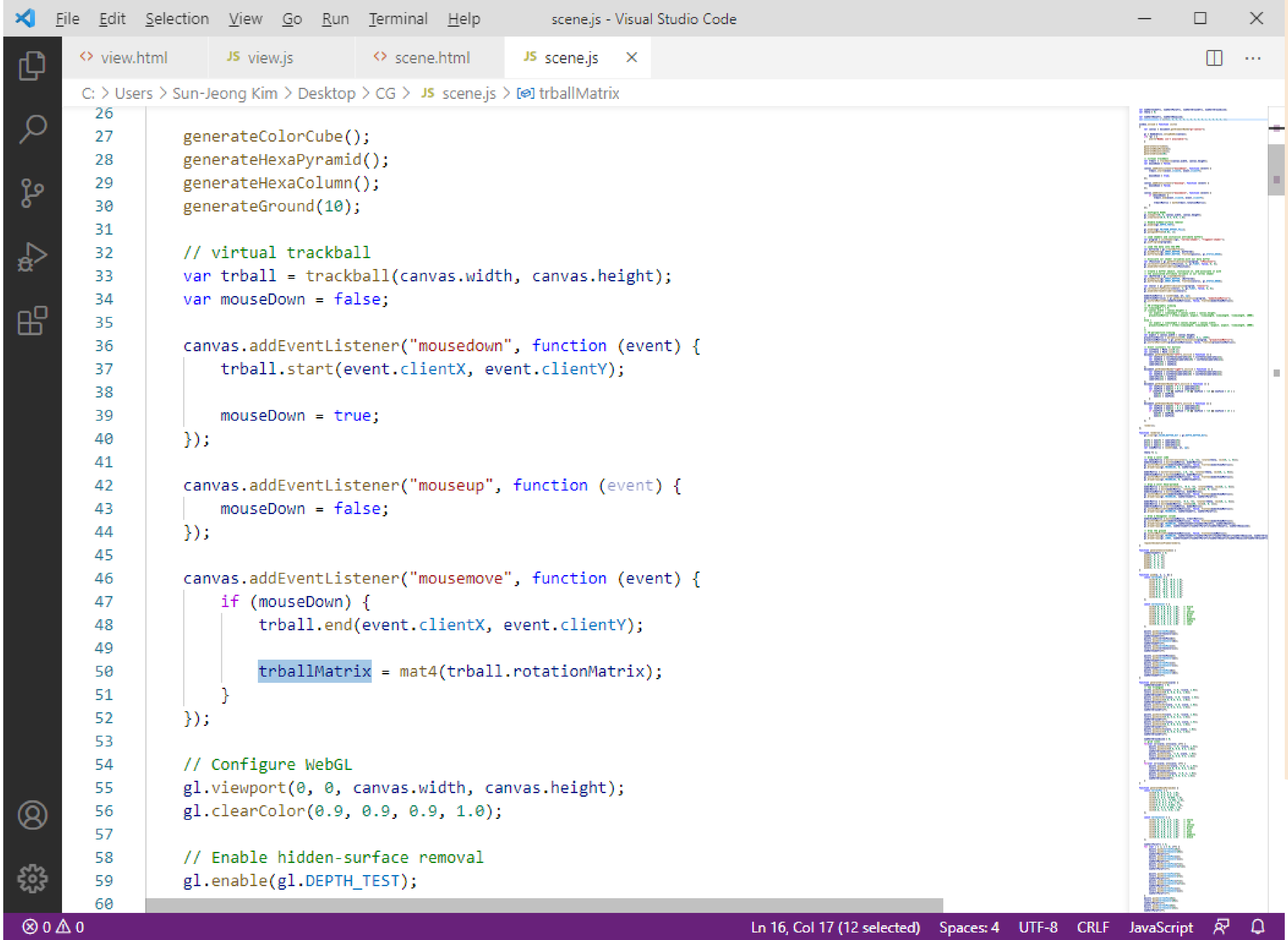
- 육각기둥을 모델링하여 화면 중앙에 위치 시키시오.
- 육각기둥은 단색으로 색칠하시오.
  - 선도 함께 그리시오.
- 육각기둥은 트랙볼로 회전 시키시오.


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
⊗ 0 ⚠ 0
Ln 31, Col 6
Spaces: 4
UTF-8
CRLF
HTML

```
        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><br>
        <div style="width:800px; text-align:center;">
            <button id="left">◀</button>
            <button id="up">▲</button>
            <button id="right">▶</button><br>
            <button id="down">▼</button>
        </div>
    </body>
</html>
```

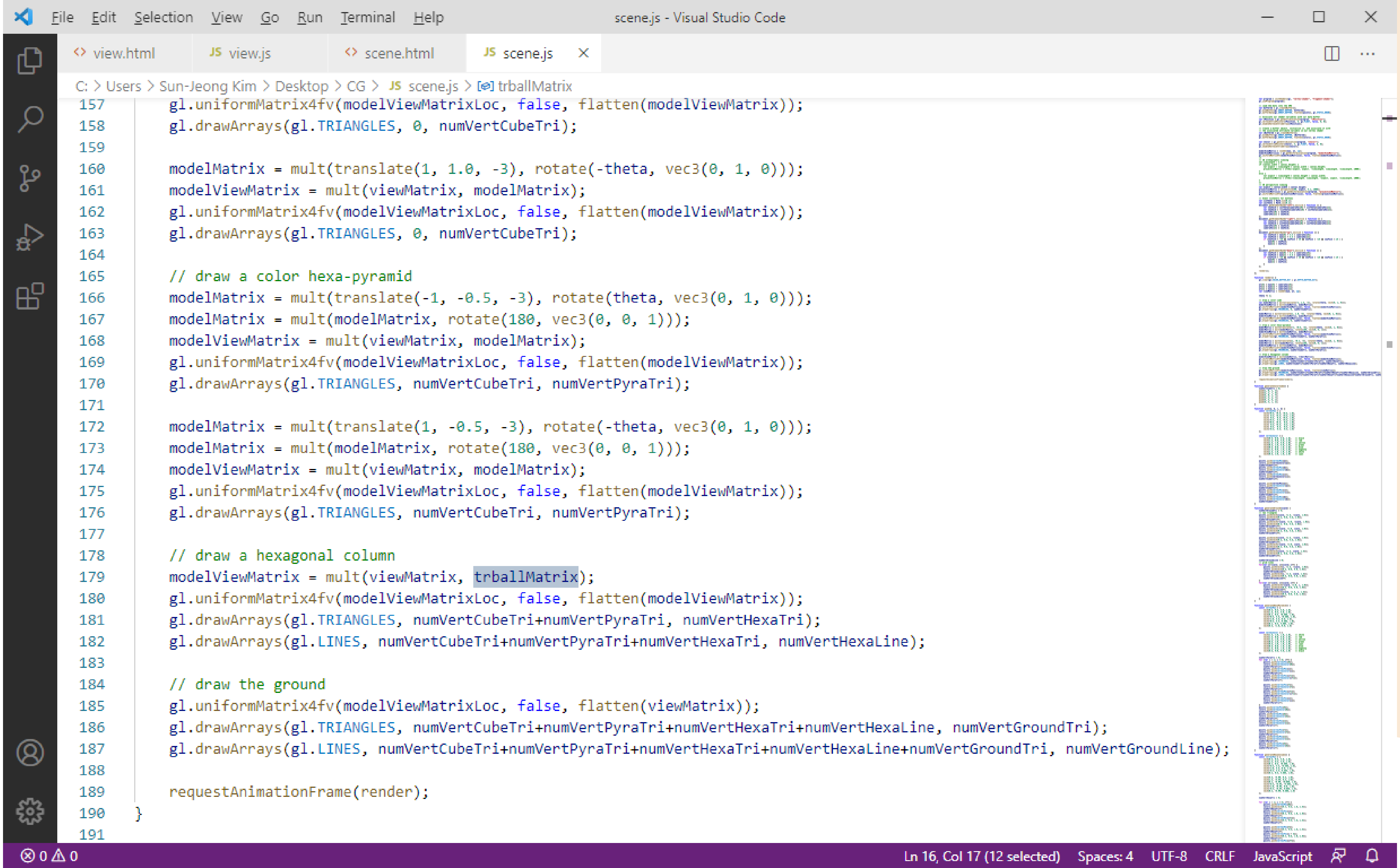

File
Edit
Selection
View
Go
Run
Terminal
Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > trballMatrix
⊗ 0 ⚠ 0
Ln 16, Col 17 (12 selected)
Spaces: 4
UTF-8
CRLF
JavaScript

```
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;

var numVertHexaTri, numVertHexaLine;
var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);

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

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

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

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

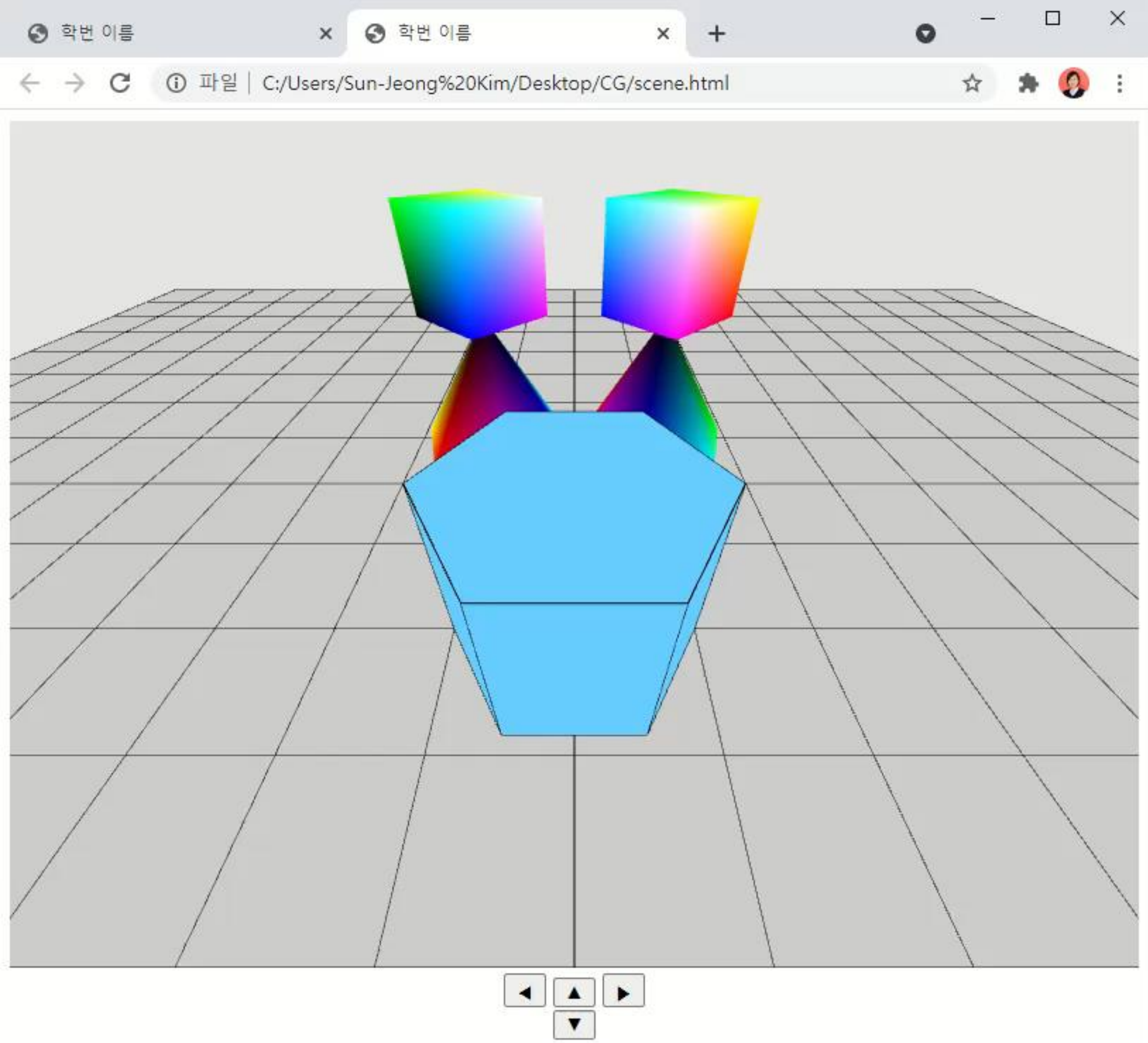


File Edit Selection View Go Run Terminal Help
scene.js - Visual Studio Code
view.html
view.js
scene.html
scene.js
C: > Users > Sun-Jeong Kim > Desktop > CG > scene.js > trballMatrix
⊗ 0 ⚠ 0
Ln 16, Col 17 (12 selected)
Spaces: 4
UTF-8
CRLF
JavaScript

```javascript

    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);

```


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

```
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    modelMatrix = mult(translate(1, 1.0, -3), rotate(-theta, vec3(0, 1, 0)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, 0, numVertCubeTri);

    // draw a color hexa-pyramid
    modelMatrix = mult(translate(-1, -0.5, -3), rotate(theta, vec3(0, 1, 0)));
    modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);

    modelMatrix = mult(translate(1, -0.5, -3), rotate(-theta, vec3(0, 1, 0)));
    modelMatrix = mult(modelMatrix, rotate(180, vec3(0, 0, 1)));
    modelViewMatrix = mult(viewMatrix, modelMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri, numVertPyraTri);

    // draw a hexagonal column
    modelViewMatrix = mult(viewMatrix, trballMatrix);
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri, numVertHexaTri);
    gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertHexaTri, numVertHexaLine);

    // draw the ground
    gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
    gl.drawArrays(gl.TRIANGLES, numVertCubeTri+numVertPyraTri+numVertHexaTri+numVertHexaLine, numVertGroundTri);
    gl.drawArrays(gl.LINES, numVertCubeTri+numVertPyraTri+numVertHexaTri+numVertHexaLine+numVertGroundTri, numVertGroundLine);

    requestAnimationFrame(render);
}

```


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

수고하셨습니다