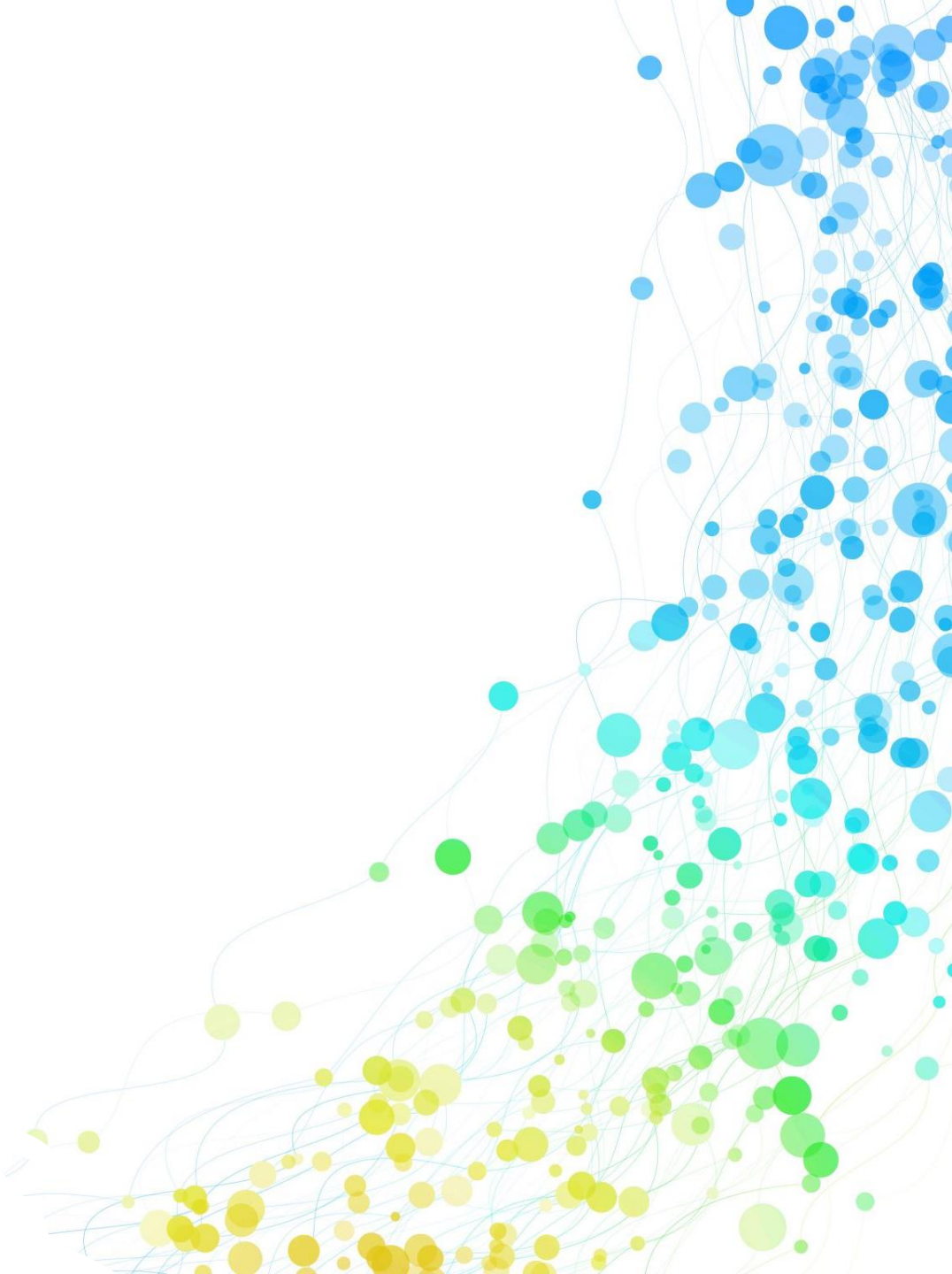
Lighting

10TH WEEK, 2021



Point Light Sources

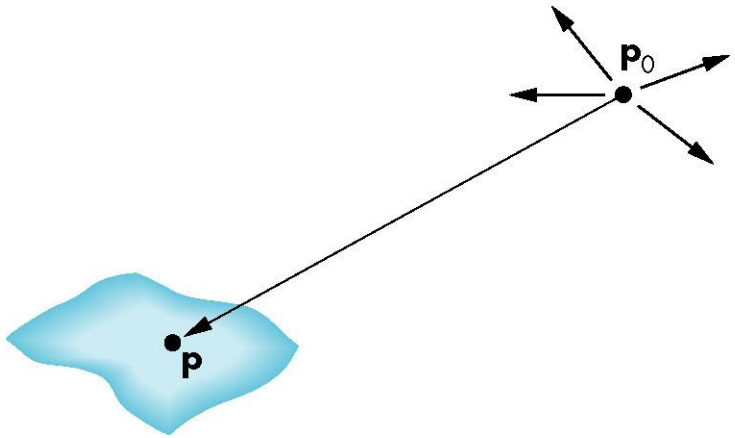
- Emitting light equally in all directions
  - $\mathbf{p}_0$: the location of a point light source

$$\mathbf{I}(\mathbf{p}_0) = \begin{bmatrix} I_r(\mathbf{p}_0) \\ I_g(\mathbf{p}_0) \\ I_b(\mathbf{p}_0) \end{bmatrix}$$

  - Attenuation
    - Proportional to the inverse square distance

$$\mathbf{I}(\mathbf{p}, \mathbf{p}_0) = \frac{1}{|\mathbf{p} - \mathbf{p}_0|^2} \mathbf{I}(\mathbf{p}_0)$$

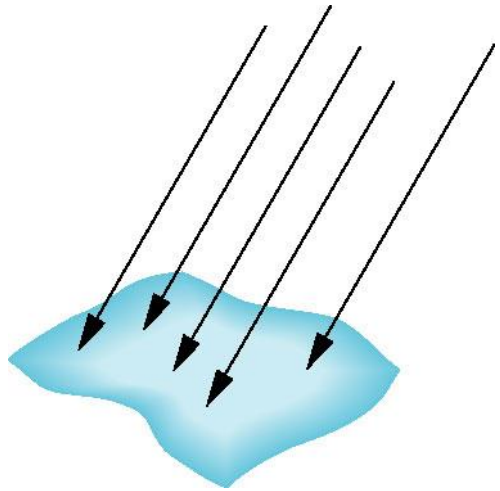
$$\mathbf{I}(\mathbf{p}, \mathbf{p}_0) = \frac{1}{k_c + k_l d + k_q d^2} \mathbf{I}(\mathbf{p}_0)$$



Directional Light Sources

- Parallel direction of lights
  - Infinite distance away from the surface
  - Location → direction

$$\mathbf{p}_0 = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow \mathbf{p}_0 = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$



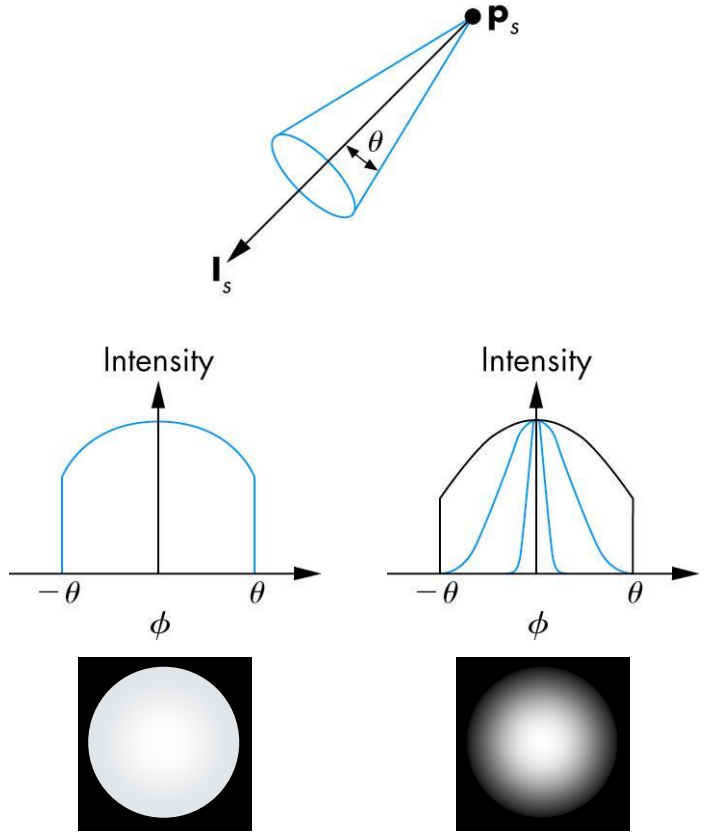
Spotlight Sources

- Characterized by a narrow range of angle through which light is emitted
  - $\mathbf{p}_s$: apex of a cone
  - $\mathbf{l}_s$: direction of pointing
  - θ : angle to determine width
- Distribution of light
  - Concentrating in the center

$$\cos\phi = \mathbf{s}\cdot\mathbf{l}_s$$

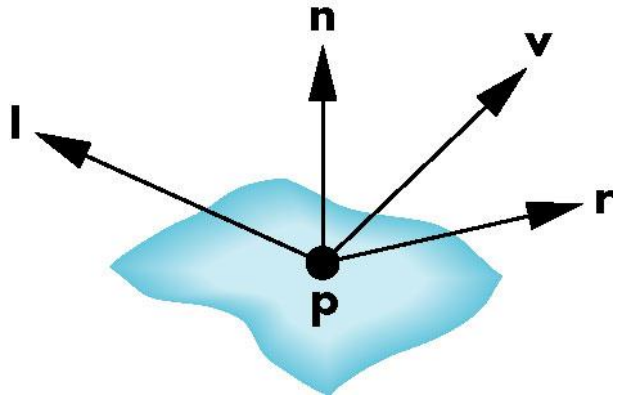
  - Light intensity drop off

$$\cos^e\phi = (\mathbf{s}\cdot\mathbf{l}_s)^e$$



Phong Reflection Model

- A simple model that can compute rapidly
- Three components (light-material interactions)
  - Ambient
  - Diffuse
  - Specular
- Using four vectors
  - **n**: normal
  - **v**: to the viewer or COP
  - **l**: to light source
  - **r**: perfect reflector



Ambient Reflection

- Same at every point on the surface
- Ambient reflection coefficient

$$\mathbf{k}_a = (k_{ar},\ k_{ag},\ k_{ab}), \quad 0 \le k_{ar},\ k_{ag},\ k_{ab} \le 1$$

  - Amount reflected
    - Some is absorbed and some is reflected
  - Three components (red, green, blue)
- Ambient reflection term in rendering equation

$$\mathbf{I}_a = \mathbf{k}_a \mathbf{L}_a$$

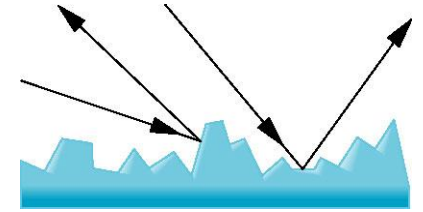
  - Can be any of the individual light sources
  - Can be a global ambient term

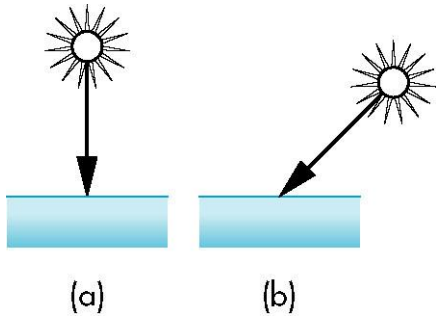
Diffuse Reflection

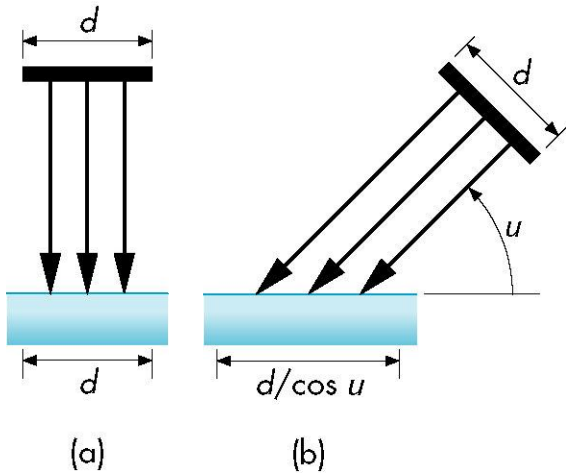
- Light scattered equally in all directions
  - Perfectly diffuse surface

  ➔ So rough that there is no preferred angle of reflection
- Lambert's law
  - Amount of light reflected is proportional to vertical component of incoming light
    - Reflected light $\propto \cos u$ $\quad \cos u = \mathbf{l} \cdot \mathbf{n}$
- Diffuse reflection term $\quad \mathbf{I}_d = \mathbf{k}_d (\mathbf{l} \cdot \mathbf{n}) \mathbf{L}_d$
  - Incorporating a distance term

$$\mathbf{I}_{\mathrm{d}} = \frac{\mathbf{k}_{\mathrm{d}}}{k_c + k_l d + k_q d^2} (\mathbf{l} \cdot \mathbf{n}) \mathbf{L}_{\mathrm{d}}$$

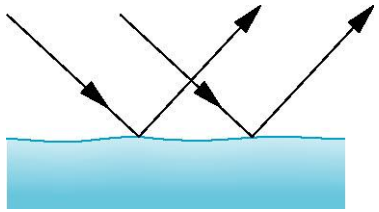


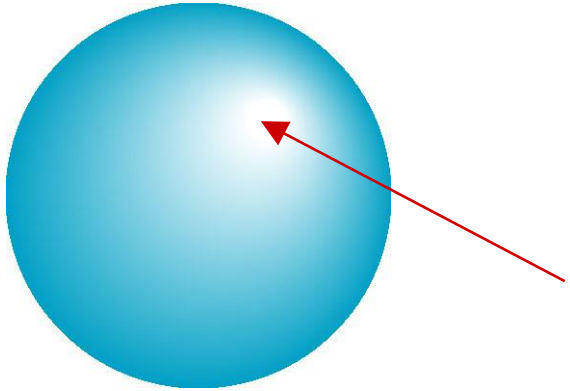




Specular Reflection

- Smooth surfaces show specular highlights

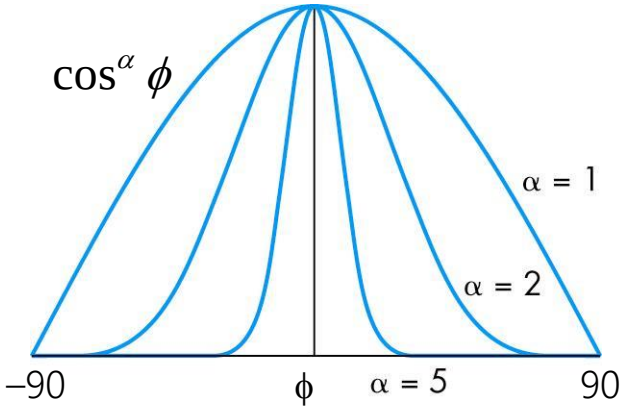




- Specular reflection term

$$\mathbf{I}_s = \mathbf{k}_s (\mathbf{r} \cdot \mathbf{v})^{\alpha} \mathbf{L}_s$$

  - α: <u>shininess</u> coefficient
    - α → infinite : mirror
    - 100 < α < 200 : metal
    - 5< α < 10 : plastic



Drawing a Color Cube

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

light.html | light.js

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

```
15
16          <script id="fragment-shader" type="x-shader/x-fragment">
17          precision mediump float;
18          varying vec4 fColor;
19
20          void main() {
21              gl_FragColor = fColor;
22          }
23          </script>
24
25          <script type="text/javascript" src="Common/webgl-utils.js"></script>
26          <script type="text/javascript" src="Common/initShaders.js"></script>
27          <script type="text/javascript" src="Common/MV.js"></script>
28          <script type="text/javascript" src="trackball.js"></script>
29          <script type="text/javascript" src="light.js"></script>
30      </head>
31      <body>
32          <canvas id="gl-canvas" width="512" height="512">
33              Oops... your browser doesn't support the HTML5 canvas element!
34          </canvas>
35          <div style="width: 512px; text-align: center;">
36              <button id="xButton">Rotate X</button>
37              <button id="yButton">Rotate Y</button>
38              <button id="zButton">Rotate Z</button>
39              <button id="buttonT">Toggle Rotate</button>
40          </div>
41      </body>
42  </html>
```

⊗ 0 ⚠ 0 Ln 4, Col 21 Spaces: 4 UTF-8 CRLF HTML

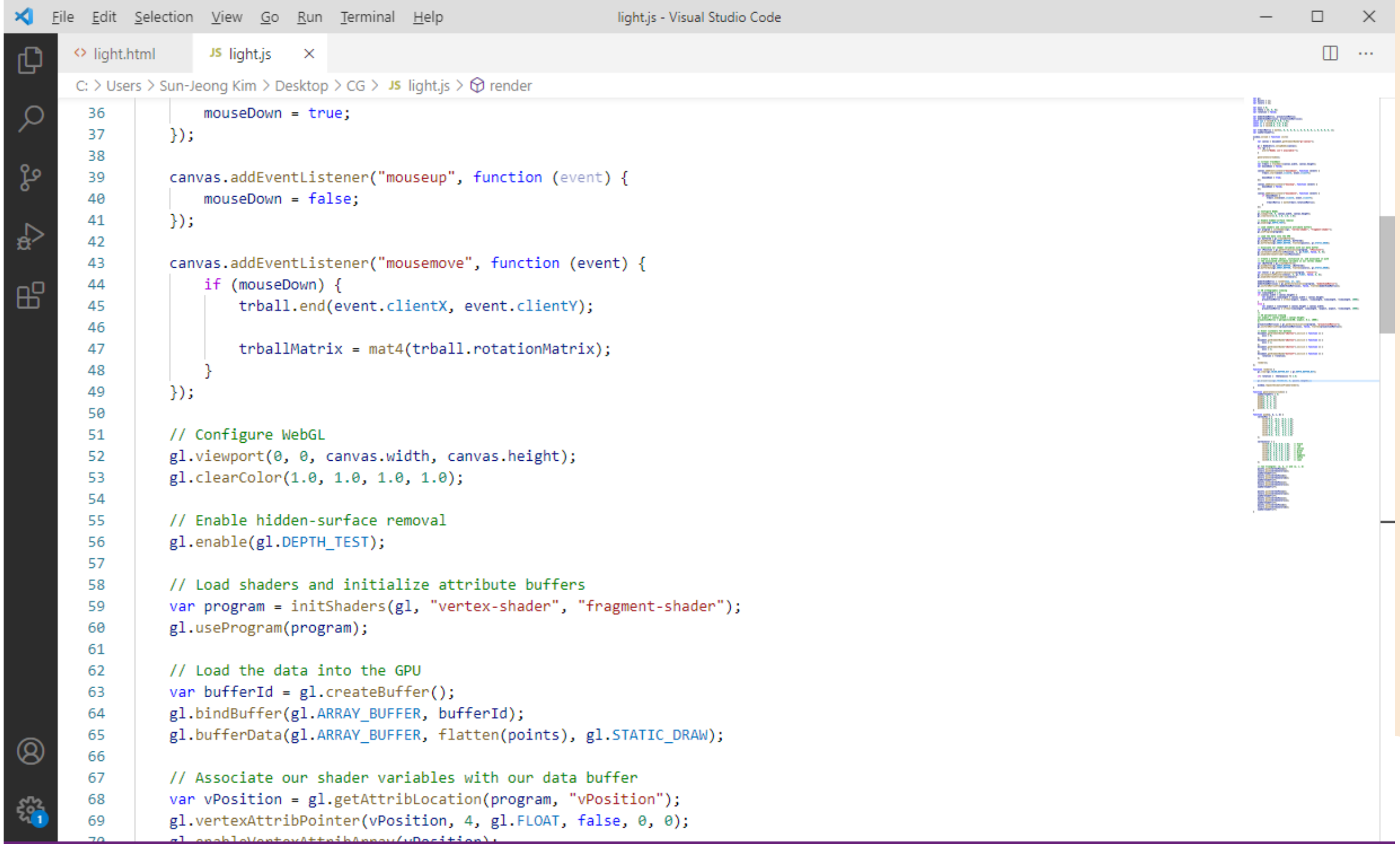
File Edit Selection View Go Run Terminal Help light.js - Visual Studio Code

light.html | light.js

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

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

var axis = 0;
var theta = [0, 0, 0];
var rotation = false;

var modelViewMatrix, projectionMatrix;
var modelViewMatrixLoc, projectionMatrixLoc;
const eye = vec3(0.0, 0.0, 1.0);
const at = vec3(0.0, 0.0, 0.0);
const up = vec3(0.0, 1.0, 0.0);

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

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

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

    generateColorCube();

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

    canvas.addEventListener("mousedown", function (event) {
        trball.start(event.clientX, event.clientY);
```

⊗ 0 ⚠ 0 Ln 126, Col 51 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
light.js - Visual Studio Code
light.html
light.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > render
36 mouseDown = true;
37 });
38
39 canvas.addEventListener("mouseup", function (event) {
40 mouseDown = false;
41 });
42
43 canvas.addEventListener("mousemove", function (event) {
44 if (mouseDown) {
45 trball.end(event.clientX, event.clientY);
46
47 trballMatrix = mat4(trball.rotationMatrix);
48 }
49 });
50
51 // Configure WebGL
52 gl.viewport(0, 0, canvas.width, canvas.height);
53 gl.clearColor(1.0, 1.0, 1.0, 1.0);
54
55 // Enable hidden-surface removal
56 gl.enable(gl.DEPTH_TEST);
57
58 // Load shaders and initialize attribute buffers
59 var program = initShaders(gl, "vertex-shader", "fragment-shader");
60 gl.useProgram(program);
61
62 // Load the data into the GPU
63 var bufferId = gl.createBuffer();
64 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
65 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
66
67 // Associate our shader variables with our data buffer
68 var vPosition = gl.getAttribLocation(program, "vPosition");
69 gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
0 0
Ln 126, Col 51 Spaces: 4 UTF-8 CRLF JavaScript

light.html | light.js

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

```
70      gl.enableVertexAttribArray(vPosition);
71
72      // Create a buffer object, initialize it, and associate it with
73      // the associated attribute variable in our vertex shader
74      var cBufferId = gl.createBuffer();
75      gl.bindBuffer(gl.ARRAY_BUFFER, cBufferId);
76      gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
77
78      var vColor = gl.getAttribLocation(program, "vColor");
79      gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
80      gl.enableVertexAttribArray(vColor);
81
82      modelViewMatrix = lookAt(eye, at, up);
83      modelViewMatrixLoc = gl.getUniformLocation(program, "modelViewMatrix");
84      gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
85
86      // 3D orthographic viewing
87      var viewLength = 1.0;
88      if (canvas.width > canvas.height) {
89          var aspect = viewLength * canvas.width / canvas.height;
90          projectionMatrix = ortho(-aspect, aspect, -viewLength, viewLength, -viewLength, 1000);
91      }
92      else {
93          var aspect = viewLength * canvas.height / canvas.width;
94          projectionMatrix = ortho(-viewLength, viewLength, -aspect, aspect, -viewLength, 1000);
95      }
96      /*
97      // 3D perspective viewing
98      var aspect = canvas.width / canvas.height;
99      projectionMatrix = perspective(90, aspect, 0.1, 1000);
100     */
101     projectionMatrixLoc = gl.getUniformLocation(program, "projectionMatrix");
102     gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projectionMatrix));
103
104     // Event listeners for buttons
```

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

light.html | light.js

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

```
104     // Event listeners for buttons
105     document.getElementById("xButton").onclick = function () {
106         axis = 0;
107     };
108     document.getElementById("yButton").onclick = function () {
109         axis = 1;
110     };
111     document.getElementById("zButton").onclick = function () {
112         axis = 2;
113     };
114     document.getElementById("buttonT").onclick = function () {
115         rotation = !rotation;
116     };
117
118     render();
119 };
120
121 function render() {
122     gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
123
124     if( rotation )  theta[axis] += 2.0;
125
126     gl.drawArrays(gl.TRIANGLES, 0, points.length);
127
128     window.requestAnimationFrame(render);
129 }
130
131 function generateColorCube() {
132     numVertCubeTri = 0;
133     quad(1, 0, 3, 2);
134     quad(2, 3, 7, 6);
135     quad(3, 0, 4, 7);
136     quad(4, 5, 6, 7);
137     quad(5, 4, 0, 1);
```

⊗ 0 ⚠ 0 Ln 126, Col 51 Spaces: 4 UTF-8 CRLF JavaScript

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

light.html | light.js

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

```
138     quad(6, 5, 1, 2);
139 }
140
141 function quad(a, b, c, d) {
142     vertexPos = [
143         vec4(-0.5, -0.5, -0.5, 1.0),
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     ];
152
153     vertexColor = [
154         vec4(0.0, 0.0, 0.0, 1.0),   // black
155         vec4(1.0, 0.0, 0.0, 1.0),   // red
156         vec4(1.0, 1.0, 0.0, 1.0),   // yellow
157         vec4(0.0, 1.0, 0.0, 1.0),   // green
158         vec4(0.0, 0.0, 1.0, 1.0),   // blue
159         vec4(1.0, 0.0, 1.0, 1.0),   // magenta
160         vec4(1.0, 1.0, 1.0, 1.0),   // white
161         vec4(0.0, 1.0, 1.0, 1.0)    // cyan
162     ];
163
164     // two triangles: (a, b, c) and (a, c, d)
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]);
```

⊗ 0 ⚠ 0 Ln 126, Col 51 Spaces: 4 UTF-8 CRLF JavaScript

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

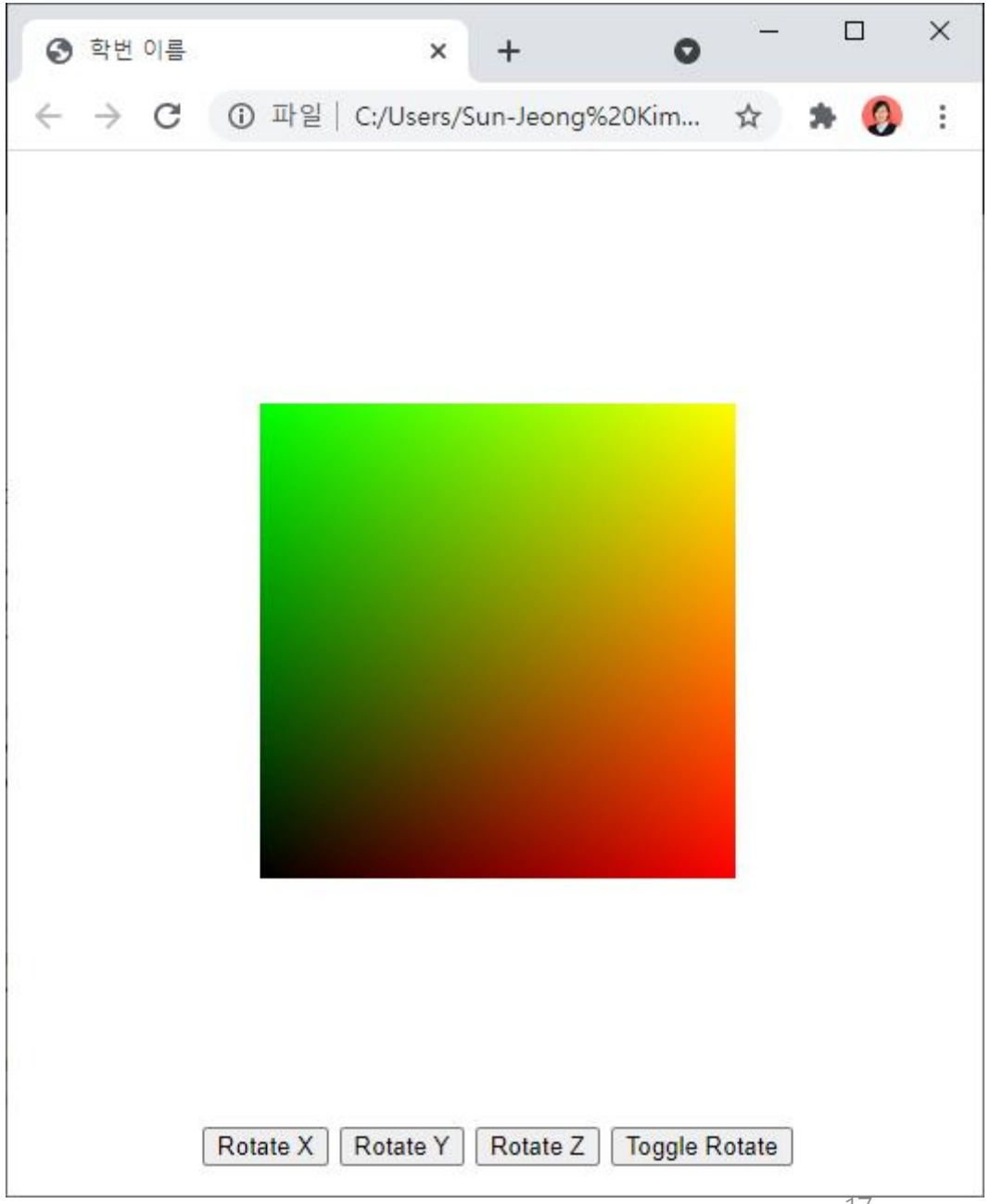
light.html | light.js

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

```
153     vertexColor = [
154         vec4(0.0, 0.0, 0.0, 1.0),   // black
155         vec4(1.0, 0.0, 0.0, 1.0),   // red
156         vec4(1.0, 1.0, 0.0, 1.0),   // yellow
157         vec4(0.0, 1.0, 0.0, 1.0),   // green
158         vec4(0.0, 0.0, 1.0, 1.0),   // blue
159         vec4(1.0, 0.0, 1.0, 1.0),   // magenta
160         vec4(1.0, 1.0, 1.0, 1.0),   // white
161         vec4(0.0, 1.0, 1.0, 1.0)    // cyan
162     ];
163
164     // two triangles: (a, b, c) and (a, c, d)
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
```

⊗ 0 ⚠ 0 Ln 126, Col 51 Spaces: 4 UTF-8 CRLF JavaScript

What's Wrong?


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

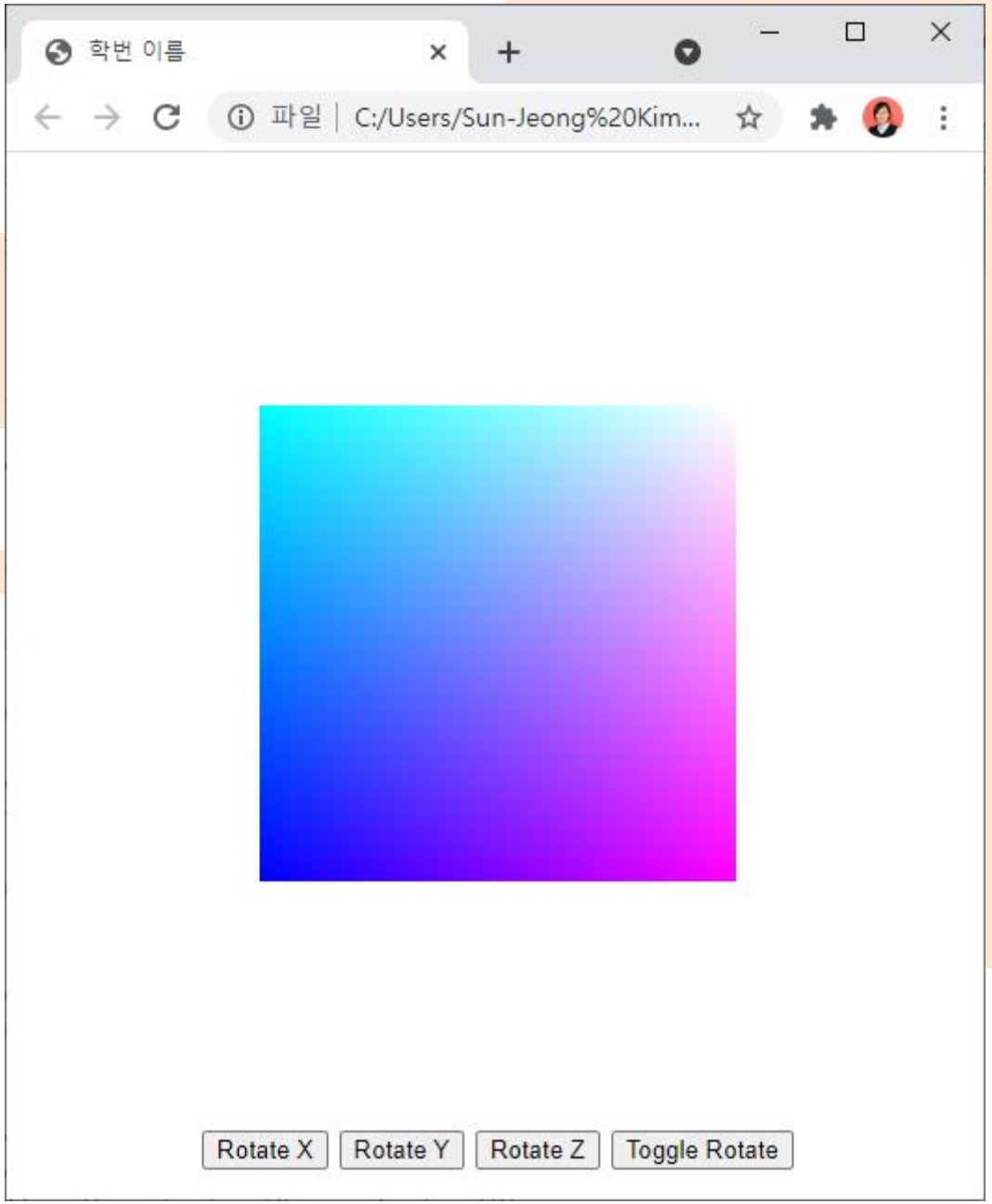
File Edit Selection View Go Run Terminal Help light.html - Visual Studio Code

light.html × light.js

C: > Users > Sun-Jeong Kim > Desktop > CG > light.html > html > head > script#vertex-shader

```
<!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="light.js"></script>
    </head>
    <body>
        <canvas id="gl-canvas" width="512" height="512">
            Oops... your browser doesn't support the HTML5 canvas element!
```

⊗ 0 ⚠ 0 Ln 13, Col 61 (15 selected) Spaces: 4 UTF-8 CRLF HTML


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

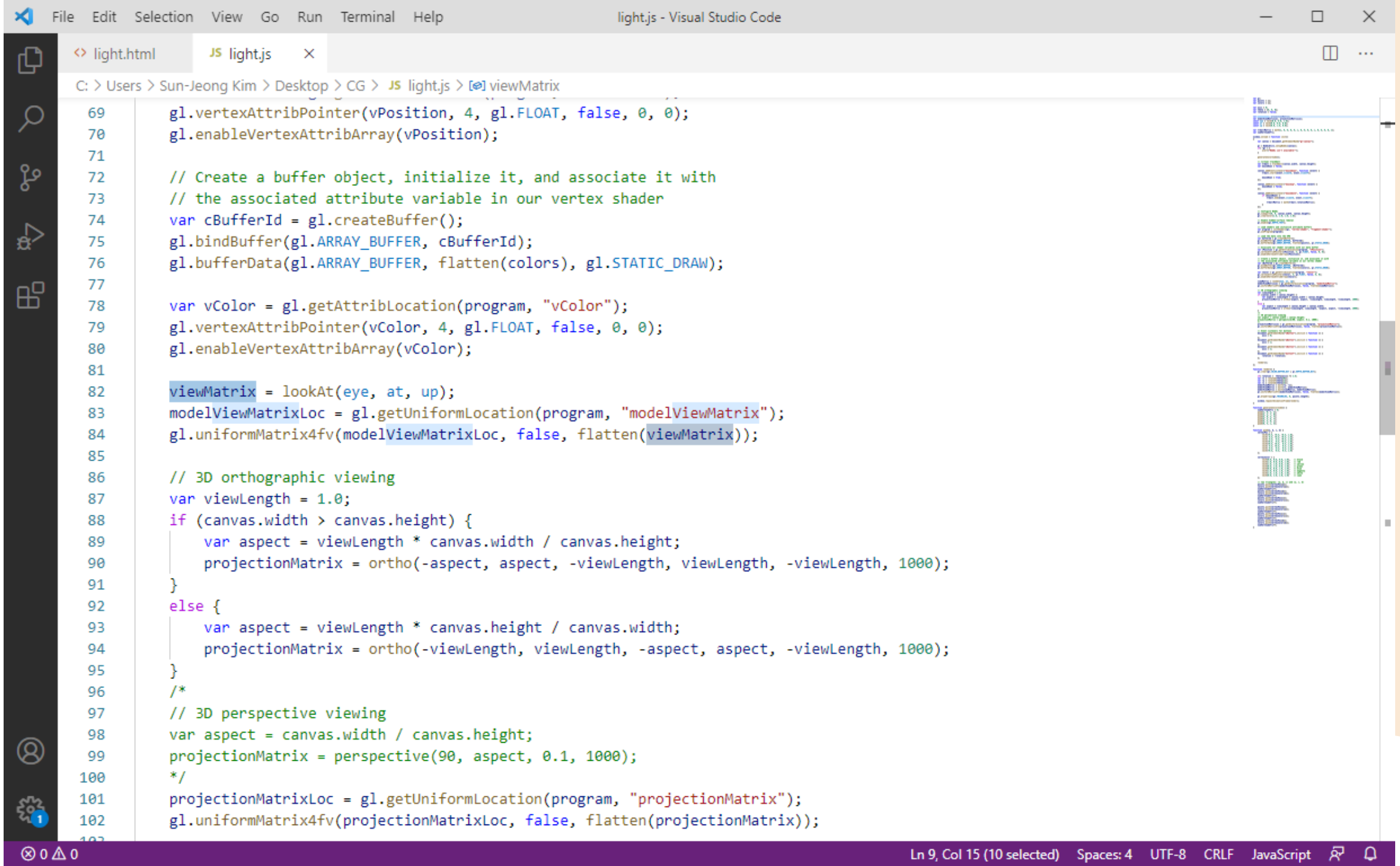
File Edit Selection View Go Run Terminal Help light.js - Visual Studio Code

light.html | light.js

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

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

var axis = 0;
var theta = [0, 0, 0];
var rotation = false;

var viewMatrix, projectionMatrix;
var modelViewMatrixLoc, projectionMatrixLoc;
const eye = vec3(0.0, 0.0, 1.0);
const at = vec3(0.0, 0.0, 0.0);
const up = vec3(0.0, 1.0, 0.0);

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

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

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

    generateColorCube();

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

    canvas.addEventListener("mousedown", function (event) {
        trball.start(event.clientX, event.clientY);
```

⊗ 0 ⚠ 0 Ln 9, Col 15 (10 selected) Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
light.js - Visual Studio Code
light.html
light.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > viewMatrix
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);
viewMatrix = lookAt(eye, at, up);
modelViewMatrixLoc = gl.getUniformLocation(program, "modelViewMatrix");
gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
// 3D orthographic viewing
var viewLength = 1.0;
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);
*/
projectionMatrixLoc = gl.getUniformLocation(program, "projectionMatrix");
gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projectionMatrix));
0 0
Ln 9, Col 15 (10 selected)
Spaces: 4
UTF-8
CRLF
JavaScript

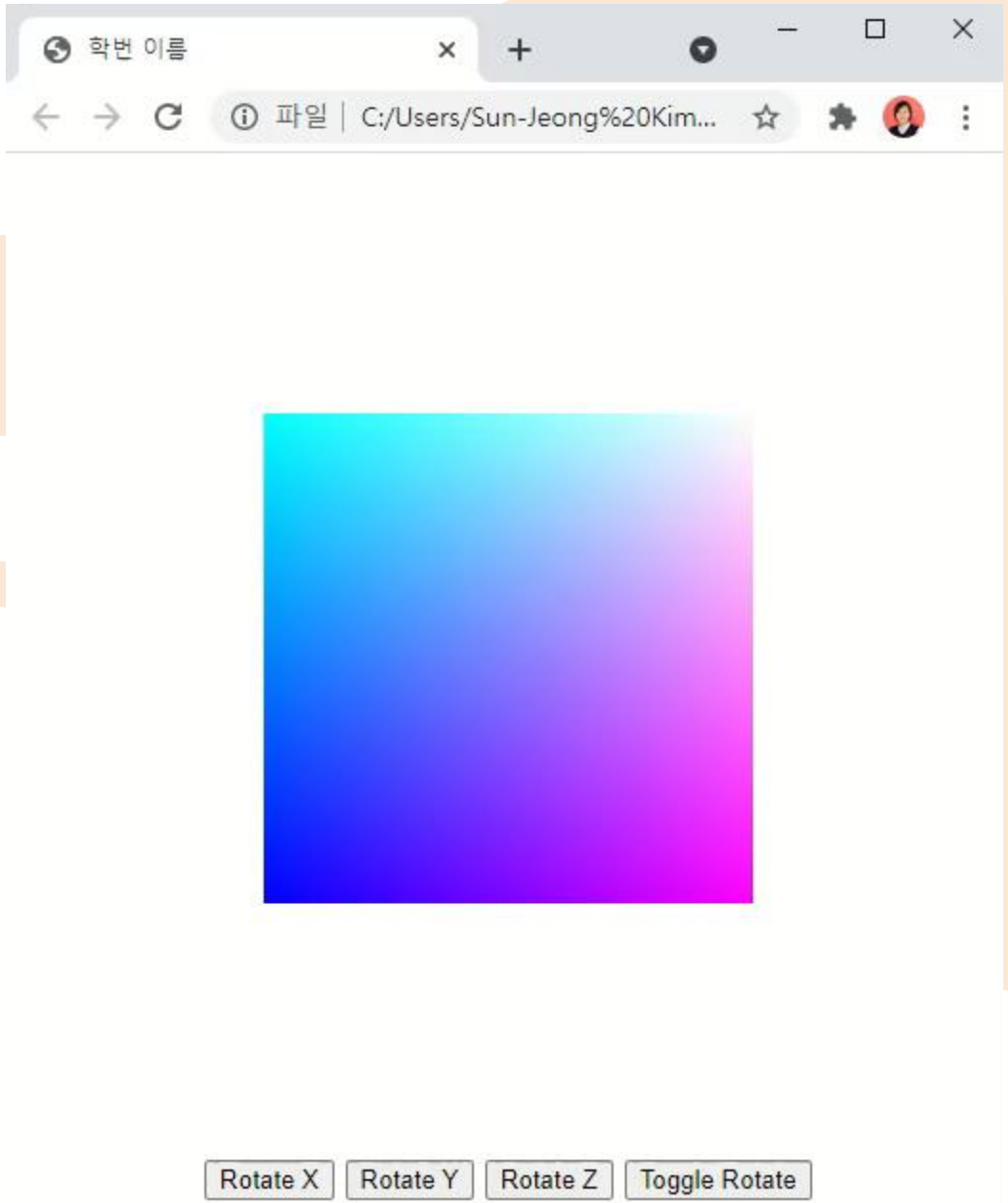
File Edit Selection View Go Run Terminal Help light.js - Visual Studio Code

light.html | light.js

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

```
117
118     render();
119 };
120
121 function render() {
122     gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
123
124     if( rotation )  theta[axis] += 2.0;
125     var rx = rotateX(theta[0]);
126     var ry = rotateY(theta[1]);
127     var rz = rotateZ(theta[2]);
128     modelViewMatrix = mult(ry, rx);
129     modelViewMatrix = mult(rz, modelViewMatrix);
130     modelViewMatrix = mult(viewMatrix, modelViewMatrix);
131     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
132
133     gl.drawArrays(gl.TRIANGLES, 0, points.length);
134
135     window.requestAnimationFrame(render);
136 }
137
138 function generateColorCube() {
139     numVertCubeTri = 0;
140     quad(1, 0, 3, 2);
141     quad(2, 3, 7, 6);
142     quad(3, 0, 4, 7);
143     quad(4, 5, 6, 7);
144     quad(5, 4, 0, 1);
145     quad(6, 5, 1, 2);
146 }
147
148 function quad(a, b, c, d) {
149     vertexPos = [
150         vec4(-0.5, -0.5, -0.5, 1.0),
151         vec4( 0.5, -0.5, -0.5, 1.0),
```

⊗ 0 ⚠ 0 Ln 9, Col 15 (10 selected) Spaces: 4 UTF-8 CRLF JavaScript


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

OpenGL Shading

- Need
  - Normals
  - Material properties
  - Lights

- State-based shading functions have been deprecated (**`glNormal`**, **`glMaterial`**, **`glLight`**)

- Get computer in application or in shaders

Normalization

- Cosine terms in lighting calculations can be computed using dot product
- Unit length vectors simplify calculation
- Usually we want to set the magnitudes to have unit length but
  - Length can be affected by transformations
  - Note that scaling does not preserved length
- GLSL has a normalization function

Specifying a Directional Light Source

- For each light source, we can set an RGBA for the diffuse, specular, and ambient component, and for the direction

```
vec4 diffuse0 = vec4(1.0, 1.0, 0.0, 1.0);
vec4 ambient0 = vec4(1.0, 1.0, 1.0, 1.0);
vec4 specular0 = vec4(0.0, 1.0, 1.0, 1.0);
vec4 light0_dir =vec4(1.0, 2.0, 3,0, 0.0);
```

Direction and Position

- The source colors are specified in RGBA
- The position is given in homogeneous coordinates
  - If w=1.0, we are specifying a finite location
  - If w=0.0, we are specifying a parallel source with the given direction vector
- The coefficient in distance terms are usually quadratic ($1/(k_c + k_l*d + k_q*d*d)$) where d is the distance from the point being rendered to the light source

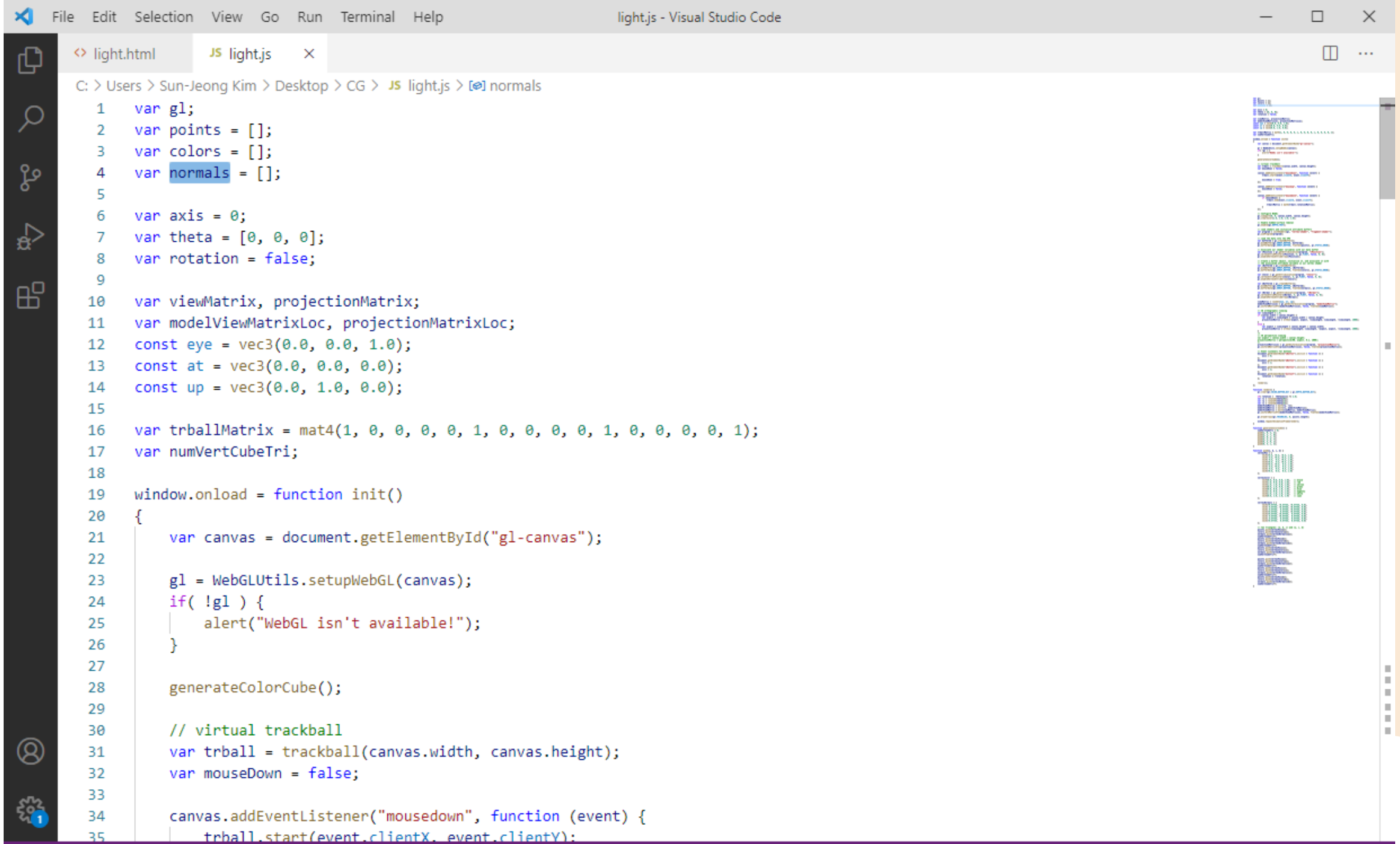
Normal Vectors

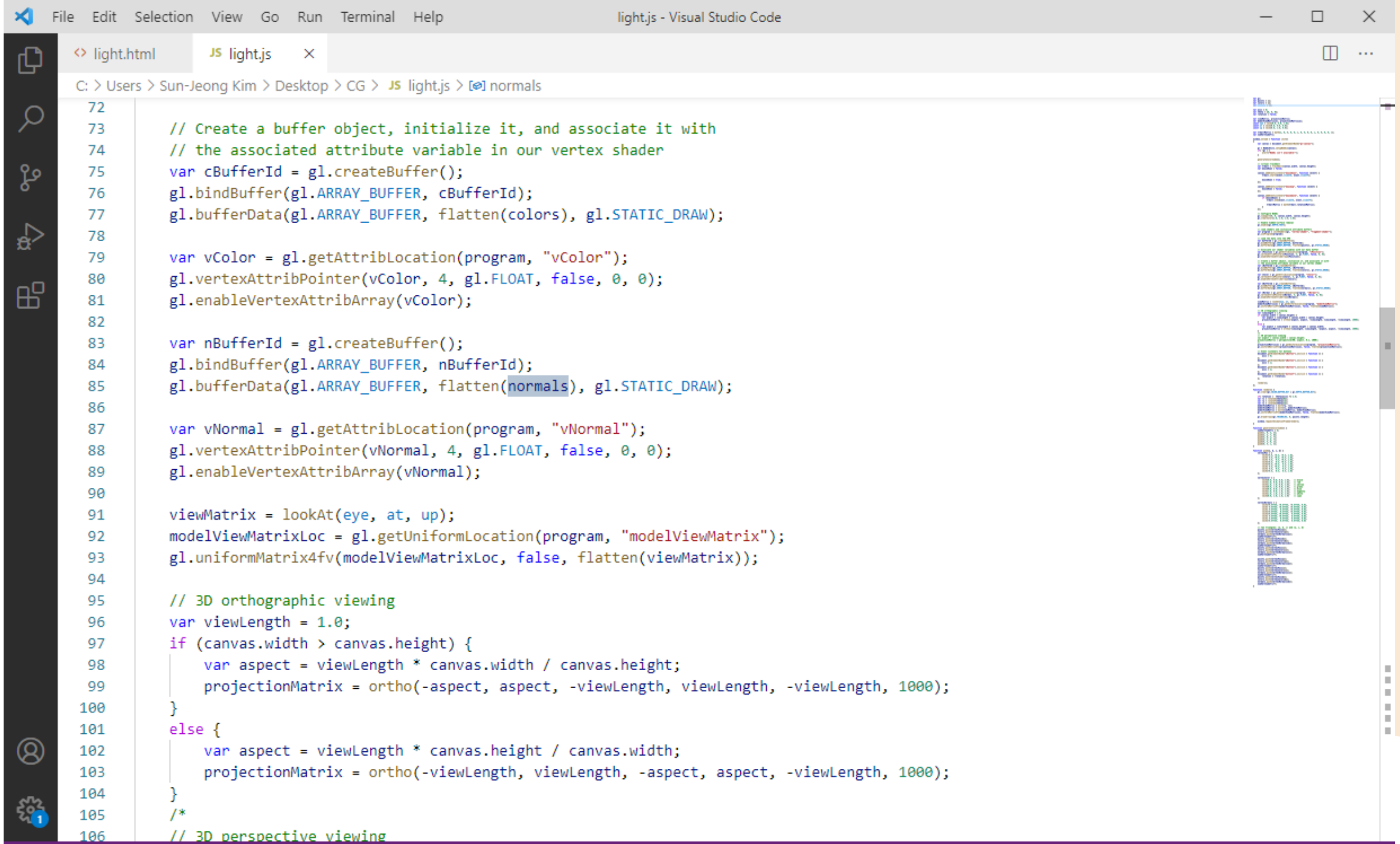
File Edit Selection View Go Run Terminal Help light.html - Visual Studio Code

light.html | light.js

C: > Users > Sun-Jeong Kim > Desktop > CG > light.html > html > head > script#vertex-shader

```
<!DOCTYPE html>
<html>
    <head>
        <title>학번 이름</title>
        <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;
        attribute vec4 vNormal;
        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;
```


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


File Edit Selection View Go Run Terminal Help
light.js - Visual Studio Code
light.html
light.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > normals
72
73 // Create a buffer object, initialize it, and associate it with
74 // the associated attribute variable in our vertex shader
75 var cBufferId = gl.createBuffer();
76 gl.bindBuffer(gl.ARRAY_BUFFER, cBufferId);
77 gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
78
79 var vColor = gl.getAttribLocation(program, "vColor");
80 gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
81 gl.enableVertexAttribArray(vColor);
82
83 var nBufferId = gl.createBuffer();
84 gl.bindBuffer(gl.ARRAY_BUFFER, nBufferId);
85 gl.bufferData(gl.ARRAY_BUFFER, flatten(normals), gl.STATIC_DRAW);
86
87 var vNormal = gl.getAttribLocation(program, "vNormal");
88 gl.vertexAttribPointer(vNormal, 4, gl.FLOAT, false, 0, 0);
89 gl.enableVertexAttribArray(vNormal);
90
91 viewMatrix = lookAt(eye, at, up);
92 modelViewMatrixLoc = gl.getUniformLocation(program, "modelViewMatrix");
93 gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(viewMatrix));
94
95 // 3D orthographic viewing
96 var viewLength = 1.0;
97 if (canvas.width > canvas.height) {
98 var aspect = viewLength * canvas.width / canvas.height;
99 projectionMatrix = ortho(-aspect, aspect, -viewLength, viewLength, -viewLength, 1000);
100 }
101 else {
102 var aspect = viewLength * canvas.height / canvas.width;
103 projectionMatrix = ortho(-viewLength, viewLength, -aspect, aspect, -viewLength, 1000);
104 }
105 /*
106 // 3D perspective viewing
0 0
Ln 4, Col 12 (7 selected) Spaces: 4 UTF-8 CRLF JavaScript

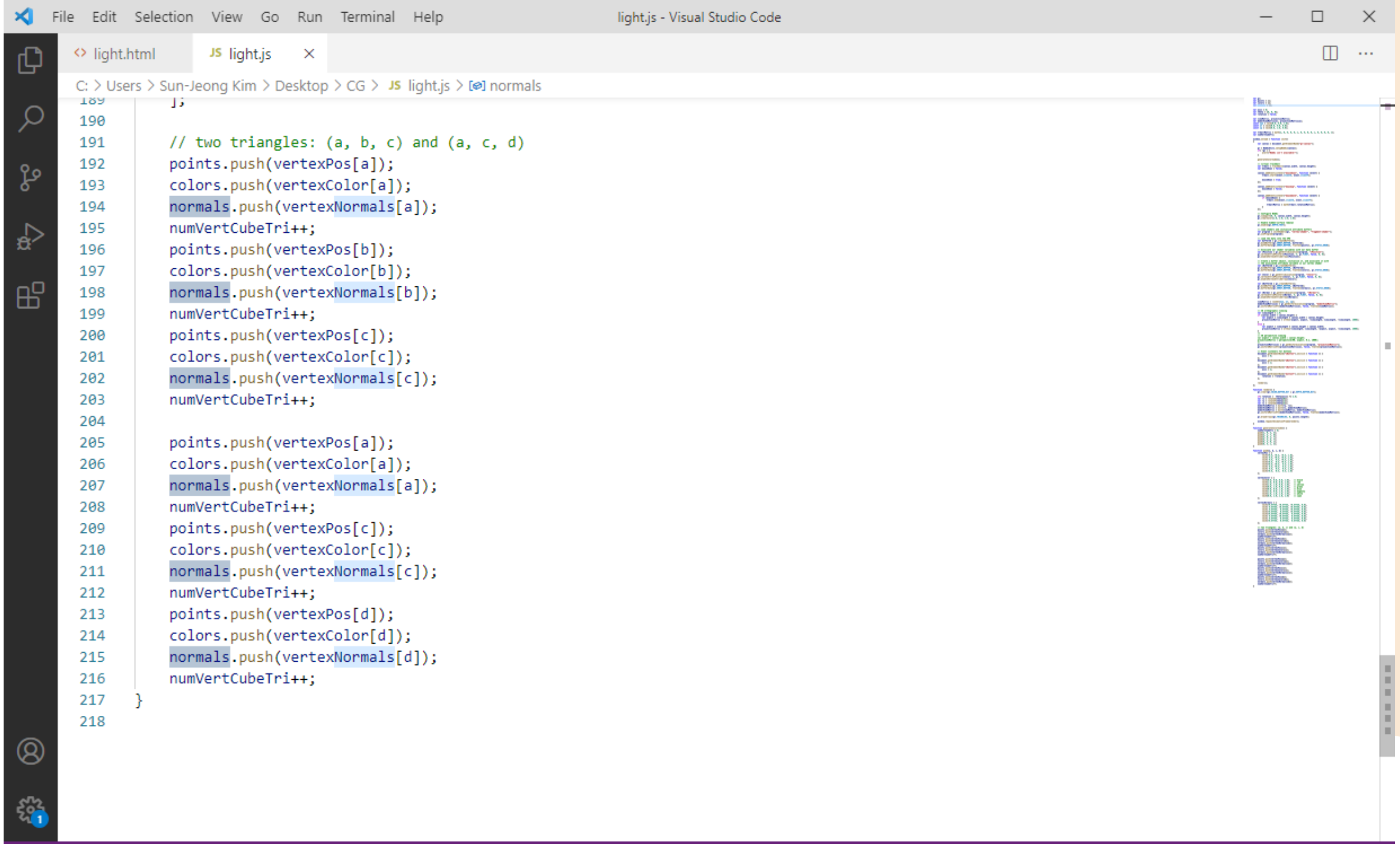
File Edit Selection View Go Run Terminal Help light.js - Visual Studio Code

light.html | light.js

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

```
156
157 function quad(a, b, c, d) {
158     vertexPos = [
159         vec4(-0.5, -0.5, -0.5, 1.0),
160         vec4( 0.5, -0.5, -0.5, 1.0),
161         vec4( 0.5,  0.5, -0.5, 1.0),
162         vec4(-0.5,  0.5, -0.5, 1.0),
163         vec4(-0.5, -0.5,  0.5, 1.0),
164         vec4( 0.5, -0.5,  0.5, 1.0),
165         vec4( 0.5,  0.5,  0.5, 1.0),
166         vec4(-0.5,  0.5,  0.5, 1.0)
167     ];
168
169     vertexColor = [
170         vec4(0.0, 0.0, 0.0, 1.0),   // black
171         vec4(1.0, 0.0, 0.0, 1.0),   // red
172         vec4(1.0, 1.0, 0.0, 1.0),   // yellow
173         vec4(0.0, 1.0, 0.0, 1.0),   // green
174         vec4(0.0, 0.0, 1.0, 1.0),   // blue
175         vec4(1.0, 0.0, 1.0, 1.0),   // magenta
176         vec4(1.0, 1.0, 1.0, 1.0),   // white
177         vec4(0.0, 1.0, 1.0, 1.0)    // cyan
178     ];
179
180     vertexNormals = [
181         vec4(-0.57735, -0.57735, -0.57735, 0.0),
182         vec4( 0.57735, -0.57735, -0.57735, 0.0),
183         vec4( 0.57735,  0.57735, -0.57735, 0.0),
184         vec4(-0.57735,  0.57735, -0.57735, 0.0),
185         vec4(-0.57735, -0.57735,  0.57735, 0.0),
186         vec4( 0.57735, -0.57735,  0.57735, 0.0),
187         vec4( 0.57735,  0.57735,  0.57735, 0.0),
188         vec4(-0.57735,  0.57735,  0.57735, 0.0)
189     ];
190
```

⊗ 0 ⚠ 0 Ln 4, Col 12 (7 selected) Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
light.js - Visual Studio Code
light.html
light.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > normals
];
// two triangles: (a, b, c) and (a, c, d)
points.push(vertexPos[a]);
colors.push(vertexColor[a]);
normals.push(vertexNormals[a]);
numVertCubeTri++;
points.push(vertexPos[b]);
colors.push(vertexColor[b]);
normals.push(vertexNormals[b]);
numVertCubeTri++;
points.push(vertexPos[c]);
colors.push(vertexColor[c]);
normals.push(vertexNormals[c]);
numVertCubeTri++;
points.push(vertexPos[a]);
colors.push(vertexColor[a]);
normals.push(vertexNormals[a]);
numVertCubeTri++;
points.push(vertexPos[c]);
colors.push(vertexColor[c]);
normals.push(vertexNormals[c]);
numVertCubeTri++;
points.push(vertexPos[d]);
colors.push(vertexColor[d]);
normals.push(vertexNormals[d]);
numVertCubeTri++;
}
0 0
Ln 4, Col 12 (7 selected)
Spaces: 4
UTF-8
CRLF
JavaScript

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

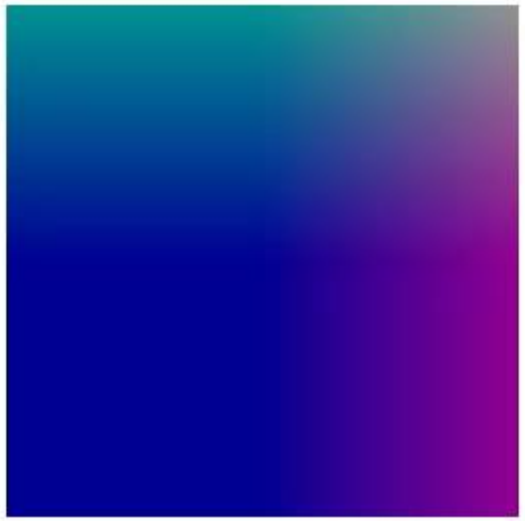
light.html | light.js

C: > Users > Sun-Jeong Kim > Desktop > CG > light.html > html > head > script#vertex-shader

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

        void main() {
            gl_Position = projectionMatrix * modelViewMatrix * vPosition;
            //fColor = vColor;
            fColor = vec4(vNormal.xyz, 1.0);    // Check normal vectors
        }
        </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="light.js"></script>
    </head>
    <body>
```

학번 이름

파일 | C:/Users/Sun-Jeong%20Kim...



Rotate X Rotate Y Rotate Z Toggle Rotate

0 0 Ln 16, Col 34 (7 selected) Spaces: 4 UTF-8 CRLF HTML

Ambient Reflection

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

light.html | light.js

C: > Users > Sun-Jeong Kim > Desktop > CG > light.html > html > head > script#vertex-shader

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

        uniform vec4 lightAmbient;

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

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

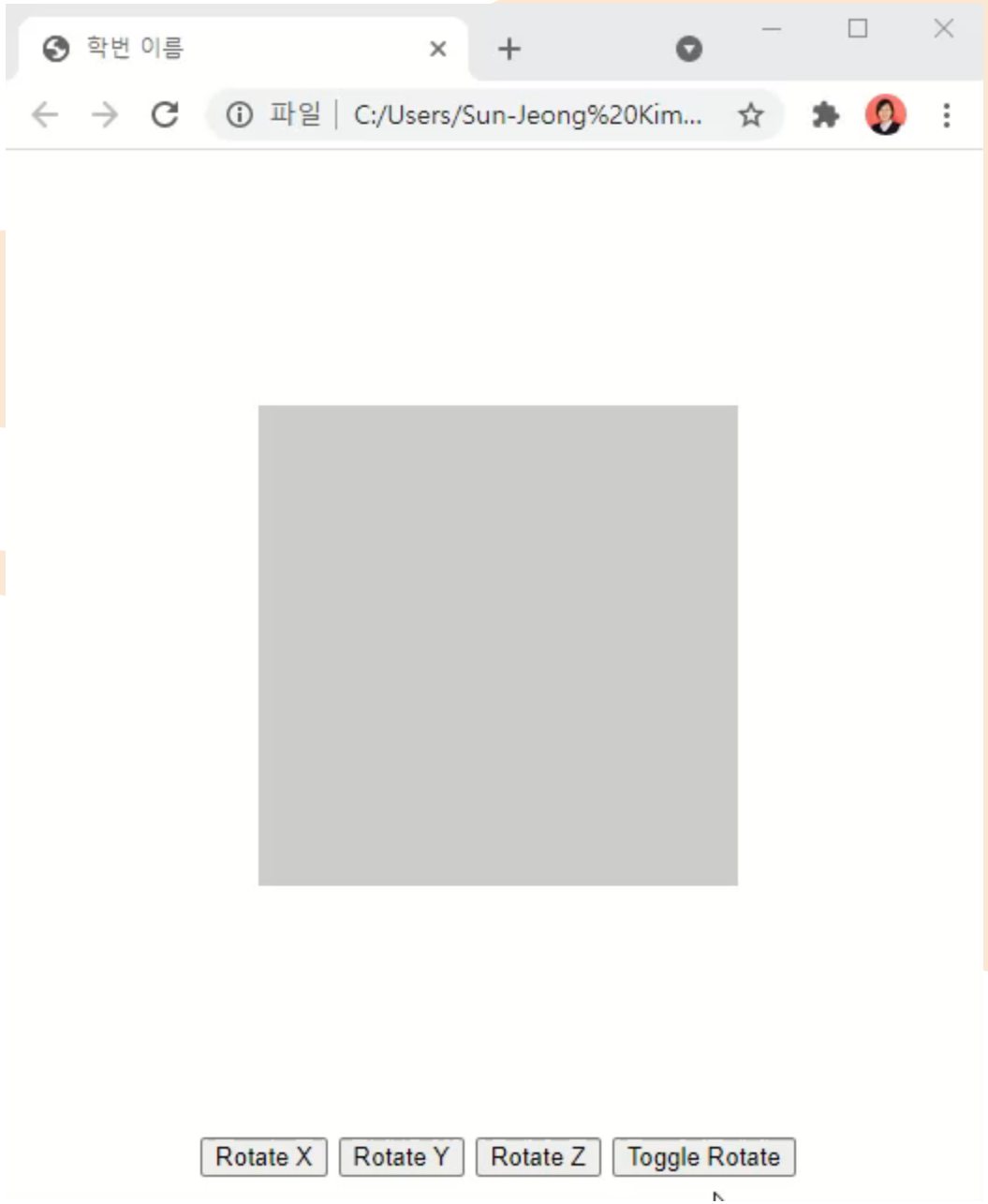
File Edit Selection View Go Run Terminal Help light.js - Visual Studio Code

light.html | light.js

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

```
126
127     setLighting(program);
128
129     render();
130 };
131
132 function render() {
133     gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
134
135     if( rotation )  theta[axis] += 2.0;
136     var rx = rotateX(theta[0]);
137     var ry = rotateY(theta[1]);
138     var rz = rotateZ(theta[2]);
139     modelViewMatrix = mult(ry, rx);
140     modelViewMatrix = mult(rz, modelViewMatrix);
141     modelViewMatrix = mult(viewMatrix, modelViewMatrix);
142     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
143
144     gl.drawArrays(gl.TRIANGLES, 0, points.length);
145
146     window.requestAnimationFrame(render);
147 }
148
149 function setLighting(program) {
150     var lightAmbient = [0.8, 0.8, 0.8, 1.0];
151
152     var lightAmbientLoc = gl.getUniformLocation(program, "lightAmbient");
153     gl.uniform4fv(lightAmbientLoc, lightAmbient);
154 }
155
156 function generateColorCube() {
157     numVertCubeTri = 0;
158     quad(1, 0, 3, 2);
159     quad(2, 3, 7, 6);
```

⊗ 0 ⚠ 0 Ln 127, Col 16 (11 selected) Spaces: 4 UTF-8 CRLF JavaScript


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

Materials


File Edit Selection View Go Run Terminal Help
light.html - Visual Studio Code
light.html
light.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.html > html > head > script#vertex-shader
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>학번 이름</title>
5 <script id="vertex-shader" type="x-shader/x-vertex">
6 attribute vec4 vPosition;
7 attribute vec4 vNormal;
8 attribute vec4 vColor;
9 uniform mat4 modelViewMatrix;
10 uniform mat4 projectionMatrix;
11 varying vec4 fColor;
12
13 uniform vec4 lightAmbient;
14 uniform vec4 matAmbient;
15
16 void main() {
17 gl_Position = projectionMatrix * modelViewMatrix * vPosition;
18 //fColor = vColor;
19 fColor = lightAmbient * matAmbient;
20 }
21 </script>
22
23 <script id="fragment-shader" type="x-shader/x-fragment">
24 precision mediump float;

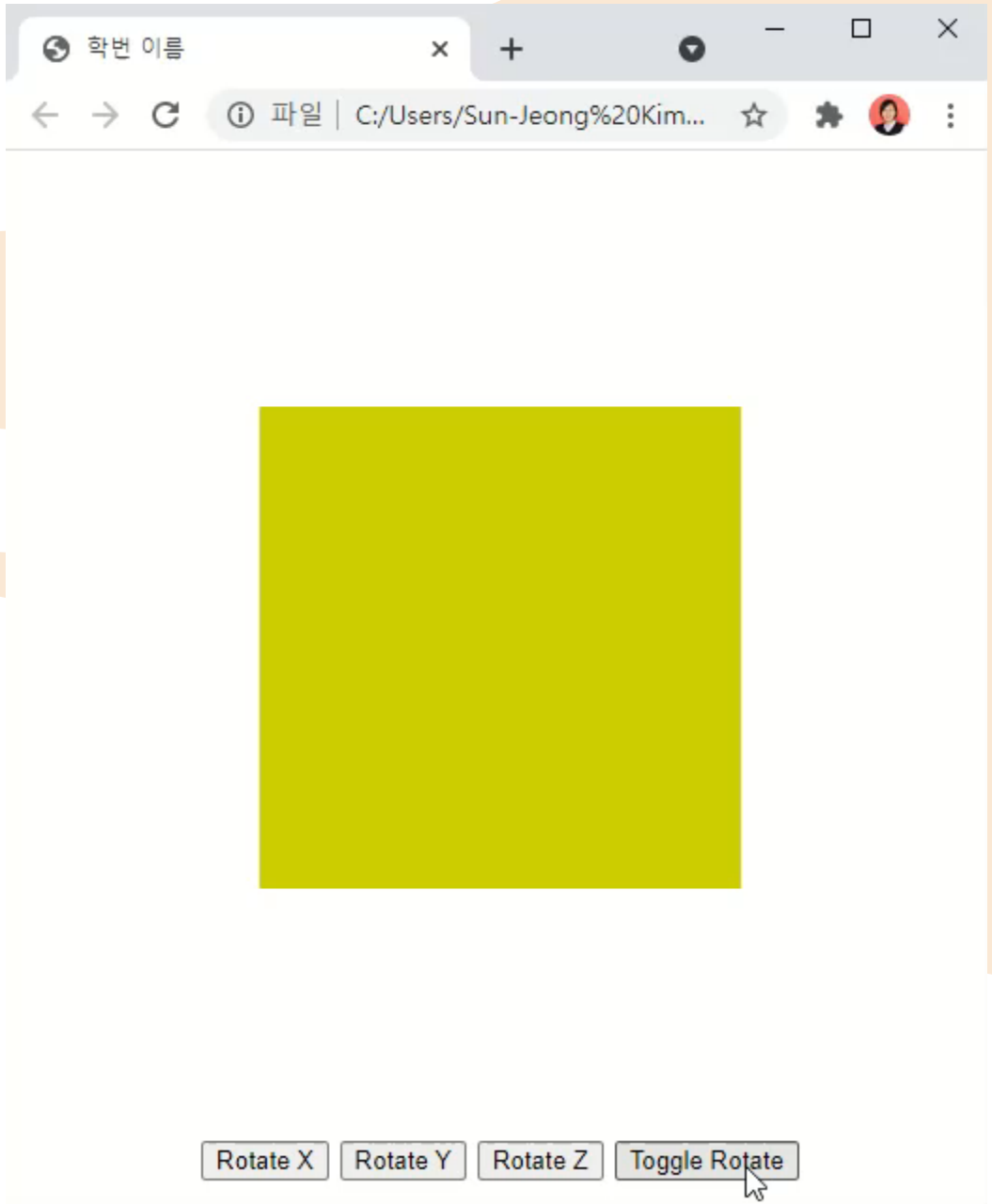
File Edit Selection View Go Run Terminal Help light.js - Visual Studio Code

light.html | light.js

C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > setLighting > matAmbient

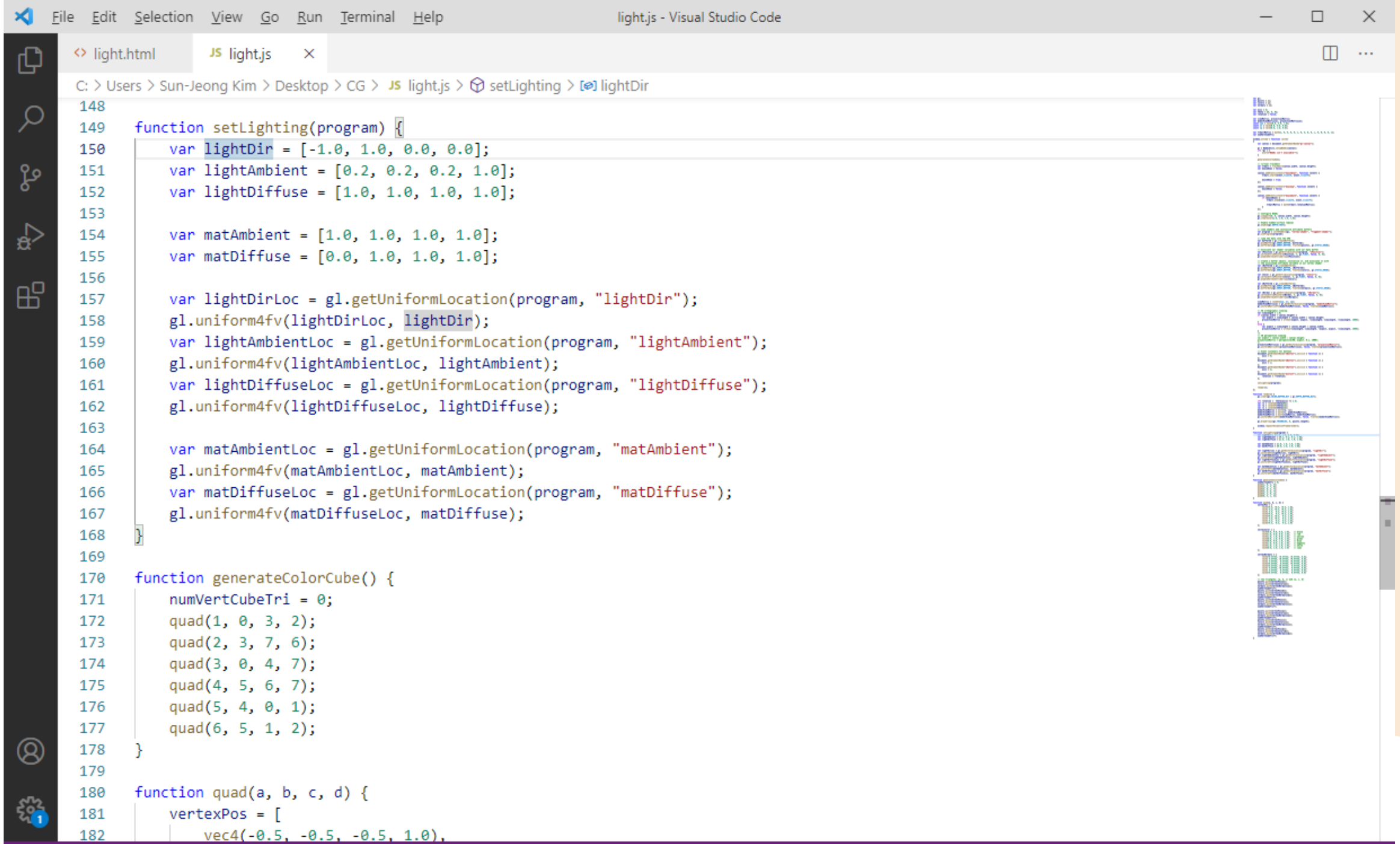
```
126
127     setLighting(program);
128
129     render();
130 };
131
132 function render() {
133     gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
134
135     if( rotation )  theta[axis] += 2.0;
136     var rx = rotateX(theta[0]);
137     var ry = rotateY(theta[1]);
138     var rz = rotateZ(theta[2]);
139     modelViewMatrix = mult(ry, rx);
140     modelViewMatrix = mult(rz, modelViewMatrix);
141     modelViewMatrix = mult(viewMatrix, modelViewMatrix);
142     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
143
144     gl.drawArrays(gl.TRIANGLES, 0, points.length);
145
146     window.requestAnimationFrame(render);
147 }
148
149 function setLighting(program) {
150     var lightAmbient = [0.8, 0.8, 0.8, 1.0];
151     var matAmbient = [1.0, 1.0, 0.0, 1.0];
152
153     var lightAmbientLoc = gl.getUniformLocation(program, "lightAmbient");
154     gl.uniform4fv(lightAmbientLoc, lightAmbient);
155     var matAmbientLoc = gl.getUniformLocation(program, "matAmbient");
156     gl.uniform4fv(matAmbientLoc, matAmbient);
157 }
158
159 function generateColorCube() {
160     numVertCubeTri = 0;
```

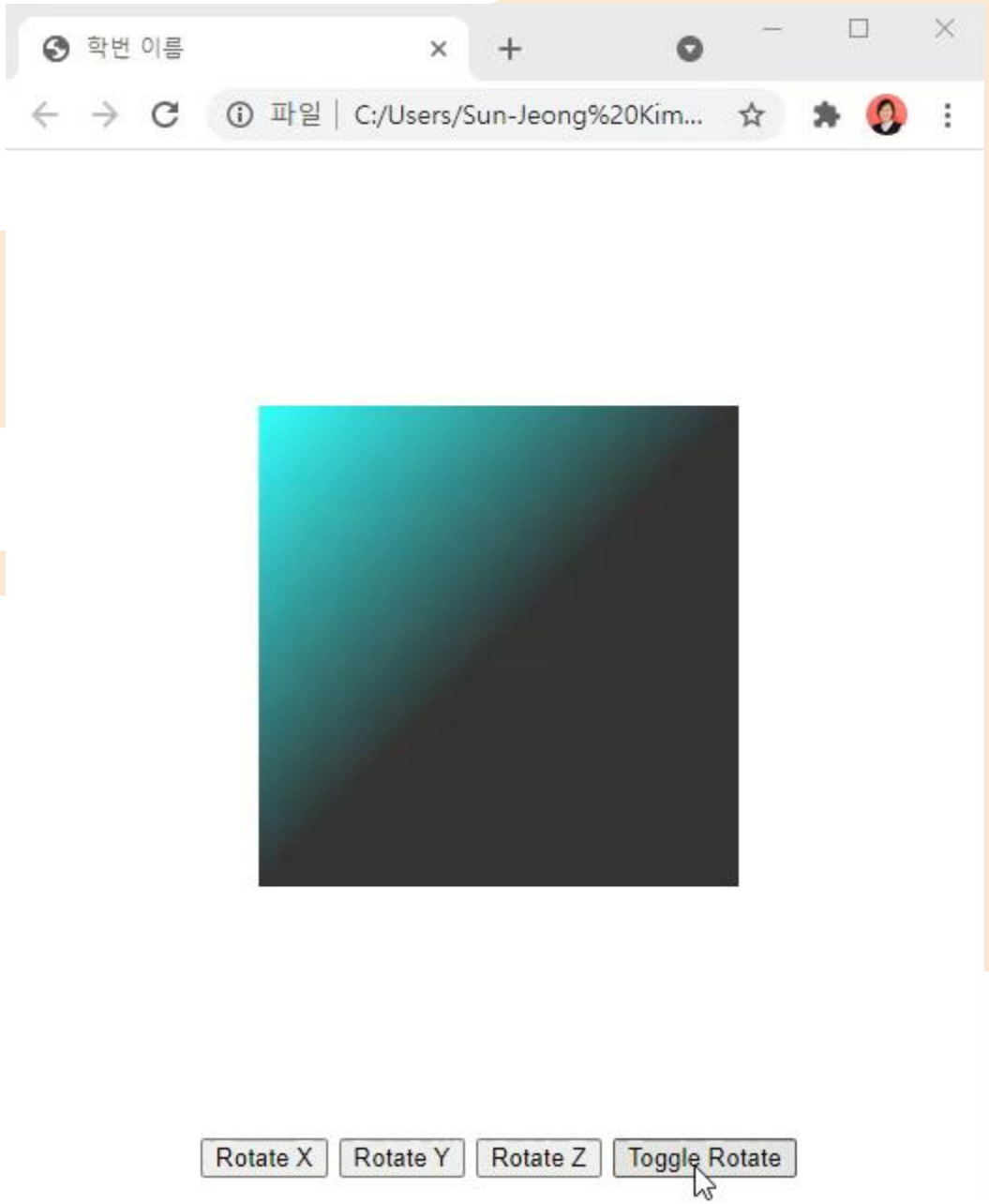
0 0 | Ln 151, Col 19 (10 selected) Spaces: 4 UTF-8 CRLF JavaScript


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

Diffuse Reflection

```
<script id="vertex-shader" type="x-shader/x-vertex">
attribute vec4 vPosition;
attribute vec4 vNormal;
attribute vec4 vColor;
uniform mat4 modelViewMatrix;
uniform mat4 projectionMatrix;
varying vec4 fColor;

uniform vec4 lightDir, lightAmbient, lightDiffuse;
uniform vec4 matAmbient, matDiffuse;

void main() {
    gl_Position = projectionMatrix * modelViewMatrix * vPosition;
    //fColor = vColor;
    vec4 ambient = lightAmbient * matAmbient;

    vec3 N = normalize(modelViewMatrix * vNormal).xyz;
    vec3 L = normalize(lightDir.xyz);
    float kd = max(dot(L, N), 0.0);
    vec4 diffuse = kd * lightDiffuse * matDiffuse;

    fColor = ambient + diffuse;
    fColor.a = 1.0;
}
</script>

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


File Edit Selection View Go Run Terminal Help
light.js - Visual Studio Code
light.html
light.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > setLighting > lightDir
148
149 function setLighting(program) {
150 var lightDir = [-1.0, 1.0, 0.0, 0.0];
151 var lightAmbient = [0.2, 0.2, 0.2, 1.0];
152 var lightDiffuse = [1.0, 1.0, 1.0, 1.0];
153
154 var matAmbient = [1.0, 1.0, 1.0, 1.0];
155 var matDiffuse = [0.0, 1.0, 1.0, 1.0];
156
157 var lightDirLoc = gl.getUniformLocation(program, "lightDir");
158 gl.uniform4fv(lightDirLoc, lightDir);
159 var lightAmbientLoc = gl.getUniformLocation(program, "lightAmbient");
160 gl.uniform4fv(lightAmbientLoc, lightAmbient);
161 var lightDiffuseLoc = gl.getUniformLocation(program, "lightDiffuse");
162 gl.uniform4fv(lightDiffuseLoc, lightDiffuse);
163
164 var matAmbientLoc = gl.getUniformLocation(program, "matAmbient");
165 gl.uniform4fv(matAmbientLoc, matAmbient);
166 var matDiffuseLoc = gl.getUniformLocation(program, "matDiffuse");
167 gl.uniform4fv(matDiffuseLoc, matDiffuse);
168 }
169
170 function generateColorCube() {
171 numVertCubeTri = 0;
172 quad(1, 0, 3, 2);
173 quad(2, 3, 7, 6);
174 quad(3, 0, 4, 7);
175 quad(4, 5, 6, 7);
176 quad(5, 4, 0, 1);
177 quad(6, 5, 1, 2);
178 }
179
180 function quad(a, b, c, d) {
181 vertexPos = [
182 vec4(-0.5, -0.5, -0.5, 1.0),
0 0
Ln 150, Col 17 Spaces: 4 UTF-8 CRLF JavaScript


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

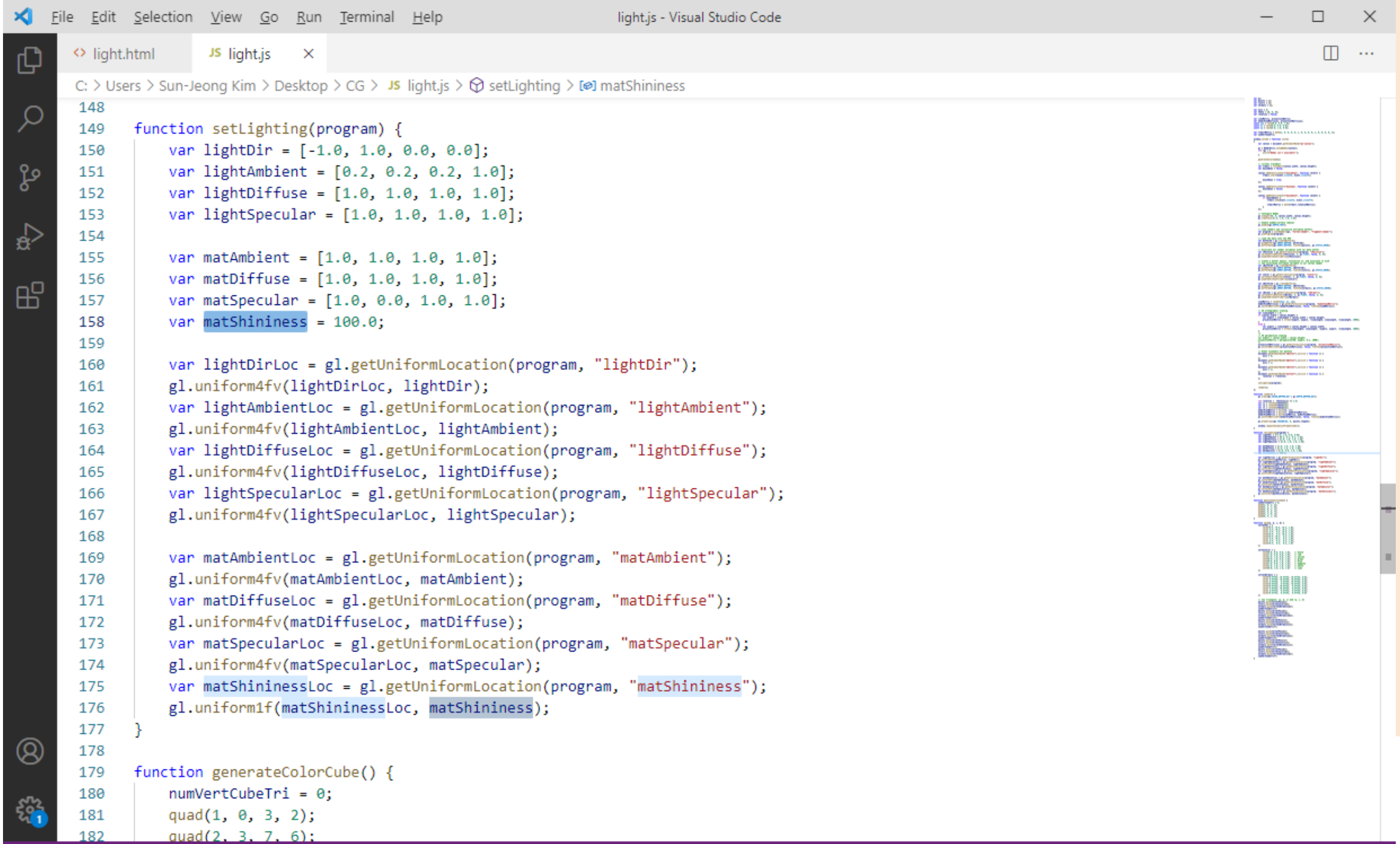
Specular Reflection

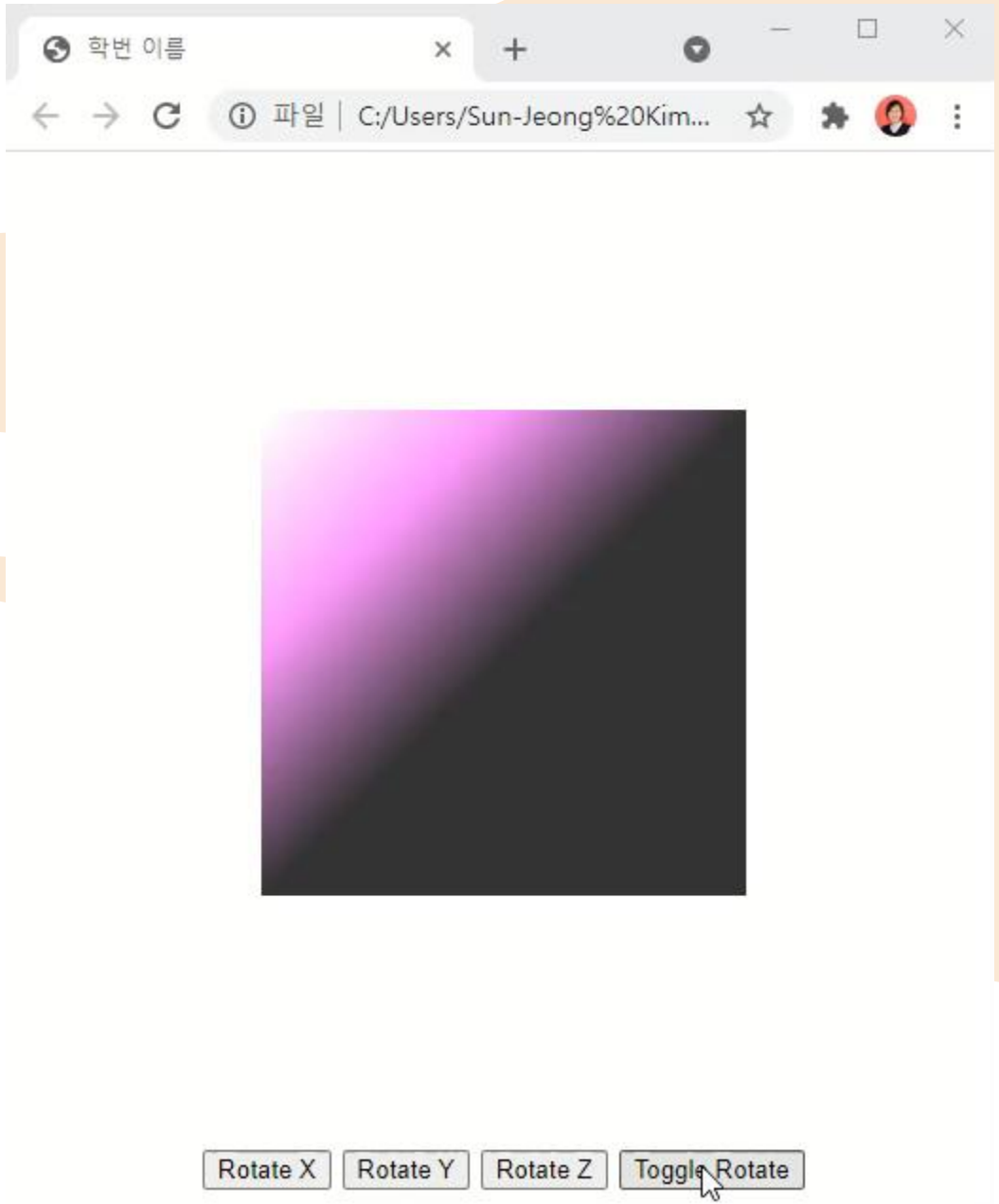
Edit Selection View Go Run Terminal Help light.html - Visual Studio Code

light.html × light.js

C: > Users > Sun-Jeong Kim > Desktop > CG > light.html > html > head > script#vertex-shader

```
<script id="vertex-shader" type="x-shader/x-vertex">
attribute vec4 vPosition;
attribute vec4 vNormal;
attribute vec4 vColor;
uniform mat4 modelViewMatrix;
uniform mat4 projectionMatrix;
varying vec4 fColor;

uniform vec4 lightDir, lightAmbient, lightDiffuse, lightSpecular;
uniform vec4 matAmbient, matDiffuse, matSpecular;
uniform float matShininess;

void main() {
    gl_Position = projectionMatrix * modelViewMatrix * vPosition;
    //fColor = vColor;
    vec4 ambient = lightAmbient * matAmbient;

    vec3 N = normalize(modelViewMatrix * vNormal).xyz;
    vec3 L = normalize(lightDir.xyz);
    float kd = max(dot(L, N), 0.0);
    vec4 diffuse = kd * lightDiffuse * matDiffuse;

    vec3 V = normalize(modelViewMatrix*vPosition).xyz;      // origin: camera position
    vec3 H = normalize(L - V).xyz;
    float ks = pow(max(dot(N, H), 0.0), matShininess);
    vec4 specular = ks * lightSpecular * matSpecular;

    if (dot(L, N) < 0.0)    specular = vec4(0.0, 0.0, 0.0, 1.0);

    fColor = ambient + diffuse + specular;
    fColor.a = 1.0;
}
</script>

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


File Edit Selection View Go Run Terminal Help
light.js - Visual Studio Code
light.html
light.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > setLighting > matShininess
148
149 function setLighting(program) {
150 var lightDir = [-1.0, 1.0, 0.0, 0.0];
151 var lightAmbient = [0.2, 0.2, 0.2, 1.0];
152 var lightDiffuse = [1.0, 1.0, 1.0, 1.0];
153 var lightSpecular = [1.0, 1.0, 1.0, 1.0];
154
155 var matAmbient = [1.0, 1.0, 1.0, 1.0];
156 var matDiffuse = [1.0, 1.0, 1.0, 1.0];
157 var matSpecular = [1.0, 0.0, 1.0, 1.0];
158 var matShininess = 100.0;
159
160 var lightDirLoc = gl.getUniformLocation(program, "lightDir");
161 gl.uniform4fv(lightDirLoc, lightDir);
162 var lightAmbientLoc = gl.getUniformLocation(program, "lightAmbient");
163 gl.uniform4fv(lightAmbientLoc, lightAmbient);
164 var lightDiffuseLoc = gl.getUniformLocation(program, "lightDiffuse");
165 gl.uniform4fv(lightDiffuseLoc, lightDiffuse);
166 var lightSpecularLoc = gl.getUniformLocation(program, "lightSpecular");
167 gl.uniform4fv(lightSpecularLoc, lightSpecular);
168
169 var matAmbientLoc = gl.getUniformLocation(program, "matAmbient");
170 gl.uniform4fv(matAmbientLoc, matAmbient);
171 var matDiffuseLoc = gl.getUniformLocation(program, "matDiffuse");
172 gl.uniform4fv(matDiffuseLoc, matDiffuse);
173 var matSpecularLoc = gl.getUniformLocation(program, "matSpecular");
174 gl.uniform4fv(matSpecularLoc, matSpecular);
175 var matShininessLoc = gl.getUniformLocation(program, "matShininess");
176 gl.uniform1f(matShininessLoc, matShininess);
177 }
178
179 function generateColorCube() {
180 numVertCubeTri = 0;
181 quad(1, 0, 3, 2);
182 quad(2, 3, 7, 6);
⊗ 0 ⚠ 0
Ln 158, Col 21 (12 selected)
Spaces: 4
UTF-8
CRLF
JavaScript


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

연습 문제 (1)

- light.js에 있는 setLighting( ) 함수에서,
  - lightDir의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) var lightDir = vec4(1.0, 0.0, 0.0, 0.0);
  - lightAmbient, lightDiffuse, lightSpecular의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) var lightAmbient = vec4(1.0, 0.0, 0.0, 1.0);
      var lightDiffuse = vec4(0.0, 1.0, 0.0, 1.0);
      var lightSpecular = vec4(0.0, 0.0, 1.0, 1.0);
  - matShininess의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) var matShininess = 20.0

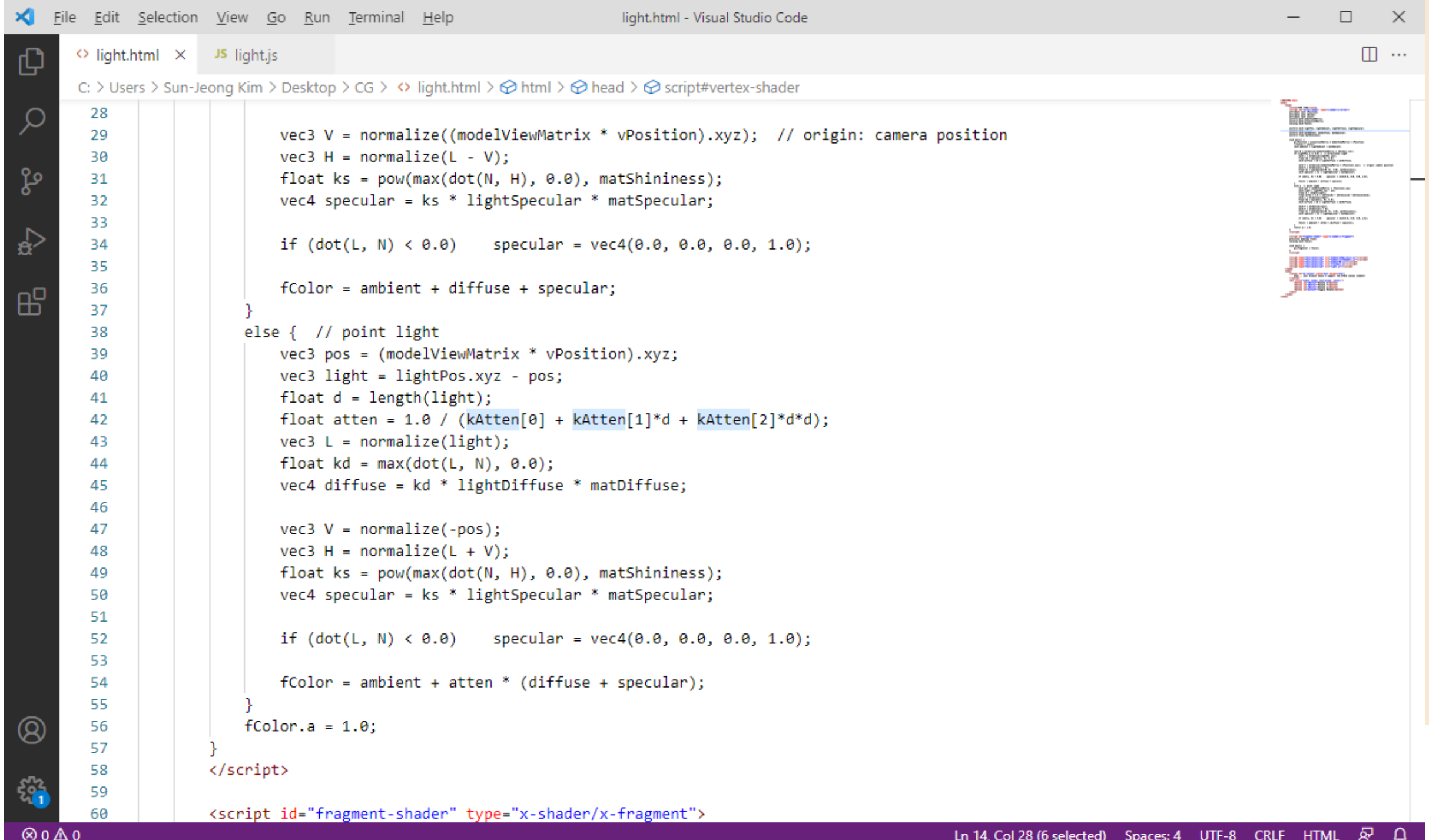
Point Light

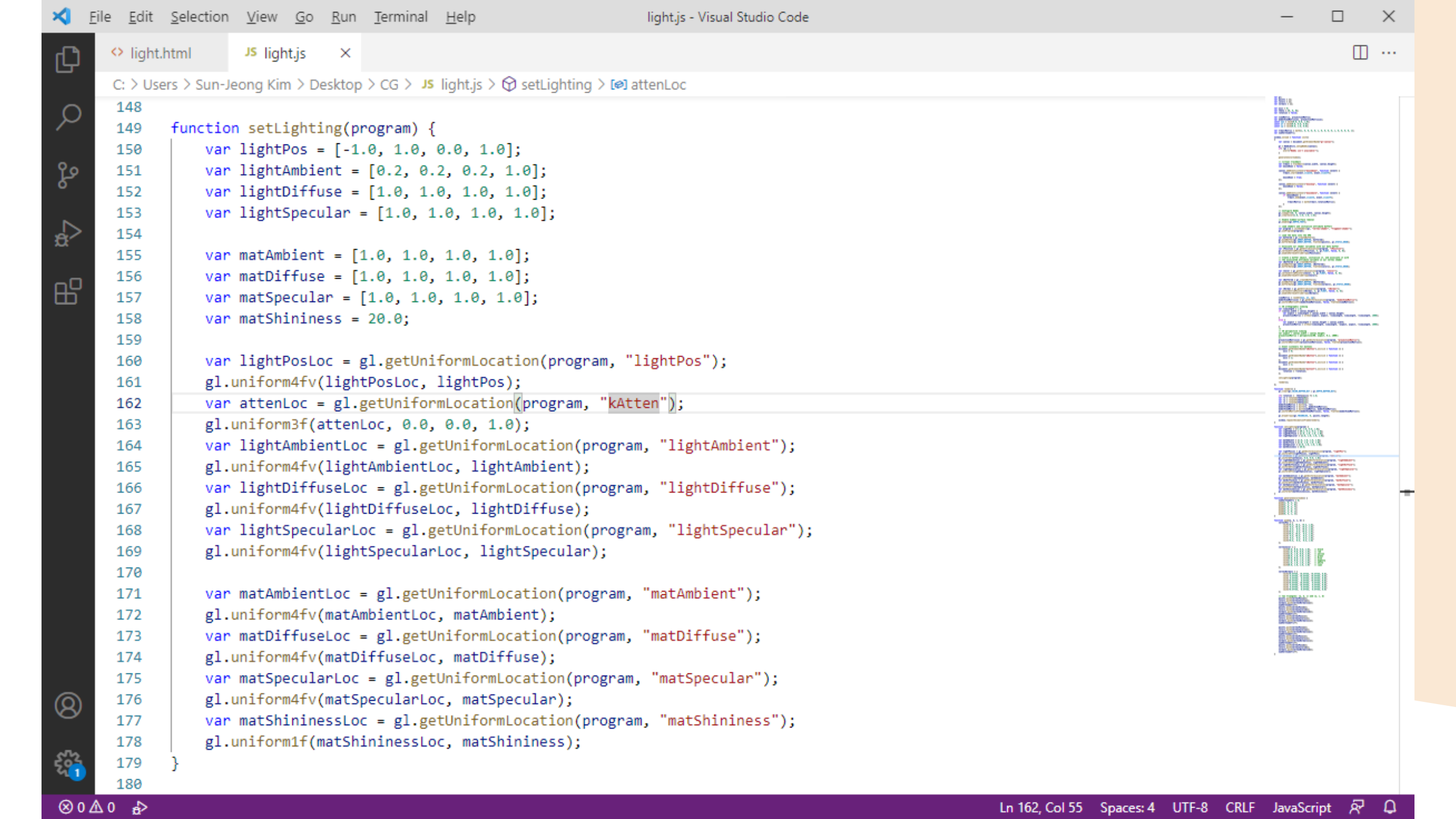
Edit Selection View Go Run Terminal Help light.html - Visual Studio Code

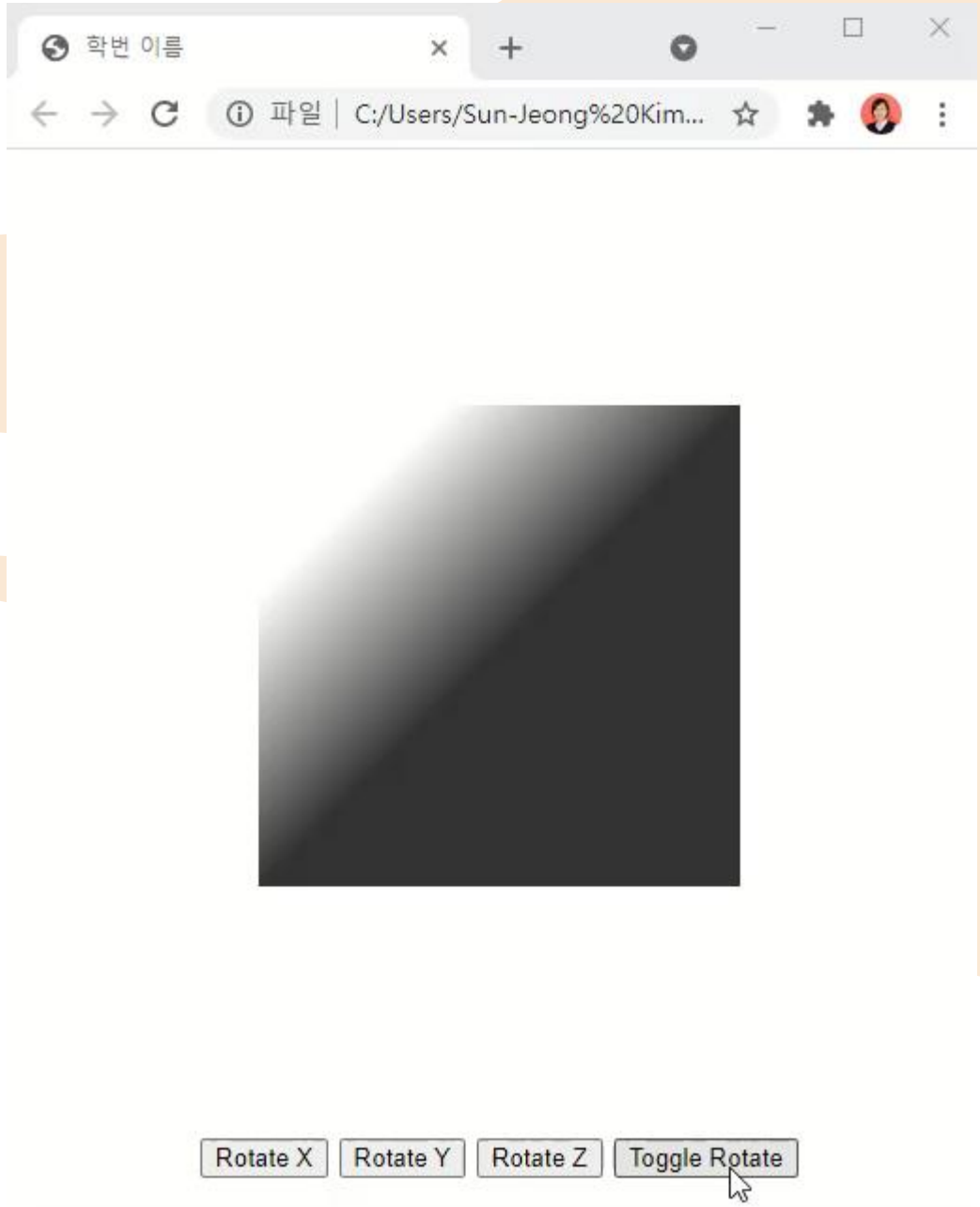
light.html × light.js

C: > Users > Sun-Jeong Kim > Desktop > CG > light.html > html > head > script#vertex-shader

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

        uniform vec4 lightPos, lightAmbient, lightDiffuse, lightSpecular;
        uniform vec3 kAtten;
        uniform vec4 matAmbient, matDiffuse, matSpecular;
        uniform float matShininess;

        void main() {
            gl_Position = projectionMatrix * modelViewMatrix * vPosition;
            //fColor = vColor;
            vec4 ambient = lightAmbient * matAmbient;

            vec3 N = normalize((modelViewMatrix * vNormal).xyz);
            if (lightPos.w == 0.0) {  // directional light
                vec3 L = normalize(lightPos.xyz);
                float kd = max(dot(L, N), 0.0);
                vec4 diffuse = kd * lightDiffuse * matDiffuse;

                vec3 V = normalize((modelViewMatrix * vPosition).xyz);  // origin: camera
                vec3 H = normalize(L - V);
                float ks = pow(max(dot(N, H), 0.0), matShininess);
                vec4 specular = ks * lightSpecular * matSpecular;
```


File Edit Selection View Go Run Terminal Help
light.html - Visual Studio Code
light.html
light.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.html > html > head > script#vertex-shader
28
29 vec3 V = normalize((modelViewMatrix * vPosition).xyz); // origin: camera position
30 vec3 H = normalize(L - V);
31 float ks = pow(max(dot(N, H), 0.0), matShininess);
32 vec4 specular = ks * lightSpecular * matSpecular;
33
34 if (dot(L, N) < 0.0) specular = vec4(0.0, 0.0, 0.0, 1.0);
35
36 fColor = ambient + diffuse + specular;
37 }
38 else { // point light
39 vec3 pos = (modelViewMatrix * vPosition).xyz;
40 vec3 light = lightPos.xyz - pos;
41 float d = length(light);
42 float atten = 1.0 / (kAtten[0] + kAtten[1]*d + kAtten[2]*d*d);
43 vec3 L = normalize(light);
44 float kd = max(dot(L, N), 0.0);
45 vec4 diffuse = kd * lightDiffuse * matDiffuse;
46
47 vec3 V = normalize(-pos);
48 vec3 H = normalize(L + V);
49 float ks = pow(max(dot(N, H), 0.0), matShininess);
50 vec4 specular = ks * lightSpecular * matSpecular;
51
52 if (dot(L, N) < 0.0) specular = vec4(0.0, 0.0, 0.0, 1.0);
53
54 fColor = ambient + atten * (diffuse + specular);
55 }
56 fColor.a = 1.0;
57 }
58 </script>
59
60 <script id="fragment-shader" type="x-shader/x-fragment">
0 0
Ln 14, Col 28 (6 selected) Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
light.js - Visual Studio Code
light.html
light.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > setLighting > attenLoc
148
149 function setLighting(program) {
150 var lightPos = [-1.0, 1.0, 0.0, 1.0];
151 var lightAmbient = [0.2, 0.2, 0.2, 1.0];
152 var lightDiffuse = [1.0, 1.0, 1.0, 1.0];
153 var lightSpecular = [1.0, 1.0, 1.0, 1.0];
154
155 var matAmbient = [1.0, 1.0, 1.0, 1.0];
156 var matDiffuse = [1.0, 1.0, 1.0, 1.0];
157 var matSpecular = [1.0, 1.0, 1.0, 1.0];
158 var matShininess = 20.0;
159
160 var lightPosLoc = gl.getUniformLocation(program, "lightPos");
161 gl.uniform4fv(lightPosLoc, lightPos);
162 var attenLoc = gl.getUniformLocation(program, "kAtten");
163 gl.uniform3f(attenLoc, 0.0, 0.0, 1.0);
164 var lightAmbientLoc = gl.getUniformLocation(program, "lightAmbient");
165 gl.uniform4fv(lightAmbientLoc, lightAmbient);
166 var lightDiffuseLoc = gl.getUniformLocation(program, "lightDiffuse");
167 gl.uniform4fv(lightDiffuseLoc, lightDiffuse);
168 var lightSpecularLoc = gl.getUniformLocation(program, "lightSpecular");
169 gl.uniform4fv(lightSpecularLoc, lightSpecular);
170
171 var matAmbientLoc = gl.getUniformLocation(program, "matAmbient");
172 gl.uniform4fv(matAmbientLoc, matAmbient);
173 var matDiffuseLoc = gl.getUniformLocation(program, "matDiffuse");
174 gl.uniform4fv(matDiffuseLoc, matDiffuse);
175 var matSpecularLoc = gl.getUniformLocation(program, "matSpecular");
176 gl.uniform4fv(matSpecularLoc, matSpecular);
177 var matShininessLoc = gl.getUniformLocation(program, "matShininess");
178 gl.uniform1f(matShininessLoc, matShininess);
179 }
180
Ln 162, Col 55 Spaces: 4 UTF-8 CRLF JavaScript


학번 이름
파일 | C:/Users/Sun-Jeong%20Kim...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

연습 문제 (2)

- light.js에 있는 setLighting( ) 함수에서,
  - lightPos의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) var lightPos = vec4(1.0, 0.0, 0.0, 1.0);
  - kAtten의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) gl.uniform3f(attenLoc, 1.0, 0.0, 0.0);

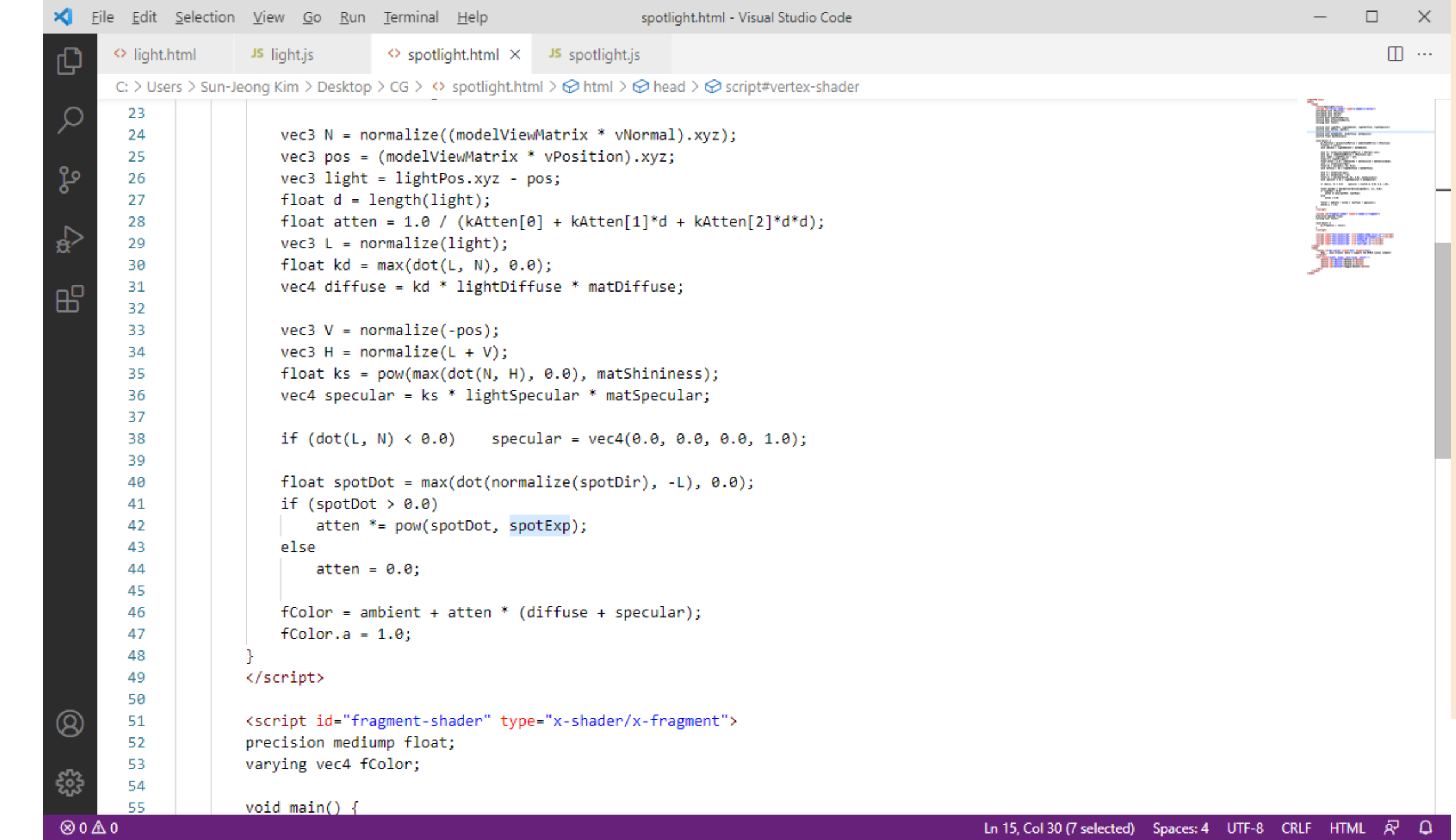
Spotlight

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

light.html | light.js | spotlight.html | spotlight.js

C: > Users > Sun-Jeong Kim > Desktop > CG > spotlight.html > html > head > script#vertex-shader

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

        uniform vec4 lightPos, lightAmbient, lightDiffuse, lightSpecular;
        uniform vec3 kAtten, spotDir;
        uniform float spotExp;
        uniform vec4 matAmbient, matDiffuse, matSpecular;
        uniform float matShininess;

        void main() {
            gl_Position = projectionMatrix * modelViewMatrix * vPosition;
            //fColor = vColor;
            vec4 ambient = lightAmbient * matAmbient;
```


File Edit Selection View Go Run Terminal Help
spotlight.html - Visual Studio Code
light.html
light.js
spotlight.html
spotlight.js
C: > Users > Sun-Jeong Kim > Desktop > CG > spotlight.html > html > head > script#vertex-shader
vec3 N = normalize((modelViewMatrix * vNormal).xyz);
vec3 pos = (modelViewMatrix * vPosition).xyz;
vec3 light = lightPos.xyz - pos;
float d = length(light);
float atten = 1.0 / (kAtten[0] + kAtten[1]*d + kAtten[2]*d*d);
vec3 L = normalize(light);
float kd = max(dot(L, N), 0.0);
vec4 diffuse = kd * lightDiffuse * matDiffuse;
vec3 V = normalize(-pos);
vec3 H = normalize(L + V);
float ks = pow(max(dot(N, H), 0.0), matShininess);
vec4 specular = ks * lightSpecular * matSpecular;
if (dot(L, N) < 0.0) specular = vec4(0.0, 0.0, 0.0, 1.0);
float spotDot = max(dot(normalize(spotDir), -L), 0.0);
if (spotDot > 0.0)
atten *= pow(spotDot, spotExp);
else
atten = 0.0;
fColor = ambient + atten * (diffuse + specular);
fColor.a = 1.0;
}
</script>
<script id="fragment-shader" type="x-shader/x-fragment">
precision mediump float;
varying vec4 fColor;
void main() {
0 0
Ln 15, Col 30 (7 selected)
Spaces: 4
UTF-8
CRLF
HTML

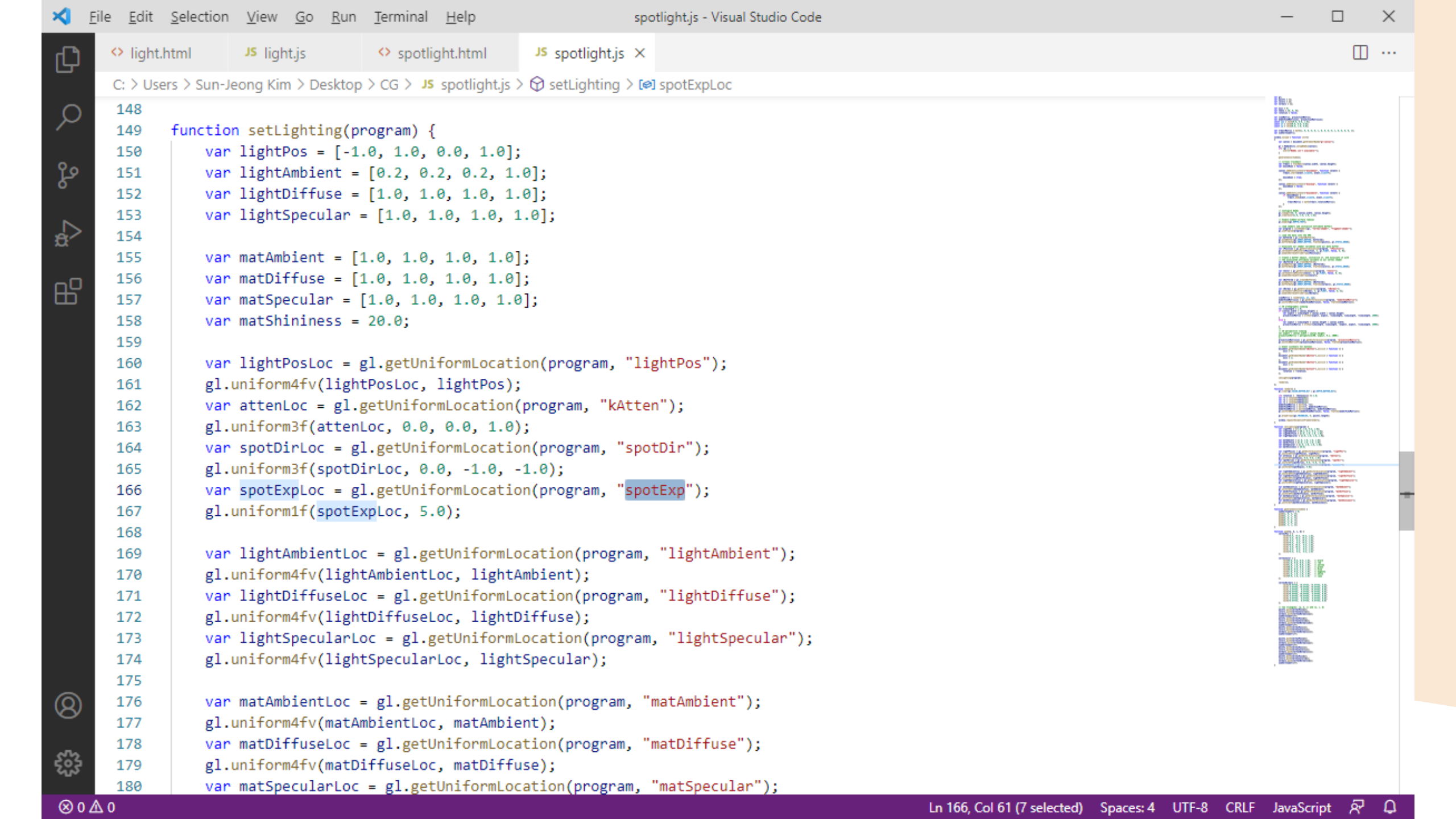
File Edit Selection View Go Run Terminal Help spotlight.html - Visual Studio Code

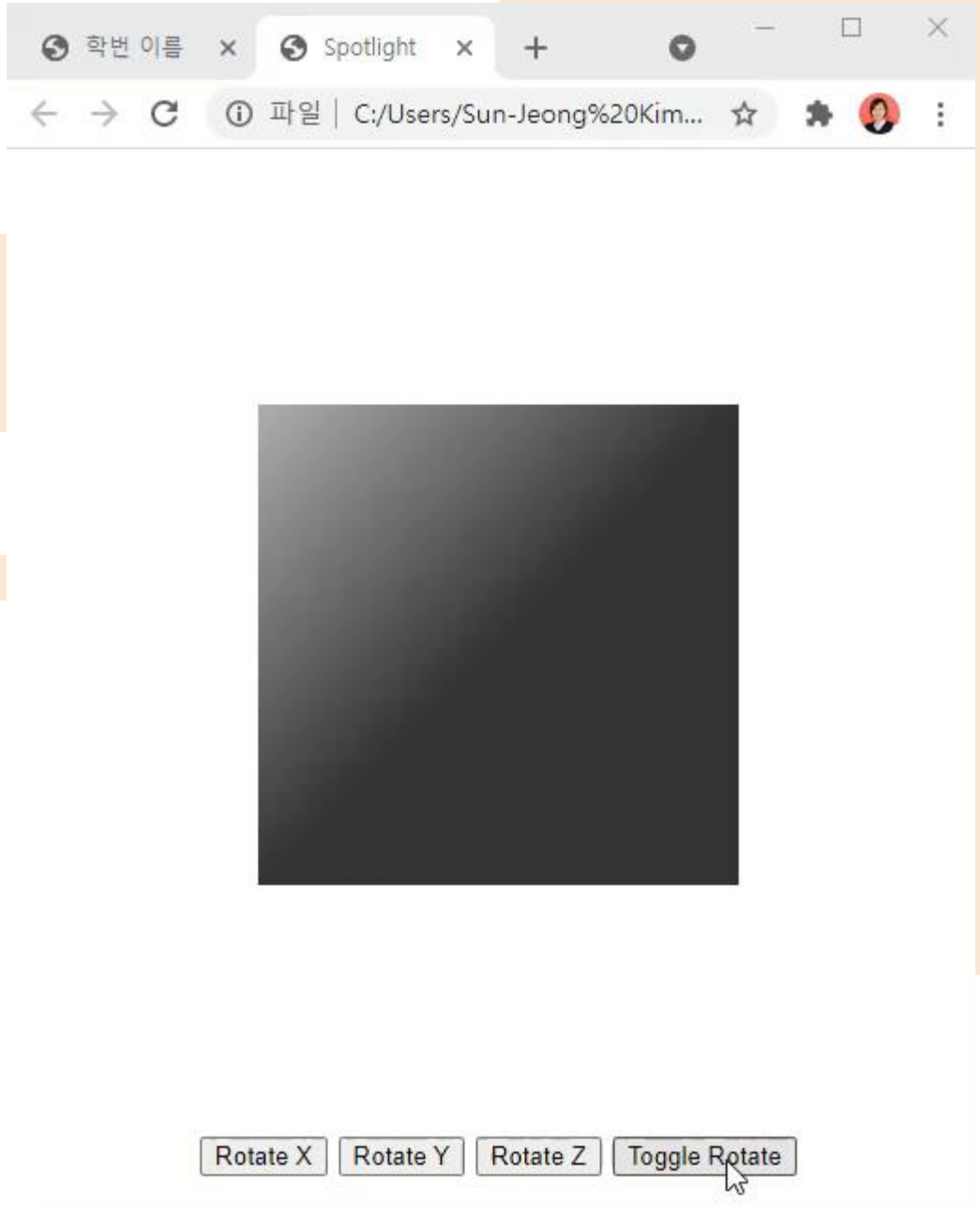
light.html | light.js | spotlight.html | spotlight.js

C: > Users > Sun-Jeong Kim > Desktop > CG > spotlight.html > html > head > script#vertex-shader

```
50
51          <script id="fragment-shader" type="x-shader/x-fragment">
52          precision mediump float;
53          varying vec4 fColor;
54
55          void main() {
56              gl_FragColor = fColor;
57          }
58          </script>
59
60          <script type="text/javascript" src="Common/webgl-utils.js"></script>
61          <script type="text/javascript" src="Common/initShaders.js"></script>
62          <script type="text/javascript" src="Common/MV.js"></script>
63          <script type="text/javascript" src="trackball.js"></script>
64          <script type="text/javascript" src="spotlight.js"></script>
65      </head>
66      <body>
67          <canvas id="gl-canvas" width="512" height="512">
68              Oops... your browser doesn't support the HTML5 canvas element!
69          </canvas>
70          <div style="width: 512px; text-align: center;">
71              <button id="xButton">Rotate X</button>
72              <button id="yButton">Rotate Y</button>
73              <button id="zButton">Rotate Z</button>
74              <button id="buttonT">Toggle Rotate</button>
75          </div>
76      </body>
77  </html>
```

⊗ 0 ⚠ 0 Ln 15, Col 30 (7 selected) Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
spotlight.js - Visual Studio Code
light.html
light.js
spotlight.html
spotlight.js
C: > Users > Sun-Jeong Kim > Desktop > CG > spotlight.js > setLighting > spotExpLoc
148
149 function setLighting(program) {
150 var lightPos = [-1.0, 1.0, 0.0, 1.0];
151 var lightAmbient = [0.2, 0.2, 0.2, 1.0];
152 var lightDiffuse = [1.0, 1.0, 1.0, 1.0];
153 var lightSpecular = [1.0, 1.0, 1.0, 1.0];
154
155 var matAmbient = [1.0, 1.0, 1.0, 1.0];
156 var matDiffuse = [1.0, 1.0, 1.0, 1.0];
157 var matSpecular = [1.0, 1.0, 1.0, 1.0];
158 var matShininess = 20.0;
159
160 var lightPosLoc = gl.getUniformLocation(program, "lightPos");
161 gl.uniform4fv(lightPosLoc, lightPos);
162 var attenLoc = gl.getUniformLocation(program, "kAtten");
163 gl.uniform3f(attenLoc, 0.0, 0.0, 1.0);
164 var spotDirLoc = gl.getUniformLocation(program, "spotDir");
165 gl.uniform3f(spotDirLoc, 0.0, -1.0, -1.0);
166 var spotExpLoc = gl.getUniformLocation(program, "spotExp");
167 gl.uniform1f(spotExpLoc, 5.0);
168
169 var lightAmbientLoc = gl.getUniformLocation(program, "lightAmbient");
170 gl.uniform4fv(lightAmbientLoc, lightAmbient);
171 var lightDiffuseLoc = gl.getUniformLocation(program, "lightDiffuse");
172 gl.uniform4fv(lightDiffuseLoc, lightDiffuse);
173 var lightSpecularLoc = gl.getUniformLocation(program, "lightSpecular");
174 gl.uniform4fv(lightSpecularLoc, lightSpecular);
175
176 var matAmbientLoc = gl.getUniformLocation(program, "matAmbient");
177 gl.uniform4fv(matAmbientLoc, matAmbient);
178 var matDiffuseLoc = gl.getUniformLocation(program, "matDiffuse");
179 gl.uniform4fv(matDiffuseLoc, matDiffuse);
180 var matSpecularLoc = gl.getUniformLocation(program, "matSpecular");
⊗ 0 ⚠ 0
Ln 166, Col 61 (7 selected)
Spaces: 4
UTF-8
CRLF
JavaScript

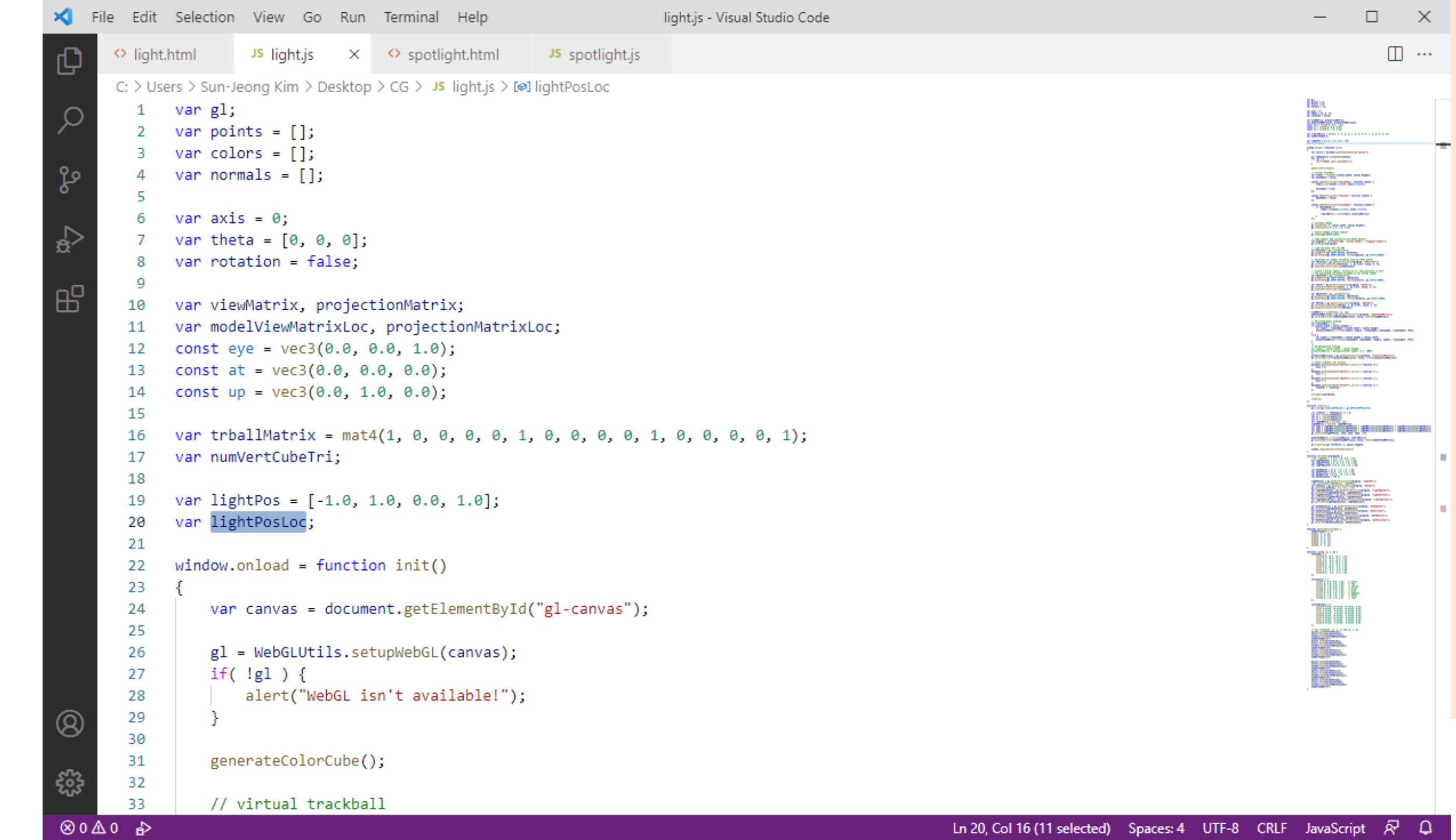

학번 이름
Spotlight
파일 | C:/Users/Sun-Jeong%20Kim...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

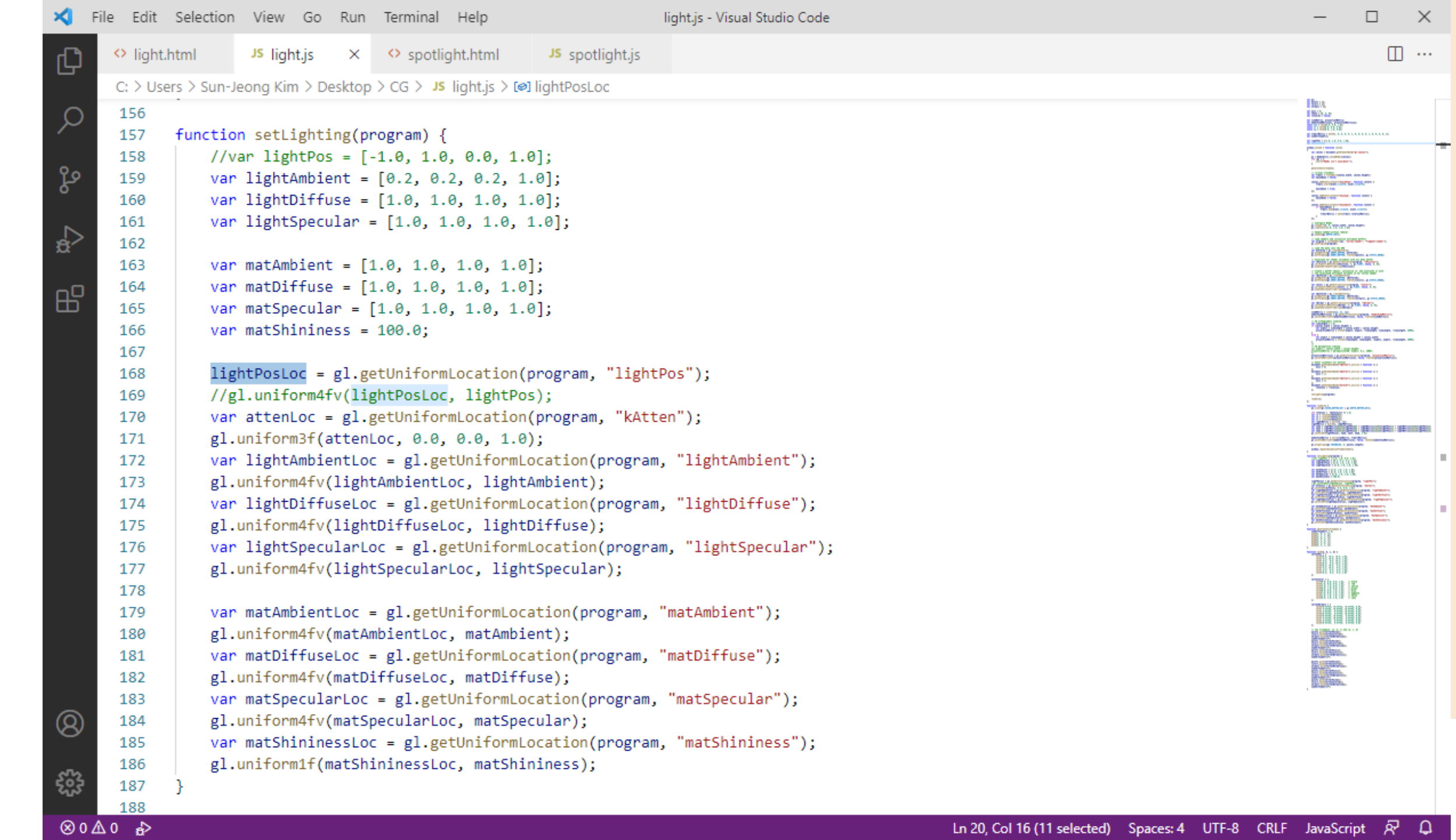
연습 문제 (3)

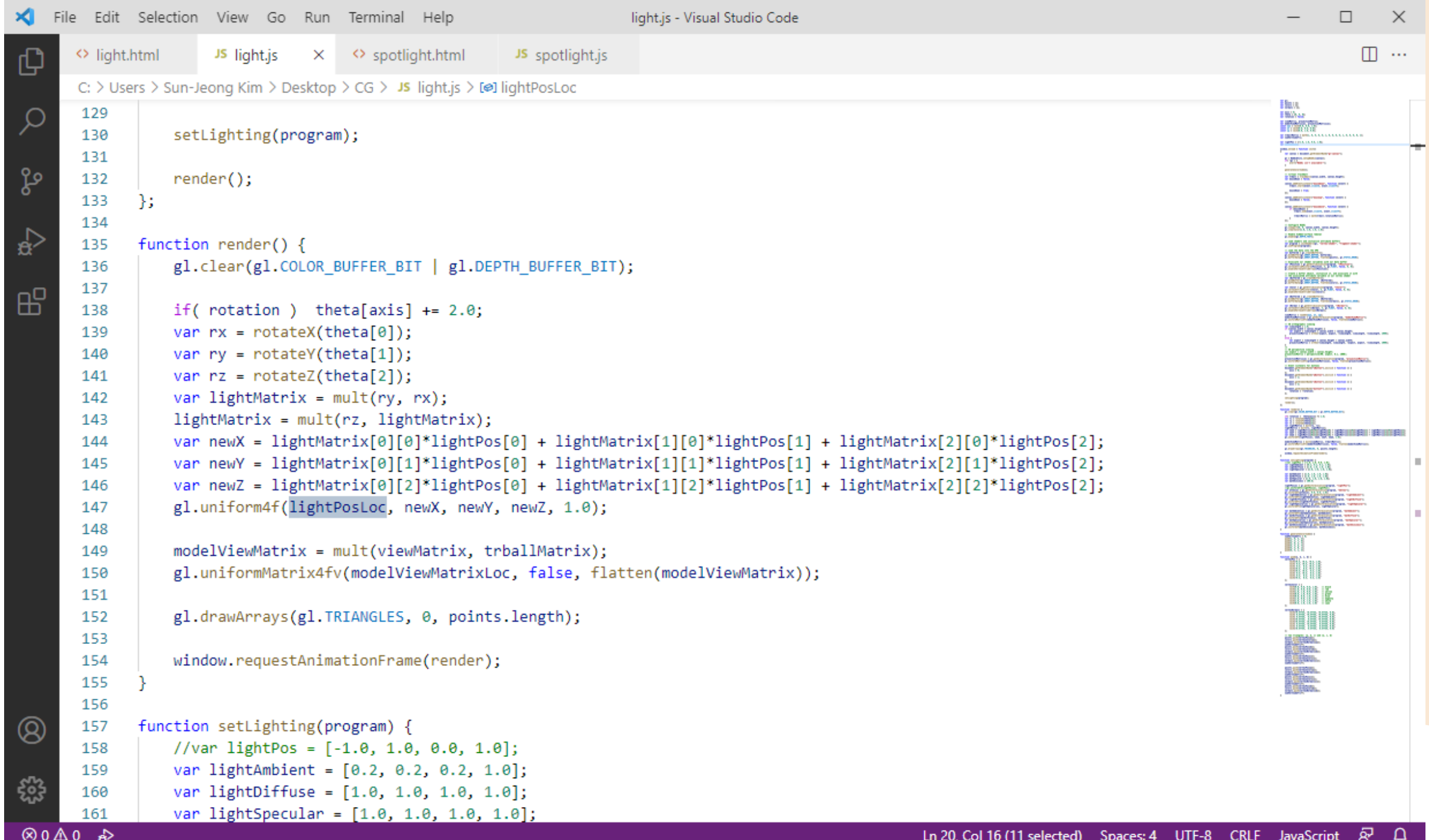
- light.js에 있는 setLighting( ) 함수에서,
  - spotDir의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) gl.uniform3f(spotDirLoc, 0.0, 0.0, -1.0);
  - spotExp의 값을 변경하면, 결과가 어떻게 바뀌는지 설명하시오.
    - 예) gl.uniform1f(spotExpLoc, 10.0);

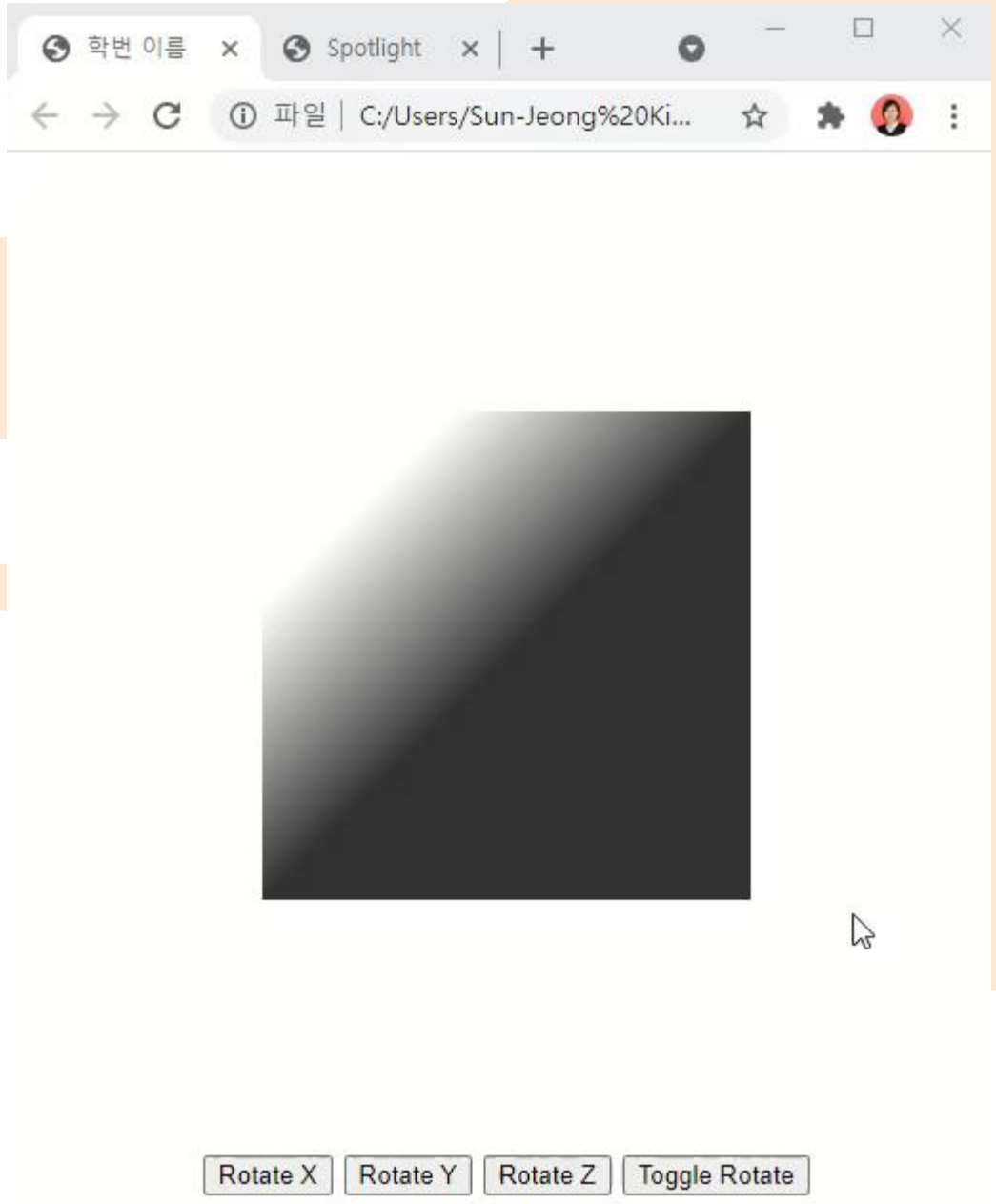
Moving Light Sources

- Light sources are geometric objects whose positions or directions are affected by the model-view matrix
- Depending on where we place the position (direction) setting function, we can
  - Move the light source(s) with object(s)
  - Fix the object(s) and move the light source(s)
  - Fix the light source(s) and move the object(s)
  - Move the light source(s) and object(s) independently


File Edit Selection View Go Run Terminal Help
light.js - Visual Studio Code
light.html
light.js
spotlight.html
spotlight.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > lightPosLoc
var gl;
var points = [];
var colors = [];
var normals = [];
var axis = 0;
var theta = [0, 0, 0];
var rotation = false;
var viewMatrix, projectionMatrix;
var modelViewMatrixLoc, projectionMatrixLoc;
const eye = vec3(0.0, 0.0, 1.0);
const at = vec3(0.0, 0.0, 0.0);
const up = vec3(0.0, 1.0, 0.0);
var trballMatrix = mat4(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);
var numVertCubeTri;
var lightPos = [-1.0, 1.0, 0.0, 1.0];
var lightPosLoc;
window.onload = function init()
{
var canvas = document.getElementById("gl-canvas");
gl = WebGLUtils.setupWebGL(canvas);
if(!gl) {
alert("WebGL isn't available!");
}
generateColorCube();
// virtual trackball
Ln 20, Col 16 (11 selected) Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
light.js - Visual Studio Code
light.html
light.js
spotlight.html
spotlight.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > lightPosLoc
156
157 function setLighting(program) {
158 //var lightPos = [-1.0, 1.0, 0.0, 1.0];
159 var lightAmbient = [0.2, 0.2, 0.2, 1.0];
160 var lightDiffuse = [1.0, 1.0, 1.0, 1.0];
161 var lightSpecular = [1.0, 1.0, 1.0, 1.0];
162
163 var matAmbient = [1.0, 1.0, 1.0, 1.0];
164 var matDiffuse = [1.0, 1.0, 1.0, 1.0];
165 var matSpecular = [1.0, 1.0, 1.0, 1.0];
166 var matShininess = 100.0;
167
168 lightPosLoc = gl.getUniformLocation(program, "lightPos");
169 //gl.uniform4fv(lightPosLoc, lightPos);
170 var attenLoc = gl.getUniformLocation(program, "kAtten");
171 gl.uniform3f(attenLoc, 0.0, 0.0, 1.0);
172 var lightAmbientLoc = gl.getUniformLocation(program, "lightAmbient");
173 gl.uniform4fv(lightAmbientLoc, lightAmbient);
174 var lightDiffuseLoc = gl.getUniformLocation(program, "lightDiffuse");
175 gl.uniform4fv(lightDiffuseLoc, lightDiffuse);
176 var lightSpecularLoc = gl.getUniformLocation(program, "lightSpecular");
177 gl.uniform4fv(lightSpecularLoc, lightSpecular);
178
179 var matAmbientLoc = gl.getUniformLocation(program, "matAmbient");
180 gl.uniform4fv(matAmbientLoc, matAmbient);
181 var matDiffuseLoc = gl.getUniformLocation(program, "matDiffuse");
182 gl.uniform4fv(matDiffuseLoc, matDiffuse);
183 var matSpecularLoc = gl.getUniformLocation(program, "matSpecular");
184 gl.uniform4fv(matSpecularLoc, matSpecular);
185 var matShininessLoc = gl.getUniformLocation(program, "matShininess");
186 gl.uniform1f(matShininessLoc, matShininess);
187 }
188
⊗ 0 ⚠ 0
Ln 20, Col 16 (11 selected) Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
light.js - Visual Studio Code
light.html
light.js
spotlight.html
spotlight.js
C: > Users > Sun-Jeong Kim > Desktop > CG > light.js > lightPosLoc
129
130 setLighting(program);
131
132 render();
133 };
134
135 function render() {
136 gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
137
138 if(rotation) theta[axis] += 2.0;
139 var rx = rotateX(theta[0]);
140 var ry = rotateY(theta[1]);
141 var rz = rotateZ(theta[2]);
142 var lightMatrix = mult(ry, rx);
143 lightMatrix = mult(rz, lightMatrix);
144 var newX = lightMatrix[0][0]*lightPos[0] + lightMatrix[1][0]*lightPos[1] + lightMatrix[2][0]*lightPos[2];
145 var newY = lightMatrix[0][1]*lightPos[0] + lightMatrix[1][1]*lightPos[1] + lightMatrix[2][1]*lightPos[2];
146 var newZ = lightMatrix[0][2]*lightPos[0] + lightMatrix[1][2]*lightPos[1] + lightMatrix[2][2]*lightPos[2];
147 gl.uniform4f(lightPosLoc, newX, newY, newZ, 1.0);
148
149 modelViewMatrix = mult(viewMatrix, trballMatrix);
150 gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(modelViewMatrix));
151
152 gl.drawArrays(gl.TRIANGLES, 0, points.length);
153
154 window.requestAnimationFrame(render);
155 }
156
157 function setLighting(program) {
158 //var lightPos = [-1.0, 1.0, 0.0, 1.0];
159 var lightAmbient = [0.2, 0.2, 0.2, 1.0];
160 var lightDiffuse = [1.0, 1.0, 1.0, 1.0];
161 var lightSpecular = [1.0, 1.0, 1.0, 1.0];
⊗ 0 ⚠ 0
Ln 20, Col 16 (11 selected) Spaces: 4 UTF-8 CRLF JavaScript


학번 이름
Spotlight
파일 | C:/Users/Sun-Jeong%20Ki...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate