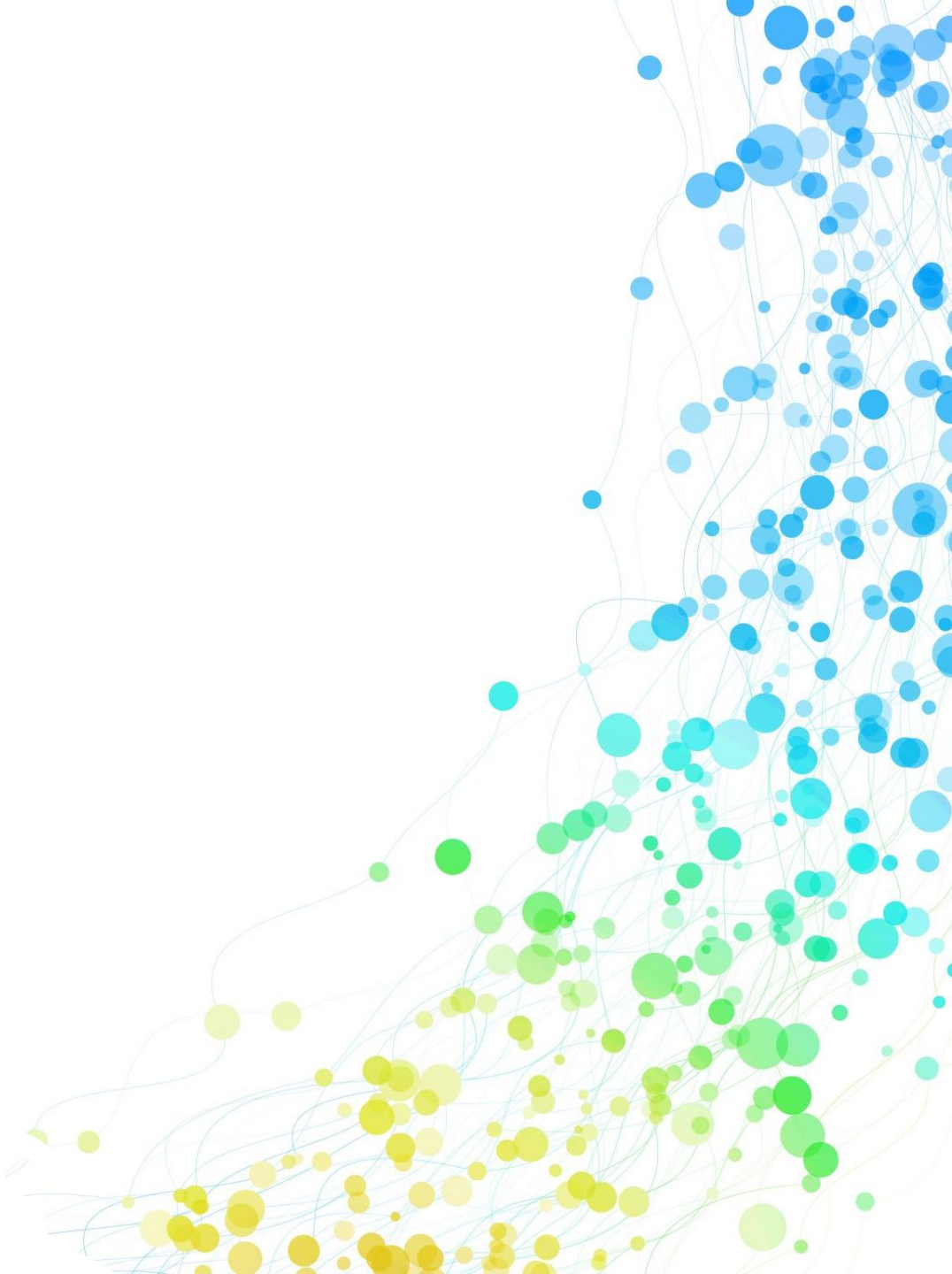
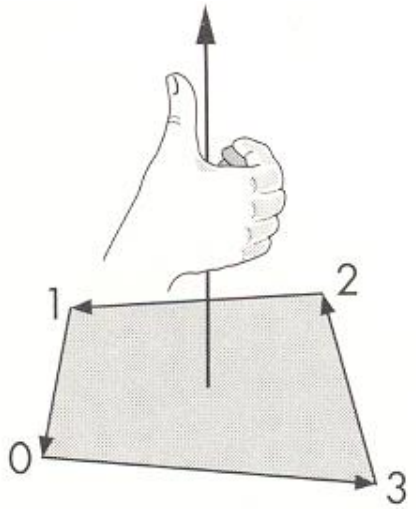
Geometric Objects and Transformations

6TH WEEK, 2021



Modeling a Cube

- Surface-based model
  - Outward-pointing face
    - Right-hand rule: counterclockwise order
- Data structure
  - Geometry: location of vertices

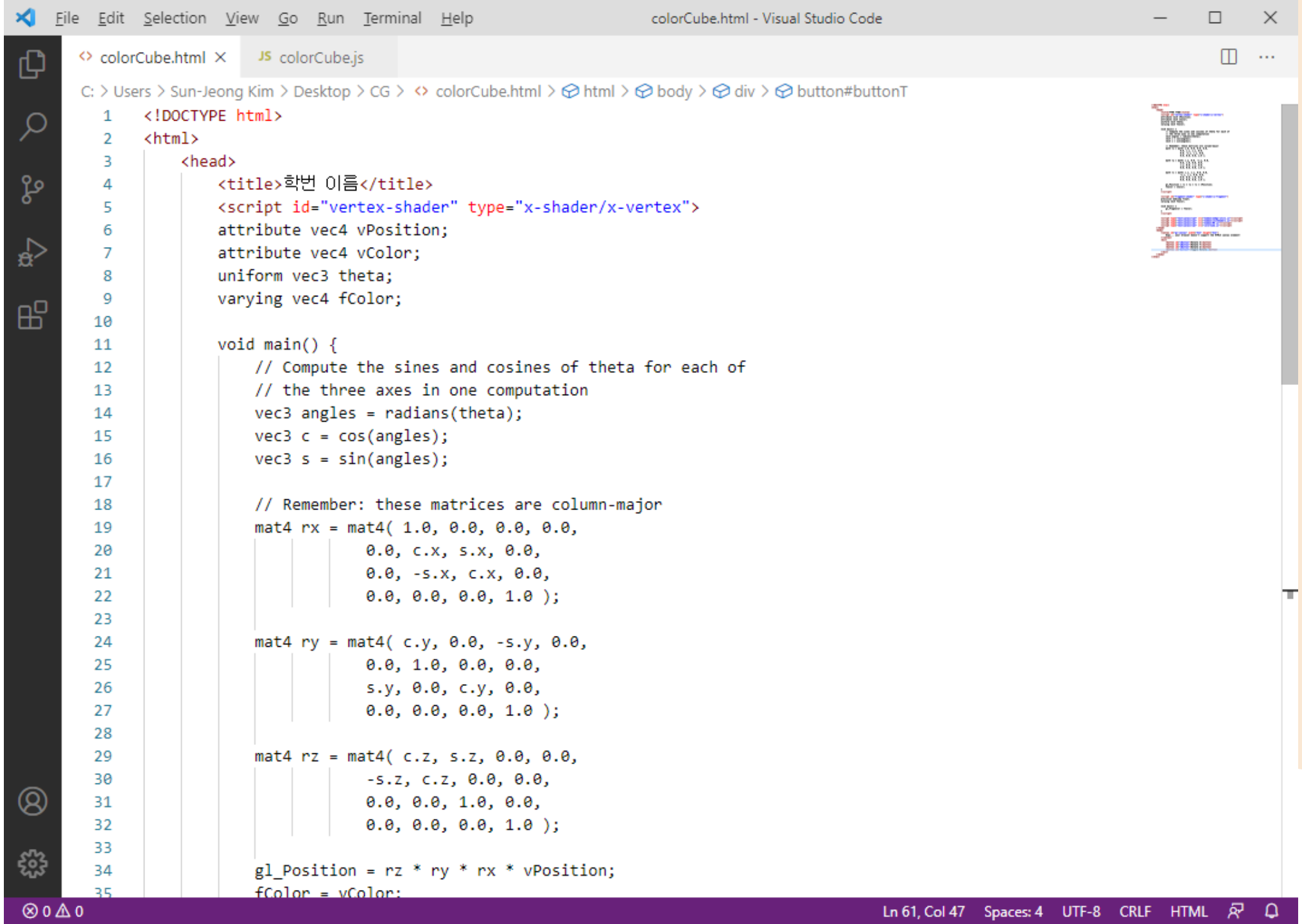


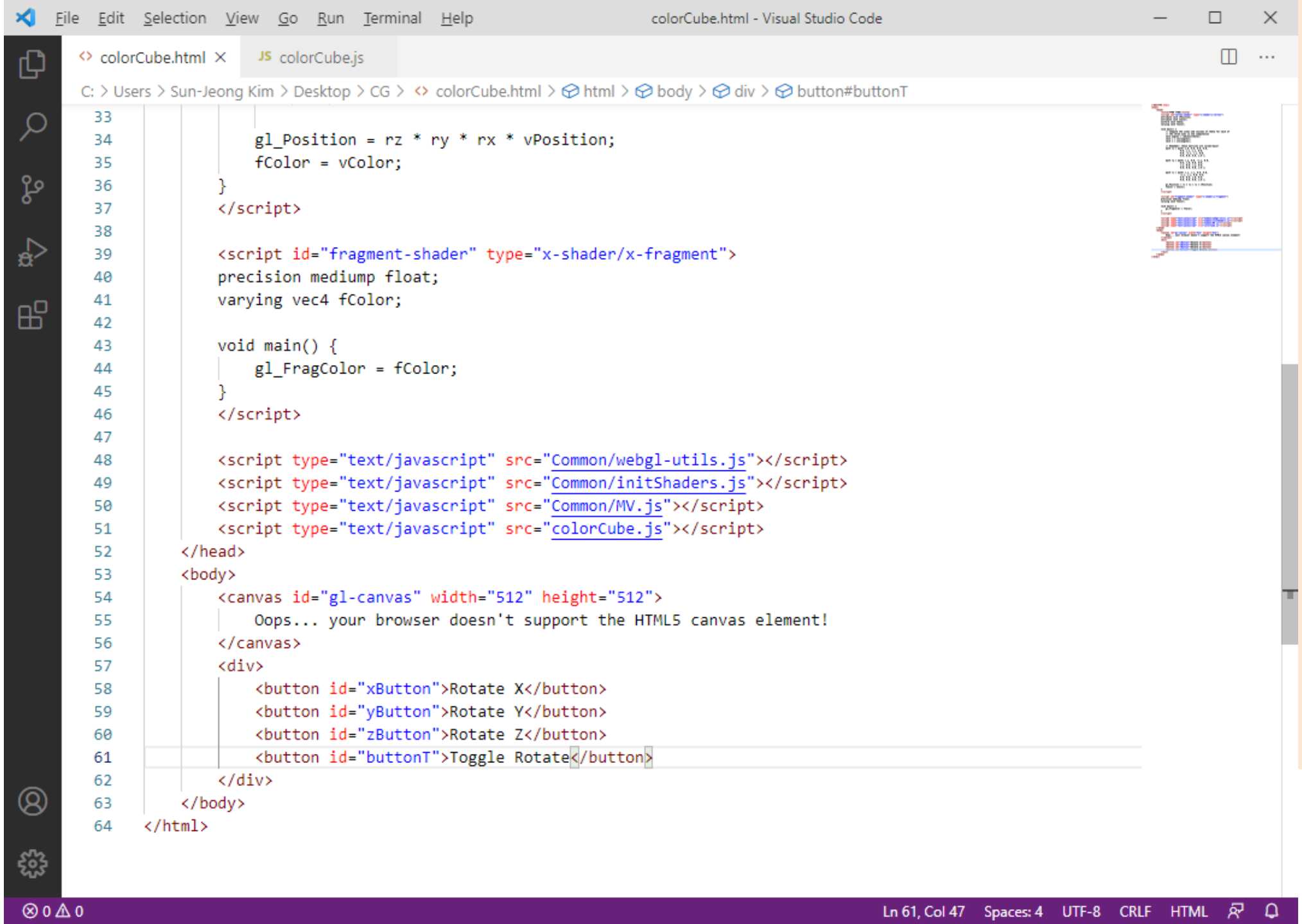

```
float vertices[8][3]= { {-1.0, -1.0, -1.0}, {1.0, -1.0, -1.0},
        {1.0, 1.0, -1.0}, {-1.0, 1.0, -1.0}, {-1.0, -1.0, 1.0},
        {1.0, -1.0, 1.0}, {1.0, 1.0, 1.0}, {-1.0, 1.0, 1.0}};
```

  - Topology: connectivity

```
var indices = [ 0, 3, 2, 0, 2, 1 ];
for ( var i = 0; i < indices.length; ++i )
        points.push( vertices[indices[i]] );

gl.drawArrays( gl.TRIANGLES, 0, NumVertices );
```


File Edit Selection View Go Run Terminal Help
colorCube.html - Visual Studio Code
colorCube.html
colorCube.js
C: > Users > Sun-Jeong Kim > Desktop > CG > colorCube.html > html > body > div > button#buttonT
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 vColor;
8 uniform vec3 theta;
9 varying vec4 fColor;
10
11 void main() {
12 // Compute the sines and cosines of theta for each of
13 // the three axes in one computation
14 vec3 angles = radians(theta);
15 vec3 c = cos(angles);
16 vec3 s = sin(angles);
17
18 // Remember: these matrices are column-major
19 mat4 rx = mat4(1.0, 0.0, 0.0, 0.0,
20 0.0, c.x, s.x, 0.0,
21 0.0, -s.x, c.x, 0.0,
22 0.0, 0.0, 0.0, 1.0);
23
24 mat4 ry = mat4(c.y, 0.0, -s.y, 0.0,
25 0.0, 1.0, 0.0, 0.0,
26 s.y, 0.0, c.y, 0.0,
27 0.0, 0.0, 0.0, 1.0);
28
29 mat4 rz = mat4(c.z, s.z, 0.0, 0.0,
30 -s.z, c.z, 0.0, 0.0,
31 0.0, 0.0, 1.0, 0.0,
32 0.0, 0.0, 0.0, 1.0);
33
34 gl_Position = rz * ry * rx * vPosition;
35 fColor = vColor;
Ln 61, Col 47 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
colorCube.html - Visual Studio Code
colorCube.html
colorCube.js
C: > Users > Sun-Jeong Kim > Desktop > CG > colorCube.html > html > body > div > button#buttonT
gl_Position = rz * ry * rx * 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="colorCube.js"></script>
</head>
<body>
<canvas id="gl-canvas" width="512" height="512">
Oops... your browser doesn't support the HTML5 canvas element!
</canvas>

<button id="xButton">Rotate X</button>
<button id="yButton">Rotate Y</button>
<button id="zButton">Rotate Z</button>
<button id="buttonT">Toggle Rotate</button>

</body>
</html>
Ln 61, Col 47 Spaces: 4 UTF-8 CRLF HTML

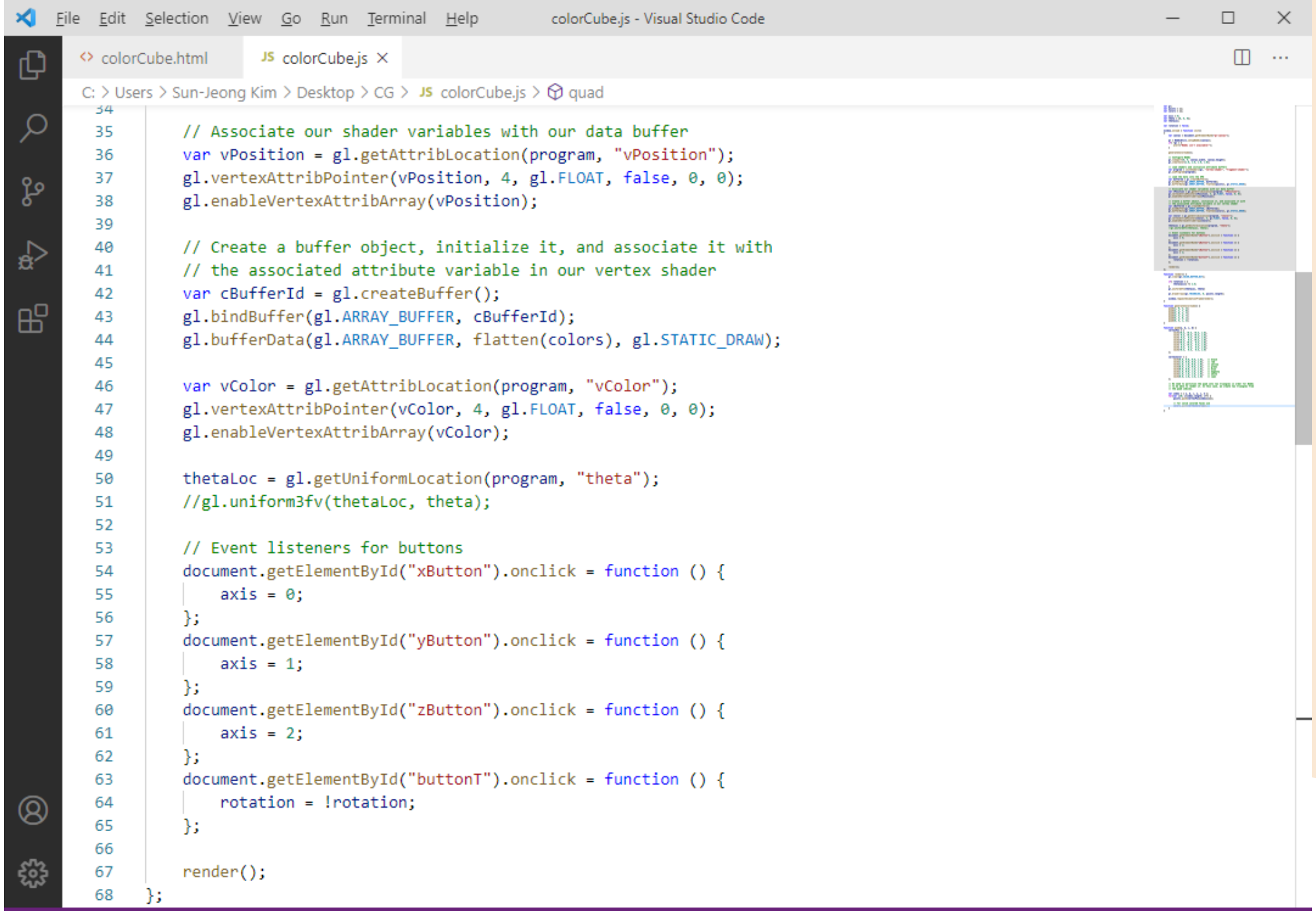
File Edit Selection View Go Run Terminal Help colorCube.js - Visual Studio Code

colorCube.html | colorCube.js

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

```
1   var gl;
2   var points = [];
3   var colors = [];
4
5   var axis = 0;
6   var theta = [0, 0, 0];
7   var thetaLoc;
8
9   var rotation = false;
10
11  window.onload = function init()
12  {
13      var canvas = document.getElementById("gl-canvas");
14
15      gl = WebGLUtils.setupWebGL(canvas);
16      if( !gl ) {
17          alert("WebGL isn't available!");
18      }
19
20      generateColorCube();
21
22      // Configure WebGL
23      gl.viewport(0, 0, canvas.width, canvas.height);
24      gl.clearColor(1.0, 1.0, 1.0, 1.0);
25
26      // Load shaders and initialize attribute buffers
27      var program = initShaders(gl, "vertex-shader", "fragment-shader");
28      gl.useProgram(program);
29
30      // Load the data into the GPU
31      var bufferId = gl.createBuffer();
32      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
33      gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
34
35      // Associate our shader variables with our data buffer
```

⊗ 0 ⚠ 0 Ln 124, Col 37 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
colorCube.js - Visual Studio Code
colorCube.html
colorCube.js
C: > Users > Sun-Jeong Kim > Desktop > CG > colorCube.js > quad
// 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);
thetaLoc = gl.getUniformLocation(program, "theta");
//gl.uniform3fv(thetaLoc, theta);
// Event listeners for buttons
document.getElementById("xButton").onclick = function () {
axis = 0;
};
document.getElementById("yButton").onclick = function () {
axis = 1;
};
document.getElementById("zButton").onclick = function () {
axis = 2;
};
document.getElementById("buttonT").onclick = function () {
rotation = !rotation;
};
render();
};
Ln 124, Col 37 Spaces: 4 UTF-8 CRLF JavaScript

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

colorCube.html | colorCube.js

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

```
69
70  function render() {
71      gl.clear(gl.COLOR_BUFFER_BIT);
72
73      if( rotation ) {
74          theta[axis] += 2.0;
75      }
76      gl.uniform3fv(thetaLoc, theta)
77
78      gl.drawArrays(gl.TRIANGLES, 0, points.length);
79
80      window.requestAnimationFrame(render);
81  }
82
83  function generateColorCube() {
84      quad(1, 0, 3, 2);
85      quad(2, 3, 7, 6);
86      quad(3, 0, 4, 7);
87      quad(4, 5, 6, 7);
88      quad(5, 4, 0, 1);
89      quad(6, 5, 1, 2);
90  }
91
92  function quad(a, b, c, d) {
93      vertexPos = [
94          vec4(-0.5, -0.5, -0.5, 1.0),
95          vec4( 0.5, -0.5, -0.5, 1.0),
96          vec4( 0.5,  0.5, -0.5, 1.0),
97          vec4(-0.5,  0.5, -0.5, 1.0),
98          vec4(-0.5, -0.5,  0.5, 1.0),
99          vec4( 0.5, -0.5,  0.5, 1.0),
100         vec4( 0.5,  0.5,  0.5, 1.0),
101         vec4(-0.5,  0.5,  0.5, 1.0)
102     ];
103
```

⊗ 0 ⚠ 0 Ln 124, Col 37 Spaces: 4 UTF-8 CRLF JavaScript

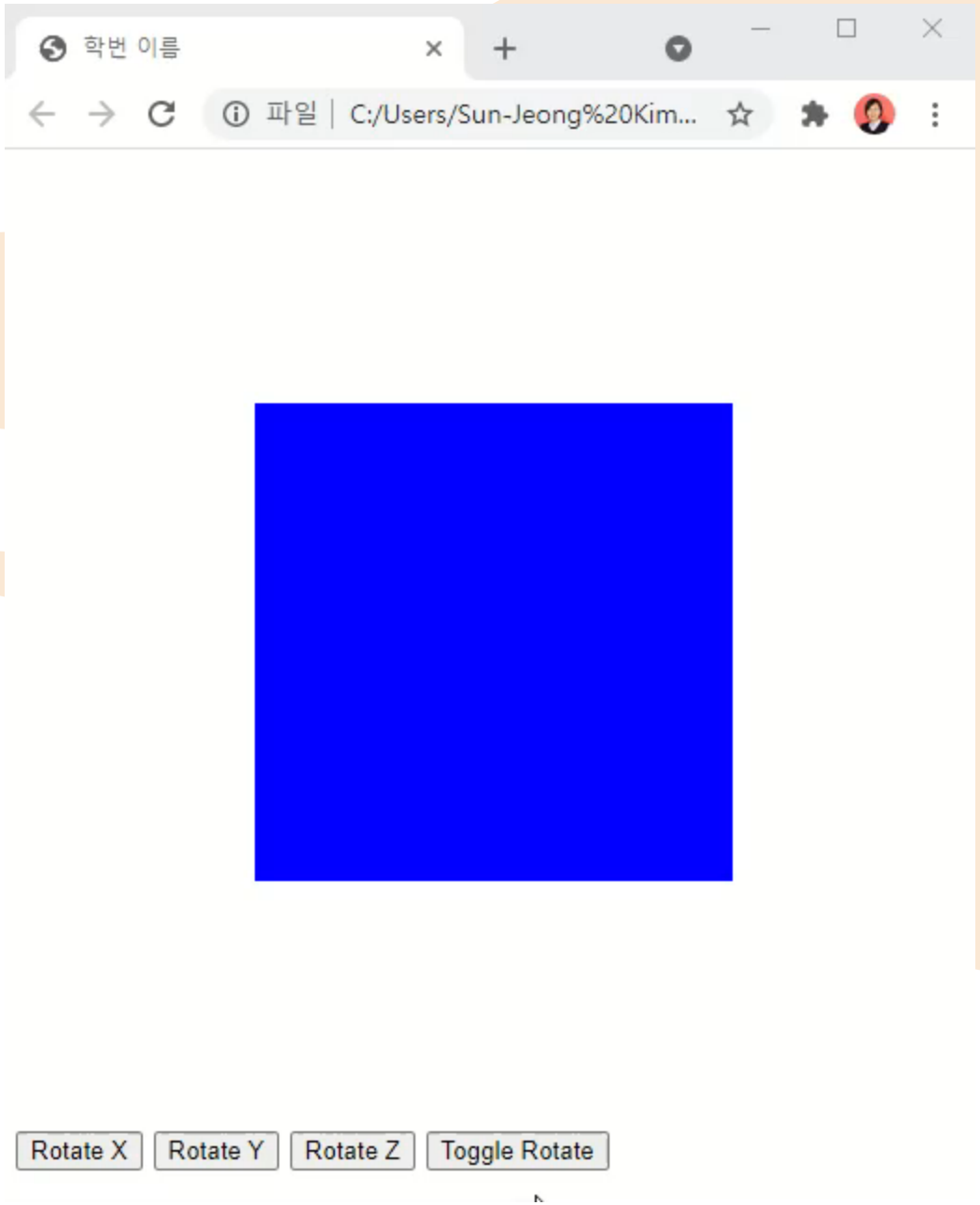
File Edit Selection View Go Run Terminal Help colorCube.js - Visual Studio Code

colorCube.html | colorCube.js

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

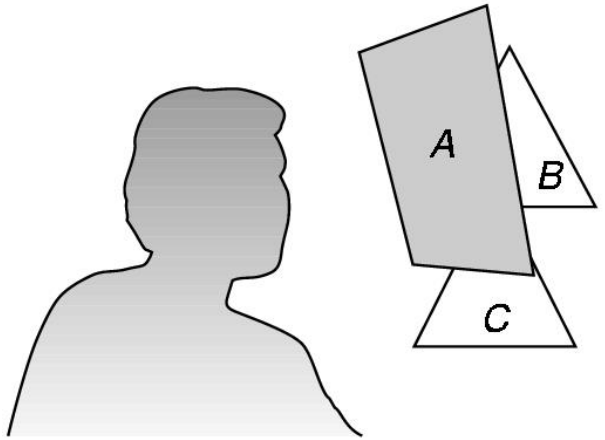
```
93      vertexPos = [
94          vec4(-0.5, -0.5, -0.5, 1.0),
95          vec4( 0.5, -0.5, -0.5, 1.0),
96          vec4( 0.5,  0.5, -0.5, 1.0),
97          vec4(-0.5,  0.5, -0.5, 1.0),
98          vec4(-0.5, -0.5,  0.5, 1.0),
99          vec4( 0.5, -0.5,  0.5, 1.0),
100         vec4( 0.5,  0.5,  0.5, 1.0),
101         vec4(-0.5,  0.5,  0.5, 1.0)
102     ];
103
104     vertexColor = [
105         vec4(0.0, 0.0, 0.0, 1.0),   // black
106         vec4(1.0, 0.0, 0.0, 1.0),   // red
107         vec4(1.0, 1.0, 0.0, 1.0),   // yellow
108         vec4(0.0, 1.0, 0.0, 1.0),   // green
109         vec4(0.0, 0.0, 1.0, 1.0),   // blue
110         vec4(1.0, 0.0, 1.0, 1.0),   // magenta
111         vec4(1.0, 1.0, 1.0, 1.0),   // white
112         vec4(0.0, 1.0, 1.0, 1.0)    // cyan
113     ];
114
115     // We need to partition the quad into two triangles in order for WebGL
116     // to be able to render it. In this case, we create two triangles from
117     // the quad indices.
118     var index = [ a, b, c, a, c, d ];
119     for(var i=0; i<index.length; i++) {
120         points.push(vertexPos[index[i]]);
121
122         // for solid colored faces use
123         colors.push(vertexColor[a]);
124     }
125 }
126
```

⊗ 0 ⚠ 0 Ln 123, Col 35 Spaces: 4 UTF-8 CRLF JavaScript


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

Hidden-Surface Removal

- To see only those surfaces in front of other surfaces
- Visible-surface algorithms or hidden-surface-removal algorithm
  - Algorithms for ordering objects
  - OpenGL uses the z-buffer algorithm that saves depth information



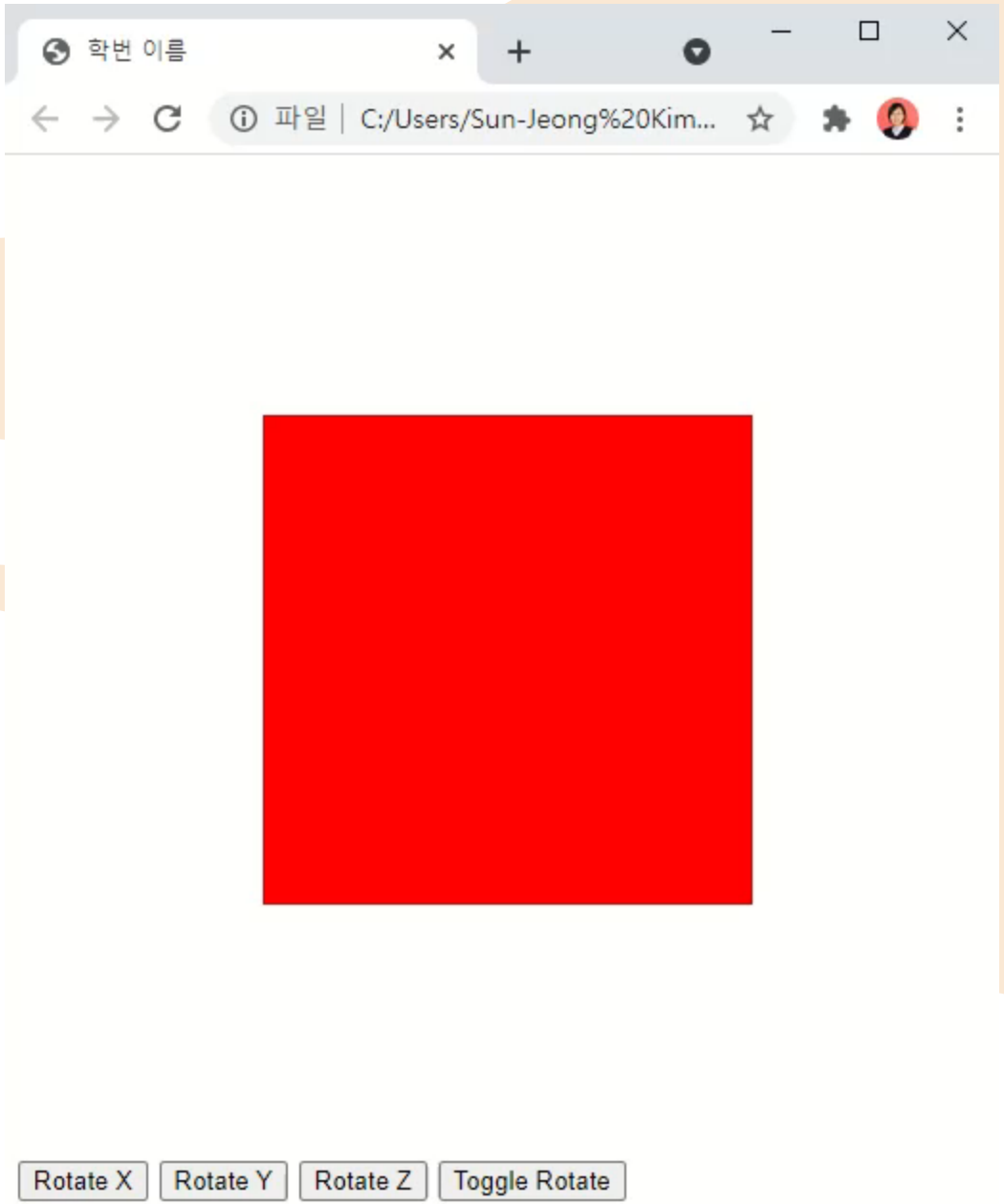
Hidden-Surface Problem

Using the Z-buffer Algorithm

- The algorithm uses an extra buffer, the z-buffer, to store depth information as geometry travels down the pipeline
- Depth buffer is required to available in WebGL
- It must be
  - Enabled
    - `gl.enable( gl.DEPTH_TEST );`
  - Cleared in for each render
    - `gl.clear( gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT );`

```
21
22      // Configure WebGL
23      gl.viewport(0, 0, canvas.width, canvas.height);
24      gl.clearColor(1.0, 1.0, 1.0, 1.0);
25
26      // Enable hidden-surface removal
27      gl.enable(gl.DEPTH_TEST);
28
29      // Load shaders and initialize attribute buffers
30      var program = initShaders(gl, "vertex-shader", "fragment-shader");
31      gl.useProgram(program);
32
```

```
73  function render() {
74      gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
75
76      if( rotation ) {
77          theta[axis] += 2.0;
78      }
79      gl.uniform3fv(thetaLoc, theta)
80
81      gl.drawArrays(gl.TRIANGLES, 0, points.length);
82
83      window.requestAnimationFrame(render);
84  }
```


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

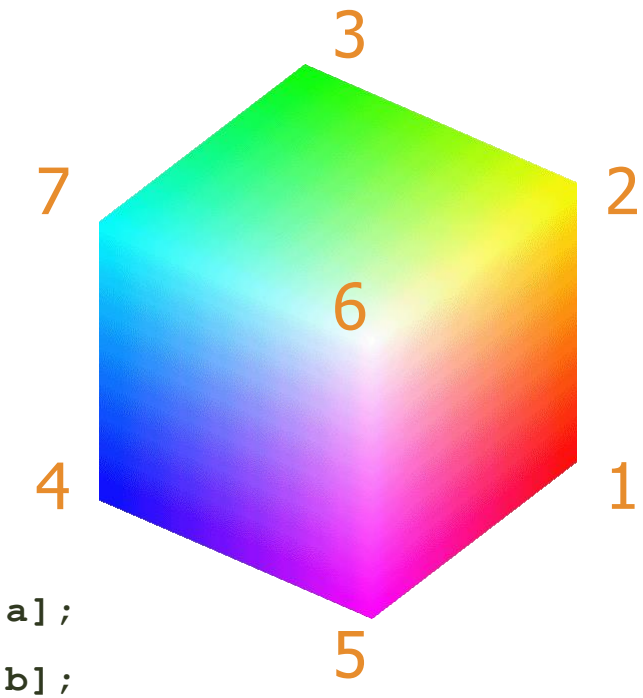
Color Cube (1)

```
float vertex_pos[8][3]= { {-1.0, -1.0, -1.0}, {1.0, -1.0, -1.0},
        {1.0, 1.0, -1.0}, {-1.0, 1.0, -1.0}, {-1.0, -1.0, 1.0},
        {1.0, -1.0, 1.0}, {1.0, 1.0, 1.0}, {-1.0, 1.0, 1.0}};

float vertex_color[8][3] = { {0.0, 0.0, 0.0}, {1.0, 0.0, 0.0},
        {1.0, 1.0, 0.0}, {0.0, 1.0, 0.0}, {0.0, 0.0, 1.0},
        {1.0, 0.0, 1.0}, {1.0, 1.0, 1.0}, {0.0, 1.0, 1.0}};

int Index = 0;
void quad( int a, int b, int c, int d )
{
    colors[Index] = vertex_color[a];    points[Index] = vertex_pos[a];
    Index++;
    colors[Index] = vertex_color[b];    points[Index] = vertex_pos[b];
    Index++;
    colors[Index] = vertex_color[c];    points[Index] = vertex_pos[c];
    Index++;

    colors[Index] = vertex_color[a];    points[Index] = vertex_pos[a];
    Index++;
    colors[Index] = vertex_color[c];    points[Index] = vertex_pos[c];
    Index++;
    colors[Index] = vertex_color[b];    points[Index] = vertex_pos[b];
    Index++;
}
```

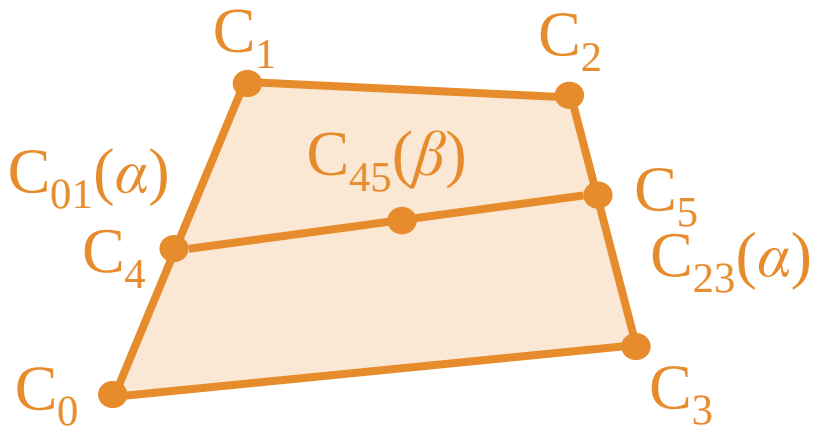


Color Cube (2)

```
void generateColorCube()
{
    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 );
}
```

- Bilinear interpolation

$$C_{01}(\alpha) = (1-\alpha)C_0 + \alpha C_1$$
$$C_{23}(\alpha) = (1-\alpha)C_2 + \alpha C_3$$
$$C_{45}(\beta) = (1-\beta)C_4 + \beta C_5$$



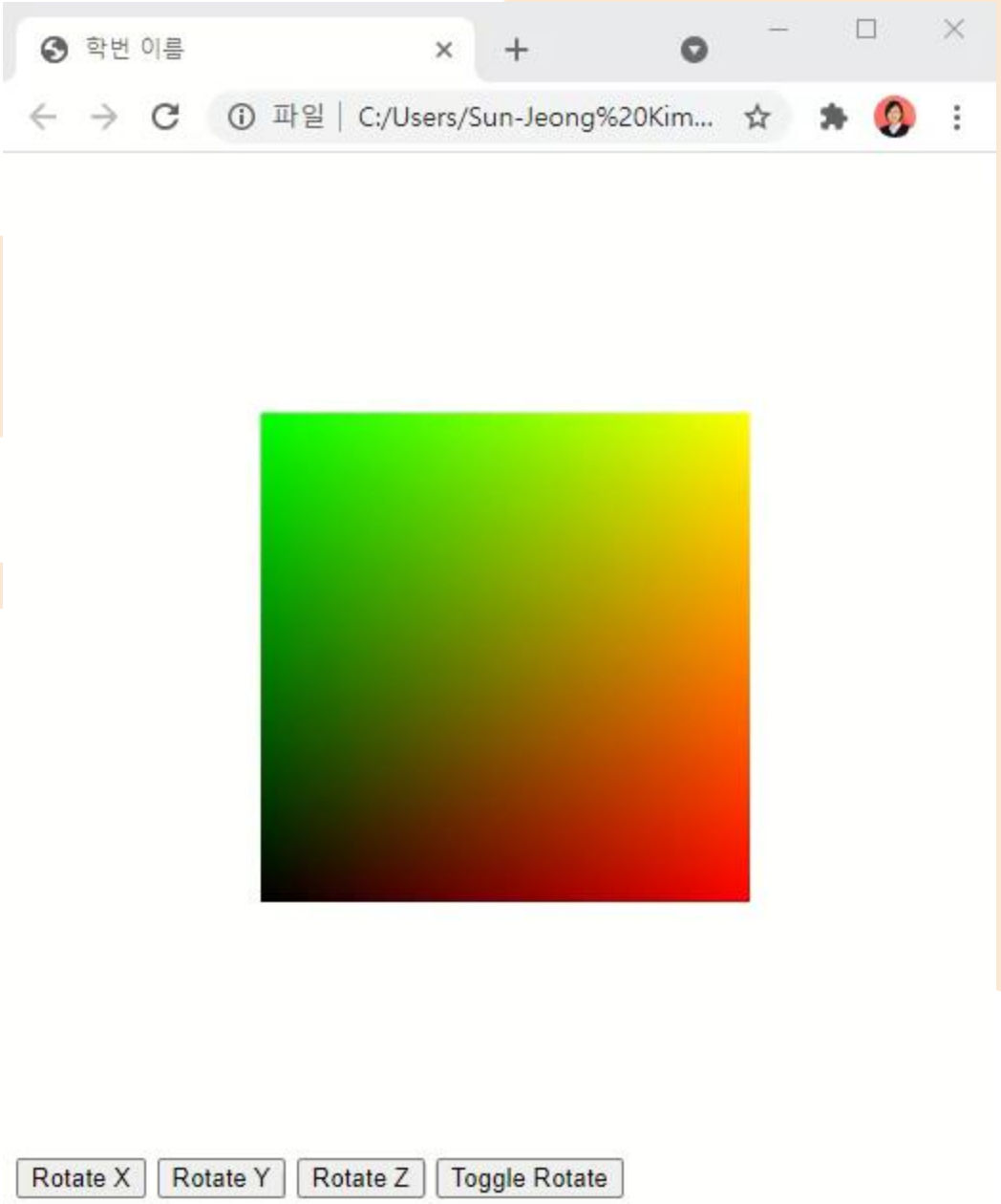
File Edit Selection View Go Run Terminal Help colorCube.js - Visual Studio Code

colorCube.html | colorCube.js

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

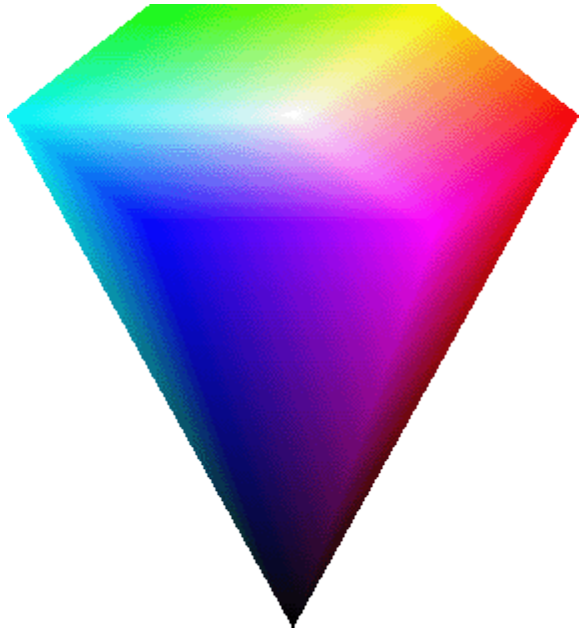
```
106
107     vertexColor = [
108         vec4(0.0, 0.0, 0.0, 1.0),   // black
109         vec4(1.0, 0.0, 0.0, 1.0),   // red
110         vec4(1.0, 1.0, 0.0, 1.0),   // yellow
111         vec4(0.0, 1.0, 0.0, 1.0),   // green
112         vec4(0.0, 0.0, 1.0, 1.0),   // blue
113         vec4(1.0, 0.0, 1.0, 1.0),   // magenta
114         vec4(1.0, 1.0, 1.0, 1.0),   // white
115         vec4(0.0, 1.0, 1.0, 1.0)    // cyan
116     ];
117
118     // We need to partition the quad into two triangles in order for WebGL
119     // to be able to render it. In this case, we create two triangles from
120     // the quad indices.
121     var index = [ a, b, c, a, c, d ];
122     for(var i=0; i<index.length; i++) {
123         points.push(vertexPos[index[i]]);
124         colors.push(vertexColor[index[i]]);
125
126         // for solid colored faces use
127         //colors.push(vertexColor[a]);
128     }
129 }
130
```

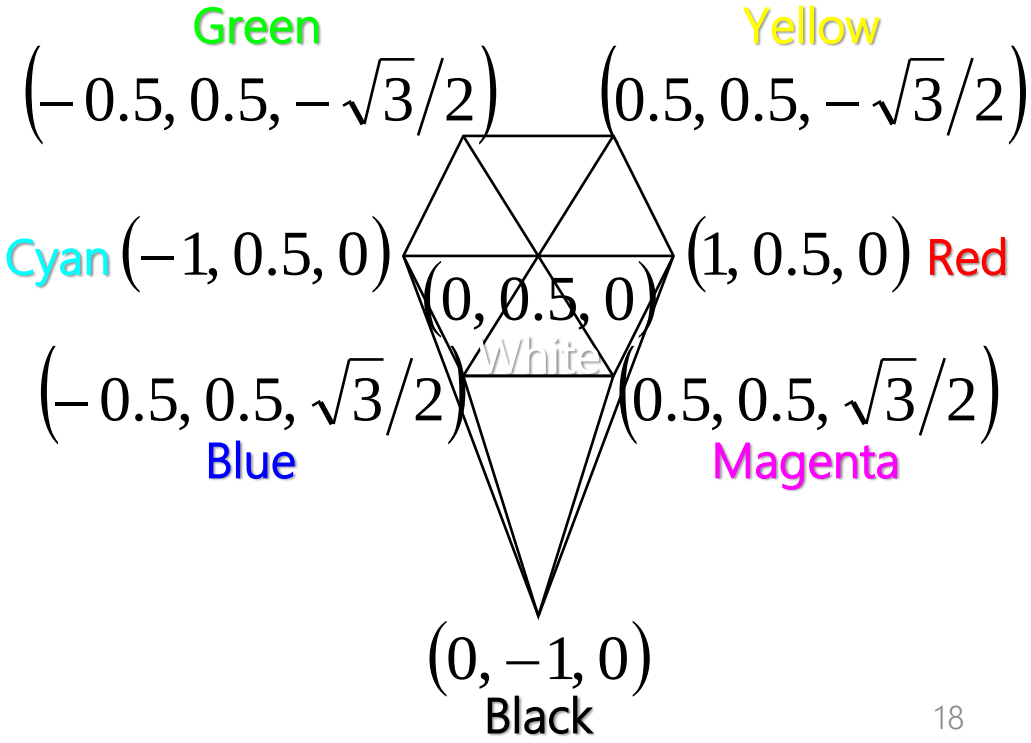
⊗ 0 ⚠ 0 Ln 124, Col 44 Spaces: 4 UTF-8 CRLF JavaScript

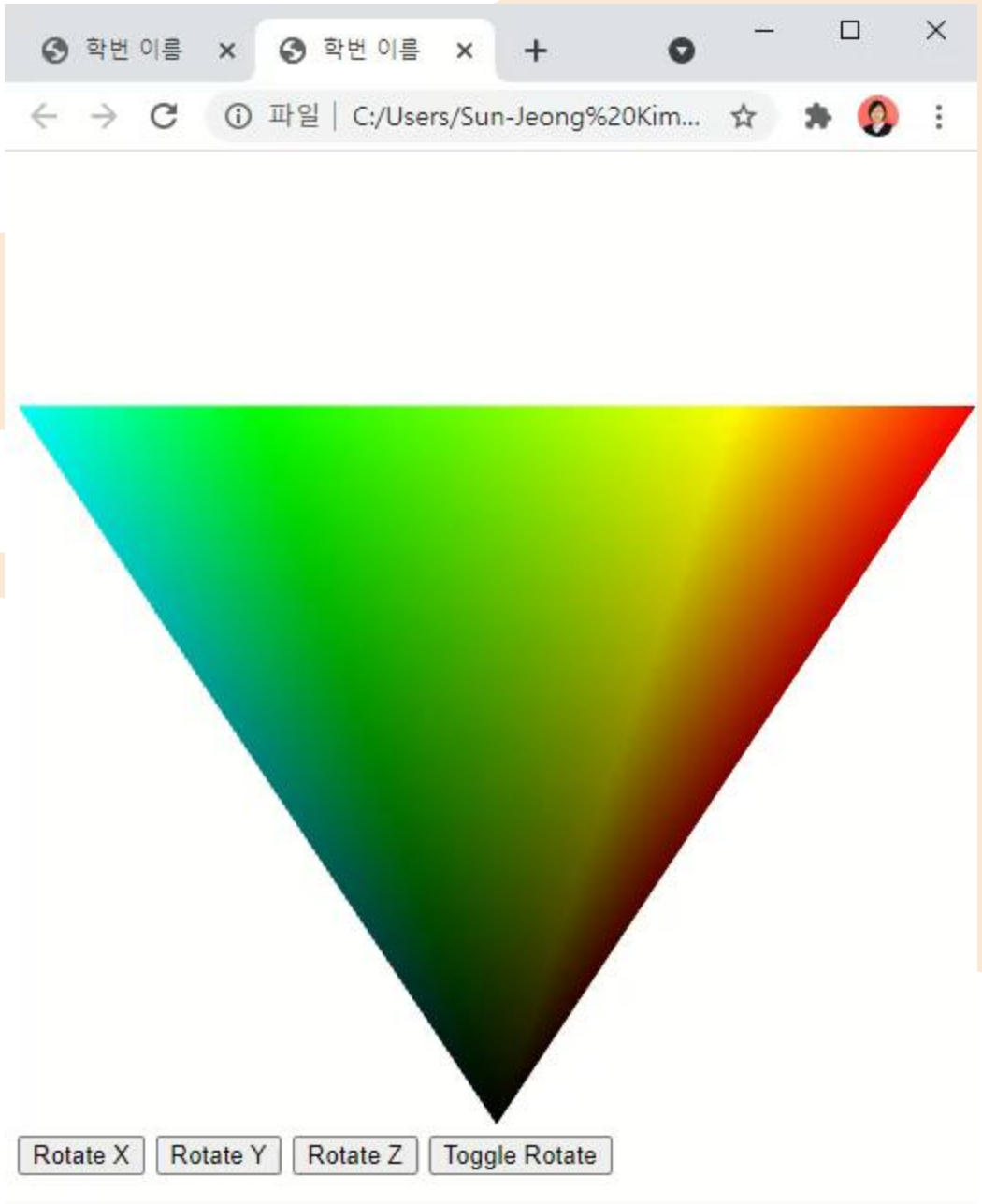

학번 이름
파일 | C:/Users/Sun-Jeong%20Kim...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

연습 문제 (1)

- Hexagonal Pyramid를 그리시오.
  - colorCube.html, js ➔ hexaPyramid.html, js로 복사
  - Vertex 8개
  - Triangle 12개






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

Line-Preserving Transformations

- Two classes of transformations of importance to CG that preserves lines
  - Affine and projective transformations
- Homogeneous coordinates
  - Vectors: $(x, y, z, 0)$, points: $(x, y, z, w)=(x/w, y/w, z/w, 1)$
  - 4x4 matrices: modeling, viewing, and projection
- OpenGL pipeline

  - Model-view: to position objects relative to camera
  - Projection: to specify clipping volume and map vertices to a normalized coordinate system

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

transform.html × transform.js

C: > Users > Sun-Jeong Kim > Desktop > CG > transform.html > html > body > div

```
<!DOCTYPE html>
<html>
    <head>
        <title>Transformations</title>
        <script id="vertex-shader" type="x-shader/x-vertex">
            attribute vec4 vPosition;
            attribute vec4 vColor;
            uniform vec3 theta;
            varying vec4 fColor;

            void main() {
                // Compute the sines and cosines of theta for each of
                // the three axes in one computation
                vec3 angles = radians(theta);
                vec3 c = cos(angles);
                vec3 s = sin(angles);

                // Remember: these matrices are column-major
                mat4 rx = mat4( 1.0, 0.0, 0.0, 0.0,
                                0.0, c.x, s.x, 0.0,
                                0.0, -s.x, c.x, 0.0,
                                0.0, 0.0, 0.0, 1.0 );

                mat4 ry = mat4( c.y, 0.0, -s.y, 0.0,
                                0.0, 1.0, 0.0, 0.0,
                                s.y, 0.0, c.y, 0.0,
                                0.0, 0.0, 0.0, 1.0 );

                mat4 rz = mat4( c.z, s.z, 0.0, 0.0,
                                -s.z, c.z, 0.0, 0.0,
                                0.0, 0.0, 1.0, 0.0,
                                0.0, 0.0, 0.0, 1.0 );

                gl_Position = rz * ry * rx * vPosition;
                fColor = vColor;
```

⊗ 0 ⚠ 0 Ln 62, Col 15 Spaces: 4 UTF-8 CRLF HTML

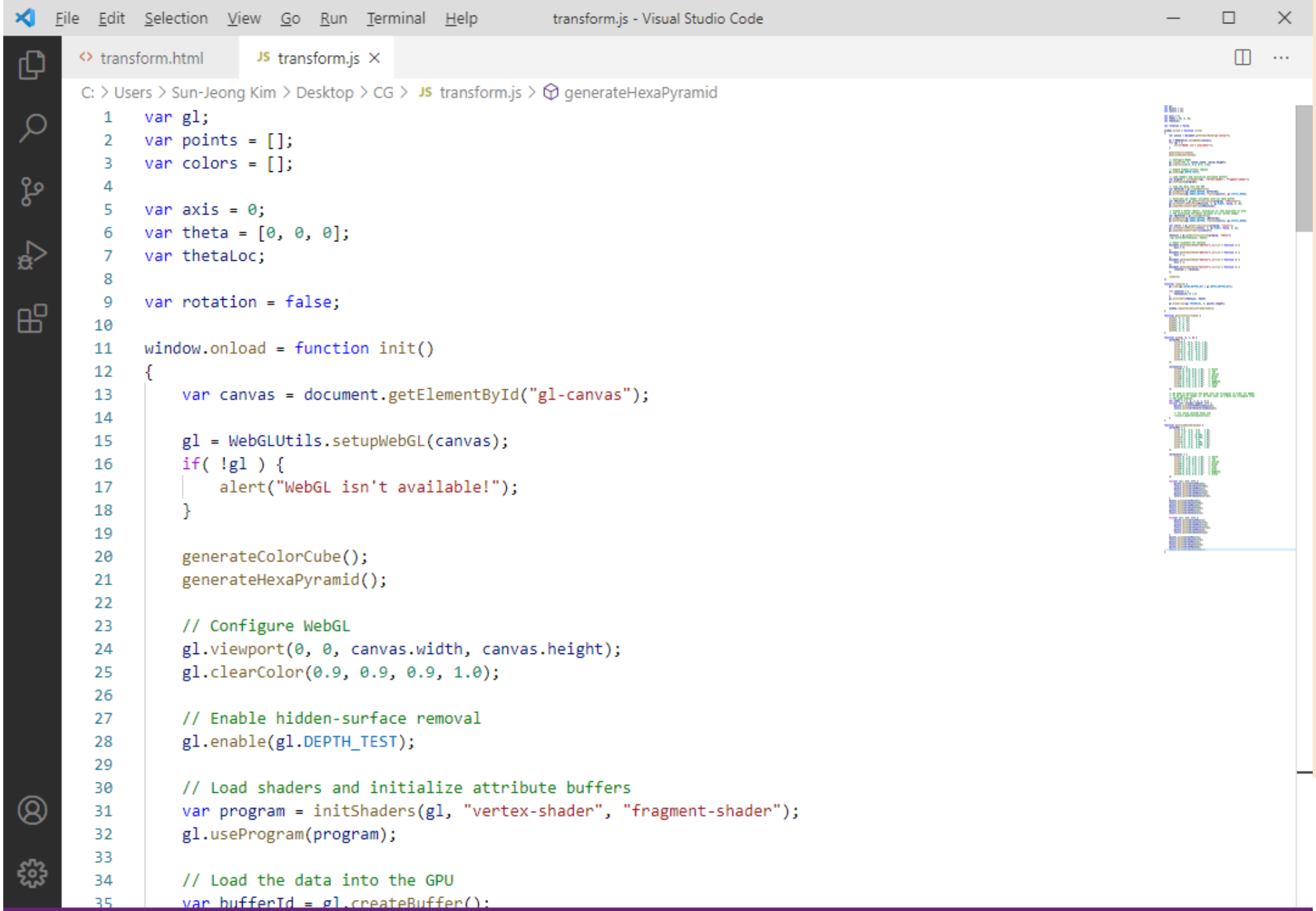
File Edit Selection View Go Run Terminal Help transform.html - Visual Studio Code

transform.html × transform.js

C: > Users > Sun-Jeong Kim > Desktop > CG > transform.html > html > body > div > button#buttonT

```
33
34                  gl_Position = rz * ry * rx * vPosition;
35                  fColor = vColor;
36              }
37              </script>
38
39              <script id="fragment-shader" type="x-shader/x-fragment">
40              precision mediump float;
41              varying vec4 fColor;
42
43              void main() {
44                  gl_FragColor = fColor;
45              }
46              </script>
47
48          <script type="text/javascript" src="Common/webgl-utils.js"></script>
49          <script type="text/javascript" src="Common/initShaders.js"></script>
50          <script type="text/javascript" src="Common/MV.js"></script>
51          <script type="text/javascript" src="transform.js"></script>
52      </head>
53      <body>
54          <canvas id="gl-canvas" width="512" height="512">
55              Oops... your browser doesn't support the HTML5 canvas element!
56          </canvas>
57          <div>
58              <button id="xButton">Rotate X</button>
59              <button id="yButton">Rotate Y</button>
60              <button id="zButton">Rotate Z</button>
61              <button id="buttonT">Toggle Rotate</button>
62          </div>
63      </body>
64  </html>
```

⊗ 0 ⚠ 0 Ln 61, Col 47 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.js > generateHexaPyramid
1 var gl;
2 var points = [];
3 var colors = [];
4
5 var axis = 0;
6 var theta = [0, 0, 0];
7 var thetaLoc;
8
9 var rotation = false;
10
11 window.onload = function init()
12 {
13 var canvas = document.getElementById("gl-canvas");
14
15 gl = WebGLUtils.setupWebGL(canvas);
16 if(!gl) {
17 alert("WebGL isn't available!");
18 }
19
20 generateColorCube();
21 generateHexaPyramid();
22
23 // Configure WebGL
24 gl.viewport(0, 0, canvas.width, canvas.height);
25 gl.clearColor(0.9, 0.9, 0.9, 1.0);
26
27 // Enable hidden-surface removal
28 gl.enable(gl.DEPTH_TEST);
29
30 // Load shaders and initialize attribute buffers
31 var program = initShaders(gl, "vertex-shader", "fragment-shader");
32 gl.useProgram(program);
33
34 // Load the data into the GPU
35 var bufferId = gl.createBuffer();
⊗ 0 ⚠ 0
Ln 183, Col 33 Spaces: 4 UTF-8 CRLF JavaScript

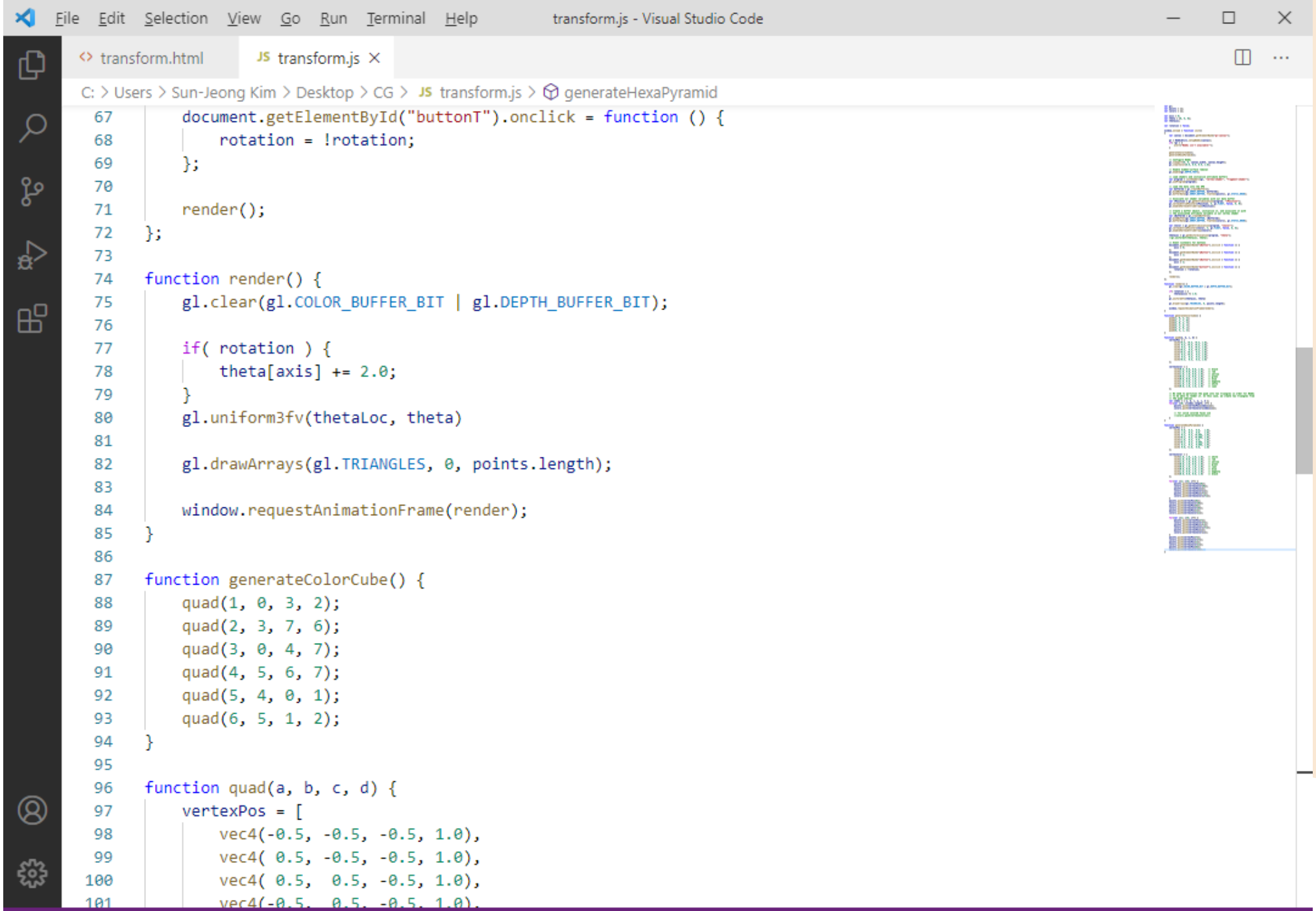
File Edit Selection View Go Run Terminal Help transform.js - Visual Studio Code

transform.html | transform.js

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

```
34    // Load the data into the GPU
35    var bufferId = gl.createBuffer();
36    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
37    gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
38
39    // Associate our shader variables with our data buffer
40    var vPosition = gl.getAttribLocation(program, "vPosition");
41    gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
42    gl.enableVertexAttribArray(vPosition);
43
44    // Create a buffer object, initialize it, and associate it with
45    // the associated attribute variable in our vertex shader
46    var cBufferId = gl.createBuffer();
47    gl.bindBuffer(gl.ARRAY_BUFFER, cBufferId);
48    gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
49
50    var vColor = gl.getAttribLocation(program, "vColor");
51    gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
52    gl.enableVertexAttribArray(vColor);
53
54    thetaLoc = gl.getUniformLocation(program, "theta");
55    //gl.uniform3fv(thetaLoc, theta);
56
57    // Event listeners for buttons
58    document.getElementById("xButton").onclick = function () {
59        axis = 0;
60    };
61    document.getElementById("yButton").onclick = function () {
62        axis = 1;
63    };
64    document.getElementById("zButton").onclick = function () {
65        axis = 2;
66    };
67    document.getElementById("buttonT").onclick = function () {
68        rotation = !rotation;
```

⊗ 0 ⚠ 0 Ln 181, Col 2 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.js > generateHexaPyramid
document.getElementById("buttonT").onclick = function () {
rotation = !rotation;
};
render();
};
function render() {
gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
if(rotation) {
theta[axis] += 2.0;
}
gl.uniform3fv(thetaLoc, theta)
gl.drawArrays(gl.TRIANGLES, 0, points.length);
window.requestAnimationFrame(render);
}
function generateColorCube() {
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) {
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),
Ln 183, Col 33
Spaces: 4
UTF-8
CRLF
JavaScript

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

transform.html | transform.js

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

```
96  function quad(a, b, c, d) {
97      vertexPos = [
98          vec4(-0.5, -0.5, -0.5, 1.0),
99          vec4( 0.5, -0.5, -0.5, 1.0),
100         vec4( 0.5,  0.5, -0.5, 1.0),
101         vec4(-0.5,  0.5, -0.5, 1.0),
102         vec4(-0.5, -0.5,  0.5, 1.0),
103         vec4( 0.5, -0.5,  0.5, 1.0),
104         vec4( 0.5,  0.5,  0.5, 1.0),
105         vec4(-0.5,  0.5,  0.5, 1.0)
106     ];
107
108     vertexColor = [
109         vec4(0.0, 0.0, 0.0, 1.0),   // black
110         vec4(1.0, 0.0, 0.0, 1.0),   // red
111         vec4(1.0, 1.0, 0.0, 1.0),   // yellow
112         vec4(0.0, 1.0, 0.0, 1.0),   // green
113         vec4(0.0, 0.0, 1.0, 1.0),   // blue
114         vec4(1.0, 0.0, 1.0, 1.0),   // magenta
115         vec4(1.0, 1.0, 1.0, 1.0),   // white
116         vec4(0.0, 1.0, 1.0, 1.0)    // cyan
117     ];
118
119     // We need to partition the quad into two triangles in order for WebGL
120     // to be able to render it. In this case, we create two triangles from
121     // the quad indices.
122     var index = [ a, b, c, a, c, d ];
123     for(var i=0; i<index.length; i++) {
124         points.push(vertexPos[index[i]]);
125         colors.push(vertexColor[index[i]]);
126     }
127 }
128
129 function generateHexaPyramid() {
```

⊗ 0 ⚠ 0 Ln 181, Col 2 Spaces: 4 UTF-8 CRLF JavaScript

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

transform.html | transform.js

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

```
129  function generateHexaPyramid() {
130      vertexPos = [
131          vec4( 0.0,  0.5,  0.0,   1.0),
132          vec4( 1.0,  0.5,  0.0,   1.0),
133          vec4( 0.5,  0.5, -0.866, 1.0),
134          vec4(-0.5,  0.5, -0.866, 1.0),
135          vec4(-1.0,  0.5,  0.0,   1.0),
136          vec4(-0.5,  0.5,  0.866, 1.0),
137          vec4( 0.5,  0.5,  0.866, 1.0),
138          vec4( 0.0, -1.0,  0.0,   1.0)
139      ];
140
141      vertexColor = [
142          vec4(1.0, 1.0, 1.0, 1.0),    // white
143          vec4(1.0, 0.0, 0.0, 1.0),    // red
144          vec4(1.0, 1.0, 0.0, 1.0),    // yellow
145          vec4(0.0, 1.0, 0.0, 1.0),    // green
146          vec4(0.0, 1.0, 1.0, 1.0),    // cyan
147          vec4(0.0, 0.0, 1.0, 1.0),    // blue
148          vec4(1.0, 0.0, 1.0, 1.0),    // magenta
149          vec4(0.0, 0.0, 0.0, 1.0)     // black
150      ];
151
152      for(var i=1; i<6; i++) {
153          points.push(vertexPos[0]);
154          colors.push(vertexColor[0]);
155          points.push(vertexPos[i]);
156          colors.push(vertexColor[i]);
157          points.push(vertexPos[i+1]);
158          colors.push(vertexColor[i+1]);
159      }
160      points.push(vertexPos[0]);
161      colors.push(vertexColor[0]);
162      points.push(vertexPos[6]);
```

⊗ 0 ⚠ 0 Ln 181, Col 2 Spaces: 4 UTF-8 CRLF JavaScript

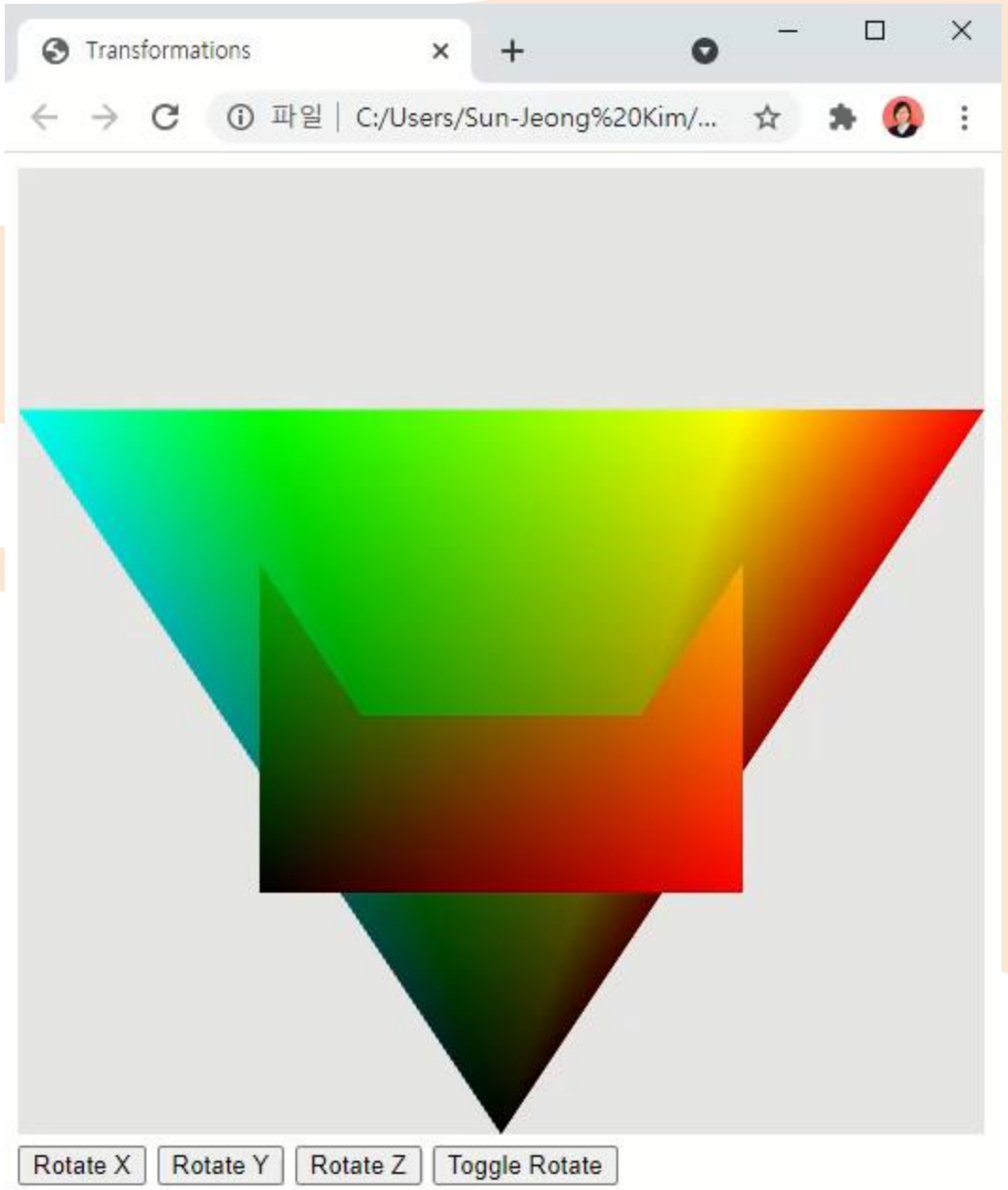
File Edit Selection View Go Run Terminal Help transform.js - Visual Studio Code

transform.html | transform.js

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

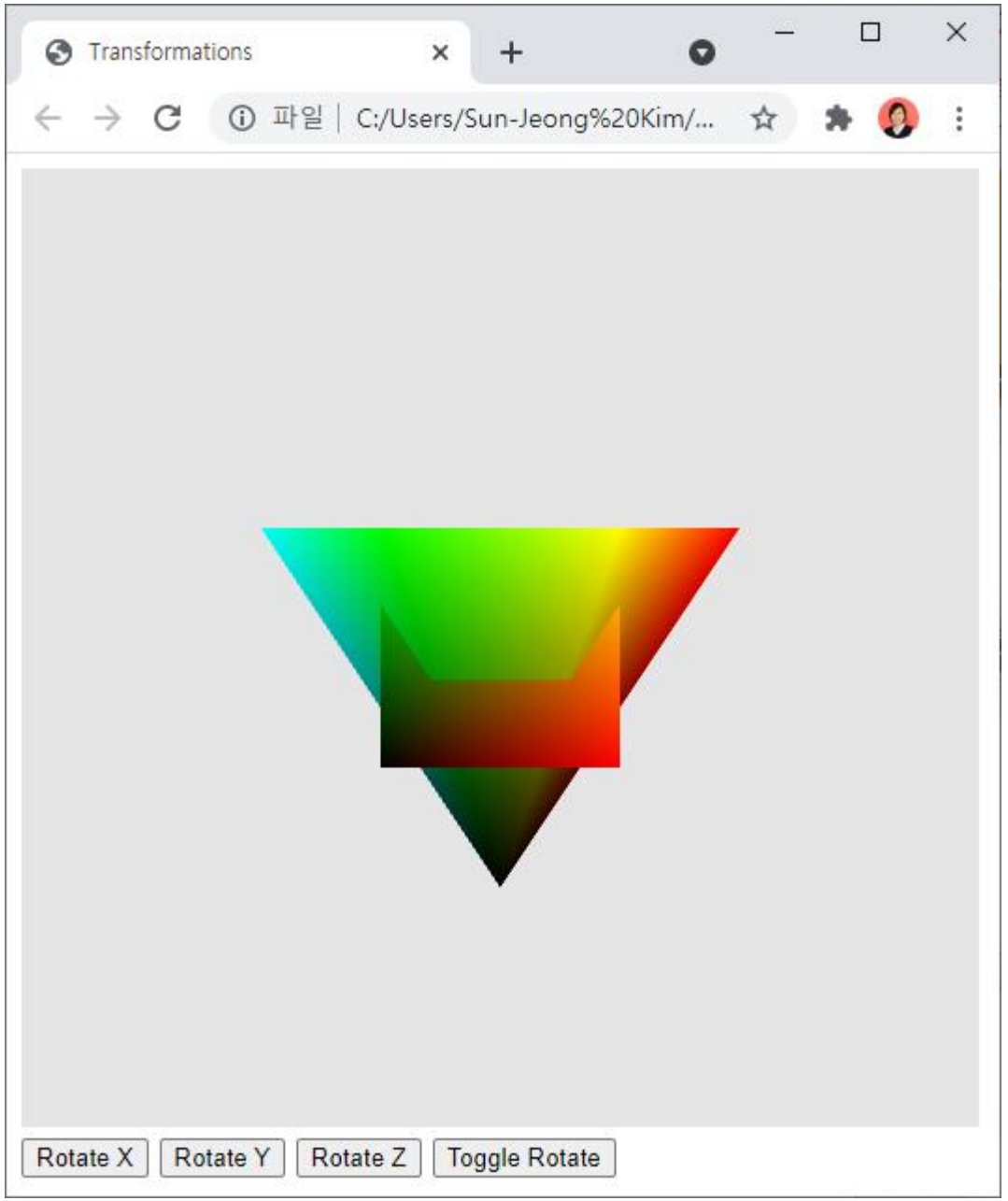
```
151
152     for(var i=1; i<6; i++) {
153         points.push(vertexPos[0]);
154         colors.push(vertexColor[0]);
155         points.push(vertexPos[i]);
156         colors.push(vertexColor[i]);
157         points.push(vertexPos[i+1]);
158         colors.push(vertexColor[i+1]);
159     }
160     points.push(vertexPos[0]);
161     colors.push(vertexColor[0]);
162     points.push(vertexPos[6]);
163     colors.push(vertexColor[6]);
164     points.push(vertexPos[1]);
165     colors.push(vertexColor[1]);
166
167     for(var i=1; i<6; i++) {
168         points.push(vertexPos[7]);
169         colors.push(vertexColor[7]);
170         points.push(vertexPos[i+1]);
171         colors.push(vertexColor[i+1]);
172         points.push(vertexPos[i]);
173         colors.push(vertexColor[i]);
174     }
175     points.push(vertexPos[7]);
176     colors.push(vertexColor[7]);
177     points.push(vertexPos[1]);
178     colors.push(vertexColor[1]);
179     points.push(vertexPos[6]);
180     colors.push(vertexColor[6]);
181 }
182
```

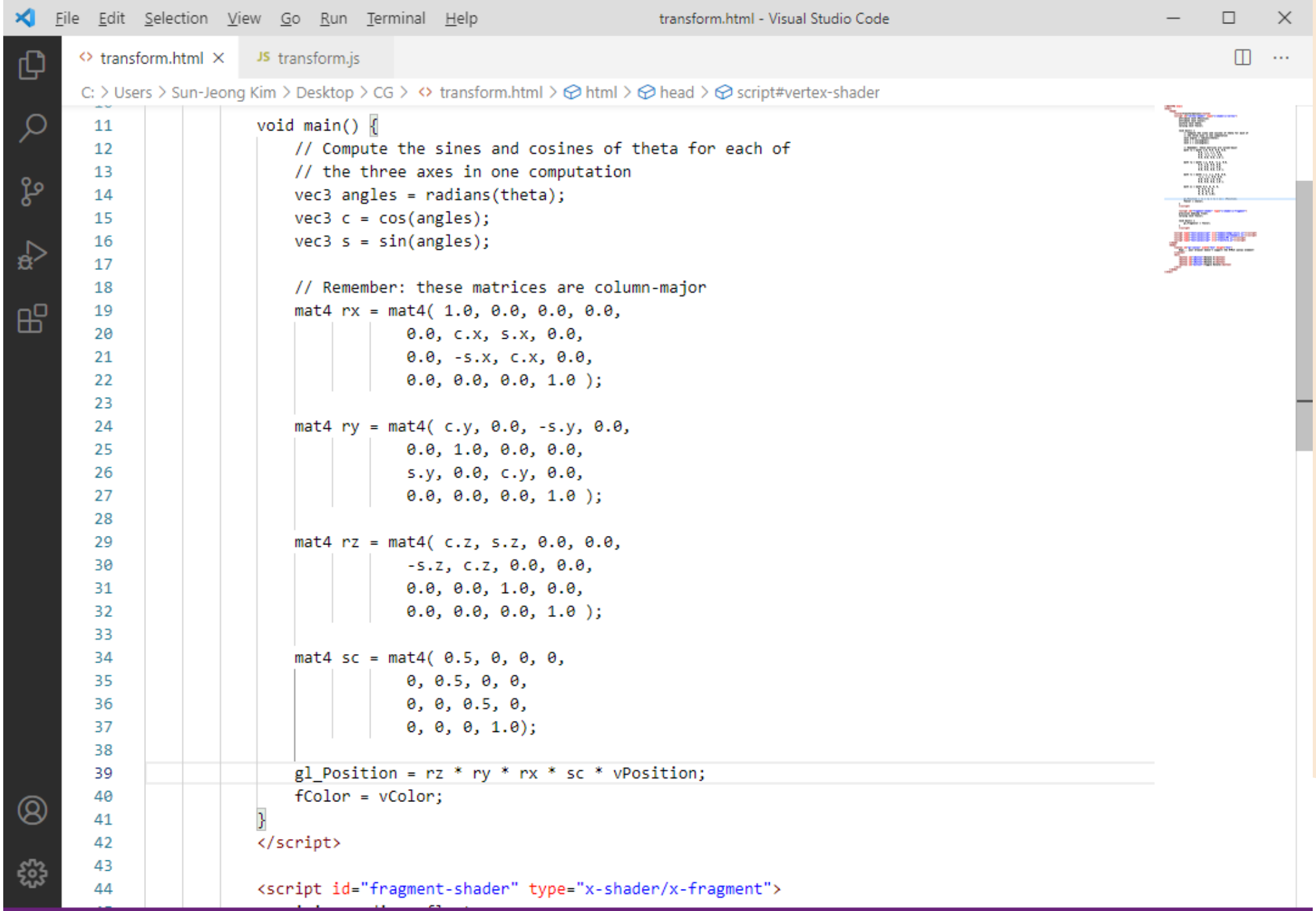
⊗ 0 ⚠ 0 Ln 181, Col 2 Spaces: 4 UTF-8 CRLF JavaScript


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

연습 문제 (2)

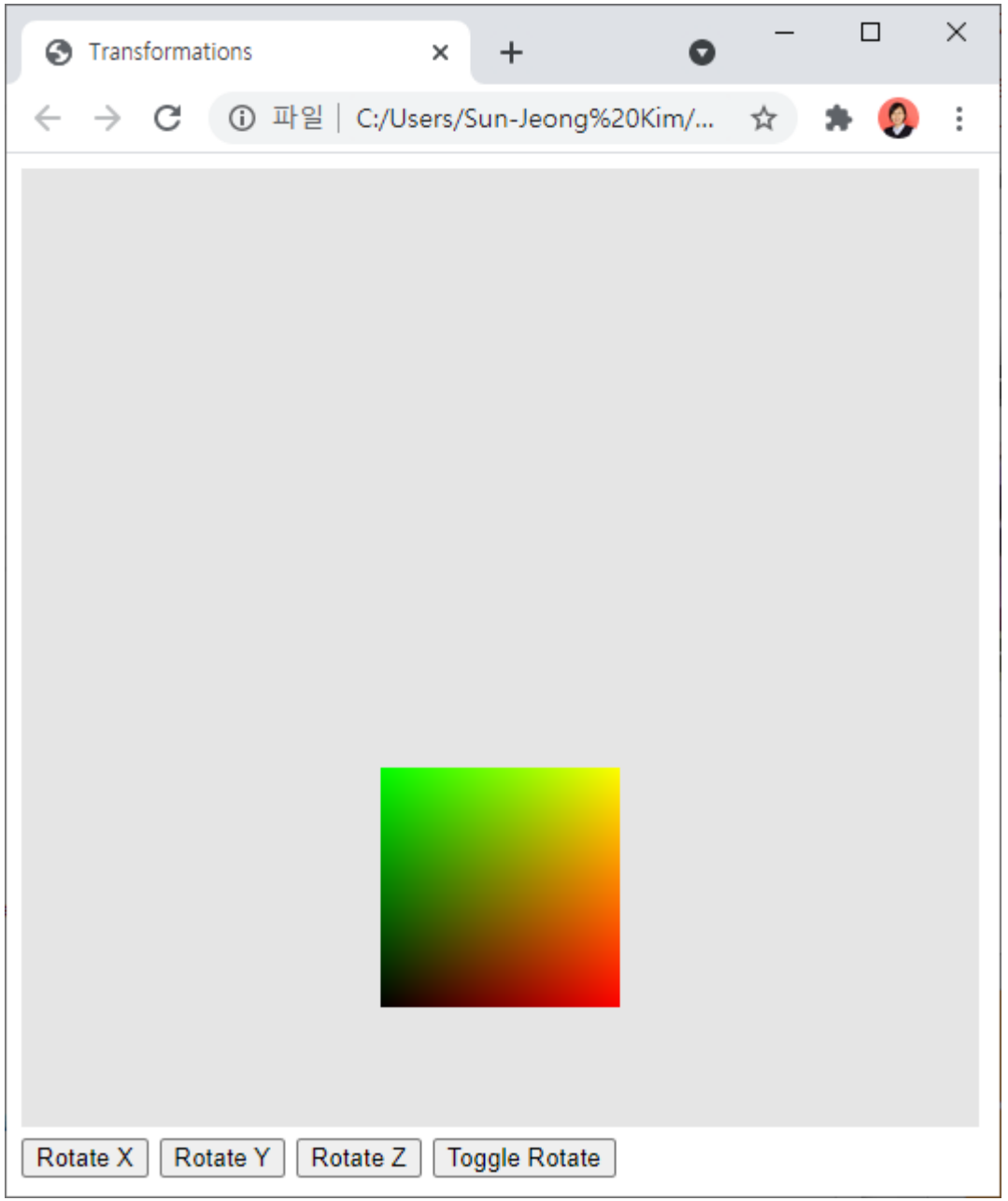
- 오브젝트의 크기를 반으로 줄이시오.

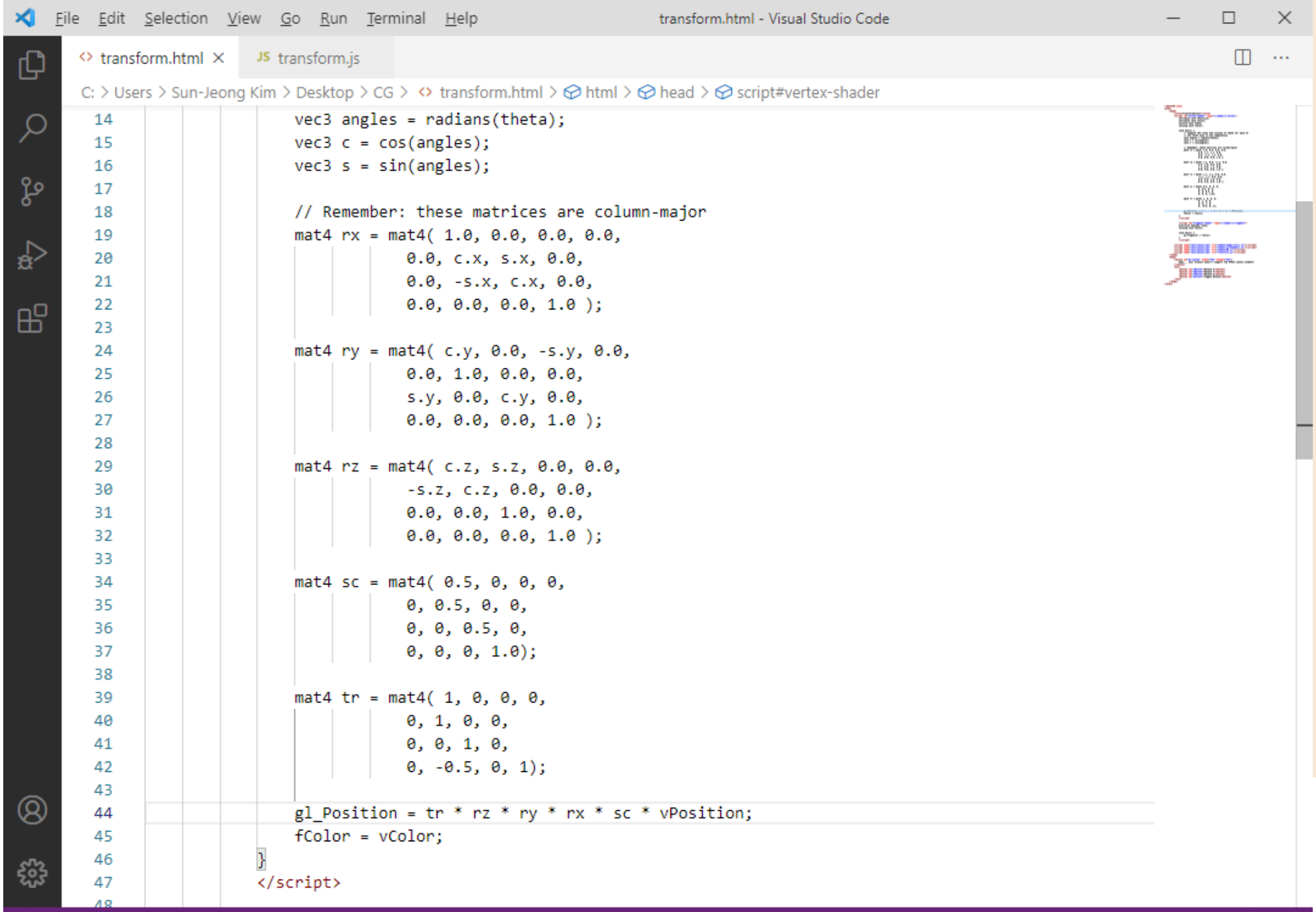


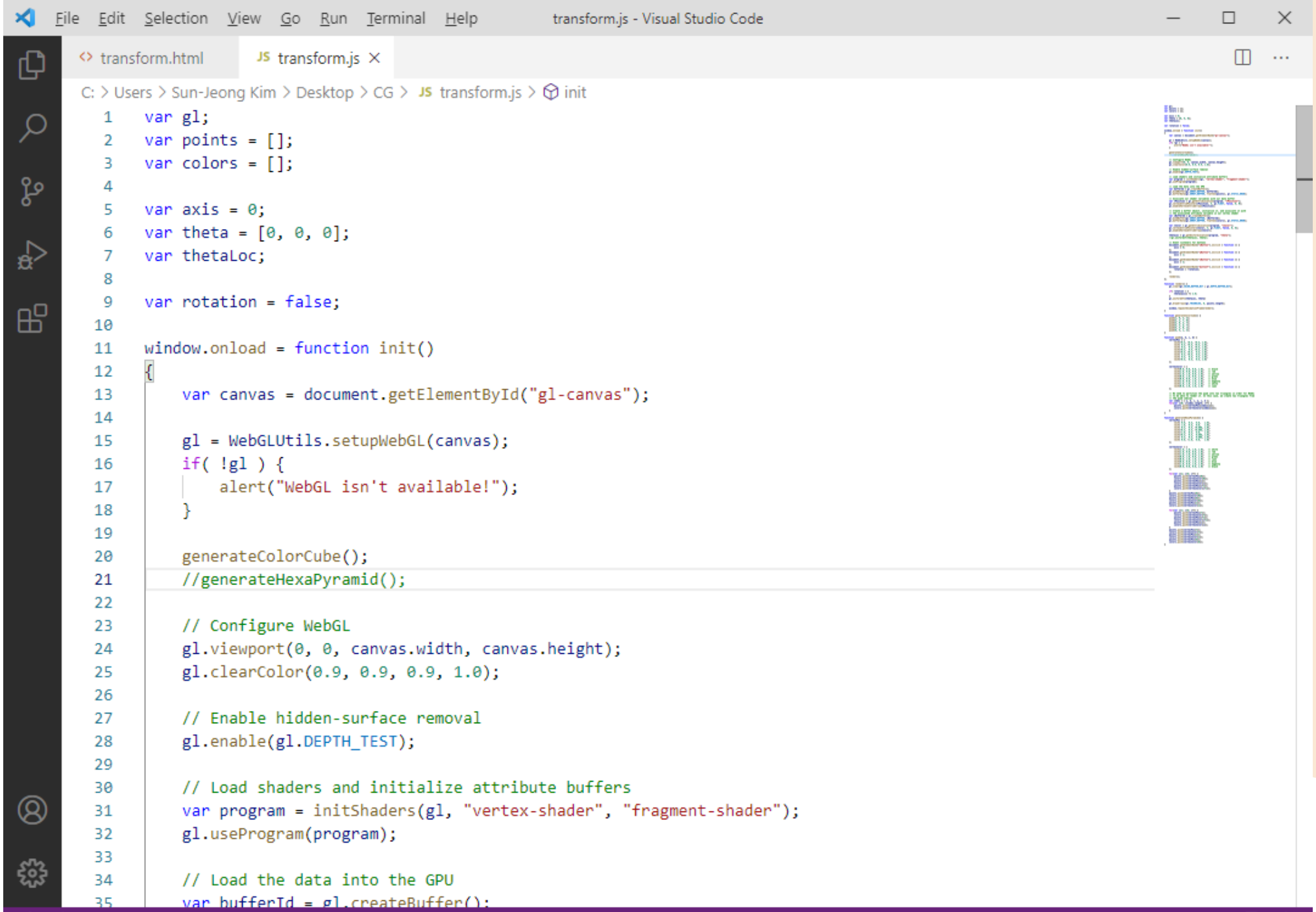

File Edit Selection View Go Run Terminal Help
transform.html - Visual Studio Code
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.html > html > head > script#vertex-shader
void main() {
// Compute the sines and cosines of theta for each of
// the three axes in one computation
vec3 angles = radians(theta);
vec3 c = cos(angles);
vec3 s = sin(angles);
// Remember: these matrices are column-major
mat4 rx = mat4(1.0, 0.0, 0.0, 0.0,
0.0, c.x, s.x, 0.0,
0.0, -s.x, c.x, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 ry = mat4(c.y, 0.0, -s.y, 0.0,
0.0, 1.0, 0.0, 0.0,
s.y, 0.0, c.y, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 rz = mat4(c.z, s.z, 0.0, 0.0,
-s.z, c.z, 0.0, 0.0,
0.0, 0.0, 1.0, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 sc = mat4(0.5, 0, 0, 0,
0, 0.5, 0, 0,
0, 0, 0.5, 0,
0, 0, 0, 1.0);
gl_Position = rz * ry * rx * sc * vPosition;
fColor = vColor;
}
</script>
<script id="fragment-shader" type="x-shader/x-fragment">
Ln 39, Col 48 Spaces: 4 UTF-8 CRLF HTML

연습 문제 (3)

- Cube를 y축 방향으로 -0.5 만큼 이동시키시오.

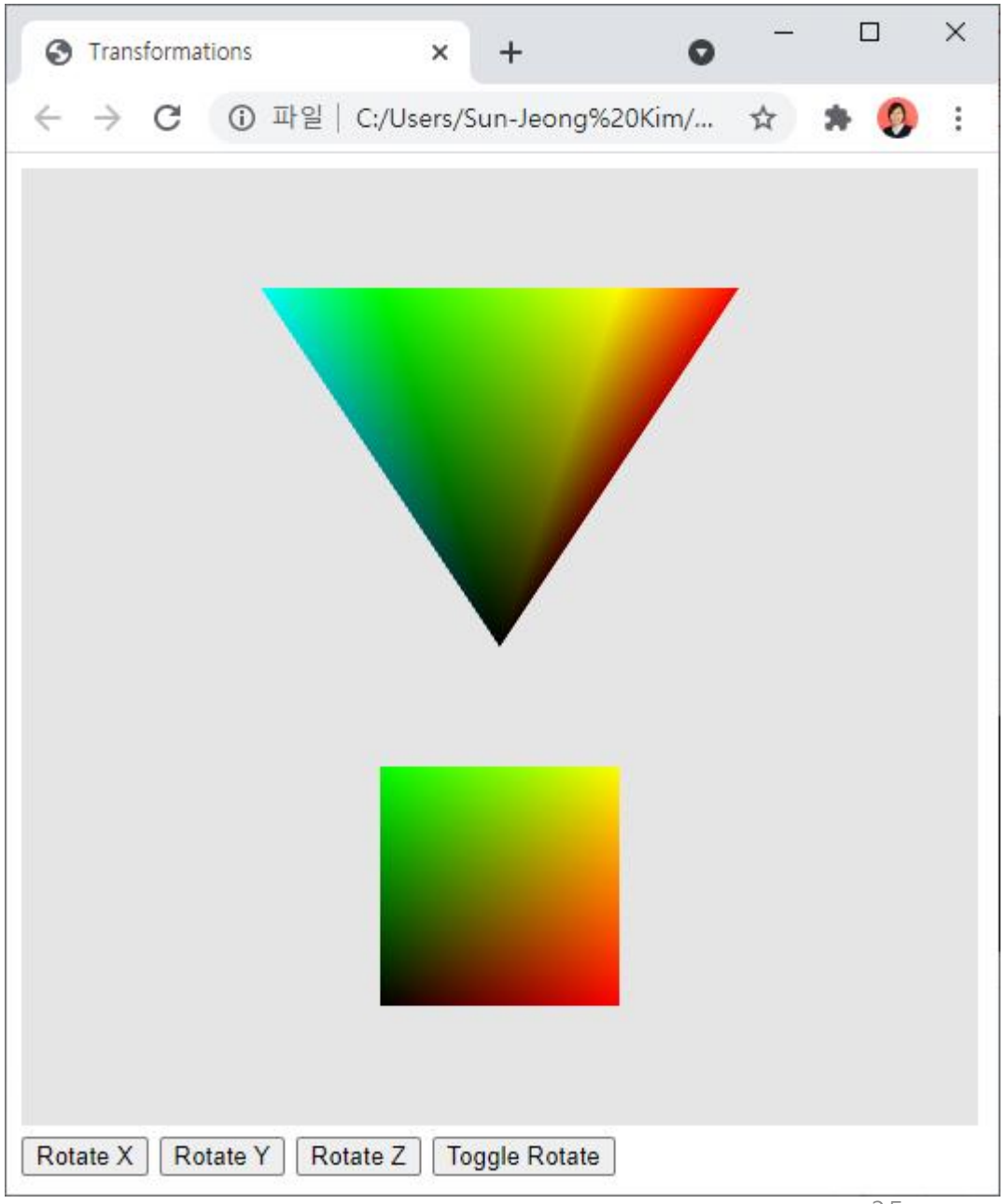



File Edit Selection View Go Run Terminal Help
transform.html - Visual Studio Code
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.html > html > head > script#vertex-shader
vec3 angles = radians(theta);
vec3 c = cos(angles);
vec3 s = sin(angles);
// Remember: these matrices are column-major
mat4 rx = mat4(1.0, 0.0, 0.0, 0.0,
0.0, c.x, s.x, 0.0,
0.0, -s.x, c.x, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 ry = mat4(c.y, 0.0, -s.y, 0.0,
0.0, 1.0, 0.0, 0.0,
s.y, 0.0, c.y, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 rz = mat4(c.z, s.z, 0.0, 0.0,
-s.z, c.z, 0.0, 0.0,
0.0, 0.0, 1.0, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 sc = mat4(0.5, 0, 0, 0,
0, 0.5, 0, 0,
0, 0, 0.5, 0,
0, 0, 0, 1.0);
mat4 tr = mat4(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, -0.5, 0, 1);
gl_Position = tr * rz * ry * rx * sc * vPosition;
fColor = vColor;
}
</script>
Ln 44, Col 36 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.js > init
1 var gl;
2 var points = [];
3 var colors = [];
4
5 var axis = 0;
6 var theta = [0, 0, 0];
7 var thetaLoc;
8
9 var rotation = false;
10
11 window.onload = function init()
12 {
13 var canvas = document.getElementById("gl-canvas");
14
15 gl = WebGLUtils.setupWebGL(canvas);
16 if(!gl) {
17 alert("WebGL isn't available!");
18 }
19
20 generateColorCube();
21 //generateHexaPyramid();
22
23 // Configure WebGL
24 gl.viewport(0, 0, canvas.width, canvas.height);
25 gl.clearColor(0.9, 0.9, 0.9, 1.0);
26
27 // Enable hidden-surface removal
28 gl.enable(gl.DEPTH_TEST);
29
30 // Load shaders and initialize attribute buffers
31 var program = initShaders(gl, "vertex-shader", "fragment-shader");
32 gl.useProgram(program);
33
34 // Load the data into the GPU
35 var bufferId = gl.createBuffer();
⊗ 0 ⚠ 0
Ln 21, Col 7 Spaces: 4 UTF-8 CRLF JavaScript

연습 문제 (4)

- Hexagonal Pyramid를 y축 방향으로 0.5만큼 이동 시키시오.



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

transform.html × transform.js

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

```
<!DOCTYPE html>
<html>
    <head>
        <title>Transformations</title>
        <script id="vertex-shader" type="x-shader/x-vertex">
            attribute vec4 vPosition;
            attribute vec4 vColor;
            uniform vec3 theta;
            uniform vec3 dist;
            varying vec4 fColor;

            void main() {
                // Compute the sines and cosines of theta for each of
                // the three axes in one computation
                vec3 angles = radians(theta);
                vec3 c = cos(angles);
                vec3 s = sin(angles);

                // Remember: these matrices are column-major
                mat4 rx = mat4( 1.0, 0.0, 0.0, 0.0,
                                0.0, c.x, s.x, 0.0,
                                0.0, -s.x, c.x, 0.0,
                                0.0, 0.0, 0.0, 1.0 );

                mat4 ry = mat4( c.y, 0.0, -s.y, 0.0,
                                0.0, 1.0, 0.0, 0.0,
                                s.y, 0.0, c.y, 0.0,
                                0.0, 0.0, 0.0, 1.0 );

                mat4 rz = mat4( c.z, s.z, 0.0, 0.0,
                                -s.z, c.z, 0.0, 0.0,
                                0.0, 0.0, 1.0, 0.0,
                                0.0, 0.0, 0.0, 1.0 );

                mat4 sc = mat4( 0.5, 0, 0, 0,
```

⊗ 0 ⚠ 0 Ln 42, Col 40 Spaces: 4 UTF-8 CRLF HTML

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

transform.html × transform.js

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

```
15              vec3 angles = radians(theta);
16              vec3 c = cos(angles);
17              vec3 s = sin(angles);
18
19              // Remember: these matrices are column-major
20              mat4 rx = mat4( 1.0, 0.0, 0.0, 0.0,
21                              0.0, c.x, s.x, 0.0,
22                              0.0, -s.x, c.x, 0.0,
23                              0.0, 0.0, 0.0, 1.0 );
24
25              mat4 ry = mat4( c.y, 0.0, -s.y, 0.0,
26                              0.0, 1.0, 0.0, 0.0,
27                              s.y, 0.0, c.y, 0.0,
28                              0.0, 0.0, 0.0, 1.0 );
29
30              mat4 rz = mat4( c.z, s.z, 0.0, 0.0,
31                              -s.z, c.z, 0.0, 0.0,
32                              0.0, 0.0, 1.0, 0.0,
33                              0.0, 0.0, 0.0, 1.0 );
34
35              mat4 sc = mat4( 0.5, 0, 0, 0,
36                              0, 0.5, 0, 0,
37                              0, 0, 0.5, 0,
38                              0, 0, 0, 1.0);
39
40              mat4 tr = mat4( 1, 0, 0, 0,
41                              0, 1, 0, 0,
42                              0, 0, 1, 0,
43                              dist.x, dist.y, dist.z, 1);
44
45              gl_Position = tr * rz * ry * rx * sc * vPosition;
46              fColor = vColor;
47          }
48          </script>
```

⊗ 0 ⚠ 0 Ln 43, Col 51 Spaces: 4 UTF-8 CRLF HTML

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

transform.html | transform.js

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

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

var axis = 0;
var theta = [0, 0, 0];
var thetaLoc;
var dist = [0, 0, 0];
var distLoc;

var rotation = false;

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

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

    generateColorCube();
    generateHexaPyramid();

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

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

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

⊗ 0 ⚠ 0 Ln 94, Col 15 Spaces: 4 UTF-8 CRLF JavaScript

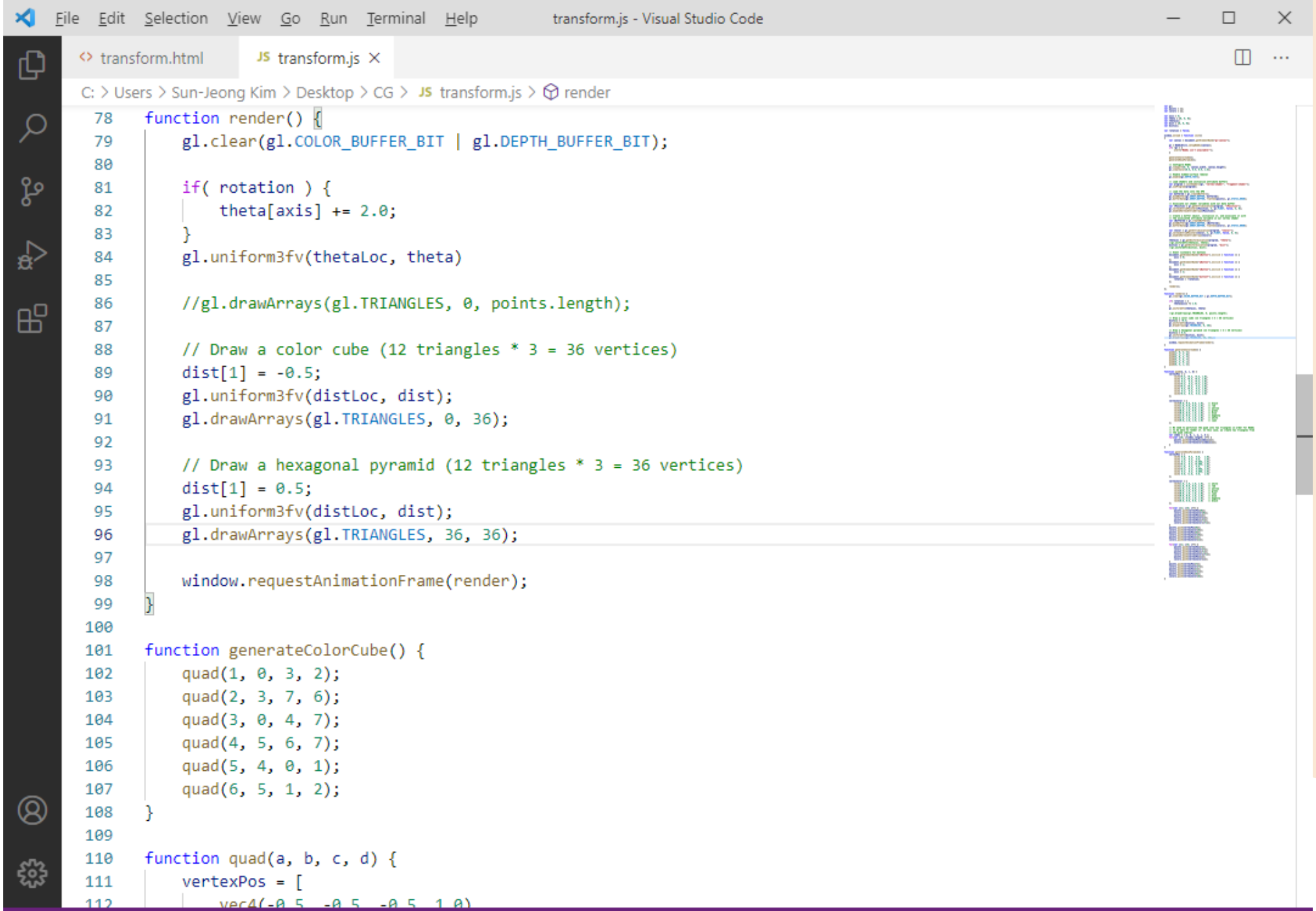
File Edit Selection View Go Run Terminal Help transform.js - Visual Studio Code

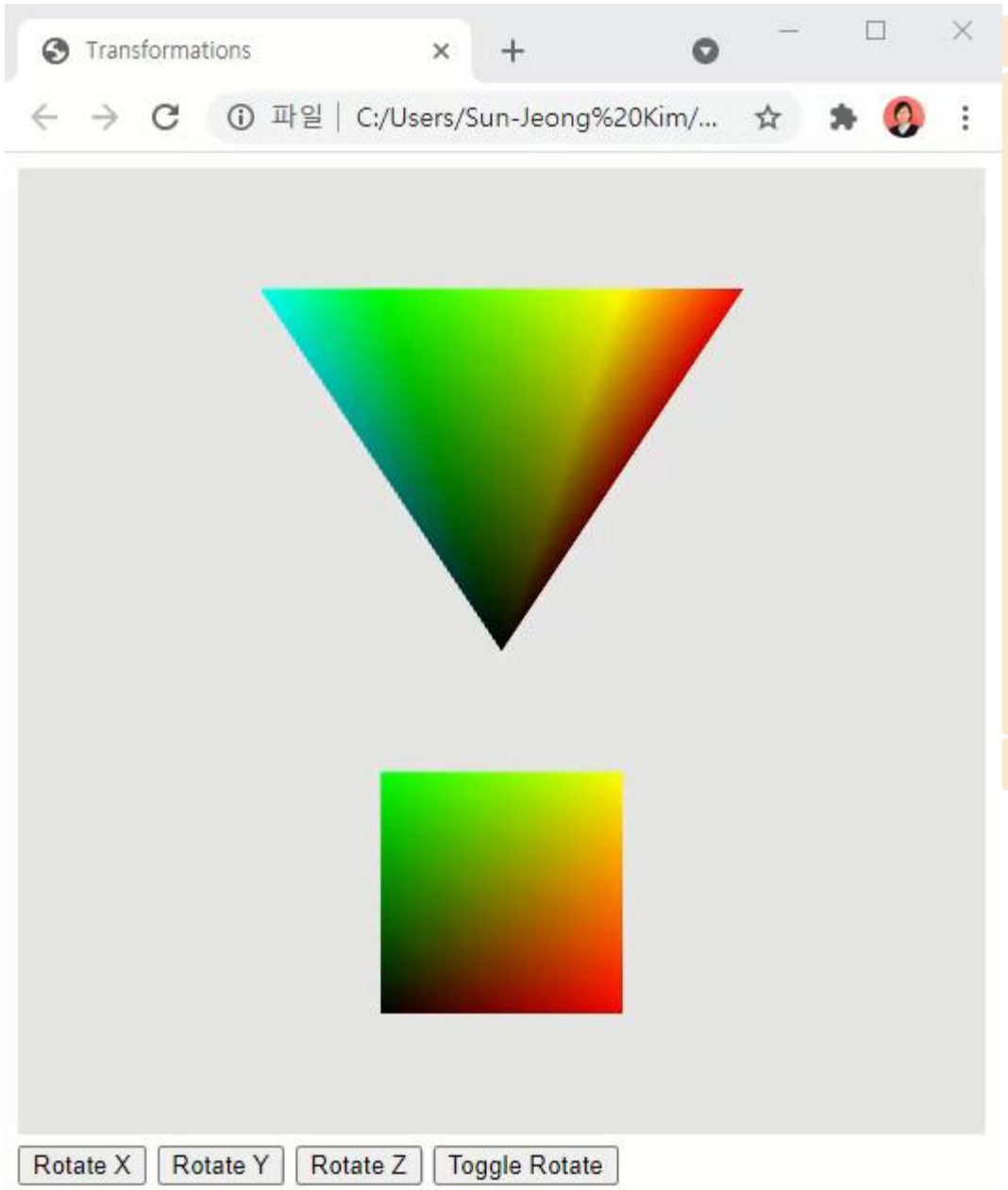
transform.html | transform.js

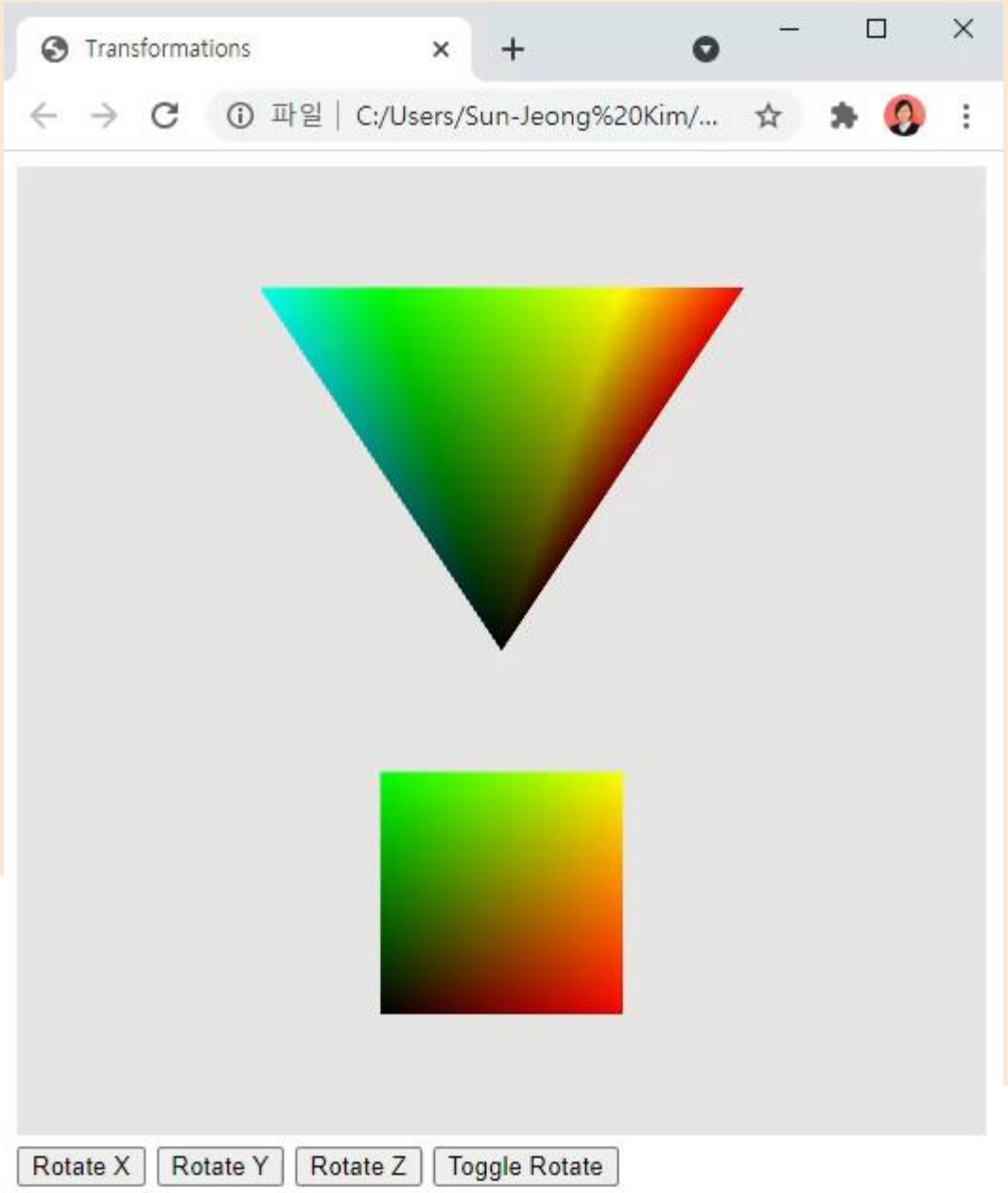
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.js > render

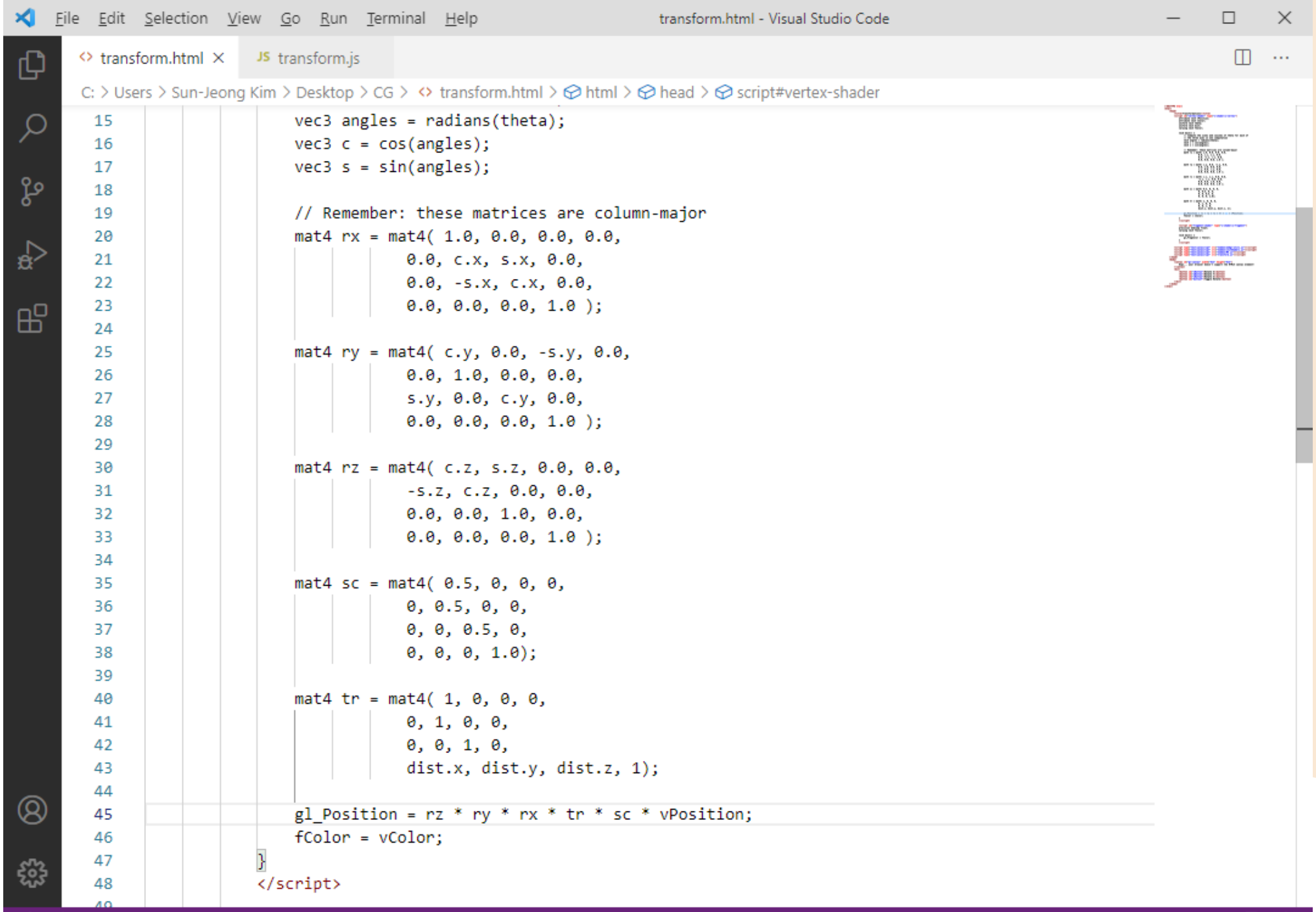
```
46      // Create a buffer object, initialize it, and associate it with
47      // the associated attribute variable in our vertex shader
48      var cBufferId = gl.createBuffer();
49      gl.bindBuffer(gl.ARRAY_BUFFER, cBufferId);
50      gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
51
52      var vColor = gl.getAttribLocation(program, "vColor");
53      gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
54      gl.enableVertexAttribArray(vColor);
55
56      thetaLoc = gl.getUniformLocation(program, "theta");
57      //gl.uniform3fv(thetaLoc, theta);
58      distLoc = gl.getUniformLocation(program, "dist");
59      //gl.uniform3fv(distLoc, dist);
60
61      // Event listeners for buttons
62      document.getElementById("xButton").onclick = function () {
63          axis = 0;
64      };
65      document.getElementById("yButton").onclick = function () {
66          axis = 1;
67      };
68      document.getElementById("zButton").onclick = function () {
69          axis = 2;
70      };
71      document.getElementById("buttonT").onclick = function () {
72          rotation = !rotation;
73      };
74
75      render();
76  };
77
78  function render() {
79      gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
80
```

⊗ 0 ⚠ 0 Ln 94, Col 15 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.js > render
78 function render() {
79 gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
80
81 if(rotation) {
82 theta[axis] += 2.0;
83 }
84 gl.uniform3fv(thetaLoc, theta)
85
86 //gl.drawArrays(gl.TRIANGLES, 0, points.length);
87
88 // Draw a color cube (12 triangles * 3 = 36 vertices)
89 dist[1] = -0.5;
90 gl.uniform3fv(distLoc, dist);
91 gl.drawArrays(gl.TRIANGLES, 0, 36);
92
93 // Draw a hexagonal pyramid (12 triangles * 3 = 36 vertices)
94 dist[1] = 0.5;
95 gl.uniform3fv(distLoc, dist);
96 gl.drawArrays(gl.TRIANGLES, 36, 36);
97
98 window.requestAnimationFrame(render);
99 }
100
101 function generateColorCube() {
102 quad(1, 0, 3, 2);
103 quad(2, 3, 7, 6);
104 quad(3, 0, 4, 7);
105 quad(4, 5, 6, 7);
106 quad(5, 4, 0, 1);
107 quad(6, 5, 1, 2);
108 }
109
110 function quad(a, b, c, d) {
111 vertexPos = [
112 vec4(-0.5, -0.5, -0.5, 1.0)
⊗ 0 ⚠ 0
Ln 96, Col 41 Spaces: 4 UTF-8 CRLF JavaScript

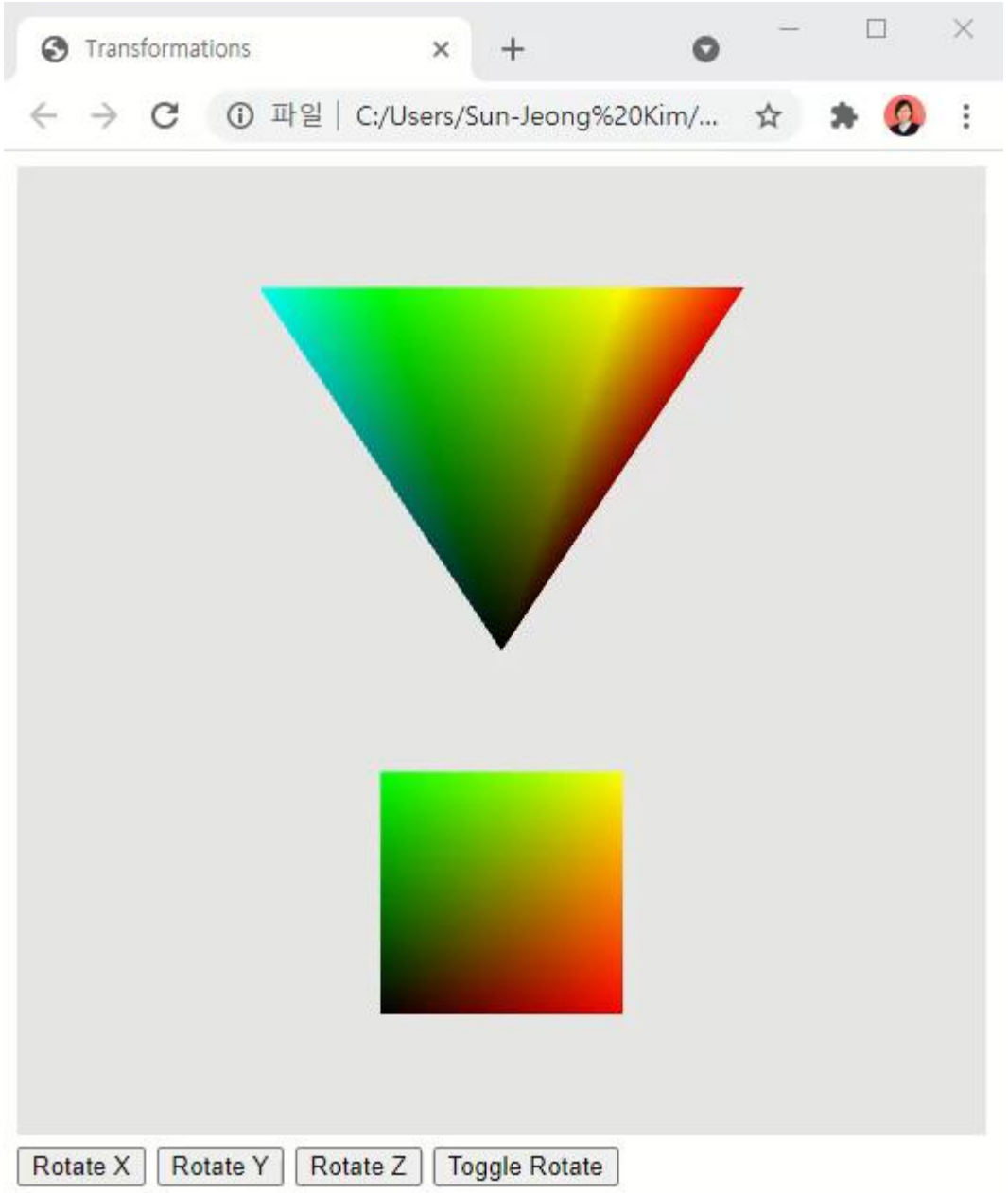

Transformations
파일 | C:/Users/Sun-Jeong%20Kim/...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate


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


File Edit Selection View Go Run Terminal Help
transform.html - Visual Studio Code
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.html > html > head > script#vertex-shader
vec3 angles = radians(theta);
vec3 c = cos(angles);
vec3 s = sin(angles);
// Remember: these matrices are column-major
mat4 rx = mat4(1.0, 0.0, 0.0, 0.0,
0.0, c.x, s.x, 0.0,
0.0, -s.x, c.x, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 ry = mat4(c.y, 0.0, -s.y, 0.0,
0.0, 1.0, 0.0, 0.0,
s.y, 0.0, c.y, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 rz = mat4(c.z, s.z, 0.0, 0.0,
-s.z, c.z, 0.0, 0.0,
0.0, 0.0, 1.0, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 sc = mat4(0.5, 0, 0, 0,
0, 0.5, 0, 0,
0, 0, 0.5, 0,
0, 0, 0, 1.0);
mat4 tr = mat4(1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
dist.x, dist.y, dist.z, 1);
gl_Position = rz * ry * rx * tr * sc * vPosition;
fColor = vColor;
}
</script>
Ln 45, Col 51 Spaces: 4 UTF-8 CRLF HTML

연습 문제 (5)

- 두 오브젝트가 서로 반대 방향으로 회전하도록 구현하시오.

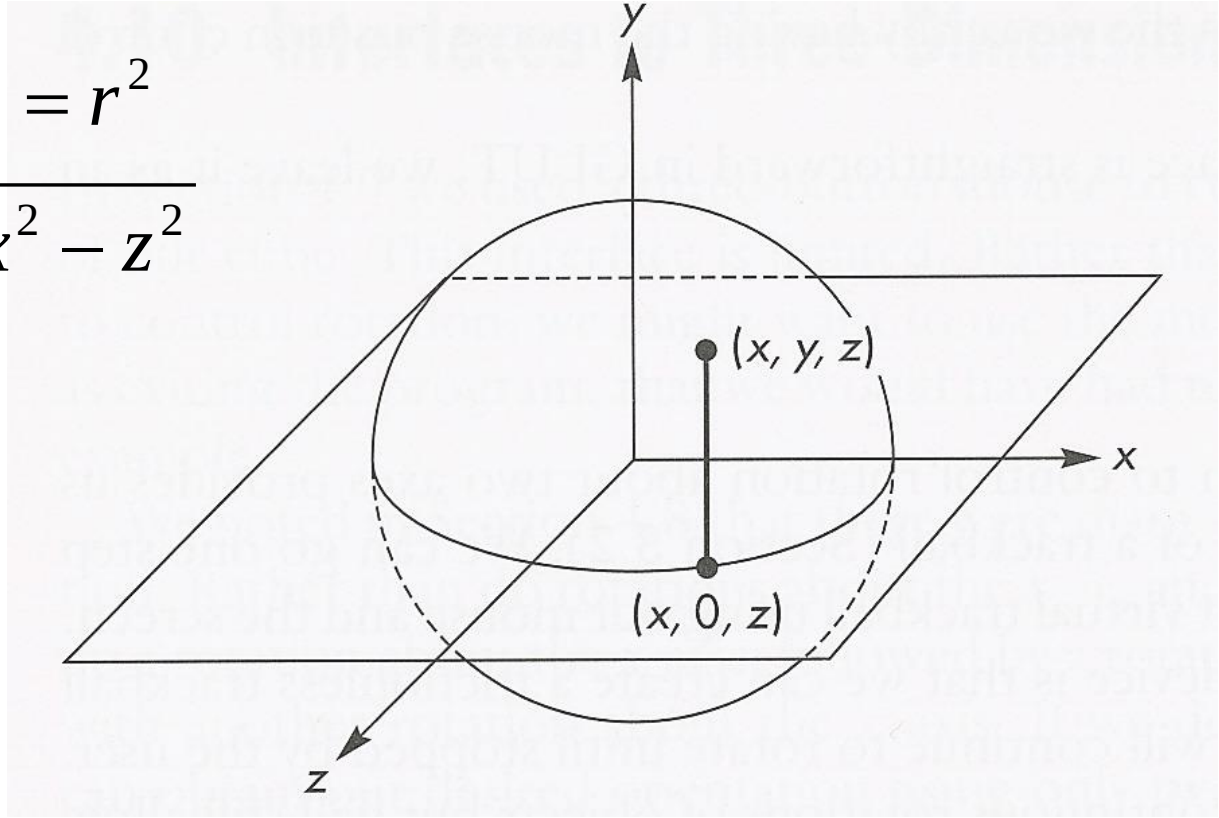


Rotation with a Virtual Trackball (1)

- Projection of the trackball position to the plane

$$x^2 + y^2 + z^2 = r^2$$

$$\therefore y = \sqrt{r^2 - x^2 - z^2}$$



Rotation with a Virtual Trackball (2)

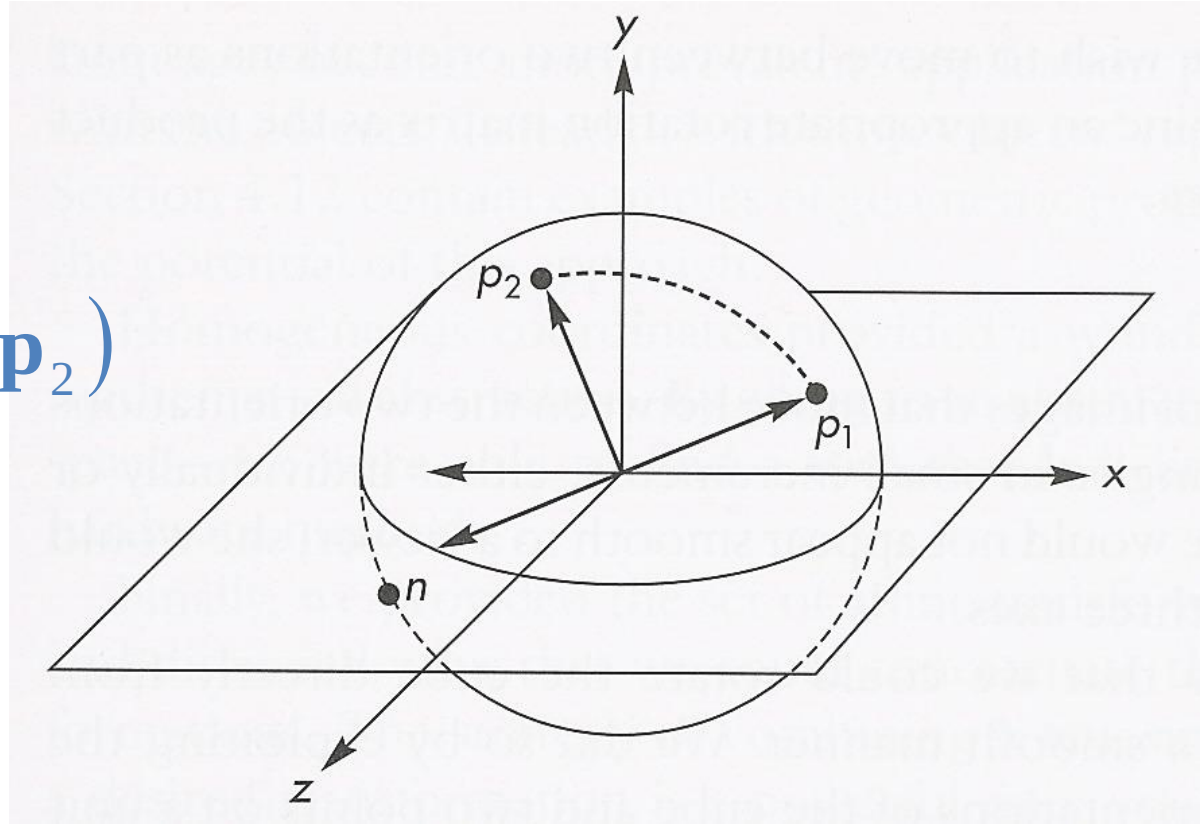
- Determination of the orientation of a plane

$$\mathbf{n} = \mathbf{p}_1 \times \mathbf{p}_2$$

- Rotation angle

$$\theta = \cos^{-1}\left(\mathbf{p}_1 \cdot \mathbf{p}_2\right)$$

➔ *Quaternions*



Rotations with Quaternions (1)

- Rotation about an arbitrary axis
  - Setting up a unit quaternion (**u**: unit vector)

$$s = \cos\frac{\theta}{2},\ \mathbf{v} = \mathbf{u}\sin\frac{\theta}{2} = (a, b, c)$$

  - Representing any point position **P** in quaternion notation (**p** = (x, y, z))

$$\mathbf{P} = (0,\ \mathbf{p})$$

  - Carrying out with the quaternion operation (q^{-1}=$(s, -\mathbf{v})$)

$$\mathbf{P}' = q\mathbf{P}q^{-1}$$

  - Producing the new quaternion

$$\mathbf{P}' = (0,\ \mathbf{p}')$$

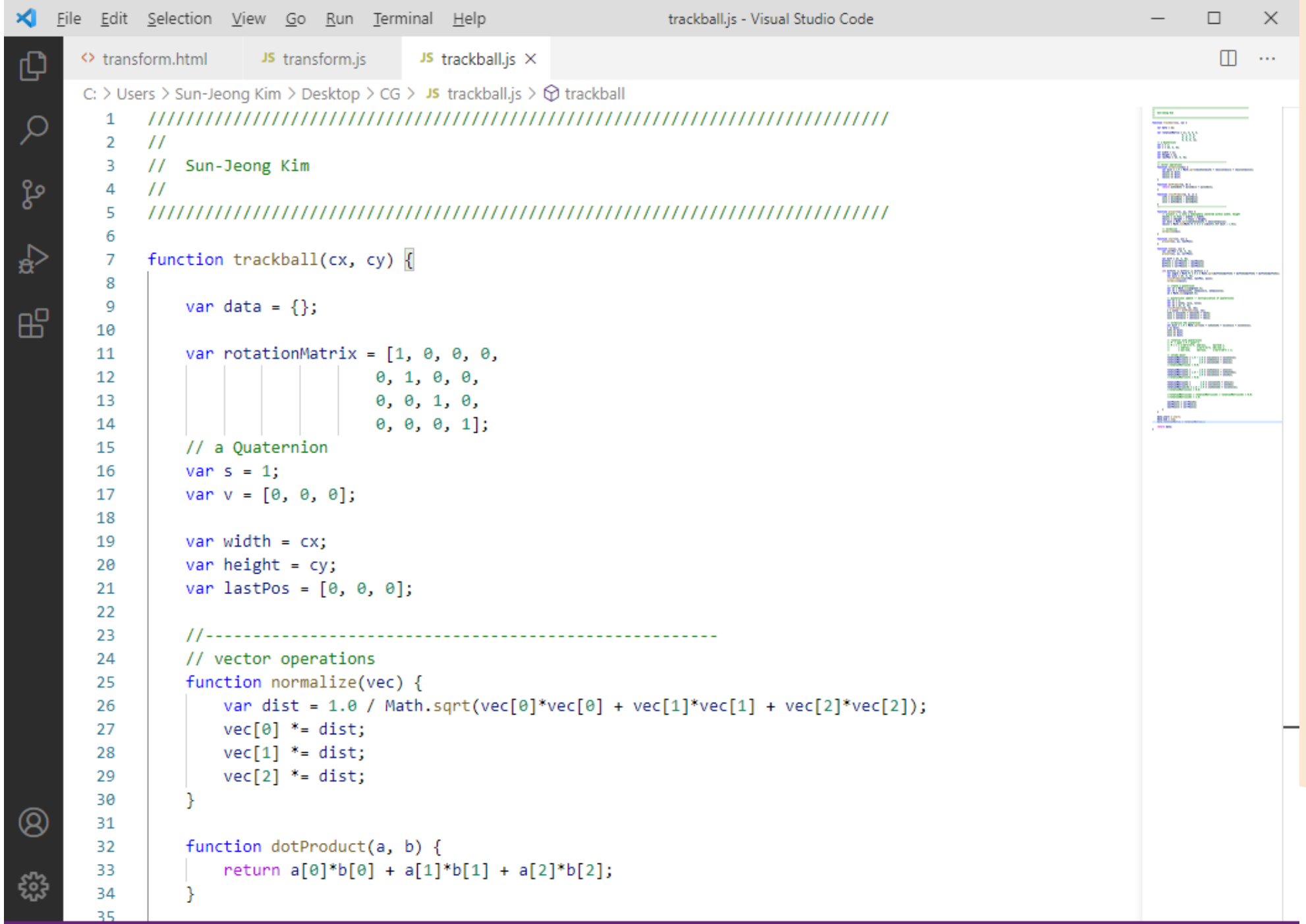
$$\mathbf{p}' = s^2\mathbf{p} + \mathbf{v}(\mathbf{p}\cdot\mathbf{v}) + 2s(\mathbf{v}\times\mathbf{p}) + \mathbf{v}\times(\mathbf{v}\times\mathbf{p})$$

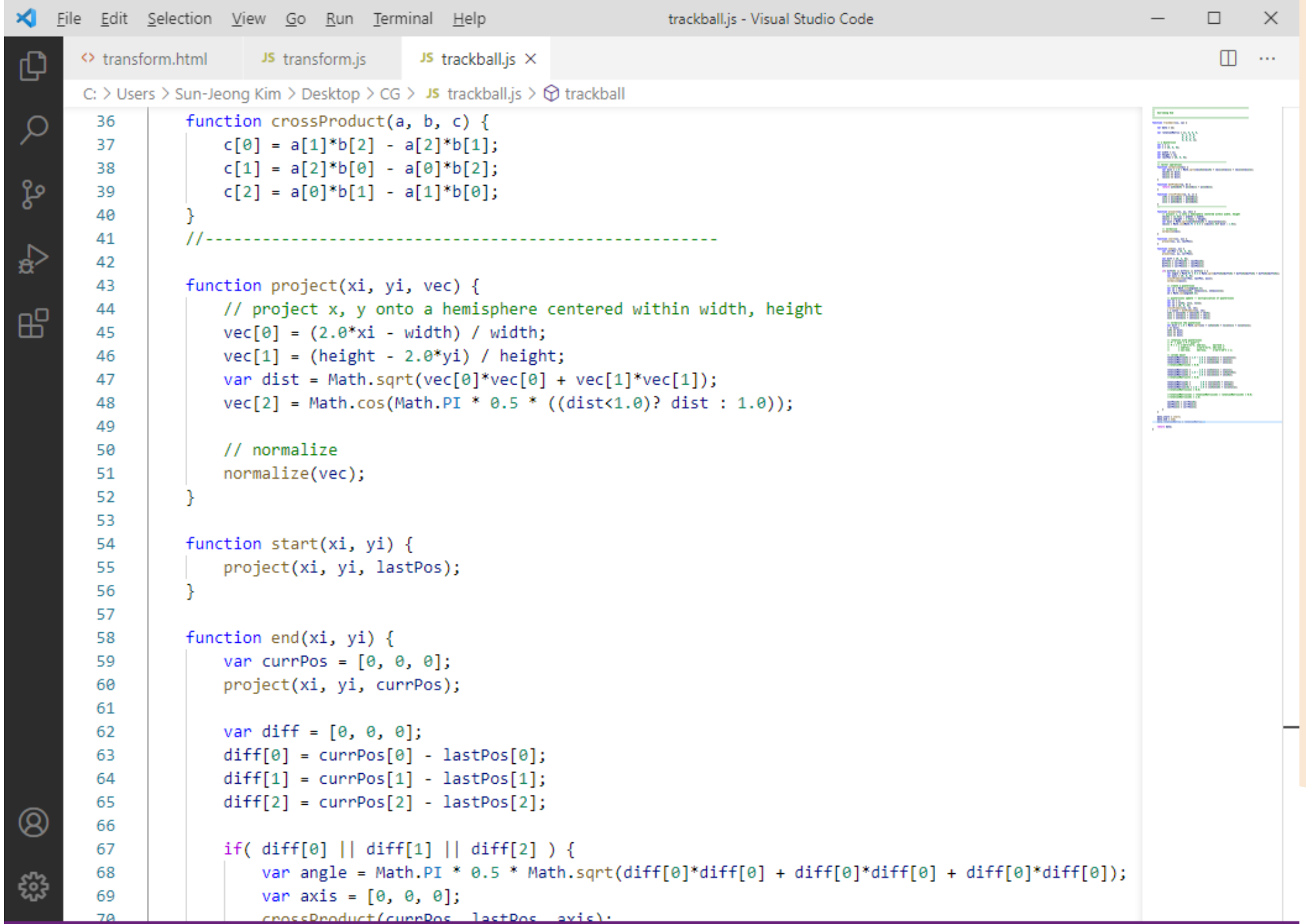
Rotations with Quaternions (2)

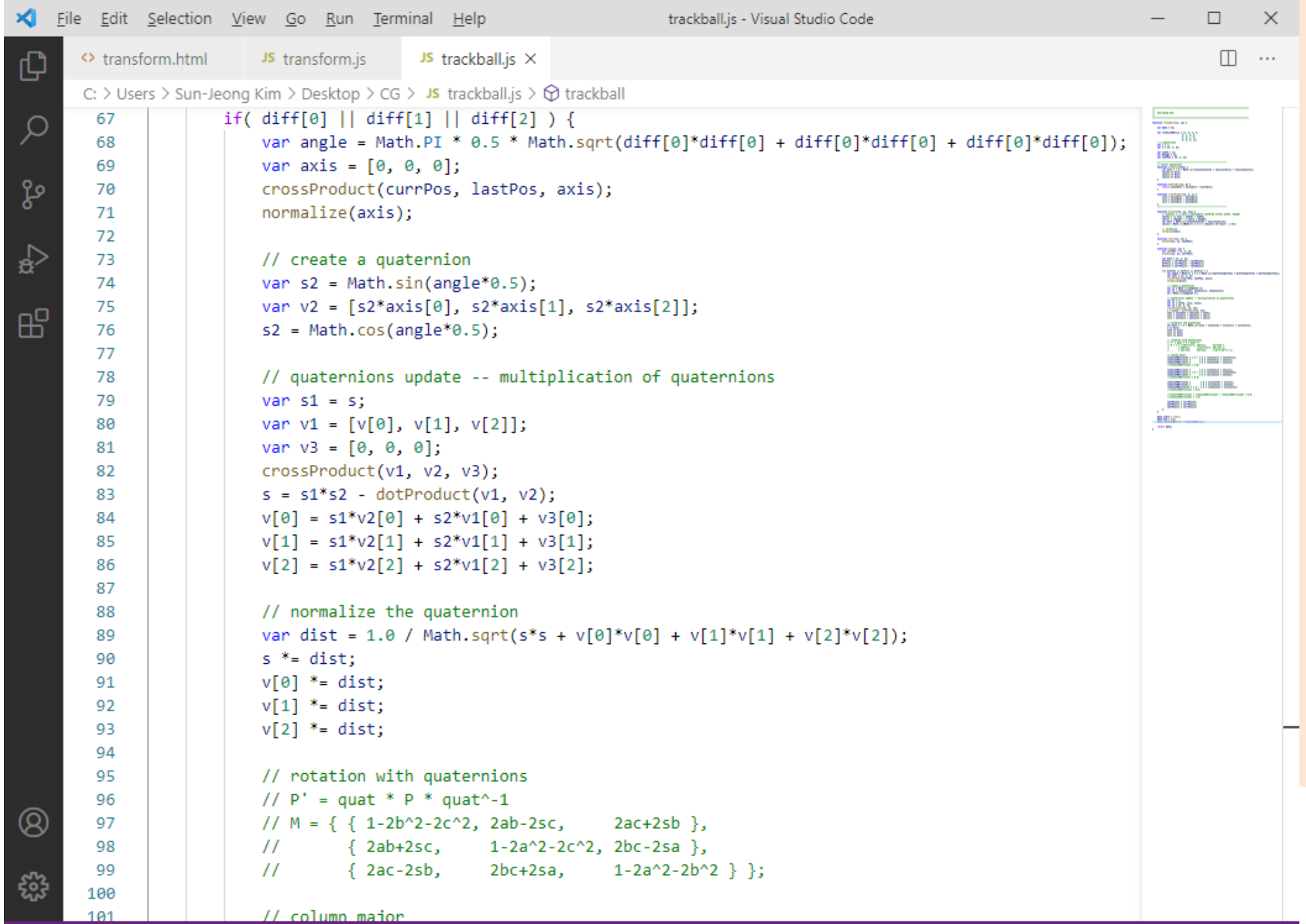
- Obtaining the rotation matrix by quaternion multiplication

$$\mathbf{M}_R(\theta) = \begin{bmatrix} 1-2b^2-2c^2 & 2ab-2sc & 2ac+2sb \\ 2ab+2sc & 1-2a^2-2c^2 & 2bc-2sa \\ 2ac-2sb & 2bc+2sa & 1-2a^2-2b^2 \end{bmatrix}$$

$$= \mathbf{R}_x(-\theta_x)\mathbf{R}_y(-\theta_y)\mathbf{R}_z(\theta)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)$$


File Edit Selection View Go Run Terminal Help
trackball.js - Visual Studio Code
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > trackball.js > trackball
//
//
// Sun-Jeong Kim
//
//
function trackball(cx, cy) {
var data = {};
var rotationMatrix = [1, 0, 0, 0,
0, 1, 0, 0,
0, 0, 1, 0,
0, 0, 0, 1];
// a Quaternion
var s = 1;
var v = [0, 0, 0];
var width = cx;
var height = cy;
var lastPos = [0, 0, 0];
//--
// vector operations
function normalize(vec) {
var dist = 1.0 / Math.sqrt(vec[0]*vec[0] + vec[1]*vec[1] + vec[2]*vec[2]);
vec[0] *= dist;
vec[1] *= dist;
vec[2] *= dist;
}
function dotProduct(a, b) {
return a[0]*b[0] + a[1]*b[1] + a[2]*b[2];
}
⊗ 0 ⚠ 0
Ln 128, Col 42
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
trackball.js - Visual Studio Code
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > trackball.js > trackball
36 function crossProduct(a, b, c) {
37 c[0] = a[1]*b[2] - a[2]*b[1];
38 c[1] = a[2]*b[0] - a[0]*b[2];
39 c[2] = a[0]*b[1] - a[1]*b[0];
40 }
41 //----------
42
43 function project(xi, yi, vec) {
44 // project x, y onto a hemisphere centered within width, height
45 vec[0] = (2.0*xi - width) / width;
46 vec[1] = (height - 2.0*yi) / height;
47 var dist = Math.sqrt(vec[0]*vec[0] + vec[1]*vec[1]);
48 vec[2] = Math.cos(Math.PI * 0.5 * ((dist<1.0)? dist : 1.0));
49
50 // normalize
51 normalize(vec);
52 }
53
54 function start(xi, yi) {
55 project(xi, yi, lastPos);
56 }
57
58 function end(xi, yi) {
59 var currPos = [0, 0, 0];
60 project(xi, yi, currPos);
61
62 var diff = [0, 0, 0];
63 diff[0] = currPos[0] - lastPos[0];
64 diff[1] = currPos[1] - lastPos[1];
65 diff[2] = currPos[2] - lastPos[2];
66
67 if(diff[0] || diff[1] || diff[2]) {
68 var angle = Math.PI * 0.5 * Math.sqrt(diff[0]*diff[0] + diff[0]*diff[0] + diff[0]*diff[0]);
69 var axis = [0, 0, 0];
Ln 128, Col 42 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
trackball.js - Visual Studio Code
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > trackball.js > trackball
67 if(diff[0] || diff[1] || diff[2]) {
68 var angle = Math.PI * 0.5 * Math.sqrt(diff[0]*diff[0] + diff[0]*diff[0] + diff[0]*diff[0]);
69 var axis = [0, 0, 0];
70 crossProduct(currPos, lastPos, axis);
71 normalize(axis);
72
73 // create a quaternion
74 var s2 = Math.sin(angle*0.5);
75 var v2 = [s2*axis[0], s2*axis[1], s2*axis[2]];
76 s2 = Math.cos(angle*0.5);
77
78 // quaternions update -- multiplication of quaternions
79 var s1 = s;
80 var v1 = [v[0], v[1], v[2]];
81 var v3 = [0, 0, 0];
82 crossProduct(v1, v2, v3);
83 s = s1*s2 - dotProduct(v1, v2);
84 v[0] = s1*v2[0] + s2*v1[0] + v3[0];
85 v[1] = s1*v2[1] + s2*v1[1] + v3[1];
86 v[2] = s1*v2[2] + s2*v1[2] + v3[2];
87
88 // normalize the quaternion
89 var dist = 1.0 / Math.sqrt(s*s + v[0]*v[0] + v[1]*v[1] + v[2]*v[2]);
90 s *= dist;
91 v[0] *= dist;
92 v[1] *= dist;
93 v[2] *= dist;
94
95 // rotation with quaternions
96 // P' = quat * P * quat^-1
97 // M = { { 1-2b^2-2c^2, 2ab-2sc, 2ac+2sb },
98 // { 2ab+2sc, 1-2a^2-2c^2, 2bc-2sa },
99 // { 2ac-2sb, 2bc+2sa, 1-2a^2-2b^2 } };
100
101 // column major
Ln 128, Col 42 Spaces: 4 UTF-8 CRLF JavaScript

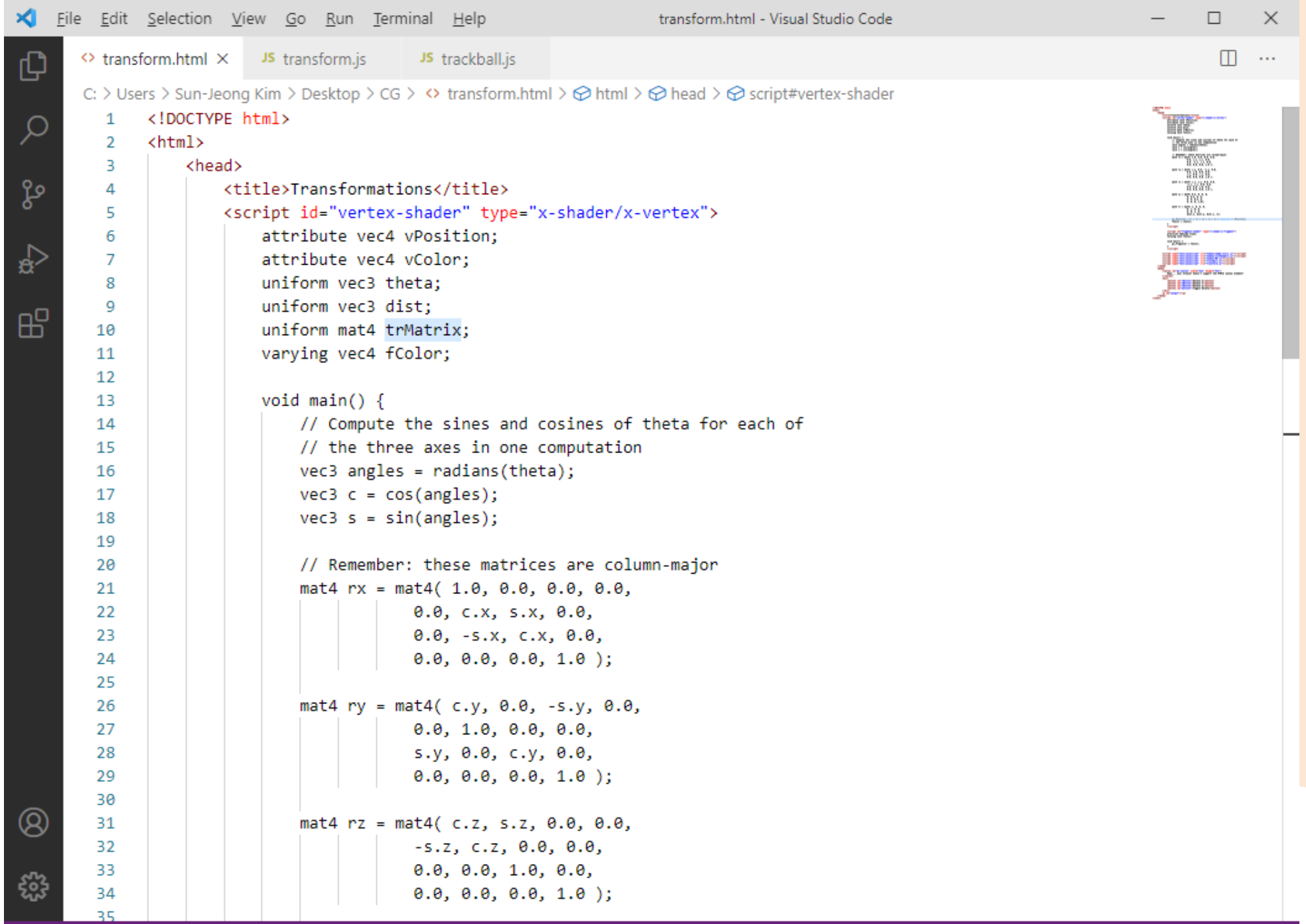
File Edit Selection View Go Run Terminal Help trackball.js - Visual Studio Code

transform.html transform.js trackball.js

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

```
100
101             // column major
102             rotationMatrix[0] = 1.0 - 2.0 * (v[1]*v[1] + v[2]*v[2]);
103             rotationMatrix[1] =       2.0 * (v[0]*v[1] + s*v[2]);
104             rotationMatrix[2] =       2.0 * (v[2]*v[0] - s*v[1]);
105             //rotationMatrix[3] = 0.0;
106
107             rotationMatrix[4] =       2.0 * (v[0]*v[1] - s*v[2]);
108             rotationMatrix[5] = 1.0 - 2.0 * (v[2]*v[2] + v[0]*v[0]);
109             rotationMatrix[6] =       2.0 * (v[1]*v[2] + s*v[0]);
110             //rotationMatrix[7] = 0.0;
111
112             rotationMatrix[8] =        2.0 * (v[2]*v[0] + s*v[1]);
113             rotationMatrix[9] =        2.0 * (v[1]*v[2] - s*v[0]);
114             rotationMatrix[10] = 1.0 - 2.0 * (v[0]*v[0] + v[1]*v[1]);
115             //rotationMatrix[11] = 0.0;
116
117             //rotationMatrix[12] = rotationMatrix[13] = rotationMatrix[14] = 0.0;
118             //rotationMatrix[15] = 1.0;
119
120             lastPos[0] = currPos[0];
121             lastPos[1] = currPos[1];
122             lastPos[2] = currPos[2];
123         }
124     }
125
126     data.start = start;
127     data.end = end;
128     data.rotationMatrix = rotationMatrix;
129
130     return data;
131 }
132
```

⊗ 0 ⚠ 0 Ln 128, Col 42 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
transform.html - Visual Studio Code
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.html > html > head > script#vertex-shader
<!DOCTYPE html>
<html>
<head>
<title>Transformations</title>
<script id="vertex-shader" type="x-shader/x-vertex">
attribute vec4 vPosition;
attribute vec4 vColor;
uniform vec3 theta;
uniform vec3 dist;
uniform mat4 trMatrix;
varying vec4 fColor;
void main() {
// Compute the sines and cosines of theta for each of
// the three axes in one computation
vec3 angles = radians(theta);
vec3 c = cos(angles);
vec3 s = sin(angles);
// Remember: these matrices are column-major
mat4 rx = mat4(1.0, 0.0, 0.0, 0.0,
0.0, c.x, s.x, 0.0,
0.0, -s.x, c.x, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 ry = mat4(c.y, 0.0, -s.y, 0.0,
0.0, 1.0, 0.0, 0.0,
s.y, 0.0, c.y, 0.0,
0.0, 0.0, 0.0, 1.0);
mat4 rz = mat4(c.z, s.z, 0.0, 0.0,
-s.z, c.z, 0.0, 0.0,
0.0, 0.0, 1.0, 0.0,
0.0, 0.0, 0.0, 1.0);
Ln 46, Col 64 (8 selected) Spaces: 4 UTF-8 CRLF HTML

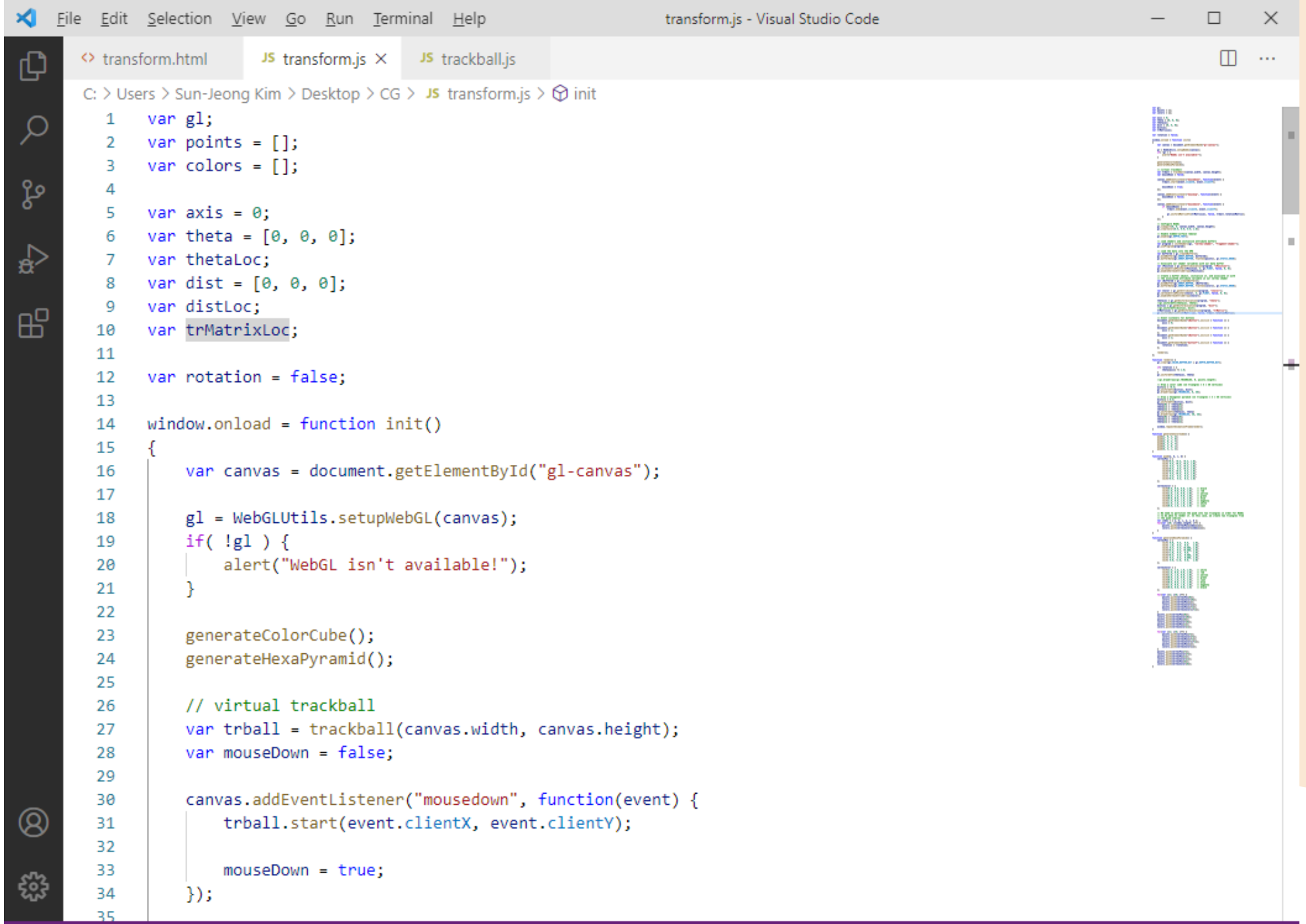
File Edit Selection View Go Run Terminal Help transform.html - Visual Studio Code

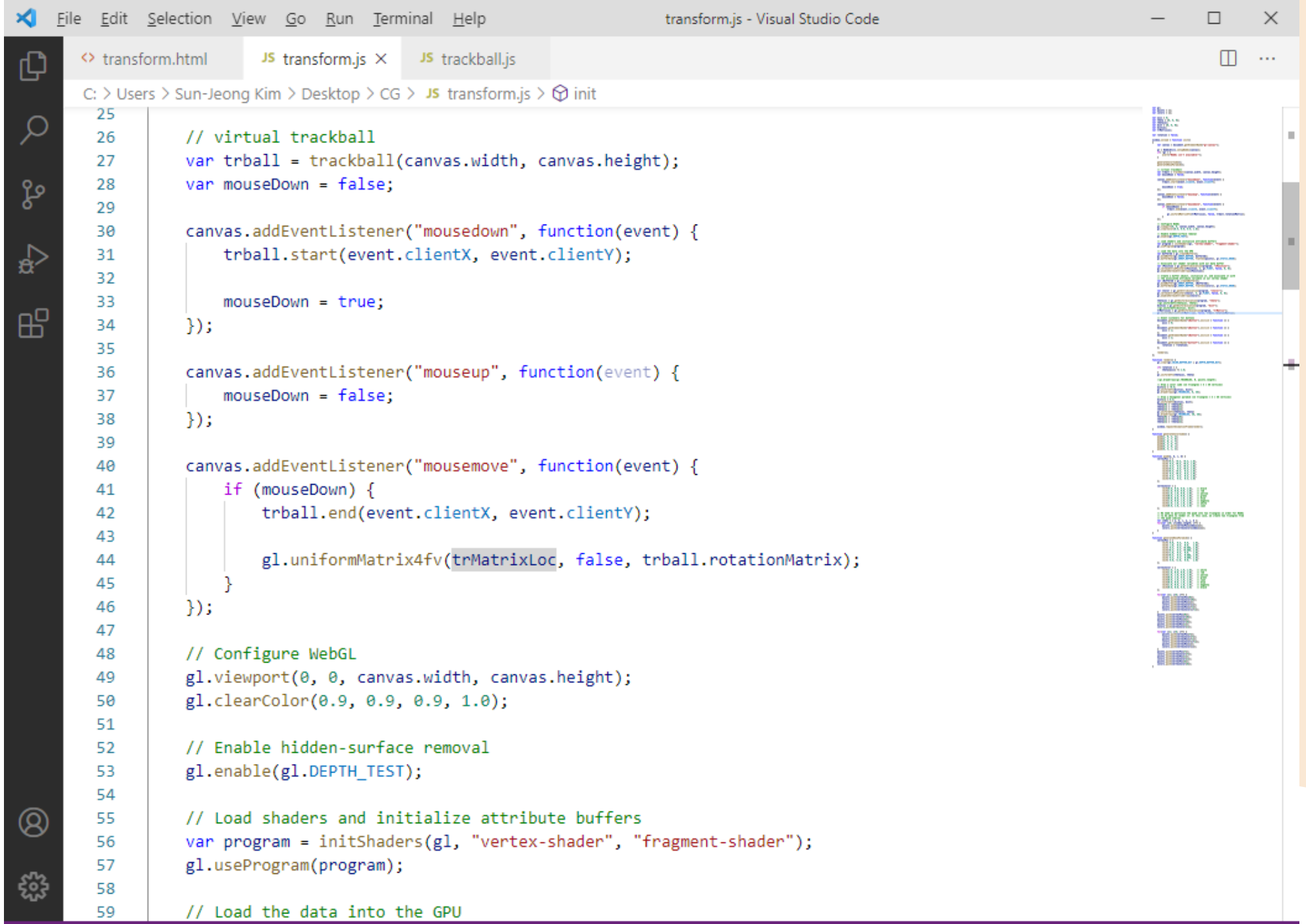
transform.html | transform.js | trackball.js

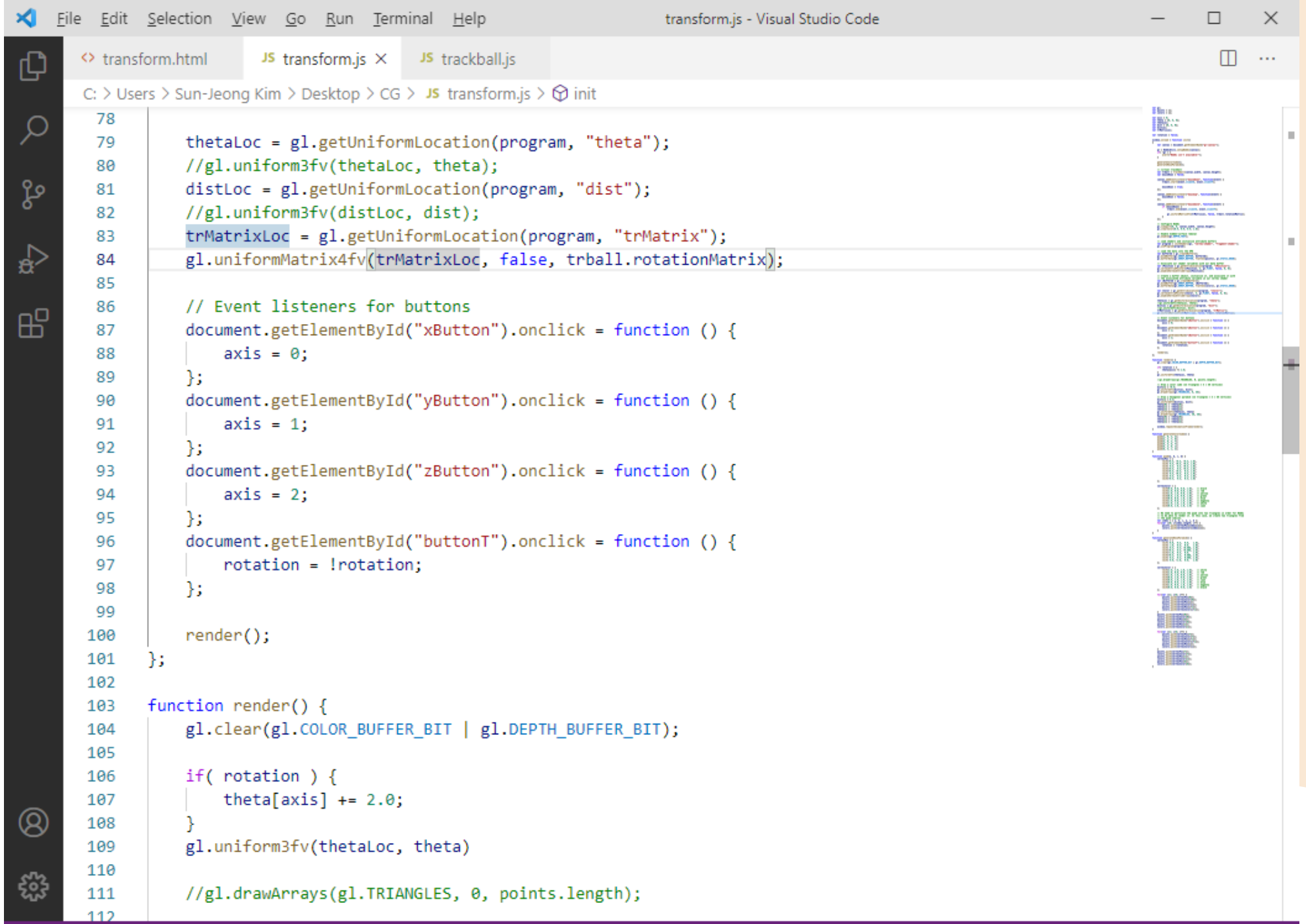
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.html > html > head > script#vertex-shader

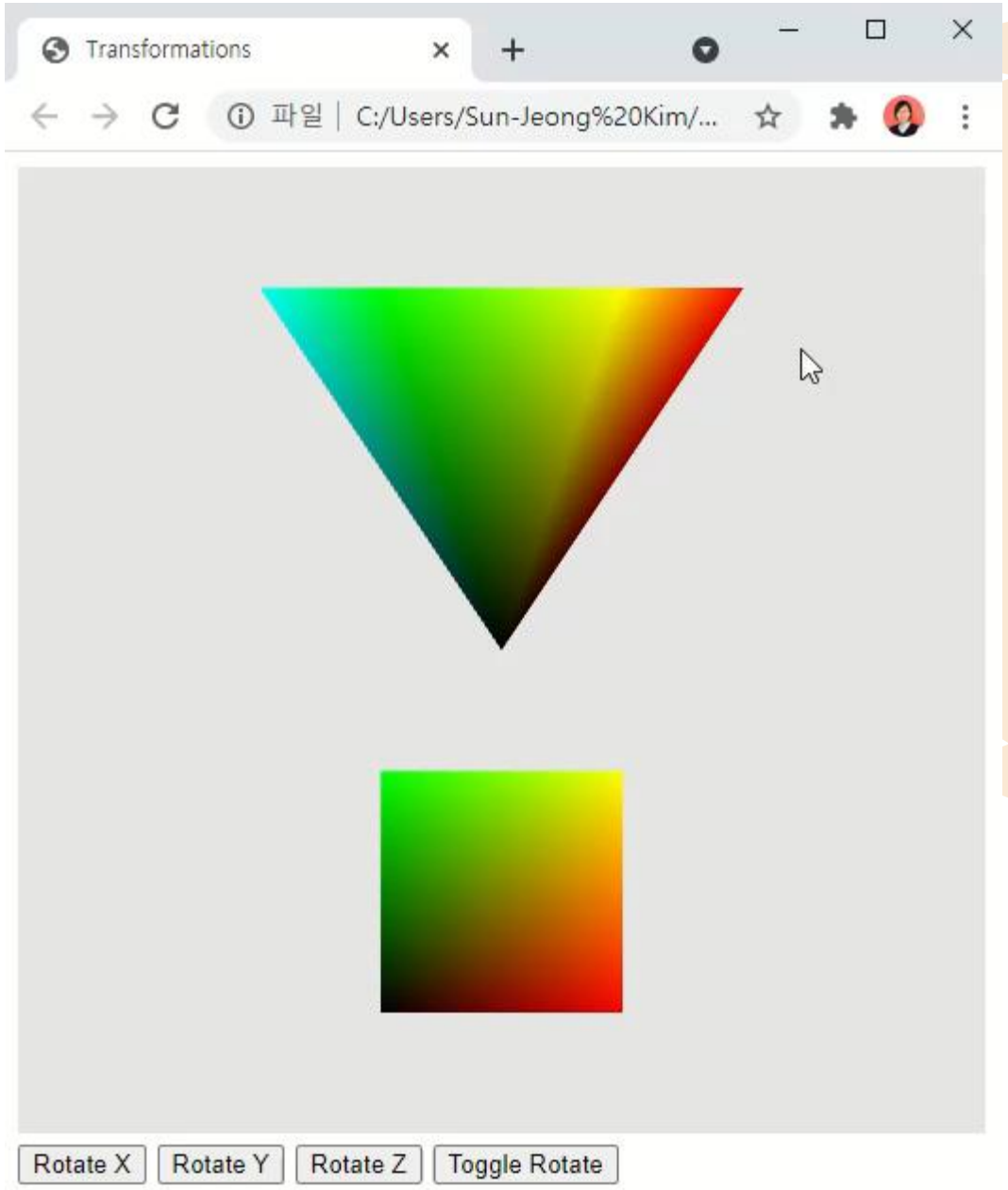
```
31              mat4 rz = mat4( c.z, s.z, 0.0, 0.0,
32                              -s.z, c.z, 0.0, 0.0,
33                              0.0, 0.0, 1.0, 0.0,
34                              0.0, 0.0, 0.0, 1.0 );
35
36              mat4 sc = mat4( 0.5, 0, 0, 0,
37                              0, 0.5, 0, 0,
38                              0, 0, 0.5, 0,
39                              0, 0, 0, 1.0);
40
41              mat4 tr = mat4( 1, 0, 0, 0,
42                              0, 1, 0, 0,
43                              0, 0, 1, 0,
44                              dist.x, dist.y, dist.z, 1);
45
46              gl_Position = tr * rz * ry * rx * sc * trMatrix * vPosition;
47              fColor = vColor;
48          }
49          </script>
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="transform.js"></script>
65  </head>
```

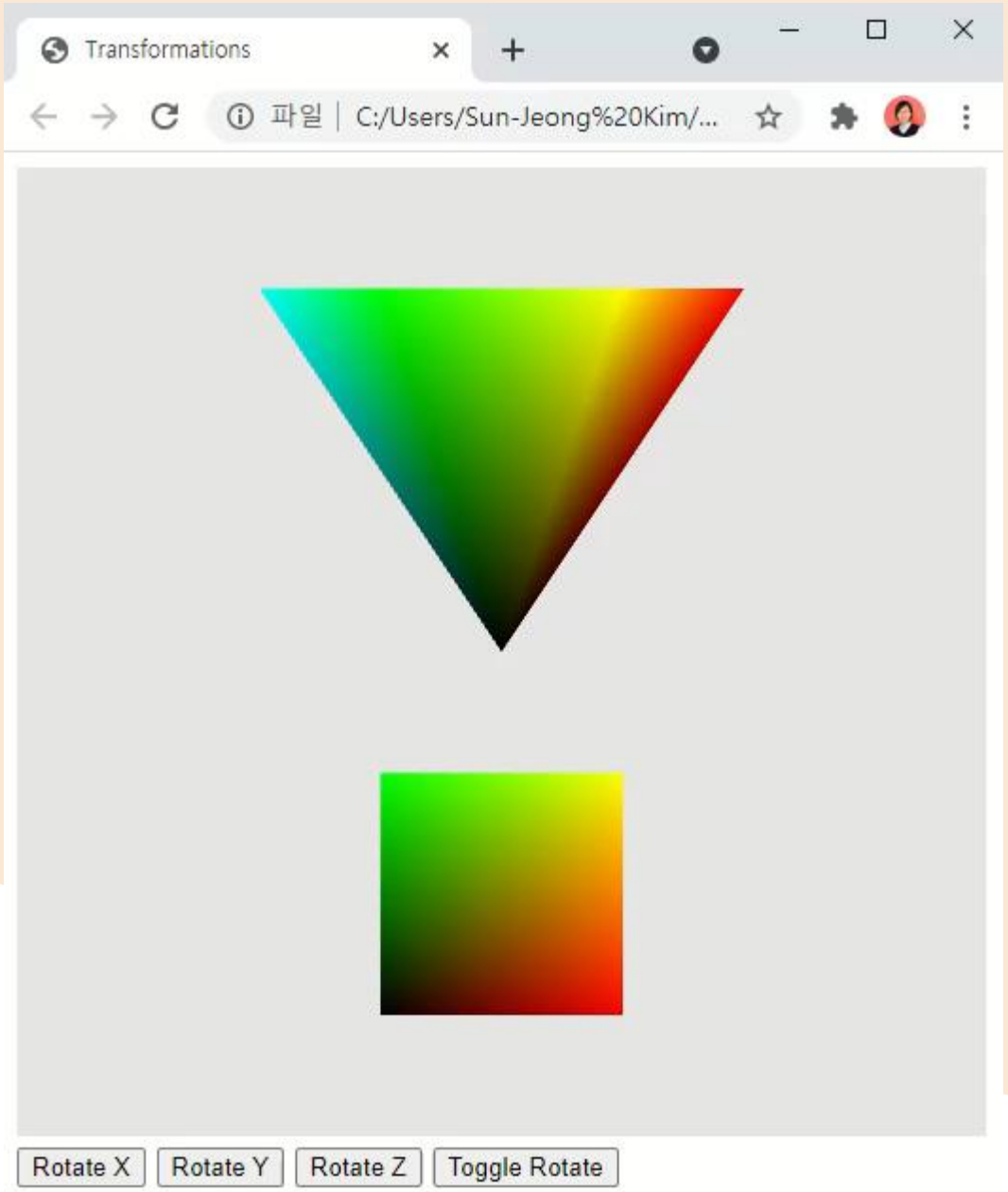
⊗ 0 ⚠ 0 Ln 46, Col 64 (8 selected) Spaces: 4 UTF-8 CRLF HTML

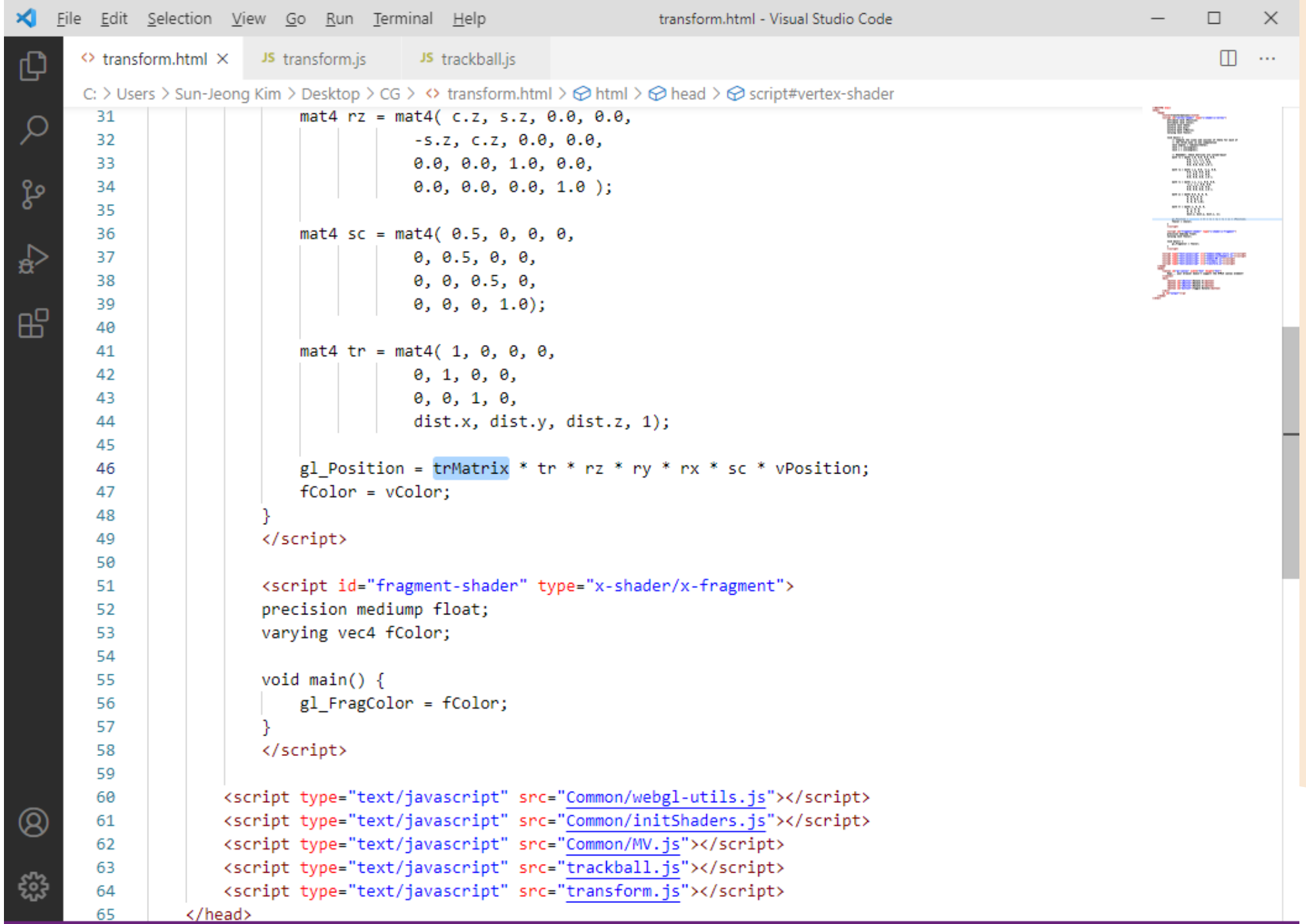

File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.js > init
var gl;
var points = [];
var colors = [];
var axis = 0;
var theta = [0, 0, 0];
var thetaLoc;
var dist = [0, 0, 0];
var distLoc;
var trMatrixLoc;
var rotation = false;
window.onload = function init()
{
var canvas = document.getElementById("gl-canvas");
gl = WebGLUtils.setupWebGL(canvas);
if(!gl) {
alert("WebGL isn't available!");
}
generateColorCube();
generateHexaPyramid();
// 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;
});
Ln 84, Col 31
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.js > init
25
26 // virtual trackball
27 var trball = trackball(canvas.width, canvas.height);
28 var mouseDown = false;
29
30 canvas.addEventListener("mousedown", function(event) {
31 trball.start(event.clientX, event.clientY);
32
33 mouseDown = true;
34 });
35
36 canvas.addEventListener("mouseup", function(event) {
37 mouseDown = false;
38 });
39
40 canvas.addEventListener("mousemove", function(event) {
41 if (mouseDown) {
42 trball.end(event.clientX, event.clientY);
43
44 gl.uniformMatrix4fv(trMatrixLoc, false, trball.rotationMatrix);
45 }
46 });
47
48 // Configure WebGL
49 gl.viewport(0, 0, canvas.width, canvas.height);
50 gl.clearColor(0.9, 0.9, 0.9, 1.0);
51
52 // Enable hidden-surface removal
53 gl.enable(gl.DEPTH_TEST);
54
55 // Load shaders and initialize attribute buffers
56 var program = initShaders(gl, "vertex-shader", "fragment-shader");
57 gl.useProgram(program);
58
59 // Load the data into the GPU
0 0
Ln 84, Col 31 Spaces: 4 UTF-8 CRLF JavaScript


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.js > init
78
79 thetaLoc = gl.getUniformLocation(program, "theta");
80 //gl.uniform3fv(thetaLoc, theta);
81 distLoc = gl.getUniformLocation(program, "dist");
82 //gl.uniform3fv(distLoc, dist);
83 trMatrixLoc = gl.getUniformLocation(program, "trMatrix");
84 gl.uniformMatrix4fv(trMatrixLoc, false, trball.rotationMatrix);
85
86 // Event listeners for buttons
87 document.getElementById("xButton").onclick = function () {
88 axis = 0;
89 };
90 document.getElementById("yButton").onclick = function () {
91 axis = 1;
92 };
93 document.getElementById("zButton").onclick = function () {
94 axis = 2;
95 };
96 document.getElementById("buttonT").onclick = function () {
97 rotation = !rotation;
98 };
99
100 render();
101 };
102
103 function render() {
104 gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
105
106 if(rotation) {
107 theta[axis] += 2.0;
108 }
109 gl.uniform3fv(thetaLoc, theta)
110
111 //gl.drawArrays(gl.TRIANGLES, 0, points.length);
112
0 0
Ln 84, Col 31
Spaces: 4
UTF-8
CRLF
JavaScript


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


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


File Edit Selection View Go Run Terminal Help
transform.html - Visual Studio Code
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > transform.html > html > head > script#vertex-shader
31 mat4 rz = mat4(c.z, s.z, 0.0, 0.0,
32 -s.z, c.z, 0.0, 0.0,
33 0.0, 0.0, 1.0, 0.0,
34 0.0, 0.0, 0.0, 1.0);
35
36 mat4 sc = mat4(0.5, 0, 0, 0,
37 0, 0.5, 0, 0,
38 0, 0, 0.5, 0,
39 0, 0, 0, 1.0);
40
41 mat4 tr = mat4(1, 0, 0, 0,
42 0, 1, 0, 0,
43 0, 0, 1, 0,
44 dist.x, dist.y, dist.z, 1);
45
46 gl_Position = trMatrix * tr * rz * ry * rx * sc * vPosition;
47 fColor = vColor;
48 }
49 </script>
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="transform.js"></script>
65 </head>
Ln 46, Col 39 (8 selected) Spaces: 4 UTF-8 CRLF HTML

연습 문제 (6)

- 4개 오브젝트를 그리시오.
  - 상하 오브젝트들은 서로 반대 방향으로 회전
  - 좌우 오브젝트들도 서로 반대 방향으로 회전