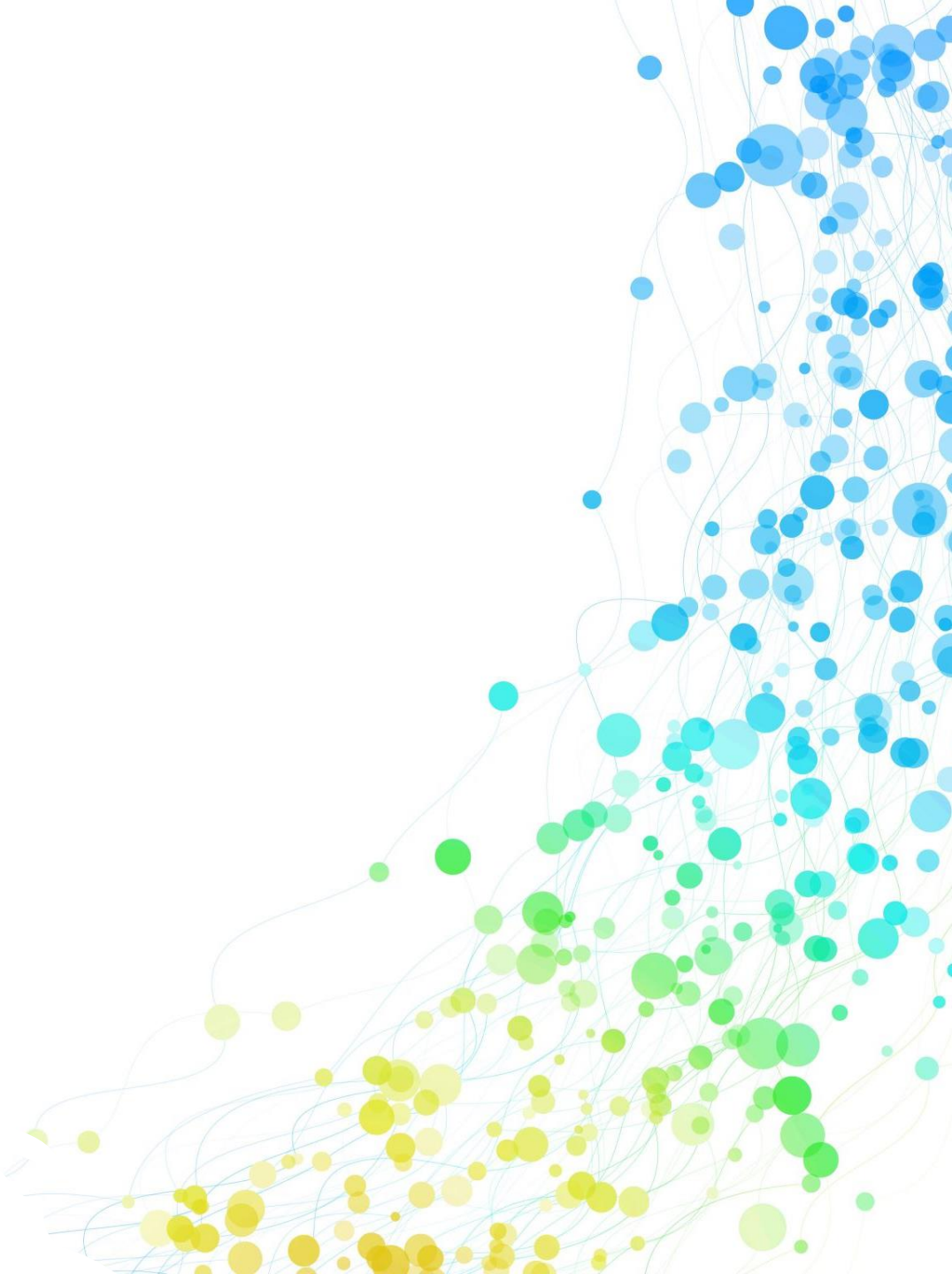
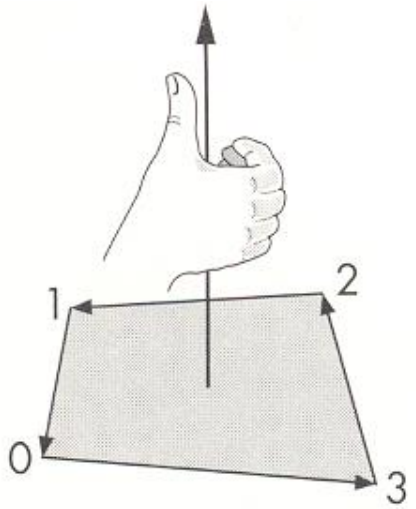
Geometric Objects and Transformations

6TH WEEK, 2022



Modeling a Cube

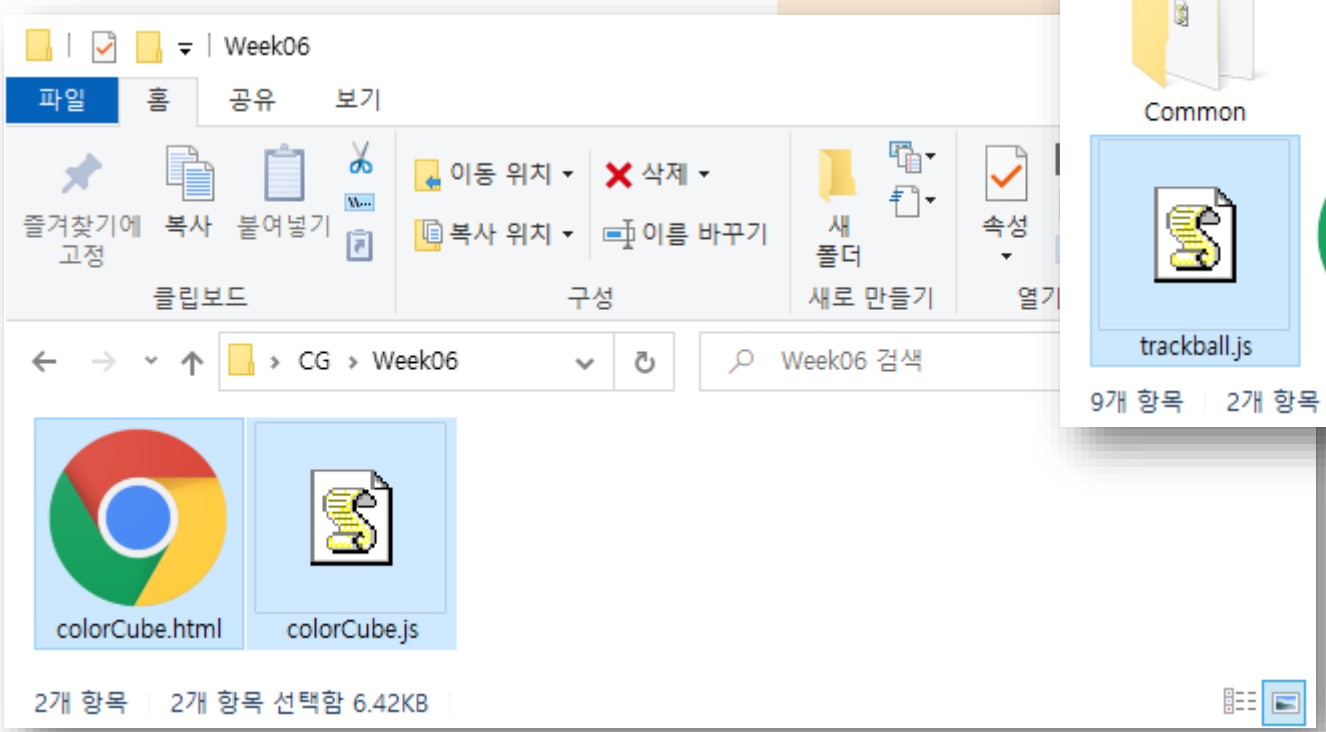
- Surface-based model
  - Outward-pointing face
    - Right-hand rule: counterclockwise order
- Data structure
  - Geometry: location of vertices

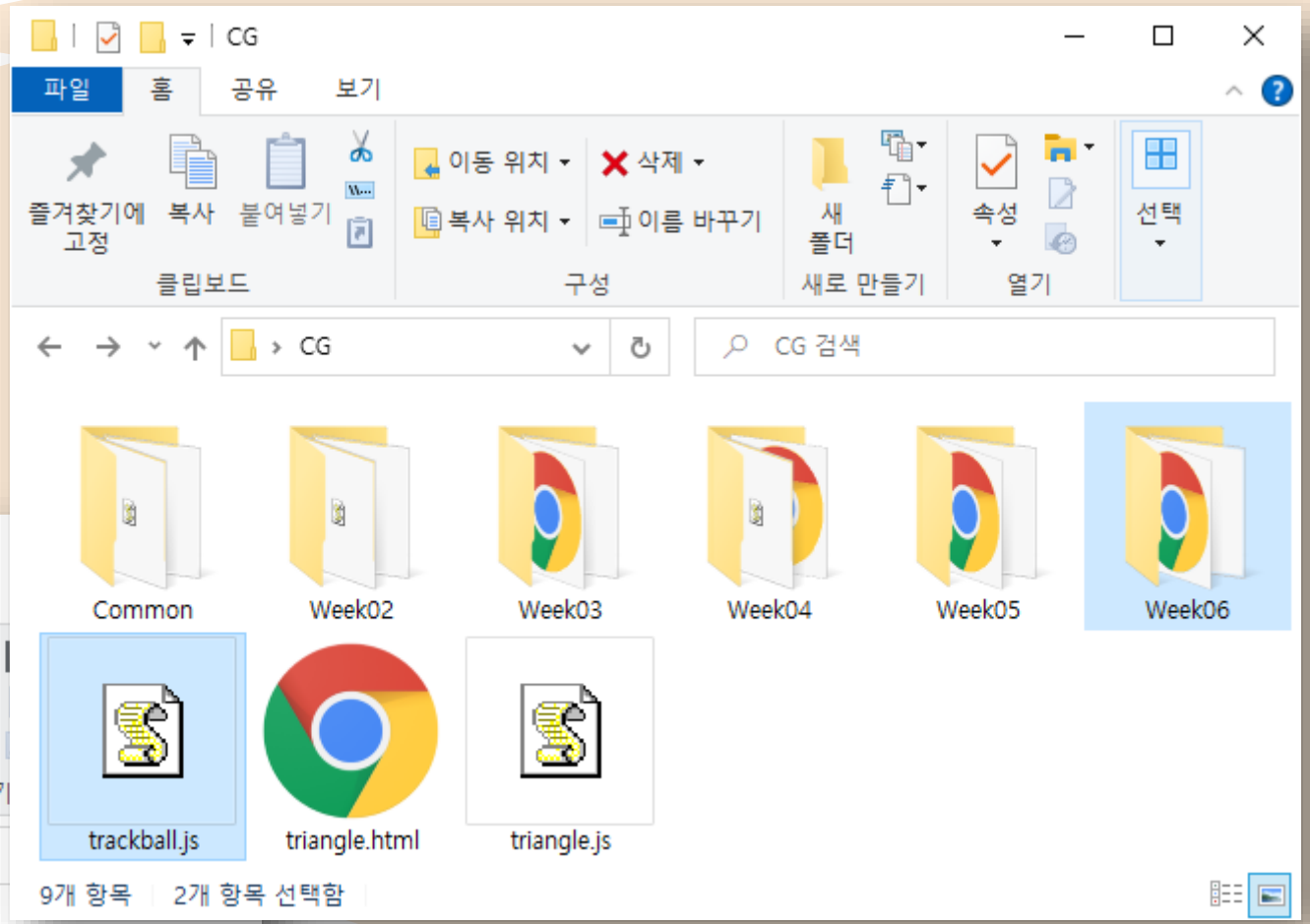



```
float vertices[8][3]= { {-1.0, -1.0, -1.0}, {1.0, -1.0, -1.0},
        {1.0, 1.0, -1.0}, {-1.0, 1.0, -1.0}, {-1.0, -1.0, 1.0},
        {1.0, -1.0, 1.0}, {1.0, 1.0, 1.0}, {-1.0, 1.0, 1.0}};
```

  - 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 );
```


Week06
파일
홈
공유
보기
즐겨찾기에 고정
복사
붙여넣기
이동 위치
복사 위치
삭제
이름 바꾸기
새 폴더
속성
클립보드
구성
새로 만들기
열기
CG › Week06
Week06 검색
colorCube.html
colorCube.js
2개 항목
2개 항목 선택함 6.42KB


CG
파일
홈
공유
보기
즐겨찾기에 고정
복사
붙여넣기
이동 위치
복사 위치
삭제
이름 바꾸기
새 폴더
속성
선택
클립보드
구성
새로 만들기
열기
CG
CG 검색
Common
Week02
Week03
Week04
Week05
Week06
trackball.js
triangle.html
triangle.js
9개 항목
2개 항목 선택함

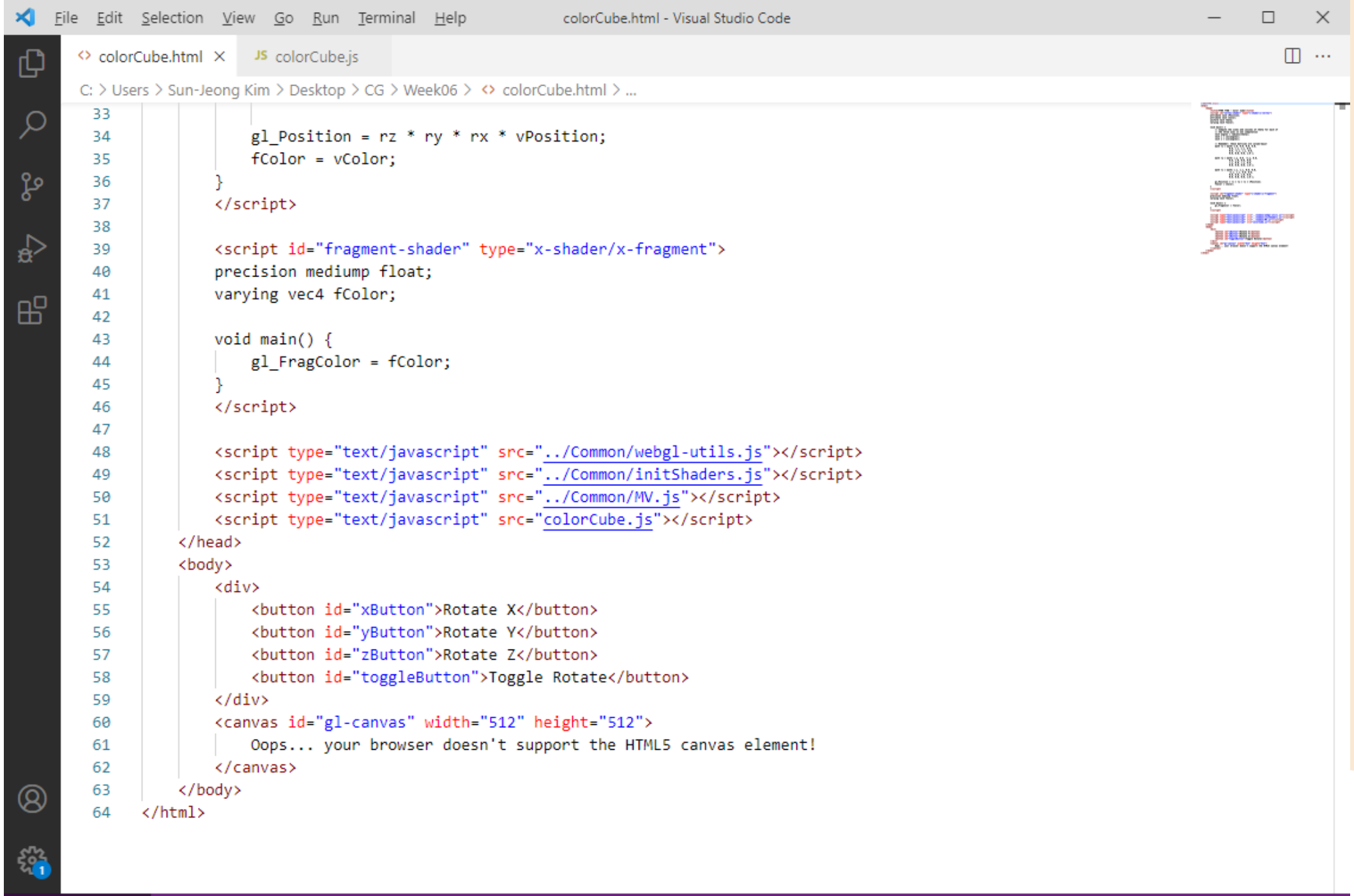
File Edit Selection View Go Run Terminal Help colorCube.html - Visual Studio Code

colorCube.html × colorCube.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > colorCube.html > ...

```
<!DOCTYPE html>
<html>
    <head>
        <title>학번 이름 - Color Cube</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;
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 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 > Week06 > colorCube.html > ...
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="colorCube.js"></script>
52 </head>
53 <body>
54
55 <button id="xButton">Rotate X</button>
56 <button id="yButton">Rotate Y</button>
57 <button id="zButton">Rotate Z</button>
58 <button id="toggleButton">Toggle Rotate</button>
59
60 <canvas id="gl-canvas" width="512" height="512">
61 Oops... your browser doesn't support the HTML5 canvas element!
62 </canvas>
63 </body>
64 </html>
Restricted Mode
0 0
Ln 1, Col 1
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 > Week06 > colorCube.js > ...

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

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

var rotation = false;

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

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

    // axes
    points.push(vec4(0.0, 0.0, 0.0, 1.0));
    points.push(vec4(1.0, 0.0, 0.0, 1.0));  // x-axis
    colors.push(vec4(1.0, 0.0, 0.0, 1.0));
    colors.push(vec4(1.0, 0.0, 0.0, 1.0));

    points.push(vec4(0.0, 0.0, 0.0, 1.0));
    points.push(vec4(0.0, 1.0, 0.0, 1.0));  // y-axis
    colors.push(vec4(0.0, 1.0, 0.0, 1.0));
    colors.push(vec4(0.0, 1.0, 0.0, 1.0));

    points.push(vec4(0.0, 0.0, 0.0, 1.0));
    points.push(vec4(0.0, 0.0, 1.0, 1.0));  // z-axis
    colors.push(vec4(0.0, 0.0, 1.0, 1.0));
    colors.push(vec4(0.0, 0.0, 1.0, 1.0));

```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 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 > Week06 > colorCube.js > ...

```
36      generateColorCube();
37
38      // Configure WebGL
39      gl.viewport(0, 0, canvas.width, canvas.height);
40      gl.clearColor(1.0, 1.0, 1.0, 1.0);
41
42      // Load shaders and initialize attribute buffers
43      var program = initShaders(gl, "vertex-shader", "fragment-shader");
44      gl.useProgram(program);
45
46      // Load the data into the GPU
47      var bufferId = gl.createBuffer();
48      gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
49      gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
50
51      // Associate our shader variables with our data buffer
52      var vPosition = gl.getAttribLocation(program, "vPosition");
53      gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
54      gl.enableVertexAttribArray(vPosition);
55
56      // Create a buffer object, initialize it, and associate it with
57      // the associated attribute variable in our vertex shader
58      var cBufferId = gl.createBuffer();
59      gl.bindBuffer(gl.ARRAY_BUFFER, cBufferId);
60      gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
61
62      var vColor = gl.getAttribLocation(program, "vColor");
63      gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
64      gl.enableVertexAttribArray(vColor);
65
66      thetaLoc = gl.getUniformLocation(program, "theta");
67      //gl.uniform3fv(thetaLoc, theta);
68
69      // Event listeners for buttons
70      document.getElementById("xButton").onclick = function () {
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 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 > Week06 > colorCube.js > ...

```
71          axis = 0;
72      };
73      document.getElementById("yButton").onclick = function () {
74          axis = 1;
75      };
76      document.getElementById("zButton").onclick = function () {
77          axis = 2;
78      };
79      document.getElementById("toggleButton").onclick = function () {
80          rotation = !rotation;
81      };
82
83      render();
84  };
85
86  function render() {
87      gl.clear(gl.COLOR_BUFFER_BIT);
88
89      if( rotation ) {
90          theta[axis] += 2.0;
91      }
92      gl.uniform3fv(thetaLoc, theta)
93
94      gl.drawArrays(gl.LINES, 0, 6);
95      gl.drawArrays(gl.TRIANGLES, 6, points.length-6);
96
97      window.requestAnimationFrame(render);
98  }
99
100 function generateColorCube() {
101     quad(1, 0, 3, 2);
102     //quad(2, 3, 7, 6);
103     //quad(3, 0, 4, 7);
104     //quad(4, 5, 6, 7);
105     //quad(5, 4, 0, 1);
```

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

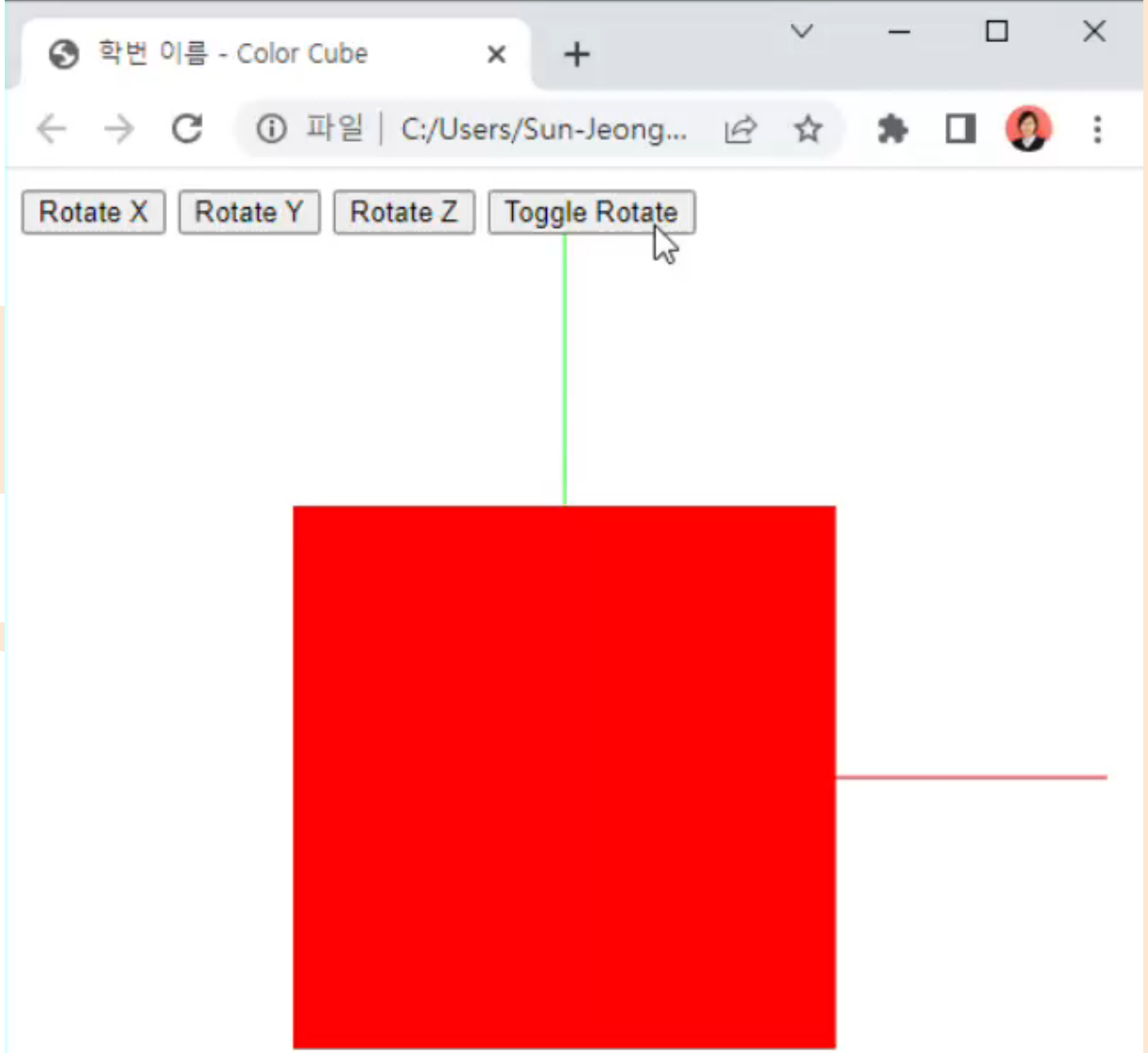
File Edit Selection View Go Run Terminal Help colorCube.js - Visual Studio Code

colorCube.html | colorCube.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > colorCube.js > ...

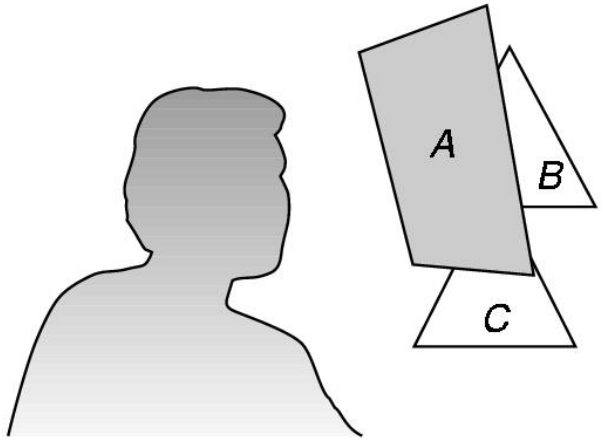
```
106     //quad(6, 5, 1, 2);
107 }
108
109 function quad(a, b, c, d) {
110     vertexPos = [
111         vec4(-0.5, -0.5, -0.5, 1.0),
112         vec4( 0.5, -0.5, -0.5, 1.0),
113         vec4( 0.5,  0.5, -0.5, 1.0),
114         vec4(-0.5,  0.5, -0.5, 1.0),
115         vec4(-0.5, -0.5,  0.5, 1.0),
116         vec4( 0.5, -0.5,  0.5, 1.0),
117         vec4( 0.5,  0.5,  0.5, 1.0),
118         vec4(-0.5,  0.5,  0.5, 1.0)
119     ];
120
121     vertexColor = [
122         vec4(0.0, 0.0, 0.0, 1.0),   // black
123         vec4(1.0, 0.0, 0.0, 1.0),   // red
124         vec4(1.0, 1.0, 0.0, 1.0),   // yellow
125         vec4(0.0, 1.0, 0.0, 1.0),   // green
126         vec4(0.0, 0.0, 1.0, 1.0),   // blue
127         vec4(1.0, 0.0, 1.0, 1.0),   // magenta
128         vec4(1.0, 1.0, 1.0, 1.0),   // white
129         vec4(0.0, 1.0, 1.0, 1.0)    // cyan
130     ];
131
132     // We need to partition the quad into two triangles in order for WebGL
133     // to be able to render it. In this case, we create two triangles from
134     // the quad indices.
135     var index = [ a, b, c, a, c, d ];
136     for(var i=0; i<index.length; i++) {
137         points.push(vertexPos[index[i]]);
138         colors.push(vertexColor[a]);
139     }
140 }
```

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


학번 이름 - Color Cube
파일 | C:/Users/Sun-Jeong...
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 );`

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

colorCube.html | colorCube.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > colorCube.js > init

```
36    generateColorCube();
37
38    // Configure WebGL
39    gl.viewport(0, 0, canvas.width, canvas.height);
40    gl.clearColor(1.0, 1.0, 1.0, 1.0);
41
42    // Enable hidden-surface removal
43    gl.enable(gl.DEPTH_TEST);
44
45    // Load shaders and initialize attribute buffers
46    var program = initShaders(gl, "vertex-shader", "fragment-shader");
47    gl.useProgram(program);
48
49    // Load the data into the GPU
50    var bufferId = gl.createBuffer();
51    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
52    gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
53
54    // Associate our shader variables with our data buffer
55    var vPosition = gl.getAttribLocation(program, "vPosition");
56    gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
57    gl.enableVertexAttribArray(vPosition);
58
59    // Create a buffer object, initialize it, and associate it with
60    // the associated attribute variable in our vertex shader
61    var cBufferId = gl.createBuffer();
62    gl.bindBuffer(gl.ARRAY_BUFFER, cBufferId);
63    gl.bufferData(gl.ARRAY_BUFFER, flatten(colors), gl.STATIC_DRAW);
64
65    var vColor = gl.getAttribLocation(program, "vColor");
66    gl.vertexAttribPointer(vColor, 4, gl.FLOAT, false, 0, 0);
67    gl.enableVertexAttribArray(vColor);
68
69    thetaLoc = gl.getUniformLocation(program, "theta");
70    //gl.uniform3fv(thetaLoc, theta);
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 43, Col 30 Spaces: 4 UTF-8 CRLF JavaScript

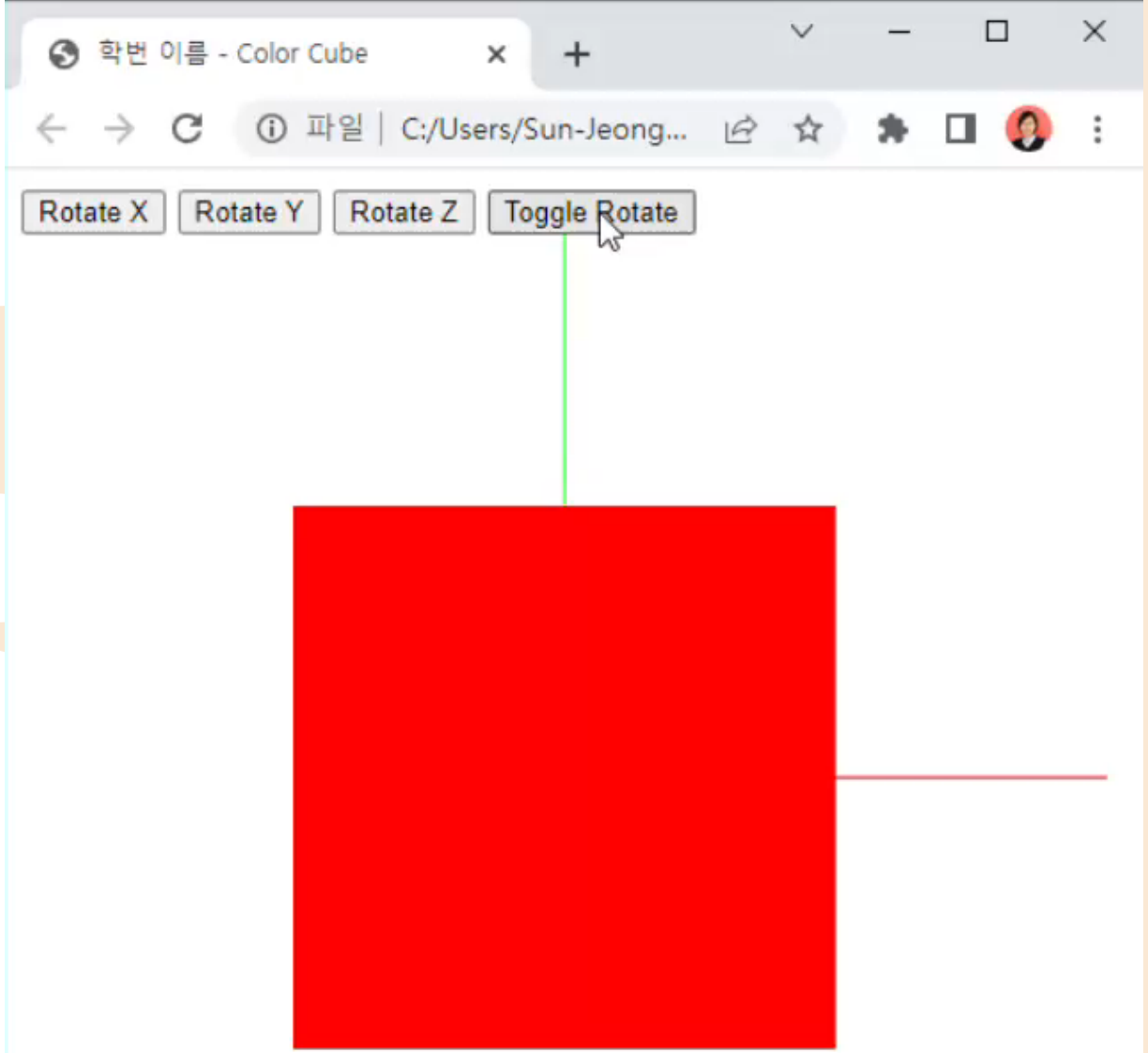
File Edit Selection View Go Run Terminal Help colorCube.js - Visual Studio Code

colorCube.html | colorCube.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > colorCube.js > render

```
71
72      // Event listeners for buttons
73      document.getElementById("xButton").onclick = function () {
74          axis = 0;
75      };
76      document.getElementById("yButton").onclick = function () {
77          axis = 1;
78      };
79      document.getElementById("zButton").onclick = function () {
80          axis = 2;
81      };
82      document.getElementById("toggleButton").onclick = function () {
83          rotation = !rotation;
84      };
85
86      render();
87  };
88
89  function render() {
90      gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
91
92      if( rotation ) {
93          theta[axis] += 2.0;
94      }
95      gl.uniform3fv(thetaLoc, theta)
96
97      gl.drawArrays(gl.LINES, 0, 6);
98      gl.drawArrays(gl.TRIANGLES, 6, points.length-6);
99
100     window.requestAnimationFrame(render);
101 }
102
103 function generateColorCube() {
104     quad(1, 0, 3, 2);
105     //quad(2, 3, 7, 6);
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 90, Col 57 Spaces: 4 UTF-8 CRLF JavaScript


학번 이름 - Color Cube
파일 | C:/Users/Sun-Jeong...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

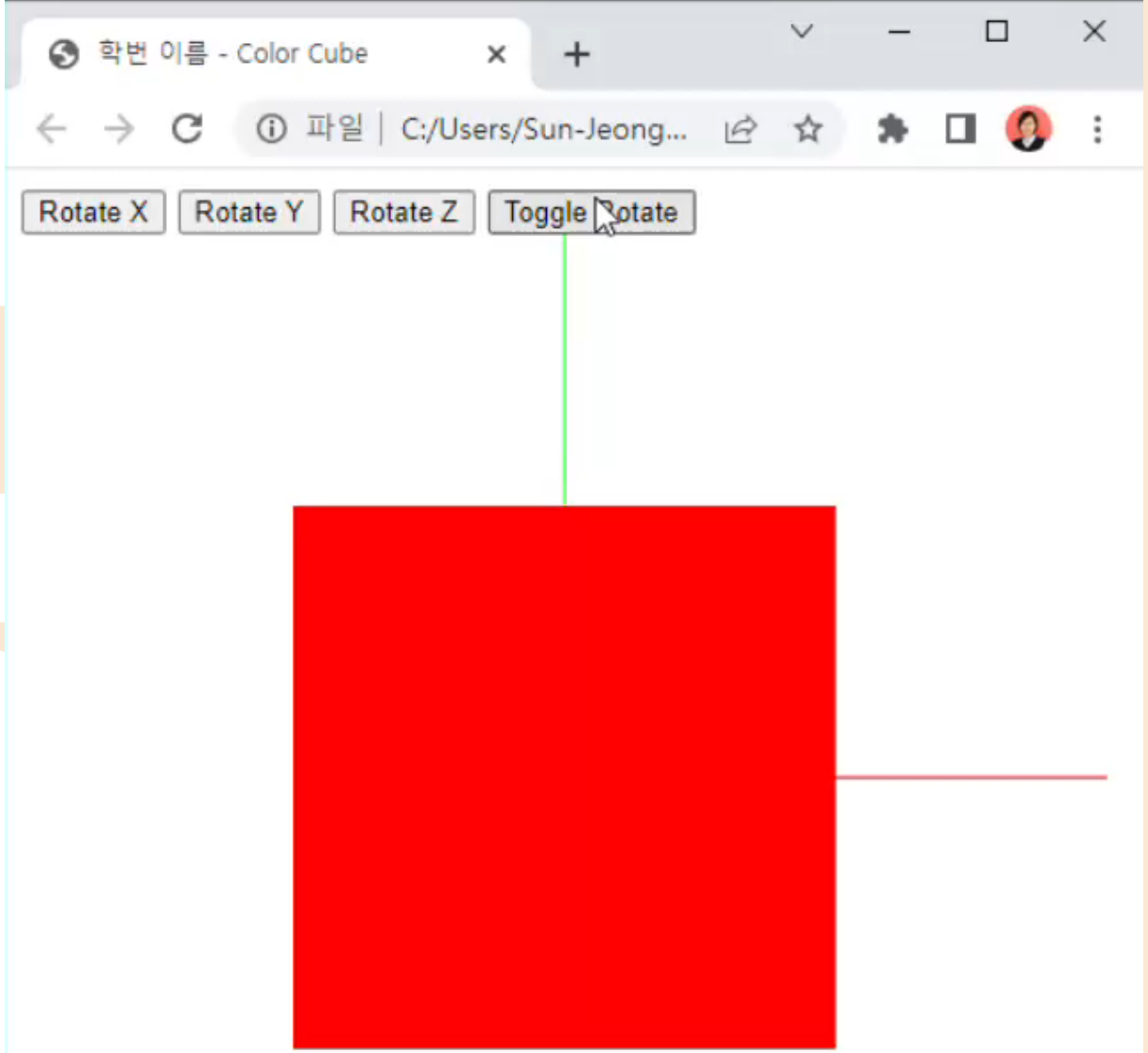
File Edit Selection View Go Run Terminal Help colorCube.js - Visual Studio Code

colorCube.html | colorCube.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > colorCube.js > generateColorCube

```
102
103 function generateColorCube() {
104     quad(1, 0, 3, 2);
105     quad(2, 3, 7, 6);
106     quad(3, 0, 4, 7);
107     //quad(4, 5, 6, 7);
108     quad(5, 4, 0, 1);
109     quad(6, 5, 1, 2);
110 }
111
112 function quad(a, b, c, d) {
113     vertexPos = [
114         vec4(-0.5, -0.5, -0.5, 1.0),
115         vec4( 0.5, -0.5, -0.5, 1.0),
116         vec4( 0.5,  0.5, -0.5, 1.0),
117         vec4(-0.5,  0.5, -0.5, 1.0),
118         vec4(-0.5, -0.5,  0.5, 1.0),
119         vec4( 0.5, -0.5,  0.5, 1.0),
120         vec4( 0.5,  0.5,  0.5, 1.0),
121         vec4(-0.5,  0.5,  0.5, 1.0)
122     ];
123
124     vertexColor = [
125         vec4(0.0, 0.0, 0.0, 1.0),   // black
126         vec4(1.0, 0.0, 0.0, 1.0),   // red
127         vec4(1.0, 1.0, 0.0, 1.0),   // yellow
128         vec4(0.0, 1.0, 0.0, 1.0),   // green
129         vec4(0.0, 0.0, 1.0, 1.0),   // blue
130         vec4(1.0, 0.0, 1.0, 1.0),   // magenta
131         vec4(1.0, 1.0, 1.0, 1.0),   // white
132         vec4(0.0, 1.0, 1.0, 1.0)    // cyan
133     ];
134
135     // We need to partition the quad into two triangles in order for WebGL
136     // to be able to render it. In this case, we create two triangles from
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 109, Col 5 Spaces: 4 UTF-8 CRLF JavaScript


학번 이름 - Color Cube
파일 | C:/Users/Sun-Jeong...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

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

colorCube.html | colorCube.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > colorCube.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      // axes
21      points.push(vec4(0.0, 0.0, 0.0, 1.0));
22      points.push(vec4(1.0, 0.0, 0.0, 1.0));  // x-axis
23      colors.push(vec4(1.0, 0.0, 0.0, 1.0));
24      colors.push(vec4(1.0, 0.0, 0.0, 1.0));
25
26      points.push(vec4(0.0, 0.0, 0.0, 1.0));
27      points.push(vec4(0.0, 1.0, 0.0, 1.0));  // y-axis
28      colors.push(vec4(0.0, 1.0, 0.0, 1.0));
29      colors.push(vec4(0.0, 1.0, 0.0, 1.0));
30
31      points.push(vec4(0.0, 0.0, 0.0, 1.0));
32      points.push(vec4(0.0, 0.0, 1.0, 1.0));  // z-axis
33      colors.push(vec4(0.0, 0.0, 1.0, 1.0));
34      colors.push(vec4(0.0, 0.0, 1.0, 1.0));
35  */
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 35, Col 3 Spaces: 4 UTF-8 CRLF JavaScript

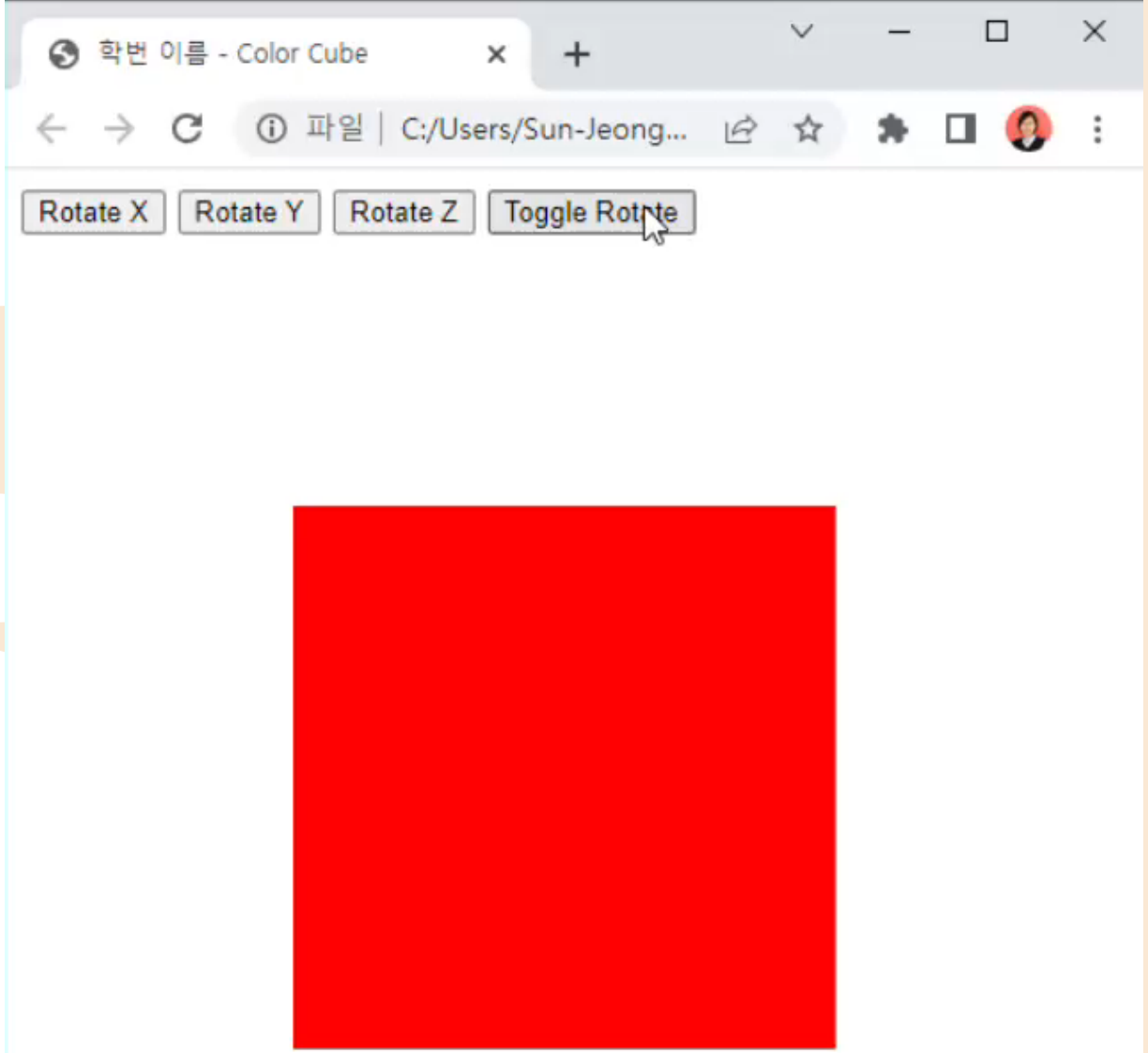
File Edit Selection View Go Run Terminal Help colorCube.js - Visual Studio Code

colorCube.html | colorCube.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > colorCube.js > generateColorCube

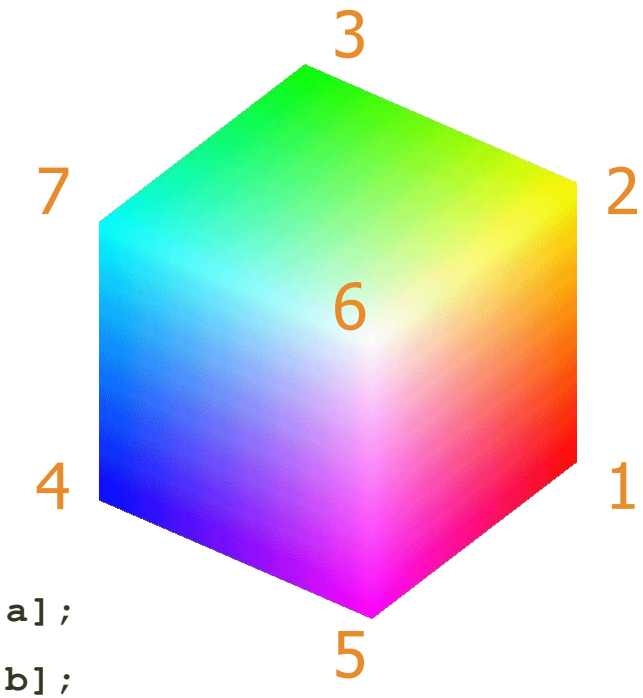
```
76      document.getElementById("yButton").onclick = function () {
77          axis = 1;
78      };
79      document.getElementById("zButton").onclick = function () {
80          axis = 2;
81      };
82      document.getElementById("toggleButton").onclick = function () {
83          rotation = !rotation;
84      };
85
86      render();
87  };
88
89  function render() {
90      gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
91
92      if( rotation ) {
93          theta[axis] += 2.0;
94      }
95      gl.uniform3fv(thetaLoc, theta)
96
97      //gl.drawArrays(gl.LINES, 0, 6);
98      gl.drawArrays(gl.TRIANGLES, 0, points.length);
99
100     window.requestAnimationFrame(render);
101 }
102
103 function generateColorCube() {
104     quad(1, 0, 3, 2);
105     quad(2, 3, 7, 6);
106     quad(3, 0, 4, 7);
107     quad(4, 5, 6, 7);
108     quad(5, 4, 0, 1);
109     quad(6, 5, 1, 2);
110 }
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 107, Col 5 Spaces: 4 UTF-8 CRLF JavaScript


학번 이름 - Color Cube
파일 | C:/Users/Sun-Jeong...
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++;
}
```

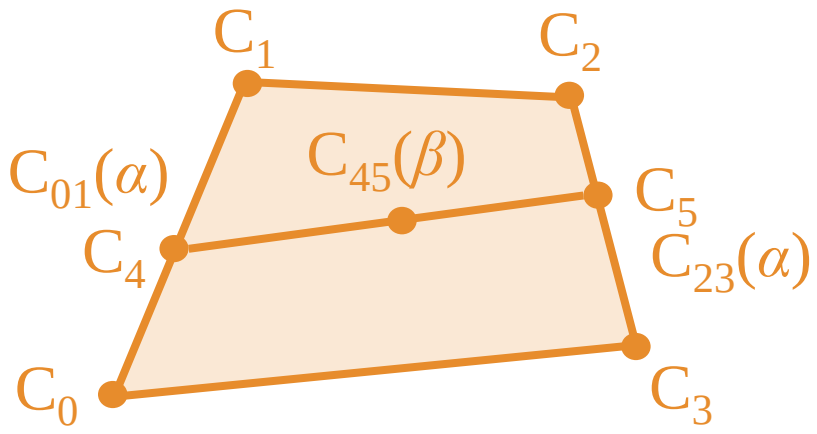


Color Cube (2)

```
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

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



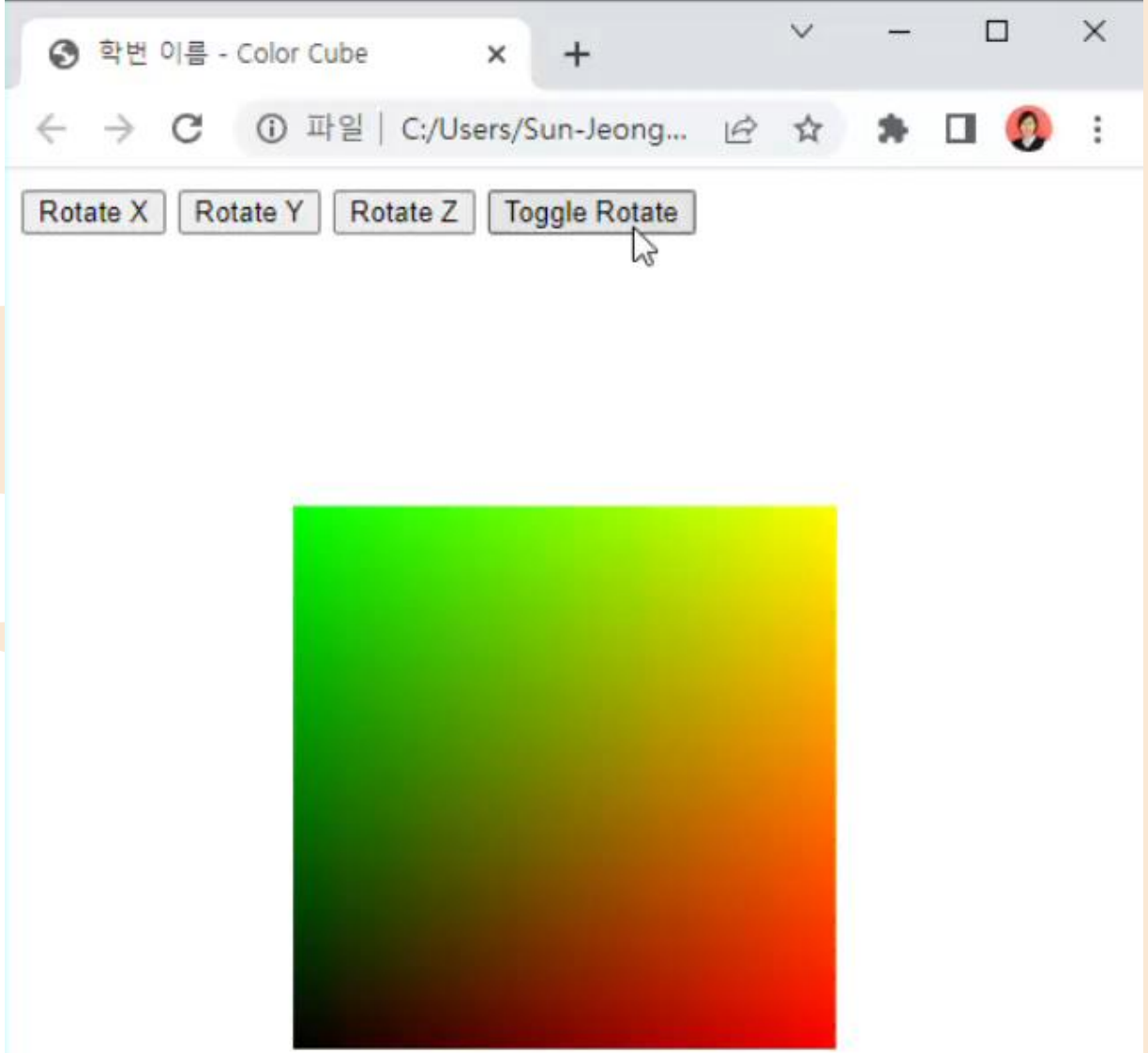
File Edit Selection View Go Run Terminal Help colorCube.js - Visual Studio Code

colorCube.html | colorCube.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > colorCube.js > quad

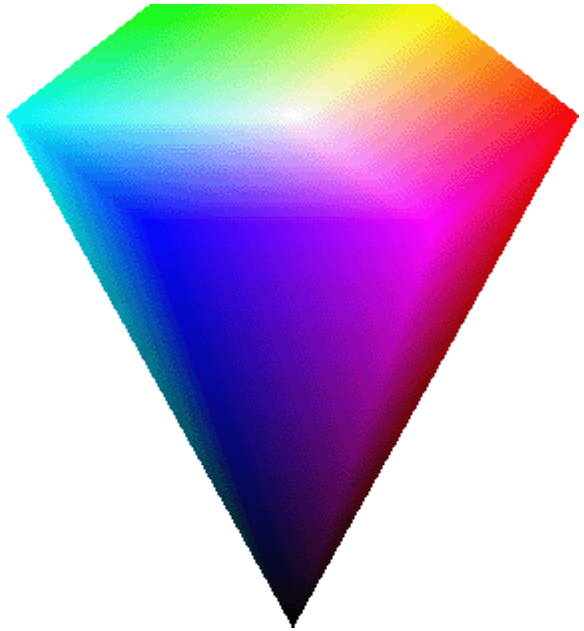
```
111
112 function quad(a, b, c, d) {
113     vertexPos = [
114         vec4(-0.5, -0.5, -0.5, 1.0),
115         vec4( 0.5, -0.5, -0.5, 1.0),
116         vec4( 0.5,  0.5, -0.5, 1.0),
117         vec4(-0.5,  0.5, -0.5, 1.0),
118         vec4(-0.5, -0.5,  0.5, 1.0),
119         vec4( 0.5, -0.5,  0.5, 1.0),
120         vec4( 0.5,  0.5,  0.5, 1.0),
121         vec4(-0.5,  0.5,  0.5, 1.0)
122     ];
123
124     vertexColor = [
125         vec4(0.0, 0.0, 0.0, 1.0),   // black
126         vec4(1.0, 0.0, 0.0, 1.0),   // red
127         vec4(1.0, 1.0, 0.0, 1.0),   // yellow
128         vec4(0.0, 1.0, 0.0, 1.0),   // green
129         vec4(0.0, 0.0, 1.0, 1.0),   // blue
130         vec4(1.0, 0.0, 1.0, 1.0),   // magenta
131         vec4(1.0, 1.0, 1.0, 1.0),   // white
132         vec4(0.0, 1.0, 1.0, 1.0)    // cyan
133     ];
134
135     // We need to partition the quad into two triangles in order for WebGL
136     // to be able to render it. In this case, we create two triangles from
137     // the quad indices.
138     var index = [ a, b, c, a, c, d ];
139     for(var i=0; i<index.length; i++) {
140         points.push(vertexPos[index[i]]);
141         //colors.push(vertexColor[a]);
142         colors.push(vertexColor[index[i]]);
143     }
144 }
145
```

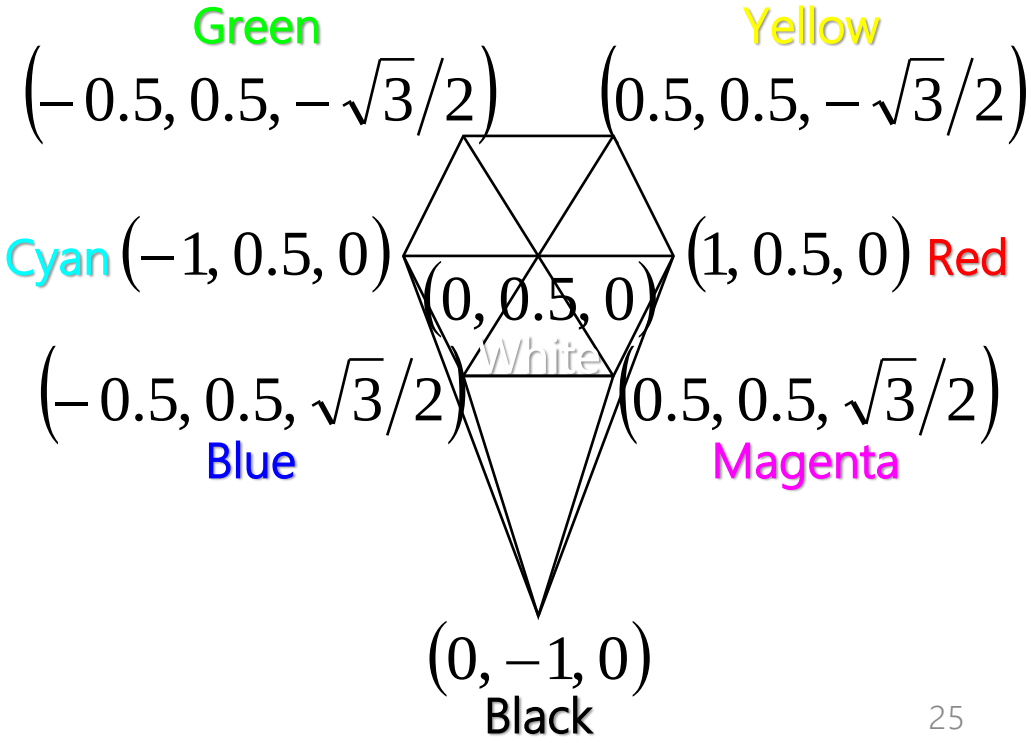
Restricted Mode ⊗ 0 ⚠ 0 Ln 142, Col 44 Spaces: 4 UTF-8 CRLF JavaScript

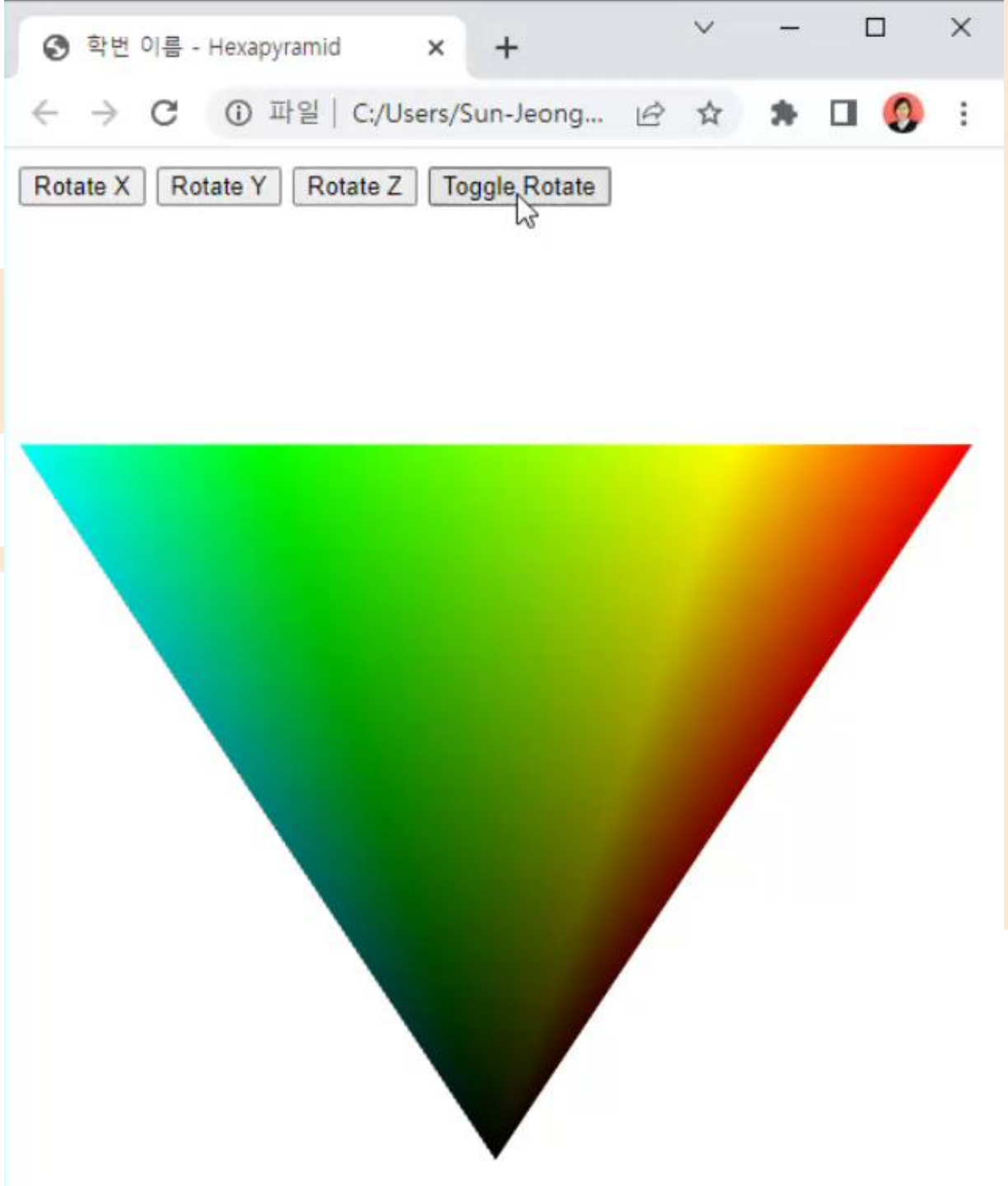

학번 이름 - Color Cube
파일 | C:/Users/Sun-Jeong...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

연습 문제 (1)

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



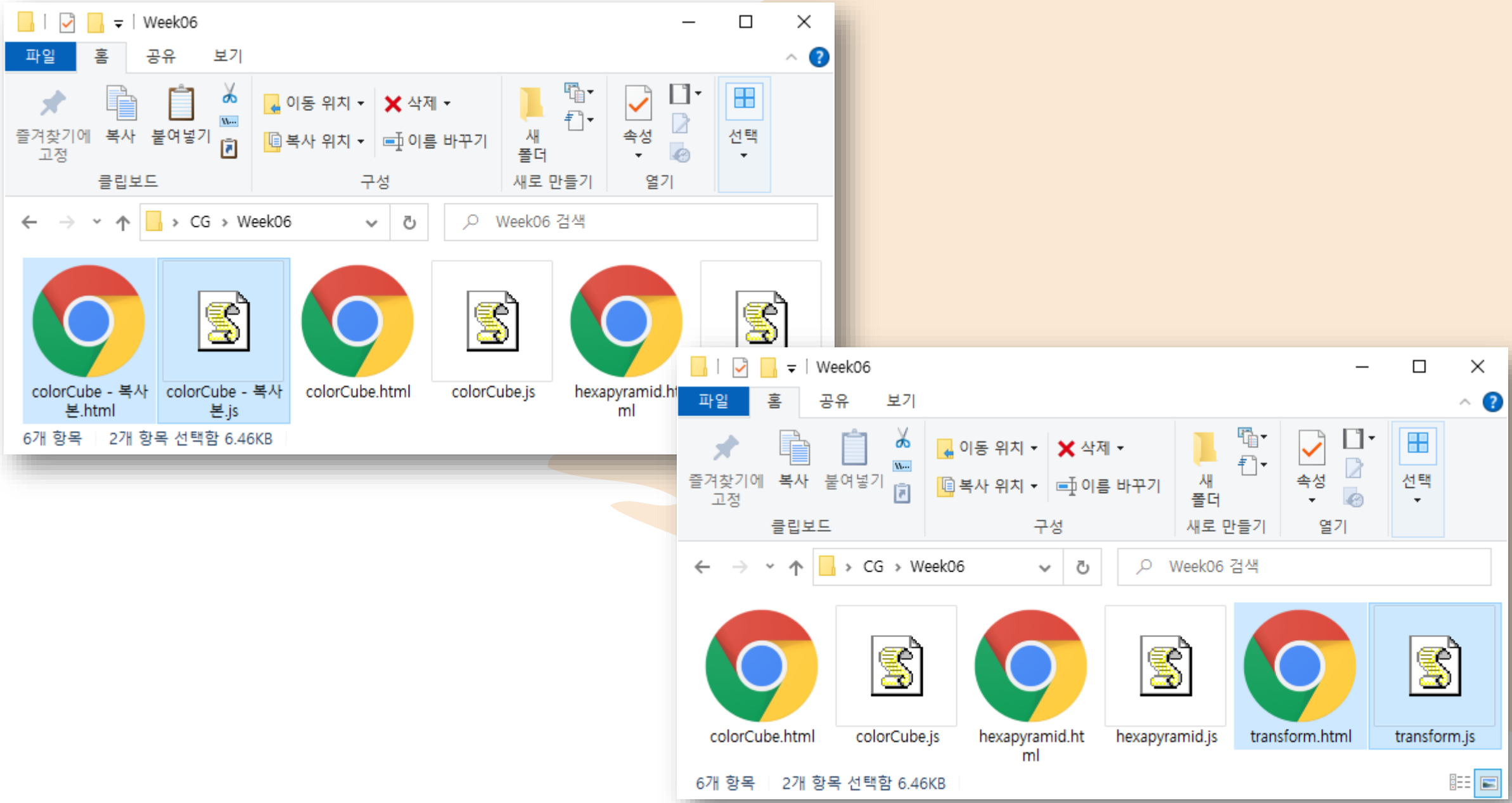


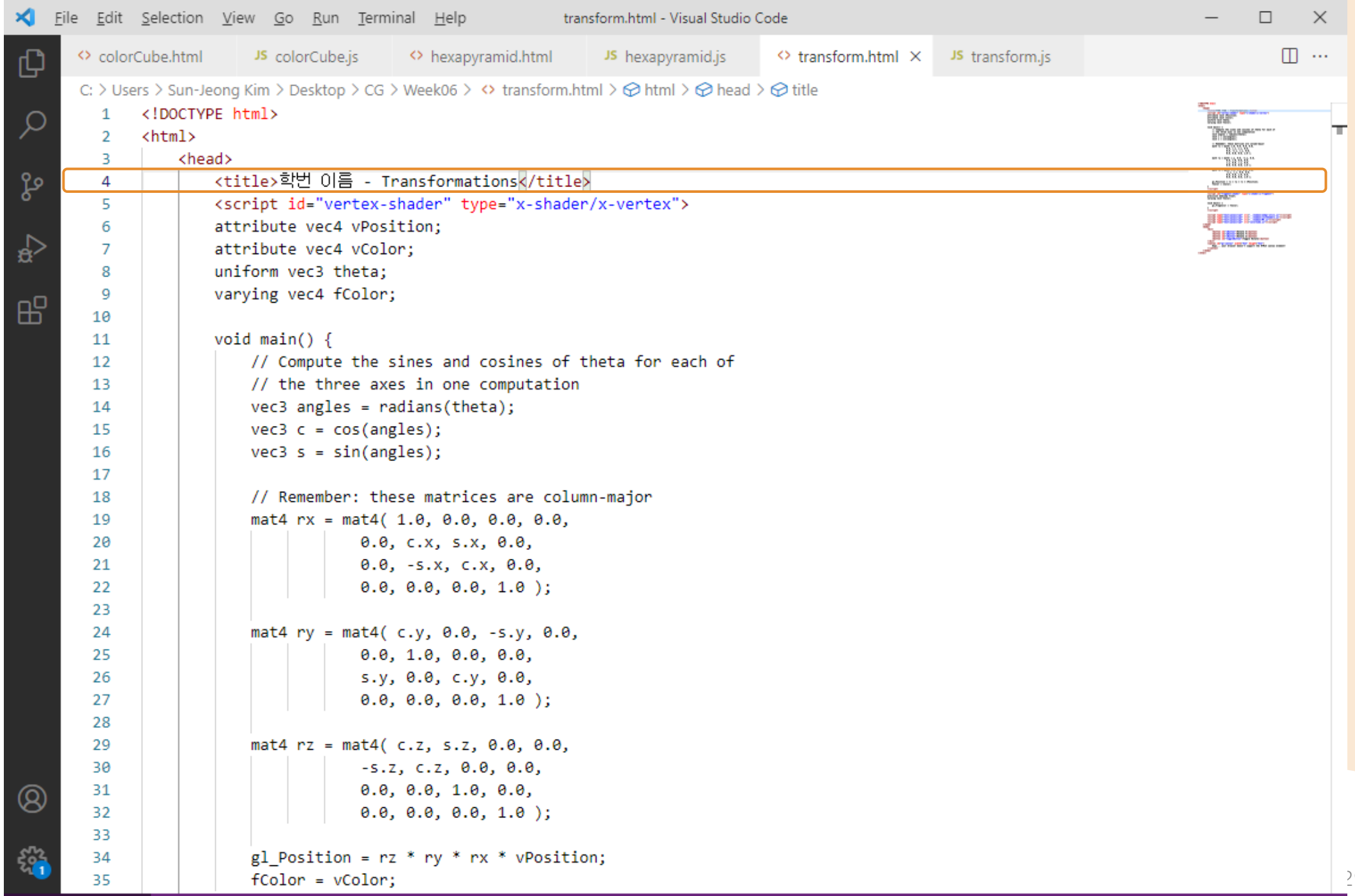

학번 이름 - Hexapyramid
파일 | C:/Users/Sun-Jeong...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

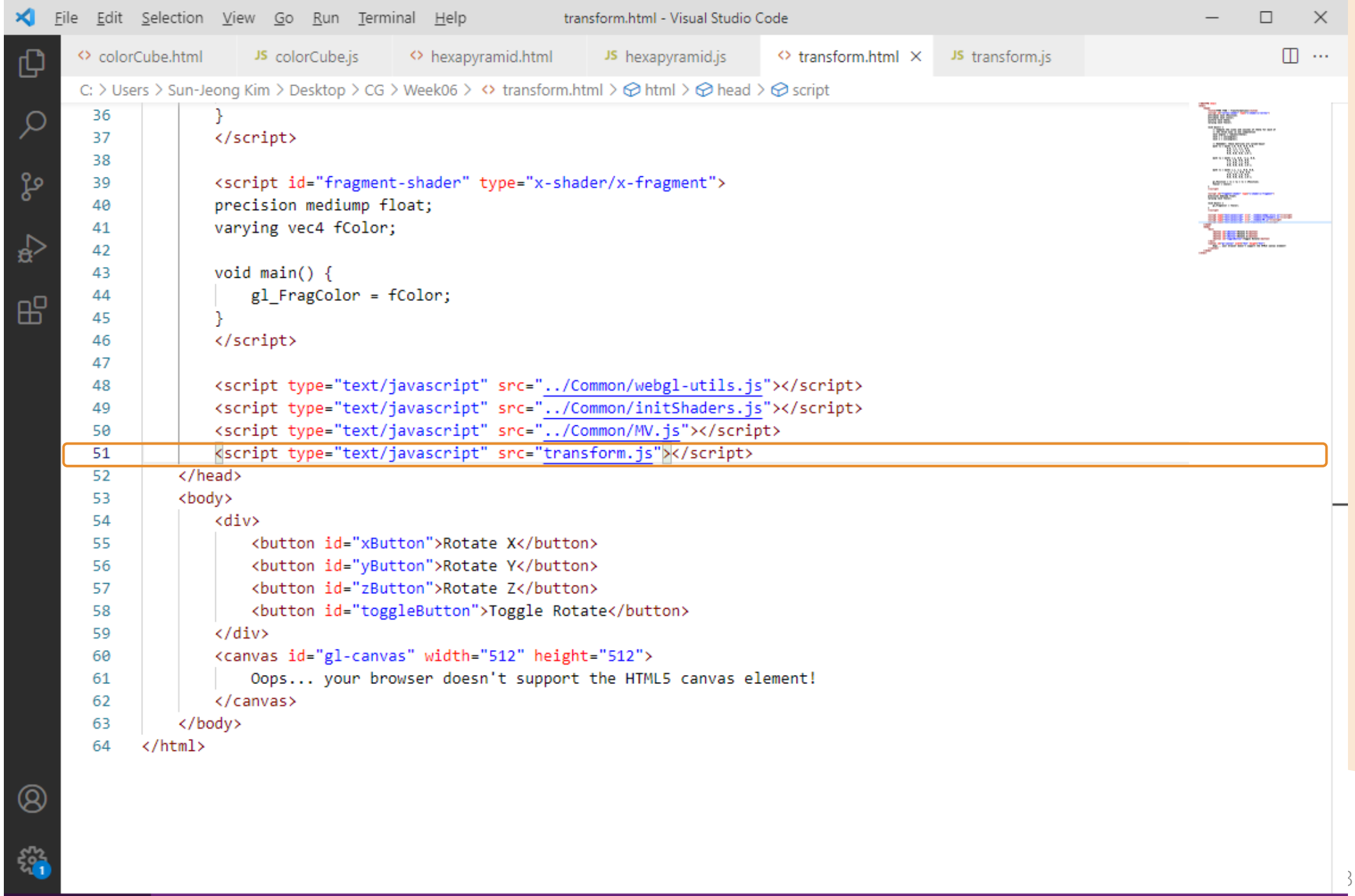
Line-Preserving Transformations

- 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


Week06
파일
홈
공유
보기
즐겨찾기에 고정
복사
붙여넣기
이동 위치
복사 위치
삭제
이름 바꾸기
새 폴더
속성
선택
클립보드
구성
새로 만들기
열기
CG › Week06
Week06 검색
colorCube - 복사본.html
colorCube - 복사본.js
colorCube.html
colorCube.js
hexapyramid.html
6개 항목
2개 항목 선택함 6.46KB
Week06
파일
홈
공유
보기
즐겨찾기에 고정
복사
붙여넣기
이동 위치
복사 위치
삭제
이름 바꾸기
새 폴더
속성
선택
클립보드
구성
새로 만들기
열기
CG › Week06
Week06 검색
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
6개 항목
2개 항목 선택함 6.46KB


File Edit Selection View Go Run Terminal Help
transform.html - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.html > html > head > title
<!DOCTYPE html>
<html>
<head>
<title>학번 이름 - 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;
Restricted Mode
0 0
Ln 4, Col 39
Spaces: 4
UTF-8
CRLF
HTML


File Edit Selection View Go Run Terminal Help
transform.html - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.html > html > head > script
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
55 <button id="xButton">Rotate X</button>
56 <button id="yButton">Rotate Y</button>
57 <button id="zButton">Rotate Z</button>
58 <button id="toggleButton">Toggle Rotate</button>
59
60 <canvas id="gl-canvas" width="512" height="512">
61 Oops... your browser doesn't support the HTML5 canvas element!
62 </canvas>
63 </body>
64 </html>
Restricted Mode
0 0
Ln 51, Col 54
Spaces: 4
UTF-8
CRLF
HTML

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

colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > 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();
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 25, Col 39 Spaces: 4 UTF-8 CRLF JavaScript

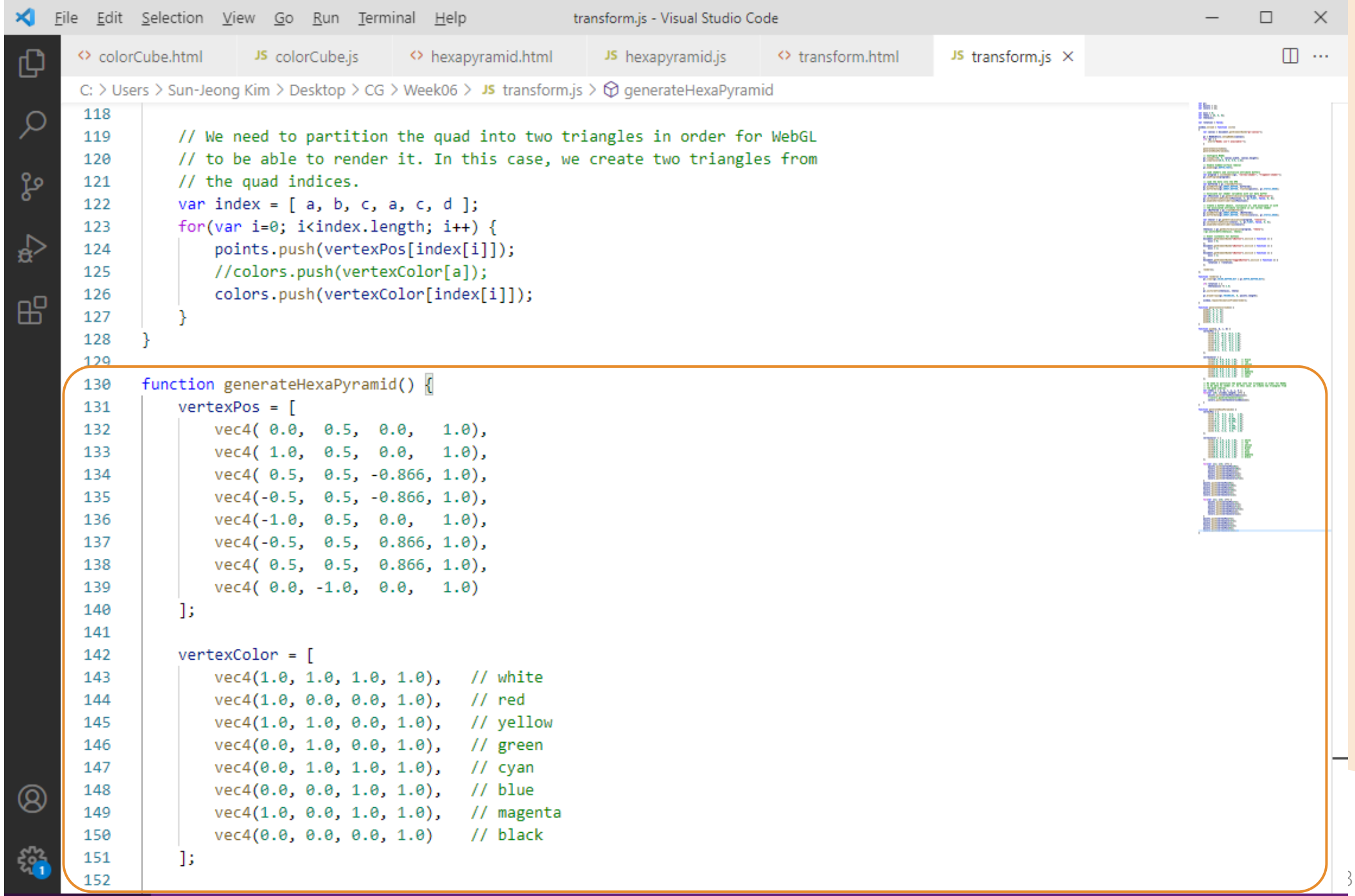
File Edit Selection View Go Run Terminal Help transform.js - Visual Studio Code

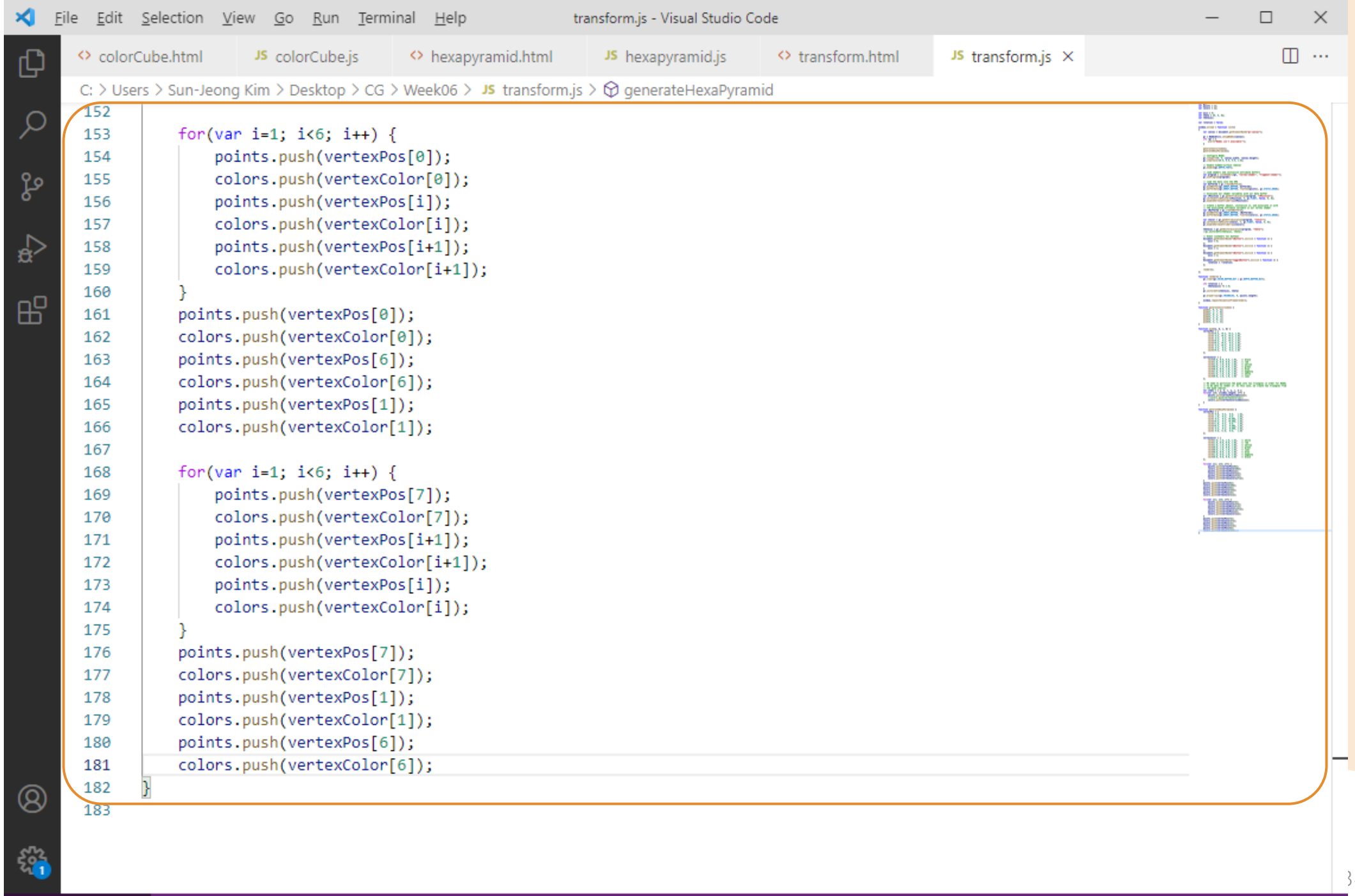
colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js

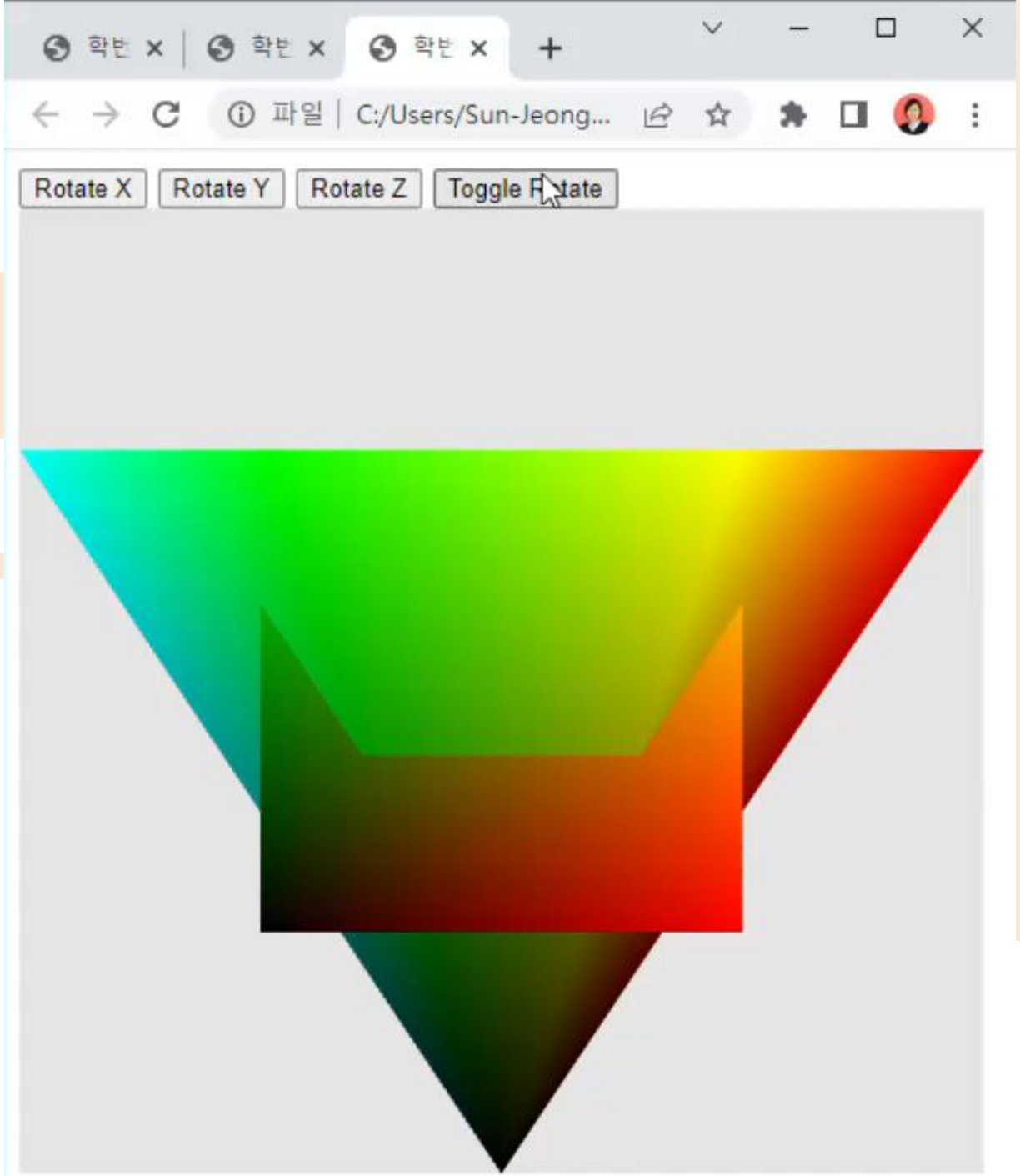
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.js > render

```
73
74  function render() {
75      gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
76
77      if( rotation ) {
78          theta[axis] += 2.0;
79      }
80      gl.uniform3fv(thetaLoc, theta)
81
82      gl.drawArrays(gl.TRIANGLES, 0, points.length);
83
84      window.requestAnimationFrame(render);
85  }
86
87  function generateColorCube() {
88      quad(1, 0, 3, 2);
89      quad(2, 3, 7, 6);
90      quad(3, 0, 4, 7);
91      quad(4, 5, 6, 7);
92      quad(5, 4, 0, 1);
93      quad(6, 5, 1, 2);
94  }
95
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
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 81, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

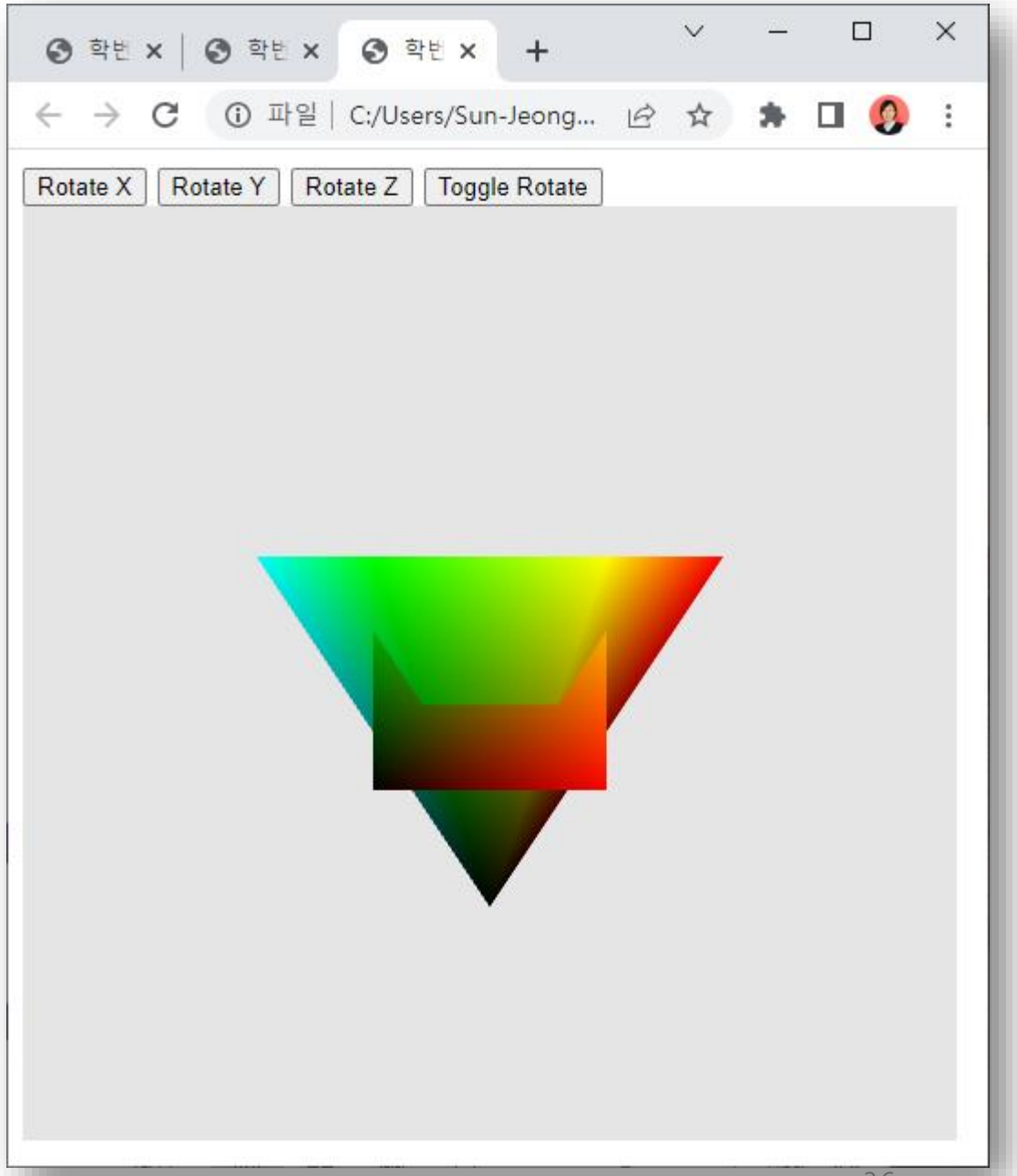

File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.js > generateHexaPyramid
118
119 // We need to partition the quad into 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[a]);
126 colors.push(vertexColor[index[i]]);
127 }
128 }
129
130 function generateHexaPyramid() {
131 vertexPos = [
132 vec4(0.0, 0.5, 0.0, 1.0),
133 vec4(1.0, 0.5, 0.0, 1.0),
134 vec4(0.5, 0.5, -0.866, 1.0),
135 vec4(-0.5, 0.5, -0.866, 1.0),
136 vec4(-1.0, 0.5, 0.0, 1.0),
137 vec4(-0.5, 0.5, 0.866, 1.0),
138 vec4(0.5, 0.5, 0.866, 1.0),
139 vec4(0.0, -1.0, 0.0, 1.0)
140];
141
142 vertexColor = [
143 vec4(1.0, 1.0, 1.0, 1.0), // white
144 vec4(1.0, 0.0, 0.0, 1.0), // red
145 vec4(1.0, 1.0, 0.0, 1.0), // yellow
146 vec4(0.0, 1.0, 0.0, 1.0), // green
147 vec4(0.0, 1.0, 1.0, 1.0), // cyan
148 vec4(0.0, 0.0, 1.0, 1.0), // blue
149 vec4(1.0, 0.0, 1.0, 1.0), // magenta
150 vec4(0.0, 0.0, 0.0, 1.0) // black
151];
152
Restricted Mode
0 0
Ln 181, Col 33
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.js > generateHexaPyramid
152
153 for(var i=1; i<6; i++) {
154 points.push(vertexPos[0]);
155 colors.push(vertexColor[0]);
156 points.push(vertexPos[i]);
157 colors.push(vertexColor[i]);
158 points.push(vertexPos[i+1]);
159 colors.push(vertexColor[i+1]);
160 }
161 points.push(vertexPos[0]);
162 colors.push(vertexColor[0]);
163 points.push(vertexPos[6]);
164 colors.push(vertexColor[6]);
165 points.push(vertexPos[1]);
166 colors.push(vertexColor[1]);
167
168 for(var i=1; i<6; i++) {
169 points.push(vertexPos[7]);
170 colors.push(vertexColor[7]);
171 points.push(vertexPos[i+1]);
172 colors.push(vertexColor[i+1]);
173 points.push(vertexPos[i]);
174 colors.push(vertexColor[i]);
175 }
176 points.push(vertexPos[7]);
177 colors.push(vertexColor[7]);
178 points.push(vertexPos[1]);
179 colors.push(vertexColor[1]);
180 points.push(vertexPos[6]);
181 colors.push(vertexColor[6]);
182 }
183
Restricted Mode
0 0
Ln 181, Col 33
Spaces: 4
UTF-8
CRLF
JavaScript


학반
학반
학반
파일 | C:/Users/Sun-Jeong...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

연습 문제 (2)

- 오브젝트들의 크기를 반으로 줄이시오.



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

colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js

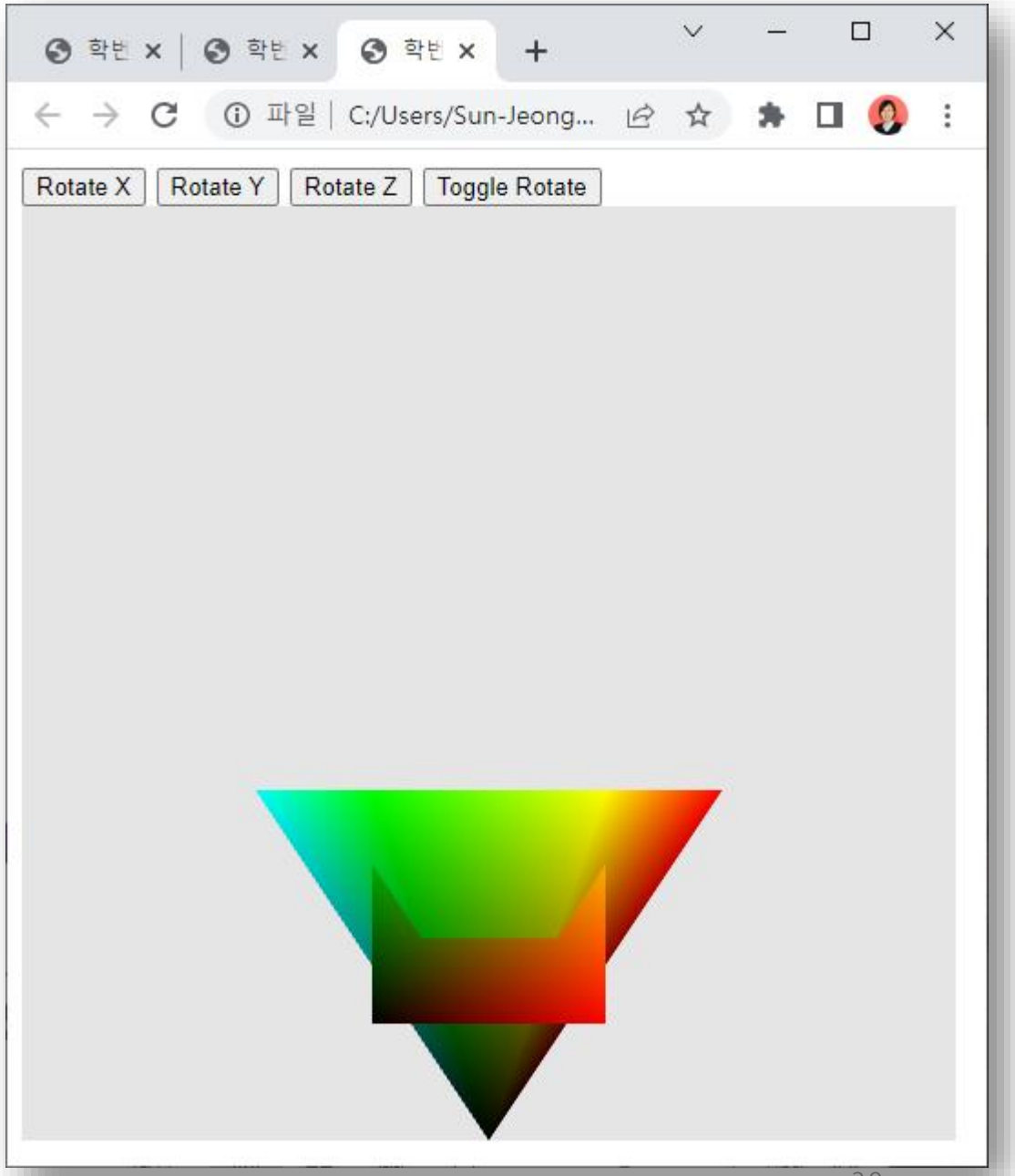
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.html > html > head > script#vertex-shader

```
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          mat4 sc = mat4( 0.5, 0.0, 0.0, 0.0,
35                          0.0, 0.5, 0.0, 0.0,
36                          0.0, 0.0, 0.5, 0.0,
37                          0.0, 0.0, 0.0, 1.0 );
38
39          gl_Position = rz * ry * rx * sc * vPosition;
40          fColor = vColor;
41      }
42      </script>
43
44      <script id="fragment-shader" type="x-shader/x-fragment">
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 39, Col 45 Spaces: 4 UTF-8 CRLF HTML

연습 문제 (3)

- 오브젝트들을 y축 방향으로 -0.5 만큼 이동 시키시오.



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

colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js

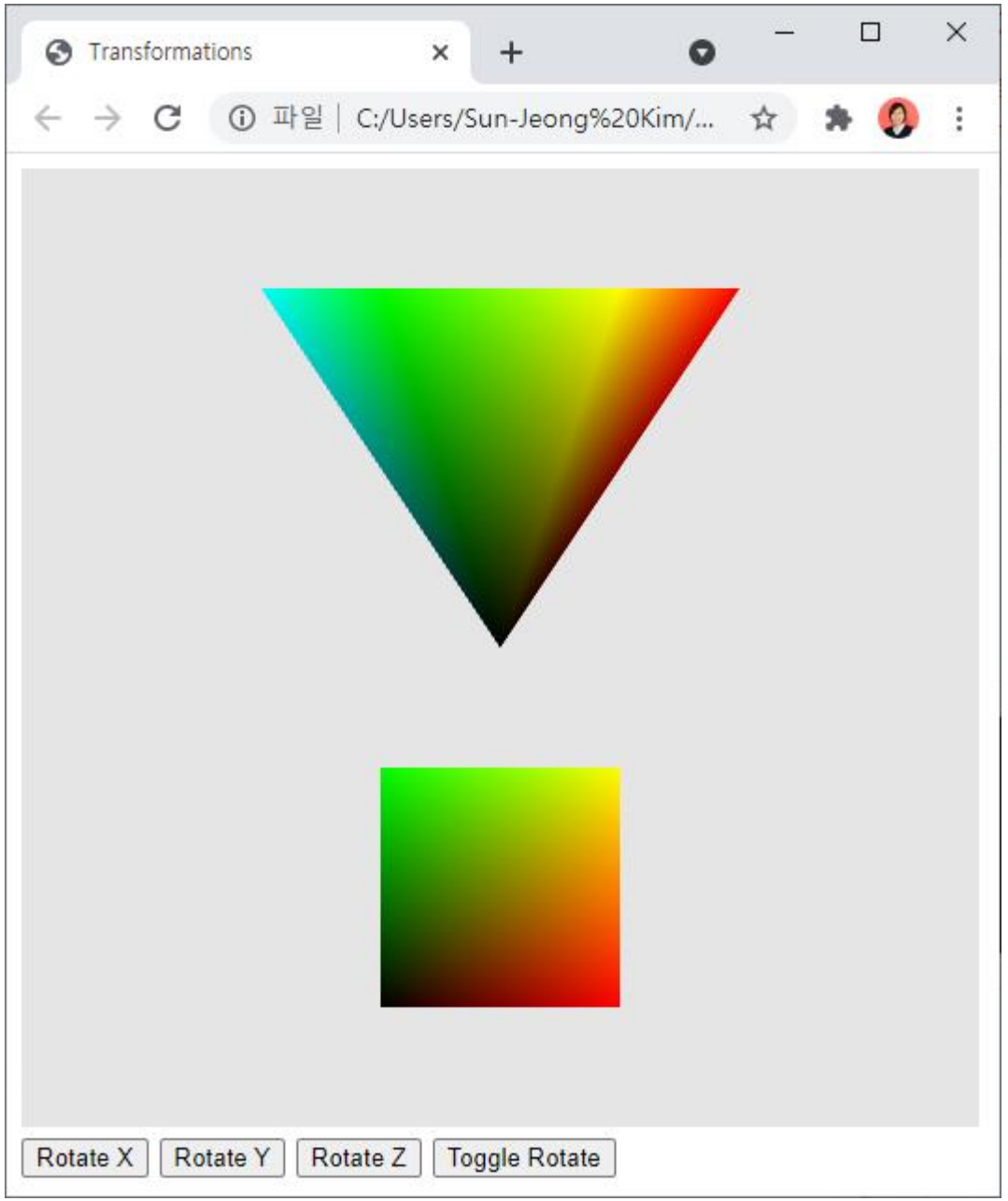
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.html > html > head > script#vertex-shader

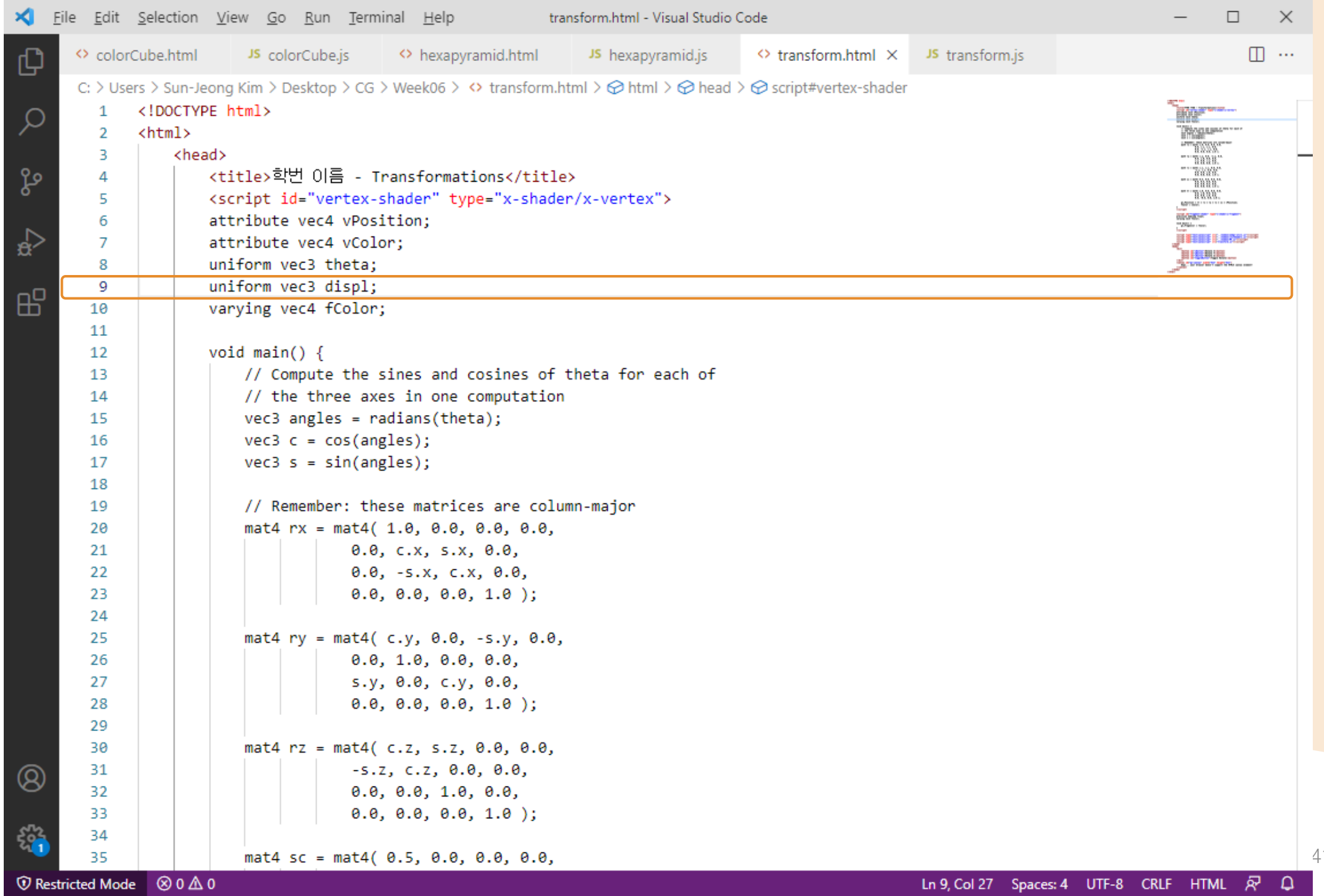
```
11      void main() {
12          // Compute the sines and cosines of theta for each of
13          // the three axes in one computation
14          vec3 angles = radians(theta);
15          vec3 c = cos(angles);
16          vec3 s = sin(angles);
17
18          // Remember: these matrices are column-major
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          mat4 sc = mat4( 0.5, 0.0, 0.0, 0.0,
35                          0.0, 0.5, 0.0, 0.0,
36                          0.0, 0.0, 0.5, 0.0,
37                          0.0, 0.0, 0.0, 1.0 );
38
39          mat4 tr = mat4( 1.0, 0.0, 0.0, 0.0,
40                          0.0, 1.0, 0.0, 0.0,
41                          0.0, 0.0, 1.0, 0.0,
42                          0.0, -0.5, 0.0, 1.0 );
43
44          gl_Position = tr * rz * ry * rx * sc * vPosition;
45          fColor = vColor;
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 44, Col 32 Spaces: 4 UTF-8 CRLF HTML

연습 문제 (4)

- Hexagonal Pyramid를 y축 방향으로 0.5만큼 이동 시키시오.




File Edit Selection View Go Run Terminal Help
transform.html - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.html > html > head > script#vertex-shader
<!DOCTYPE html>
<html>
<head>
<title>학번 이름 - Transformations</title>
<script id="vertex-shader" type="x-shader/x-vertex">
attribute vec4 vPosition;
attribute vec4 vColor;
uniform vec3 theta;
uniform vec3 displ;
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, 0.0,
Restricted Mode
Ln 9, Col 27
Spaces: 4
UTF-8
CRLF
HTML

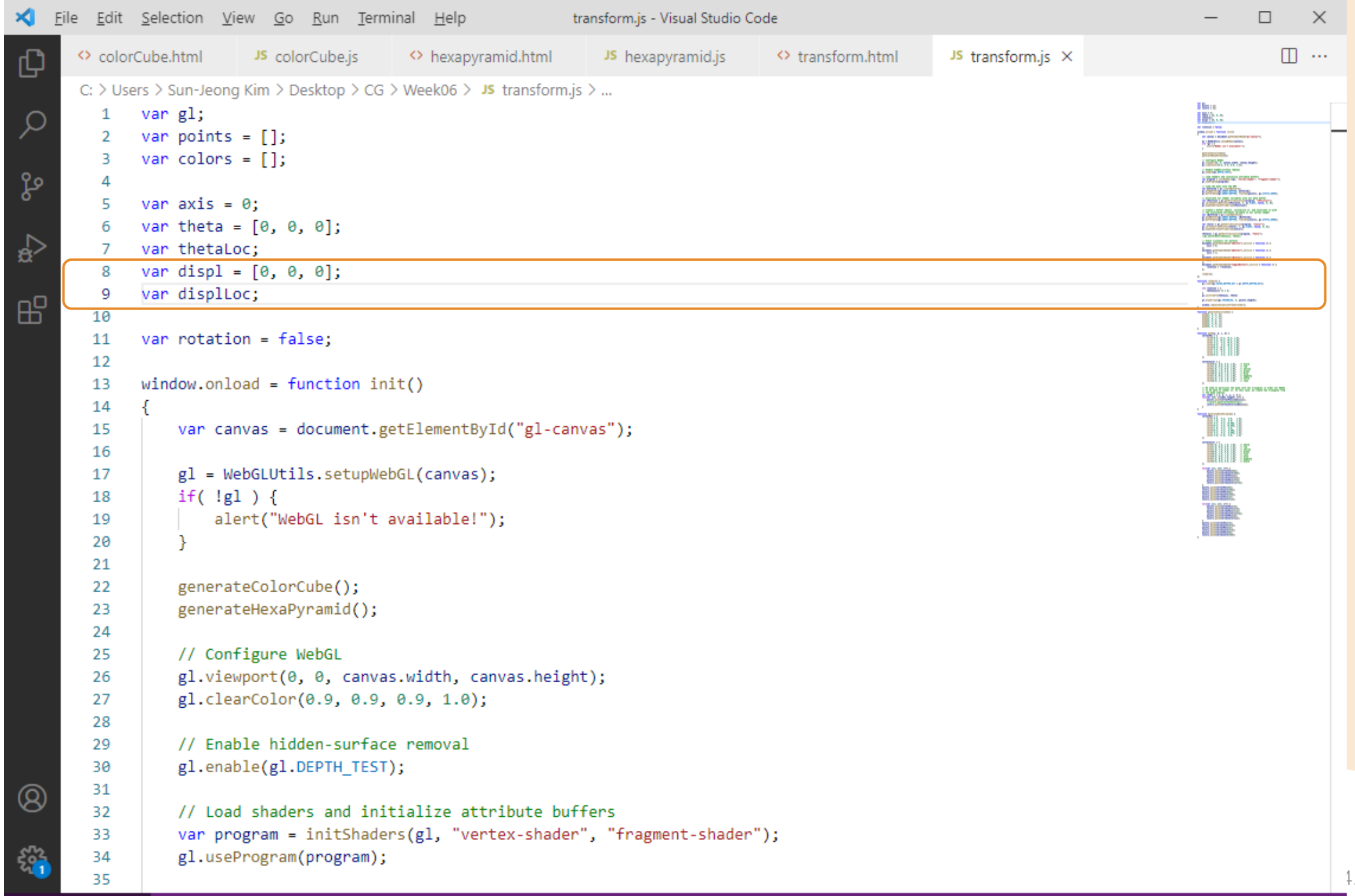
File Edit Selection View Go Run Terminal Help transform.html - Visual Studio Code

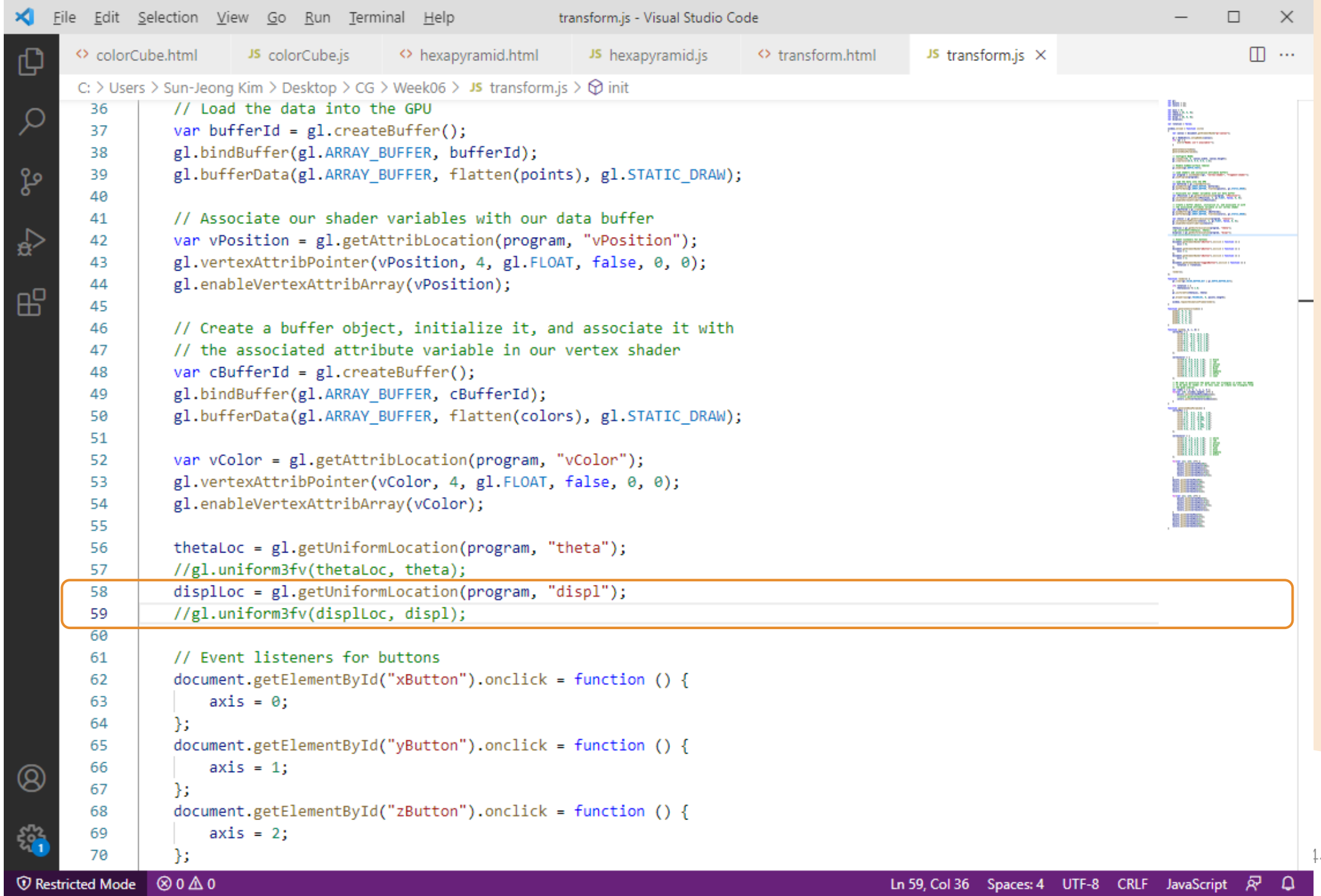
colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.html > html > head > script#vertex-shader

```
12      void main() {
13          // Compute the sines and cosines of theta for each of
14          // the three axes in one computation
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, 0.0, 0.0,
36                          0.0, 0.5, 0.0, 0.0,
37                          0.0, 0.0, 0.5, 0.0,
38                          0.0, 0.0, 0.0, 1.0 );
39
40          mat4 tr = mat4( 1.0, 0.0, 0.0, 0.0,
41                          0.0, 1.0, 0.0, 0.0,
42                          0.0, 0.0, 1.0, 0.0,
43                          displ.x, displ.y, displ.z, 1.0 );
44
45          gl_Position = tr * rz * ry * rx * sc * vPosition;
46          fColor = vColor;
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 43, Col 50 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.js > ...
1 var gl;
2 var points = [];
3 var colors = [];
4
5 var axis = 0;
6 var theta = [0, 0, 0];
7 var thetaLoc;
8 var displ = [0, 0, 0];
9 var displLoc;
10
11 var rotation = false;
12
13 window.onload = function init()
14 {
15 var canvas = document.getElementById("gl-canvas");
16
17 gl = WebGLUtils.setupWebGL(canvas);
18 if(!gl) {
19 alert("WebGL isn't available!");
20 }
21
22 generateColorCube();
23 generateHexaPyramid();
24
25 // Configure WebGL
26 gl.viewport(0, 0, canvas.width, canvas.height);
27 gl.clearColor(0.9, 0.9, 0.9, 1.0);
28
29 // Enable hidden-surface removal
30 gl.enable(gl.DEPTH_TEST);
31
32 // Load shaders and initialize attribute buffers
33 var program = initShaders(gl, "vertex-shader", "fragment-shader");
34 gl.useProgram(program);
35
Restricted Mode
0 0
Ln 9, Col 14
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.js > init
36 // Load the data into the GPU
37 var bufferId = gl.createBuffer();
38 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
39 gl.bufferData(gl.ARRAY_BUFFER, flatten(points), gl.STATIC_DRAW);
40
41 // Associate our shader variables with our data buffer
42 var vPosition = gl.getAttribLocation(program, "vPosition");
43 gl.vertexAttribPointer(vPosition, 4, gl.FLOAT, false, 0, 0);
44 gl.enableVertexAttribArray(vPosition);
45
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 displLoc = gl.getUniformLocation(program, "displ");
59 //gl.uniform3fv(displLoc, displ);
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 };
Restricted Mode
Ln 59, Col 36
Spaces: 4
UTF-8
CRLF
JavaScript

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

colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.js > render

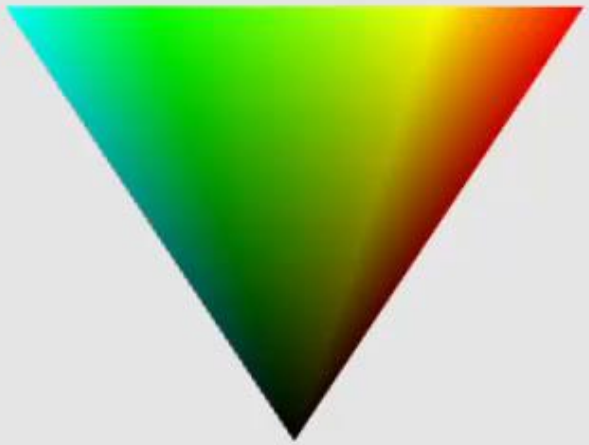
```
71      document.getElementById("toggleButton").onclick = function () {
72          rotation = !rotation;
73      };
74
75      render();
76  };
77
78  function render() {
79      gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
80
81      if( rotation ) {
82          theta[axis] += 2.0;
83      }
84      gl.uniform3fv(thetaLoc, theta)
85
86      // Draw a color cube (12 triangles * 3 = 36 vertices)
87      displ[1] = -0.5;
88      gl.uniform3fv(displLoc, displ);
89      gl.drawArrays(gl.TRIANGLES, 0, 36);
90
91      // Draw a hexagonal pyramid (12 triangles * 3 = 36 vertices)
92      displ[1] = 0.5;
93      gl.uniform3fv(displLoc, displ);
94      gl.drawArrays(gl.TRIANGLES, 36, 36);
95
96      window.requestAnimationFrame(render);
97  }
98
99  function generateColorCube() {
100     quad(1, 0, 3, 2);
101     quad(2, 3, 7, 6);
102     quad(3, 0, 4, 7);
103     quad(4, 5, 6, 7);
104     quad(5, 4, 0, 1);
105     quad(6, 5, 1, 2);
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 94, Col 35 Spaces: 4 UTF-8 CRLF JavaScript

학반 학반 학반

파일 | C:/Users/Sun-Jeong...

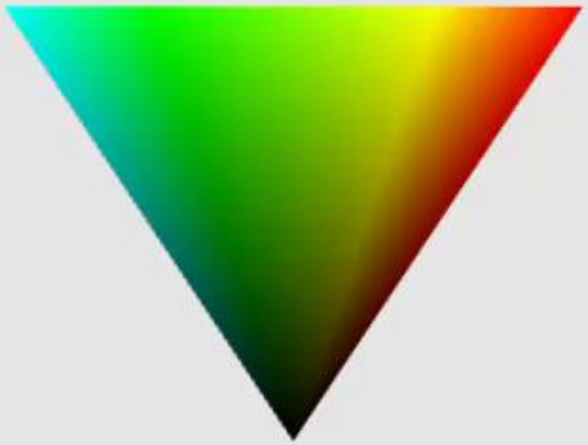
Rotate X Rotate Y Rotate Z Toggle Rotate



학반 학반 학반

파일 | C:/Users/Sun-Jeong...

Rotate X Rotate Y Rotate Z Toggle Rotate



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

colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js

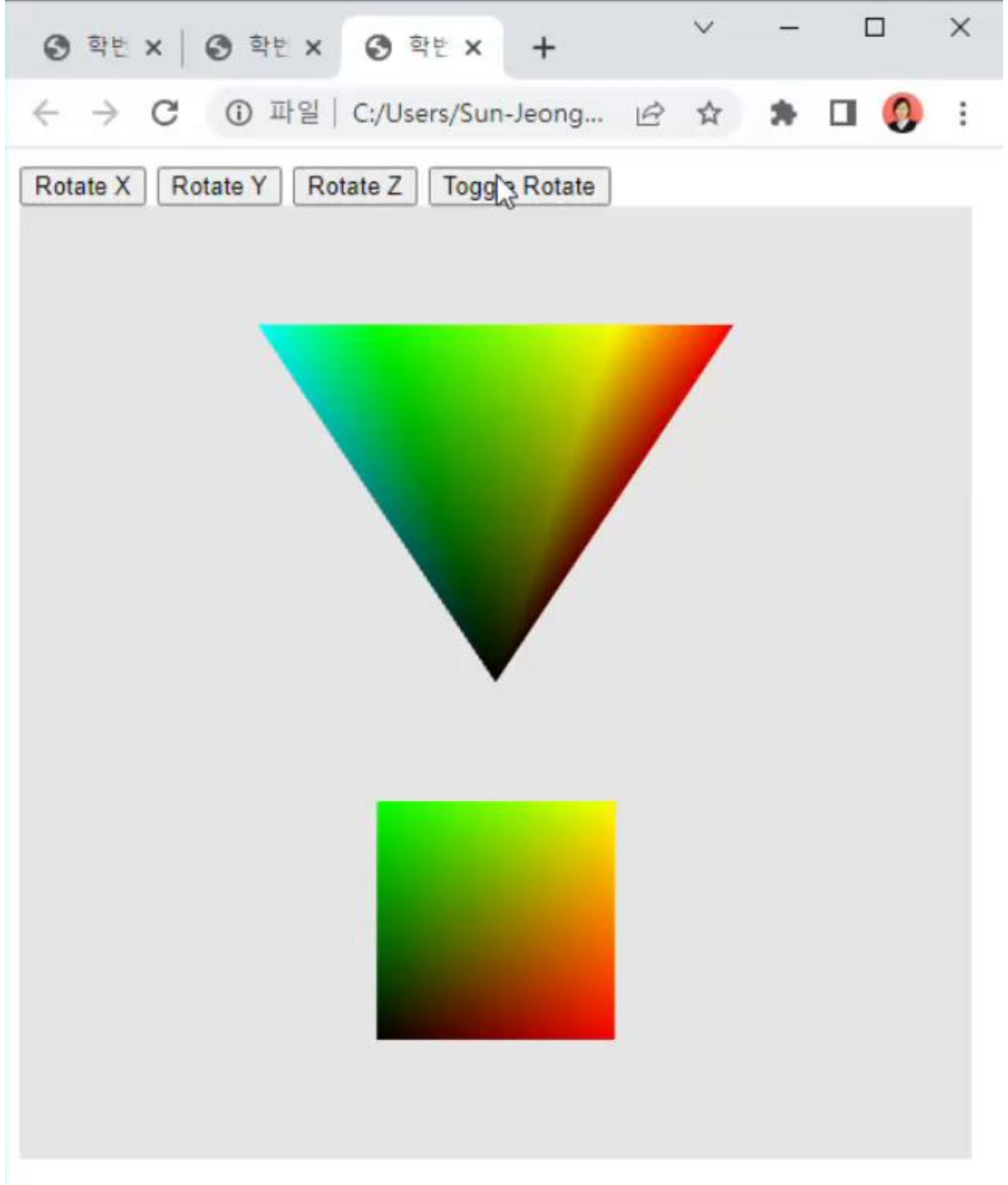
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.html > html > head > script#vertex-shader

```
12      void main() {
13          // Compute the sines and cosines of theta for each of
14          // the three axes in one computation
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, 0.0, 0.0,
36                          0.0, 0.5, 0.0, 0.0,
37                          0.0, 0.0, 0.5, 0.0,
38                          0.0, 0.0, 0.0, 1.0 );
39
40          mat4 tr = mat4( 1.0, 0.0, 0.0, 0.0,
41                          0.0, 1.0, 0.0, 0.0,
42                          0.0, 0.0, 1.0, 0.0,
43                          displ.x, displ.y, displ.z, 1.0 );
44
45          gl_Position = rz * ry * rx * tr * sc * vPosition;
46          fColor = vColor;
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 45, Col 47 Spaces: 4 UTF-8 CRLF HTML

연습 문제 (5)

- 두 오브젝트가 서로 반대 방향으로 회전하도록 구현하시오.

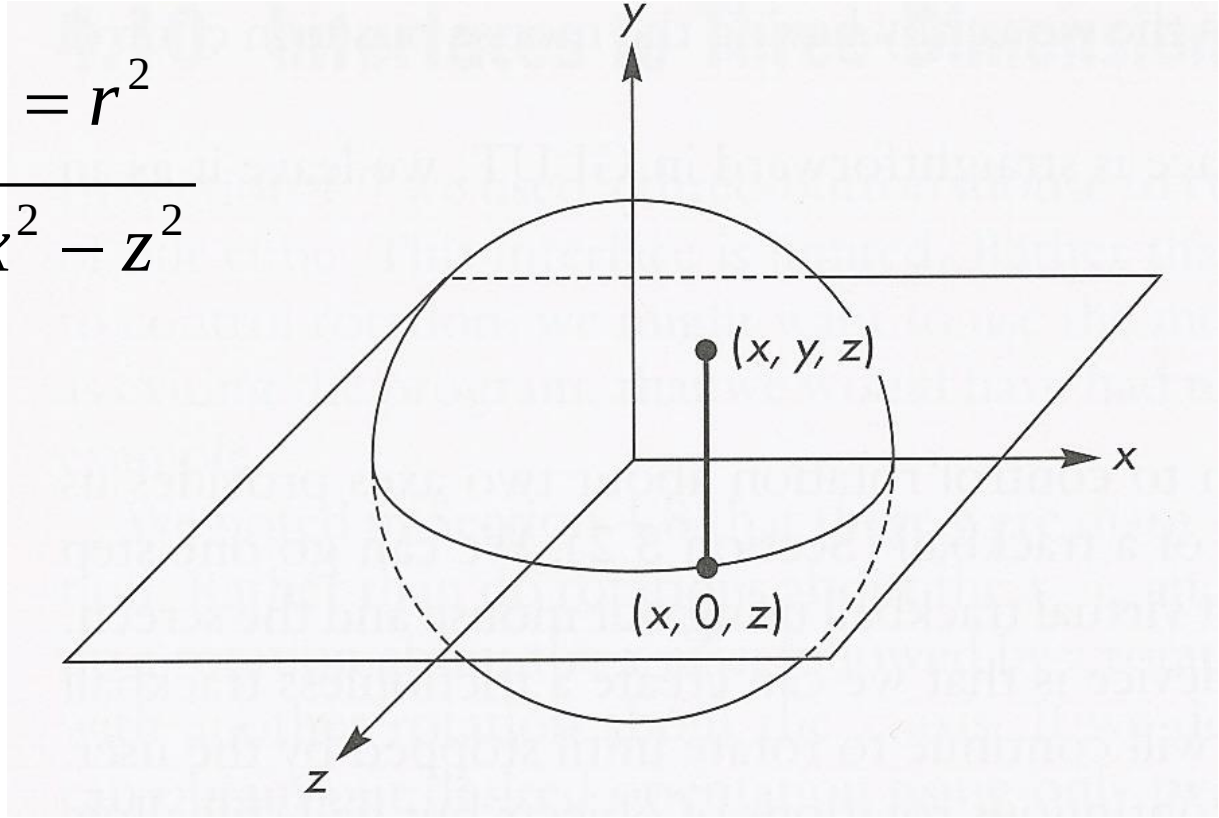


Rotation with a Virtual Trackball (1)

- 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)

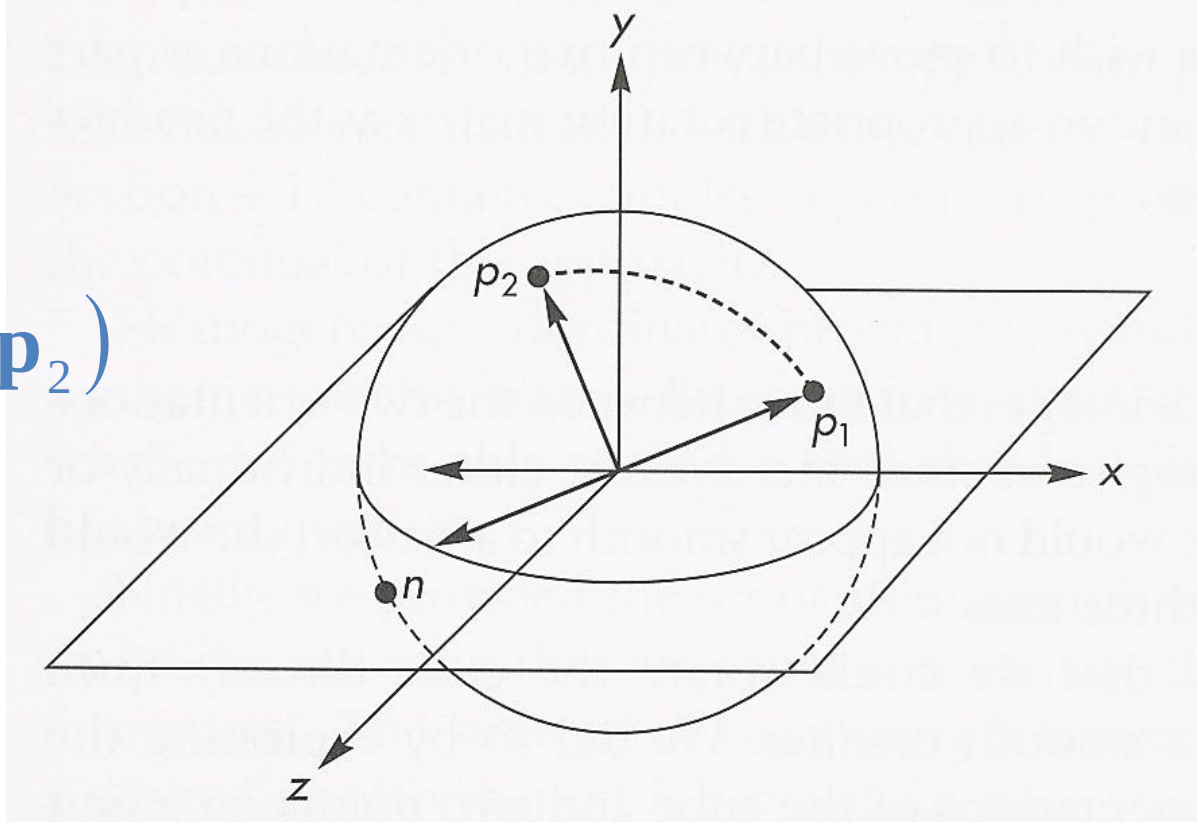
- Determination of the orientation of a plane

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

- 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

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

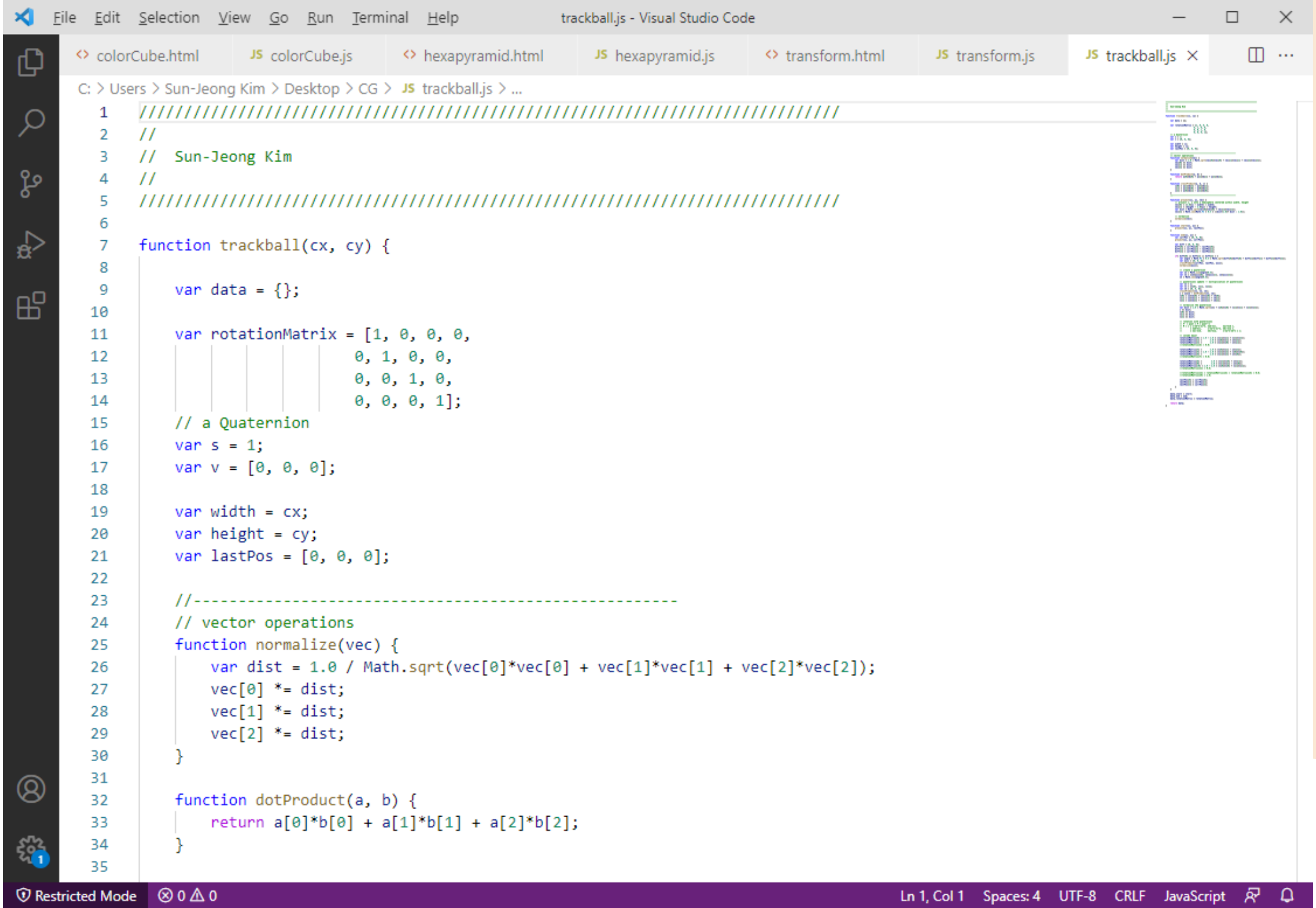
$$\mathbf{p}' = s^2\mathbf{p} + \mathbf{v}(\mathbf{p}\cdot\mathbf{v}) + 2s(\mathbf{v}\times\mathbf{p}) + \mathbf{v}\times(\mathbf{v}\times\mathbf{p})$$

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
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > trackball.js > ...
//
//
// 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];
}
Restricted Mode
0 0
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript

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

colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js | trackball.js

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

```
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[1]*diff[1] + diff[2]*diff[2]);
69          var axis = [0, 0, 0];
70          crossProduct(currPos, lastPos, axis);
```

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

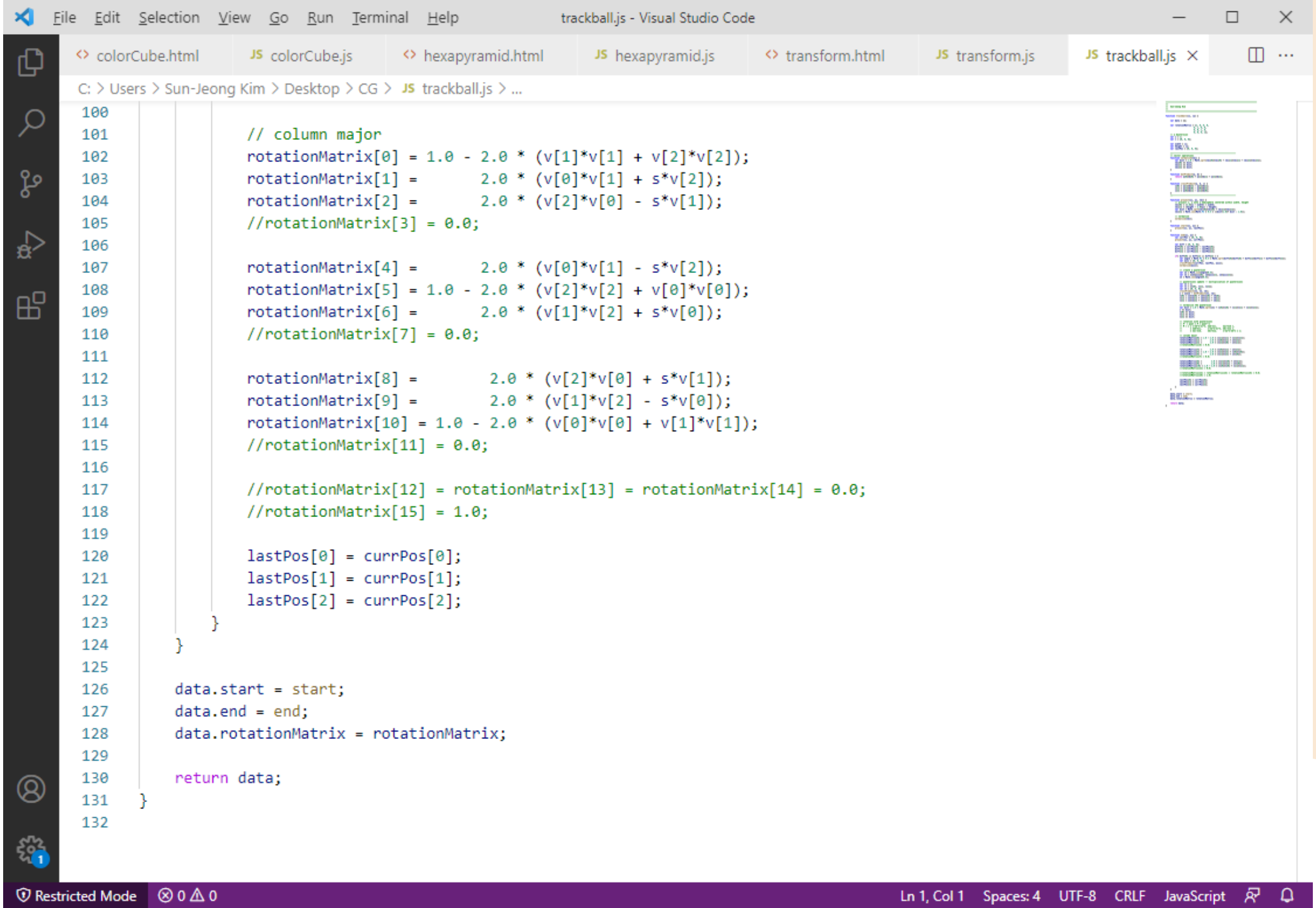
File Edit Selection View Go Run Terminal Help trackball.js - Visual Studio Code

colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js | trackball.js

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

```
67      if( diff[0] || diff[1] || diff[2] ) {
68          var angle = Math.PI * 0.5 * Math.sqrt(diff[0]*diff[0] + diff[1]*diff[1] + diff[2]*diff[2]);
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
```

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


File Edit Selection View Go Run Terminal Help
trackball.js - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > trackball.js > ...
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
Restricted Mode
0 0
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript

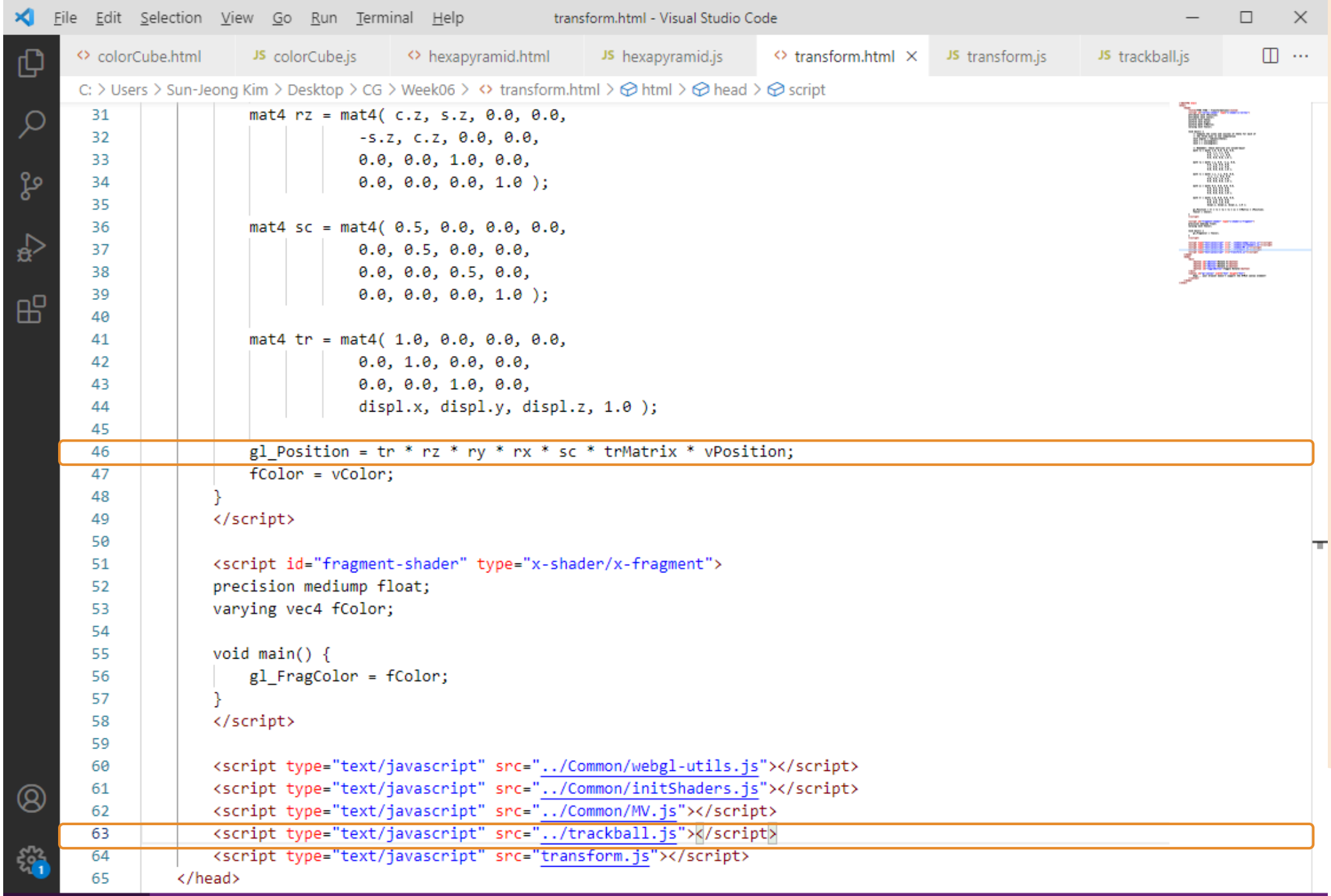
File Edit Selection View Go Run Terminal Help transform.html - Visual Studio Code

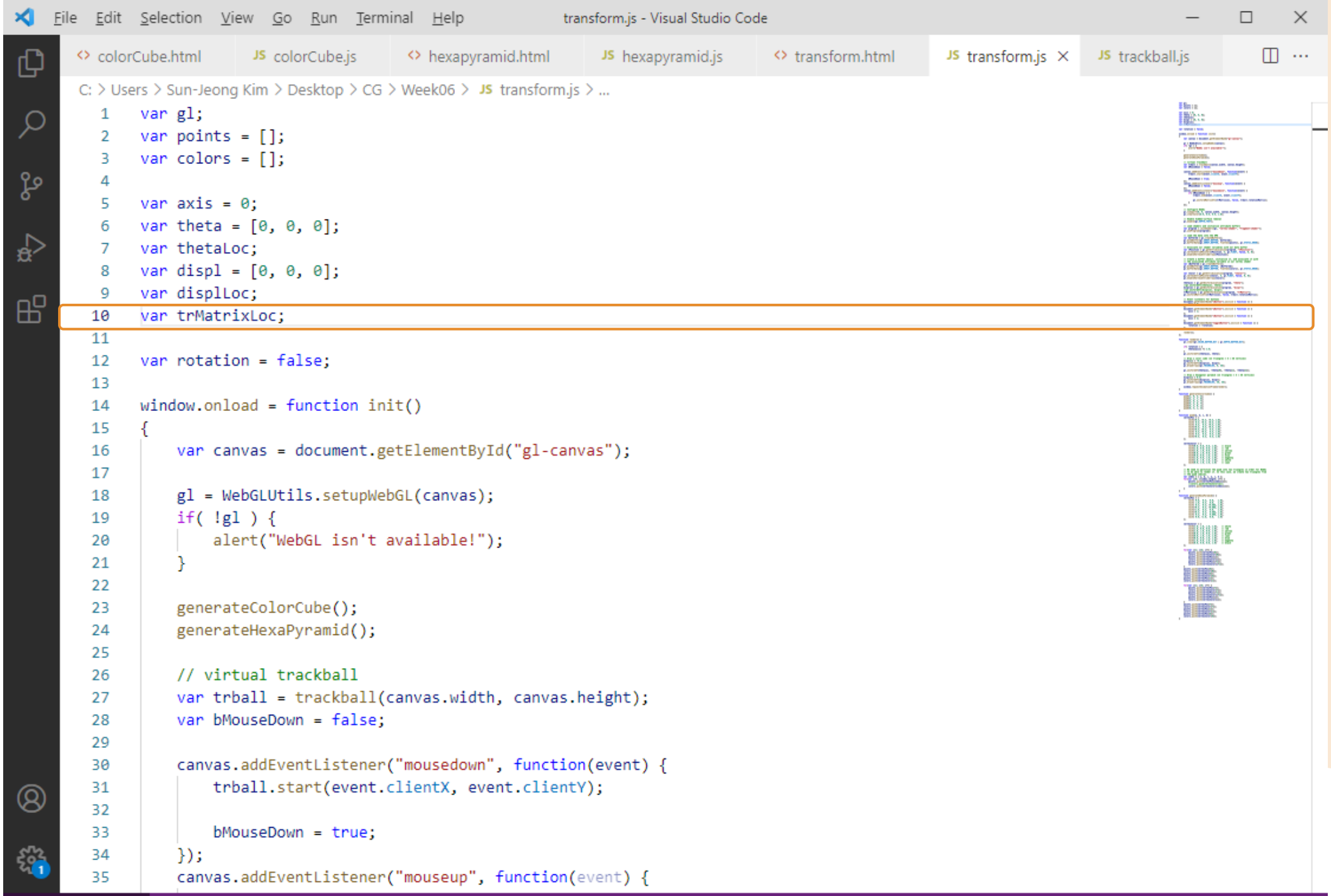
colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js | trackball.js

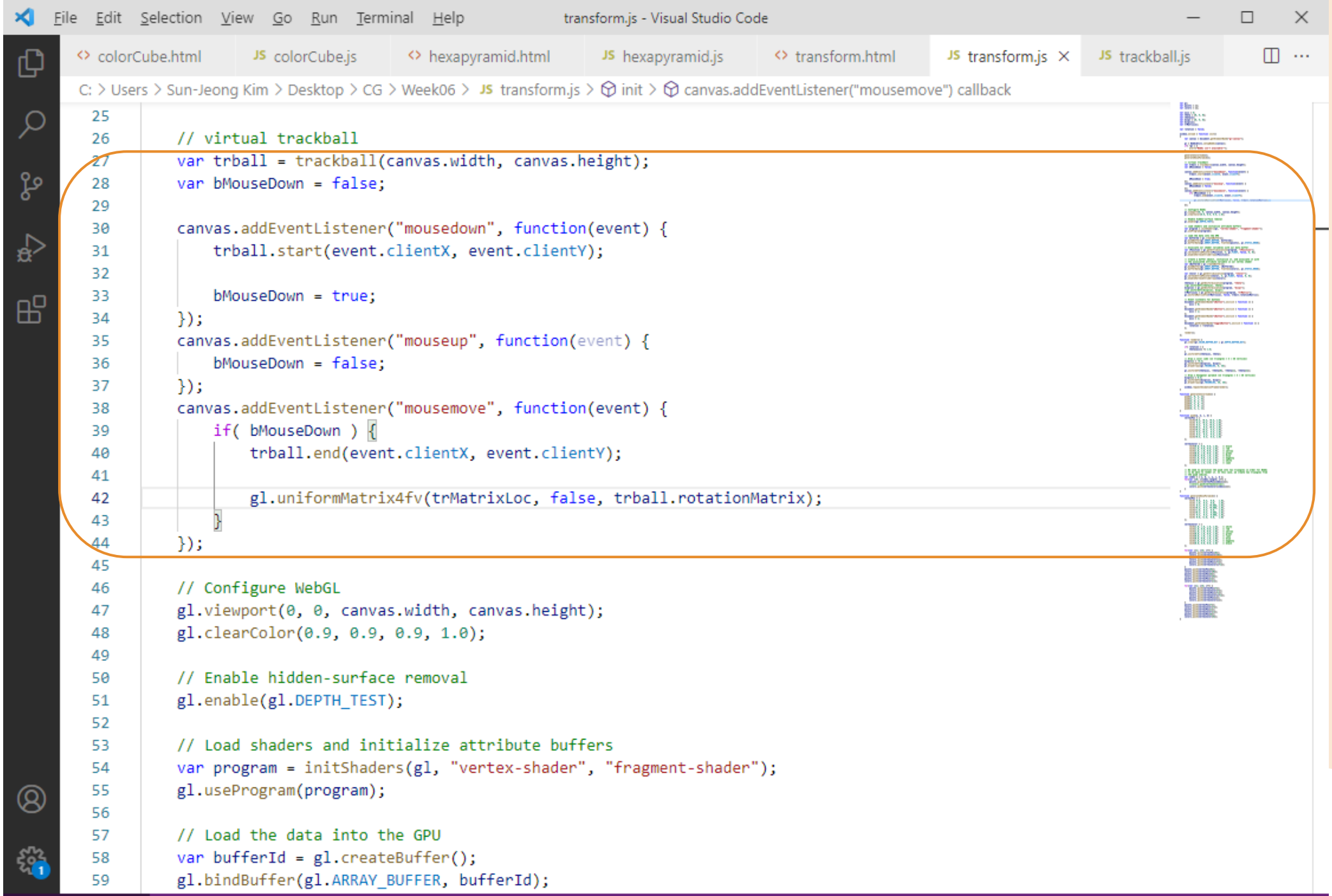
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.html > html > head > script#vertex-shader

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

```

Restricted Mode ⊗ 0 ⚠ 0 Ln 10, Col 31 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
transform.html - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.html > html > head > script
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, 0.0, 0.0,
37 0.0, 0.5, 0.0, 0.0,
38 0.0, 0.0, 0.5, 0.0,
39 0.0, 0.0, 0.0, 1.0);
40
41 mat4 tr = mat4(1.0, 0.0, 0.0, 0.0,
42 0.0, 1.0, 0.0, 0.0,
43 0.0, 0.0, 1.0, 0.0,
44 displ.x, displ.y, displ.z, 1.0);
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>
Restricted Mode
0 0
Ln 63, Col 62
Spaces: 4
UTF-8
CRLF
HTML


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.js > ...
var gl;
var points = [];
var colors = [];
var axis = 0;
var theta = [0, 0, 0];
var thetaLoc;
var displ = [0, 0, 0];
var displLoc;
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 bMouseDown = false;
canvas.addEventListener("mousedown", function(event) {
trball.start(event.clientX, event.clientY);
bMouseDown = true;
});
canvas.addEventListener("mouseup", function(event) {
Restricted Mode
Ln 10, Col 17
Spaces: 4
UTF-8
CRLF
JavaScript


File Edit Selection View Go Run Terminal Help
transform.js - Visual Studio Code
colorCube.html
colorCube.js
hexapyramid.html
hexapyramid.js
transform.html
transform.js
trackball.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.js > init > canvas.addEventListener("mousemove") callback
25
26 // virtual trackball
27 var trball = trackball(canvas.width, canvas.height);
28 var bMouseDown = false;
29
30 canvas.addEventListener("mousedown", function(event) {
31 trball.start(event.clientX, event.clientY);
32
33 bMouseDown = true;
34 });
35 canvas.addEventListener("mouseup", function(event) {
36 bMouseDown = false;
37 });
38 canvas.addEventListener("mousemove", function(event) {
39 if(bMouseDown) {
40 trball.end(event.clientX, event.clientY);
41
42 gl.uniformMatrix4fv(trMatrixLoc, false, trball.rotationMatrix);
43 }
44 });
45
46 // Configure WebGL
47 gl.viewport(0, 0, canvas.width, canvas.height);
48 gl.clearColor(0.9, 0.9, 0.9, 1.0);
49
50 // Enable hidden-surface removal
51 gl.enable(gl.DEPTH_TEST);
52
53 // Load shaders and initialize attribute buffers
54 var program = initShaders(gl, "vertex-shader", "fragment-shader");
55 gl.useProgram(program);
56
57 // Load the data into the GPU
58 var bufferId = gl.createBuffer();
59 gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
Restricted Mode
0 0
Ln 42, Col 76
Spaces: 4
UTF-8
CRLF
JavaScript

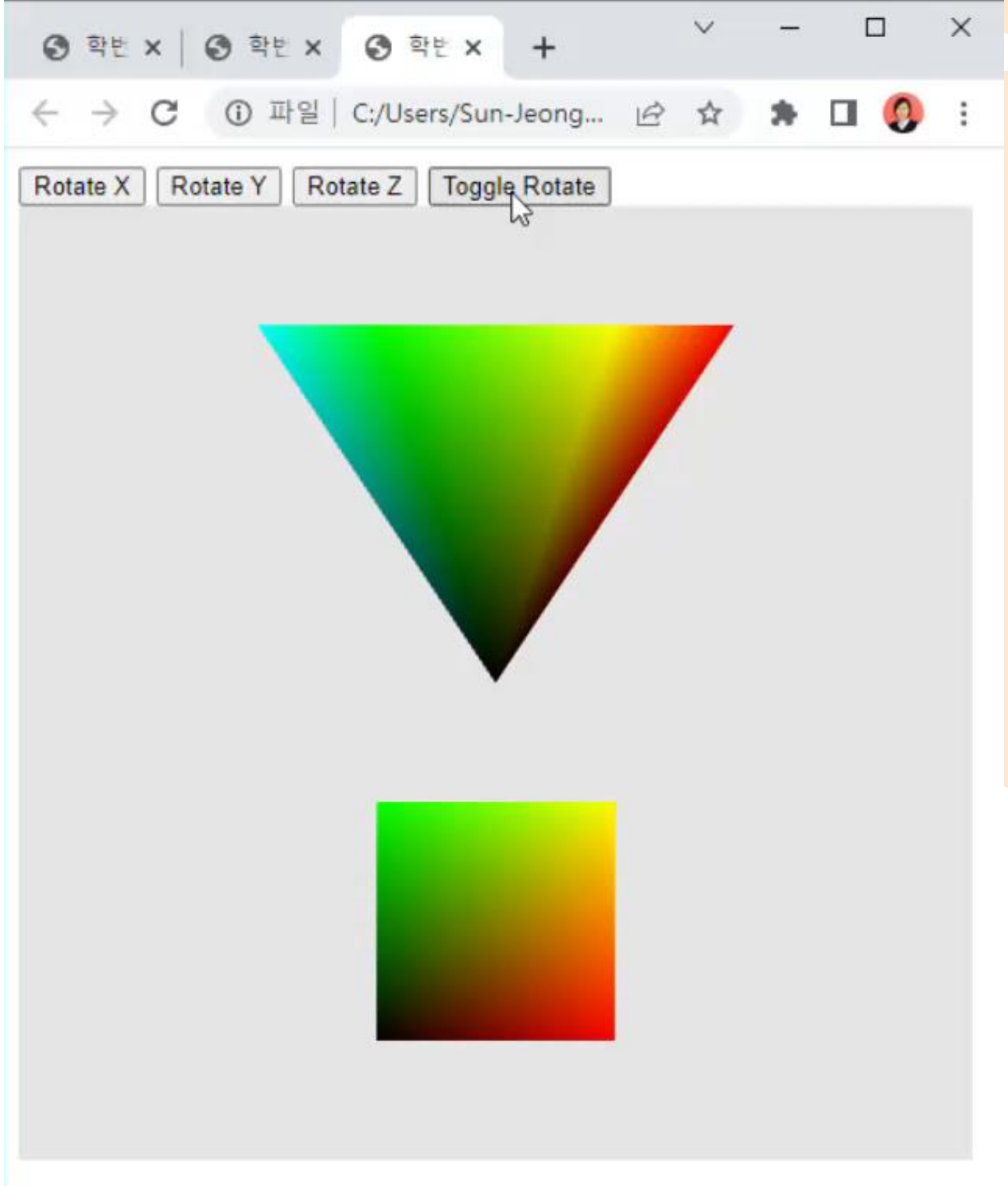
File Edit Selection View Go Run Terminal Help transform.js - Visual Studio Code

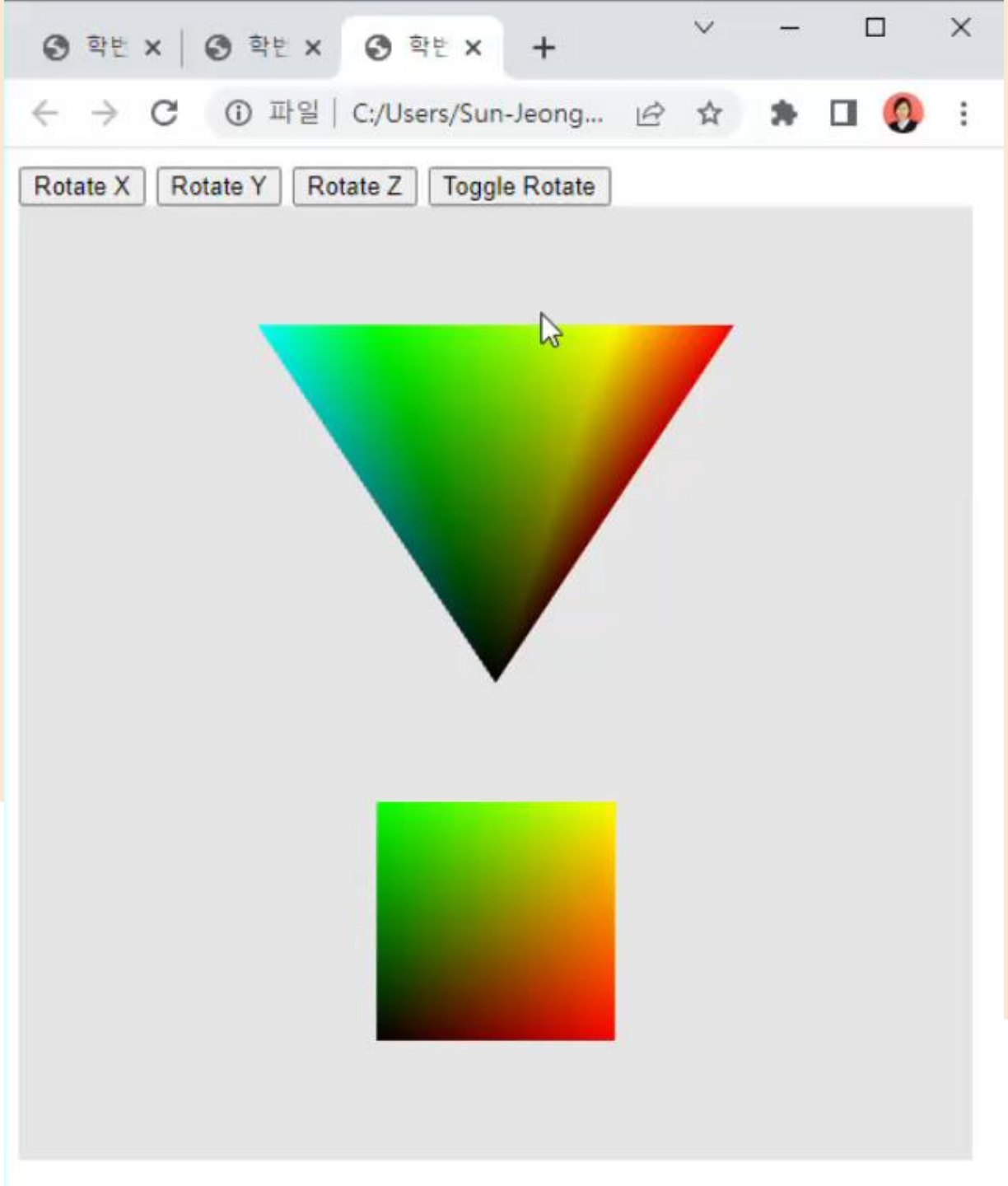
colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js | trackball.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > transform.js > init

```
76
77      thetaLoc = gl.getUniformLocation(program, "theta");
78      //gl.uniform3fv(thetaLoc, theta);
79      displLoc = gl.getUniformLocation(program, "displ");
80      //gl.uniform3fv(displLoc, displ);
81      trMatrixLoc = gl.getUniformLocation(program, "trMatrix");
82      gl.uniformMatrix4fv(trMatrixLoc, false, trball.rotationMatrix);
83
84      // Event listeners for buttons
85      document.getElementById("xButton").onclick = function () {
86          axis = 0;
87      };
88      document.getElementById("yButton").onclick = function () {
89          axis = 1;
90      };
91      document.getElementById("zButton").onclick = function () {
92          axis = 2;
93      };
94      document.getElementById("toggleButton").onclick = function () {
95          rotation = !rotation;
96      };
97
98      render();
99  };
100
101 function render() {
102     gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
103
104     if( rotation ) {
105         theta[axis] += 2.0;
106     }
107     gl.uniform3fv(thetaLoc, theta);
108
109     // Draw a color cube (12 triangles * 3 = 36 vertices)
110     displ[1] = -0.5;
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 82, Col 68 Spaces: 4 UTF-8 CRLF JavaScript


학반
학반
학반
파일 | C:/Users/Sun-Jeong...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate


학반
학반
학반
파일 | C:/Users/Sun-Jeong...
Rotate X
Rotate Y
Rotate Z
Toggle Rotate

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

colorCube.html | colorCube.js | hexapyramid.html | hexapyramid.js | transform.html | transform.js | trackball.js

C: > Users > Sun-Jeong Kim > Desktop > CG > Week06 > 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, 0.0, 0.0,
37                              0.0, 0.5, 0.0, 0.0,
38                              0.0, 0.0, 0.5, 0.0,
39                              0.0, 0.0, 0.0, 1.0 );
40
41              mat4 tr = mat4( 1.0, 0.0, 0.0, 0.0,
42                              0.0, 1.0, 0.0, 0.0,
43                              0.0, 0.0, 1.0, 0.0,
44                              displ.x, displ.y, displ.z, 1.0 );
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>
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 46, Col 35 Spaces: 4 UTF-8 CRLF HTML

연습 문제 (6)

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