

---

# REPORT

---



## 과제 2: Rotating Pyramid

|     |                     |       |               |
|-----|---------------------|-------|---------------|
| 과목명 | 컴퓨터그래픽스<br>(SW)-1분반 | 담당교수  | 송인식           |
| 학 번 | 32221741            | 전 공   | 소프트웨어         |
| 이 름 | 박성현                 | 제 출 일 | 2023-10-19(수) |

# 내용

|                 |   |
|-----------------|---|
| 1. 개발환경.....    | 3 |
| 2. 설계.....      | 3 |
| 3. 구현.....      | 4 |
| 4. 테스트 결과 ..... | 7 |
| 5. 소스 코드.....   | 8 |

# 1. 개발환경

OS : Windows 10/11

Editor : Visual Studio Code

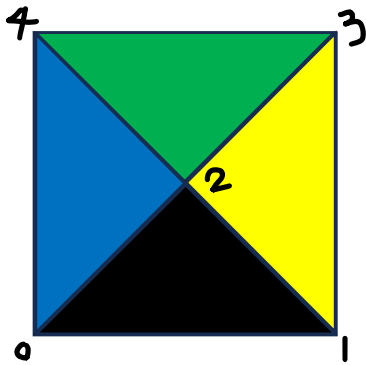
Browser : Microsoft Edge

---

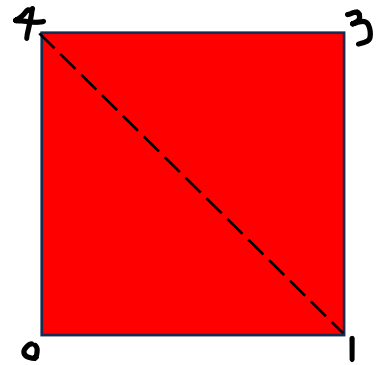
## 2. 설계

1. 예제 'cube.html'과 'cube.js'를 다운로드 받아 이것들을 기반으로 과제를 진행한다.
2. 슬라이더와 선택트(html의 select 태그)를 이용하여 회전 속도를 제어할 수 있도록 한다.
3. cube.js에서 정점의 개수, 좌표 값 등을 수정하여 큐브가 아닌 피라미드 형상을 가지도록 한다.  
아래는 피라미드 정점(vertices)이 인덱스 별로 어떻게 위치하고 있는지 설계한 것이다.

위에서 봤을 때:



아래에서 봤을 때:



### 3. 구현

#### rotatingPyramid.html

```
6  <script id="vertex-shader" type="x-shader/x-vertex">
7
8  attribute vec4 vPosition;
9  attribute vec4 vColor;
10 varying vec4 fColor;
11
12 uniform vec3 theta;
13
14 void main()
15 {
16     // Compute the sines and cosines of theta for each of
17     // the three axes in one computation.
18     vec3 angles = radians( theta );
19     vec3 c = cos( angles );
20     vec3 s = sin( angles );
21
22     // Remember: these matrices are column-major
23     mat4 rx = mat4( 1.0, 0.0, 0.0, 0.0,
24                    0.0, c.x, s.x, 0.0,
25                    0.0, -s.x, c.x, 0.0,
26                    0.0, 0.0, 0.0, 1.0 );
27
28     mat4 ry = mat4( c.y, 0.0, -s.y, 0.0,
29                    0.0, 1.0, 0.0, 0.0,
30                    s.y, 0.0, c.y, 0.0,
31                    0.0, 0.0, 0.0, 1.0 );
32
33
34     mat4 rz = mat4( c.z, s.z, 0.0, 0.0,
35                    -s.z, c.z, 0.0, 0.0,
36                    0.0, 0.0, 1.0, 0.0,
37                    0.0, 0.0, 0.0, 1.0 );
38
39     fColor = vColor;
40     gl_Position = rz * ry * rx * vPosition;
41     gl_Position.z = -gl_Position.z;
42 }
43 </script>
```

cube.html에서 가져온 vertex-shader 스크립트 부분이다.

---

```
71 <div>
72     speed 0%
73     <input id="slider" type="range" min="0" max="100" step="10" value="50" />
74     100%
75 </div>
76 <button id="xButton">Rotate X</button>
77 <button id="yButton">Rotate Y</button>
78 <button id="zButton">Rotate Z</button>
79 <select id="Controls" size="2">
80     <option value="0">Spin Faster</option>
81     <option value="1">Spin Slower</option>
```

cube.html에서 가져온 x,y,zButton들과 새로 추가한 slider와 select 요소다. 브라우저 상에서 이를 통해 회전 방향 및 속도를 제어한다.

## rotatingPyramid.js

```
7 | var NumVertices = 18; // 밑면 삼각형 두개(6개) + 옆면 삼각형 각각 3개(3*4=12) = 18
```

피라미드의 밑면은 삼각형 두 개로 표현한다. 즉,  $3 \times 2 = 6$ 개의 정점이 사용된다.

나머지 옆면들은 각각 삼각형 한 개로 표현하고, 총 4면이므로  $3 \times 4 = 12$ 개의 정점이 사용된다.

즉, 총 18개의 정점이 사용된다.

---

```
30 | | colorPyramid();
```

이 colorPyramid() function(예제에서는 colorCube())로 피라미드를 그리게 된다.

---

```
64 | document.getElementById("slider").onchange = function(event) {
65 |     speed = 100 - event.target.value;
66 | };
67 | document.getElementById("Controls").onclick = function(event) {
68 |     switch(event.target.index) {
69 |         case 0:
70 |             speed /= 2.0;
71 |             break;
72 |         case 1:
73 |             speed *= 2.0;
74 |             break;
75 |     }
76 | };
77 | window.onkeydown = function(event) {
78 |     var key = String.fromCharCode(event.keyCode);
79 |     switch(key) {
80 |         case '1':
81 |             speed /= 2.0;
82 |             break;
83 |         case '2':
84 |             speed *= 2.0;
85 |             break;
86 |     }
87 | };
88 | document.getElementById( "xButton" ).onclick = function () {
89 |     axis = xAxis;
90 | };
91 | document.getElementById( "yButton" ).onclick = function () {
92 |     axis = yAxis;
93 | };
94 | document.getElementById( "zButton" ).onclick = function () {
95 |     axis = zAxis;
96 | };
```

slider는 왼쪽에 가까울수록 회전이 빨라지고 반대는 느려진다. 즉, 100에서 slider 값을 빼서 speed에 저장한다.

select 요소는 인덱스 0이면 회전이 빨라지고 1이면 느려지므로 각각 클릭 시 speed를 1/2, 2배 하도록 한다.

키보드의 키로 1, 2를 입력하면 각각 회전이 빨라지고, 느려지도록 하였다.

x,y,zButton을 누르면 각각 axis에 x,y,zAxis를 저장한다. x,y,zAxis는 각각 0,1,2가 저장되어 있으며, axis는 render() function과정에 사용되며, 회전 방향을 바꾸게 된다.

```

101 function colorPyramid()
102 {
103     triple( 0, 1, 2 ); // 앞면
104     triple( 2, 1, 3 ); // 우측면
105     triple( 3, 4, 2 ); // 뒷면
106     triple( 4, 0, 2 ); // 좌측면
107     triple( 1, 0, 4 ); // 밑면1
108     triple( 1, 4, 3 ); // 밑면2
109 }
110
111 function triple(a, b, c)
112 {
113     var vertices = [
114         vec4( -0.5, -0.5, 0.5, 1.0 ), // 좌측 앞
115         vec4( 0.5, -0.5, 0.5, 1.0 ), // 우측 앞
116         vec4( 0.0, 0.5, 0.0, 1.0 ), // 꼭대기
117         vec4( 0.5, -0.5, -0.5, 1.0 ), // 우측 뒤
118         vec4( -0.5, -0.5, -0.5, 1.0 ) // 좌측 뒤
119     ];
120
121     var vertexColors = [
122         [ 0.0, 0.0, 0.0, 1.0 ], // black
123         [ 1.0, 0.0, 0.0, 1.0 ], // red
124         [ 1.0, 1.0, 0.0, 1.0 ], // yellow
125         [ 0.0, 1.0, 0.0, 1.0 ], // green
126         [ 0.0, 0.0, 1.0, 1.0 ] // blue
127     ];
128
129     //vertex color assigned by the index of the vertex
130
131     var indices = [ a, b, c ];
132
133     for ( var i = 0; i < indices.length; ++i ) {
134         points.push( vertices[indices[i]] );
135         // colors.push( vertexColors[indices[i]] );
136
137         // for solid colored faces use
138         colors.push(vertexColors[a]);
139     }
140 }
141

```

앞서 설명한 colorPyramid() function은 또, triple() function으로 삼각형을 만들 정점의 인덱스를 묶어준다.

triple의 인자로 넘겨진 3개의 정점 배열의 인덱스로 삼각형이 만들어지고 첫 번째 인자를 vertexColors의 인덱스로 사용하여 만들어진 삼각형 면의 색상을 정해준다.

정점은 앞선 보고서의 '설계' 부분에서 표현한 것처럼 묶어주었다.

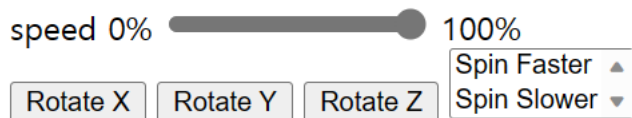
```

143 function render()
144 {
145     gl.clear( gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
146
147     theta[axis] += 2.0;
148     gl.uniform3fv(thetaLoc, theta);
149
150     gl.drawArrays( gl.TRIANGLES, 0, NumVertices );
151
152     setTimeout(
153         function() {requestAnimationFrame( render );},
154         speed
155     );
156 }

```

render() function은 cube.js의 것과 같으나, setTimeout()과 requestAnimationFrame(), speed를 통해 회전 속도를 제어하기 위해 조작할 때, 결과가 나타날 수 있도록 하였다.

## 4. 테스트 결과



정상적으로 피라미드가 출력되고, 회전 속도 및 방향이 제어됨을 확인하였다.

## 5. 소스 코드

소스코드는 GitHub 레포지토리에 있습니다.

[https://github.com/ParkSeonghyeon2003/DKU\\_ComputerGraphics](https://github.com/ParkSeonghyeon2003/DKU_ComputerGraphics)