

컴퓨터 그래픽스

과제 2

학과 : AI 컴퓨터공학부

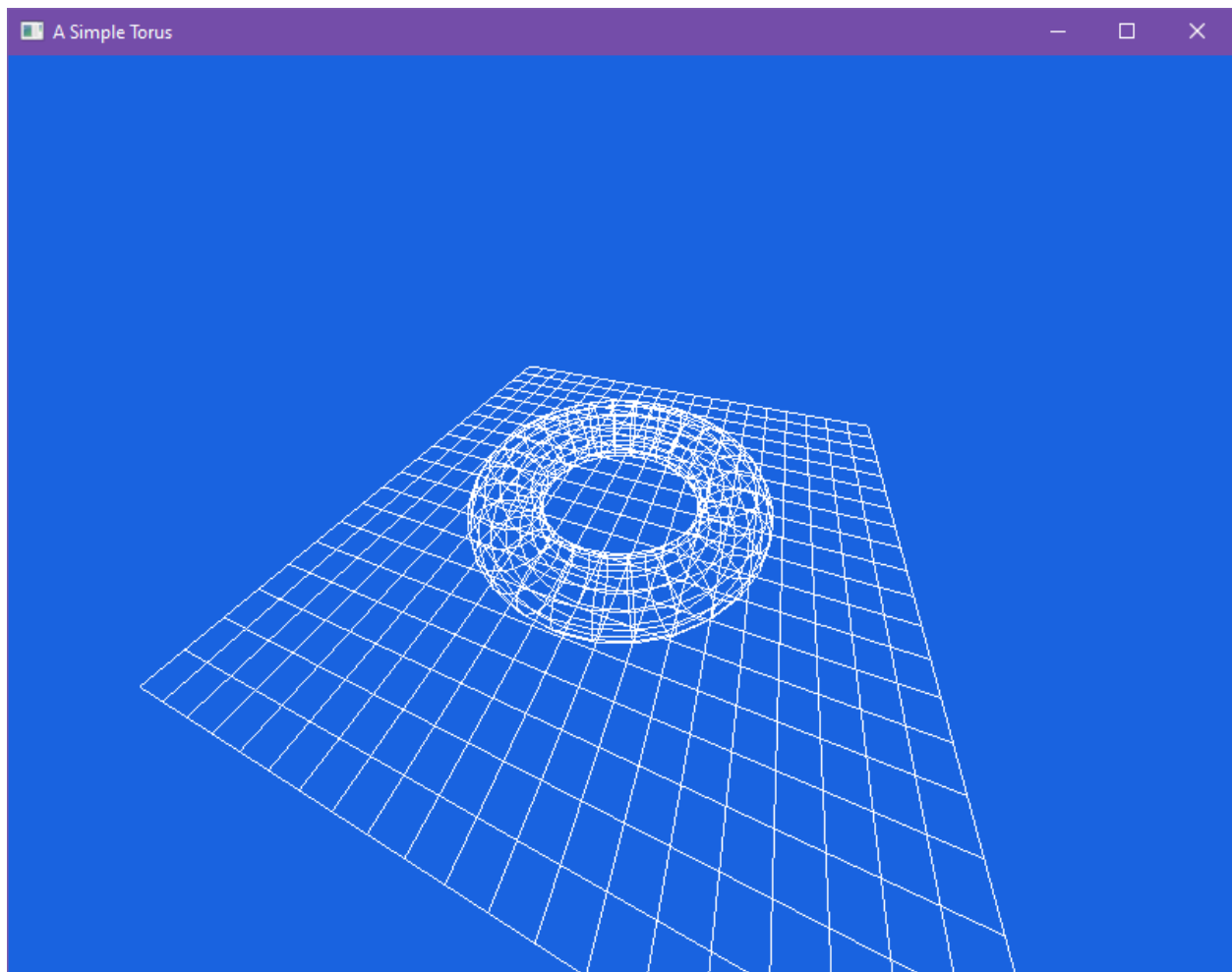
학번 : 201811302

이름 : 손현태

제출 날짜 : 2022-10-02

- (1). Tetrahedron 샘플 프로그램을 수행해 보고,
다음 내용을 반영하여 프로그램을 변경 하여 프로그램과 출력 결과를 제출 하시오.
- (1)-①. Displays a static picture of a torus (instead of a Tetrahedron)
sitting on a grid "floor".

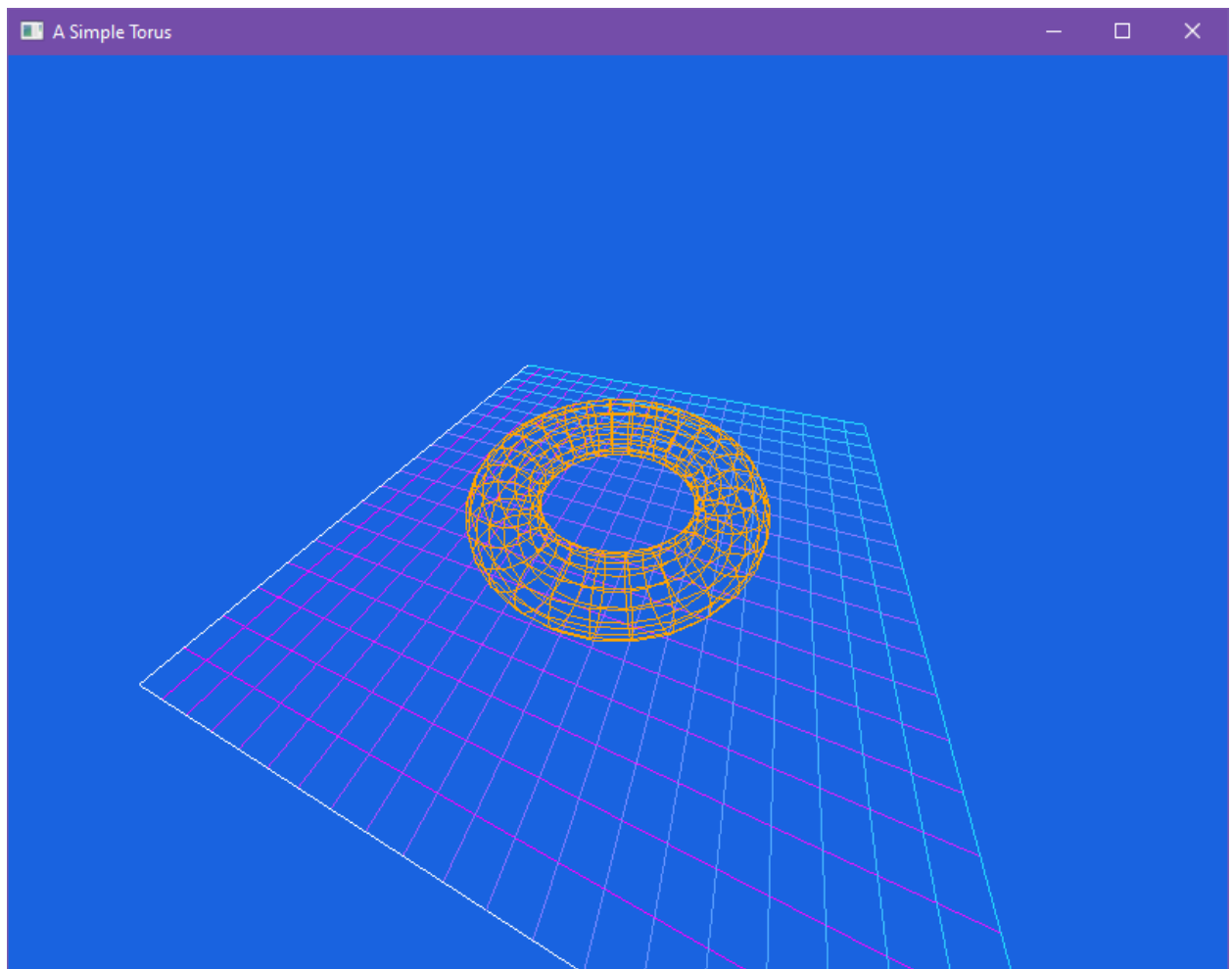
```
P01_01.cpp
P01_01 (Global Scope)
1 #include <GL/glut.h>
2 #include <GL/GL.h>
3 #include <GL/GLU.h>
4
5 // Clears the window and draws the torus.
6 void display() {
7     glClear(GL_COLOR_BUFFER_BIT);
8
9     // Draw a white grid "floor" for the torus to sit on.
10    glColor3f(1, 1, 1);
11    glBegin(GL_LINES);
12    for (GLfloat i = -2.5; i <= 2.5; i += 0.25) {
13        glVertex3f(i, 0, 2.5); glVertex3f(i, 0, -2.5);
14        glVertex3f(2.5, 0, i); glVertex3f(-2.5, 0, i);
15    }
16    glEnd();
17
18    // Draw the torus.
19    glRotatef(-90, 1, 0, 0);
20    glTranslatef(0.0f, 0.0f, 0.3f);
21    glutWireTorus(0.3, 1, 15, 30);
22
23    glFlush();
24 }
25
26 void init() {
27     glClearColor(0.1f, 0.39f, 0.88f, 1.0f);
28     glColor3f(1.0, 1.0, 1.0);
29
30     glEnable(GL_CULL_FACE);
31     glCullFace(GL_BACK);
32
33     glMatrixMode(GL_PROJECTION);
34     glLoadIdentity();
35     glFrustum(-2, 2, -1.5, 1.5, 1, 40);
36
37     glMatrixMode(GL_MODELVIEW);
38     glLoadIdentity();
39     glTranslatef(0, 0, -3);
40     glRotatef(50, 1, 0, 0);
41     glRotatef(70, 0, 1, 0);
42 }
43
44 int main(int argc, char** argv) {
45     glutInit(&argc, argv);
46     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
47     glutInitWindowPosition(80, 80);
48     glutInitWindowSize(800, 600);
49     glutCreateWindow("A Simple Torus");
50     glutDisplayFunc(display);
51     init();
52     glutMainLoop();
53 }
```



- (1)-②. Torus 에 적절한 컬러 색상을 반영하고,
아울러 grid floor 에도 임의의 컬러 색상을 할당하여 출력하시오.

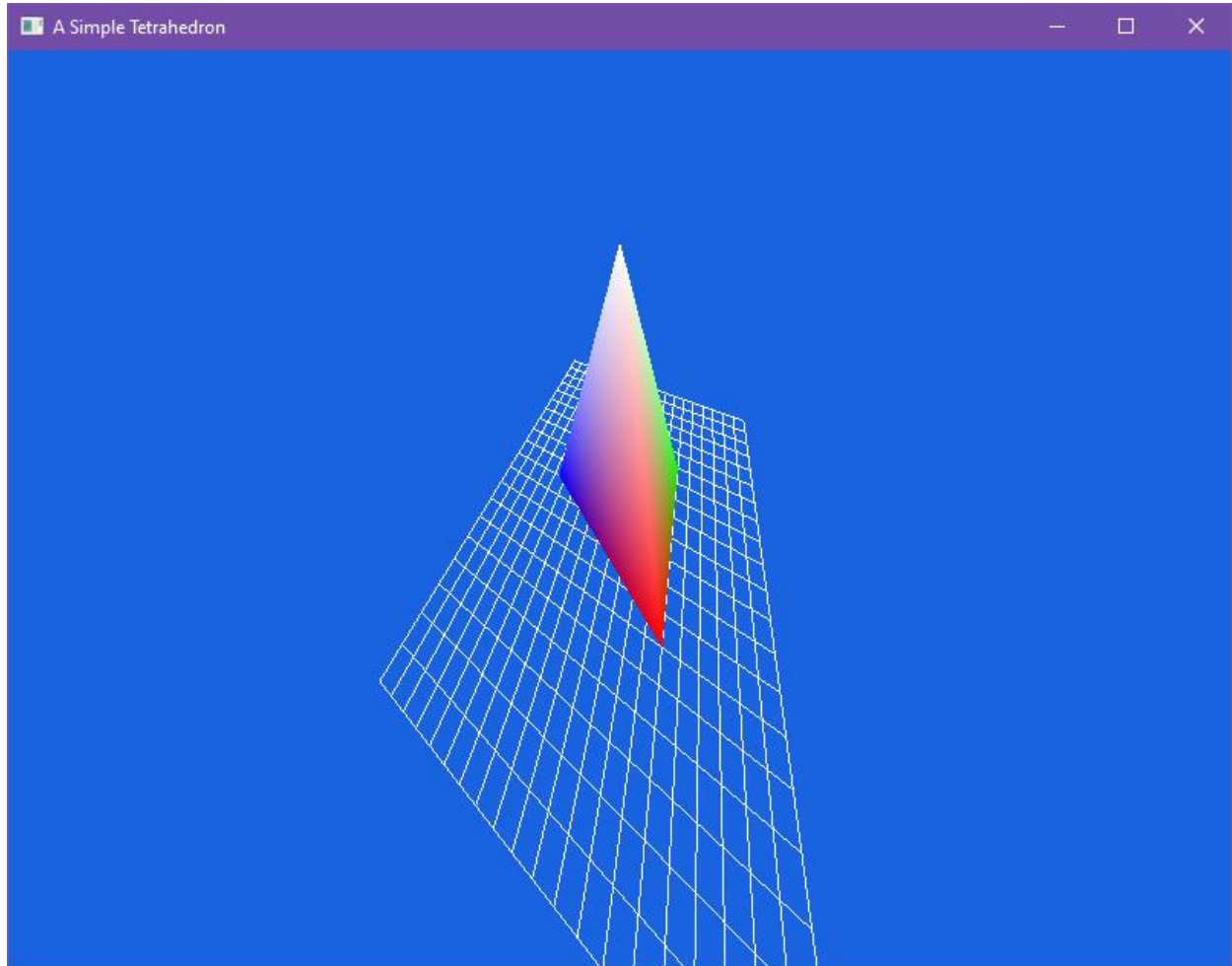
```
P01_02.cpp
P01_02 (Global Scope)

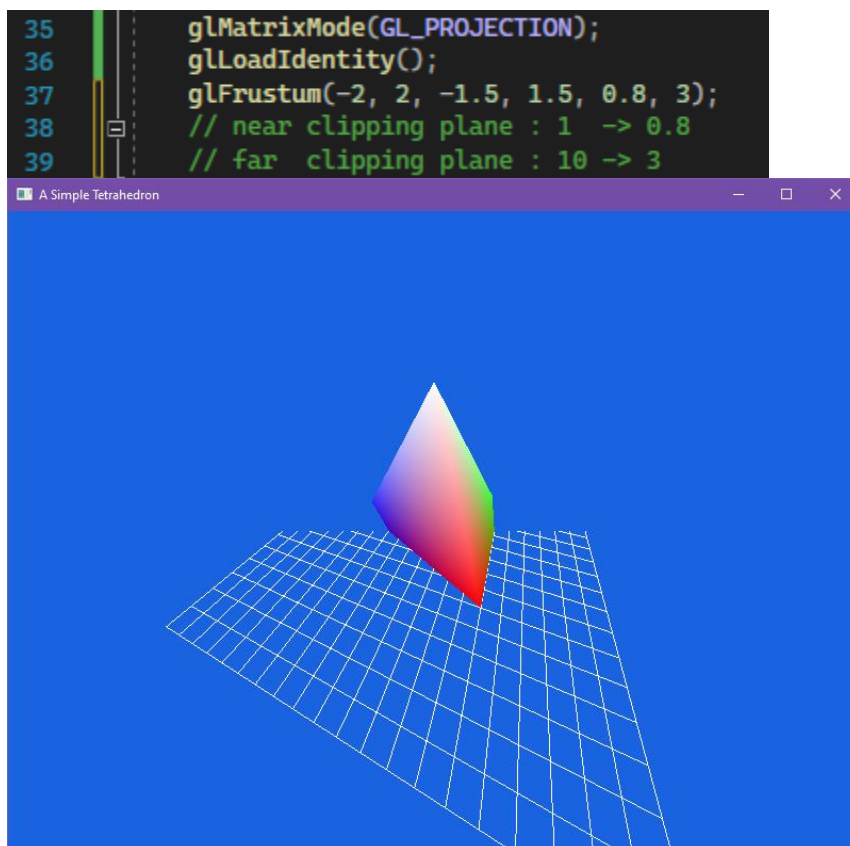
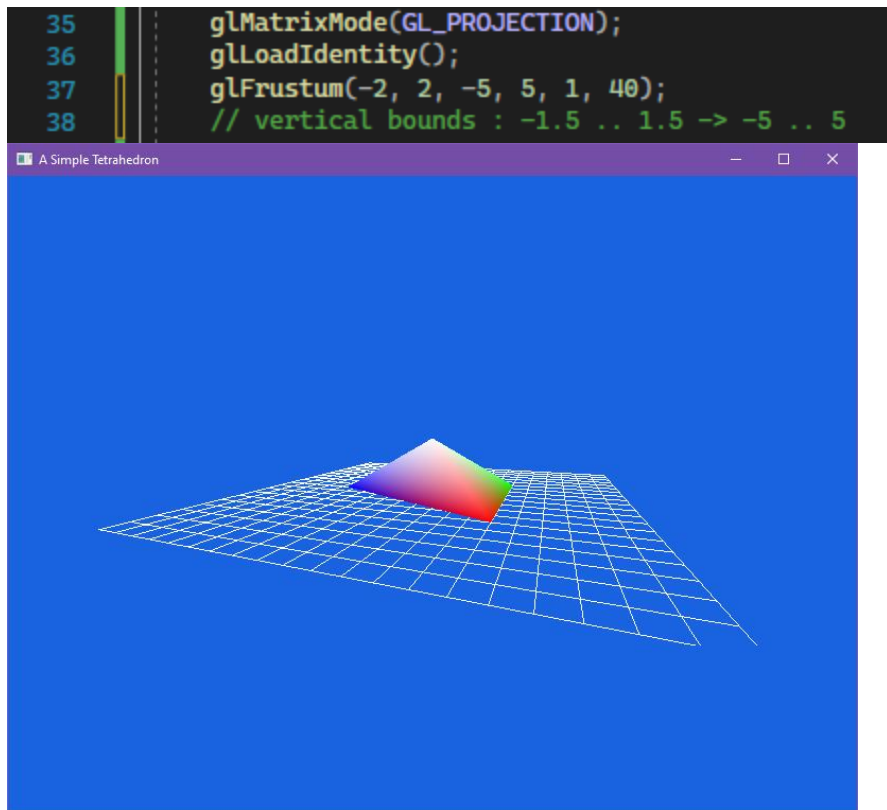
1  #include <GL/glut.h>
2  #include <GL/GL.h>
3  #include <GL/GLU.h>
4
5  // Clears the window and draws the torus.
6  void display() {
7      glClear(GL_COLOR_BUFFER_BIT);
8
9      // Draw a white grid "floor" for the torus to sit on.
10     // variable r and g is change each color step by step
11     glBegin(GL_LINES);
12     float r = 1.0f; float g = 0.0f;
13     for (GLfloat i = -2.5; i <= 2.5; i += 0.25) {
14         glVertex3f(i, 0, 2.5); glVertex3f(i, 0, -2.5);
15         glVertex3f(2.5, 0, i); glVertex3f(-2.5, 0, i);
16         glColor3f(r, g, 1);
17         r -= 0.045f;
18         g += 0.045f;
19     }
20     glEnd();
21
22     // Draw the torus which is orange.
23     glColor3f(1, (float)165 / 255, 0);
24     glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
25     glTranslatef(0.0f, 0.0f, 0.3f);
26     glutWireTorus(0.3, 1, 15, 30);
27
28     glFlush();
29 }
30
31 void init() {
32     glClearColor(0.1, 0.39, 0.88, 1.0);
33     glColor3f(1.0, 1.0, 1.0);
34
35     glEnable(GL_CULL_FACE);
36     glCullFace(GL_BACK);
37
38     glMatrixMode(GL_PROJECTION);
39     glLoadIdentity();
40     glFrustum(-2, 2, -1.5, 1.5, 1, 40);
41
42     glMatrixMode(GL_MODELVIEW);
43     glLoadIdentity();
44     glTranslatef(0, 0, -3);
45     glRotatef(50, 1, 0, 0);
46     glRotatef(70, 0, 1, 0);
47 }
48
49 int main(int argc, char** argv) {
50     glutInit(&argc, argv);
51     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
52     glutInitWindowPosition(80, 80);
53     glutInitWindowSize(800, 600);
54     glutCreateWindow("A Simple Torus");
55     glutDisplayFunc(display);
56     init();
57     glutMainLoop();
58 }
```



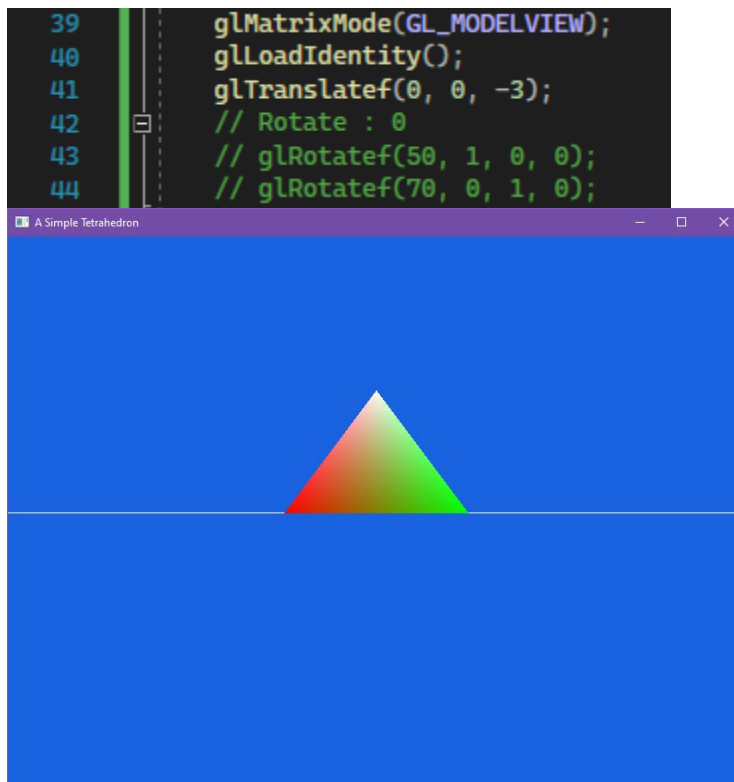
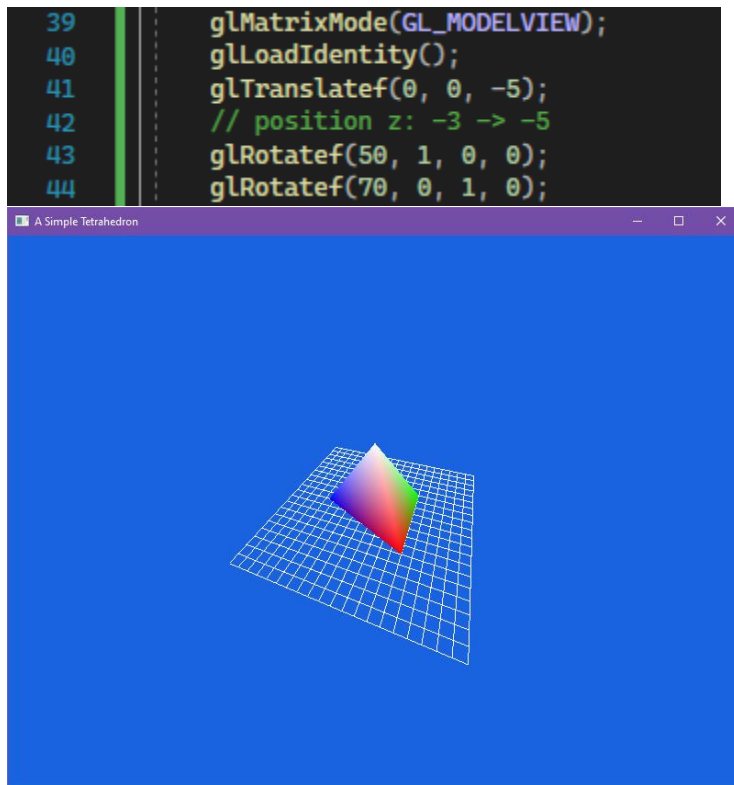
(1)-③. 샘플 프로그램 내 아래 코드에서 Viewing Volume 의 크기를 임의로 조정 해보고, 카메라 렌즈로부터 clipping plane 간의 거리를 가깝거나 더 멀리 조정해서 그 결과를 확인해 보시오.

```
35  glMatrixMode(GL_PROJECTION);  
36  glLoadIdentity();  
37  glFrustum(-4, 4, -1.5, 1.5, 1, 40);  
38  // horizontal bounds : -2 .. 2 -> -4 .. 4
```

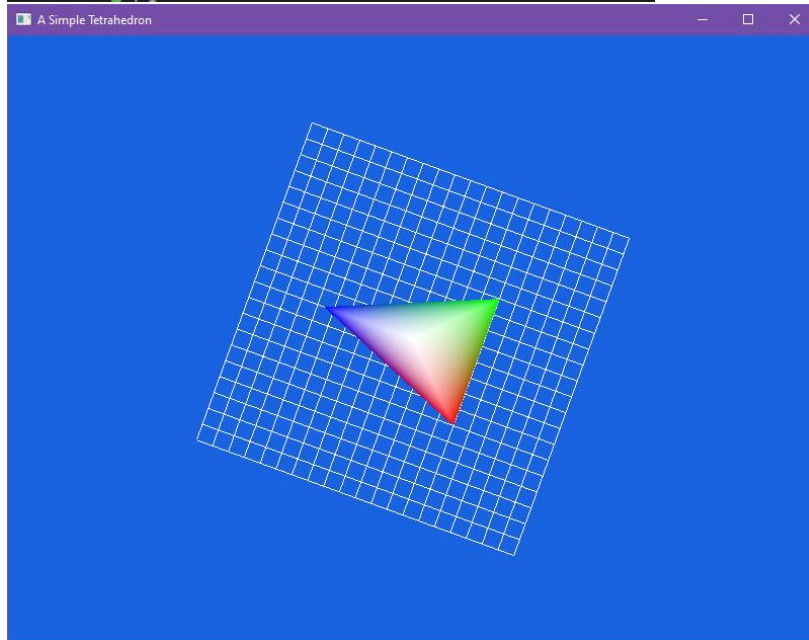




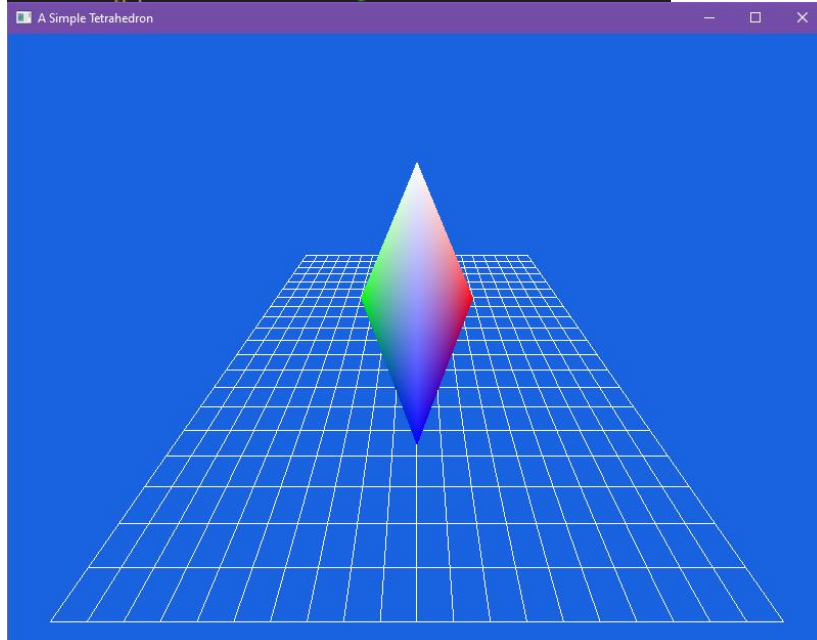
(1)-④. 아래 물체 변경 명령어내의 매개변수 값을 임의로 수정하여 물체의 변환 결과를 보이시오.




```
39  glmMatrixMode(GL_MODELVIEW);
40  glLoadIdentity();
41  glTranslatef(0, 0, -3);
42  glRotatef(90, 1, 0, 0);
43  // Rotate x: 50 -> 90
44  glRotatef(70, 0, 1, 0);
```



```
39  glmMatrixMode(GL_MODELVIEW);
40  glLoadIdentity();
41  glTranslatef(0, 0, -3);
42  glRotatef(50, 1, 0, 0);
43  glRotatef(180, 0, 1, 0);
44  // rotate y: 70 -> 180
```



(2). Spinning Square code 는 마우스(mouse)를 이용하여 4 면체를 Rotating(회전)하는 프로그램입니다. 다음의 조건이 반영된 프로그램으로 변경하여 수행하고, 변경된 프로그램과 결과를 제출하시오.

(2)-①. Square 를 임의의 3 차원 물체(Cube, Sphere 등)로 변경.

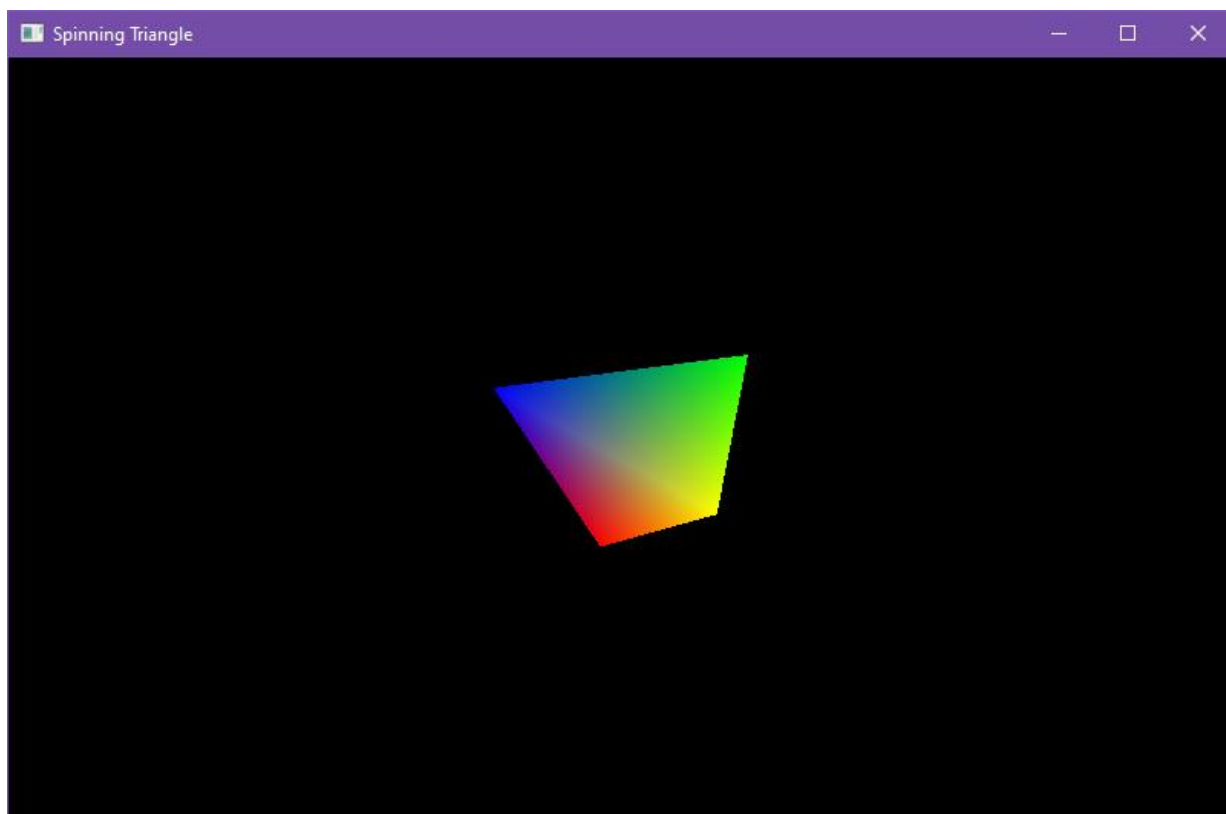
```
P2_01.cpp
P2_01
(Global Scope)

1  #include <GL/glut.h>
2  #include <GL/GLU.h>
3  #include <GL/GL.h>
4
5  static bool spinning = true;
6  static const int FPS = 60;
7  static GLfloat currentAngleOfRotation = 0.0;
8
9  void reshape(GLint w, GLint h) {
10     glViewport(0, 0, w, h);
11     GLfloat aspect = (GLfloat)w / (GLfloat)h;
12     glMatrixMode(GL_PROJECTION);
13     glLoadIdentity();
14     if (w <= h) {
15         glOrtho(-50.0, 50.0, -50.0 / aspect, 50.0 / aspect, -50.0, 50.0);
16     }
17     else {
18         glOrtho(-50.0 * aspect, 50.0 * aspect, -50.0, 50.0, -50.0, 50.0);
19     }
20 }
21
22 void display() {
23     glClear(GL_COLOR_BUFFER_BIT);
24     glMatrixMode(GL_MODELVIEW);
25     glLoadIdentity();
26     glRotatef(currentAngleOfRotation, 0.0, 0.0, 1.0);
27
28     // make tetrahedron
29     glBegin(GL_TRIANGLE_STRIP);
30     glColor3f(1.0f, 0.0f, 0.0f); // color : red
31
32     glVertex3f( 0.0f, 20.0f, 0.0f); // point #01
33     glVertex3f(-15.0f, -10.0f, 10.0f); // point #02
34     glVertex3f( 10.0f, -10.0f, 15.0f); // point #03
35
36     glColor3f(1.0f, 1.0f, 0.0f); // color : yellow
37
38     glVertex3f( 15.0f, 5.0f, 40.0f); // point #04
39
40     glColor3f(0.0f, 0.0f, 1.0f); // color : blue
41
42     glVertex3f(-15.0f, -10.0f, 10.0f); // point #02
43
44     glColor3f(0.0f, 1.0f, 0.0f); // color : green
45
46     glVertex3f( 0.0f, 20.0f, 0.0f); // point #01
47
48     glEnd();
49
50     glFlush();
51     glutSwapBuffers();
52 }
```

```

53
54 void timer(int v) {
55     if (spinning) {
56         currentAngleOfRotation += 1.0;
57         if (currentAngleOfRotation > 360.0) {
58             currentAngleOfRotation -= 360.0;
59         }
60         glutPostRedisplay();
61     }
62     glutTimerFunc(1000 / FPS, timer, v);
63 }
64
65 void mouse(int button, int state, int x, int y) {
66     if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
67         spinning = true;
68     }
69     else if (button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN) {
70         spinning = false;
71     }
72 }
73
74 int main(int argc, char** argv) {
75     glutInit(&argc, argv);
76     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
77     glutInitWindowPosition(80, 80);
78     glutInitWindowSize(800, 500);
79     glutCreateWindow("Spinning Triangle");
80     glutReshapeFunc(reshape);
81     glutDisplayFunc(display);
82     glutTimerFunc(100, timer, 0);
83     glutMouseFunc(mouse);
84     glutMainLoop();
85 }

```



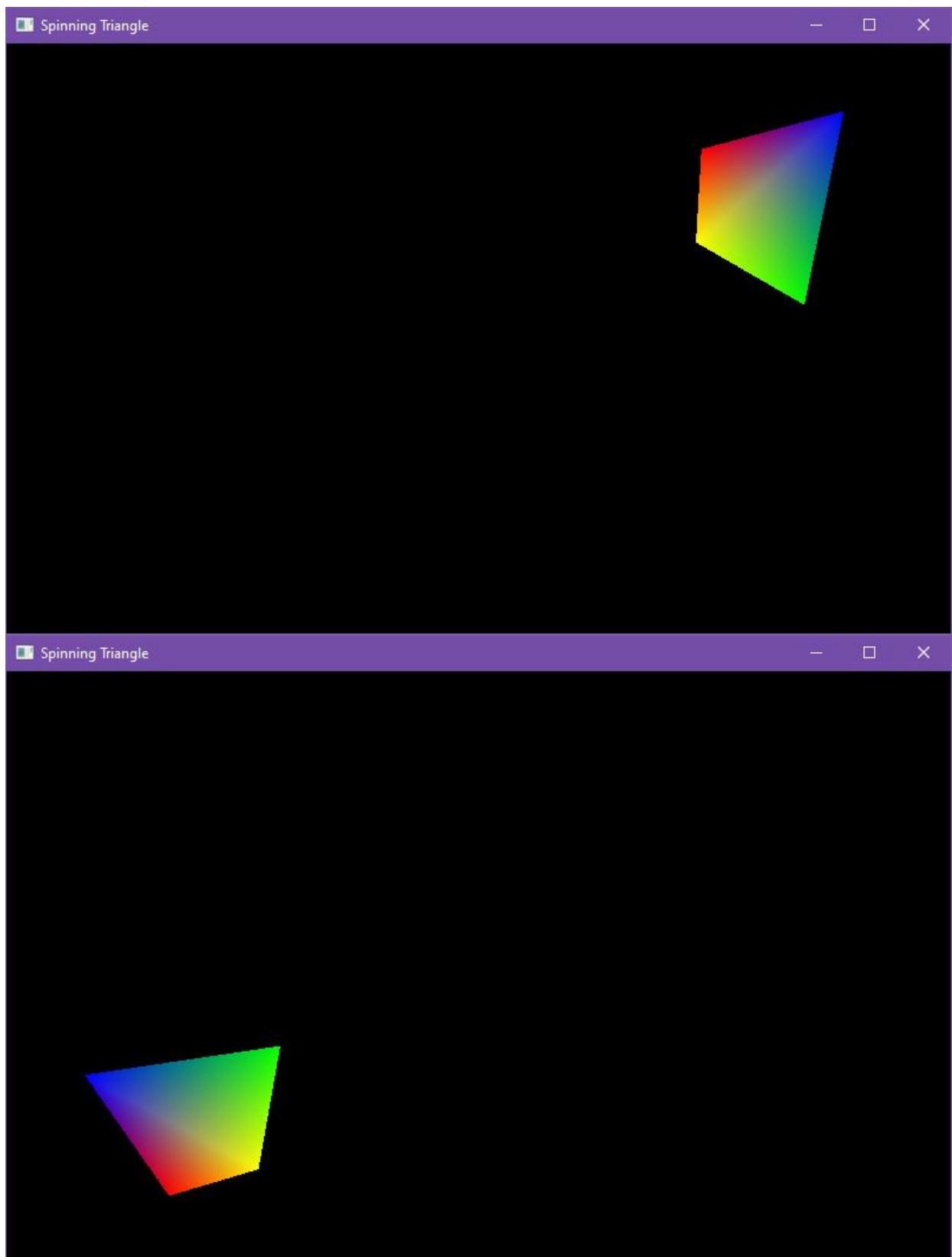
(2)-②. 현재 Mouse 버튼으로 회전 하는 기능외에 버튼을 이용해서 물체를 이동하는 기능 추가 후 결과 확인.

```
P2_01.cpp ×
P2_01 (Global Scope)
1  #include <GL/glut.h>
2  #include <GL/GLU.h>
3  #include <GL/GL.h>
4
5  static bool spinning = true;
6  static const int FPS = 60;
7  static GLfloat currentAngleOfRotation = 0.0;
8
9  // object movement point
10 GLfloat positionX;
11 GLfloat positionY;
12
13 void reshape(GLint w, GLint h) {
14     glViewport(0, 0, w, h);
15     GLfloat aspect = (GLfloat)w / (GLfloat)h;
16     glMatrixMode(GL_PROJECTION);
17     glLoadIdentity();
18     if (w <= h) {
19         glOrtho(-50.0, 50.0, -50.0 / aspect, 50.0 / aspect, -50.0, 50.0);
20     }
21     else {
22         glOrtho(-50.0 * aspect, 50.0 * aspect, -50.0, 50.0, -50.0, 50.0);
23     }
24 }
25
26 void display() {
27     glClear(GL_COLOR_BUFFER_BIT);
28     glMatrixMode(GL_MODELVIEW);
29     glLoadIdentity();
30     // Camera movement to make the object appear to be moving
31     glTranslatef(positionX, positionY, 0.0f);
32     glRotatef(currentAngleOfRotation, 0.0, 0.0, 1.0);
33
34     // make tetrahedron
35     glBegin(GL_TRIANGLE_STRIP);
36     glColor3f(1.0f, 0.0f, 0.0f); // color : red
37
38     glVertex3f(+0.0f, +20.0f, 0.0f); // point #01
39     glVertex3f(-15.0f, -10.0f, 10.0f); // point #02
40     glVertex3f(+10.0f, -10.0f, 15.0f); // point #03
41
42     glColor3f(1.0f, 1.0f, 0.0f); // color : yellow
43
44     glVertex3f(+15.0f, +5.0f, 40.0f); // point #04
45
46     glColor3f(0.0f, 0.0f, 1.0f); // color : blue
47
48     glVertex3f(-15.0f, -10.0f, 10.0f); // point #02
49
50     glColor3f(0.0f, 1.0f, 0.0f); // color : green
51
52     glVertex3f(+0.0f, +20.0f, 0.0f); // point #01
53
54     glEnd();
55
56     glFlush();
57     glutSwapBuffers();
58 }
```

```

60 void timer(int v) {
61     if (spinning) {
62         currentAngleOfRotation += 1.0;
63         if (currentAngleOfRotation > 360.0) {
64             currentAngleOfRotation -= 360.0;
65         }
66         glutPostRedisplay();
67     }
68     glutTimerFunc(1000 / FPS, timer, v);
69 }
70
71 void mouse(int button, int state, int x, int y) {
72     if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
73         spinning = true;
74     }
75     else if (button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN) {
76         spinning = false;
77     }
78 }
79
80 // Key Input
81 void keys(int key, int x, int y) {
82     if (key == GLUT_KEY_RIGHT) {
83         positionX += 5.0f;
84     }
85     if (key == GLUT_KEY_LEFT) {
86         positionX -= 5.0f;
87     }
88     if (key == GLUT_KEY_UP) {
89         positionY += 5.0f;
90     }
91     if (key == GLUT_KEY_DOWN) {
92         positionY -= 5.0f;
93     }
94 }
95
96 int main(int argc, char** argv) {
97     glutInit(&argc, argv);
98     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
99     glutInitWindowPosition(80, 80);
100    glutInitWindowSize(800, 500);
101    glutCreateWindow("Spinning Triangle");
102    glutReshapeFunc(reshape);
103    glutDisplayFunc(display);
104    glutTimerFunc(100, timer, 0);
105    glutMouseFunc(mouse);
106    // Key Input Call
107    glutSpecialFunc(keys);
108    glutMainLoop();
109 }

```



(3). Comet Ride 는 태양 주위를 회전하는 지구의 모습을 animation 한 내용입니다. 다음의 조건이 반영된 프로그램으로 변경하여 수행하고, 변경된 프로그램과 결과를 제출 하시오.

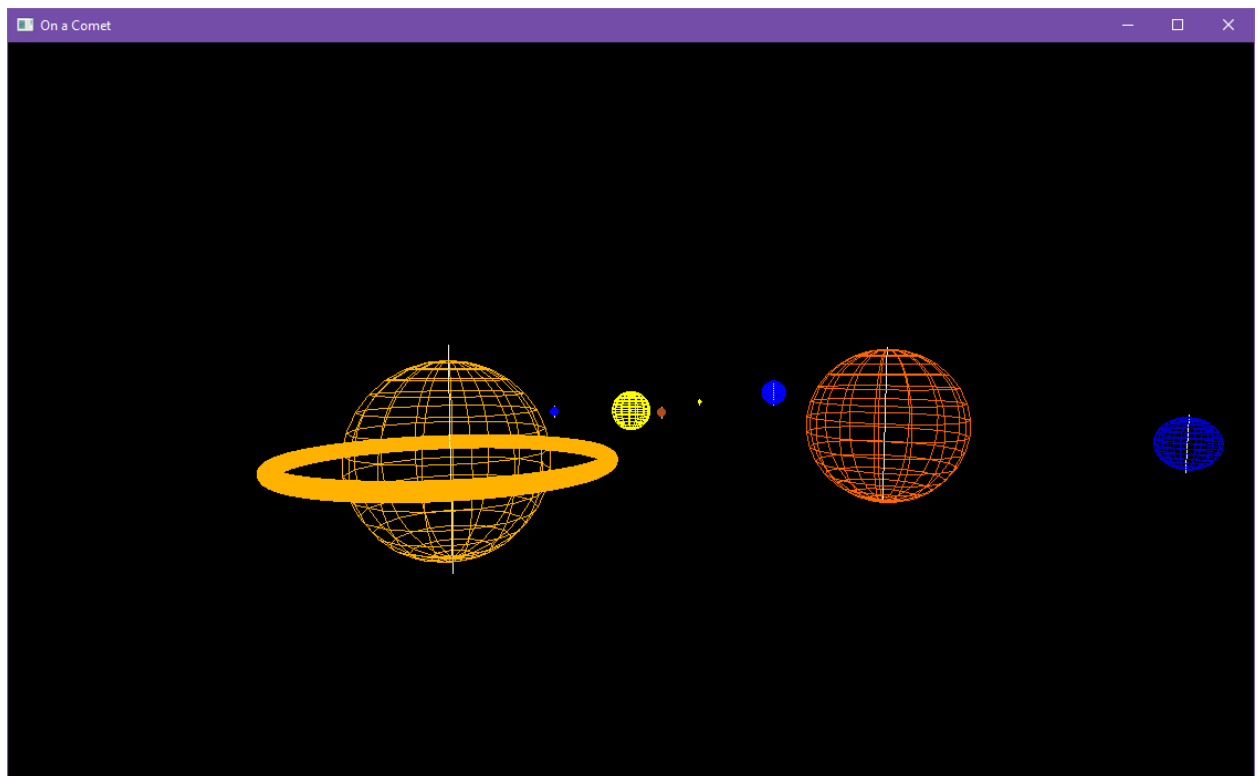
(3)-①. 현재 태양과 그 주변을 도는 지구로 된 위성에 화성(목성) 등 위성을 추가하여 animation 해보세요.
(가능한 태양계 위성을 모두 추가하길 기대합니다만 의무는 아님)

```
P03_01.cpp
P03_01 (Global Scope)
1  #include <GL/glut.h>
2  #include <GL/GL.h>
3  #include <GL/GLU.h>
4  #include <cmath>
5
6  void myWireSphere(GLfloat radius, int slices, int stacks) {
7      glPushMatrix();
8      glRotatef(-90.0, 1.0, 0.0, 0.0);
9      glutWireSphere(radius, slices, stacks);
10     glPopMatrix();
11 }
12
13 static int year = 0, day = 0;
14
15 void drawPlanet(float distance, float size, int speed) {
16     glPushMatrix();
17     glRotatef((GLfloat)year * speed, 0.0, 1.0, 0.0);
18     glTranslatef(distance, 0.0, 0.0);
19     glRotatef((GLfloat)day, 0.0, 1.0, 0.0);
20     myWireSphere(size, 15, 15);
21     glColor3f(1, 1, 1);
22     glBegin(GL_LINES);
23     glVertex3f(0, -(size + 0.1), 0);
24     glVertex3f(0, size + 0.1, 0);
25     glEnd();
26     glPopMatrix();
27 }
28
29 void display() {
30     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
31     glPushMatrix();
32
33     // Draw the Sun
34     glColor3f(1.0, 1.0, 0.0);
35     myWireSphere(1.0, 15, 15);
36
37     // Draw planet : Mercury
38     glColor3f(0.2, 0.2, 0.2);
39     drawPlanet(1.5, 0.07, 4.2);
40
41     // Draw planet : Venus
42     glColor3f(0.7, 0.3, 0.1);
43     drawPlanet(2.0, 0.2, 3.0);
44
45     // Draw planet : Earth
46     glColor3f(0, 0, 1);
47     drawPlanet(4.0, 0.2, 2.7);
48
49     // Draw planet : Mars
50     glColor3f(1.0, 1.0, 0.0);
51     drawPlanet(9.0, 0.1, 1.8);
52
53     // Draw planet : Jupiter
54     glColor3f(6.0, 0.4, 0.0);
55     drawPlanet(13.0, 3.0, 3.8);
56 }
```

```

57 // Draw planet : Saturn
58 glColor3f(1.0, 0.7, 0.0);
59 glPushMatrix();
60 glRotatef((GLfloat)year * 2.5, 0.0, 1.0, 0.0);
61 glTranslatef(18, 0.0, 0.0);
62 glRotatef((GLfloat)day, 0.0, 1.0, 0.0);
63 myWireSphere(2.5, 15, 15);
64 glColor3f(1, 1, 1);
65 glBegin(GL_LINES);
66 glVertex3f(0, -(2.8 + 0.1), 0);
67 glVertex3f(0, 2.8 + 0.1, 0);
68 glEnd();
69 glRotatef(90, 1.0, 0.0, 0.0);
70 glColor3f(1.0, 0.7, 0.0);
71 glutWireTorus(0.2, 4, 60, 60);
72 glPopMatrix();
73
74 // Draw planet : Uranus
75 glColor3f(0.0, 0.0, 0.8);
76 drawPlanet(22.0, 0.8, 3.0);
77
78 // Draw planet : Neptune
79 glColor3f(0.0, 0.0, 1.0);
80 drawPlanet(25.0, 1.0, 1);
81
82 glPopMatrix();
83 glFlush();
84 glutSwapBuffers();
85 }
86
87 static GLfloat u = 0.0;
88 static GLfloat du = 0.1;
89
90 void timer(int v) {
91     u += du;
92     day = (day + 1) % 360;
93     year = (year + 2) % 360;
94     glLoadIdentity();
95     gluLookAt(20 * cos(u / 8.0) + 12, 5 * sin(u / 8.0) + 1, 10 * cos(u / 8.0) + 2, 0, 0, 0, 0, 1, 0);
96     glutPostRedisplay();
97     glutTimerFunc(1000 / 60, timer, v);
98 }
99
100 void reshape(GLint w, GLint h) {
101     glViewport(0, 0, w, h);
102     glMatrixMode(GL_PROJECTION);
103     glLoadIdentity();
104     gluPerspective(60.0, (GLfloat)w / (GLfloat)h, 1.0, 80.0);
105     glMatrixMode(GL_MODELVIEW);
106 }
107
108 int main(int argc, char** argv) {
109     glutInit(&argc, argv);
110     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
111     glutInitWindowSize(800, 600);
112     glutCreateWindow("On a Comet");
113     glutDisplayFunc(display);
114     glutReshapeFunc(reshape);
115     glutTimerFunc(100, timer, 0);
116     glEnable(GL_DEPTH_TEST);
117     glutMainLoop();
118 }

```

(3)-②. 현재 Meshed model 인 태양과 지구 그리고 추가되는 화성 목성을 모두 solid model 로 변경해서 실행.

```
5  
6 void myWireSphere(GLfloat radius, int slices, int stacks) {  
7     glPushMatrix();  
8     glRotatef(-90.0, 1.0, 0.0, 0.0);  
9     glutSolidSphere(radius, slices, stacks);  
10    glPopMatrix();  
11 }
```

