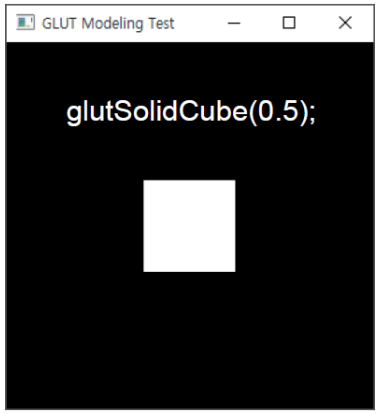
9/24(금) 실습내용(1) : GLUT Modeling(1)

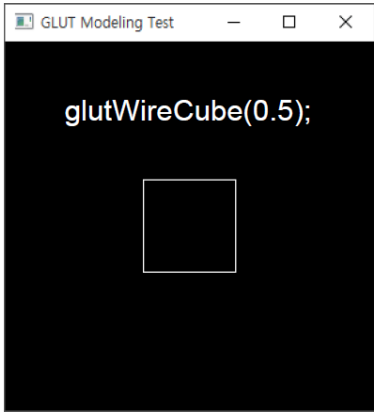
- 아래 코드를 참조하여 Cube, Sphere, Torus, Cone, Tetrahedron, Icosahederon, Teapot 그려보기

```
#include <gl/glut.h>
void MyDisplay(){
        물체별 코드 입력

   glFlush();
}

int main(){
        glutCreateWindow("GLUT Modeling Test");
        glutDisplayFunc(MyDisplay);
        glutMainLoop();

       return 0;
}
```

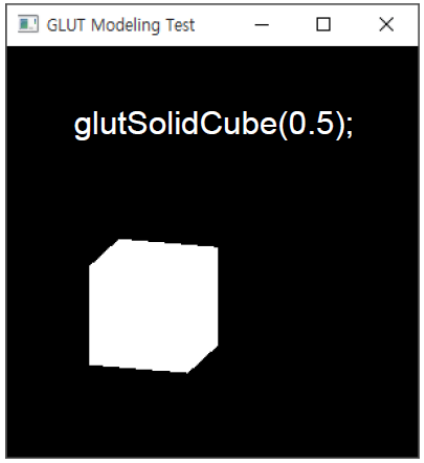
9/24(금) 실습내용(1) : GLUT Modeling(2)

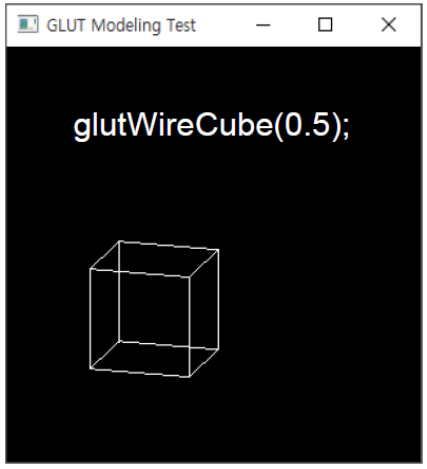
glutSolidCube(Gldouble size); **glutWireCube(Gldouble size);**



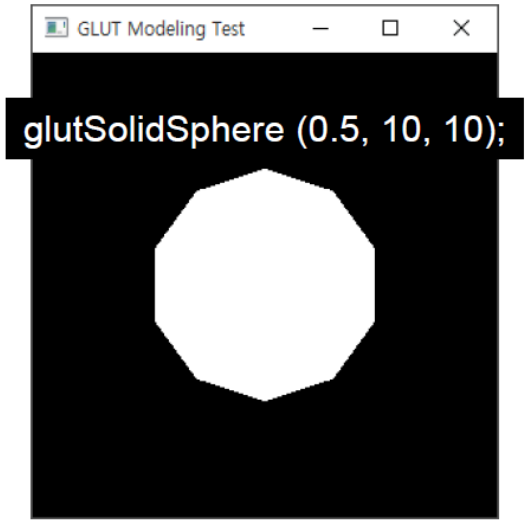


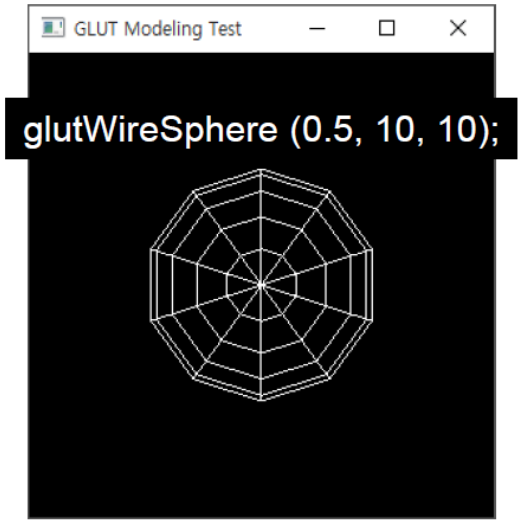

```
void MyDisplay(){
      gluLookAt(0.3, 0.3, 0.0, 0.0, 0.0, -1.0, 0.0, 1.0, 0.0);
      glutSolidCube(0.5);
      glFlush();
}
```

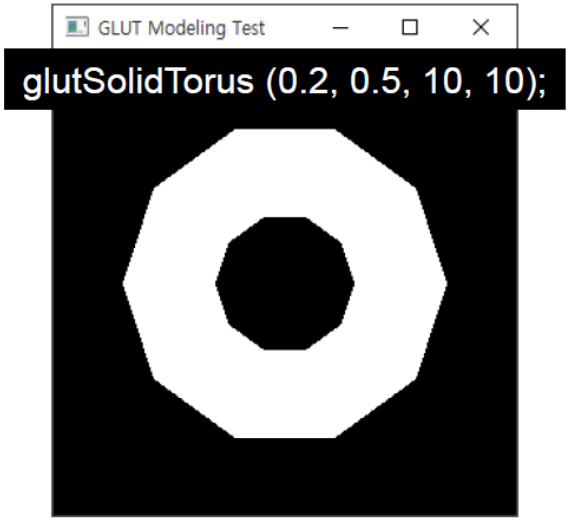


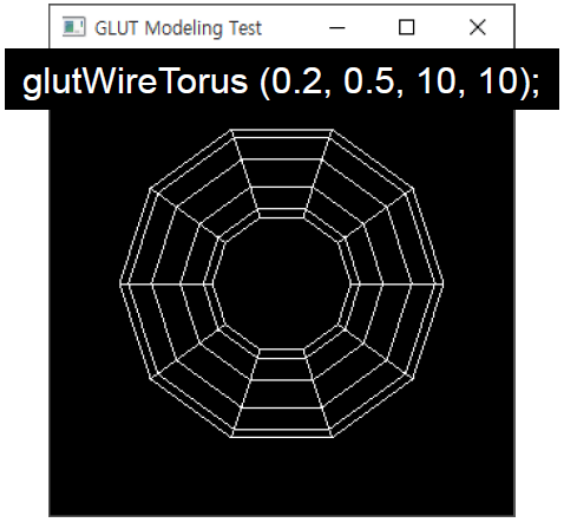


9/24(금) 실습내용(1) : GLUT Modeling(3)

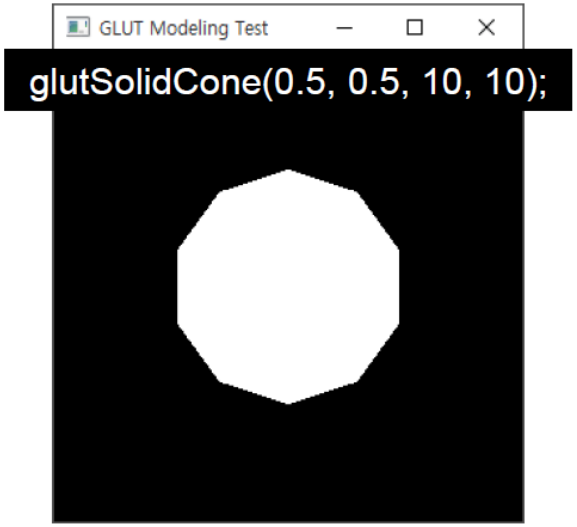


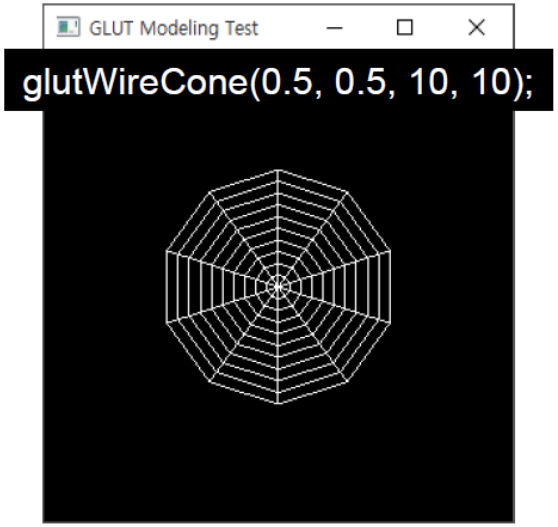


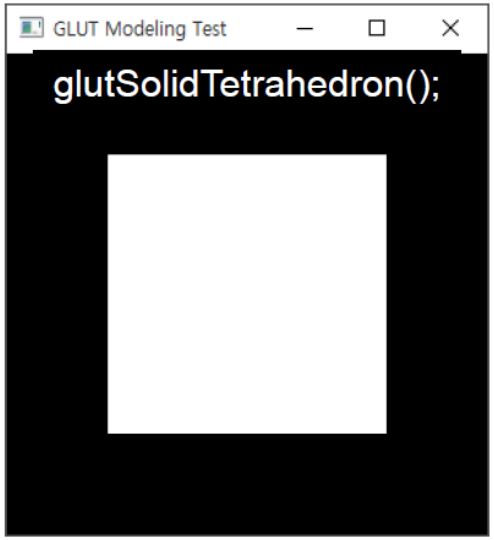


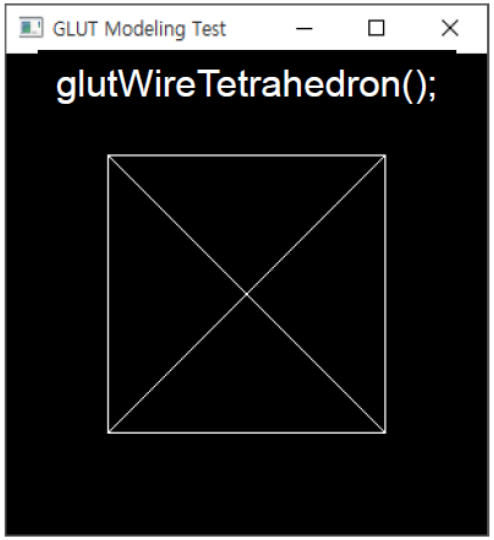


9/24(금) 실습내용(1) : GLUT Modeling(4)

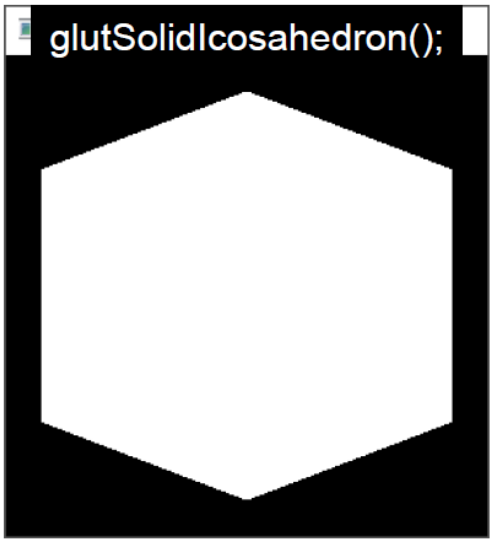


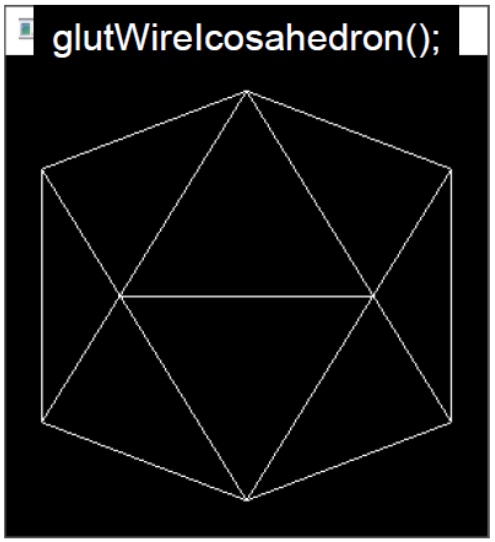


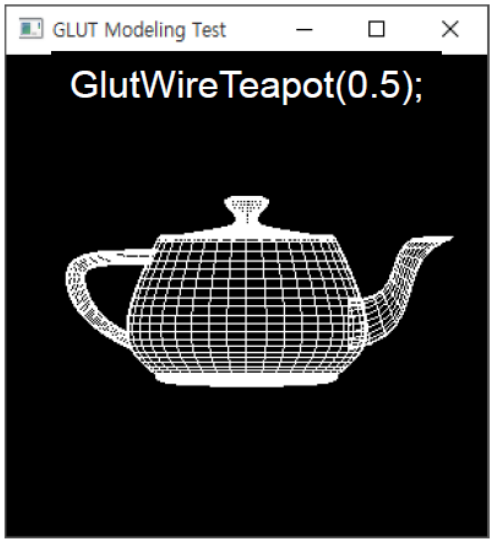




9/24(금) 실습내용(1) : GLUT Modeling(5)





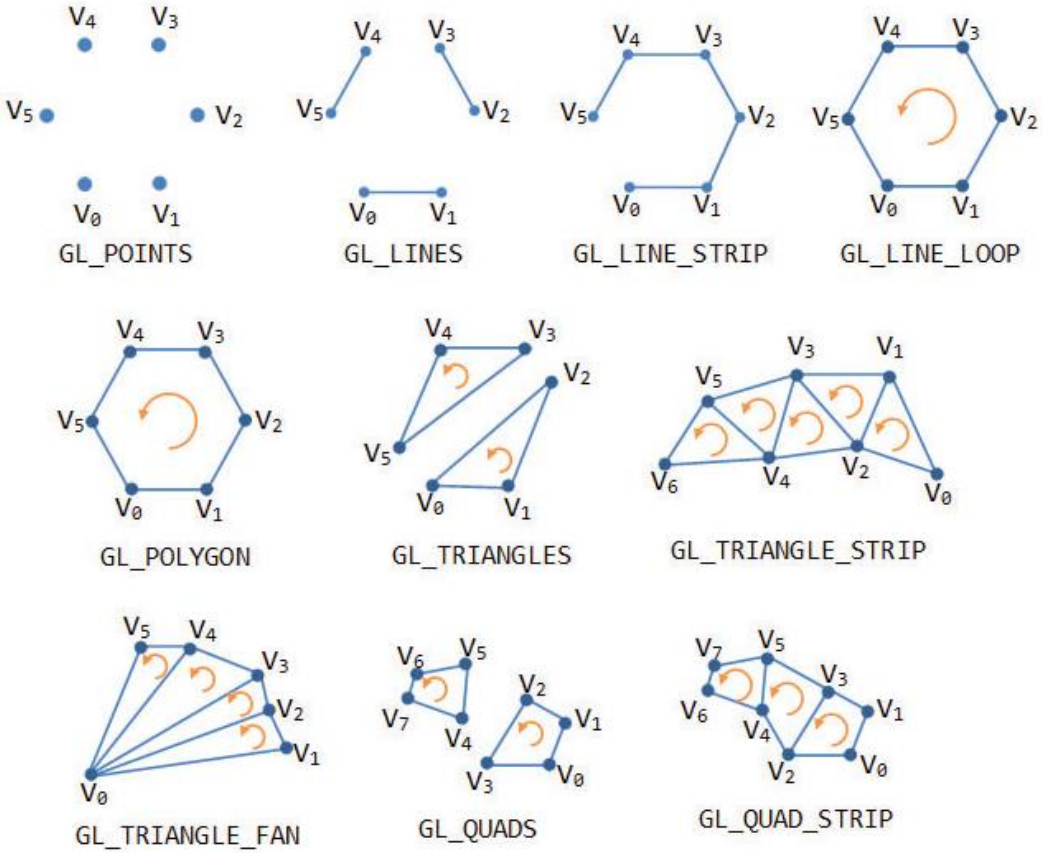


9/24(금) 실습내용(2) : GL Modeling – 2D Primitives 구현(1)

- [실습과제2-1] 코드4-2를 참조하여 다음과 같은 2D Primitive들을 구현
  - Points, Lines, LineStrip, LineLoop
  - TrangleFan, Triangles, TriangleStrip
  - Polygons, Quads, QuadStrip

- [제한사항]
  - z 값은 모두 0으로 지정하고, x축과 y축 값만 고려

- [Due Date]
  - 9/30(목) 23:59

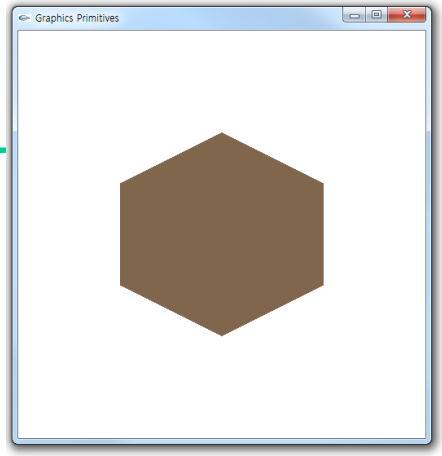


9/24(금) 실습내용(2) : GL Modeling – 2D Primitives 구현(2)

다각형(Polygon)으로 생성한 Polygon

코드 4-2

```
#include <GL/glut.h>
void MyDisplay( ) {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.5, 0.4, 0.3);
    glBegin(GL_POLYGON);
        glVertex3f( 0.0, 1.0, 0.0);     glVertex3f(-1.0, 0.5, 0.0);
        glVertex3f(-1.0, -0.5, 0.0);    glVertex3f( 0.0, -1.0, 0.0);
        glVertex3f( 1.0, -0.5, 0.0);    glVertex3f( 1.0, 0.5, 0.0);
    glEnd();
glFlush();
}
…
```



9/24(금) 실습내용(3) : GLU Modeling – 3D Primitives 구현(1)

- [실습과제2-2] 코드4-5와 코드4-7을 참고하여 다음과 같은 3D Primitive 구현 : Sphere, Cylinder, Disk, PartialDisk

- [제한사항]
  - Display List를 사용한다.
  - Solid, Wireframe 등 DrawSyle을 여러가지로 해본다.

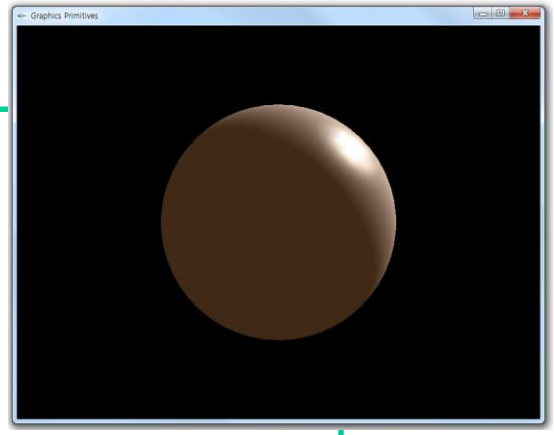
```
Gluint list[4];
list[0] = glGenLists(4);   // Sphere
list[1] = list[0] + 1;     // Cylinder
…                          // Disk
…                          // Partial Disk
```

```
gluQuadricDrawStyle(qobj, GLU_FILL);
gluQuadricDrawStyle(qobj, GLU_LINE);
gluQuadricDrawStyle(qobj, GLU_SILHOUETTE);
gluQuadricDrawStyle(qobj, GLU_POINT);
```

- [Due Date]
  - 9/30(목) 23:59

9/24(금) 실습내용(3) : GLU Modeling – 3D Primitives 구현(2)

Quadrics로 생성한 Sphere



코드 4-5

```
GLUquadricObj *qobj = gluNewQuadric(); // 새로운 Quadric 생성
..
void MyDisplay( ) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glMatrixMode( GL_MODELVIEW );
    glLoadIdentity();

    glEnable(GL_LIGHTING);
    glShadeModel (GL_SMOOTH);

    gluQuadricDrawStyle( qobj, GLU_FILL );  // Quadric을 위해 원하는 Draw Style 지정
    gluQuadricNormals( qobj, GLU_SMOOTH );  // 법선벡터 제어
    gluQuadricOrientation( qobj, GLU_OUTSIDE ); // 법선벡터 내부 및 외부 등과 같은 방향 지정
    gluQuadricTexture( qobj, GL_FALSE ); // Texture 좌표 사용할 것인지에 대한 여부

    gluSphere( qobj, 1.5, 50, 50 );  // Sphere 생성

    glutSwapBuffers();
}
...
```

9/24(금) 실습내용(3) : GLU Modeling – 3D Primitives 구현(3)

DisplayList의 사용#

코드 4-7

```
#include <GL/glut.h>
#include <gl.h>
#include <glu.h>

int MyListID; // DisplayList는 정수 ID에 의해 식별
// Display List 생성
void MyCreateList() {
    MyListID = glGenLists(1);   // DisplayList 선언
  // 아이디를 가진 리스트 1개를 새로 만들되,
  // 실행하지 않고 컴파일 완료된 버전을 만든다.
   glNewList(MyListID, GL_COMPILE);

   glBegin(GL_POLYGON);   // 리스트를 구성하는 함수 나열
      glColor3f(0.5, 0.5, 0.5);
      glVertex3f(-0.5, -0.5, 0.0);
      glVertex3f(0.5, -0.5, 0.0);
      glVertex3f(0.5, 0.5, 0.0);
      glVertex3f(-0.5, 0.5, 0.0);
   glEnd( );
   glEndList();
}
```

```
void MyDisplay() {
   glClear(GL_COLOR_BUFFER_BIT);
   glViewport(0, 0, 300, 300);
// 컴파일이 완료된 리스트가 실제로 실행
   glCallList(MyListID);
   glFlush();
}

int main(int argc, char** argv) {
   …
   glutDisplayFunc(MyDisplay);
   MyCreateList(); // Display List 생성
   glutMainLoop();
   return 0;
}
```