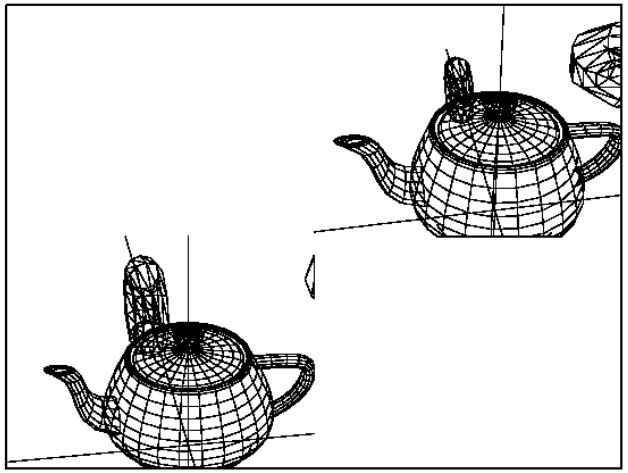
11/05 (금) 실습내용(1) : 뷰폿변환 이해(1)

- [실습과제6] 코드7-2를 참조하여 다음과 같은 결과를 낼 수 있는 프로그램 구현
  - 여기서 보여지는 결과는 한 예일 뿐 학생들 스스로 정의해서 어떤 물체를 그려도 상관 없음. 단 2개 이상의 물체를 한 뷰폿에 그려야 함.
  - 하나의 윈도우를 4개의 뷰폿으로 분할(윈도우 및 뷰폿의 각각의 크기는 필요에 따라 변형하여 설정 가능)
  - 우상단의 뷰폿 : 원근투영 – gluPerspective()
  - 좌하단의 뷰폿 : 평행투영 – glOrtho()
  - 뷰폿에 있는 물체들은 각각 Keyboard Callback 또는 Mouse Callback 함수를 사용하여 Camera의 이동(시점변환)을 로 변환이 수행되도록 함)



  - 모든 뷰폿들은 객체를 기본적으로 뷰폿의 중심부에 위치하도록 하고. Keyboard나 Mouse로 제어할 경우에도 항상 중심부에 위치하도록 함.
  - 윈도우의 크기를 임의로 변경하거나 Full Screen으로 변경하였을 경우에는 윈도우의 크기에 상관없이 객체의 형태가 왜곡되지 않도록 함.

- [Due Date]
  - 11/11(목) 23:59

11/05 (금) 실습내용(1) : 뷰폿변환 이해(2)

코드 7-2

```
#include <GL/glut.h>
…
void MyDisplay(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);

    glViewport(0, 0, Width/2, Height/2);
    glPushMatrix();
        gluLookAt(0.0, 0.0, 1.0,   0.0, 0.0, 0.0,   0.0, 1.0, 0.0);
        DrawScene();
    glPopMatrix();

    glViewport(Width/2, 0, Width/2, Height/2);
    glPushMatrix();
        gluLookAt(1.0, 0.0, 0.0, 0.0, 0.0, 0.0,  0.0, 1.0, 0.0);
        DrawScene();
    glPopMatrix();

    glViewport(0, Height/2, Width/2, Height/2);
    glPushMatrix();
        gluLookAt(0.0, 1.0, 0.0,   0.0, 0.0, 0.0,   0.0, 0.0, -1.0);
        DrawScene();
    glPopMatrix();
```

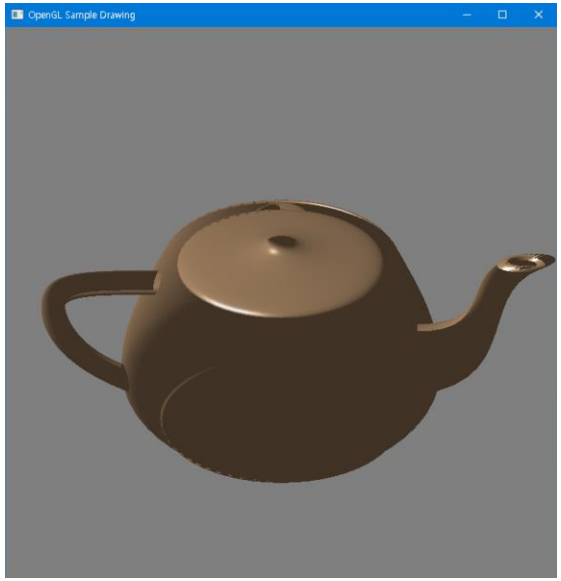
```
    glViewport(Width/2, Height/2, Width/2, Height/2);
    glMatrixMode(GL_PROJECTION);
    glPushMatrix();
        glLoadIdentity();
        gluPerspective(30, 1.0, 3.0, 50.0);

        glMatrixMode(GL_MODELVIEW);
        glPushMatrix();
            gluLookAt(5.0, 5.0, 5.0,  0.0, 0.0, 0.0,  0.0, 1.0, 0.0);
            DrawScene();
        glPopMatrix();

        glMatrixMode(GL_PROJECTION);
    glPopMatrix();

   glFlush();
}
…
```

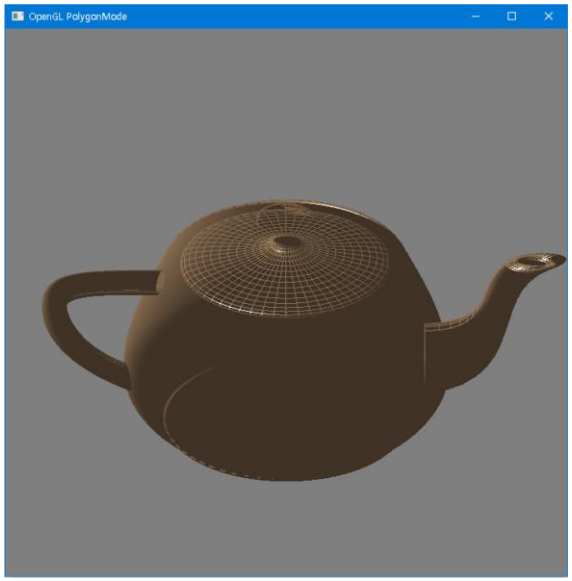
11/05 (금) 실습내용(2) : 후면 및 은면제거

- 코드8-2의 파라미터들을 바꿔보며 후면 및 은면 제거 이해하기

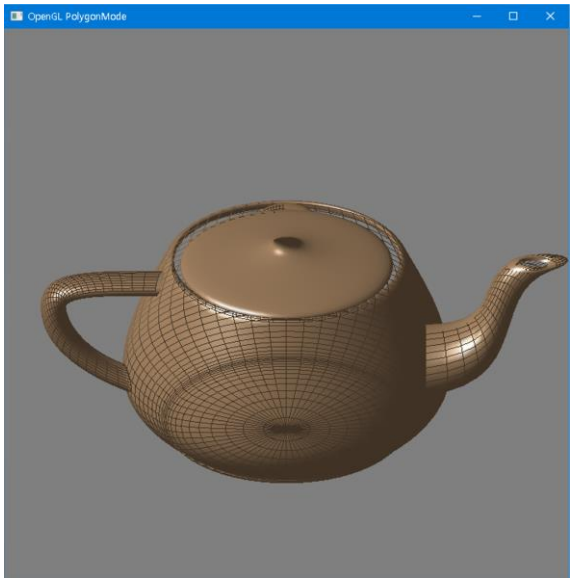
1) 후면 및 은면을 모두 제거하지 않았을 경우

glutSolidTeapot(0.58);





뒷면:FILL 앞면:LINE



뒷면:LINE 앞면:FILL

11/05 (금) 실습내용(2) : 후면 및 은면제거(2)

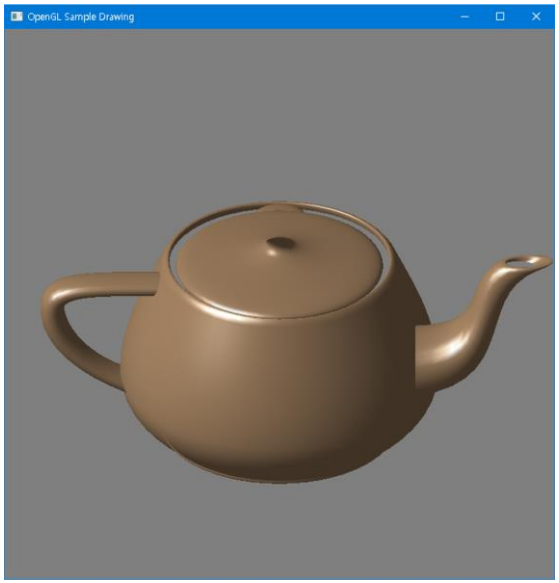
코드 8-2

```
#include <GL/glut.h>
…
void InitVisibility() {
    glEnable(GL_CULL_FACE);
    glFrontFace(GL_CW);
    glPolygonMode(GL_FRONT, GL_LINE);
    glPolygonMode(GL_BACK, GL_FILL);
    glCullFace(GL_FRONT);
    glEnable(GL_DEPTH_TEST);
}

void MyDisplay() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0.0, 0.4, 0.5, 0.0, -0.5, -1.0, 0.0, 1.0, 0.0);
    glutSolidTeapot(0.58);
    glFlush();
}
int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA | GLUT_DEPTH);
    …
    InitVisibility();
   …
    return 0;
}
```

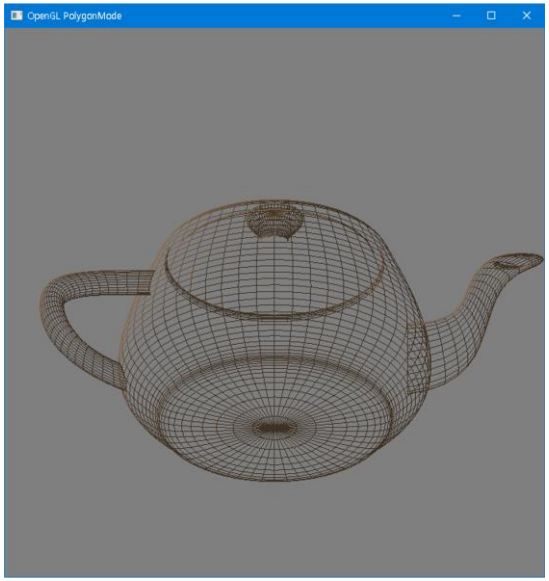
11/05 (금) 실습내용(2) : 후면 및 은면제거(3)

2) 후면 제거만 활성화했을 경우

glutSolidTeapot(0.58);

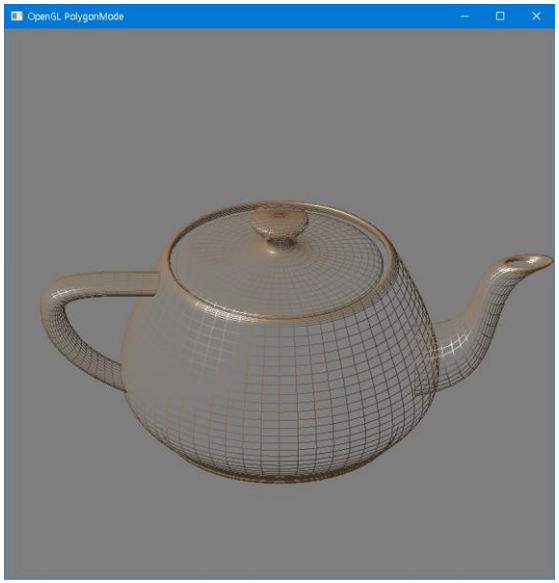


주전자 바닥면은 보이지 않음.
(후면 제거됨. 앞면만 보임)
은면제거가 되지 않아, 뚜껑 손잡이에 가려 보이지 않아야 할, z좌표상 뒤에 있는 뚜껑 덮개가 앞에 있어 뚜껑손잡이가 일부 보이지 않음.



앞면 제거
: 뒷면이 LINE으로 보임.

뒷면:LINE 앞면:FILL



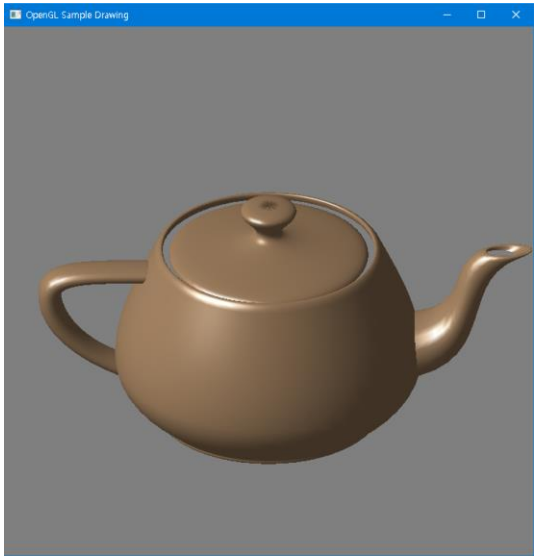
뒷면 제거
: 앞면이 LINE으로 보임.

뒷면:FILL 앞면:LINE

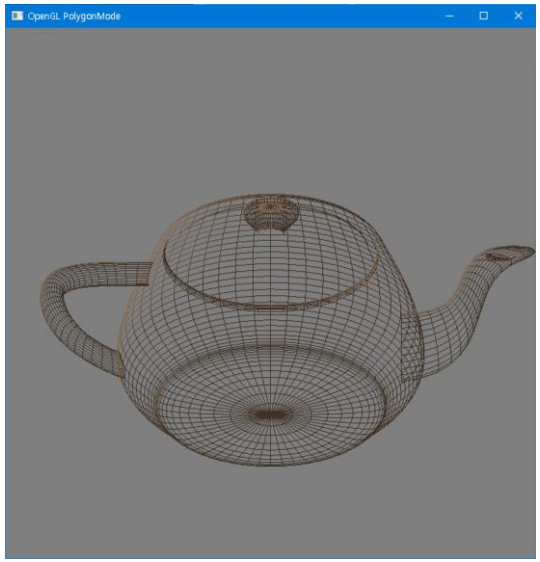
11/05 (금) 실습내용(2) : 후면 및 은면제거(4)

3) 후면 제거, 은면 제거 모두 활성화했을 경우

glutSolidTeapot(0.58);

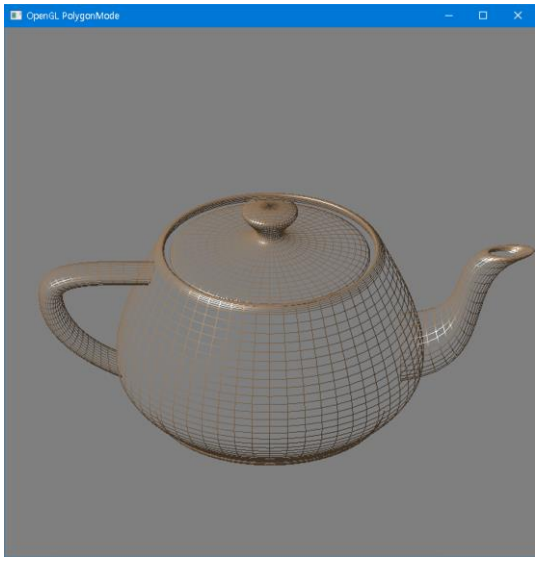


앞면만 보임.
주전자 손잡이가
뚜껑 윗부분보다 앞에 있으므로
은면제거 후, 잘 보임.



앞면 제거, 은면 제거
: 뒷면이 LINE으로 보임.

뒷면:LINE 앞면:FILL



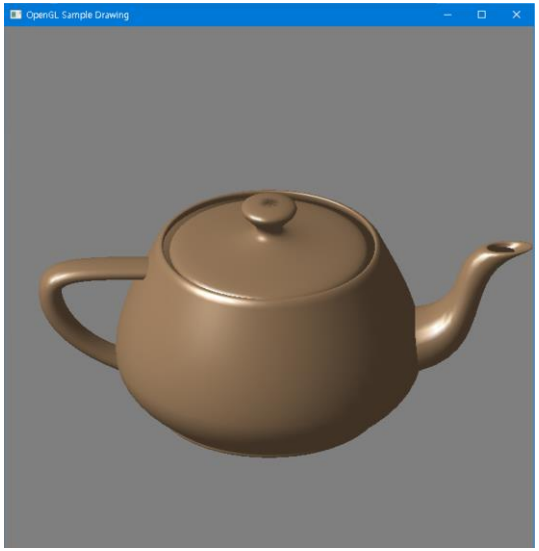
뒷면 제거, 은면 제거
: 앞면이 LINE으로 보임.

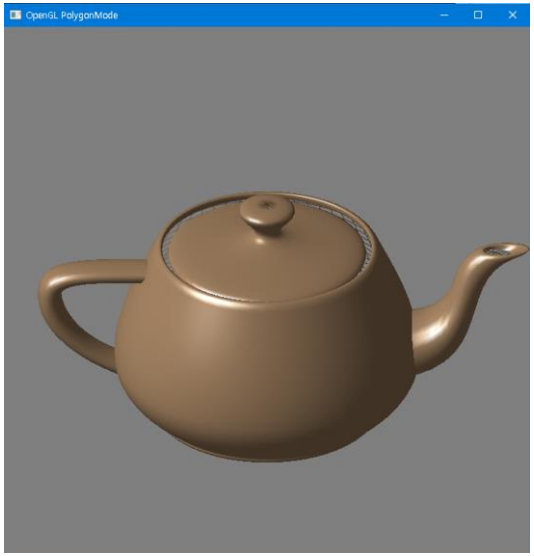
뒷면:FILL 앞면:LINE

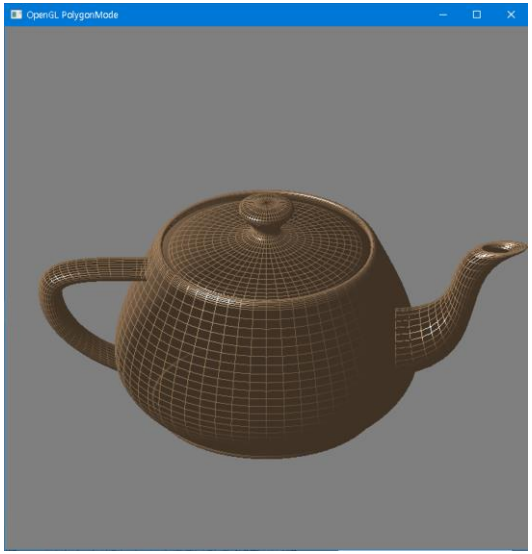
11/05 (금) 실습내용(2) : 후면 및 은면제거(5)

4) 은면 제거만 활성화했을 경우

glutSolidTeapot(0.58);







앞면, 뒷면이 모두 보임.
주전자 뒤쪽의 뒷면(이면) 보임.

뒷면:LINE 앞면:FILL

뒷면:FILL 앞면:LINE