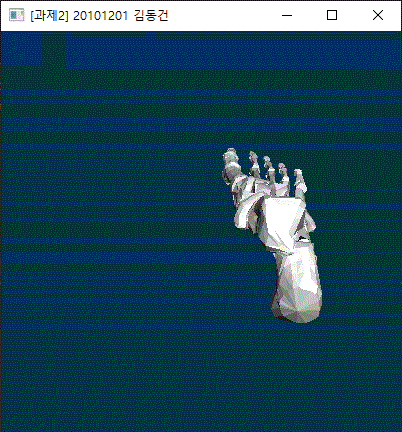
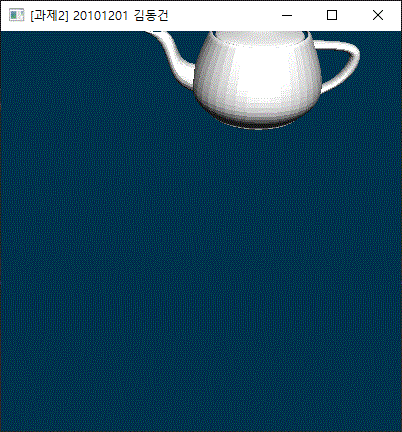
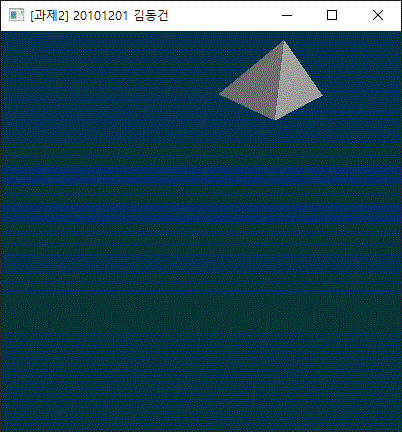
Computer Assignment 3

20101201 김동건

* 실행화면(bones.obj, teapot.obj, tetrahedron.obj)

 (\*타이틀 바에 과제2라고 적혀있는데 나중에 과제 3으로 바꿨습니다)

모두 오브젝트 기준 y축으로 회전하는 동시에 월드 y축 기준으로도 회전한다. 캡처본이라서 정지해 있지만, 실행해보면 잘 돌아갑니다.

* 주요 코드(나머지 대부분의 코드는 과제2에서와 똑같습니다.)

void transform()

{

// Model matrix : an identity matrix (model will be at the origin)

Model = glm::mat4(1.0f);

Model = glm::rotate(Model, angle, glm::vec3(0.0, 1.0, 0)) \*

glm::translate(Model, glm::vec3(angle2, 0.0f, 0.0f)) \* glm::rotate(Model, angle, glm::vec3(0.0, 1.0, 0));

// ModelViewProjection

mvp = Projection \* View \* Model;

glUniformMatrix4fv(MatrixID, 1, GL\_FALSE, &mvp[0][0]);

glUniformMatrix4fv(ViewMatrixID, 1, GL\_FALSE, &View[0][0]);

glUniformMatrix4fv(ModelMatrixID, 1, GL\_FALSE, &Model[0][0]);

glm::vec3 lightPos = glm::vec3(10, 20, 10);

glUniform3f(LightID, lightPos.x, lightPos.y, lightPos.z);

}

단순히 회전이 아니라 모델 좌표계와 월드 좌표계의 y축을 모두 회전해야되기 때문에 Model에 변화를 주었다. 동시에 자전과 공전을 같이 해야되기 때문에 rotate \* translate \* rotate를 해주어 자전과 동시에 공전이 가능하게 하였다. translate에서는 vec3(angle2, 0.0f, 0.0f)로 점차 증가하는 angle2를 받아 시점에서 점점 멀어지도록 하였다. 점진적으로 멀어지기 때문에 공전은 나선형을 그리면서 공전한다.

void timer(int value) {

angle += glm::radians(5.0f);

angle2 += glm::radians(1.5f);

glutPostRedisplay();

glutTimerFunc(20, timer, 0);

}

20msec마다 일정각도로 회전해야 되기 때문에 glutTImerFunc에 20을 주어 주기를 20msec로 하였다. angle변수를 모두 사용하려 했지만, 5도씩 멀어지게 하면 너무 빨리 화면에서 사라져서 angle2라는 변수를 새로 만들고 1.5f만큼 증가하도록 하였다.