

Graphics HW4 Report

컴퓨터공학부 2013-11425 이창영

1. 사용 언어

Python 3.6

2. 사용한 라이브러리

1) 설치 라이브러리

PyOpenGL – pip install PyOpenGL로 설치할 수 있다.

2) 기본 라이브러리

json - 데이터 관리 용이함을 위해 사용

math - cos, sin sqrt 등의 수학 연산을 위해 사용

sys - 실행 쿼리에 인자를 받기 위해 사용

3) 만든 라이브러리

xyz.py - 점, 벡터, 그들의 연산을 정의함.

quaternion.py - 쿼터니언과 그것의 연산을 정의함.

cube.py - 정육면체를 나타내는 객체.

3. 실행 방법

python hw4.py

4. 구현한 것

- 1) 이전 과제에서 구현한 surface 그리기
- 2) 1)과 겹치는 반투명한 cube 그리기
- 3) 2)를 그릴 때 법선의 각도를 이용하여 depth ordering하여 그리기
- 4) 광원 생성
- 5) 1)과 겹치는 material을 입힌 cube 그리기

5. 구현하지 못한 것

- 1) BSP tree를 이용한 depth ordering

6. 조작법

1) virtual trackball rotate

마우스 왼쪽 버튼을 클릭하여 드래그 하면 virtual trackball 방식으로 rotate하는 모습을 볼 수 있다.

2) translate screen

키보드 t버튼을 누른 상태로 유지하며 마우스를 드래그 하면 마우스 방향으로 전체가 translate된다.

3) zoom in/out

키보드 z버튼을 누른 상태로 방향키 위 버튼을 누르면 zoom/in 되고, 방향키 아래 버튼을 누르면 zoom out 된다.

4) dolly in/out

키보드 d버튼을 누른 상태로 방향키 위 버튼을 누르면 dolly in 되고, 방향키 아래 버튼을 누르면 dolly out 된다.

7. 구현 내용, 방법

1) sweep polygonal surface 그리기

저번 과제에서 진행했던 것 그대로 myinput.txt를 읽어 고려청자 모양의 surface를 render하였다.

2) 반투명한 cube 그리기 (법선을 이용한 depth ordering 사용)

한쪽 면이 sweep polygonal surface와 겹치도록 반투명한 cube를 만들었다.

glColor4f를 이용하였고 마지막 인자는 0.5로 하였다.

이 때 여러 polygon들을 render하는 순서가 중요했다. 일단 불투명한 물체들을 가장 먼저 render하였고 그 다음 cube 여섯개 반투명한 polygon은 depth ordering 하여 순서를 정하여 render하였다. 각각의 면은 빨(1, 0, 0, 0.5), 주(1, 153/255, 0, 0.5), 노(1, 1, 0, 0.5), 초(0, 1, 0, 0.5), 파(0, 0, 1, 0.5), 보(153/255, 0, 153/255, 0.5) 색을 가지고 있다.

depth ordering하는 방법은 법선 벡터와 카메라 벡터 사이의 각도를 이용하였다. 처음에는 무조건 눈에 3면이 앞쪽 나머지 3면이 뒤쪽이라고 생각하여 그 각도가 90도보다 큰 3면을 먼저 무작위로 render하고 나머지 3면을 나중에 render하면 될 것이라고 생각하였으나 구현하고 보니 시야 각도 때문에 무조건 3개의 면이 앞에 보이는것은 아니여서 잘 작동하지 않았다. 그래서 각도가 가장 큰 것부터 차례대로 render하니 잘 작동하였다.

3) material을 입힌 cube 및 광원 생성

이것 또한 불투명한 물체이기 때문에 반투명한 cube를 그리기 전에 먼저 render하였다. material이 입혀진 모습을 보기 위해선 우선 광원이 필요하다. opengl의 glLight를 이용하여 광원을 6개

생성하였다. (0, 0, 100), (0, 0, -100), (0, 100, 0), (0, -100, 0), (100, 0, 0), (-100, 0, 0)에 생성하였다. 광원에는 ambient, diffuse, specular 등의 값을 적용할 수 있다. ambient light는 특정 방향을 가지지 않고 주변을 덮고 있는 주변광이다. diffuse light는 모든 방향으로 균일하게 반사하는 분산광이다. specular light는 특정 방향으로 빛을 반사하는 빛이다. shininess 값은 매끈매끈함의 정도이다. x축, z축에 위치한 4개의 광원에는 ambient값이 (0, 0, 0)인 광원을 배치하였다. y축 방향에서 바라봤을 때 너무 어두워서 y축에는 ambient값만 (0.2, 0.2, 0.2)를 가진 광원을 두개 배치하였다. 그 다음 cube에 material를 입혔다. glMaterial 메소드를 이용하는데 이도 마찬가지로 ambient, diffuse, specular, shininess 값을 가진다. 이 과제에선 6개의 면을 각각 cyan plastic, pearl, silver, yellow rubber, gold, obsidian으로 구성해 보았다. 이 때 각각의 면의 ambient, diffuse, specular, shininess 값은 아래와 같았다.

	ambient	diffuse	specular	shininess
cyan plastic	(0, 0.1, 0.06)	(0, 0.5, 0.59)	(0.50, 0.50, 0.50)	0.25
pearl	(0.25, 0.21, 0.21)	(1, 0.83, 0.83)	(0.3, 0.3, 0.3)	0.088
silver	(0.25, 0.2, 0.07)	(0.75, 0.61, 0.23)	(0.63, 0.56, 0.37)	0.4
yellow rubber	(0.05, 0.05, 0.0)	(0.5, 0.5, 0.4)	(0.7, 0.7, 0.04)	0.078125
gold	(0.19, 0.19, 0.19)	(0.51, 0.51, 0.51)	(0.51, 0.51, 0.51)	0.4
obsidian	(0.05, 0.05, 0.07)	(0.18, 0.17, 0.23)	(0.33, 0.33, 0.35)	0.3

참조 - <http://devernay.free.fr/cours/opengl/materials.html>