**[CSE4170/AIE4012] 기초 컴퓨터 그래픽스**

**HW3: OpenGL API 함수를 사용한 3차원 뷰잉**

담당교수: 임인성

학번: 20231632

이름: Jumagul Alua

**1. [목적]**

(a) OpenGL API 함수(Core Profile)를 사용하여 3차원 모델링 변환을 구현한다.

(b) 가상 동적 카메라를 기반으로 OpenGL 3차원 뷰잉 파이프라인을 구현한다.

**2. 요구 사항 구현**

(a)i. 3D 정적 물체 배치

5개의 정적 3D 기하학적 물체를 건물 내부에 자연스럽게 배치했다. 원래의 방 구조와 저명 물체들은 그대로 두고 나머지는 없어버렸다. 그리고 dragon, godzilla, cat, optimus, helicopter을 배치하였다. 이들은 모두 건물 내에 적절히 위치하고 크기 조정되어 장면을 풍성하게 만들었다. 각 물체는 일정한 변환을 거쳐 배치되었고, 그 결과 건물 내부의 자연스러운 3D 공간을 구현할 수 있었다. 또한, define\_object 함수 안에서 diffuse 변수로 rgb 색깔을 사용해서 물체 각자 다영한 색으로 칠했다. 이는 rgb color/255로 0부터 1까지의 값으로 정해서 설정되었다.

(a)ii. 3D 동적 물체 구현

동적인 물체로는 '벤'과 '스파이더' 두 개의 물체를 선택하여 구현했다. 이 물체들은 각각 건물 내부에서 지정된 경로를 따라 움직이며, 이를 위해 물체마다 적절한 모델링 변환이 적용되었다. 벤은 좌우로 자연스럽게 움직였고, 스파이더는 건물 내부에서 상하로(cos 라인을 따라) 움직이는 애니메이션을 구현했다. 경로 상에 물체들이 벽을 통과하거나 비현실적인 움직임을 보이지 않도록 충분히 고려하여 설계했으며, 결과적으로 두 동적 물체는 자연스러운 경로를 따라 이동하게 되었다. 여기서도, 둘이 핑크색과 보라색을 가지게끔 설정되었다.

(b)i. 주 카메라 이동 구현

주 카메라는 마우스 및 키보드를 통해 세 축 방향으로 이동할 수 있도록 구현했다. 'w', 's', 'a', 'D', 'e', 'q' 키를 사용하여 카메라의 이동이 가능했다. 각 축 방향으로의 이동은 일정한 속도로 이루어졌으며, 사용자가 쉽게 카메라를 조작할 수 있도록 했다. 카메라 이동은 부드럽게 이루어졌고, 3D 공간 내에서 원하는 위치로 자유롭게 이동할 수 있었다. 하지만, 소문자 ‘d’는 이미 depth를 관리하는 것으로 구현되었기에, 이동은 대문자로 설정한 점이 불편하게 느낄 수 있었다는 생각이 들었다.

(b)ii. 카메라 회전 구현

주 카메라는 키보드를 통해 3가지 축에 대해 회전할 수 있도록 구현했다. 'j', 'l', 'i', 'k', 'u', 'o' 키를 이용하여 카메라의 yaw(좌우 회전), pitch(상하 회전), roll(회전)을 제어할 수 있었다. 이 회전은 각 축에 대해 자유롭게 이루어졌으며, 사용자는 원하는 방향으로 카메라를 회전시킬 수 있었다. 회전 구현은 직관적이고 유연하게 설계하려고 노력했다.

(b)iii. 카메라 줌 기능

카메라는 '+'와 '-' 키를 이용한 줌 기능을 지원했다. 이를 통해 사용자는 카메라의 시야각(fovy)을 확대하거나 축소할 수 있었다. 또한, 마우스 휠을 통해서도 줌 기능을 조정할 수 있었다. 줌 인과 줌 아웃 기능은 자연스럽게 동작했으며, 주 카메라의 시야를 변경하여 보다 가까운 또는 먼 시점에서 장면을 볼 수 있었다.

(b)iv. 정적 CCTV 카메라

정적 CCTV 카메라는 건물 내에 고정된 위치에서 장면을 감시하는 역할을 했다. 이 카메라는 사용자가 화면을 전환하는 동안 고정된 위치와 방향을 유지했다. 총 3개의 CCTV 카메라를 배치했으며, 각각은 고유의 시점에서 장면을 보여주었다. CCTV 카메라는 각기 다른 시점을 제공하여 사용자가 건물 내부를 다양한 각도에서 볼 수 있도록 했으며, 또한 'Z' 키를 눌러 정적 CCTV와 직교 투영 카메라 간의 전환이 가능했다.

(b)v. 동적 CCTV 카메라

동적 카메라의 뷰를 윈도우에 상단에 놓었고 이는 정적 카메라랑 달리 항상 그 자리에 있다. 이 또한, 건문 내부에 위치되어 있으며, 아무것도 안 누른 상태에서는 ben의 앞에 배치되었으며 이때, Shift+좌측 마우스 클릭을 누르면 카메라가 y축으로 이동하여 거미의 움직임을 감시할 수 있게 된다. 그리고, 사용자는 ‘O’, ‘P’를 눌러 카메라의 시점을 이동하거나 확대/축소할 수 있었다. 이 동적 CCTV 카메라는 카메라의 이동과 방향을 자유롭게 조정할 수 있는 장점이 있어, 사용자가 더 많은 자유도를 가지고 장면을 관찰할 수 있게 했다.

(b)vi. 직교 투영 카메라

직교 투영을 사용하는 세 가지 카메라(정면도, 측면도, 상면도)를 구현했다. 이 카메라는 각기 다른 뷰포트를 통해 장면을 보여주었으며, 'Z' 키를 눌러 활성화할 수 있었다. 배치된 위치는 건물의 끝 좌표인 (230,160)의 절반인 (115, 80) 에서의 위치를 사용해 설정되었다. 따라서, 직교 카메라는 시점의 정확한 비율로 장면을 보여주기 때문에, 건물의 내부나 외부를 명확하게 볼 수 있는 장점을 제공했다.

(b)vii. 카메라 프레임

주 카메라와 CCTV 카메라 모두 카메라 프레임(RGB 축)을 적절한 키 입력을 통해 토글할 수 있도록 구현했다. 이는 't' 키를 통해 주 카메라와 CCTV 카메라에서 RGB 좌표축을 쉽게 표시하거나 숨길 수 있도록 했다.

* 스스로 채점한 점수: 필수적으로 구현해야 되는 부분을 구현하여, 2 (a), (b)는 만점을 주고, (c)는 다른 학생들의 작품을 못 봤으니 비교는 못 하겠지만, 제 프로그램의 복잡도가 많이 아 없어서 추가 점수의 1/3정도는 줄 것 같다.

**3. 프로그램 사용법**

이동:

* w: 앞쪽으로 이동
* s: 뒤쪽으로 이동
* a: 왼쪽으로 이동
* D: 오른쪽으로 이동
* q: 아래로 이동
* e:위로 이동

회전:

* j: 좌측으로 회전 (Yaw)
* l: 우측으로 회전 (Yaw)
* i: 위로 회전 (Pitch)
* k: 아래로 회전 (Pitch)
* u: 좌측으로 회전 (Roll)
* o: 우측으로 회전 (Roll)

줌:

* +: zoom in
* -: zoom out
* 마우스 휠: zoom in/out

CCTV:

* Z: 정적 CCTV와 직교 투영 카메라 간 전환
* t: 카메라 프레임 코글
* P: 동적 CCTV zoom in
* O: 동적 CCTV zoom out
* Shift + 좌측 마우스 클릭: 동적 CCTV Y축 이동 시작
* Escape: 프로그램 종료