

Project1: Basic OpenGL Viewer

Class

파이썬 파일을 모듈화 했을 때 변수를 수정하면 다른 파일에서 적용되지 않는다.

이를 해결하기 위해 클래스를 만들어 넘기기로 하였다.

Camera Class

- 카메라의 eye를 표현하기 위해 구면좌표계의 개념을 사용하였다.

$$\begin{aligned}x &= \rho \sin \theta \cos \phi \\y &= \rho \sin \theta \sin \phi \\z &= \rho \cos \theta\end{aligned}$$

- ρ 는 distance, ϕ 는 azimuth, θ 는 elevation을 말한다.

Drag Class

- 포인터의 x, y 좌표를 저장하는 클래스

Orbit

- 작동 방식: 마우스 왼쪽 버튼 & 드래그
- mouse_button_callback을 사용해 왼쪽 버튼을 눌렀을 때의 좌표를 저장한 뒤 cursor_position_callback을 통해 drag의 x, y offset을 구하여 각각 azimuth, elevation의 값에 더해주었다.
- sin(elevation)의 값이 음수이면 카메라의 up벡터를 (0,-1,0)으로 바꿔주고 양수면 (0,1,0) 으로 바꿔주었다.

Pan

- 작동 방식: 마우스 오른쪽 버튼 & 드래그
- mouse_button_callback을 사용해 오른쪽 버튼을 눌렀을 때의 좌표를 저장한 뒤 cursor_position_callback을 통해 drag의 x, y offset을 구한다.
- 그리고 현재 카메라의 View Matrix의 역행렬을 구한다. 이때 각 열은 u, v, w, point를 나타낸다.
- 좌우로 드래그 한 경우 u방향으로 이동하고, 상하로 드래그 한 경우 v방향으로 이동한다.
- 위의 값들을 pan벡터에 저장한다.
- 이 벡터는 나중에 glm.lookAt에 eye와 center 벡터에 더해줘 새로운 view matrix를 만든다.

Zoom

- 작동 방식: 마우스 휠
- scroll_callback을 사용해 distance의 값을 바꿔줌으로써 카메라 프레임의 w방향으로 움직이는 효과를 볼 수 있다.

Others

- key_callback을 만들어 v키를 통해 projection을 perspective / orthogonal 으로 바꿀 수 있게 한다.
- 초기 center 좌표는 (0,0,0)이다.

Screenshots

처음 두 사진은 프로그램을 키면 제일 먼저 보이는 화면과 v키를 통해 orthogonal projection으로 바꿨을 때 보이는 화면이고, 나머지는 구현 기능을 좀 더 잘 보이게 하기위해 큐브를 (0, 0, 0)에 두고 zoom-out & pan & orbit을 시행한 뒤 찍은 사진이다.

