**2021 Computer Graphics(11272) Class Assignment 1**

2015005169 최윤석

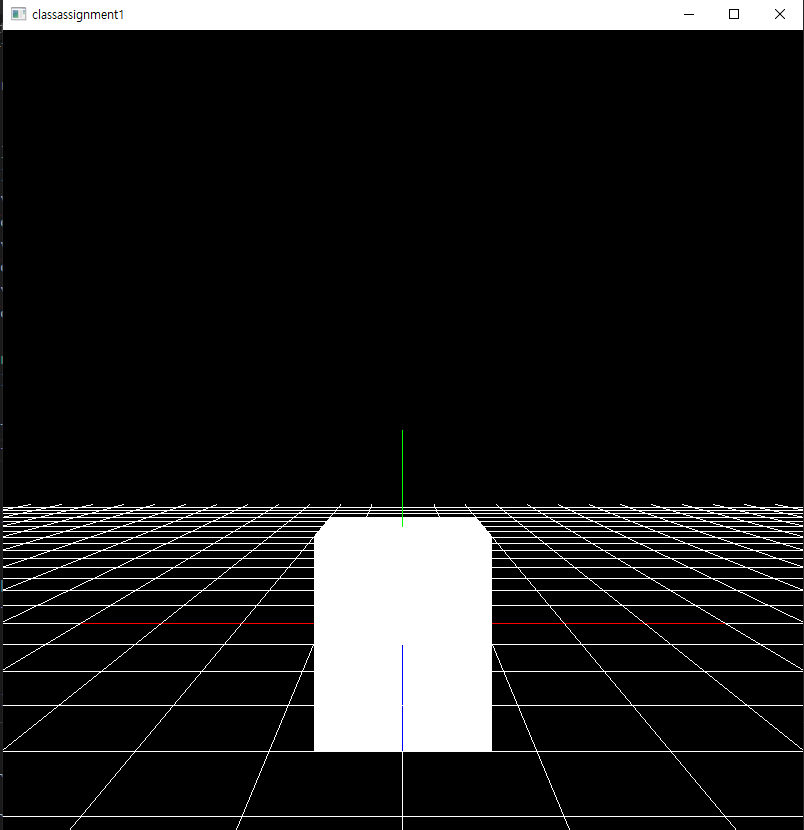
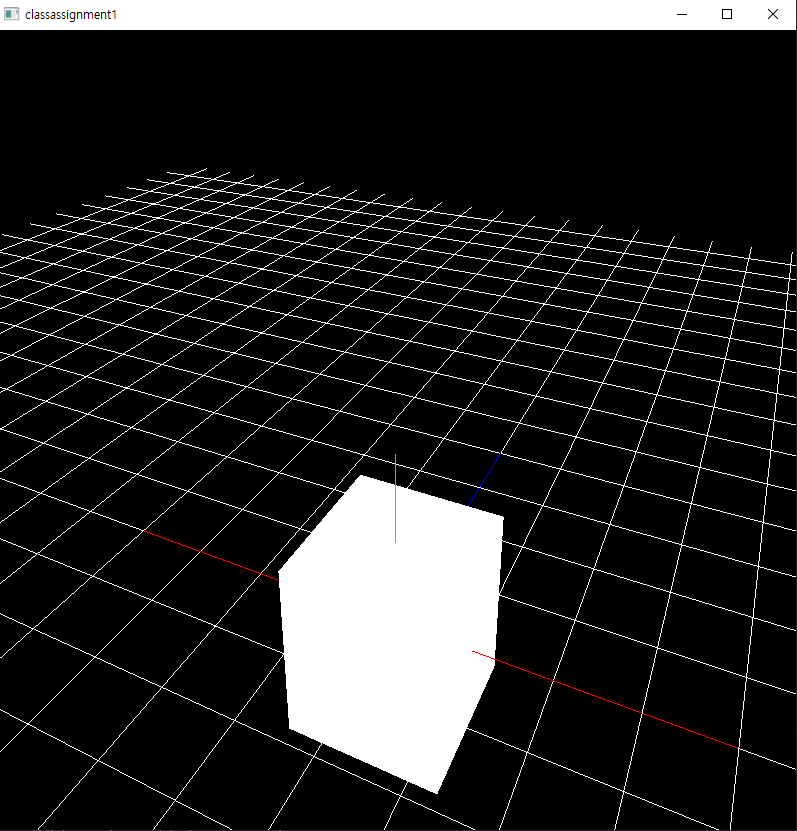
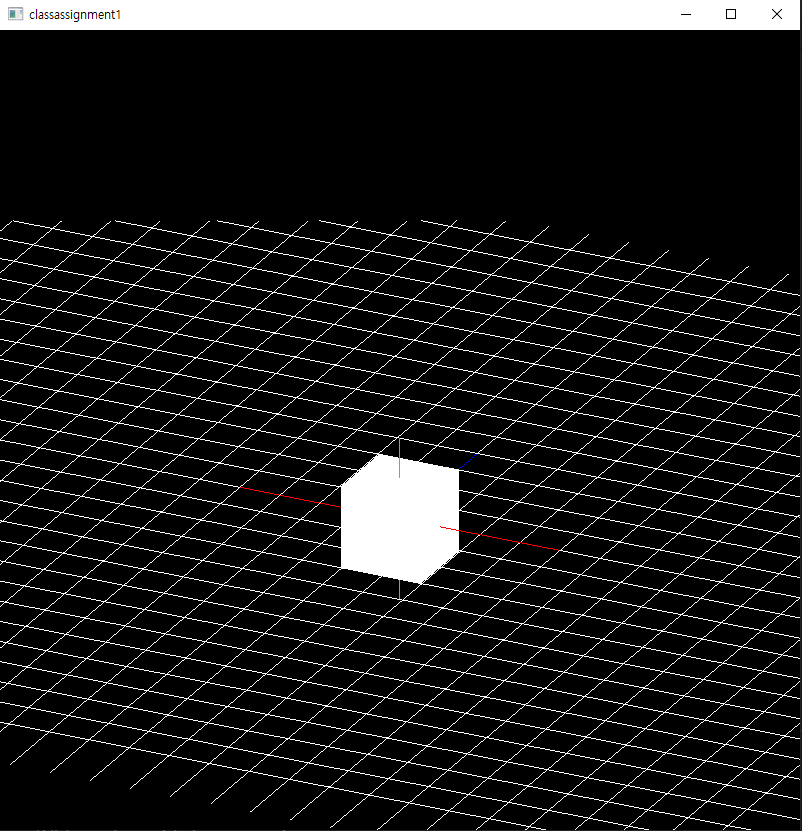
1. 프로그램 동작 방식  
 모든 요구조건을 main.py에 구현을 했기 때문에 main.py를 다운 받은 후 실행을 시키면 됩니다. Blender를 참조하라고 하여 가운데에 변의 길이가 2인 정육면체를 구현했습니다. 정육면체가 보인다면 프로그램 테스트를 진행하셔도 좋습니다.  
  
2. 요구조건 구현  
-마우스 조작-  
**A) Orbit (마우스 좌클릭 & 드래그) :** cursor\_callback함수와 button\_callback함수로 구현하였습니다. 마우스 좌클릭 입력이 들어올 경우 button\_callback에서 클릭된 순간의 x,y좌표를 글로벌 변수인 origin\_xpos, origin\_ypos에 저장해두고 cursor\_callback으로 넘어가 azimuth와 elavation을 각각 origin\_xpos, xpos, origin\_ypos, ypos값을 사용하여 구합니다. 이때 구한 각도들을 사용해 로테이션 메트릭스 a\_matrix, el\_matrix를 사용해 모든 변환이 저장되어 있는 stack\_matrix와 elav\_stack\_matrix에 곱해줍니다.

**B) Panning(마우스 우클릭 & 드래그)** : Orbit과 유사하게 button\_callback에서 마우스 오른쪽 입력이 들어오면 눌린 위치의 xpos,ypos를 origin\_panningx, origin\_panningy에 저장을 하고 cursor\_callback2로 넘어갑니다. cursor\_callback2에서는 커서가 움직일 때의 xpos, ypos와 글로벌 변수 origin\_panningx, origin\_panningy를 사용하여 translation matrix panX\_matrix와 panY\_matrix를 만들고 stack\_matrix와 elav\_stack\_matrix에 곱해주는데 이때 panningx값과 panningy값은 적절한 이동속도를 위하여 100으로 나누어주었습니다.

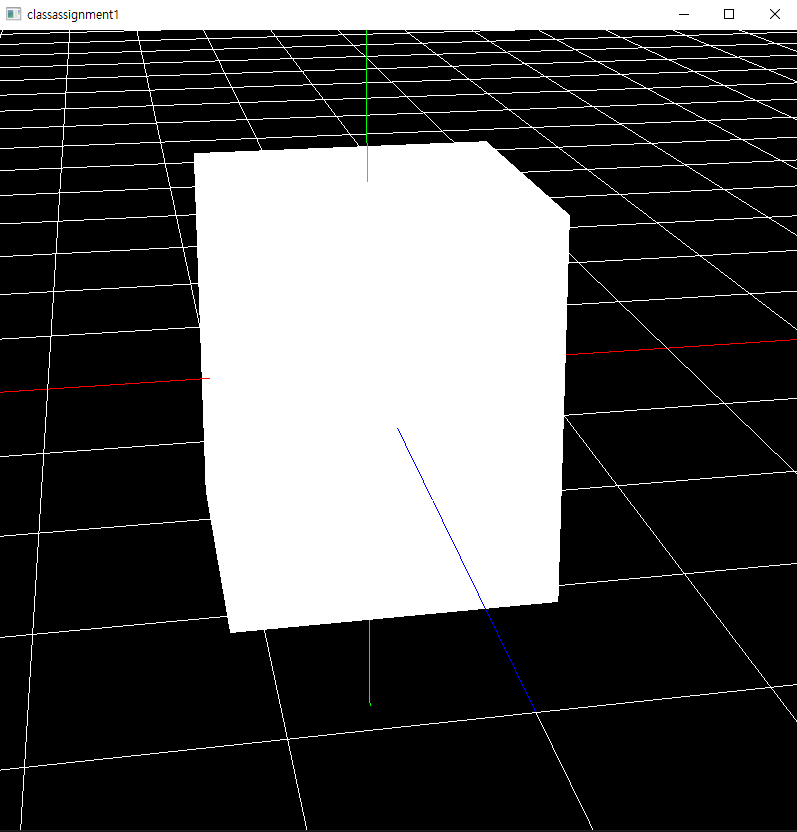
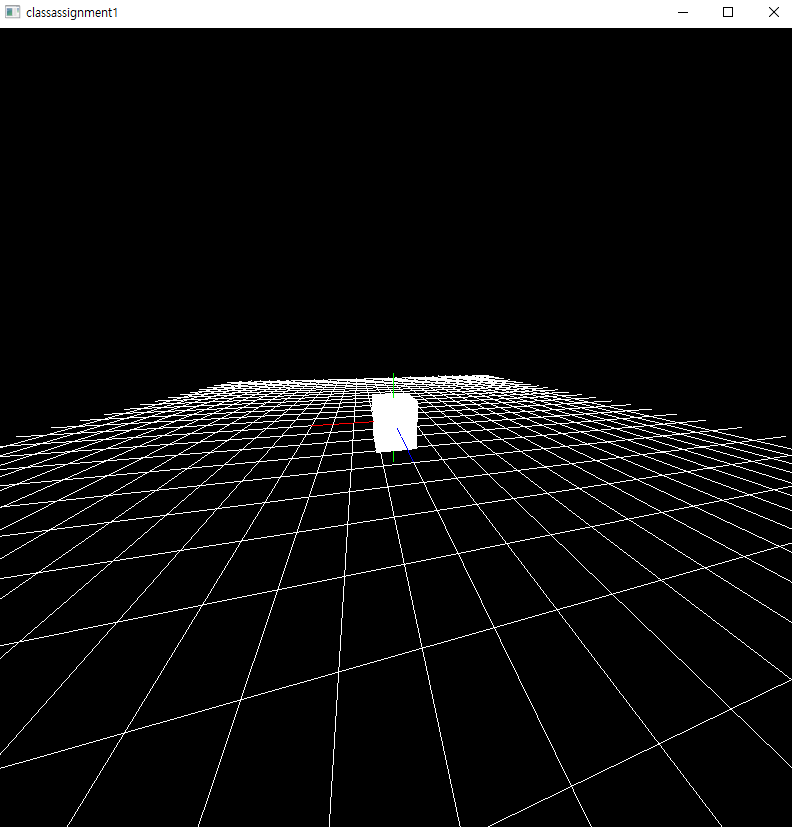
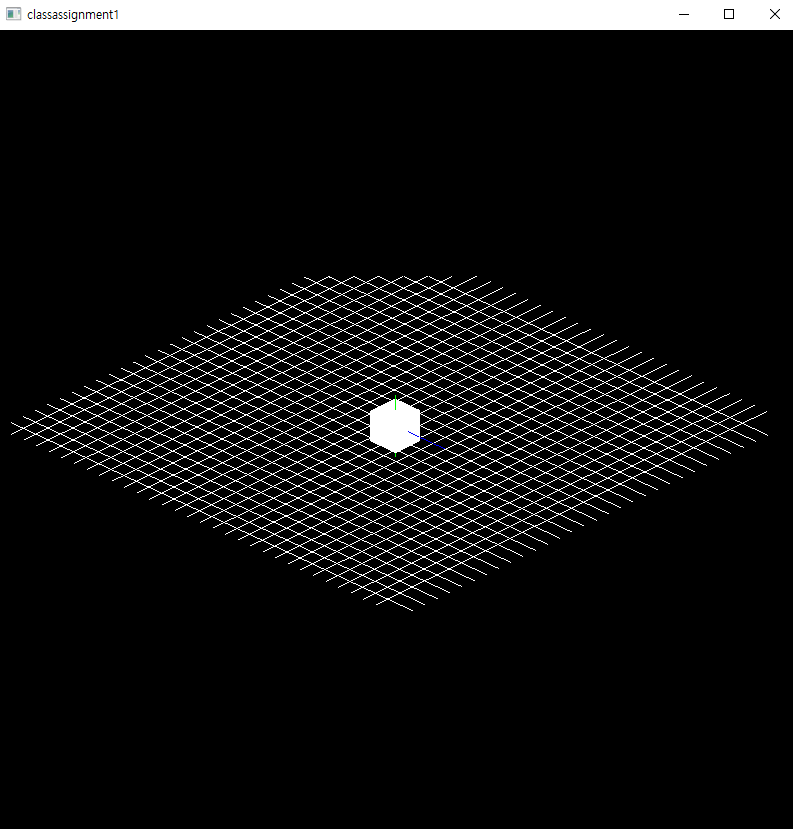
**C) Zooming(마우스 휠) :** Perspective의 경우 fovy값을 초기에 글로벌 변수 fov에 저장되어 있는 45를 넘겨주고 Orthogonal의 경우 상하 좌우 값을 fov를 5로 나눈 fov\_o값을 넘겨주어 사용하는데 이 fov값을 scroll이 발생할 때 마다 yoffet만큼의 변동을 주어 rendering시마다 변화가 일어날 수 있도록 만들었습니다. 과도한 확대와 축소를 막기위해 fov값은 1.0~120.0사이의 값이 되도록 하였습니다.

**D) V키를 사용한 Projection/Orthogonal <-> perspective 전환** : 첫 projection모드는 Perspective모드로 하였고 구분을 위하여 글로벌 변수 projection\_mode를 활용하였습니다. V키 입력이 들어올 경우 projection\_mode의 값을 P, O로 변경해가며 rendering시에 perspective로 랜더링 할지 orthogonal로 랜더링 할지 정합니다.

**E) XZ평면에 그리드 형태 구현 :** drawFrame()에서 for문을 사용하여 16행 16열의 그리드를 그렸습니다.

3.실제 실행 화면

- 첫 시작화면 - Orbit(Perspective) - Orbit(Orthogonal)



- zoom in & Panning (Perspective) - zoom out (Perspective) -zoom out (Orthogonal)