Computer Graphics Project1 Report – 2022083409 Seunggwan Kim

Implementation

1. camera.py – 카메라 클래스
   A. __init__(self, center=glm.vec3(0,0,0), radius=3.0, azimuth=0.0, elevation=0.5)
      처음 카메라의 위치를 초기화하고 zoom의 최대 최소 거리와 elevation의 최대 최소값을 초기화한다. up vector도 (0,1,0)으로 초기화한다.
   B. get_position(self)
      radius, elevation, azimuth 값을 통해 x, y, z 값을 계산해 center를 기준으로 현재 camera의 eye가 얼마나 떨어져 있는지 계산한다.
   C. get_view_matrix(self)
      get_position()에서 구한 값을 eye값으로 하는 lookAt function을 리턴한다.
   D. orbit(self, dx, dy)
      마우스의 x, y축 변화량 dx, dy 만큼 azimuth, elevation 값을 변경한다. 이때, elevation은 $[-\frac{\pi}{2}+0.01, \frac{\pi}{2}-0.01]$ 범위의 제한을 둔다.
   E. pan(self, dx, dy)
      center의 위치를 right vector, up vector 방향으로 마우스의 x, y축 변화량 dx, dy 만큼 이동시킨다.
   F. zoom(self, dy)
      마우스의 y축 변화량 dy를 radius 값을 변경한다.
2. main.py
   A. 처음 초기화
      camera 객체 생성, dx, dy 계산을 위한 last_x, last_y 초기화, orbit, span, zoom을 구별하는 drag_mode 초기화, drag를 하고 있는지 나타내는 is_dragging 초기화
   B. mouse_botton_callback(window, button, action, mods)
      LMB를 눌렀을 때 alt가 눌린 상태면 is_dragging을 true로 바꾸고 drag_mode를 alt면 orbit, alt+shift면 pan, alt+ctrl이면 zoom으로 바꾼다.
   C. cursor_position_callback(window, xpos, ypos)
      is_dragging이 false면 last_x, last_y를 현재 마우스의 x, y 위치로 계속 초기화하다가 is_dragging이 true면 그 때부터 x, y의 변화량을 계산해 drag_mode에 맞는 함수를 실행한다. 이 때 해당 함수에 맞는 키 입력이 중간에 끊기면 is_dragging=false, drag_mode=None으로 초기화한다.
   D. get_view_matrix()
      camera 클래스의 get_view_matrix를 리턴한다.
   E. prepare_vao_frame()

0.1 간격으로 RGB가 (0.7,0.7,0.7)인 grid를 만든다.

F. main()

- glfwSetMouseButtonCallback() 함수로 마우스 버튼이 입력됐을 때 mouse_button_callback() 함수를 실행한다.
- glfwSetCursorPosCallback() 함수로 마우스 위치가 변할 때마다 cursor_position_callback() 함수를 실행한다.
- V=camera.get_view_matrix() 로 view matrix를 계산한다.
- glDrawArrays(GL_LINES, 0, gird_vertex_count) gird의 vertex 개수를 계산해 grid를 그린다.