

Computer Graphics Project1 : Basic OpenGL viewer Report

2019073181 심기호

1. Which requirements you implemented

Orbit : 처음 마우스 왼쪽 버튼을 눌렀을 때의 위치를 `button_callback` 함수에서 `start_xpos`, `start_ypos`, `cur_xpos`, `cur_ypos`로 저장해두고 `leftButtonPressed`라는 변수를 `True`로 바꿉니다. 드래그를 시작할 때에는 `cursor_callback` 함수에서 `start_xpos`와 `start_ypos`를 각각 `cur_xpos`, `cur_ypos`로 받아서 예전 위치를 저장해두고 다시 `glfwGetCursorPos`로 현재 위치를 받아서 `elevation`과 `azimuth`값을 계산합니다. $\sin(\text{elevation})$ 값이 <0 일때와 >0 일때가 orbit방향이 반대로 돌기 때문에 $\sin(\text{elevation})$ 이 음수일 때에는 `azimuth`에 음수를 곱해주어 계산해주었습니다. 그렇게 계산된 `azimuth`와 `elevation`값이 `prepare_V_mat` 함수에서 camera의 frame을 계산해주는데 사용됩니다. `w_vec`, `v_vec`, `u_vec`을 외적과 내적과 정규화를 이용해서 벡터들을 계산해주고, camera pos vector를 $\text{distance} \cdot \text{w_vec} + \text{targetpos}$ 해서 계산해줍니다.

Pan : Pan도 Orbit과 마찬가지로 `button`, `cursor_callback` 함수에서 `start_xpos`, `start_ypos`, `cur_xpos`, `cur_ypos`를 계산해주고 `rightButtonPressed`라는 변수를 `True`로 바꿉니다. 그 후 `cursor_callback`에서 `rightButtonPressed`가 `true`라면 `start_pos`와 `cur_pos`간의 차이를 `x,y` 각각 `u_vec`, `v_vec` 곱해준 후 `targetpos`와 `camerapos`에 각각 더해주었습니다.

Zoom : 카메라와 물체 사이의 거리를 의미하는 `distance` 변수를 scroll할 때 마다 `distance` 변수를 바꾸게 했습니다. `Distance`가 0이 되면 안되기 때문에 `yoffset`만큼을 지수에 곱해줘서 변화를 주었습니다.

Toggle : `key_callback`에서 `V`를 누르고 안누르고에 따라서 `mode`라는 변수를 바꿔줍니다. `main` 함수의 `while`문에서 `mode`에 따라서 `P matrix`를 `perspective`또는 `ortho`로 toggle해줍니다.

Draw a grid : 먼저 `prepare_vao_grid(i)`를 이용해서 점들을 위치를 `i`만큼 변화를 조금씩 줍니다. 그리고 메인문에서 `for`문을 이용하여 `prepare_vao_grid(i)`를 호출해서 점들을 화면에 띄우게 됩니다.

2. A few screenshots of your program

