# **도표이(가) 표시된 사진 자동 생성된 설명Implements**

The description of the implementation is based on '2. Requirements - A' in the task specification document.

ii. When the program is first executed, the camera looks at the target point (0,0,0) from the position (0, 0, -1).

Figure 1

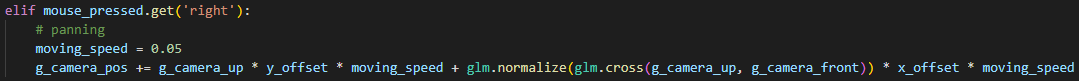
\* Explanation of the global variables used:

1. g\_azimuth, g\_elevation: Angles shown in Figure 1.
2. g\_projection\_is\_ortho: A flag that distinguishes whether it is an orthographic projection or not (since it starts with a perspective projection, the initial value is False).
3. g\_screen\_width, g\_screen\_height: Screen size.
4. last\_mouse\_x\_pos, last\_mouse\_y\_pos: Stores the most recent clicked point on the screen.
5. mouse\_pressed: A flag that stores whether the left mouse or the right mouse is clicked.
6. g\_P: Projection matrix.
7. g\_camera\_pos, g\_camera\_front, g\_camera\_up: Camera position, vector indicating the direction in which the camera is facing(-w vector in camera frame), and the vector indicating the up direction of the camera(v vector in camera frame).

iii-1. **Orbit**: As shown in Figure 1, the angle rotated in the direction from the +z-axis to the +x-axis is called azimuth, and the angle rotated in the direction from the +y-axis to the xz plane's first quadrant is called elevation. In the mouse\_button\_callback function, it checks whether the left mouse is pressed and sets the mouse\_pressed['left'] flag accordingly. When the left mouse is pressed and the cursor is moved, the logic related to rotation is processed inside the if statement where the flag is true in the cursor\_position\_callback. Assuming that the window we see is the xy plane, we add the x\_offset moved by the cursor in the x-direction to g\_azimuth and the difference moved in the y-direction, which is y\_offset, to g\_elevation.

|  |  |
| --- | --- |
| Rotation vector for camera\_front | Rotation vector for camera\_up |
| 도표이(가) 표시된 사진  자동 생성된 설명텍스트이(가) 표시된 사진  자동 생성된 설명 | 도표이(가) 표시된 사진  자동 생성된 설명텍스트이(가) 표시된 사진  자동 생성된 설명 |

iii-2. **Pan**: Until the right mouse button is clicked and released, the mouse\_button\_callback sets the flag for the right mouse to True, which allows it to enter the right if statement in the cursor\_callback function. At this point, we multiply y\_offset by the camera\_up vector direction and add it, and we multiply x\_offset by the direction vector of the cross product between camera\_up and camera\_front and add it, allowing us to move left and right in the u direction and up and down in the v direction with respect to the camera frame.



iii-3. **Zoom**: When a scroll event occurs, the scroll\_callback function is called. By multiplying y\_scroll by the vector in the direction of camera\_front and adding it to the existing camera\_pos, we can move in the direction of camera\_front vector(camera\_front vector is along w axis of the camera frame mentioned in assignment document).



However, in perspective projection, depth is represented, so zooming in and out can be visually observed when scrolling. In contrast, in orthogonal projection, depth is not represented, so zooming cannot be observed.

iv. **Toggle** **projection**: By pressing v key, the key\_callback is called and it goes into the logic in if statement according to the flag that shows the state of projection(True when orthogonal projection). By using flag, we can toggle projection. After changing the flag, the corresponding projection is calculated. In orthogonal projection, the ortho\_height is fixed and the ortho\_width is calculated according to the screen aspect ratio. Then, the values corresponding to the 6 arguments (left, right, bottom, top, near, far) of the ortho() function are filled in. In perspective projection, near and far are fixed, as well as fov, and aspect\_ratio is calculated based on the screen size and then filled in. Additionally, a framebuffer\_size\_callback is added to adjust the window size when the screen size changes(This is optional feature).

텍스트, 스크린샷, 화면, 은이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 2 toggle projection matrix Figure 3 change size of screen keeping ratio

v. **grid on xz plane**: In the prepare\_vao\_grid() function, the vertex position and color information required for the vao is stored and then returned as the vao. In draw\_grid(), the information from the vao is drawn using glDrawArrays(GL\_LINES, 0, 84).

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 4 grid VAO Figure 5 draw grid function(called in main while loop)

Optional Implement. **World frame**: x, y, z axis for red, green, blue line to help where is the direction of camera is watching. You can disable world frame by pressing ‘F’.

# **Screenshots**

차트이(가) 표시된 사진

자동 생성된 설명 차트이(가) 표시된 사진

자동 생성된 설명 차트이(가) 표시된 사진

자동 생성된 설명 차트이(가) 표시된 사진

자동 생성된 설명 웹이(가) 표시된 사진

자동 생성된 설명

