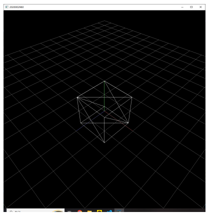
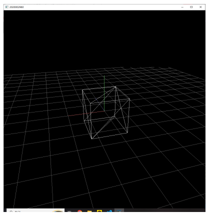
[CG] Project1 2020002960 박연진

Initial Target Point : (0, 0, 0)



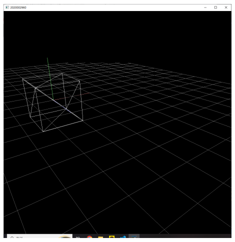
implemented by initializing global variable `g_trans` and `target_position`

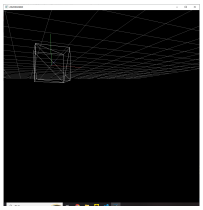
Orbit : mouse left button & drag



1. Mark if the user is clicking mouse left button in global variable `isDraggingLeft` in `mouse_button_callback` function
2. If the user is clicking mouse left button and dragging, change global variable `g_elevation` and `g_azimuth` in `mouse_callback` function
3. In \`main\` loop, set three global variables `g_w`, `g_u`, `g_v` with the modified `g_elevation` and `g_azimuth` values.
4. Set `target_position` and `camera_position` with modified `g_w`, `g_u`, `g_v` vectors.
5. Set variable `V` using `glm.lookAt` function.

Pan : mouse right button & drag

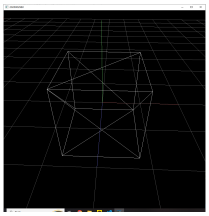




1. Mark if the user is clicking mouse right button in global variable `isDraggingRight` in `mouse_button_callback` function
2. If the user is clicking mouse right button and dragging, change global variable `g_trans` (with respect to `g_w` and `g_v`) in \`mouse_callback\` function

3. In `main` loop, set `target_position` and `camera_position` with modified `g_trans` vector.
4. Set variable `V` using `glm.lookAt` function.

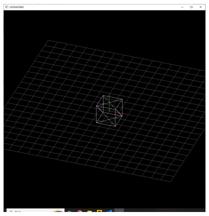
Zoom : mouse wheel



1. If user rotates mouse wheel, modify `g_distance` in `scroll_callback` function
2. In `main` function, set `camera_position` using `g_distance`
3. Set variable `V` using `glm.lookAt` function.

Perspective (orthogonal / perspective) : v key

1. Initialize `g_perspective` variable as `True`.



1. Toggle `g_perspective` variable if v key is pressed.
2. According to `g_perspective` value, set matrix `P` using `glm.ortho` function or `glm.perspective` function.

Rectangular Grids

1. Define `prepare_vao_grid` function
2. Draw one grid at a time calling `glDrawArrays(GL_LINE_LOOP, i * 4, 4)` in loop.