

Computer Graphics Project 1: Designing a Basic Viewer

Handed out: April 6, 2025

Due: 23:59, April 19, 2025 (NO SCORE for late submissions!)

- LMS course home > Weekly Learning (주차학습) > Week 5> Project 1
- Compress your files into a zip file as in the following example. The zip file can be named whatever you want.

```
+ submission.zip
- main.py
- ...
- report.pdf
```

- Your program may consist of several python source files. But the main module should be in **main.py**. That is, your program should be executed with the following command:

```
python main.py
```

1. In this project, you will build a **simple OpenGL viewer** and make your own design choices for camera control and interaction.
 - A. This project is not only about implementation, but about:
 - i. **making decisions among multiple valid approaches**
 - ii. understanding how camera and interaction affect the scene
 - iii. **explaining and justifying your design choices**
 - B. Your implementation will serve as a **foundation for future projects** (OBJ viewer, BVH viewer).

2. Program Requirements

- A. Basic Viewer
 - i. Use **perspective projection**
 - ii. Draw a **grid on the xz-plane**.
 - iii. Render at least **one object** (e.g., triangle or simple geometry) in the scene.

B. Camera System Design

- i. You must implement one camera system as your main design.
 1. A camera system includes both how the camera moves and how it is controlled by the user.
- ii. You must choose one of the following approaches:
 1. **Orbit camera** around a target point (Blender-style)
 - A. You may refer to the following examples of camera manipulation:
 - i. Blender camera controls:
 1. <https://docs.blender.org/manual/en/latest/editors/3dview/navigate/navigation.html>
 - ii. Interactive example (Three.js):
 1. https://threejs.org/examples/#misc_controls_orbit
 - Orbit: LMB+drag
 - Pan: RMB+drag
 - Zoom: scroll or MMB+drag
 - B. (Note: exact controls may vary depending on your implementation)
 2. **Free camera** (FPS-style)
 - A. You may refer to the following examples of camera manipulation:
 - i. Interactive example (Three.js):
 1. https://threejs.org/examples/#games_fps
 - Move: WASD keys
 - Look around: mouse movement
 - Jump: space
 - B. (Note: exact controls may vary depending on your implementation)
- iii. In your **report**:
 1. **Compare your approach with the other option** and clearly explain the differences and trade-offs.

- A. You are NOT required to implement both approaches. The comparison can be based on your understanding and analysis.
- 2. **Explain why your choice is suitable** for your viewer.
- iv. Optional: You may implement an additional custom camera behavior, but this will not result in additional points.

C. Interaction Design

- i. Choose how users interact with your camera.
 - 1. Input mapping is your design choice. There is no fixed requirement for specific keys or mouse inputs, as long as your design is clearly explained.
- ii. Examples:
 - 1. Mouse-based interaction
 - 2. Keyboard-based interaction
 - 3. Combination of both inputs
- iii. In your **report**:
 - 1. You must clearly describe your interaction design in your report, including:
 - A. **how the user controls** the camera (e.g., mouse, keyboard, or both)
 - B. **how the camera changes** in response to each user input
 - Example:
 - Input: drag mouse
 - Behavior: camera rotates around the target point
 - C. **why your interaction method is suitable** for your camera system (e.g., suitable for exploring objects, intuitive control, or familiar interaction style)
 - D. **any design considerations or trade-offs** in your interaction method (e.g., what your design does well and what it does not do well)

3. Video Requirements

- A. A **hyperlink to a video** demonstrating your program, uploaded to an online video streaming service (e.g., YouTube, Vimeo).
 - i. The video is intended to demonstrate functionality, not to provide a explanation.
 - ii. The video must clearly **demonstrate camera movement**. Explanations or annotations are not required.
- B. **The uploaded video MUST be publicly accessible.**
- C. The video should be **no longer than 10 seconds**.

4. Report Requirements

- A. The report should include:
 - i. **Camera System Design report requirement (B-iii)**
 - ii. **Interaction Design report requirement (C-iii)**
 - iii. **Video link (3-A)**
- B. The report should be concise (**approximately 1–3 pages**). Clarity is more important than length.
- C. You may write the report in **either English or Korean**. There is **no advantage or disadvantage based on the language you choose**.

5. What to submit: A zip file including

- A. **Python source files (.py)**
 - i. Your program may consist of several python source files. But the main module should be in **main.py**.
- B. **A report (.pdf)**
- C. Note:
 - The video must be accessible via a hyperlink included in the report.
 - If the video link is missing or not accessible, it may be treated as missing and result in a penalty.

6. **Runtime** Requirements

- A. You must implement all requirements in a single program. This project DOES NOT require each requirement to be a separate program.
- B. The window size doesn't need to be (800, 800). Use the larger window that is enough to see the details of the viewer.
- C. **Your program must use OpenGL 3.3 Core Profile**, meaning that ...
 - i. Your python code must include:

```
glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 3)
glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 3)
glfwWindowHint(GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE)
```

- ii. Your shader code must start with:

```
#version 330 core
```

- D. **Your program should be able to run on Python 3.10 with only NumPy, PyOpenGL, glfw, PyGLM, Pillow installed. Do not use any other additional python modules.**
- E. Only **glfw** is allowed for event processing and window & OpenGL context management. **Do not use glut functions for this purpose.**
- F. **If your program does not meet these requirements, it may not run on the TA's computer, which may result in a significant penalty.**

7. **Evaluation** will be primarily based on:

- A. your camera design and justification (as required in B-iii)
- B. your interaction design and justification (as required in C-iii)
- C. how clearly your video demonstrates your program
- D. Note:
 - The video and report are essential components of this project.
 - Insufficient or missing video/report may result in significant penalties.
 - Correct implementation alone is not sufficient for a high score.
 - Strong emphasis is placed on how clearly you explain your design choices.