

## Computer Graphics Assignment 1: Basic OpenGL viewer

Handed out: March 29, 2021

**Due: 23:59, April 11, 2021 (NO SCORE for late submissions!)**

- Only accept answers submitted via git push to this course project for you at <https://hconnect.hanyang.ac.kr> (<Year>\_<Course no.>\_<Class code>/<Year>\_<Course no.>\_<Student ID>.git).
- Place your files under the directory structure **<Assignment name>/<your files>** just like the following example.

```
+ 2020_ITE0000_2019000001
+ ClassAssignment1/
- main.py
- report.docx
```

- The submission time is determined not when the commit is made but when the git push is made.

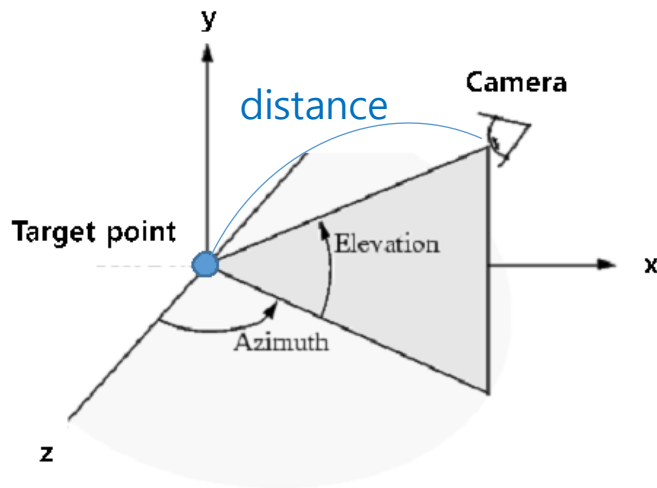
1. Implement a basic OpenGL viewer. This viewer will also be used in your future class assignments.
  - A. You have to implement all requirements in a single program. This assignment DOES NOT require each requirement to be a separate program.
  - B. The window size doesn't need to be (480, 480). Use the larger window that is enough to see the details of the viewer.
  - C. **DO NOT** set the window title to **your student ID**.
  - D. Total points: 80 pts

## 2. Requirements

- A. **Manipulate the camera with mouse movement (70 pts)**
  - i. Refer the camera manipulation of Blender software.
    1. <https://www.blender.org/download/>

ii. The camera of your program should initially look at a target point, similar to that of Blender.

1. Initialize the target point to the origin (0, 0, 0)



2.

iii. Provide the following three camera control operations.

1. **Orbit:** Rotate the camera around the target point by changing azimuth / elevation angles. (MMB (mouse middle button) in Blender) **(20 pts)**

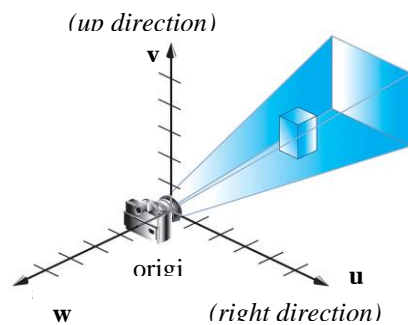
A. Do not rotate the camera about a vector from the camera to the target point.

2. **Panning:** Move both the target point and camera in left, right, up and down direction of the camera (Shift-MMB in Blender) **(20 pts)**

A. More specifically, translate both the target point and camera along u axis (left & right) and v axis (up & down) of the camera frame.

3. **Zooming:** Move the camera forward toward the target point (zoom in) and backward away from the target point (zoom out) (Ctrl-MMB in Blender) **(15 pts)**

A. A. More specifically, translate the camera along w axis of the camera frame.



B. (backward direction)

4. You MUST use the following mouse movement:
  - A. Orbit: Click **mouse left button & drag**
  - B. Panning: Click **mouse right button & drag**
  - C. Zooming: **Rotate mouse wheel**
  - D. Using above mouse movements is essential for scoring your assignment, so if you use any other set of mouse movement or keyboard shortcuts for Orbit / Panning / Zooming, **you won't get any score** for them.
- iv. Toggle perspective projection / orthogonal perspective projection by pressing '**v**' key **(10 pts)**.
  1. When the program is executed, it starts in perspective projection mode.
- v. Draw a **rectangular grid with lines (not polygons) on xz plane** as a reference ground plane (similar to Blender). Choose number of rows and columns, size as you want. **(5 pts)**

### 3. Report (10 pts)

- A. Submit a report of **at most 2 pages** in docx file format (MS Word). Do not exceed the limit.
- B. The report should include:
  - i. Which requirements you implemented (5 pts)
  - ii. A few screenshot images of your program (5 pts)
- € You do not need to try to write a long report. Just only write down the required information. Use either English or Korean.

### 4. Runtime Environment

- A. Your program should be able to run on systems only with Python 3.7 or later, NumPy, PyOpenGL, glfw. Do not use any other additional python modules.
- B. Only **glfw** is allowed for event processing and window & OpenGL context management. Do not use **glut** functions for this purpose.
- C. If your program does not meet this requirement, it will not run on TA's computer **so**

**you will not get any score for this assignment (except report).**

**5. What you have to submit:**

**A. .py files**

- i. You can use multiple .py files for this assignment. In this case, explain how to run the program in the report.

**B. .docx report file**