CSE 333/533 - Monsoon 2023
Assignment 2: Modelling, Viewing and Projection
Due date: 23:59, 19 Sep. 2023

Consider the world coordinate space with X axis to the right, Y axis up and Z axis coming out of the screen plane. The given program renders a scaled cube (a cuboid) with the bottom face centered at the world space origin and a scale factor of 2 in the Y direction. Compile and run it.

Study the program and understand how The viewing and projection matrices are constructed with GLM and are copied to the vertex shader as Uniform variables.

1. Program the arrow keys on your keyboard to change the camera view of the scene.
   (a) The camera center should change as follows:
   - Left and right arrows move the camera along -X and +X of the camera axes.
   - Down and up arrows move the camera along the -Y and +Y of the camera axes.
   - Shift + down/up arrows move the camera along the -Z and +Z of the camera axes. This will give the effect of Zoom out/in.

   Keep the look-at point to be the world space origin (0,0,0) and the camera up-vector to the world vector (0, 1, 0). Do not forget to update the viewing matrix after every change in camera center. Note that arrow keys should change camera center coordinates in the camera space (and not in the world space), which will result in a camera motion on a sphere around the origin. The `glm::lookAt()` in `setupViewTransformation()` requires camera center in world space.
   (b) Move the camera to specific positions and generate one-point perspective, two-point perspective, and the two three-point perspective views (bird's eye view and rat's eye view).

[Functionality: (a) 10 marks (b) 5 marks, Code quality and doc: 5 marks, Total: **20 marks**]

2. The given program performs a perspective projection using GLMs `perspective()` function. Keep the arrow key movement from the previous question and
   1. Program the 'o'/'O' and the 'p'/'P' keys on the keyboard to switch between orthographic[1] and perspective projections respectively. You can use `ImGui::IsKeyPressed()` function to detect keyboard keys.
   2. Using arrow keys, move the camera to specific positions to generate top view, front elevation, and side elevation. Use a modifier key on your keyboard like Ctrl (control) to snap camera center to appropriate axes/lines in order to accurately generate such views.

[Functionality: (a) 5 marks (b) 10 marks, Code quality and doc: 5 marks, Total: **20 marks**]

3. Find the inverse of the rigid body transformation. Show all of your steps.

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

where $\mathbf{R}$ is a 3 x 3 rotation matrix and $\mathbf{t}$ is a 3-vector. [**10 marks**]

[1] `glm::ortho()` will require a scale factor to the min/max window sizes to appropriately resize the model (for example to show it in similar size as that of a perspective projection).

Deliverables (as a single zipped file **Assignment02_<studentID>.zip**) containing:

- C/C++ code (make sure to upload full code and do not include any intermediate object files, delete any other temporary files).
- 2~3 page PDF Report written with with **Latex/MS Word**. Use the acmlarge option (single column) (see sample-acmlarge.tex if writing with Latex). Include screenshots within the report itself (and DO NOT attach separately).

Total marks for this assignment: 50 marks

Note: Your code should be written by you and be easy to read. You are NOT permitted to use any code that is not written by you. (Any code provided by the instructor/TA can be used with proper credits within your program). Theory questions need to be answered by you and not copied from other sources.