C++ 3D Assignment 2 (2 pts) – Week 2.5 Moving around in the MGE

All assignments are **mandatory**.
Read the whole assignment before starting.
You need to be able to explain at least all code *you* wrote.

Assignment 2.1 Camera Inverse

Use glm to calculate the matrix for a camera that has **first** been rotated 45 degrees *around* the z axis and **then** translated by (2,4,6). What is the view matrix for this camera matrix?

Explain both matrices to your lab teacher.

Assignment 2.2 Create a camera orbit behaviour

Implement a CameraOrbitBehaviour which can be attached to a **camera**.
The behaviour should take the following parameters upon creation:

- distance
- min/max tilt angle (rotation over x axis)
- rotation speed around the y axis
- a target gameobject

As you move the mouse left/right the camera should rotate around the target at the given distance controlled by the given speed. As you move the mouse up/down the camera rotates up/down between the min/max tilt angle. (Effectively the camera moves on a (restricted) sphere with the given radius.)

The camera should always be oriented (look at) the given target, and if the gameobject moves the camera should update accordingly. Make sure the behaviour is user friendly.

Assignment 2.3 Loading an .Obj file

This assignment includes 4 .obj files that are all faulty except one. Find out which .obj is the correct one and describe what is wrong with the others. Load the correct .obj file through the Mesh class and add your camera behaviour from 2.1 to the camera to be able to view it from all sides.

Assignment 2.4 Creating a new shader

Create a new Material, call it WobblingMaterial and assign it to your loaded object. The material should make the loaded object "wobble". Basically each vertex in the loaded model should be scaled towards the model's origin, but out of phase with the scale of the other vertices (eg use sin to scale for example with different phase offsets for each vertex).

Cookie bonus: animate the texture as well.