Computer Graphics Assignment #1c
Wireframe Viewer
Cameras and others

Part 3 of the first assignment will add cameras and other features. For this part please create 'Assignment1Report/Assignment1Report_part3.md', and do the following:

1. Implement a camera with an orthographic projection, and **remove the centering trick** used in the second part of the assignment. Place a mesh in the scene such that it is visible (i.e. it is inside the view volume). Allow the user to change the view volume (e.g. up, down, left, right, near, far) and compare the result with two different view volume. Note: near and far should not affect the rendering in this assignment.

2. Allow the user to reposition the camera in the camera frame and the world frame using incremental changes. Explain how you implemented them and show results.

3. Allow the user to set the view volume automatically based on the window's size, and show the result on two window sizes. The proportions of the drawn object must remain the same! Verify your result using a sanity check and explain it in the report.

4. Implement a feature in the renderer that draws the axes of the model and the world frames (as short lines or arrows). Show the difference between transforming a model in the world frame and in the model frame while its axes are visible. Transforming in the model frame should keep the model axes fixed, while transforming in the world frame will transform the axes too.

5. Load a different mesh and pick two *non-commuting* transformations T_1, T_2. Compare the results of applying T_1 in model frame and T_2 in world frame vs. T_1 in world frame and T_1 in world frame. Explain the differences.

6. Implement and display the bounding box of the object, in local and world coordinates. Demonstrate the differences between them.

7. Compute and display the face normals and vertex normals. Note that in OBJ files there can be multiple normals for one vertex, one normal for each face that coincides with the vertex. Make sure the the normals are scaled reasonably (not too long that they cover the whole screen, and not to short, that they look like points). Design a sanity check to test your implementation. In addition, show that the normals are transformed correctly when the object is rotated and translated. Demonstrate and explain in the report.

8. Load a mesh, and move the camera away from it. implement a perspective projection and compare the orthographic projection vs. the perspective projection. Make sure that the differences are clearly visible, and suggest and test a sanity check for the correctness of the projections.

9. For pairs only: Set the projection to perspective. Show the difference between changing the zoom (by changing the size of the frustum) and moving closer or farther from the model.

10. For pairs only: Set the camera position to (a, a, a) and orient it towards the origin using LookAt. Pick a such that the model is clearly visible.

11. For pairs only: Implement a dolly zoom (the vertigo effect), and use a slider to control it. Show the result.

12. For pairs only: Put at least two screenshots showing multiple models and cameras on screen.

13. Put a screenshot showing as much of your GUI as possible.

14. Describe any additional features or bonuses you implemented.