

Introduction to computer graphics. Spring 2019. Delivery 1.

Goal: The goal of this practice is to implement camera and world transformations showing a basic understanding about how the camera, world and object coordinates work in OpenGL, as well as an illumination with the phong lightning work. For this purpose, you will prepare a simple scene which loads a model of a character, and draw some elements in the background, based on the cube primitive to give it a context.

Exercises:

1. Load a 3D model of a character, and render it with the phong model. Use the cube primitive to load some additional objects (walls, furniture, etc.). Make sure the distinction between main character and background is clear in the following exercises. **(2 points)**
2. Implement rendering of the loaded model using the phong rendering model. Make sure that the parameters of the phong model can be adjusted through the user interface. **(4 points)**

To provide the user input use the ImGui method:

ImGui::DragFloat("drag float", &f1, 0.005f);

3. Inverse dolly effect **(4 points)**

- a) A close-up in perspective projection. The camera will approach the model loaded in the middle of the scene, with some objects in background.
- b) An increase of the field of view of the camera, with the objects in the middle of the scene, and some objects in background.
- c) Using the same scene and the operations implemented in 2a) and 2b) , implement a demonstrator of the inverse dolly effect, as described in <https://www.youtube.com/watch?v=Amj6RiGiTOE&t=283s>

To provide the user input use the ImGui method:

ImGui::Button("Click");

Delivery format: To do the task, use the starting project provided in class (GL_framework.sln), and update **only** the file render.cpp

The delivery must be a .zip file with the following elements:

- a) a clear readme.txt file explaining how to run the project
- b) the file render.cpp, with the code commented
- c) the .obj model used in render.cpp
- c) a .pdf explaining how you solved the movement in exercise 3. It can be a scan of a paper document if it is clear and well written.

The .zip file will be named: Delivery1_GROUP_MEMBER1_MEMBER2.zip where GROUP is your group number (M01, for example), and each member appears as the combination of its first Surname and Name

If the file render.cpp does not work immediately in the project, the rest of the delivery will NOT be corrected