Homework 1
Topic: Getting Started with OpenGL, SDL, GLM, and Transformations

This is a programming assignment. At this stage, we assume that you have already read the preliminary reading material for Homework 1 (HW1_PR.pdf) and seen the video that shows a similar end result that you should obtain when you complete this homework (HW1_demo.mp4). The goal of this assignment is for you to familiarize yourself with Visual Studio, OpenGL, SDL, and 3D transformations. For this purpose, you will implement various mouse-based transformations of your 3D test object.

Students are required to use C/C++ for this assignment. You will be writing your code in the **main.cpp** file of the starter project that is provided to you.

Please document all your answers, plots, and insights in a **single pdf** file containing all requested results (read Submission section below before you prepare the pdf file!). Finally, please submit a .zip/.7z file with all source code and the pdf you have prepared (**<Student ID>_HW<number>.zip**, e.g. 118033991234_HW1.zip) to **Piazza**.

Introduction

After you compile the starter code you should see a **single** teapot with a set of axes. The figure below (and the rest of the figures in this pdf) show three teapots, which you will implement in task 4. After task 4, your renderings should look similar to these images.

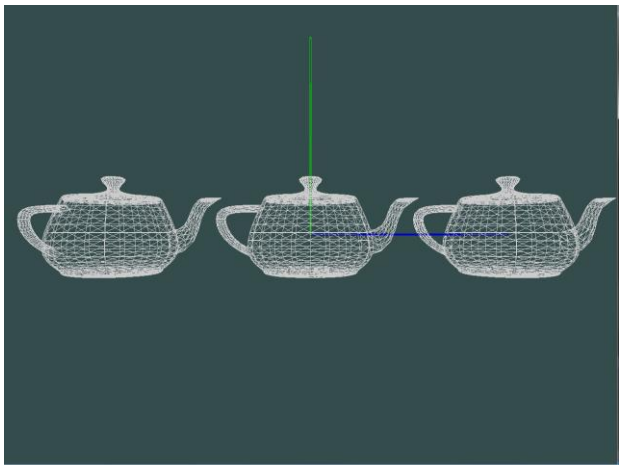

Figure 1. Default teapot position

The lines that you see define the axis of the object coordinate system. The blue line corresponds to the +xaxis, the green line to the +y-axis, and the red-line (not clearly seen in this image) to the -z-axis.

Task 1 of 4 (30 points)

Implement a simple mouse-based xy translation (panning) of the 3D object in the starter code. Use the middle mouse button.

For this purpose, create a variable that remembers the x and y coordinate when the middle mouse button was pressed. While it is pressed, continuously read the x and y coordinates of the mouse cursor and compute the differences to the coordinate where the mouse button was pressed. Use the differences to translate your 3D teapot along the x and y axes. Use the function glm::translate().

Verify that your panning motion works correctly by pressing the middle mouse button on top of your teapot and moving the mouse. The teapot should move in the direction of the mouse (while the middle mouse button is pressed).

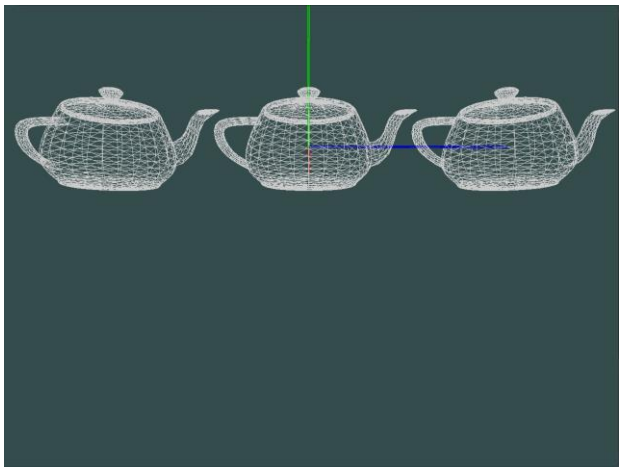

Figure 2. Teapot translation up

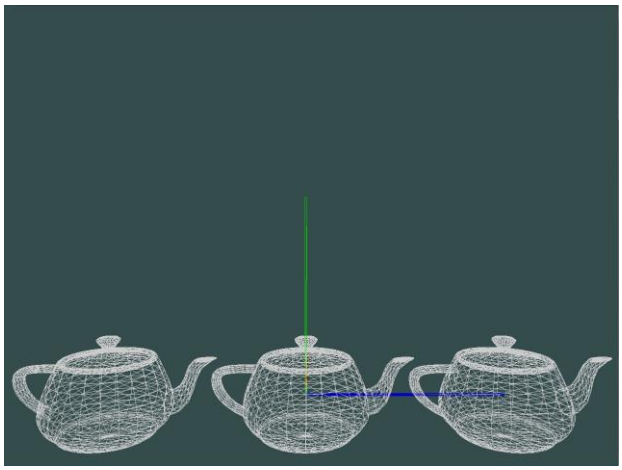

Figure 3. Teapot translation down

Task 2 of 4 (20 points)

Implement a simple mouse-based z translation of the 3D object in the starter code. Use the right mouse button.

Follow the instructions of task 1 to record the mouse-down pixel coordinates when the right button is pressed and continuously read it while the button is pressed. Calculate pixel differences in only the x mouse coordinate, i.e. horizontal mouse motion, and add a translation matrix to translate your teapot into the z dimension.

The teapot should move backward (into the screen, along -z) when you move the mouse to the right and forward (towards you, along z) when you move the mouse to the left.

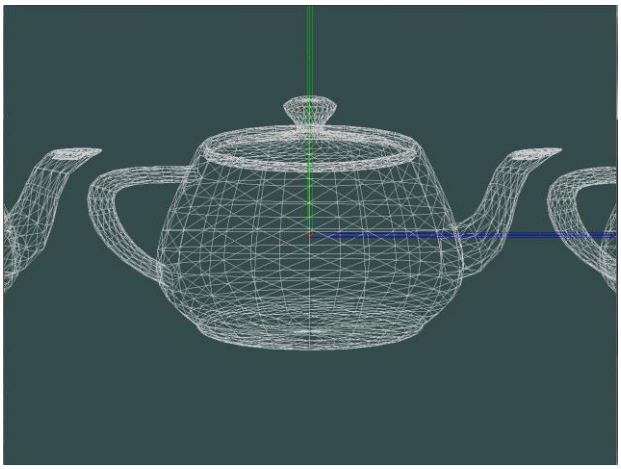

Figure 4. Teapot +z translation

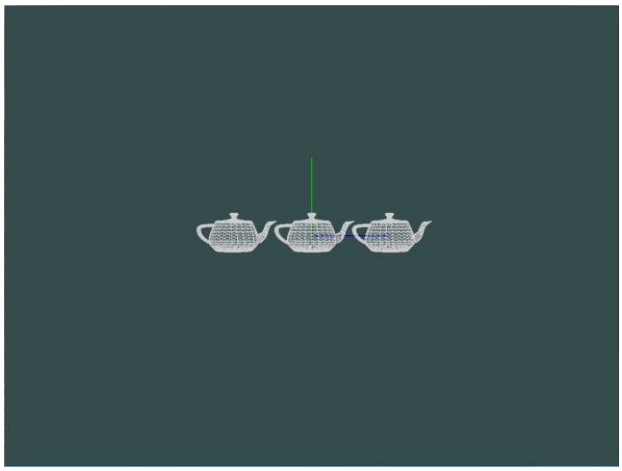

Figure 5. Teapot -z translation

Task 3 of 4 (35 points)

Implement a simple mouse-based rotation with the left mouse button. Follow the instructions of tasks 1 and 2 to record the mouse-down pixel coordinates when the left button is pressed and continuously read it while the button is pressed.

Convert pixel differences to degrees. Here, there is no strict conversion formula – just make sure your object neither rotates too fast nor too slow.

Convert x and y mouse motion to rotations around the y and x axis, respectively. So, moving your cursor left and right should result in a rotation around the y axis. Moving your cursor up and down should result in a rotation around the x axis. Implement the rotation using the glm:rotate function. Use two calls of this function, one for the rotation around the y axis and one for the x axis. When you move your mouse cursor diagonally, does the order of the two rotation operations (call to glm:rotate) matter?

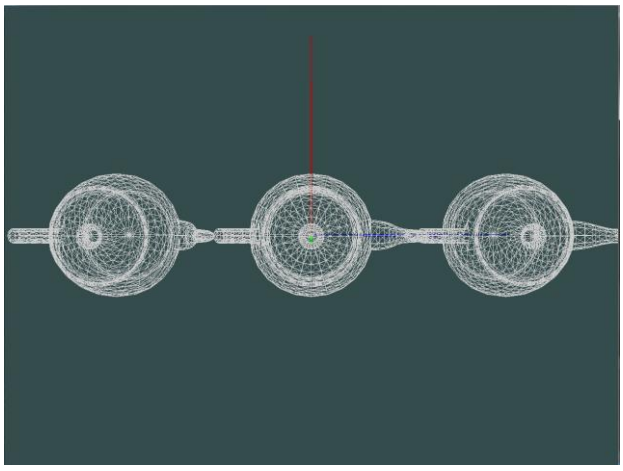

Figure 7. Rotate around x-axis.

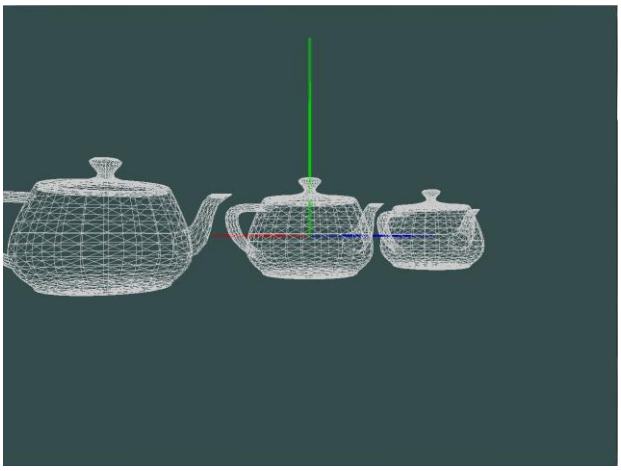

Figure 6. Rotate around y-axis.

Task 4 of 4 (15 points)

Extend your 3D scene by a few more objects.

Download at least 2 free (ideally interesting) 3D objects from this website: *www.turbosquid.com*. Look for .obj formatted models, with relatively few polygons (try not to exceed 10,000 polygons because it will take a while to load).

You can also use different website or other 3D models that you may already have. Turbosquid.com is a great resource for 3D models and may be useful for your project later on.

Use the model loader provided in your example program to load all three models (the two you downloaded and the provided teapot). All you have to do is place the .obj and .mtl (material file associated with .obj file) in the models directory and provide the path to the .obj file as the argument to the constructor for the Model class in the code. Render all three models and make sure to transform each of them individually. Take a look at how the teapot is loaded in and rendered. You will need to instantiate a different Model object for each model.

Scale the two downloaded models so that they are approximately the same size as the teapot. Then translate one to the left and one to the right. Make sure that the objects do not intersect one another.

Submission

Please add screenshots to the pdf of tasks 1 to 3 **AFTER** task 4 has been completed, meaning that there are **3 objects** in the scene.

Task 1:

- Add screenshot of default scene, with no transformations applied
- Add two screenshots of +/- y-translation similar to figures 2 and 3, one where objects are touching the touch edge of the window, and one where objects are touching the bottom edge
- Add two screenshots of +/- x-translation similar to y-translation screenshots, where objects touch left and right edges of the window

Task 2:

- Add two screenshots of zooming in and out, similar to figures 4 and 5

Task 3:

- Add two screenshots of rotation around x-axis, one where we see the top of the objects, and one where we see the bottom of the objects (similar to figure 6)
- Add two screenshots of rotation around y-axis, one where the objects are rotated counter clockwise by approximately 45°, and one where the objects are rotated clock-wise by approximately 45° (similar to figure 7)

Questions?

First, ~~Google~~ Baidu it! It's a good habit to use the internet to answer your question. For 99% of all question, the answer is easier found online than asking us. Also make sure to check out the helper resources we have added on the course webpage. If you can't figure it out this way, post your question on Piazza (preferably public question rather than a private one, so that everyone can benefit from the given answer) or send us an e-mail.