

COSC4370 Fall 20 HW2 - OpenGL

Sep 24, 2020
Due: Oct 8 at 23:59, 2020

1 Introduction

In this assignment, we will practice some of the basics of OpenGL.

2 Setup

We will be using the GLUT/freeglut library for our OpenGL code. If you are on Windows, you can find a freeglut 3.0.0 package for Visual Studio (MSVC) at <http://www.transmissionzero.co.uk/software/freeglut-devel/>. On OS X, GLUT should be installed by default. Finally, if you are on Linux and don't have GLUT already installed, you should be able to install it by running `sudo apt-get install freeglut3-dev` or `sudo yum install freeglut-devel` in a terminal. We have written the included Makefile so that the homework should compile with it on Linux or OS X. For Windows, you will need to create a project in Visual Studio rather than using the Makefile.

Please make sure you try compiling the starter code ASAP so that there are no last-minute problems with getting your environment setup correctly.

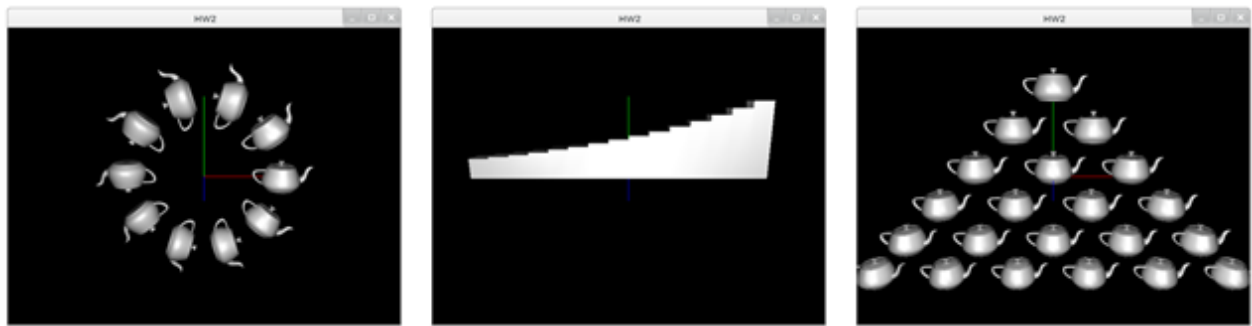
3 The Main Assignment

For this homework, you will be using your newfound OpenGL skills from class, as well as your artistic creativity, to create several 3D scenes with OpenGL.

- Complete function `display()` to determine your own Projection and Model-View matrix. You also need to setup a suitable camera position. Please read comments in `main.cpp` file.
- Write code to reproduce two of the following three images, you can pick up any two of them. Fill in function `problem1()` and `problem2()`
- Using similar techniques, create a scene of your own imagination. For example you can use primitives to create a hand as images shows below. Fill in function `problem3()`.
- Use keyboard to switch between three scenes and quit the program at any time by pressing 'q', 'Q', or the Escape key. Please complete function `keyboard()`.

4 Tips and Requirements

For each of the three reproductions, you should be able to create the image using only `glutSolidTeapot`, `glutSolidCube`, and OpenGL's transformation mechanisms like `glPushMatrix`, `glPopMatrix`, `glTranslatef`, etc. Note that you should not need any custom geometry, just the teapot and cube, to reproduce the images.



Your reproductions do not need to match exactly. However, please try to make them match the examples as closely as possible. We used nice numbers in the reference solutions, so if you find yourself using strange fractions etc. to reproduce the examples, you may be trying too hard!

For function `problem3()`, you could attempt to create a (very rough approximation of) an articulated hand:



5 Deliverables

Submit all deliverables to your Github repository.

- Code for generating each of your 3 scenes (`main.cpp`)
- Screenshots (preferably `.png`) for each of your 3 scenes

- Write a detail report in readme file of Github