CS 355 Lab #5: 3D Graphics with OpenGL

Overview

This lab introduces you to simple 3D graphics by having you set up the world-to-camera, projection (perspective or orthographic), and viewport transformations. In this lab, we will not use Jupyter notebooks. Rather, we will use command line Python to make a viewing window and the OpenGL package to do rendering.

OpenGL was originally in C, so you'll find the design is straight imperative (i.e. commands) rather than object-oriented. It also doesn't involve GUIs directly, though there have been GUI libraries written for it. You won't have event listeners but will instead poll the keyboard to see if keys are pressed.

In this lab, you will move around a *very* simple virtual world that consists of a wireframe model of a small house. (If you wish, you can play around with the model to render other scenes.) In this world, you can turn left or right in the xz plane of the world; move up or down in the y direction, or move forward, backward, left, or right *relative to the direction you are currently facing*. You should be able to switch the projection between perspective and orthographic.

When moving, you should move one unit in the world coordinates *relative to the direction you are currently facing*. When turning, you should turn one degree left or right.

User Interface

Your only interaction will be through the keyboard. Each of the following keys should perform the corresponding actions.

a	Move left
d	Move right
w	Move forward
s	Move backward
q	Turn left
e	Turn right
r	Move up
f	Move down
h	Return to the original "home" position and orientation
o	Switch to orthographic projection
p	Switch to perspective projection

Downloading OpenGL

You should already have OpenGL installed on your machine if you followed the **Getting Started** instructions for the first lab. If you still need the package, follow the directions below.

On Mac or Linux machines, run

```
$ conda install pyopengl
$ conda install pyopengl-accelerate
```

On Windows, you will need to download a current build for PyOpenGL and PyOpenGL-accelerate at http://www.lfd.uci.edu/ gohlke/pythonlibs/#pyopengl, then run

```
$ pip install pyopengL_version.whl
$ pip install pyopengL_accelerate_version.whl
```

To test if you have correctly installed OpenGL, simply run the lab file that is given. It should pop up with a black window display with nothing drawn in it. This is where you will do your rendering.

Implementation Notes

You are given a shell file for this program that takes care of some of the initial setup for OpenGL. This should allow you to foucs on rendering.

For this lab, the "model" in the model-view-controller sense is static. We provide a `drawHouse()` function that provides the location of the lines that draw the house. However, this function is useless without a definition of the rendering transformations needed to display it. You will need to implement these transformations using OpenGL commands.

You'll notice at the top of the file that the TA has explicitly imported the OpenGL commands you'll actually need for this lab. There's a *lot* more to OpenGL than what we're using, so let this import help guide you in what commands to use.

If you don't know how to use a particular command or what parameters are, go find the documentation on the internet. (It's not hard to find.)

A good place to start is http://www.glprogramming.com/red/index.html. Chapter 3 contains most of what you need to know for this lab.

Submitting Your Lab

Your code should be contained inside a single .py file. To submit the lab, simply submit this file through Learning Suite. If for some reason you used multiple files, zip these files together before submission. If you need to add any special instruction, you can add them there in the notes when you submit.

Rubric

This is tentative and may be adjusted up to or during grading, but it should give you a rough breakdown for partial credit. Please be aware that it may be hard to see enough to get partial credit unless a reasonable amount of the lab is complete.

- Correct perspective projection (30 points)
- Correct orthographic projection (20 points)
- Correct camera movement (30 points)
- Generally correct behavior otherwise (20 points)

  TOTAL: 100 points