CPE 476 WD40 for 3D programming for Professor Wood's program 1

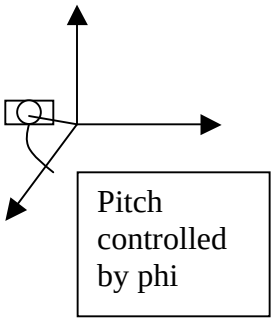
The purpose of this information is to let you start playing with some important tools for 3D game development. This exercise is not required but is a good start to working towards program 1. Consider doing the following tasks:

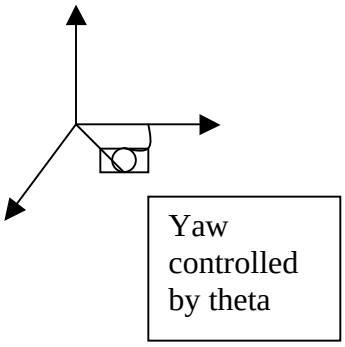
1) Create a ground plane for your game world (rendered in wireframe now is fine).
2) Enable 2D bitmap fonts in order to display text to the screen and display the frames per second for your program.
3) Enable a pitch and yaw camera with zoom and translate (using the gluLookAt command).

In addition, for visual richness for this initial program, I recommend starting with your code to display a 3D mesh. If you do not have this code, you can download it from:
http://www.csc.calpoly.edu/~zwood/teaching/csc476/material/
(The original C style file is called MeshParser_release3.cpp or there is a tar file called NewMeshParser, converted to a C++ style by a past student, using a class called BasicModel).
Example mesh files (of this format) can be found at:
http://www.csc.calpoly.edu/~zwood/teaching/csc471/data/
You must at very least have some objects appear on your grid plane (for example, if you are not displaying a mesh you must display 3 different solid and shaded glut primitives).

For **task 2**, I recommend using a variety of online code to assist you. For example, refer to:
http://www.lighthouse3d.com/opengl/glut/index.php3?bmpfont
for a tutorial on bitmapped fonts.
Refer to:
http://www.lighthouse3d.com/opengl/glut/index.php?fps
for an example program to calculate frames per second

For **task 3**, you need to write your own pitch and yaw camera controls, which include a zoom and translate mode. Use the mouse or you may also hook these commands up to keys from the keyboard.





Given a pitch and yaw angle, you can compute the eye location by:

x = radius*cos(phi)*cos(theta)
y = radius*sin(phi)
z = radius*cos(phi)*cos(90.0-theta)

You can control zoom by modifying radius.
You can compute phi and theta as some portion of the mouse's movement in the window in the vertical (phi) and horizontal (theta). Becareful as in C/C++ sin and cos expect radians!!!! And think about what theta and phi should start out as to start looking down the z axis.