Programming Assignment #2
3D Graphics in OpenGL
Due Thursday, October 25 by 11:59pm

(The grade for this assignment is out of 100 points, but you can obtain up to a maximum of 125 points by doing more optional components. Additional points beyond 100 will be applied to your final score for all programming assignments.)

Required Components: (40 pts)

(1) Display a 3-dimensional manifold object; that is, an object that encloses a volume (*i.e.* cube, sphere, tetrahedron, etc.). NOTE: You cannot use something like the GLUT helper functions (glutSolidSphere, etc.) to satisfy this requirement. (10 pts)

(2) Implement at least one of the following viewing modes. You may implement more than one. (Any additional modes would be counted toward credit for the optional components portion of the assignment.) (10 pts each)

- (1) Fix the camera on the z=0 plane, with the viewer looking down the -z axis. Allow the user to translate the camera within the z=0 plane. (that is, the user controls the location of the camera in the z=0 plane, but the direction remains -z)
- (2) Fix the location of the camera, but allow the user to control the view direction.
- (3) Fix the view direction of the camera and its x-y location. Allow the user to move the camera forward and back along the view direction, like zooming in and out.

(3) Display two 3-dimensional objects and have them rotate independently (about each object center) about different rotation axes. The animation can be user-controlled or entirely scripted, but it must animate (move without user input, such as with glutTimerFunc or glutIdleFunc in GLUT terminology), and both objects must be able to rotate simultaneously. (10 pts)

(4) Display two 3-dimensional objects, lit by a single point light source. Allow the user to toggle between flat shading and Gouraud shading. NOTE: For this to work, you need to use a model with defined normals. The GLUT helper functions (glutSolidSphere, glutSolidTeapot, etc) have defined normals, so I recommend using some of these for this portion of the assignment. (10 pts)

Optional Components: (60-85 pts)
NOTE: You MUST do at least 60 points worth of components from this section to achieve full credit on this assignment.

(1) Implement additional viewing modes:
    (1) Any of the modes discussed in part 2 above (10 pts each)
    (2) Define some point in the scene on which the camera will always center. The distance of the camera from that center point and the point itself define a sphere. Allow the user to move the camera along the surface of the sphere. The camera's look direction must always be toward the center point you choose. (20 pts)
(2) Load polygonal models defined in the TRI format and display them. NOTE: You must write your own file reader class/function; you cannot use a publicly available reader. Notes on this format are available online http://wwwx.cs.unc.edu/~jwendt/classes/COMP136/fileFormats.php; sample models are available here http://wwwx.cs.unc.edu/~jwendt/classes/COMP136/models/ (20 pts) (Credit to Jeremy Wendt for the linked resources.)
(3) Allow the user to interactively move the front and near clipping planes. (You probably want to define some minimum and/or maximum values to prevent math errors such as division-by-zero.) (15 pts)
(4) Extend part 3 from the required components above. This may mean using more objects, including other transformation types, or using more complicated transformations or scripted animation (0-25 points; negotiable)
(5) Expand a game from Programming Assignment 1 to 3D. The camera CANNOT be static; it should move in some logical way. (*i.e.* tied to the user's paddle in Pong, tied to the front of the spaceship in Asteroids, etc.) (10-50 pts; negotiable)
(6) Calculate the per-triangle normals of objects read in from a file. Use these to render the object lit and flat-shaded. (15 pts)