Assignment G5
3D Modeling and Viewing

Due Date: April 23

Purpose
There are several objectives for this project:

1. Create 3D models from scratch using techniques described in class (see below).
2. Use pre-defined 3D data sets that represent realistic models.
3. Set up multiple light sources.
4. Display the 3D data using Phong's lighting model to yield more realistic output.
5. Apply textures to surfaces.
6. Combine all of the above into a coherent scene.

Problem
Pixart was happy with your last project, and now wants you to generate realistic CG objects/scenes for their upcoming *What You See Ain't What You Get* series of movies. The first step in this is creating and/or rendering 3D objects. Your scene will be generated using some pre-defined 3D data as well as your own, custom objects. Textures can then be added to several surfaces. Finally, the scene will be rendered with Phong lighting to make it look as realistic as possible.

Input
There is **no** user-defined input data for this project. However, your program will use several files:

- at least two pre-defined 3D .obj files, as given on the course web page, one of which is a model of a chair. You can also use a file found on the interWeb or one you create with 3D modeling software, but show it to me before using it.
- at least one image used for a texture pattern.
- at least one image used for a texture image.

As in previous projects, the camera can be moved; use the numeric keypad to represent the desired camera movement (4-left, 6-right, 8-up, 2-down). The camera should have a circular motion about the scene; the distance of the camera to the scene should stay constant. Finally, 0 should represent zoom in, while 1 is zoom out.

The user can also choose various options from a standard (ugly) OpenGL menu:

- Toggle textures.
- Toggle wire-frame.
- Toggle point light source (see below).
- Switch between smooth (Phong) and flat shading.

- Quit.

Output

The output is simply a nice room scene, rendered with at least two light sources, in a pleasing default orientation. The user can then change the view of the scene by moving the camera or choosing menu options.
The scene **must** have the following properties:

- A custom-made table.
- At least two chairs by the table. These chairs should be made from the `chippchair.obj` file.
- At least one additional pre-defined, non-trivial object from a .obj file.
- The "Utah teapot" should be on the table.
- At least one *surface of revolution.* That is, one custom 3D model of a circular object. This should be generated with your code.
- Two or more "walls" or simple solid rectangular objects.
- A clear "floor" or "ground." Items should not just be floating in space.
- A "rug" on the floor with a tiled texture mapped onto it.
- At least one 3D model should have an image texture mapped onto it.
- At least one of the textures should be mapped onto a non-flat object; that is, the texture should cover multiple polygons.
- Objects should have material properties such that the effect of each of the lighting models is obvious.
- One of the light sources should be a point light source. If you make one surface shiny (the teapot?), then the viewer should see the effect of this light source easily.
- Colors are up to you, but should look as realistic as possible.

Specifics

- Once again, you are free to use any OpenGL functions.
- You must triangulate and/or carefully tessellate your custom objects to be assured that all polygons are planar.
- You must compute surface normals for each polygon (in the case of a custom-made object) and vertex so that the lighting models will work properly.
- You will need data structures to store your models. The data structure must store vertices and normals so that it is easier to render the objects. Do not hard-code hundreds (thousands?) of vertices!

Notes

The lighting parameters/materials are up to you. Try to make the rendering as realistic as possible, even if the scene is a fantasy. The better it looks, the better your grade.

Grab code liberally from my examples and from the textbook (the Code directory). We have done much of this in class, just at a smaller scale.

As before, bundle all of your files in a zip file, and email 'em over to me before 11:59:59 PM on the due date. The zip file name should include your last name and "G5" as in `gousieG5.zip`. If you use your own Makefile, include it in your zip file.

There is no reason anyone would want a computer in their home.
– Ken Olson (founder of Digital Equipment Corp., 1977)