Project G6

Due Date: May 7

Purpose

The purpose of this assignment is to:

1. Create models using techniques described in class (see below)
2. Use one or more models defined and stored in .obj data files
3. Use associated .mtl files that store material definitions
4. Construct a realistic scene
5. Use multiple light sources
6. Apply more textures to surfaces
7. Render the scene with ray tracing using POV-Ray, the Persistence of Vision Raytracer, available at povray.org

Problem

The problem is simple: Create as realistic a scene of a **recognizable location on the Wheaton campus** as possible using the POV-Ray ray tracing software.

See http://hof.povray.org for inspiration.

Input

The deliverable for this project is mainly a .pov file that contains the definition of all elements of your scene, plus any texture and other data files that are necessary to render the image. I will run POV-Ray with your input files to generate the scene.

Your scene (.pov file) must contain certain items:

- You must use at least two different texture images.
- You must have at least two light sources.
- You must have at least one "shiny" object.
- You must have at least one non-opaque object.
- At least one item must have some curvature. That is, a rectangular building built only with rectangular polygons is not sufficient.
- You may use any number of .obj objects, including any you find on the interweb. Several are listed on the course web page. It will take creativity to turn pre-made objects into something that looks like a Wheaton location. Alternatively, you will have to make one or more models from scratch or from glut objects.

In addition, you may want to use one or more material files (.mtl) to define what an object is made of.

Output

The output will be the ray-traced rendering of your .pov file definition and associated texture and material files:

- The scene should be as realistic as possible, with all elements clearly identifiable.
- There must be at least two light sources (the sun plus at least one artificial source) and the objects should be well lit and implemented with smooth shading; that is, try to make the lighting look as realistic as possible. This means you will have to include vertex normals.
- There should be multiple types of surfaces on the various objects, including a very shiny surface and one which is non-opaque. The surfaces should match the kind of object; that is, you should not have a super-shiny tree, for example.
- Good textures should be applied to the surfaces of appropriate objects in the scene.
- Although the scene is up to you, your grade will improve if
  - the scene better shows lighting effects (e.g., shadows or light coming through windows)
  - the scene is more realistic (e.g., a good model of a real building)
  - objects have texture maps that make them look real (e.g., bark on a tree trunk)

Notes

Although you are not submitting any C++ code (wow!!), you may have to write one or more programs or find code that will transform files from one format to another. For example, you will probably have to transform an .obj file to data that conforms to the .pov standard. You may need other (small) programs as well. In general, how you put your .pov file together is up to you!

A significant portion of the grade will depend on (a) if you implement and/or compute and/or otherwise define surface normals correctly on an arbitrary object, which in turn makes the lighting/shading look realistic, (b) how complex the surface is to which you apply a texture map, and (c) how nice/realistic your surface models are. The largest portion of the grade, however, is that POV-Ray runs with your .pov file and renders a scene with realistic lighting effects.

Deliverables include: your .pov file that defines your scene, texture files, any other necessary files (perhaps material data in .mtl format, for example), and a .png of your rendered scene. Both of these should be sent electronically in one archive file, as usual.

You've got to get the fundamentals down because otherwise the fancy stuff isn't going to work.
– Randy Paush, Carnegie Mellon CS Prof (1960-2008)