CSC 471: Program IV – Due Friday, November 18th at 11:55pm

Introduction: Your goal for this program is to create a simple world scene and allow the user to explore it. Your world must contain at least, a ground plane and two different types of models scattered throughout the world. At least one of these models should be hierarchically modeled using primitives and the other should be an obj file. The user is allowed to move through the world (by moving the camera). The user is allowed to move forward, backward, and side to side (relative to their current gaze). In addition, the user can rotate their view to look all round and up and down. **You are allowed the freedom of creativity in terms of creating and arranging models. You are encouraged to make a visually interesting world.**

This is an individual assignment. There are a number of tutorials that closely follow the requirements for this program. Make sure you implement and **understand your own solution**.

The specifics include:

- Your program will need to procedurally generate a scene. This scene will include at least:
  - a ground plane which extends well beyond the current view volume so that the user has a good deal of space to explore.
  - two different models. One models should be hierarchically modeled containing at least four different components and one model should be an obj mesh. These models should be spread through out your world across the ground plane. You must include at least 10 different instances of each of the two models (you may include more of one or the other or both, but you must include at least ten of each).
- Your program should allow the user to move around and explore the world. This will be controlled by moving the camera around in the scene. The camera can be translated using the following keyboard events:
  - "a" = move to the left of the current gaze (this should look like a translation not a rotation).
  - "d"= move to the right of the current gaze (this should look like a translation not a rotation).
  - "w"= move forward in the scene along the current gaze.
  - "s"=move backward in the scene along the current gaze.
- In addition the view can be rotated to allow the user to look around the world. Rotation of the view will be controlled using the mouse. A movement to the right with the mouse should allow the user to rotate their view to look to the left – likewise with a movement to the left. The user should be able to spin their view 360 degrees around. A mouse movement down should allow the user to look up while a movement up should allow the user to look down. Constrain the vertical movement of the view such that the user can only move a total of 160 degrees (80 degrees up and 80 down). Once the view has been rotated, all camera translations should be relative to this new view! This should allow the user to completely explore your world. You are allowed freedom to choose the exact speed of the camera motion for rotation and translation but it must be reasonable! That means a user must be able to feel like they can explore the world (i.e. not wait for lag or become confused by rapid motion).
- **You do not need to include any kind of collision detection.** This means that a user can walk through your models. You are allowed to include collision detection if you choose to (if so document this in your readme).

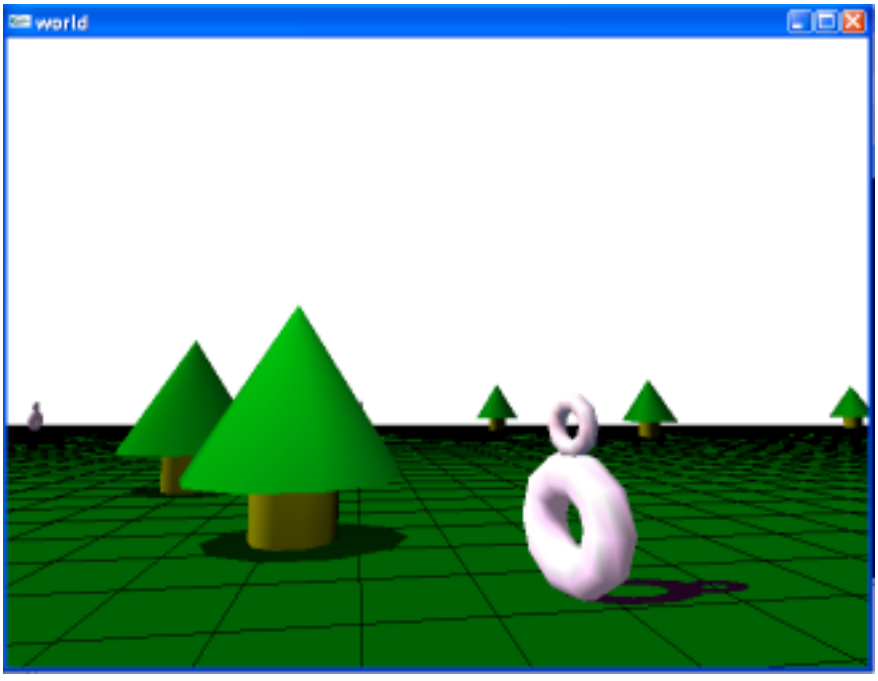
- Your program should only be rendered in perspective viewing.
- **Your program should include lighting (models should be correctly shaded).** You need to pick a lighting situation that you would like to emulate and are required to provide a sample image of the lighting situation you are trying to simulate (this could be a time of day – thus using directional lighting- or an interior scene. Note that you likely may need to use more then one light (ie many small lights) or a particular type of light (spot light)). You should include at least three different new materials on your models and you must include pixel shaded phong shading – please choose materials and lighting to show this off).

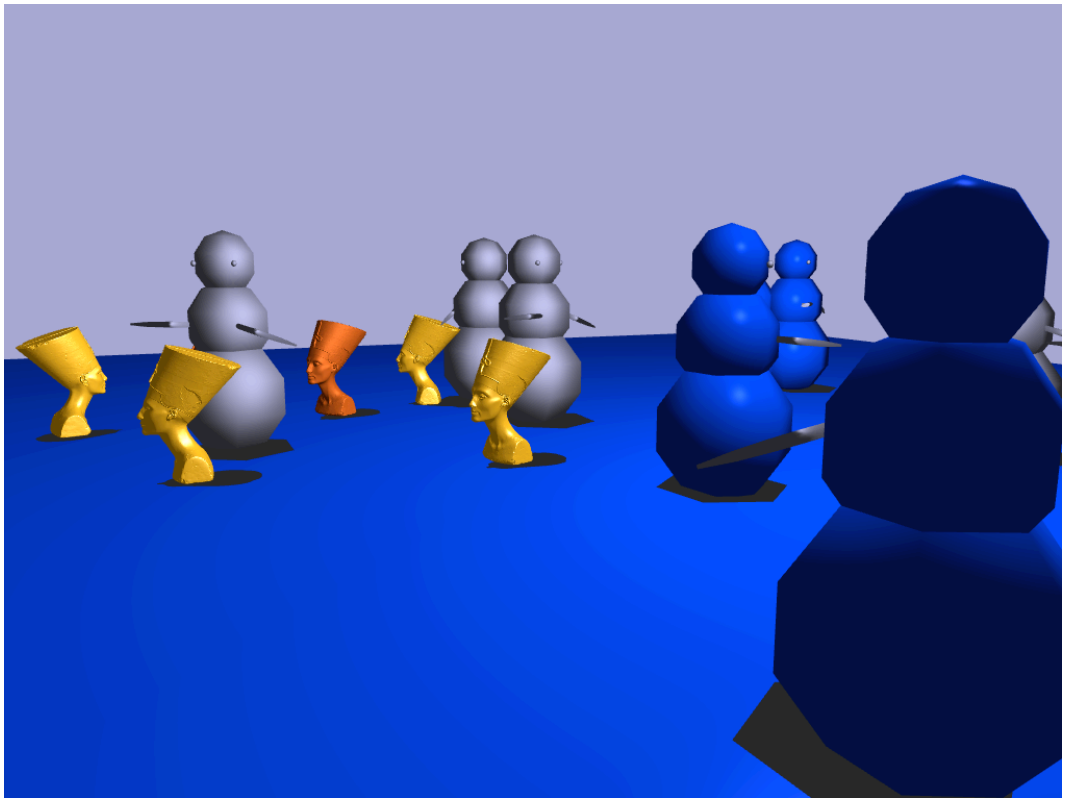
Please feel free to be creative and include as much detail as you'd like in this program, extra credit points will be awarded for extras, however, note that the bulk of the program points are for the basics so get those working first. But do feel free to include more like complex models, animating models, texture mapping, walls, fake shadows, collision detection, fancy shaders, etc.

Point break down:

20 pts for correct camera rotation
25 pts for correct camera translation
20 pts model creation and placement in the world
15 pts for simulated lighting condition
10 pts for general world construction (ie creative, interesting, good materials, etc.)
10 pts for general (code, readme, usability, etc.)

An example world with user interaction will be demonstrated in class, here are sample screen shots (including optional fake shadows – this is lighting via directional lighting light the sun – the shadows position matches the light direction):





Two views of the camera moving through a different world…