SMALL-E - Creating the Scene

CS 475/CS 675: Computer Graphics - Assignment 2, Part 2

Due Date: 17/10/2011

1 The Assignment

This assignment is aimed at producing an animated (very) short film by the end of the course. There are many steps to making an animation that involve a combination of aesthetic and technical skills. In this course we will learn many of these basic technical aspects. The aesthetic quality of the final result is, however, dependent on how much thought/effort you put into it.

This is the second part of this assignment and it deals with lighting and texturing the Small-E model you created in the first part, and creating a room scene.

TO DO:

1. Texture the Small-E model you built in the previous part of the assignment. You must at least texture map the body, arms and wheel. Textures can be any suitable images downloaded from the net. To figure out this out, read Chapter 9 of the OpenGL Programming Guide at http://www.glprogramming.com/red/chapter09.html
2. Light the model with two dim *directional* lights. To figure out this out, read Chapter 5 of the OpenGL Programming Guide at http://www.glprogramming.com/red/chapter05.html.
3. The scene you have to generate is a room with furniture - you must at least model a chair and a table. Apply proper lighting and texture to the objects, room walls and floor. Place Small-E in this room.
4. The shape, size, color of the room and the objects is your choice.
5. Figure out four camera locations in the room - one each from which the chair, the table and Small-E are in the center of the scene. The fourth is a camera placed in front and top of Small-E's head, showing the the scene from his point of view.
6. All your keyboard bindings from the previous assignment used to move Small-E should still work.

7. Bonus: Model a light fixture (bulb or tubelight) on one of the walls and make the light fixture glow. Model any kind of light switch that can turn the light on/off - the lighting in the room should change appropriately.

DO NOT:

1. Compile and produce an *a.out*. Learn how to use a Makefile.

2. Write code for non-inlined functions in header files.

3. Write untidy code - you will lose marks if you sprinkle your code with global variables, write code that is difficult to read and is unindented or write code that is not properly structured into objects, classes and files. Only the GLUT callbacks need not be encapsulated in any class and global variables are permitted in the main program file (like in toylogo.cpp from the previous assignment).

4. Make a model that looks exactly similar to some other group's model from the class - both groups will then lose marks. This is an assignment where you have enough chance to show that all of you are original thinkers - please do not hesitate to be creative. So you are free to discuss solution strategies with your classmates but make sure that your code and your models are different.

GIVEN:

1. In order to perform texture mapping you will have to read in images in your program. Source code for reading raw PPM image files into an array. This array can then be passed on to OpenGL for use with texture mapping. An image in any format may be converted to a raw PPM using the "convert" command form ImageMagick on the command line or by using the GIMP. The PPM image format is explained in the source file.

2. OpenGL Programming Guide and other tutorials - be sure to check out relevant chapters of the programming guide to figure out how to do texture mapping and lighting in OpenGL.

MARKING:

- The four camera locations should be attached to the keys $1, 2, 3, 4$ and one should be able to toggle the view between them : 20 marks

- Texturing the body, arms and wheels of Small-E correctly : 30 marks

- Lighting the scene properly with 2 directional lights : 10 marks

- Modelling the Room (walls, floor and ceiling), Chair and Table : 10+10+10 = 30 marks
- Texturing and coloring the scene objects : 10 marks
- Bonus (Glowing Light that can be switched on/off): 20 Marks (No Bonus marks will be given unless everything else is perfect - so complete the rest of the assignment before attempting the bonus!)
- Total : 90 marks + 10 Marks
- Deduction - I am expecting everybody to write properly formatted, indented and structured code from now on. Untidy code will be penalized.
- Late submission will follow a policy of graceful degradation with a 25% penalty for each day's delay (i.e., you get zero marks if the assignment is more than three days late after the due date.)

TO SUBMIT:

1. A Tar-Gzipped archive of the complete source code (and only source code). It should compile using the given Makefile on any Ubuntu system.
2. A link to a html report page on the assignment that should contain some details about what you implemented and images of some the results that you generated. Put the link in a README file in the archive you submit. Also, include all the keyboard bindings in your code that move the various parts of the robot.
3. The submission will be through the submission portal.