CS6250 Topics in Adv. Comp. Graphics

Programming Assignment #2 — Computer Science Department
Local Reflection Models — Bowling Green State University
Fall 2009

30 points
Due date: Friday, October 2, 2009
Time: Electronic submission of program files due by 2:00pm on the due date
Printout (i.e., hardcopy) is due at the beginning of the class on the due date (must be stapled)

In this programming assignment, we will consider the local reflection models. The program must be developed in C/C++ using the OpenGL graphics library with GLUT for maximum portability; do not use any other windowing/graphics package. No other libraries can be used, unless you received approval from your instructor.

Assignment: For this assignment, you will implement a program that can render an octahedron structure with can be rotated in space. Your program will support rendering of this octahedron using several reflection models. You will render the octahedron in a face-by-face manner, using hand computation of color for small sets of pixels that make up each face. An easy way to do this is to recursively divide each face into small sub-faces of size about 4 pixels in diameter and determine a uniform color for each sub-face. You can use OpenGL functions to render each sub-face as a 3D facet on the screen (i.e., you can let OpenGL determine which faces are back faces and let it do the projection for you). You can divide a triangle into 4 triangles by connecting the midpoints of the original triangle's sides.

The Octahedron: Octahedra are fairly simple to generate. The coordinates of the vertices of an octahedron centered at the origin are (±50, 0, 0), (0, ±50, 0), (0, 0, ±50). Each face is a triangle. The cross-section of the octahedron through its origin is a square. Your program should model the octahedron as a collection of 8 triangles.

The Viewing Parameters: You should view the octahedron in an orthographic view volume of size 200×200×200 that is centered at the origin. The projection plane in the viewing scenario is the x-y plane. The viewer is located on the positive z side of the view plane and is looking at a view window that has "up" in the positive y direction. The eye point could be viewed as being at (0, 0, 100).

Shading Information: There is one point light source, positioned at the eye point position. This point light source has ambient, diffuse, and specular components. The light source is white.

The octahedron is painted with very glossy bright purple paint (no green, max blue and red components). This paint is a good, but not pure, specular reflector. The specular exponent that is to be used is 3. Use a specular coefficient of a fairly high value– about 0.7.

User Interface: The following keystrokes or mouse input should be supported. No "key" of valid commands should be printed on the screen; however, a list of the valid commands should be printed to stdout.

LEFT MOUSE BUTTON : when clicked, the intensity of the light is increased by 0.2 unit. Assume that minimum intensity is 0 and the maximum is 1.
RIGHT MOUSE BUTTON : when clicked, the intensity of the light is decreased by 0.2 unit. Assume that minimum intensity is 0 and the maximum is 1.
'd' : toggles between ambient and diffuse reflection models.

's' : toggles specular reflection on and off.
'R' : rotates the octahedron by 10 degrees about its vertical axis (which passes through its centroid).
'Q' : when the Q button is pressed, the program should exit.

Lighting: When the light is at maximum intensity, it should be bright white. It should be of the maximum viewable intensity in OpenGL (corresponding to (R, G, B) value of (1, 1, 1)). Other intensities are simple linear scales of the maximal intensity value. The default light intensity is 0.6.

Extra point: A couple of options could be included for extra credit consideration. If any of these features are implemented, they MUST be clearly documented at the TOP of the program listing.

1. Allow the "L" key to toggle on and off a second light. This light will be positioned at (60, 0, 0).
2. Allow the user to change the specular reflection exponent in one unit increments (decrements). The "I" key will increase the exponent and the "i" key will decrease it. Your program must display the current value of the exponents at all times.

Structure and Documentation Note: The program should have a modular design and a reasonable amount of documentation. This means that major/important features, functions/methods, and data structures should be very clearly documented. Remember, you can reduce some documentation work that might otherwise be needed by careful choice of variable and function/method names. Object-oriented approaches must be very clearly described; careful choice of object names is not sufficient for other readers of your program to understand its structure. Programs are also expected to be reasonably efficient.
The initial comment section of the program should include a concise description of the architecture of your program. Your goal in the initial comment section should include providing a concise description of the program's methodology (including any critical functions and data structures) to the grader. Anyone reading the initial comment section should be able to very quickly understand the structure of the program.

Building and submitting the OpenGL program: Your program must include and use the *cs_456_250_setup.h* header file that is on the BlackBoard. Unless you are doing animation, you should use that file exactly as it is. If you want to use an object-based approach, you are free to minimally modify the *cs_456_250_setup.h* to work with your object-based framework. Please, no use of STL for C++ program.

You need to hand-deliver a print-out of your program's complete source code. *(Please have a separate cover page.)* In addition, you need to put your complete source code in the directory specified below on the CS server. In particular, create a Visual Studio Project under a work directory named **cs6250_prog2_yourlastname** (e.g., if you name is Tiger Woods, the directory should be named as **cs6250_prog2_woods**.) and then submit the entire folder. You are restricted to NO MORE THAN **TWO** header files (including *cs_456_250_setup.h*) and **ONE** C/C++ file.

Please check that your program works on the departmental Windows machine.

- Turn-in your softcopy of program for grading:
  1. Go to **Start → CS Classes** and login as **bgsulabs** for both username and password.
  2. Click on **lee\cs6250_turn_in**
  3. Drag your **cs6250_prog2_yourlastname** folder and drop it into the **"cs6250_turn_in"** folder**.**

  (For more details to access the CS Classes, see http://www.bgsu.edu/departments/compsci/docs/cs-classes.html)