CS6250 Topics in Adv. Comp. Graphics

Programming Assignment #3
Ray Tracing

Computer Science Department
Bowling Green State University
Fall 2009

30 points
Due date: Friday, November 6, 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)

The goal of this assignment is to learn about ray tracing technique in graphics. 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: Ray tracing is one of the fundamental techniques in computer graphics and has been used to create beautiful images of complex scenes. For this assignment, your task is to generate some simple images of a scene composed of spherical objects.

The program need only trace the rays as far as their THIRD intersection with an object. If you like, you are free to trace the rays to a depth of four (i.e., allow a third reflection) and to then use a recursive or non-recursive solution, although I encourage you to try to improve its efficiency.

Details: The program should print a message to the terminal window when it starts. This message should say something like "Press S to begin simulation. Press X to exit when you are finished viewing the image." Your program should compute its image and display it when the S is pressed in the drawing canvas. When the user presses the 'X' key in the drawing canvas, the program should exit.

The scene will contain 7 sphere-shaped planets. One sphere is of radius 6 and it centered at (0, 0, 20). It is polished silver in color. Another sphere is of radius 6 and is centered at (15, 15, 20). It is red. The third sphere is of radius 7, centered at (78, 52, 70). It is green. The fourth sphere is of radius 8, centered at (48, 51, 68). It is blue. The fifth sphere is of radius 4, centered at (50, 50, 40). It is yellow. The sixth sphere is of radius 10, centered at (-9, 11, 11) and is orange. The seventh sphere is of radius 2, centered at (3, 11, 11). It is white.

The scene is size 100 by 100 by 100. The viewer is at the origin, although you will render the scene with a parallel projection. The projection plane is the $z = 0$ plane and the viewing region is centered at (50, 50, 0). The "front" boundary of the scene is coincident with the $z = 0$ plane. The viewing plane's extents are 100 units by 100 units, and these extents coincide with the sides of the viewing region. The viewing plane will be rendered as a 200 by 200 pixel canvas on the screen, with the world origin mapped to the screen canvas origin.

All spheres are about 50% reflective. The polished one is highly (about 70%) reflective. There is a very low level of ambient light in the scene; the diffuse and specular components make up most of the illumination.

A single direct light source should be used. It should be positioned 10 units directly behind the viewer. The light passes "through" the viewer. You should support a "L" option which toggles the light between its default position and a position to the right 20 units from its default position. When the light movement option is pressed, the scene should be regenerated for the next position. You cannot store the scene in an off-screen buffer and double-buffer between the views! The light movement option will be worth about 3~4 points on the assignment. You might implement this feature last.

You are free to choose reasonable parameters for the undefined parameters. **Your program listing (or an accompanying short report) should discuss how you chose the parameters and what the parameter values are.**

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 viewer to change the view point
2. Another extra feature would be to assume that the spheres exist in a room that has a mirror along part of one wall. The mirror should be approximately 80~90% reflective and should be a "clear" color (like glass or very highly polished silver). Non-mirror areas on the walls should be strictly non-reflective (although you might decide to "paint" them a dull color).

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_prog3_yourlastname** (e.g., if you name is Tiger Woods, the directory should be named as **cs6250_prog3_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_prog3_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)