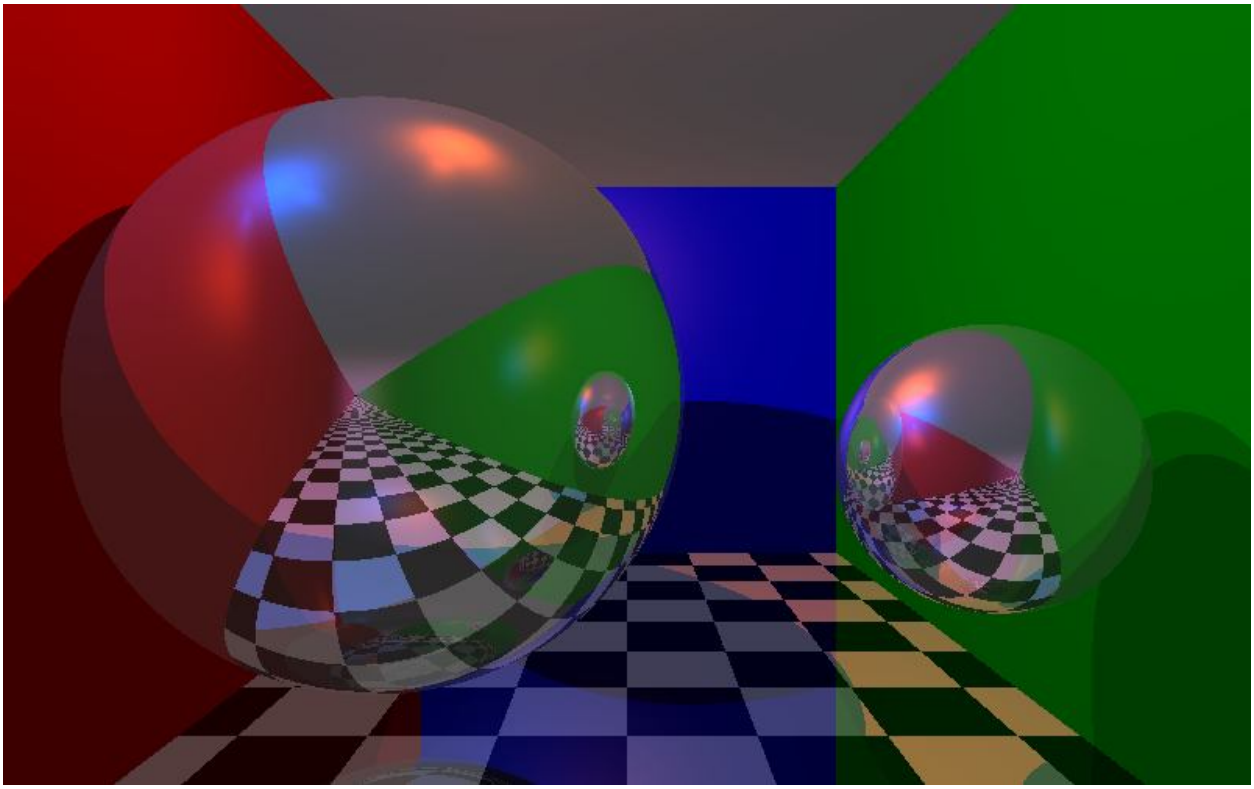
CS 488 – Fall 2013

Project 3 – Lighting & Hidden Surface Removal

Due Date: Tuesday, 12/10/13 11:59pm



Summary:

The goal of this project is to render a basic scene composing of two silver spheres, three colorful walls and one checkboard as floor. Your final scene should look similar to but not exactly the same as the picture shown above. For this project, you are given the option to render the scene based on two illumination models.

For the first option, you must render your scene based on the Gouraud or Phong illumination model while also incorporating the Warnock algorithm for hidden surface removal (going down to the level of 4 sub-pixels for antialiasing). The spheres must be constructed using 512 polygonal facets.

For the second option, you are to ray trace the same scene. Here the spheres in your scene should be modeled based on the equation of a sphere. For each ray emanating from the center of projection and passing through the center of a pixel on the screen that hits an object you should calculate the intensity of the pixel using the equation: Ip = Ia + Id + Is + R + T; where Ia: ambient lighting, Id: diffuse lighting, Is: specular lighting, R the recursive reflective light component, and T the recursive trans-missive component. Remember, you do not need to adopt a hidden surface algorithm or a perspective projection matrix to the ray tracing option since the ray tracing algorithm intrinsically provides perspective, lighting and hidden surface detail to your scene.

Compilation/Programming-Style

The program must be written using C/C++ using the GLUT library. You can program using any operating system (Windows, Linux and Mac), and you can use any IDE (Visual Studio, Eclipse, Xcode and so on). However, your final product must compile on the CS Computer Labs computers in room 2254 SEL. You can use your CS account on these machines to confirm that this requirement is fulfilled. Programming style is essential; you should document your code so that I can understand the important steps/algorithms in your code. Use block comments to clarify a section of code and line comments to explain a technical detail on the implementation. You should also format and organize your code in a readable fashion. Without documentation or ReadMe file, you will lose 10% of your final grade as penalty.

Submission

You will submit your project on Piazza. Please submit only ONE archive file (zip or tar) which includes the source code, possible header files, ReadMe file.

ReadMe

You MUST include a readme file which summarized what you did. In this file you should also specify the algorithm(s) you used and document any other information that is relevant to my understanding your algorithm. The algorithms you explain in the readme file should coordinate with your comments within the program.

Grading

This programming assignment is worth 7% of the final grade (7 points). The maximum value possible for an assignment will be reduced if handed-in late. If the assignment is turned in up to 24 hours late this maximum value is reduced by a 50% penalty (3.5 points). If an assignment is turned in up to 48 hours late this maximum value will be reduced by a 100% penalty (0 points); this also means that a program not 100% correct can receive a value less than zero but not less than -7 points on this assignment. An assignment turned in more than 48 hours late or not turned in at all will result in student losing an additional 100% loss (that means you get a -7 points on this assignment).

Grading is based on the stated requirements (about 2 points for each sphere, 3 point for the walls and floor), including comments, readme and programming style. A 100% working program handed-in on time does not necessarily receive full credit. A non-working program handed-in on time does not necessarily receive 0 points. Failing to provide sufficient cogent comments and/or a poor programming style will result in a penalization.

Extra Credit

Extra credit for this project will be a maximum of **two points** if you choose the Gouraud or Phong shading. One point will be provided if you can correctly incorporate Z-buffering hidden surface algorithm to the Warnock algorithm. One point will be provided if you can implement both Gouraud and Phong shading (use the ‘t’ key to toggle between the two).

Extra credit for this project will be a maximum of **one point** if you choose the ray tracing option, and apply the reflective lighting component to the level of three bounces.

You will also get **two points** if you can draw additional objects (one point for a cylinder and one for a cone).

Thus, the most you can earn from this project is 11 pts.