CMPU 378: Computer Graphics
Programming Assignment 4: Ray Tracing

In this assignment you will implement a recursive ray-tracing program that demonstrates techniques for analyzing scenes in which multiple objects interact with one or more light sources. Your program must be capable of analyzing scenes composed of spheres and infinite planes. A description of a format for files describing such scenes will be found among the files for this project along with several example scene files. You should use these example scene files to test and debug your program.

Your program must clearly demonstrate each of the following effects: elimination of hidden surfaces; diffuse reflection of light coming directly from a light source; specular reflection of light coming directly from a light source; a shadow cast by an opaque object on the surface of another object; reflection by one object of light coming from another object; refraction of light passing through a transparent object.

You should develop your program in a incremental manner. Start by implementing visible surface ray tracing for a simple scene with two objects. Next add a capability for modeling ambient light, directional light sources, and diffuse and specular reflection. Then add a capability for identifying shadows cast by opaque objects. All of these capabilities can be implemented without doing any recursive ray tracing. At this point you should make your program recursive in order to capture the effects of reflection and refraction.

During the debugging phase, you should use a low resolution raster and carry out the recursive ray-tracing process to a shallow depth. Once the basic mechanism appears to be working correctly, add more objects to your scenes and arrange them to obtain the effects described above. Finally, when the program and scenes appear to be correct, increase the raster resolution and depth of recursive ray tracing in order to generate the final versions of your images.

You may not use any of the packages Glut, Glu and OpenGL in your program. Your program should take the scene file name (e.g., "scene.01") as an argument. It should output images as files in **`ppm`** format, which can be viewed using Adobe Photoshop. The output file name should be the input file name with ".ppm" appended (e.g., "scene.01.ppm"). A brief description of **`ppm`** format is available among the files for this project.

Extra Credit: You many do any or all of the following things for extra credit. Each item is worth an additional 5% of grade.

- Extend the scene file format and enhance your program to include clipping planes.
- Extend the scene file format and enhance your program to render scenes composed of arbitrary quadric surfaces.
- Extend the scene file format and enhance your program to render scenes with texture maps drawn from **ppm** files placed on planar surfaces.