CSCD18 Computer Graphics, Fall 2007
Assignment 3

Part A Programming: Due at the drop-box and online on Fri, Nov. 23, 2007 at 11:59pm [50 marks]

Collaboration policy: You must work alone and submit your own work. Feel free to discuss the programming assignment with others, but you must write and submit your own code and report.

Part A [50 marks in total]

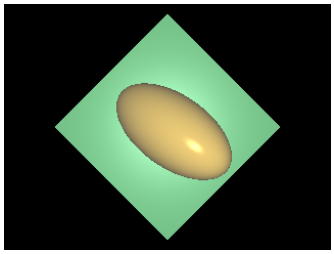
There are two parts to this assignment, a written part and a programming part. This is a description of the programming part. The written (theory) part will be available on the course website early next week.

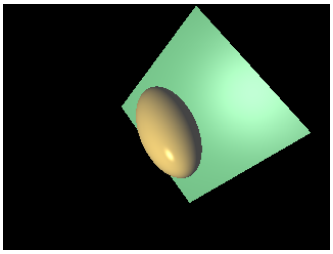
In the first programming part, you task is to implement parts of a basic ray-tracer. You are asked to render a simple scene using ray casting and local shading. We will provide starter code (see the course website) that sets up a scene comprising an ellipsoid and a plane, being illuminated by a point light source. Your job is to render the scene by implementing code fragments for object intersections and Phong shading.

To demonstrate the working of your program, you should generate three different types of renderings:

1. a scene signature where each pixel shows a unique colour identifier for the first object hit (or background). This gives you an impression of the relative positions of the camera and the objects.
2. a rendered scene with only the diffuse and ambient components of the Phong model.
3. a rendered scene with all three terms of the Phong model.

In addition to specifying your scene, the starter code has been written to generate two distinct views (images) every time you run it. Thus, for each type of rendering above you will generate two views, comprising a total of six images. In your electronic submission, please call these image files `sig1`, `sig2`, `diffuse1`, `diffuse2`, `phong1` and `phong2`. (You should also include a `readme.txt` with your submission to explain what the marker will see and what the relevant files are, as the file extensions will depend on the image file format used, which should be one of the well-known BMP, TIFF, PNG, or PPM formats.) Below are two images of what the Phong rendered scene should look like.





Finally, ray-tracing is compute-intensive, so debug your code on small images, e.g., 200x200, and debug ray intersections using the scene signature.

Also, when writing your code, keep in mind that you will need to extend it in various ways in Assignment 4, hence make sure your implementation is modular and re-usable.

Marking:

As usual, the course policy concerning extensions and late assignments are given on the course web site. Do not hand in hardcopy of your code; the electronic copy suffices. Also hand in a concise reports in the drop box. The report should be a well-structured written/diagramatic explanation of your software design and the element of your 3D scene(s) and animation (if any). The description should be a clear and concise guide to the concepts and properties of the images you generated, *not* a simple documentation of the code. In addition to correctness, you *will* also be marked on the clarity and quality of your writing. You should make specific reference to your generated images as you explain the key properties of your implementation, as well as any potential failure modes you identify. You should make reference to any interesting or unexplained aspects of your renderings handed in.