COSC342 Assignment 2

Due: 5pm, 22nd May 2017*

Building a ray tracer

You are asked to write (parts of) a ray tracer. Skeleton code is available in `/home/cshome/coursework/342/pickup/Assignment2/Assignment2OptionA/`

1. Starting with the skeleton program that we have provided, write and test routines that ray trace and shade the objects in the scene, meeting the requirements below. The skeleton program defines the required user interface, reads scene information from a file, and sets up many useful data structures. The format of the scene files and the functions and methods provided in the skeleton code are documented online.

2. Test your program with our sample files (within the `TestScenes` directory) and with scenes of your own design.

3. Write a short report, describing how your program works and how you tested it. Describe any known flaws in your program.

4. Use the `submit342` script to provide us with a directory that contains:

   - Code that we can compile and run under the MacOS environment in the CS labs using a `Makefile` or `XCode` project that you provide.
   - A sample image file, and the scene description file you used to generate it, which shows off the capabilities of your ray tracer.
   - A short report as a plain text file. (This should usually be about a page in length, and not be more than two pages long.)

Requirements for your ray tracer

We require that you complete the following tasks within the structure of the skeleton code provided.

1. The skeleton provides only ambient lighting. You should add Lambertian diffuse and Phong specular illumination.

2. Implement shadows and mirror reflections.

*The default late penalties apply.

3. Implement the rendering of cylinders and cones. (See the code documentation for more detail.)

4. Add support for a spotlight: these light sources diminish with distance and only cast light within a cone. (See the code documentation for more detail.)

Note that the scene file parser already recognises syntax relevant to the above tasks. The skeleton code is designed to provide you with implementation hints, and to allow you to focus on the graphics implementation required. See the To Do list in the documentation for information about which methods you need to update.

Marking scheme

3 marks	Report, with an emphasis on testing.
3 marks	Your sample scene image and associated scene description file.
2 marks	Diffuse (Lambertian) shading.
2 marks	Specular (Phong) shading.
2 marks	Shadows cast by objects.
2 marks	Mirror reflections.
2 marks	Correct rendering of cylinders.
2 marks	Correct rendering of cones.
2 marks	Correct lighting from spotlights.

Note that many of the components are independent. For example, you could complete diffuse shading, shadows, and cone rendering without specular reflection, CSG intersection, or mirror reflections. However, it may be difficult to test if your object normals are correct with just ambient lighting.

Final notes

Remember this is an assignment in graphics, not in production programming. You will not get any extra credit for extravagant solutions (although we may well smile while marking them). High marks will be given for a well explained solution that is easy to follow. Programs must be commented of course, but no more than necessary for us to be able to read them.

You may discuss conceptual issues relating to this assignment with others, but all the work you hand in must be your own, except for the parts of the given skeleton program.