Mathematics Assignment #3

Rendering a Triangle Based Model

Start *date: Thursday 2nd February 202*
Due Date: Thursday 19th January 2012

Overview

The objective of this assignment is to create an application to render a 3d triangle based model.

The assignment includes the following sections:

1. Create a scene description describing the camera, lights and a set of objects and associated material properties.
2. Implement an illumination model which includes ambient, diffuse, and specular terms for each light source as well as spherical and cubic reflection map.
3. Demonstrate multiple directional and point light sources, i.e. no area lights.
4. Your renderer must provide for transparency
5. Write the image data to disk.
6. Display the rendered – and rendering – image
7. Your Program must allow a user to:
   - Select a view position and view direction
   - "Render"
   - Interrupt and "Cancel" a render
   - "Save" the rendered image
   - "Exit" the program

This is a single person project.

Technical Specification

1. This is NOT a real-time project
2. The implementation will be based on the supplied DirectX framework RTVS_Lite.

Software requirements

1. It will be implemented in C++ using the supplied DirectX framework RTVS_Lite.
2. Your project should be created in Microsoft Visual Studio 2008 or 2010.
3. There must be a clear distinction between your application framework and your application (i.e. between those parts of your application that provide 'housekeeping', and those that implement your simulation).
4. You must include a complete working example in your submission (see below).
5. You must provide all dependent files (see below)

6. You must ensure that the Lecturer can compile and run your application on the Lecturer's PC prior to final hand-in (see hand in notes at beginning of this document).

Delivery and Marking

There are nine parts worth 5, 15, 15, 10, 5, 10, 10, 10, 5, 5, and 10 marks respectively:

1. Create a scene description describing the camera, lights, one or more triangle based objects and associated material properties (5).
2. Render the scene to simulate the image formed by a pin-hole camera. (15)
3. Implement an illumination model which includes ambient, diffuse, and specular terms for each material and light source. (15)
4. Demonstrate the difference between Blinn and Phong shading. (10)
5. Demonstrate multiple directional and point light sources (i.e. no area lights). (5)
6. Your renderer must provide for transparency. (10).
7. Your renderer must demonstrate anti-aliasing. (10)
8. Write the image data to disk. (10)
9. Display the rendered – and rendering (progressively displayed) – image (5).
10. Your Program must allow a user to interact with the scene description (e.g. switch lights on and off) and provide for other housekeeping functions. (5)
11. Write a description (name it "*renderer.doc*" and be 800 words or more) of your project detailing your objectives, data classes and other structures (10).

Please note that the marks awarded will not depend solely on satisfying the implementation requirements detailed above, but also on the elegance of your solution, the techniques used, the speed of your application, quality of code documentation and the quality of the images realised.

Submission requirements

When you have completed your assignment, make a copy of your project and the written description of your project, other project files, output images and other associated files in to a directory called *Ocean_Waves_Project* in your home directory. Zip this file.

Write a short *README.txt* plain text file, describing what (and why) you have included within your *Ocean_Waves_Project* project folder.

Upload the *Ocean_Waves_Project.zip* and the *README.txt* file to your assignment area (to be arranged by Gareth) as per the due dates above.