Mathematics Assignment #4

Ray Tracing a Triangle Based Model

Start Date: Thursday 15th March 2011

Ray Tracing a Triangle Based Model

The objective of this assignment is to write a basic (C++) Ray Tracing program that renders models comprised of triangles.

This is a single person project.

Suggested and required functionality

1. Create a scene description describing the camera, lights and a set of (including procedurally generated) objects and associated material properties.
2. Cast rays into the scene to simulate the image formed by a pin-hole camera. One ray per pixel is sufficient – though please feel free to attempt anti-aliasing.
3. Recursively intersect rays with the geometric primitives - triangles.
4. Implement an illumination model which includes ambient, diffuse, and specular terms for each light source as well as shadowing, reflection and refraction. You only need to handle multiple directional and point light sources, i.e. no area lights.
5. Write the image data to disk.
6. Display a rendered – and rendering – image
7. If your ray tracer renders an image by tiles or lines then these should be progressively displayed.
8. Your Program must allow a user to:
    - Select a view position and view direction
    - "Render"
    - "Cancel" a render
    - "Save" the resulting image
    - "Exit" the program

Notes on your illumination model

1. Your implementation will require you to trace rays towards each light, and may also require you to recursively trace reflected and refracted rays.
2. When tracing rays toward lights, you should look for intersections with objects, thereby rendering shadows.
3. If you intersect a semi-transparent object, you should attenuate the light, thereby rendering partial shadows, but you may ignore refraction.

Software requirements

1. It will be implemented in C++ using the supplied RTVS Lite framework.

2. Your project should be created in Microsoft Visual C++ Express 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 model of ocean waves (see the supplied DirectX framework).
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.

Delivery and Marking

There are seven parts worth 10, 10, 30, 20, 5, 5, 10 and 10 marks respectively:

1. Create a scene description describing the camera, lights, one or more triangle based object and associated material properties (10).
2. Cast rays into the scene to simulate the image formed by a pin-hole camera. One ray per pixel is sufficient (10).
3. Ray trace by recursively intersecting rays with the geometric primitives – triangles (30).
4. Implement an illumination model which includes ambient, diffuse, and specular terms for each light source as well as shadowing, reflection and refraction. You only need to handle multiple directional and point light sources, i.e. no area lights (10).
5. Display a rendered – and rendering – image (5).
6. If your ray tracer renders an image by tiles or lines then these should be progressively displayed (5).
7. Real-time ray tracing (10)
8. Write a description (name it "*raytracer.doc*" and be 500 words or more) of your project detailing your objectives, data classes and other structures (10).

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 *raytracer* in your home directory. Zip this file.

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

Upload the *raytracer.zip* and the *README.txt* file to your assignment area.