CS 563 Project 2
Due: Thursday, March 31, 2005

Overview

This goal of this assignment is to learn the Cg shader programming language and perform a few tasks in it, including combining the use of Cg with OpenGL. It is assumed that you know how to write OpenGL programs. Please see me if you cannot. The assignment is broken down into parts which should guide you.

Step 1: Set up development platform:

First, you need to set up your development platform. First, you need to make sure that your graphics card has a programmable GPU and supports vertex and fragment shaders (e.g. Nvidia FX series). The manufacturer's website should give the above details.

Next, you need to install and make sure that the following software is working on your system:

- **Visual C++ (version 6 or later) or Visual C++ .NET**
- **OpenGL and GLUT**
- **Cg toolkit:** Download the latest Version of the Cg toolkit at http://developer.nvidia.com/object/cg_toolkit.html. A few good references for setting up Cg include The "Hello Cg!" article by Alex D'Angelo, lesson 47 at http://nehe.gamedev.net
- **Cg Tutorials:** Download the Tutorial Examples that come with the book, Cg Tutorial at http://developer.nvidia.com/object/cg_tutorial_software.html

Step 2: Go through Cg tutorials

Next, we shall get used to Cg. If you already comfortable with Cg, skip to the next step. Otherwise, read the papers "Cg in Two Pages" by Mark Kilgard and the "Hello Cg!" article by Alex D'Angelo. These give you a clear idea of the basic parts of a Cg program.

Now, you are ready to run the Cg tutorial software. After installing the Cg Tutorial Examples, you can fire up the tutorial framework by typing (or double-clicking):

C:\Program Files\NVIDIA Corporation\Cg Tutorial Examples\Bin\CgTutorial.exe

From the file menu, select "Open Setup". Load .ini files, which typically render a scene and open its associated vertex and fragment programs for editing in an editor window. Load and go through a few of these example .ini files. These .ini files can also be opened and inspected using a simple text editor. Open and load a few of them to see how they are set up. Focus on the examples in chapter 5, which begin with the prefix C5_ . **Make sure you understand**

the .ini and shader code of the chapter 5 code. This chapter of the Cg tutorial book deals with lighting and shading. I've placed a copy of the book, "The Cg Tutorial", on reserve in the library.

Step 3: Study BRDF models

Look at the shader implementation slides of the Nvidia Cg introductory session at http://developer.nvidia.com/docs/IO/8229/Cg-Sessions-Introduction.pdf. These slides explain clearly the theory and Cg code for a phong shader. Next, using the C5_vertexLighting2.ini as an example, create a new .ini file vertex shader and passthrough fragment shader to implement Phong shading. The bulk of your work should be done in the vertex shader. The passthrough fragment shader simply sets the output so that you can view your phong shaders.

Now that you're an expert on implementing BRDFs ☺, read paper on "Surface Reflection Models" on BRDF models by Frank Losasso at http://www.cs.wpi.edu/~emmanuel/courses/cs563/S05/projects/surface_reflection_losasso.pdf. Implement the (Cook-Torrance) and Ward anisotropic models.

Step 4: OpenGL and Cg

Now, it's time to move beyond the Cg tutorial framework that we have been working with. Get and download the PlyLoader project, which can read and display .ply files. The PlyLoader project is at http://www.cs.wpi.edu/~emmanuel/courses/cs563/S05/projects/PlyLoader.zip. When you build and execute this project, it loads a mesh file in the .ply format. Hitting 'N' displays the next .ply file in the /ply folder and hitting 'P' displays the previous .ply file.

The .ply file is a 3D file format used widely in the Stanford 3D scanning repository and the Georgia tech large model database. You can learn more about the ply file format at http://astronomy.swin.edu.au/~pbourke/geomformats/ply/.

Create a /shaders directory in your PlyLoader project and copy your Cook-Torrance and Ward anisotropic shaders from the previous step into this directory. You'll want to expose your shader parameters. These easiest way is to use #defines in a header file. You can also use some of GLUT's pulldown menus. Make it possible to select different shading models (Phong, Cook-Torrance and Ward) from within your program using GLUT pulldown menus. Make sure to say how I can access these BRDF models in your final README file.

Step 5: Texture and environment mapping

Texture map: Map a texture onto the rendering of your .ply files. You can use one of the 2D textures in Textures/2D folder of the Cg tutorial examples.

Environment map: Add environment mapping to the rendering of your .ply files. You can use the EnvironmentMap.dds file in the Textures/cube directory of the Cg tutorial examples.

If you need more information than provided in the Nvidia Cg startup slides, chapter 3 of the Cg tutorial book also deals with texture and environment mapping.

What to submit

Create a simple README to explain what you have done. If your program has known deficiencies or aspects that you did not get to implement, say so. Zip up your Visual C++ project, source files and .ply files and submit the zip file. Email me this zip file. Also include your .ini file for your phong shading in the first part.