

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY
Department of Computer Science and Engineering

COMP 5411 : Advanced Computer Graphics

Fall 2011

Programming Assignment 2: Programmable Shading

Due date: **November 20, 2011 at 11:59PM** (using CASS)

In this assignment, your objective is to render meshes efficiently using programmable shading.

The starting mesh viewer for the assignment can be downloaded from here:

<http://www.cse.ust.hk/~psander/5411/pa2.zip>

The included mesh viewer allows you to visualize the mesh in several modes by pressing the keys 1 through 7 or using the popup menu as in previous assignments. The active mode is displayed in the console window. All but the last two “Custom Shader” modes are already implemented in the skeleton code using the fixed function pipeline. You should complete the shaders in the *.glsl files in order to successfully render the mesh using the last two modes, which should employ programmable shaders.

Loading the meshes: You should specify the path to the venus.obj or dragon.obj mesh files in the command line (e.g: ../modes/venus.obj) . For tasks 2 and 3, you must use the venus mesh because it is the only one that contains texture coordinates. For the remaining tasks, you may use either mesh.

1) Gouraud shading (50%)

Implement the “Custom Shader – Gouraud Shading” mode. To do so, you must complete the vertex shader in blinn-phong-vert.glsl that performs Blinn-Phong shading at each vertex as described in the lecture notes (no attenuation term). You should mimic what is done by the fixed function pipeline solution for the “Smooth Shading” mode by looking at the cpp source code and the information provided in the glsl file. You should use the built-in uniform variables to retrieve this information in the vertex shader. Finally, you should fill in the pixel shader so that simply interpolates the colors computed at the vertices. If correct, the result should match that of the provided “Smooth Shading” mode.

2) Texture mapping (15%)

Add a texture lookup in the pixel shader to read from the only supplied texture using the provided texture coordinates. After doing this, your “Custom Shader – Gouraud Shading” should match that of “Textured Smooth Shading”.

3) Phong shading (15%)

Implement the “Custom Shader – Phong Shading” mode in blinn-phong-frag.glsl. To do so, you must move your Blinn-Phong shading computation from the vertex shader to the pixel shader (Hint: You can mostly reuse the same code as pointed out in the glsl comments.) The vertex shader simply passes the vertex normals on to the rasterizer. The pixel shader should receive

the interpolated normals and perform the Blinn-Phong shading computation for each pixel. Important: You should add the specular contribution only after modulating by the texture.

4) **Vertex animation (20%)**

Finally, you should modify the vertex shader in `blinn-phong-frag.glsl` using an interesting animation function. The function should modify the final vertex position based on the input vertex position and a provided frame-time variable. To get full credit, your effect should be an interesting effect, similar or possibly more sophisticated than the one at this URL that uses sine functions:

<http://www.cse.ust.hk/~psander/5411/pa2animexample.avi>

Note: In order to be able to develop in your own machine, you must have a desktop or notebook with a DX10-level or above graphics chip (i.e., Radeon HD2000+ or NVIDIA GeForce 8000+/GTX) and install the latest OpenGL drivers. If you do so and the program doesn't run, email the TA Mr. Ge Chen at gechen@ust.hk and describe the error message that you are getting. If you don't have access to such a machine let us know and we will try to arrange it for you to use one of the machines in lab 4.