Practical 4a – OpenGL: GLSL Shaders

This practical is designed to improve upon your OpenGL skills that were developed in the introduction practical. In this practical you build upon concepts learned in the first prac, which means you **should** reuse the original code. You are asked to demonstrate your competence with OpenGL shaders, lighting and texturing techniques.

What you need to do

Take the setup provided by the first practical: with the object located in the centre of the world. This time your object does not need to move. However, if you succeeded in getting it to move and rotate in the previous practical, then leave this as an option. The camera must be constrained to look at the centre but must also be able to orbit around this fixed point, as before. Next you will need to add some additional steps for this practical which are outlined below.

- ### 1) Phong-shading
  You are required to add light sources to the world, minimum of 2. Each of these needs to have different colour values to show up effectively on your object. Per-pixel lighting is required for this and the accumulated value of all the lights needs to be factored in for the final pixel value.

- ### 2) Shader files
  You will need to create both a vertex and fragment shader for this purpose. This is where you will put the code necessary to have phong-shading in the scene.

- ### 3) Moving lights
  The lights that are added to the scene need to be able to move: this can either be done by transforming them individually or even having them all orbit in either a clockwise or anti-clockwise direction around the scene. This provides another method to show how the light falls on the object's surface.

- ### 4) Texturing
  You need to apply a texture file to the surface. This is quite dificult as you will need to find code online that can handle the loading of a texture file into your program. There is also the issue of assigning texture coordinates to your object. Unfortunately, STL format does not support texture coordinates so you will have to setup a texture projection (e.g., spherical, cylindrical or planar).

- ### 5) Bump Mapping
  In order to provide some challenge to the more eager students there is a final 15% available for adding bump-mapping to the surface of the object. This is achieved through shader techniques that perturb the surface of the teapot. This requires not only the texture file but a secondary image that is a bump-map.

Marking

Feature	Marks
Phong-Shading	35%
Shader files	10%
Moving Lights	25%
Texturing	15%
Bump-Mapping	15%
Total	100%

Submission Date

Hand in by 10:00am Monday, 17 August.

Useful Resources

- OpenGL Tutorials (Setup & Phong-Shading) http://lighthouse3d.com/opengl/
- OpenGL Tutorials (Various shader techniques) http://nehe.gamedev.net/