Assignment 3. Texture and Normal Mapping

Total Number of Points for this Assignment: 20 points

The first and second assignments are about the transformation, lighting and shading, for rendering. The goal of this third assignment is to extend your program to implement texturing mapping and normal mapping.

You are required to write your program using OpenGL shading language (GLSL) to do texture mapping and normal mapping. Your program should provide at least the following features:

a) Implement the texturing mapping in GLSL. The texture is defined in each image file. How to read an image will be introduced in the Appendix. You need to do a texture mapping based on the given texture coordinates. (6 points)

b) Implement normal mapping in GLSL. The normal direction is defined in each image called normal map image. Based on the given texture coordinate you can uniquely get the normal directions from the texture space. (8 points)

c) Implement a direction/point light. The direction light is with direction (0, -1, -1) in camera coordinate system. The point light is a point source rotating around the center of the triangular model. The user can choose to rotate it around X, Y, or Z-axis of the world coordinate system. You need to support interactive changing the direction/position of light through the GUI. Support interactive changing RGBA values associated with the diffuse component of the light source. In this assignment, you’re just required to consider the diffuse and specular reflection of the Phong illumination model. (3 points)

d) Turn light on and off. (1 point)

e) Turn texturing mapping on and off. (1 point)

f) Turn normal mapping on and off. (1 point)

Appendix

Read an image

You’re free to use any library that you’re familiar to read the image.

Here we are using the SOIL(Simple OpenGL Image Library) to read images.

The source code can be find: http://www.lonesock.net/soil.html

You can simply use this lib via:

```
#include <SOIL.h>

……

int img_width, img_height;
unsigned char* image = SOIL_load_image(img_path.c_str(), &img_width,
&img_height, 0, SOIL_LOAD_RGB);

glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, img_width, img_height, 0, GL_RGB,
GL_UNSIGNED_BYTE, image);

……
```

There's a lot of similar libraries. You can pick any one you like or you can read it by yourself.