Assignment 3: Texture and Normal Mapping

## Total of Points for this Assignment: 15

The previous two assignments are about the transformation, lighting and shading. The goal of this 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:

1. 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. (5points) 🡺 texture map
2. 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. (5points)🡪 normal map
3. 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 RGB 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. (2points)🡪lighting map
4. Turn light on and off. (1 point)
5. Turn texturing mapping on and off. (1 point)
6. Turn normal mapping on and off. (1 point)

# Appendix I: How to load an image as texture

Please feel free to use any [image loading libraries](https://www.khronos.org/opengl/wiki/Image_Libraries). In this course, we provide a single header image loading library by [Sean Barrett](https://github.com/nothings) called *stb\_image.h* which can be downloaded from [here](https://github.com/nothings/stb/blob/master/stb_image.h). This header file has been provided in the starter code. There are also more single-file libraries provided in [stb](https://github.com/nothings/stb), such as image resizing and writing, please feel free to include more for you own need.

For MacOS, the stb\_image.h file has been included in *Renderer.h*, you can directly call functions such as *stbi\_load()* to load image.

For Windows x64, you need to load [stb.props](https://github.com/CS6366utd/environment_setup_windows/blob/master/stb.props) in the ‘Property Manager’ under VS2019, similarly as you load [OPENGL.props](https://github.com/CS6366utd/environment_setup_windows/blob/master/OPENGL.props) and [NANOGUI.props](https://github.com/CS6366utd/environment_setup_windows/blob/master/NANOGUI.props). Please check out the tutorial video if forgot how to do so. Then you can uncomment following two lines in *Renderer.h*:

|  |
| --- |
| // #define STB\_IMAGE\_IMPLEMENTATION  // #include "stb\_image.h" |

Another tip, remember to checkout OpenGL tutorial (related to [texture](https://learnopengl.com/Getting-started/Textures)) if you do not know how to start!!