

TP 3

*Submission date:**10 October*

Problem. The main goal of this assignment is for you to become familiar with texturing in OpenGL.

1. Add a texture class to your project so far, and add a variable of its type to your subObject class. Read a ppm texture file, and then add code to access that texture from your shaders. To test if the texture is working, you can use any texture coordinates for the vertices of the object. You will have to do:
 - Add a texture class that has a read function to read an image. Texture class has the variable `texName` for storing the texture ID.
 - Add a variable of type texture class in your object class.
 - First generate the texture coordinates for the object.
 - Then create object buffers to use these texture coordinates array (similar to what you did for vertex positions and vertex normals).
 - When drawing with textures, you should activate `GL_TEXTURE1`:
`glActiveTexture(GL_TEXTURE1);`
`glBindTexture(GL_TEXTURE_2D, ID for the texture image);`
 - You will need to pass texture coordinates from the vertex shader to the fragment shader through shared variables (similar to what you did for vertex positions and vertex normals).
 - In the fragment shader, the variable for texture image is defined as:
`uniform sampler2D tex ;`
 - In the main, set this uniform variable value to 1.
 - You can access this texture image in your fragment shader via:
`texture2D(tex, yourtexturecoordinatevariable.st)`
 - Online I have given you the function we wrote in class to generate cylinder texture mappings. You can use that to test the texture mapping if you wish.
2. In the class `myObject`, add functions to generate Sphere texture coordinate mapping. Check the function on the sphere with a map to see if it is working correctly.
3. Also add functions for generating plane texture coordinate mapping, and the cylinder texture mapping that we computed in class.

The texture class:

```
class myTexture
{
public:
    int width, height, pixelsize;
    GLuint texName;
};
```