Lab 2 Toon Shading

In this lab we are going to write a simple Toon Shader. However before creating the shader we are going to modify the application code to make the process of linking to shader’s uniforms easier. Download and open up the shader.h file. You will see new functions have been added. These functions allow us to set the values of the shader’s uniform variables using “glUniform” and “glGetUniformLocation”. This code will be discussed in the lecture.

As our shader get more complex there will be certain details that we will want to pass to the shaders, in particular the model, view, and project matrices…

Todo:

In the Camera.h file write two getters, one for the view matrix and one for the projection matrix.

Create a new vertex and a new fragment shader…

Vertex Shader…

Pass in the positions and the normals, these are stored in the vbos, i.e…

layout (location = 0) in vec3 Position;

layout (location = 2) in vec3 Normals;

Create two variables one for our normals and the other for our transforms...

varying vec3 normal;

uniform mat4 transform;

Inside the main method pass our normals to the fragment shader and set our new positions…

normal = VertexNormal;

gl\_Position = transform \* vec4(VertexPosition, 1.0);

In the fragment shader…

Create two variables one two hold the light direction and on is our normals from the vertex shader….

uniform vec3 lightDir;

varying vec3 normal;

Complete the following main…

void main()

{

float intensity;

vec4 color;

to do: calculate intensity and control color based on intensity

gl\_FragColor = color;

}

Now go back to the main game class and create a method that passes the light direction to the shader and run the code.