## Multiple Lighting Sources

Create an array of Light structures in your Application class

struct Light {

glm::vec3 direction;

glm::vec3 color;

};

Light m\_light[2];

And initialise them in startup()

When we pass these through to the shader as a uniform in draw(), copy them first to primitive arrays, and pass them through using the appropriate bindUniform function.

// copy our light info into primitive arrays

glm::vec3 lightDirections[2];

glm::vec3 lightColors[2];

for (int i = 0; i < 2; i++)

{

lightDirections[i] = m\_light[i].direction;

lightColors[i] = m\_light[i].color;

}

// pass the arrays through

m\_texturedShader.bindUniform("lightDirection", 2, &lightDirections[0]);

…

In your frag shader, make the uniforms that catch this data into arrays

uniform vec3 lightDirection[2];

uniform vec3 lightColor[2];

Make the diffuse light calculation into a function and call it from a loop in your shader.

vec3 diffuseLight(vec3 N, int index)

Make the specular light calculation into a function and accumulate those terms too.