

```

1  #version 330 core
2
3  // uniform data
4  uniform mat4 uProjectionMatrix;
5  uniform mat4 uModelViewMatrix;
6  uniform vec3 uColor;
7
8  // I created these-----
9  uniform float u_ambient_coeff;
10 uniform float u_diffuse_coeff;
11 uniform float u_specular_coeff;
12 uniform float u_power_specular_coeff;
13 uniform vec3 u_light_position_coeff;
14 //-----
15
16 // mesh data
17 layout(location = 0) in vec3 aPosition;
18 layout(location = 1) in vec3 aNormal;
19 layout(location = 2) in vec2 aTexCoord;
20
21 // model data (this must match the input of the fragment shader)
22 out VertexData {
23     vec3 position;
24     vec3 normal;
25     vec2 textureCoord;
26 } v_out;
27
28 void main() {
29     // transform vertex data to viewspace
30     v_out.position = (uModelViewMatrix * vec4(aPosition, 1)).xyz;
31     v_out.normal = normalize((uModelViewMatrix * vec4(aNormal, 0)).xyz);
32     v_out.textureCoord = aTexCoord;
33
34     // set the screenspace position (needed for converting to fragment data)
35     gl_Position = uProjectionMatrix * uModelViewMatrix * vec4(aPosition, 1);
36 }
37

```