

1 One or More External 3D Model

There are a total of seven 3d models imported from an online source.

2 Structure for Lights

2.1 Spotlight

The Light struct defined in Light.hpp has a LightType member through which we can define the type of the light. For a spot light, the member is assigned an enum value for spotlight. It also has a direction, a cutoff angle, intensity exponent, and attenuation values. The direction is a vector that defines which way the spot light is shining. The cutoff angle determines how wide the light source will shine, with respect to the direction. Only those fragments that are within the cutoff angle from the direction will be shone. The intensity exponent is one that is applied to the intensity so that the intensity gradually decreases from the center to the edge through a \cos^e curve. The constant, linear and quadratic attenuation values are those that are used to soften the light as distance increases, through the equation $1/(a+bd+cd^2)$. These values are defined for each light objects and then passed on to the material class.

2.2 Point Light

A point light is similar to the spot light but there is no cutoff angle or the direction. The light propagates in all directions. Thus, cutoff angle, direction, and intensity exponent values are not used in this case. However, the three attenuation values are still used because the intensity of a color is affected by the distance from the light source.

3 Shader Programming

3.1 Phong Shading

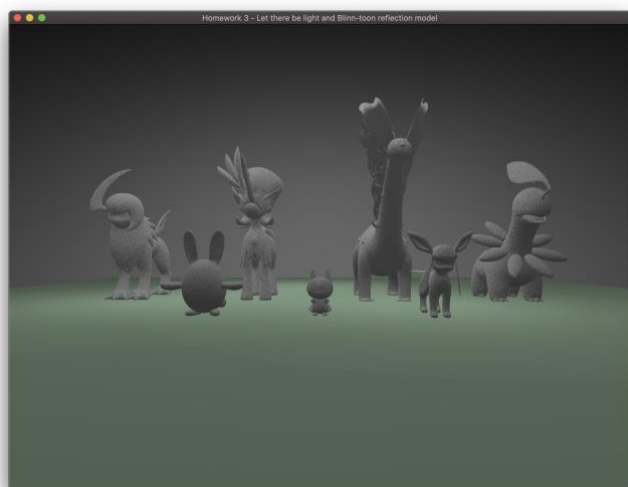
The PhongMaterial and a relevant shader is created based on the Blinn-Phong reflection model. The key part of the model is the use of half-vector. The half-vector is used instead of calculating the dot product of reflection vector (r) and the view vector (v). To calculate the half-vector, we need to know the position of the camera. So I passed the position of the camera through the material class. Then the calculated half-vector was used in the specular component of the intensity as the dot product of normal vector and half-vector. This value was then applied a shininess exponent which is defined for each material. Also, the light type is checked, and if it is a spotlight, only the fragments inside the cutoff angle are given intensity. Directional light is not supported.

3.2 Toon Shading

The ToonMaterial is also created based on the Blinn-Phong model. Therefore, it also utilizes the half-vector to calculate the specular component of the intensity. The difference from the PhongMaterial is that the intensity is discretized. I discretized the intensity based on its magnitude. If the magnitude was less than 0.25, the intensity was divided by four, if it was less than 0.5, the intensity was divided by three and so on. This created the cartoon-like effect, which is characterized by the set number of levels of intensity.

4 Scene

4.1 Cycle



Scene 1

There are three scenes in this project. The three scenes can be cycled by pressing "s" on the keyboard. Pressing "c" on the keyboard enables and disables the cursor callback, so that the effect of materials can be observed more easily by rotating a creature. The first scene includes creatures rendered with MyMaterial. There are one point light and three spotlights. The spotlights



Scene 3

orbit around the point light while moving back and forth. In the second scene, the creatures are rendered with the ToonMaterial. The colors of the ground and sky are changed. The spotlights continue orbiting. In the third scene, the creatures are rendered with PhongMaterial. The spotlights continue



Scene 2

The spotlights continue orbiting but have vibrant colors.

4.2 Animation of Lights

There are four lights in all scenes. There is a point light at the center and three

spotlights around it. I set the parent of the spotlights as the point light, so as the point light rotates, the spotlights rotate with it. As the spotlights rotate around the center, they also move back and forth, which creates the lively effect.

5 Creativity

The story of the scenes is as follows. During the day, which is Scene 1, the creatures are rock statues. In the evening, which is Scene 2, the creatures turn into cartoon characters. In the night, the creatures turn into real creatures. Then the scenes cycle. I put creativity into the MyMaterial to implement the effect of a rock statue. First, I found a formula for turning an RGB value into a grayscale value, which is $0.3R + 0.59G + 0.11B$. I applied this formula to all intensity components to create a grayscale material. I also added some noise to the fragment normal vector. I found a formula called Gold Noise to make random numbers. I used the noisy fragment normal vector to create a rough-looking material. Combining these two effects, the material became gray and rough like a rock. Using the three materials, I could create rock statue creatures, which turn into cartoon characters, then into real creatures.