

Introduction

In this assignment, we will finish the lighting model by implementing specular lighting.

The User Interface

The user interface is the same as in Assignment 10. The lighting effects should change with a change in the viewpoint. Make the shininess a maximum so that the change will be obvious.

The Lighting Model

This model will include the ambient light and the diffuse light. Each is a vector, normally set to a gray level (bright or dim white light). The ambient light is usually low and the diffuse light is usually high. We will need to introduce several new variables:

```
vec3 light_spec;  
GLfloat light_shiny; // To become mat_shiny soon  
  
GLint light_spec_loc;  
GLint light_shiny_loc;
```

These variables should be initialized and used in the usual way.

The `setLight()` Function

The `setLight()` will send the values of the uniform variables `light_spec`, `light_shiny` to the shaders.

The Vertex Shader

The vertex shader will be the same as in Assignment 10.

The Fragment Shader

In the fragment shader, add the code to calculate the specular reflection. The fragment color will be the sum of the ambient, diffuse, and specular reflections.

Due Date

This assignment is due by Friday, October 25. The next assignment will implement the material properties.