CS 355 Lab #8: 3D Lighting

Overview

In this lab, you will implement a simple 3D lighting program. You will be lighting a sphere with a directional light source. You will color the sphere according to the Phong model.

User Interface

Your only interaction will be through the keyboard. Pressing the keyboard will change the perceived location of light. **The light should rotate around the sphere, not simply move in a plane**. The following keys should rotate the light in the following ways:

a	Rotate left
d	Rotate right
w	Rotate up
s	Rotate down
q	Rotate counterclockwise
e	Rotate clockwise

See the example video in the course content for an example of what each key does. Pressing the a or d keys should cause a rotation about the Y axis, pressing the w or s keys should cause a rotation about the X axis, and pressing the q or e keys should cause a rotation about the Z axis.

Implementation Notes

All of the 3D geometry and projection is provided for you in this lab. Again, we are using the Pygame package to do this. Make sure to include the basicShapes.py and wireframe.py files in the same directory as your code. It will not run without them.

If the lab file is setup correctly, when you run the code, you should see a very faint beach ball displayed in a window. The ambient light portion of the lab is done for you, but you will need to implement the diffuse and specular reflections from the Phong model.

The code provides the viewing vector and the incoming light vector. However, you will need to calculate the reflection vector for specular lighting. The reflection vector can be calculated as

$$r = 2(l \cdot n)n - l$$

where r is the reflection vector, l is the incoming light vector, and n is the normal to the surface at that point.

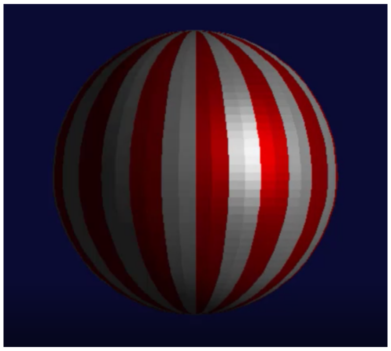

Figure 1: Final Lighting of the Sphere

Phong Model Colors and Constants

The Phong model of lighting is a very general set of equations that allows for material properties that respond differently to ambient, diffuse, and specular lighting. For this lab, we have chosen to make the material property the same color for ambient, diffuse, and specular lighting. Thus, there is only one light source color and only one material color.

There are some constants that you will need to define. Specifically, you will need to set the coefficients describing how much ambient, diffuse, and specular lighting you want. You may set these to whatever you wish as long as the coefficients sum to 1.0 and the result looks realistic. You will also need to define a value for your Phong constant (m_{gls}). Again, choose any value you wish and feel free to experiment with different values to see how it affects the overall lighting.

Don't forget to check a polygon to see if it actually is visible from the light source. **If the normal vector dotted with the light vector gives a negative value, the polygon is in shadow and should only have ambient light**. Also, don't forget to clip your final result color to be between 0.0 and 1.0 for each polygon.

Submitting Your Lab

Your code should be contained inside a single .py file. To submit the lab, simply submit this file through Learning Suite. **You do not need to include the wireframe.pyc and basicShapes.pyc files with your submission.** If you need to add any special instruction, you can add them there in the notes when you submit.

Rubric

- Correct rendering of diffuse reflection (20 points)
- Correct rendering of specular reflection (20 points)
- Correct navigation of light source (20 points)

- Generally correct behavior otherwise (10 points)

  TOTAL: 70 points