Assignment 03: Lighting and Shading

Documentation

Tanishq Jain, 2021294

Question 1 - Switch on lighting using a point light source.

a) In code, a 3-D parametric sphere is implemented where parameters are angles θ, ϕ such that a vertex on the sphere is given by

$$V(\theta, \phi) = (R\sin\phi\cos\theta, R\sin\phi\sin\theta, R\cos\phi).$$

By using formula for calculating normal (given in assignment), we get unit normal vector as

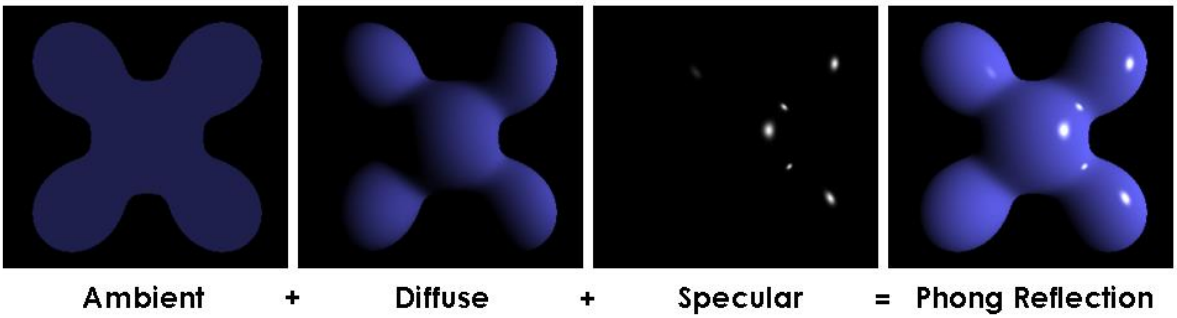
$$N(\theta, \phi) = (\sin\phi\cos\theta, \sin\phi\sin\theta, \cos\phi).$$

Using these 2 functions, we iterate over values of theta, phi to sample vertices on the sphere and their corresponding unit normal vectors.

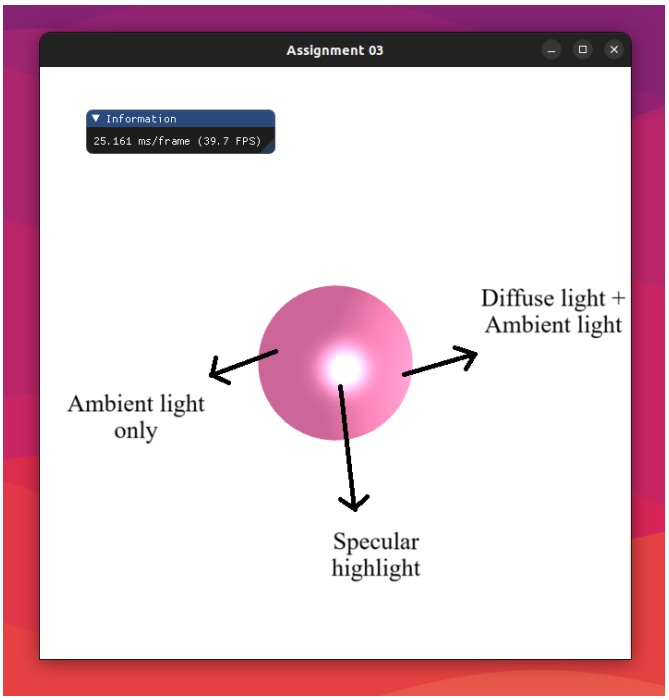
b) A point light source is identified by the location of the light and its colors. Hence, we introduce uniform vec3 variables *lightPosition* and *lightColor* corresponding to these properties. Values for these uniforms are set in main.cpp file.

c) We modify the vertex shader *(Gouraud shading)* to implement the phong lighting model. This involves computing the 3 lighting components – ambient light, diffuse light and specular light.

We set the ambient and diffuse light color same as the *vColor* uniform and the the specular light color same as the *lightColor* uniform. Also, we set the specular exponent as 32. The overall fragment color is the collective color from these 3 components. The following image (taken from internet) gives a better intuition for the described choices.



After modifying the shaders the result looked like below. Notice that the back side of object is not pitch dark, since we have added an ambient component.

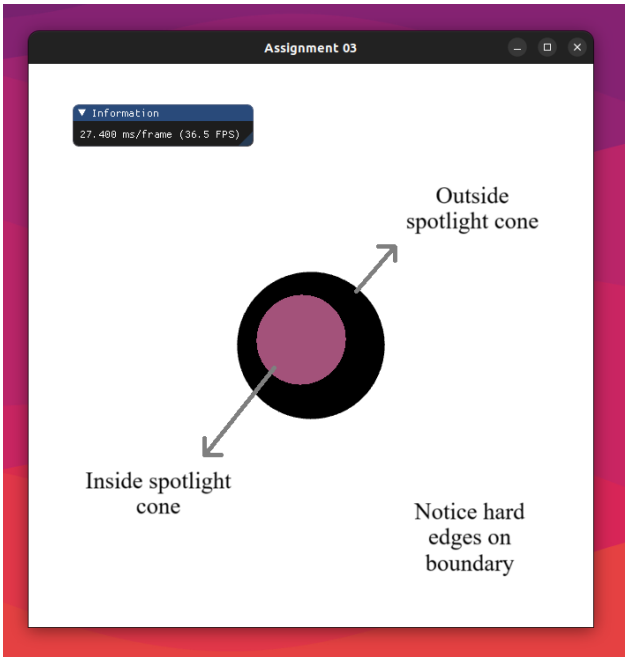


d) In order to use per-fragment *Phong shading*, we use the same lighting model with the 3 components. This time, we simply move the shading components from vertex shader to the fragment shader.

Question 2 – Create a spot light source.

a) A spot light is a light source that emits light in a cone of directions. Hence, only the parts of the object within the cone of illumination are lit. We can represent a spot light source using light location, direction, cutoff angle and color. Hence we introduce uniform vec3 variables *lightPosition*, *lightDirection*, *lightColor* and a uniform float variable *lightCutoffAngle* corresponding to these properties. We choose the *lightCutoffAngle* as 15 degrees.

The basic computations for 3 light components is similar to Question 1. We find the value of a *spotlightFactor*, which is equal to cosine of angle formed between light direction vector and light to fragment vector. If the angle is larger than cutoff angle, its cosine (= *spotlightFactor*) is smaller than cosine of cutoff angle, and hence fragment lies outside the spotlight cone. In this case, the fragment is not lit. Else, we apply the illumination model to the fragment.



b) In the above illustration, notice that the spotlight has hard edges on the boundary of illumination cone. We can implement smooth, soft edges by introducing an *outerCutoffAngle* (= *innerCutoffAngle* + 10 degrees) while *innerCutoffAngle* is same as the original input angle. Between the inner cutoff and outer cutoff, we multiply intensities with an *interpolateFactor* (goes from 1 to 0 linearly).

Hence, the intensity of a fragment falling between the inner, outer cones drops down from 1 (at inner cutoff) to 0 (at outer cutoff) in a smooth fashion, thus achieving soft edges.