

HW3 Guideline

Task: Implement Phong & Gouraud shading.

Phong shading

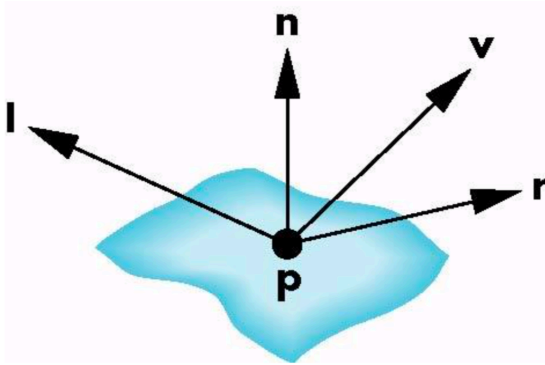
Main.cpp

1. Compute normal vector
Compute the Earth vertices' normal coordinates and pass them to vertex shader by using **glVertexAttribPointer**.
2. Set up light position
 1. You can use **gluSphere** to represent your point light source.
Remember to add color to your light source for visualization and translate it to the specific light position.(We will provide color & pos in spec.)
 2. Pass light position to vertex shader by using **glUniform3fv**.
3. Get ModelView matrix before doing translate.
 1. Using **glGetFloatv** to get current ModelView matrix.
 2. Pass ModelView matrix to vertex shader by using **glUniformMatrix4fv**.

Vertex shader

1. Compute normal matrix
 1. You have to implement normal matrix by yourself.
Hint: You can't directly use ModelView matrix to transform normal vector to view space. You can find out some formula online.
 2. Compute new normal vector
Hint: Using normal matrix & normal vector
 3. Pass new normal vector to fragment shader.
2. Compute ModelView position and pass it to fragment shader.
3. Compute ModelView light position and pass it to fragment shader.

Fragment shader



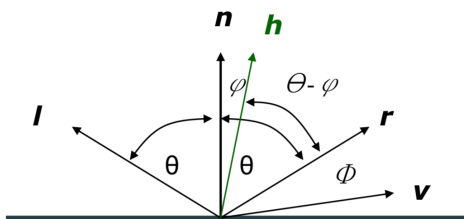
$$I = I_{ambient} + I_{diffuse} + I_{specular}$$

$$= k_a I_a + k_d I_d (l \cdot n) + k_s I_s (v \cdot r)^\alpha$$

Using the Halfway Angle

- Replace $(v \cdot r)^a$ by $(n \cdot h)^b$
- b is chosen to match shininess
- Note that halway angle is half of angle between r and v if vectors are coplanar

$$\text{Halfway vector : } h = (l + v) / |l + v|$$



$$\theta + \varphi = \theta - \varphi + \phi$$

$$2\varphi = \phi$$

1. Initialize parameters by yourself: k_a , k_d , k_s , α , β . ($0 \leq k_d, k_s \leq 1$)
2. Compute four vectors:
 1. To source l
 2. To viewer v
 3. Normal n
 4. Perfect reflector r or you can use h(mentioned above)
3. Compute Ambient, Diffuse, Specular by using these parameters.
4. Output final fragment color.

Gouraud shading

Hint: You should change the order of shader to implement.