CS 419: Production Rendering

Eric Shaffer
Phong Reflectance Model

- Simple model of light reflection from a surface
- Ambient light
- Diffuse light
- Specular light

- Ambient is a hack
  - Simply light throughout space

- Diffuse scatters in all directions

- Specular focuses in the mirror reflection direction

\[ I_p = k_a i_a + \sum_{m \in \text{lights}} (k_d (\hat{L}_m \cdot \hat{N}) i_{m,d} + k_s (\hat{R}_m \cdot \hat{V})^\alpha i_{m,s}). \]
Phong Reflectance Model

- $V$ is the vector from surface point to eye
  - Only used for specular term
- $N$ is the surface normal
  - How can you compute it for sphere?
- $L$ is the vector from surface point to light
- $\alpha$ is the shininess factor
  - What does a larger value do to reflections?
  - What makes a surface look smoother?
- Use unit-length vectors
- Dot products are cosine of angle

$$I_p = k_a i_a + \sum_{m \in \text{lights}} (k_d (\hat{L}_m \cdot \hat{N}) i_{m,d} + k_s (\hat{R}_m \cdot \hat{V})^\alpha i_{m,s}).$$
Phong Reflectance Model

- Simple model of light reflection from a surface
  - $k_a$, $k_d$, $k_s$ reflectance factors for the surface ambient, diffuse, specular all in RGB with values in $[0,1]$
  - $i_a$, $i_d$, $i_s$ illumination from the lights diffuse and specular all in RGB with values in $[0,1]$
  - the $m$ subscript indicates which light

$$I_p = k_a i_a + \sum_{m \in \text{lights}} (k_d (\hat{L}_m \cdot \hat{N}) i_{m,d} + k_s (\hat{R}_m \cdot \hat{V})^{\alpha} i_{m,s}).$$
Walking through some code

- You can grab a simple Python implementation
  
  https://github.com/shaffer1/Ullinois_Rendering

- Code covers
  - Main rendering loop
  - Sphere class
  - ViewPort class
  - Ray class
  - Diffuse Phong shading

- Code doesn’t include
  - Viewing
  - Plane plane or triangle
  - Sampling
  - Rendering multiple objects