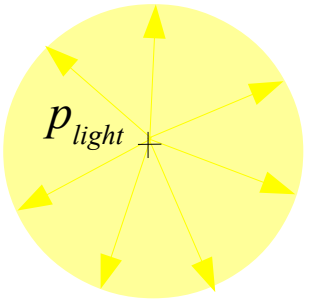
Light Sources

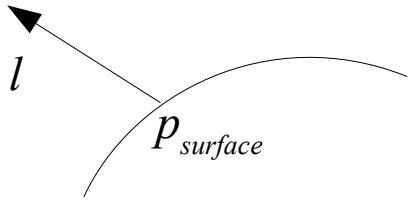
- Point light, directional light, spotlight
- Wrap lighting
- Rim lighting
- Hemispheric lighting
- Trilight

Light source models

Point light (Omni light)

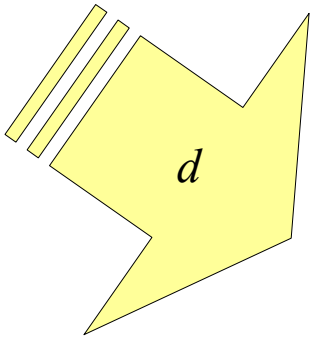
- Parameters
  - Color: L
  - Position: p_{light}
  - $l = (p_{light}-p_{surface})/||p_{light}-p_{surface}||$

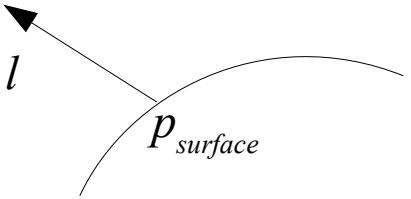




Directional light

- Parameters
  - Color: L
  - Direction: d
  - $l = -d$



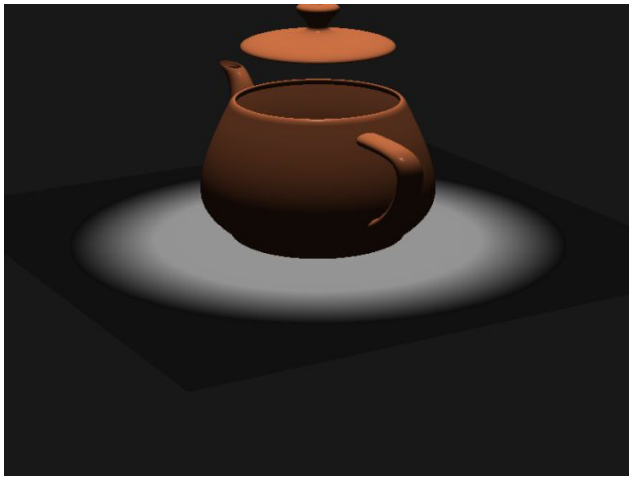


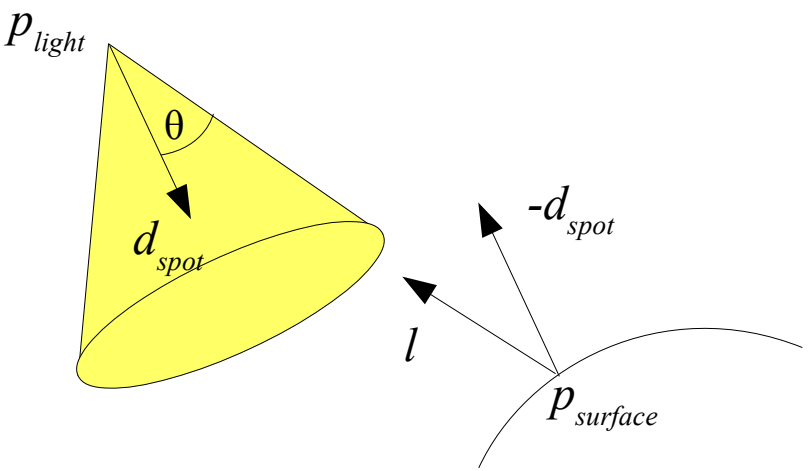
Light source models

Warn Spotlight

- Parameters
  - Color: L
  - Position: p_{light}
  - Direction: d_{spot}
  - Cone falloff: e
  - Cone cutoff: θ

- $l = (p_{light} - p_{surface}) / ||p_{light} - p_{surface}||$
- If $(-l \cdot d_{spot}) < cos(\theta)$ then $I = 0$
- else $I = L\,(-l \cdot d_{spot})^e$





General Purpose Lighting for Games

- “The Trilight - A simple general-purpose lighting model for games”, Tom Forsyth, 2007.

  Generalizations of Lambertian diffuse lighting

  - Wrap lighting
  - Rim lighting
  - Hemispheric lighting
  - Trilight

- http://tomforsyth1000.github.io/papers/trilight/trilight.html

Wrap lighting

- Allow diffuse shading on the dark side of an object.
- More realistic than constant ambient color.

$$I_{diffuse} = L_d \frac{max(0, n \cdot l + f)}{1 + f}$$

- f : wrap factor
  - How far light is allowed to wrap around the dark side.

f = 0.0

f = 0.2

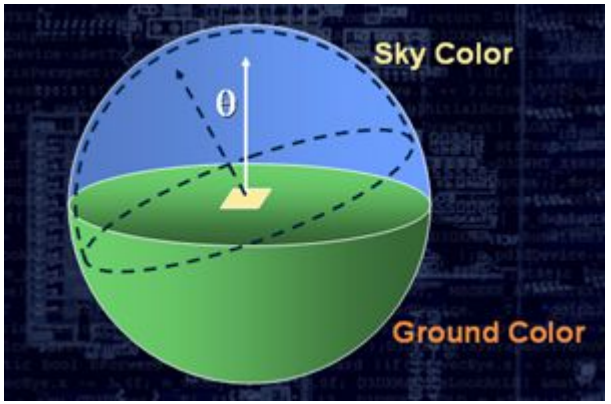
f = 0.4

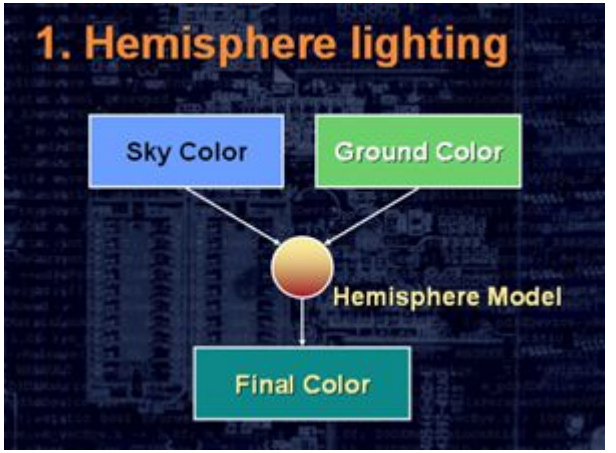
Rim lighting

- Back lighting from behind object
  - $l = -v$

$$I_{diffuse} = L_d \frac{max(0, -(n \cdot v)^e + f)}{1 + f}$$

Hemispheric Outdoor Lighting





- Simulate indirect lighting from a pair of hemispheres
  - Sky
  - Ground
- If z is up in your world:

```
vec4 AmbientLightColor = mix(GroundColor, SkyColor, (worldNormal.z + 1.0)/2.0);
```

GLSL:

- `mix(vec x, vec y, float a)` returns the linear blend of x and y : $(1- a)x + a\,y$

Bidirectional light

- Key light (L_0) and Fill light (L_2)

$$I = L_0 max(0, n \cdot l) + L_2 max(0, -n \cdot l)$$

The Trilight

- Can get similar effects to wrap lighting, hemispherical lighting, and bidirectional lighting

$$I = L_0 max(0, n \cdot l) + L_1(1 - |n \cdot l|) + L_2 max(0, -n \cdot l)$$

- Try different light source models in you project
- It can change the look of your scene with a few lines of shader code