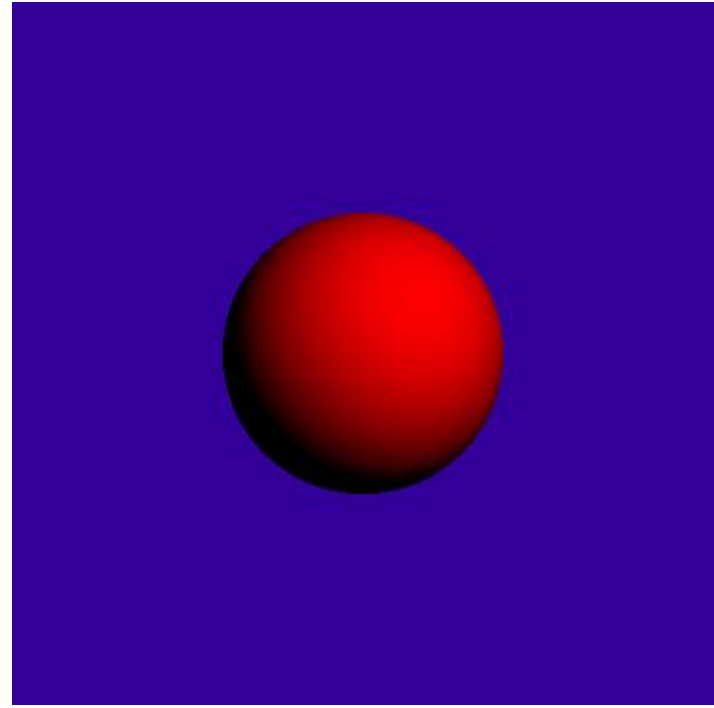
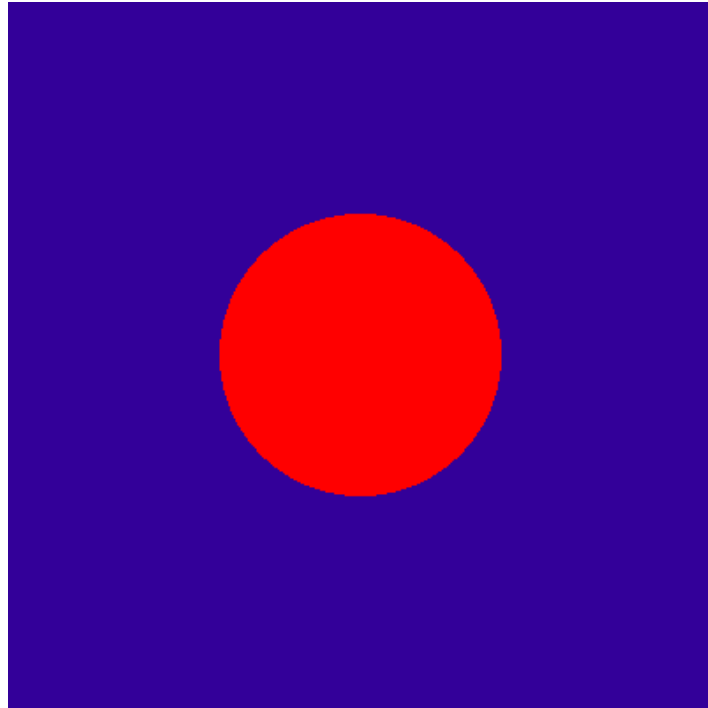


# Computer Graphics 16: Illumination

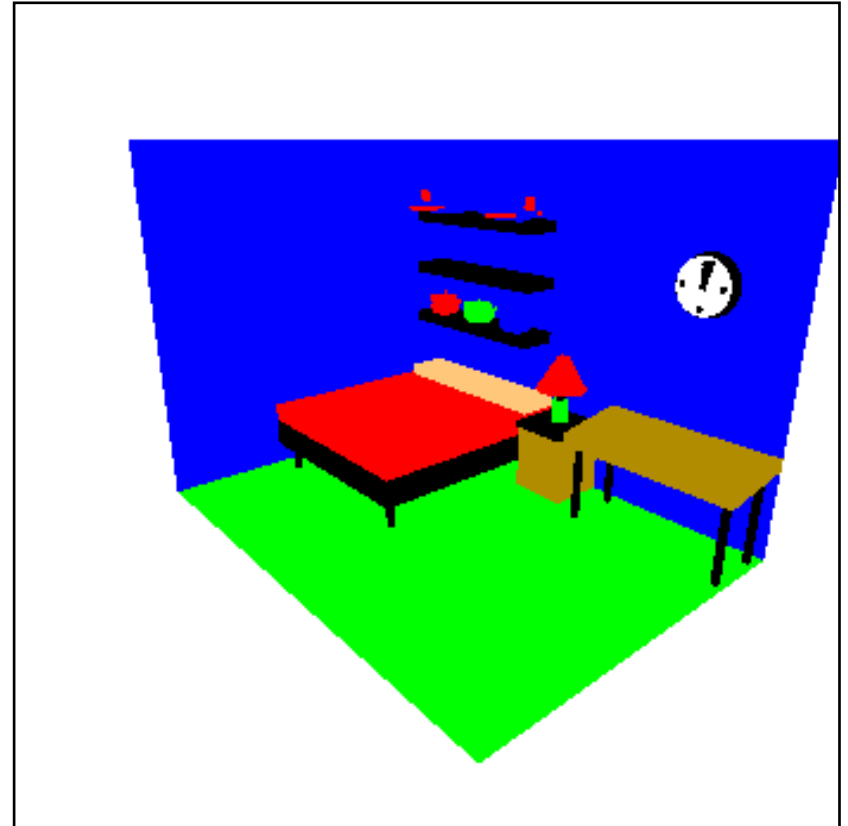
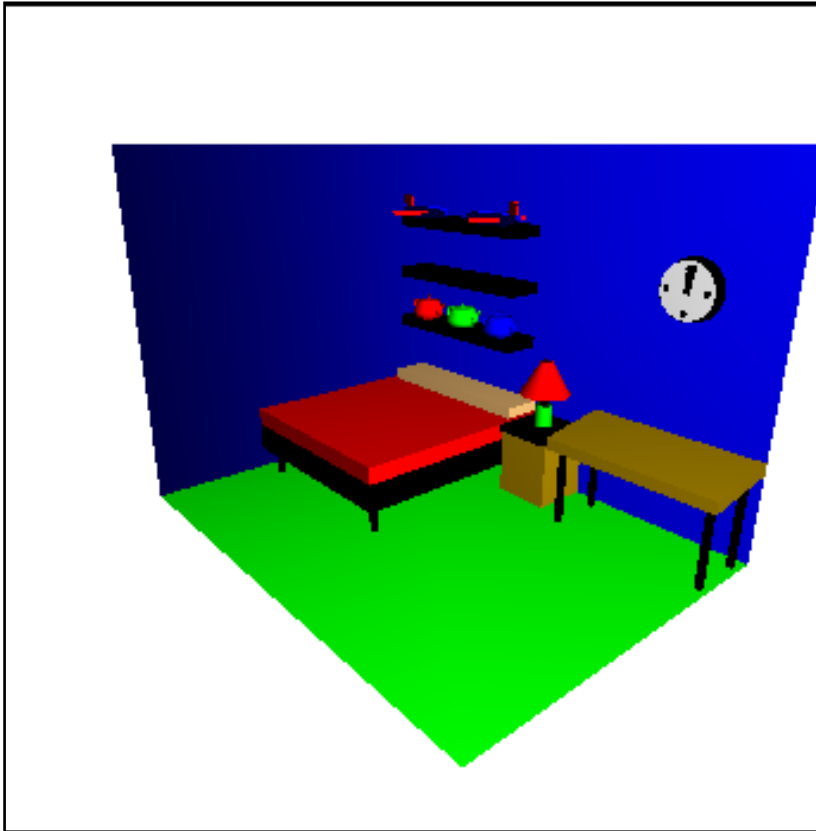
Today we will start to look at illumination models in computer graphics

- Why do we need illumination models?
- Different kinds of lights
- Different kinds of reflections
- Basic lighting model

If we don't have lighting effects nothing looks three dimensional!



# Why Lighting? (cont...)



# Point Light Sources

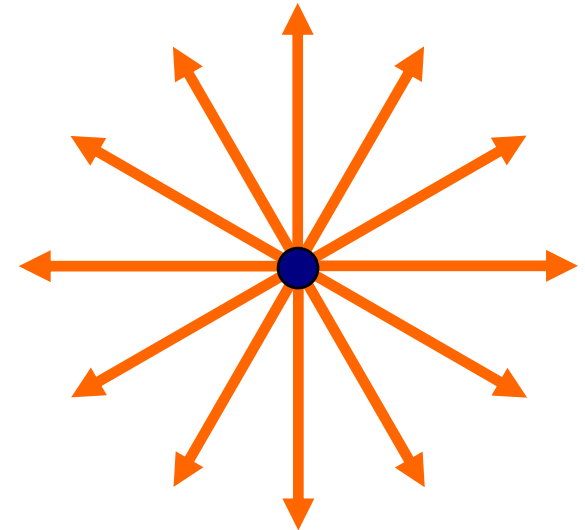
A point source is the simplest model we can use for a light source

We simply define:

- The position of the light
- The RGB values for the colour of the light

Light is emitted in all directions

Useful for small light sources



# Radial Intensity Attenuation

As light moves from a light source its intensity diminished

At any distance  $d_l$  away from the light source the intensity diminishes by a factor of  $\frac{1}{d_l^2}$

However, using the factor  $\frac{1}{d_l^2}$  does not produce very good results so we use something different

# Radial Intensity Attenuation (cont...)

We use instead inverse quadratic function of the form:

$$f_{radatten}(d_l) = \frac{1}{a_0 + a_1 d_l + a_2 d_l^2}$$

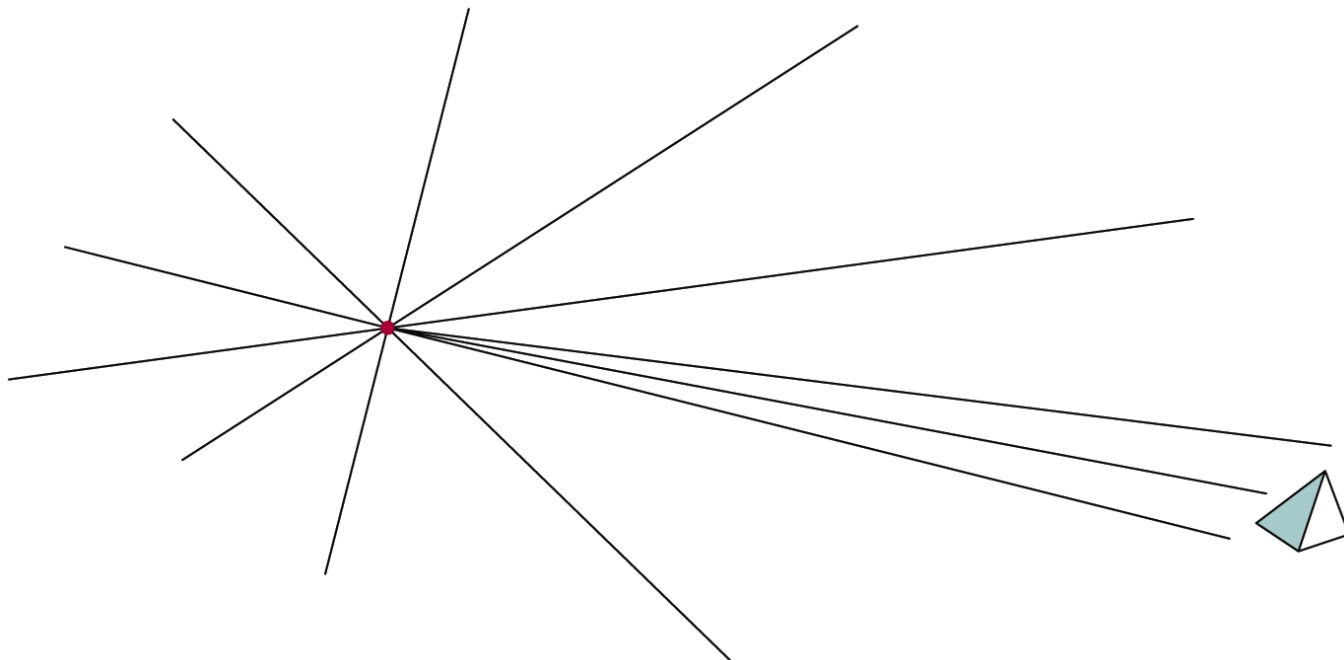
where the coefficients  $a_0$ ,  $a_1$ , and  $a_2$  can be varied to produce optimal results

# Infinitely Distant Light Sources

A large light source, like the sun, can be modelled as a point light source

However, it will have very little directional effect

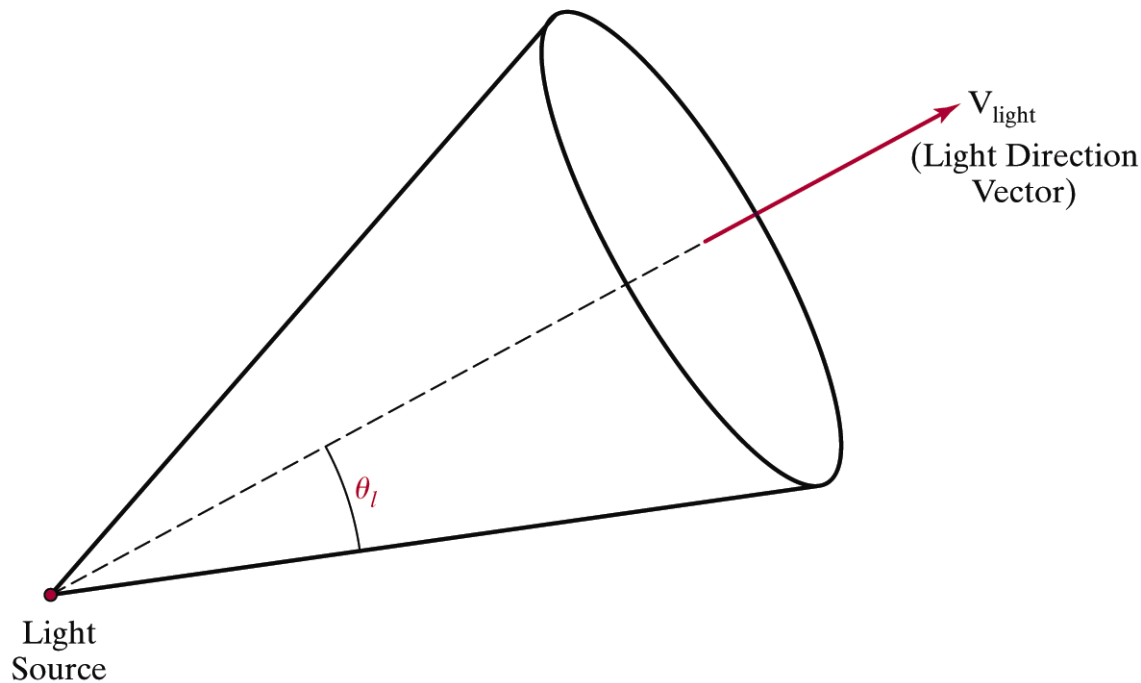
Radial intensity attenuation is not used





# Directional Light Sources & Spotlights

To turn a point light source into a spotlight we simply add a vector direction and an angular limit  $\theta_l$

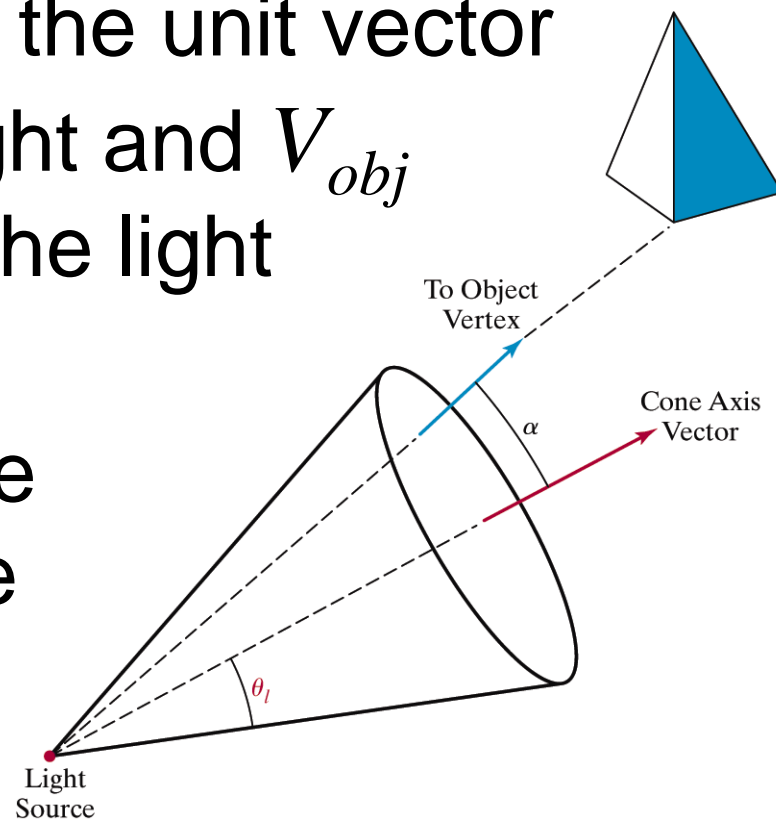


# Directional Light Sources & Spotlights (cont...)

We can denote  $V_{light}$  as the unit vector in the direction of the light and  $V_{obj}$  as the unit vector from the light source to an object

The dot-product of these two vectors gives us the angle between them

$$V_{obj} \cdot V_{light} = \cos \alpha$$



If this angle is inside the light's angular limit then the object is within the spotlight

# Angular Intensity Attenuation

As well as light intensity decreasing as we move away from a light source, it also decreases angularly

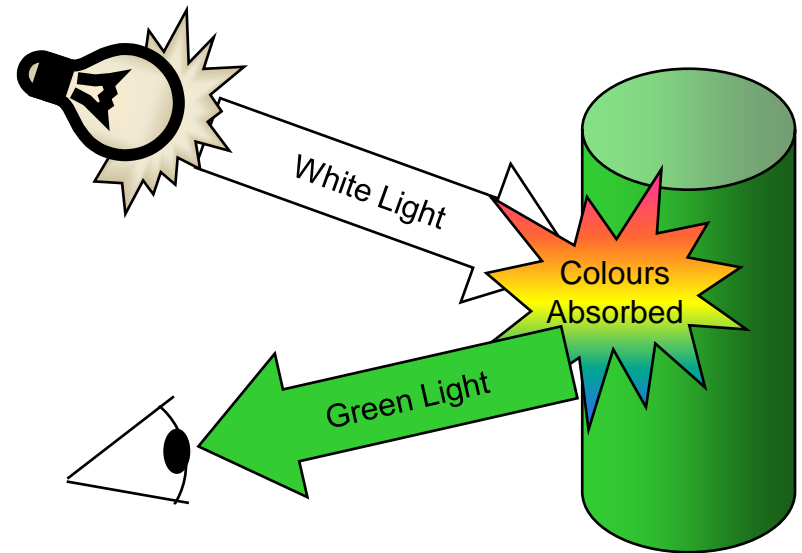
A commonly used function for calculating angular attenuation is:

$$f_{angatten}(\phi) = \cos^{a_l} \phi \quad 0^\circ \leq \phi \leq \theta$$

where the attenuation exponent  $a_l$  is assigned some positive value and angle  $\phi$  is measured from the cone axis

The colours that we perceive are determined by the nature of the light reflected from an object

For example, if white light is shone onto a green object most wavelengths are absorbed, while green light is reflected from the object



# Surface Lighting Effects

The amount of incident light reflected by a surface depends on the type of material

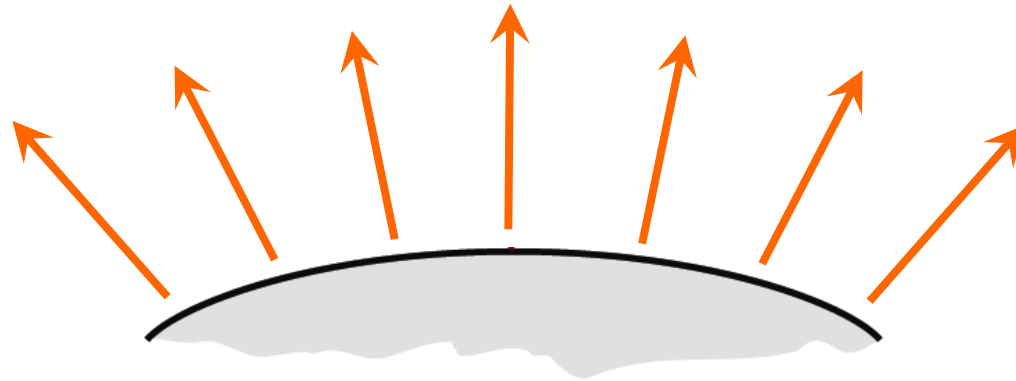
Shiny materials reflect more of the incident light and dull surfaces absorb more of the incident light

For transparent surfaces some of the light is also transmitted through the material

# Diffuse Reflection

Surfaces that are rough or grainy tend to reflect light in all directions

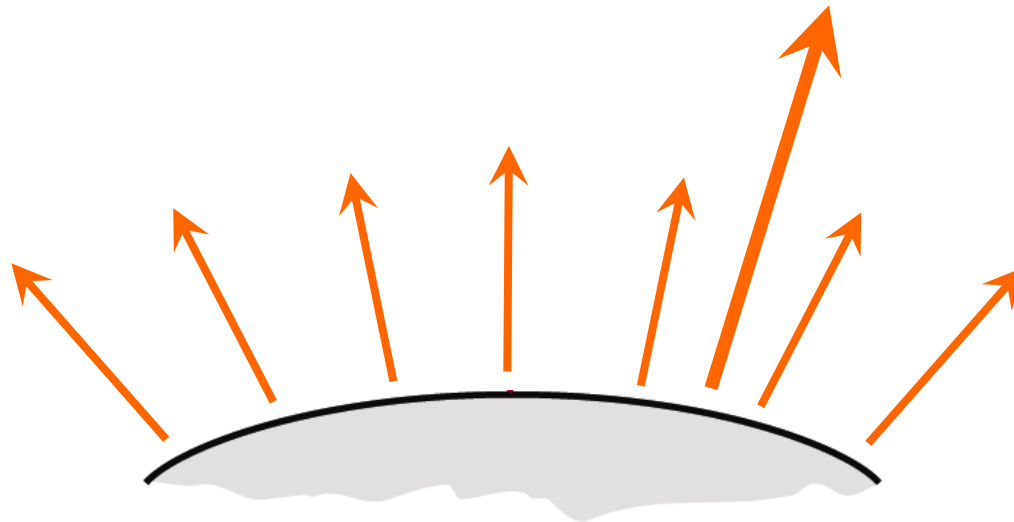
This scattered light is called **diffuse reflection**



# Specular Reflection

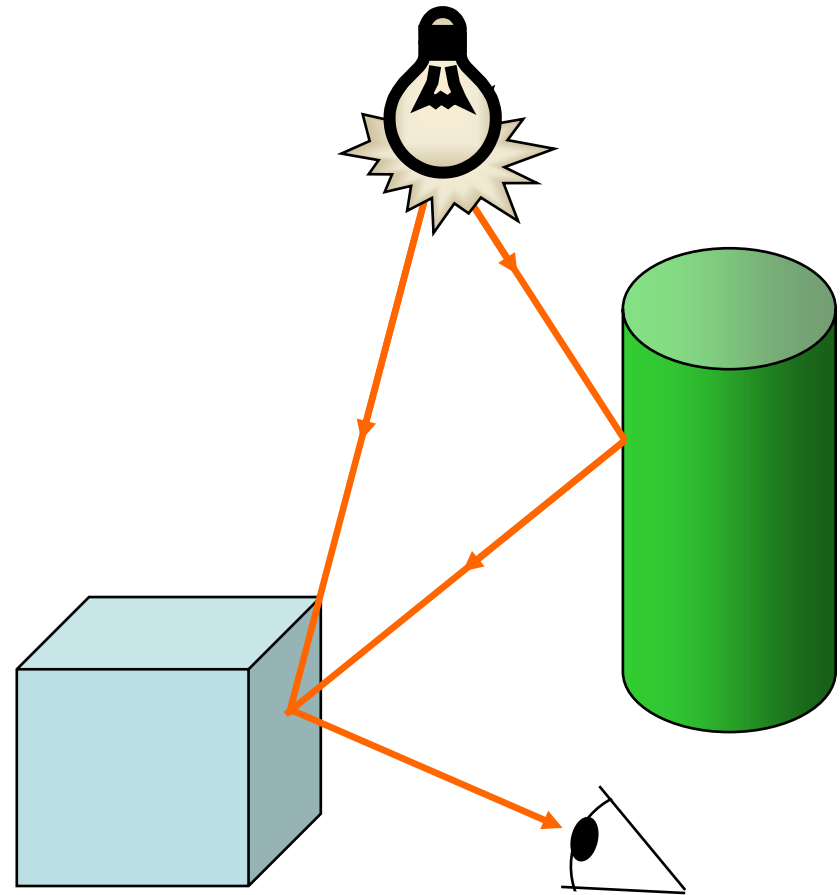
Additionally to diffuse reflection some of the reflected light is concentrated into a highlight or bright spot

This is called **specular reflection**

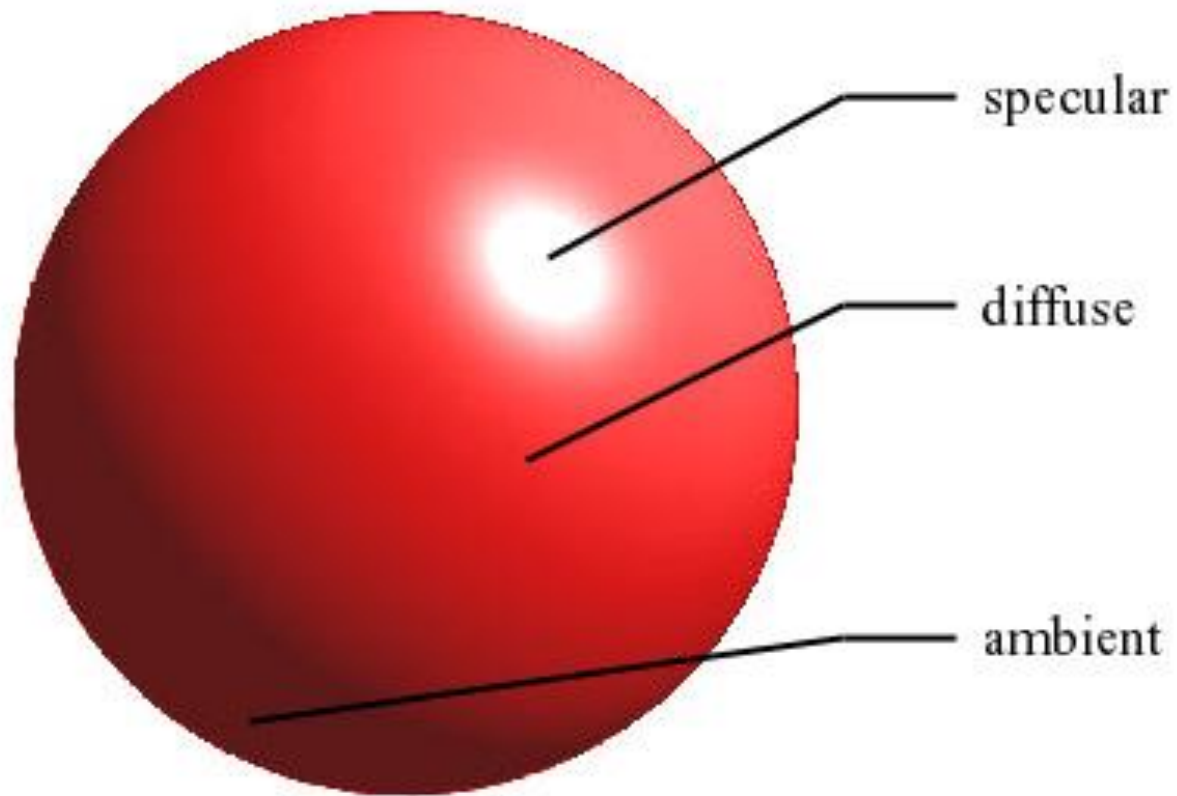


A surface that is not exposed to direct light may still be lit up by reflections from other nearby objects – **ambient light**

The total reflected light from a surface is the sum of the contributions from light sources and reflected light



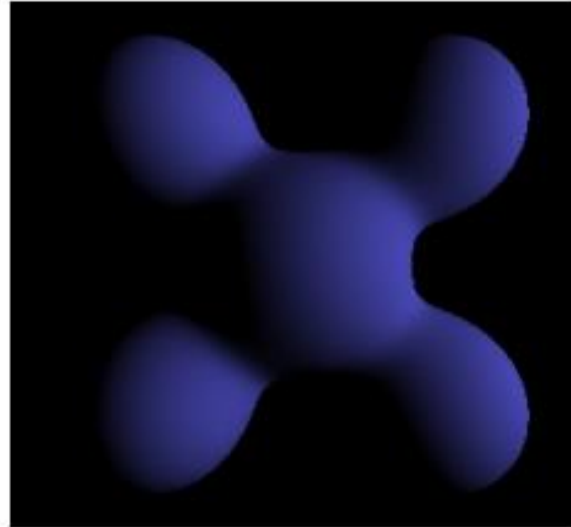




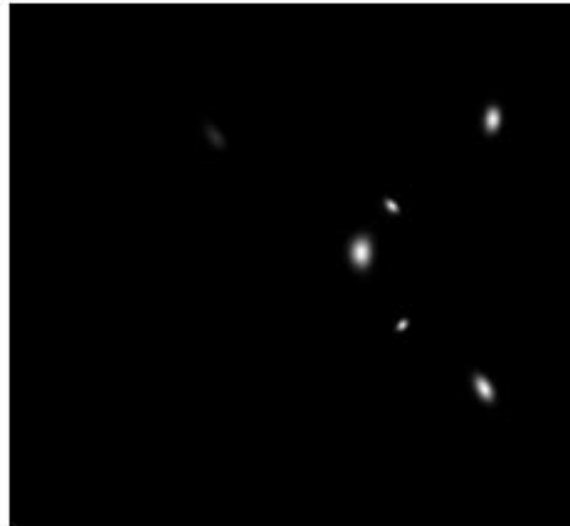
Ambient



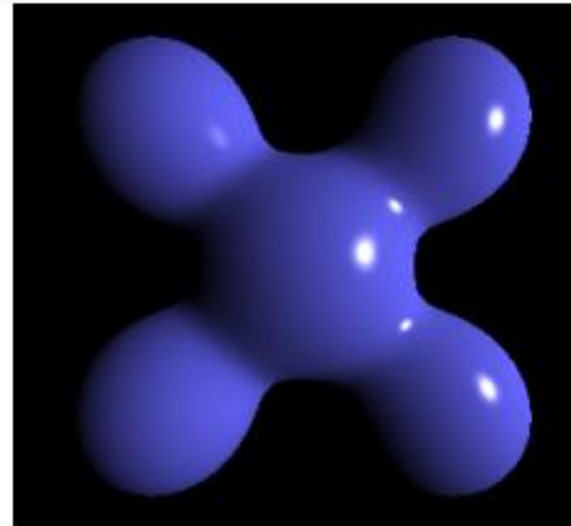
Diffuse



Specular




Final  
Image



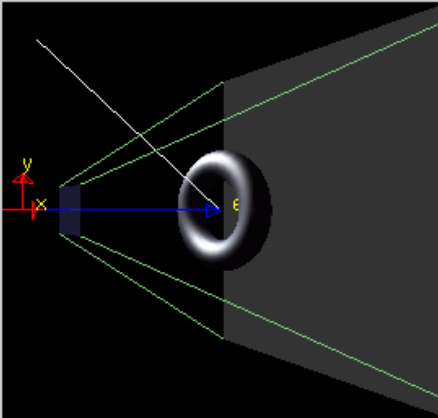
# Nate Robin's Tutorial

Light & Material

Screen-space view

A 3D rendering of a torus (donut shape) with a smooth, reflective surface, centered in a dark gray square view.

World-space view

A 3D rendering of the same torus in a wireframe-like perspective, showing its position relative to a coordinate system with red, green, and blue axes. The torus is positioned on a gray plane.

Command manipulation window

```
GLfloat light_pos[ ] = { -2.71 , 3.10 , 2.00 , 1.00 };
GLfloat light_Ka[ ] = { 0.00 , 0.00 , 0.00 , 1.00 };
GLfloat light_Kd[ ] = { 1.00 , 1.00 , 1.00 , 1.00 };
GLfloat light_Ks[ ] = { 1.00 , 1.00 , 1.00 , 1.00 };

glLightfv(GL_LIGHT0, GL_POSITION, light_pos);
glLightfv(GL_LIGHT0, GL_AMBIENT, light_Ka);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_Kd);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_Ks);

GLfloat material_Ka[ ] = { 0.11 , 0.06 , 0.11 , 1.00 };
GLfloat material_Kd[ ] = { 0.44 , 0.47 , 0.54 , 1.00 };
GLfloat material_Ks[ ] = { 0.99 , 1.00 , 0.98 , 1.00 };
GLfloat material_Ke[ ] = { 0.00 , 0.00 , 0.00 , 0.00 };
GLfloat material_Se = 10 ;

glMaterialfv(GL_FRONT, GL_AMBIENT, material_Ka);
glMaterialfv(GL_FRONT, GL_DIFFUSE, material_Kd);
glMaterialfv(GL_FRONT, GL_SPECULAR, material_Ks);
glMaterialfv(GL_FRONT, GL_EMISSION, material_Ke);
glMaterialfv(GL_FRONT, GL_SHININESS, material_Se);
```

Click on the arguments and move the mouse to modify values.

# Basic Illumination Model

We will consider a basic illumination model which gives reasonably good results and is used in most graphics systems

The important components are:

- Ambient light
- Diffuse reflection
- Specular reflection

For the most part we will consider only monochromatic light

To incorporate background light we simply set a general brightness level for a scene

This approximates the global diffuse reflections from various surfaces within the scene

We will denote this value as  $I_a$

First we assume that surfaces reflect incident light with equal intensity in all directions

Such surfaces are referred to as **ideal diffuse reflectors** or **Lambertian reflectors**

# Diffuse Reflection (cont...)

A parameter  $k_d$  is set for each surface that determines the fraction of incident light that is to be scattered as diffuse reflections from that surface

This parameter is known as the **diffuse-reflection coefficient** or the **diffuse reflectivity**

$k_d$  is assigned a value between 0.0 and 1.0

- 0.0: dull surface that absorbs almost all light
- 1.0: shiny surface that reflects almost all light

# Diffuse Reflection – Ambient Light

For background lighting effects we can assume that every surface is fully illuminated by the scene's ambient light  $I_a$

Therefore the ambient contribution to the diffuse reflection is given as:

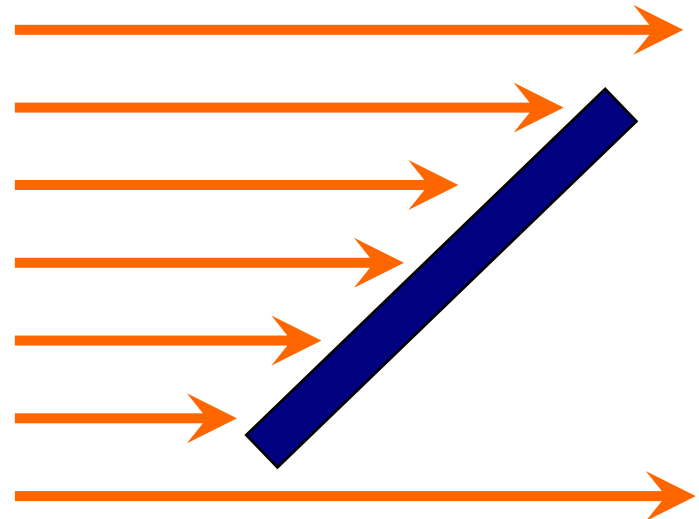
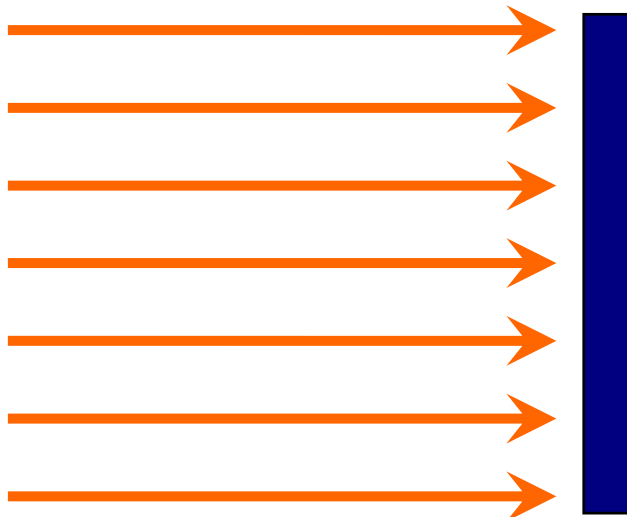
$$I_{ambdiff} = k_d I_a$$

Ambient light alone is very uninteresting so we need some other lights in a scene as well



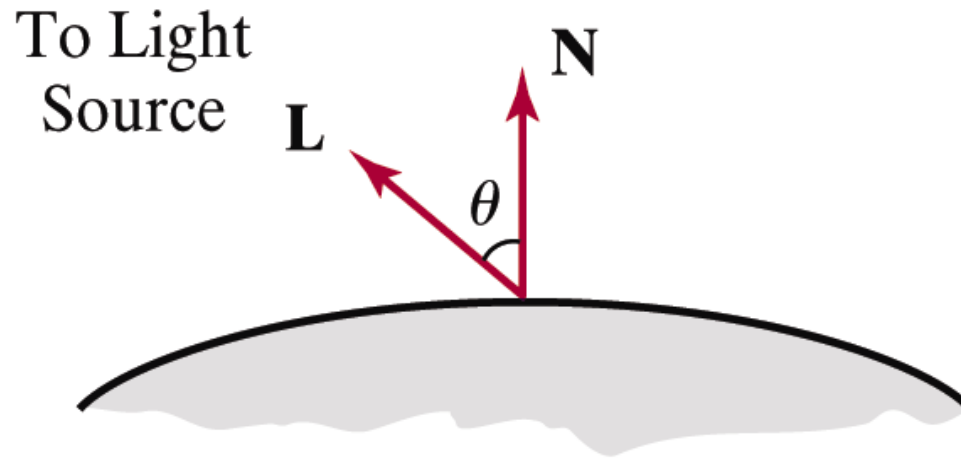
# Diffuse Reflection (cont...)

When a surface is illuminated by a light source, the amount of incident light depends on the orientation of the surface relative to the light source direction



# Diffuse Reflection

The angle between the incoming light direction and a surface normal is referred to as the **angle of incidence** given as  $\theta$



# Diffuse Reflection (cont...)

So the amount of incident light on a surface is given as:

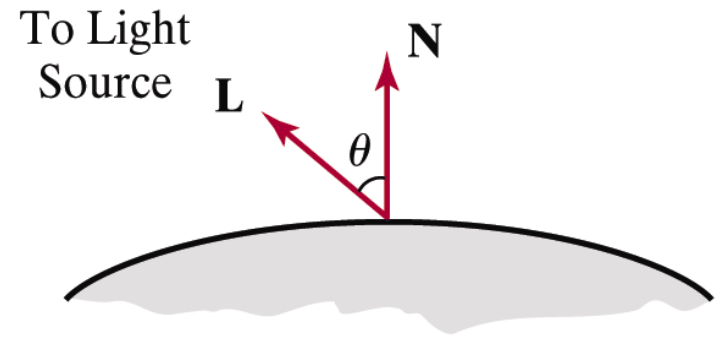
$$I_{l,incident} = I_l \cos \theta$$

So we can model the diffuse reflections as:

$$\begin{aligned} I_{l,diff} &= k_d I_{l,incident} \\ &= k_d I_l \cos \theta \end{aligned}$$

# Diffuse Reflection (cont...)

Assuming we denote the normal for a surface as  $N$  and the unit direction vector to the light source as  $L$  then:



$$N \cdot L = \cos \theta$$

So:

$$I_{l,diff} = \begin{cases} k_d I_l (N \cdot L) & \text{if } N \cdot L > 0 \\ 0 & \text{if } N \cdot L \leq 0 \end{cases}$$

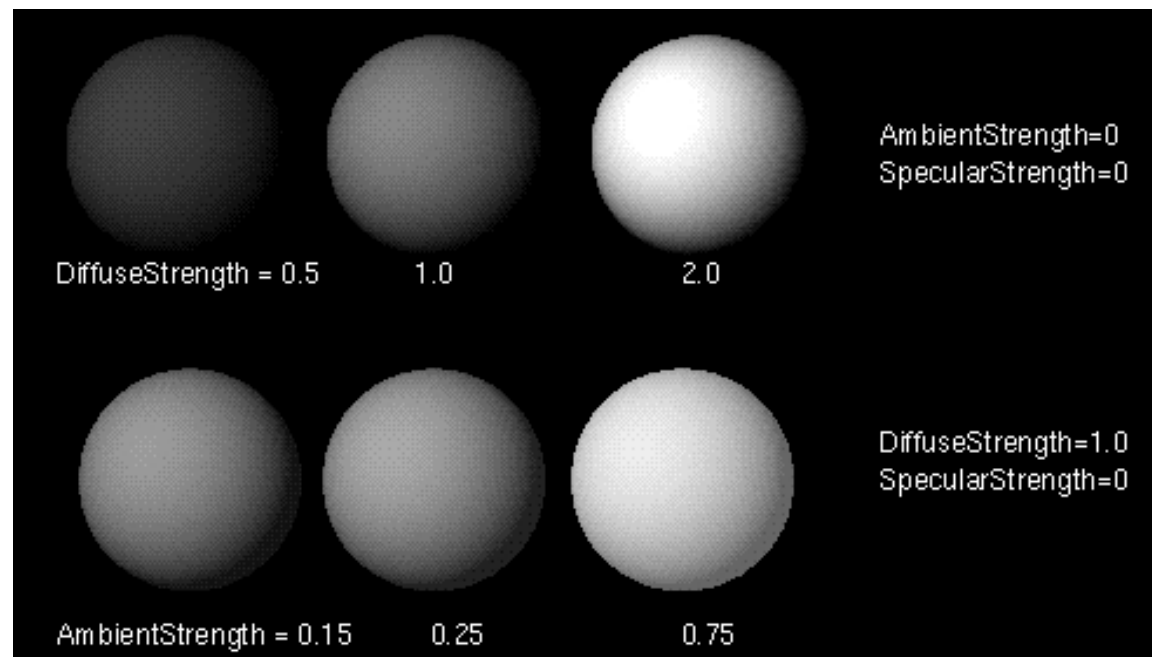
# Combining Ambient And Incident Diffuse Reflections

To combine the diffuse reflections arising from ambient and incident light most graphics packages use two separate diffuse-reflection coefficients:

- $k_a$  for ambient light
- $k_d$  for incident light

The total diffuse reflection equation for a single point source can then be given as:

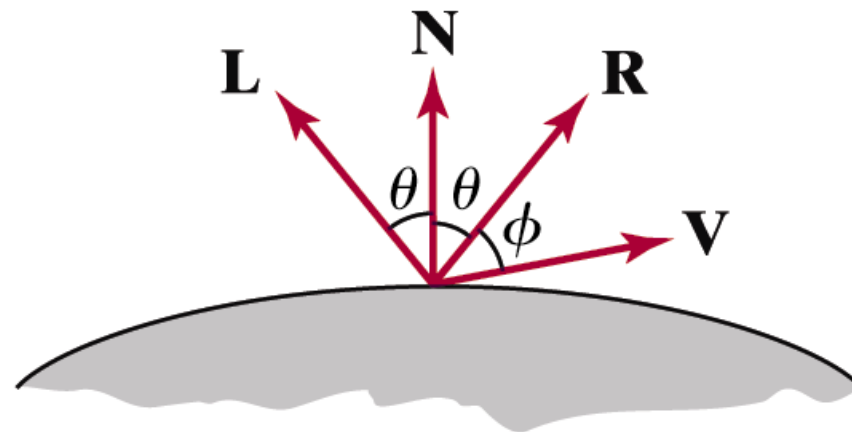
$$I_{diff} = \begin{cases} k_a I_a + k_d I_l (N \cdot L) & \text{if } N \cdot L > 0 \\ k_a I_a & \text{if } N \cdot L \leq 0 \end{cases}$$



# Specular Reflection

The bright spot that we see on a shiny surface is the result of near total of the incident light in a concentrated region around the **specular reflection angle**

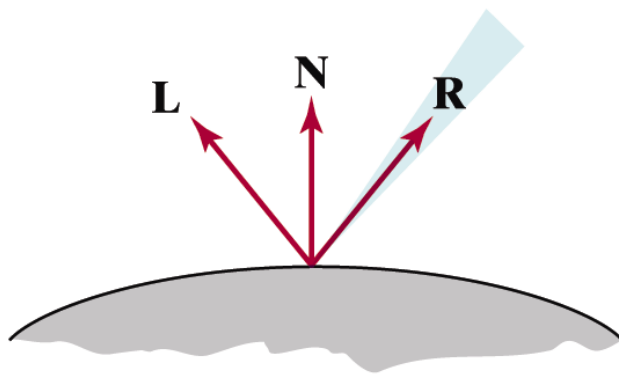
The specular reflection angle equals the angle of the incident light



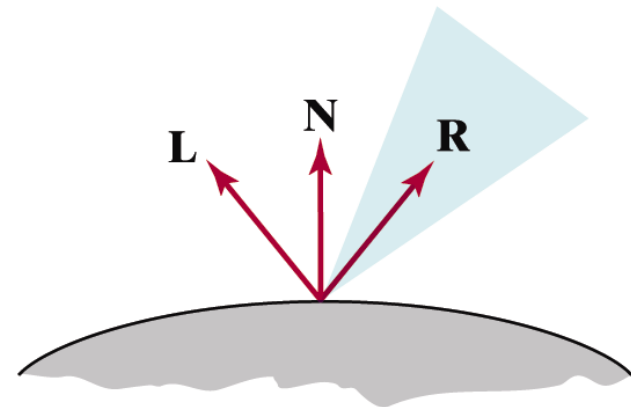
# Specular Reflection (cont...)

A perfect mirror reflects light only in the specular-reflection direction

Other objects exhibit specular reflections over a finite range of viewing positions around vector  $R$



Shiny Surface  
(Large  $n_s$ )



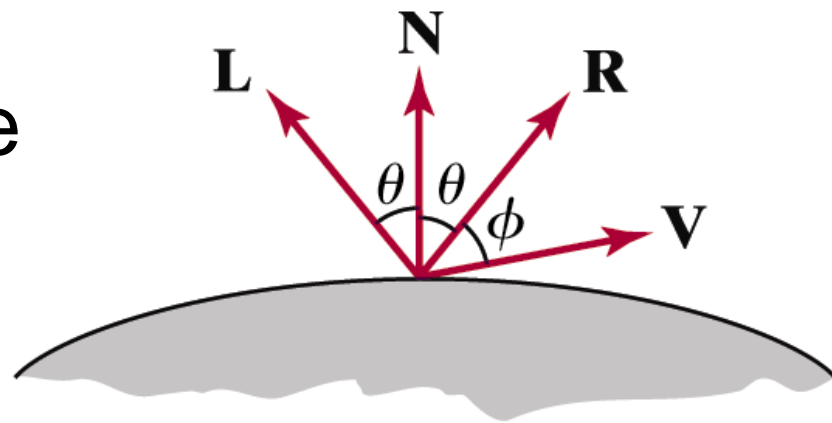
Dull Surface  
(Small  $n_s$ )



# The Phong Specular Reflection Model

The **Phong specular reflection** model or **Phong model** is an empirical model for calculating specular reflection range developed in 1973 by Phong Bui Tuong

The Phong model sets the intensity of specular reflection as proportional to the angle between the viewing vector and the specular reflection vector



# The Phong Specular Reflection Model (cont...)

So, the specular reflection intensity is proportional to  $\cos^{n_s} \phi$

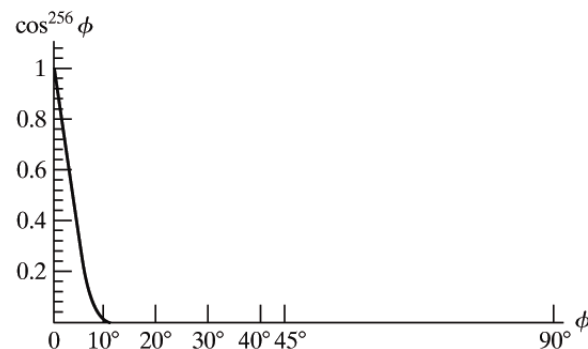
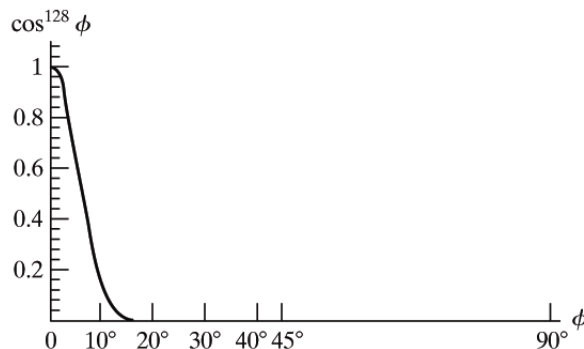
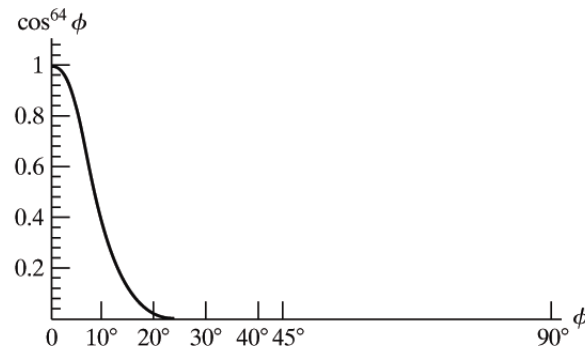
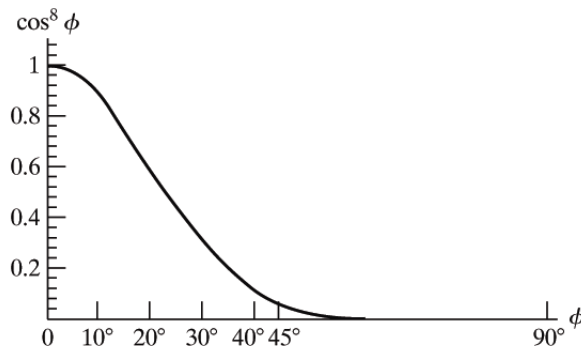
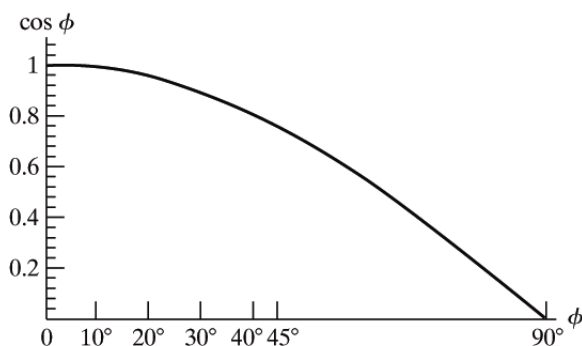
The angle  $\phi$  can be varied between  $0^\circ$  and  $90^\circ$  so that  $\cos\phi$  varies from 1.0 to 0.0

The **specular-reflection exponent**,  $n_s$  is determined by the type of surface we want to display

- Shiny surfaces have a very large value ( $>100$ )
- Rough surfaces would have a value near 1

# The Phong Specular Reflection Model (cont...)

The graphs below show the effect of  $n_s$  on the angular range in which we can expect to see specular reflections



# The Phong Specular Reflection Model (cont...)

For some materials the amount of specular reflection depends heavily on the angle of the incident light

*Fresnel's Laws of Reflection* describe in great detail how specular reflections behave

However, we don't need to worry about this and instead approximate the specular effects with a constant specular reflection coefficient  $\underline{k_s}$

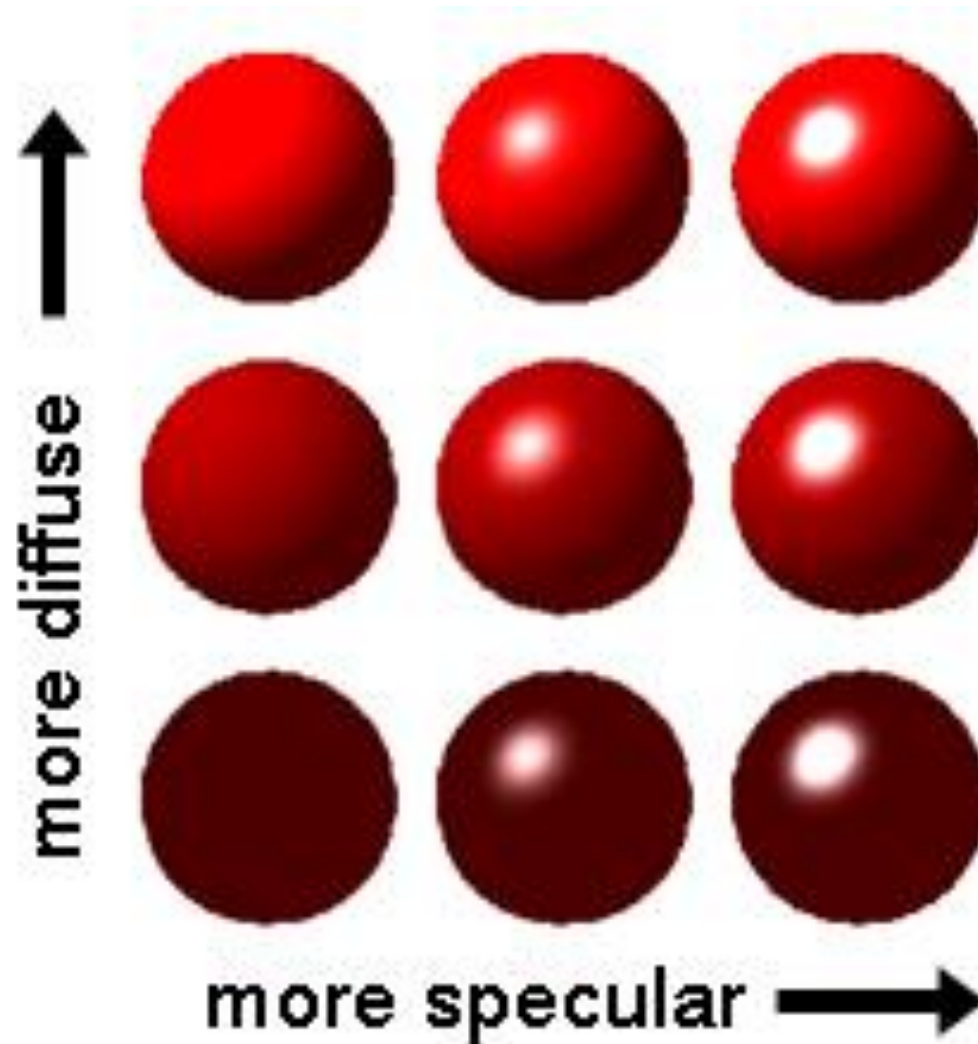
# The Phong Specular Reflection Model (cont...)

So the specular reflection intensity is given as:

$$I_{l,spec} = k_s I_l \cos^{n_s} \phi$$

Remembering that  $V \cdot R = \cos \phi$  we can say:

$$I_{l,spec} = \begin{cases} k_s I_l (V \cdot R)^{n_s} & \text{if } V \cdot R > 0 \text{ and } N \cdot L > 0 \\ 0.0 & \text{if } V \cdot R < 0 \text{ or } N \cdot L \leq 0 \end{cases}$$



# Combining Diffuse & Specular Reflections

For a single light source we can combine the effects of diffuse and specular reflections simply as follows:

$$\begin{aligned} I &= I_{diff} + I_{spec} \\ &= k_a I_a + k_d I_l (N \cdot L) + k_s I_l (V \cdot R)^{n_s} \end{aligned}$$

# Diffuse & Specular Reflections From Multiple Light Sources

We can place any number of light sources in a scene

We compute the diffuse and specular reflections as sums of the contributions from the various sources

$$\begin{aligned} I &= I_{ambdiff} + \sum_{l=1}^n [I_{l,diff} + I_{l,spec}] \\ &= k_a I_a + \sum_{l=1}^n I_l [k_d (N \cdot L) + k_s (V \cdot R)^{n_s}] \end{aligned}$$

Common  
Exam Question



# Adding Intensity Attenuation

To incorporate radial and angular intensity attenuation into our model we simply adjust our equation to take these into account

So, light intensity is now given as:

$$I = I_{ambdiff} + \sum_{l=1}^n \left[ f_{l,radatten} f_{l,angatten} (I_{l,diff} + I_{l,spec}) \right]$$

where  $f_{radatten}$  and  $f_{angatten}$  are as discussed previously

# RGB Colour Considerations

For an RGB colour description each intensity specification is a three element vector

So, for each light source:

$$I_l = (I_{lR}, I_{lG}, I_{lB})$$

Similarly all parameters are given as vectors:

$$k_a = (k_{aR}, k_{aG}, k_{aB}) \quad k_d = (k_{dR}, k_{dG}, k_{dB})$$

$$k_s = (k_{sR}, k_{sG}, k_{sB})$$

# RGB Colour Considerations (cont...)

Each component of the surface colour is then calculated with a separate expression

For example:

$$I_{lR,diff} = k_{dR} I_{lR} (N \cdot L)$$

$$I_{lG,diff} = k_{dG} I_{lG} (N \cdot L)$$

$$I_{lB,diff} = k_{dB} I_{lB} (N \cdot L)$$

To create realistic (or even semi-realistic) looking scenes we must model light correctly

To successfully model lighting effects we need to consider:

- Ambient light
- Diffuse reflections
- Specular reflections