# Shading

# Reading

**Required**

- Foley, Section16.1

**Optional**

- Hearn & Baker, sections 14.1,14.2,14.5

# Introduction

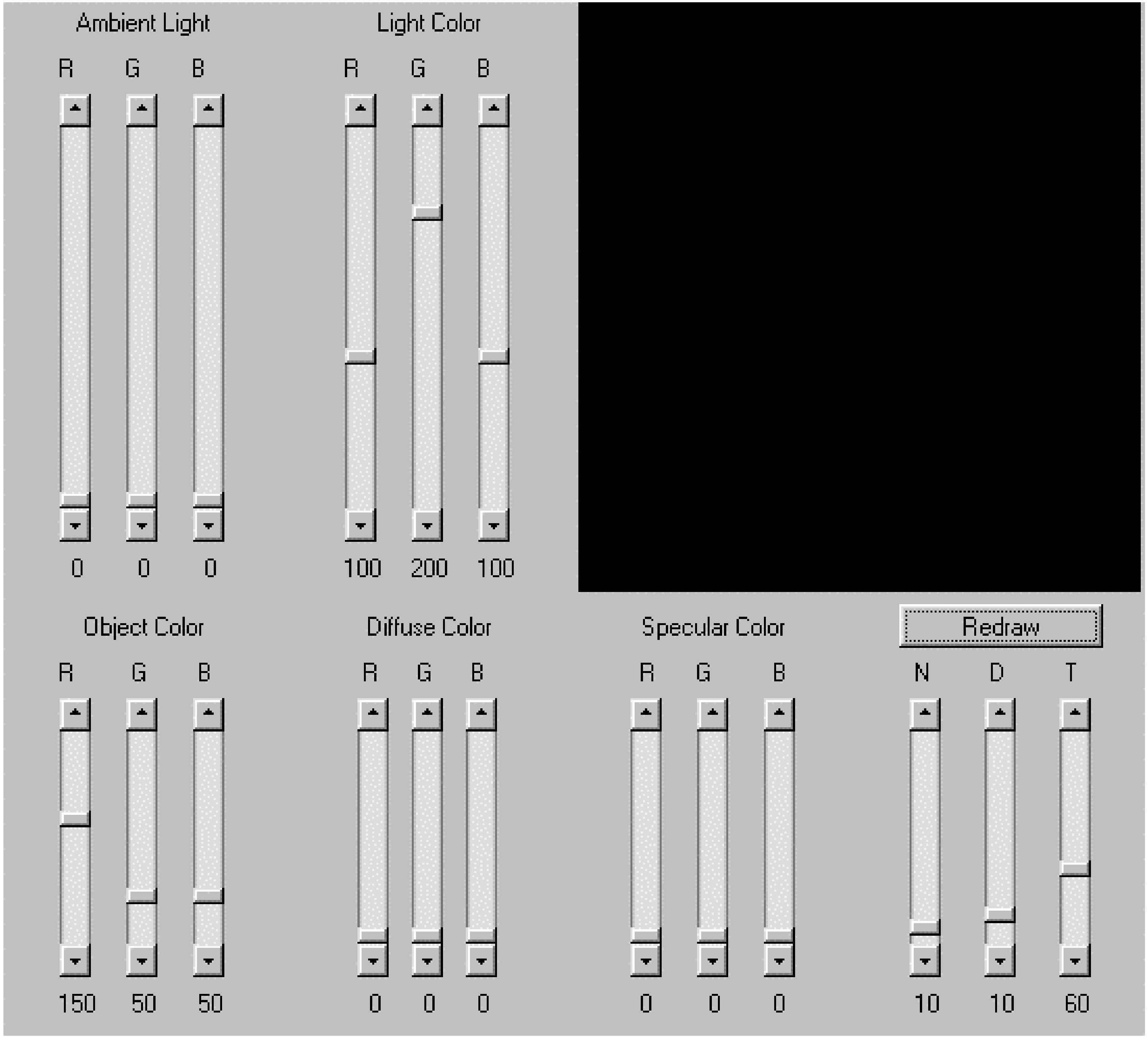
- So far, we’ve talked exclusively about geometry.
  - What is the shape of an object?
  - How do I place it in a virtual 3D space?
  - How do I know which pixels it covers?
  - How do I know which of the pixels I should actually draw?
- Once we’ve answered all those, we have to ask one more important question:
  - What value do I set each pixel to?
- Answering this question is the job of the **shading model**.
- (Of course, people also call it a lighting model, a light reflection model, a local illumination model, a reflectance model, etc., etc.)

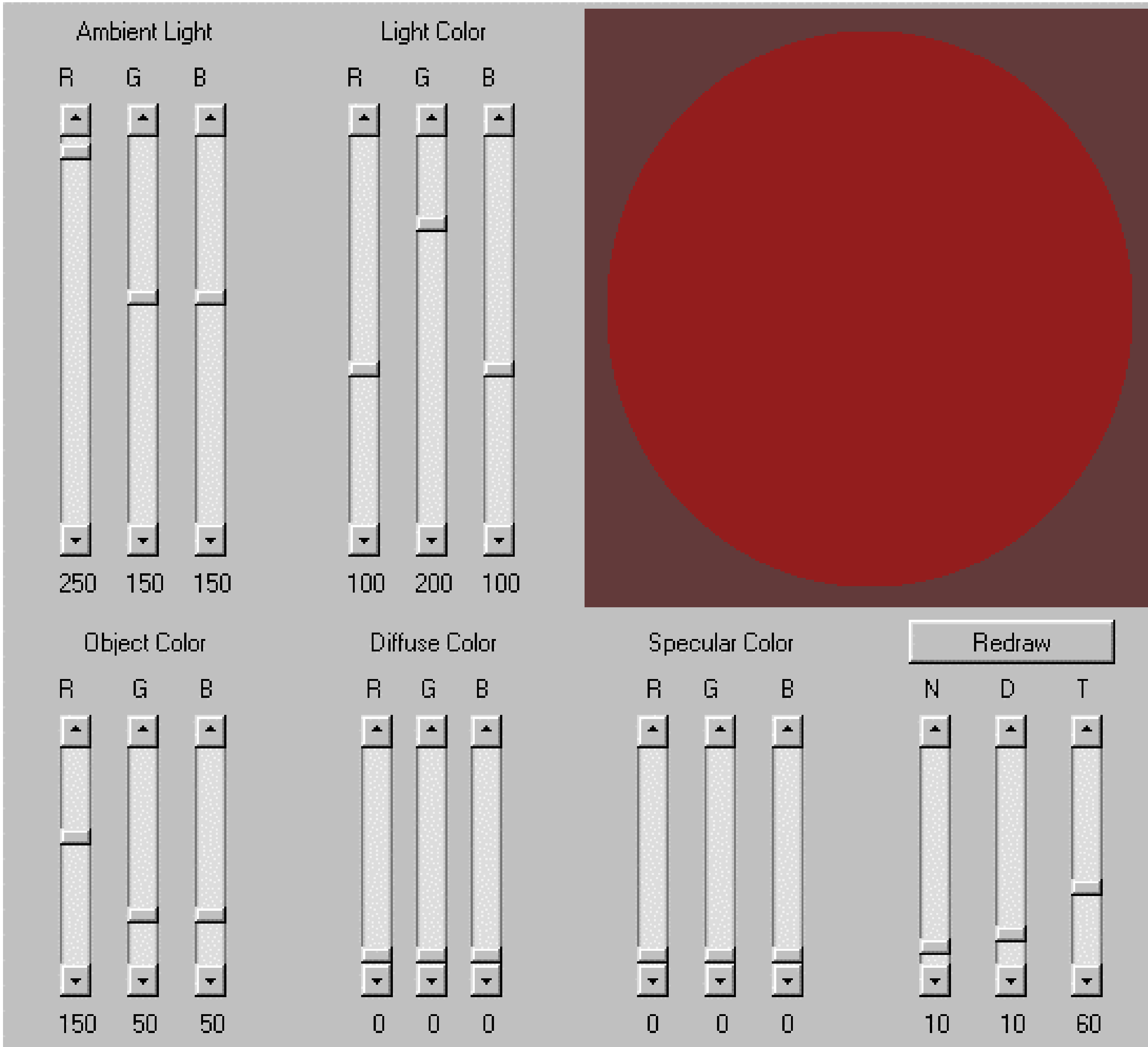
# Tedious Reality

- Properly determining the right color is *really hard.*
- Look around the room. Each light source has different characteristics. Trillions of photons are pouring out every second.
- These photons can:
  - interact with the atmosphere, or with things in the atmosphere
  - strike a surface and
    - be absorbed
    - be reflected
    - cause fluorescence or phosphorescence
- of course, none of the surfaces in here are perfect spheres or cylinders. At some microscopic level (very important for photons) they're all really bumpy.
- also, everything depends on wavelength.

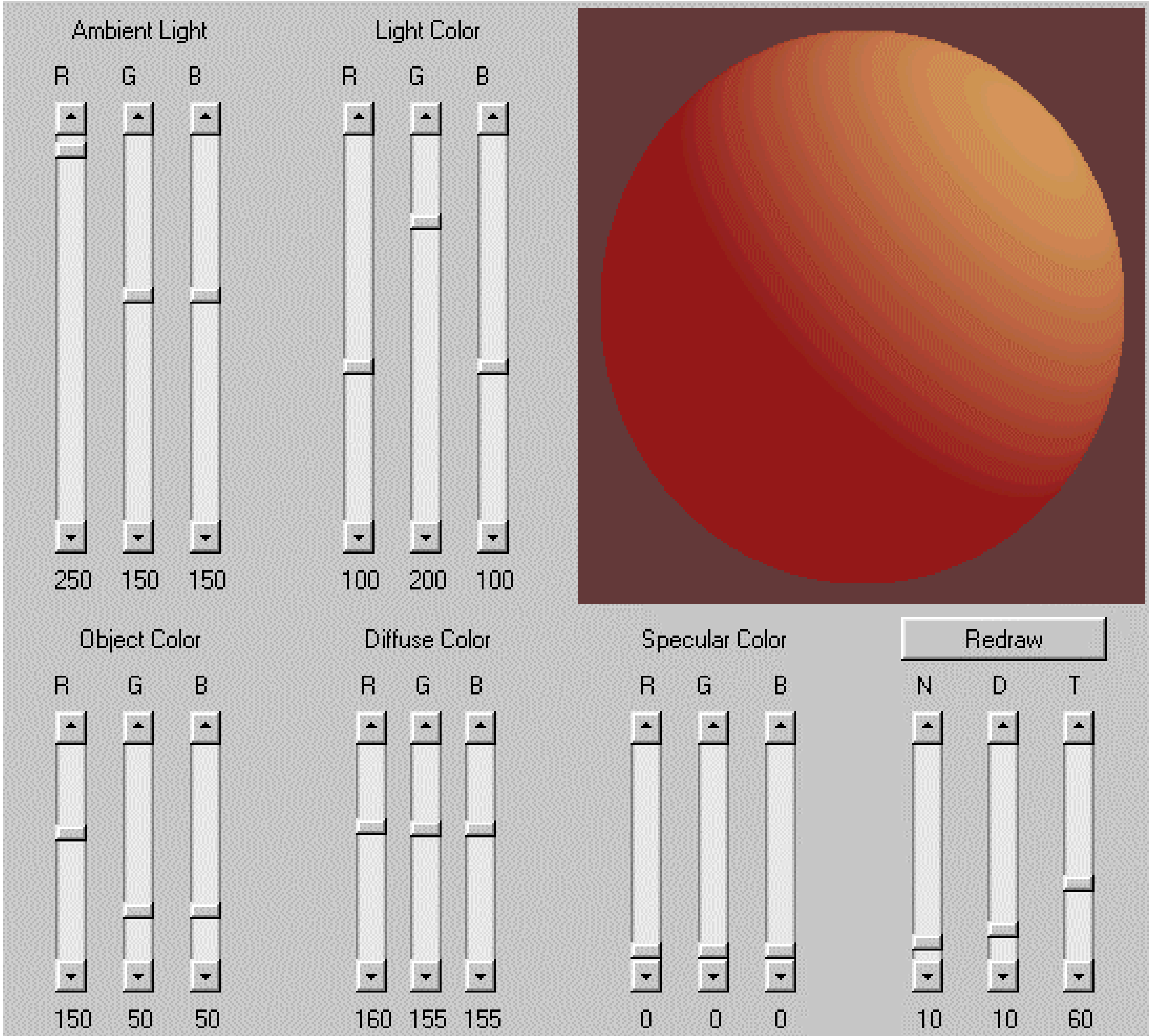
# Our Problem

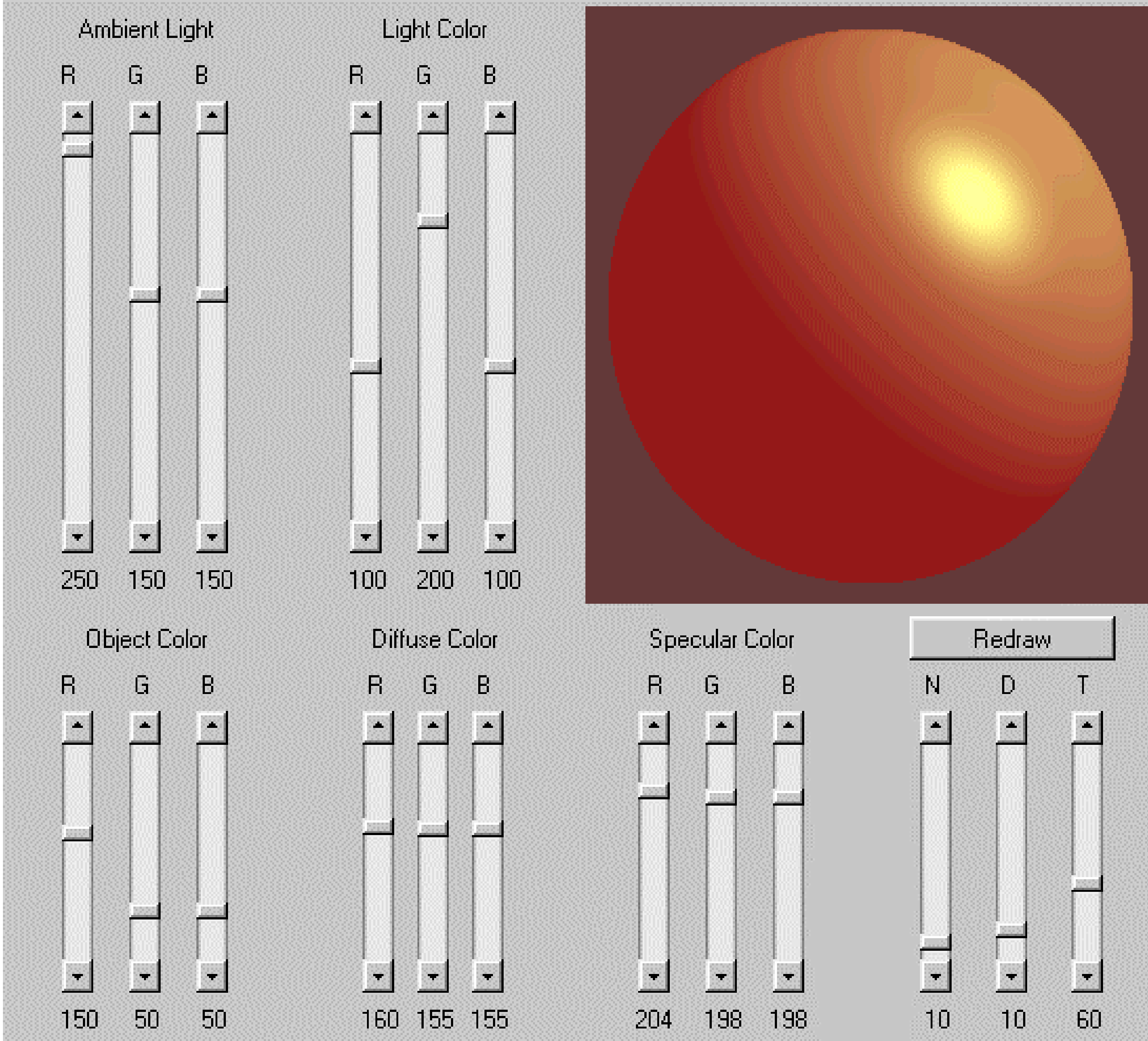
- We're going to build up to an *approximation* of reality called the **Phong illumination model**.
- It has the following characteristics:
  - *not* physically based
  - gives a first-order *approximation* to physical light reflection
  - very fast
  - widely used

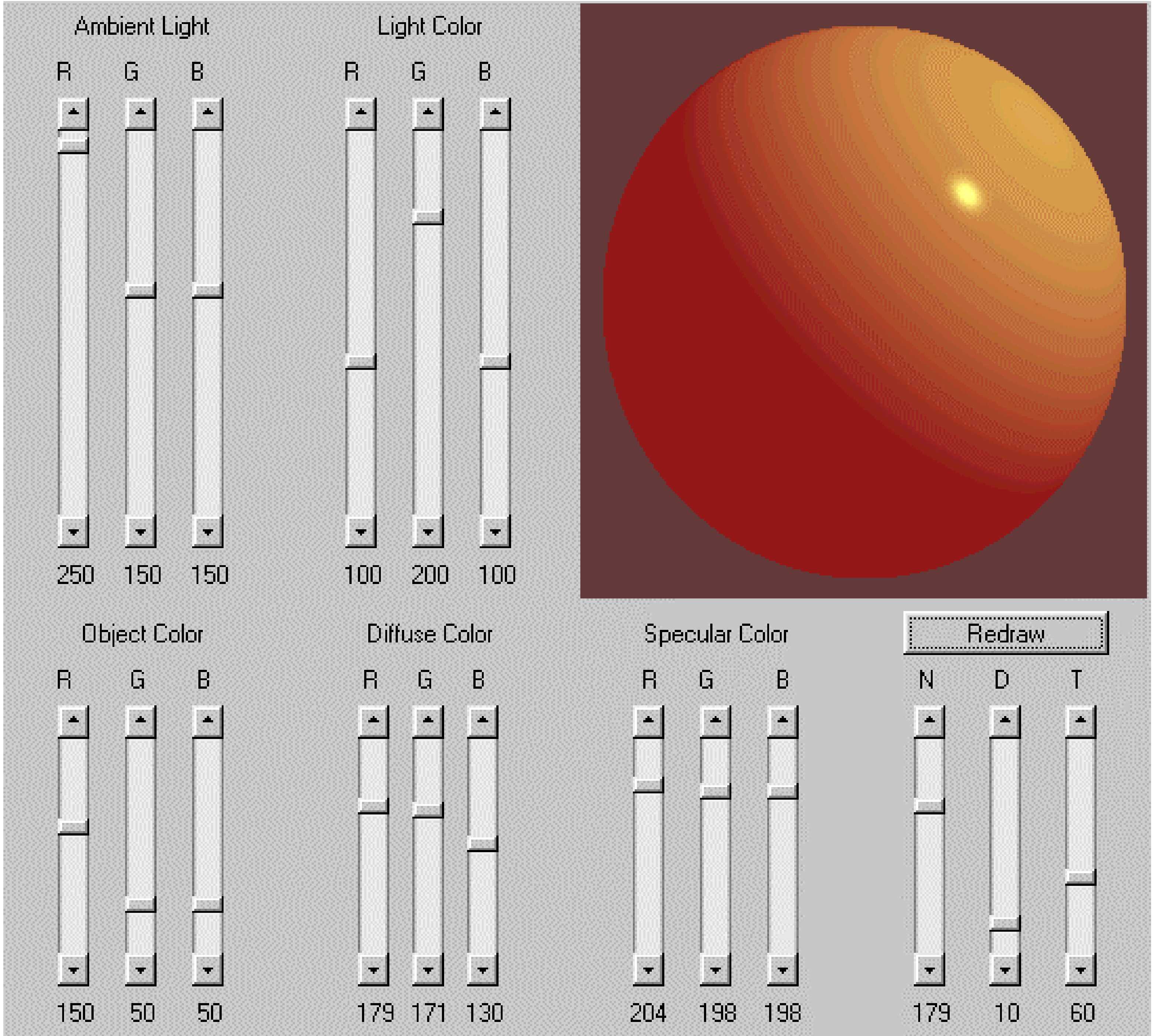

Ambient Light
R G B
0 0 0
Light Color
R G B
100 200 100
Object Color
R G B
150 50 50
Diffuse Color
R G B
0 0 0
Specular Color
R G B
0 0 0
Redraw
N D T
10 10 60



Ambient Light
R G B
250 150 150
Light Color
R G B
100 200 100
Object Color
R G B
150 50 50
Diffuse Color
R G B
0 0 0
Specular Color
R G B
0 0 0
Redraw
N D T
10 10 60


Ambient Light
R G B
250 150 150
Light Color
R G B
100 200 100
Object Color
R G B
150 50 50
Diffuse Color
R G B
160 155 155
Specular Color
R G B
0 0 0
Redraw
N D T
10 10 60


Ambient Light
R G B
250 150 150
Light Color
R G B
100 200 100
Object Color
R G B
150 50 50
Diffuse Color
R G B
160 155 155
Specular Color
R G B
204 198 198
Redraw
N D T
10 10 60


Ambient Light
R G B
250 150 150
Light Color
R G B
100 200 100
Object Color
R G B
150 50 50
Diffuse Color
R G B
179 171 130
Specular Color
R G B
204 198 198
Redraw
N D T
179 10 60

# Iteration Zero

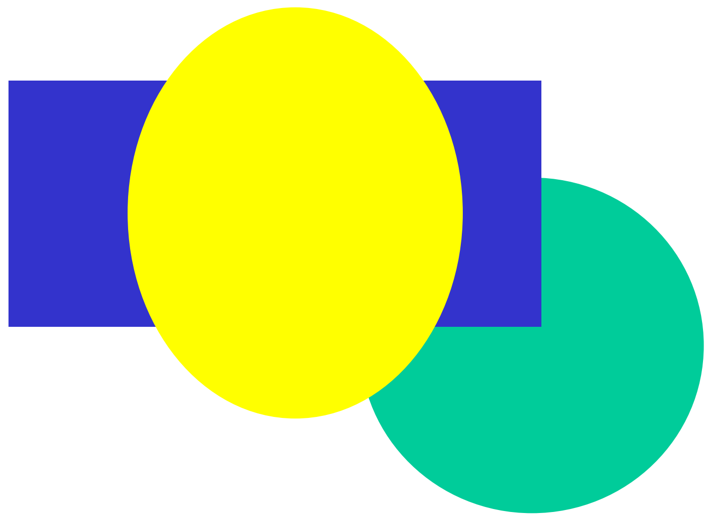
- Given:
  - a point $P$ on a surface ($P$ is determined by ray-object intersection, for instance)
  - visible through pixel $p$
- Assign each polygon a single color:

$$I = k_e$$

where

  - $I$ is the resulting intensity
  - $k_e$ is the intrinsic shade associated with the object
- This has some special-purpose uses, but not really good for drawing a scene.

# What will it like?



# Iteration One

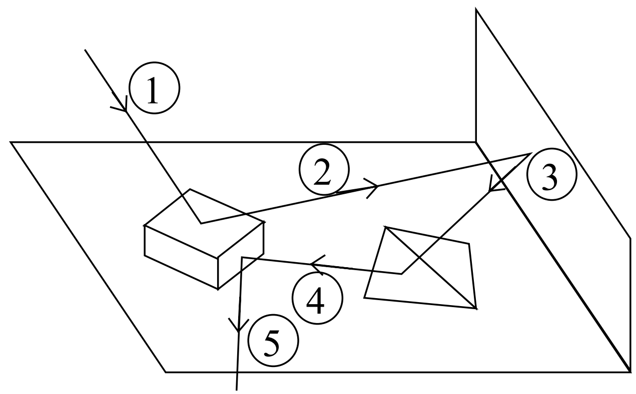
- Let’s make the color at least dependent on the overall quantity of light available in the scene:

$$I = k_a I_a$$

- $k_a$ is the **ambient reflection coefficient**.
  - really the reflectance of ambient light
  - “ambient” light is assumed to be equal in all directions
- $I_a$ is the **ambient intensity**.
- Physically, what is “ambient” light?
  - Answer on next page.

# Indirect Illumination (Ambient)

- Some surfaces are illuminated even it is in shadow. Why?
- There is indirect lighting (background lighting) reflected from other surfaces




- Each surface illuminated becomes itself a source of light for illumination on other surfaces of the scene.
- Each of these surfaces, in turn, reflect light to other surfaces, including the original one, thus achieving an "infinite regression" of reflections and illumination.
- Heuristic: Simply assume the indirect lighting is constant (same to all objects in the scene) in most graphics systems.

# Wavelength Dependence

- Really, $k_a$ and $I_a$ are functions over all wavelengths $\kappa$.
- Ideally, we would do the calculation on these functions:

$$I(\kappa) = k_a(\kappa)\, I_a(\kappa)$$

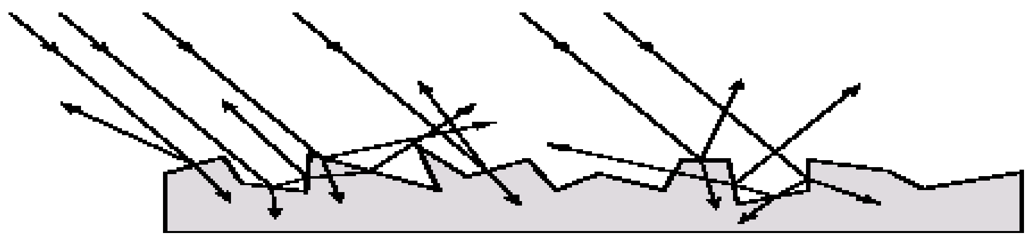
- then we would find good RGB values to represent the spectrum $I_a(\kappa)$.
- Traditionally, though, $k_a$ and $I_a$ are represented as RGB triples, and the computation is performed on each color channel separately.

# Diffuse Reflection

- Let’s examine the ambient shading model:
  - objects have different colors
  - we can control the overall light intensity
    - what happens when we turn off the lights?
    - what happens as the light intensity increases?
    - what happens if we change the color of the lights?
- So far, objects are uniformly lit.
  - not the way things really appear
  - in reality, light sources are directional
- Diffuse, or **Lambertian** reflection will allow reflected intensity to vary with the direction of the light.

# Diffuse Reflector

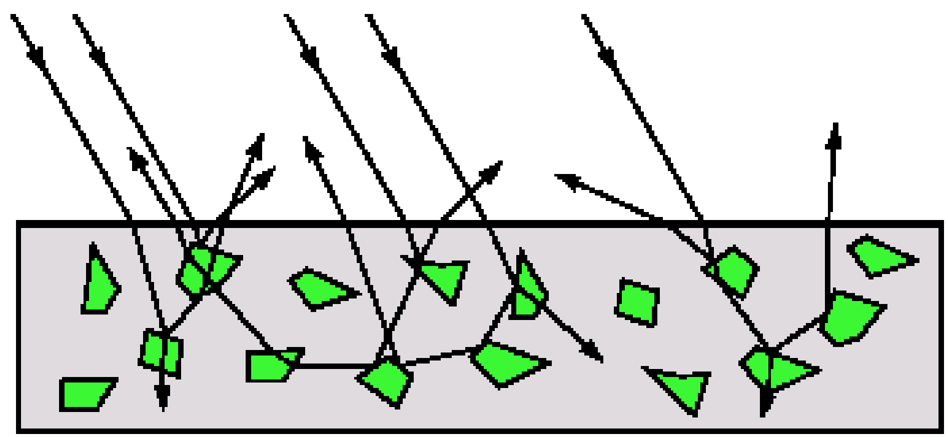
- Diffuse reflection occurs from dull, matte surfaces, like latex paint, or chalk.
- These **diffuse** or **Lambertian** reflectors reradiate light equally in all directions.
- Picture a rough surface with lots of tiny **microfacets**.



- Note:
  - Light may actually penetrate the surface, bounce around, and then reflect back out.
  - Accounts for colorization of diffusely reflected light by plastics.

# Diffuse Reflector

- The reflected intensity from a diffuse surface does not depend on the direction of the viewer. The incoming light, though, does depend on the direction of the light source.



- Q: Why is the North Pole cold? Why is winter cold?

# Iteration Two

- The incoming energy is proportional to cos $\eta$, giving the diffuse reflection equations:

$$\begin{aligned} I &= k_e + k_a I_a + k_d\, I_l \cos\eta \\ &= k_e + k_a I_a + k_d\, I_l\, (\mathbf{N} \cdot \mathbf{L})_+ \end{aligned}$$

- where:
  - $k_d$ is the **diffuse reflection coefficient.**
  - $I_l$ is the intensity of the light source
  - **N** is the normal to the surface (unit vector)
  - **L** is the direction to the light source (unit vector)
  - $(x)_+$ means max $\{0, x\}$
- OpenGL supports different kinds of lights: point, directional, and spot. How do these work?

# Ideal Light Sources

**Light Sources:**

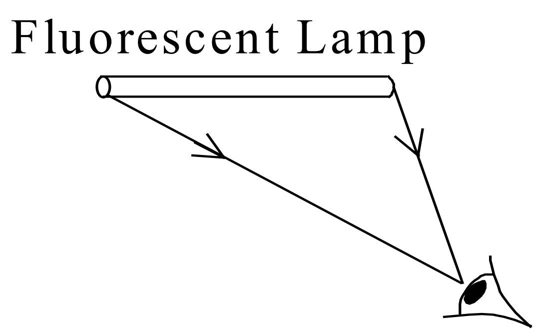
- In computer graphics, two types of light sources are commonly used
  - **<u>point source</u>** The light source is a zero-volume point
  - **<u>directional source</u>** The point source that are infinite far away

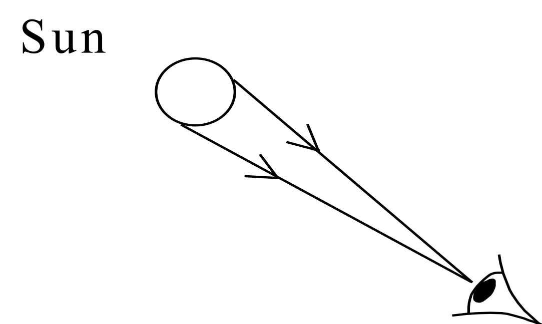

- Both types of light sources are ideal light sources (i.e. not realistic)
- But they are easy for computation.

# Extended Light Sources

- A more realistic class of light source is **extended source.**
- The light source is actually a surface or a volume, not a point.

e.g

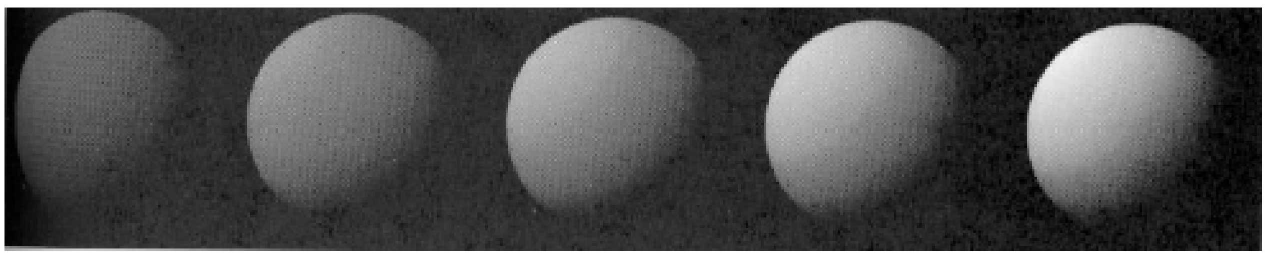


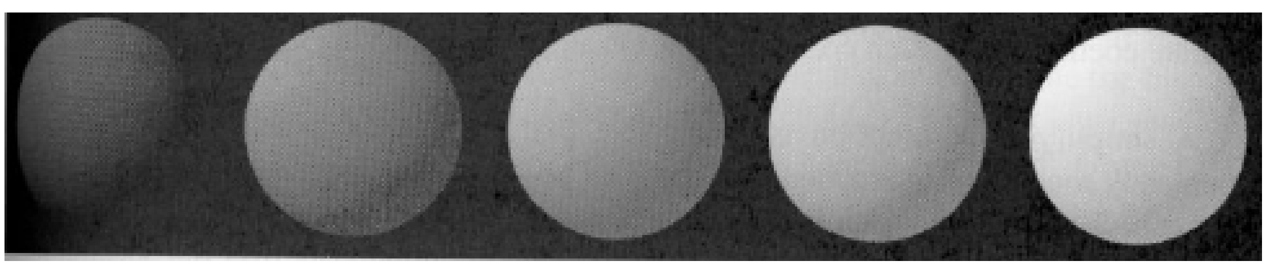

- Sun is really an extended source
- Fluorescent light is a typical extended source.

# Ambient and Diffuse Examples

- Increasing the diffuse coefficient:



- Increasing the ambient term while keeping the diffuse term constant:



# Intensity drop-off with distance

- The laws of physics state that the intensity of a point light source must drop off with its distance squared.
- We can incorporate this effect by multiplying $I_l$ by $1/d^2$.
- Sometimes, this distance-squared drop off is considered too "harsh." Angel suggests using with user-supplied constants for *a*, *b*, and *c*.

$$f(d) = \frac{1}{a + bd + cd^2}$$

- with user-supplied constants for $a$, $b$, and $c$.

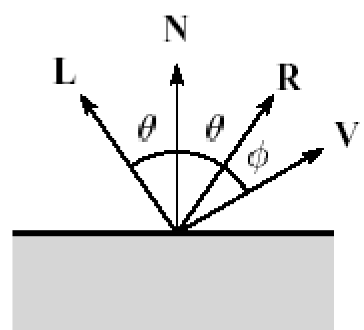
$$f(d) = \min\left(1, \frac{1}{a + bd + cd^2}\right)$$

# Specular Reflection

- **Specular reflection** accounts for the highlight that you see on some objects.
- It is particularly important for *smooth, shiny* surfaces, such as:
  - metal
  - polished stone
  - plastics
  - apples
- Specular reflection depends on the viewing direction $V$. The color is often determined solely by the color of the light.
  - corresponds to absence of internal reflections

# Specular Reflection Derivation

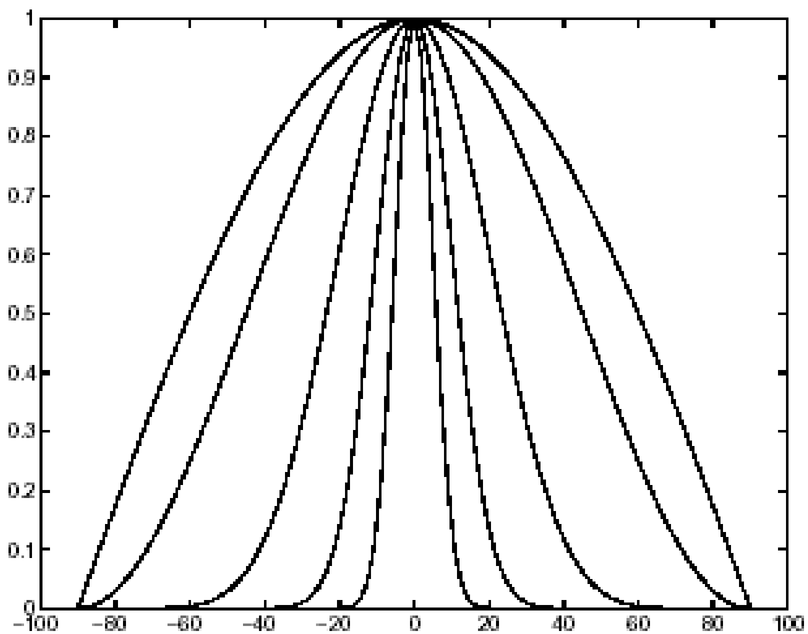
- For a perfect mirror reflector, light is reflected about ***N***, so

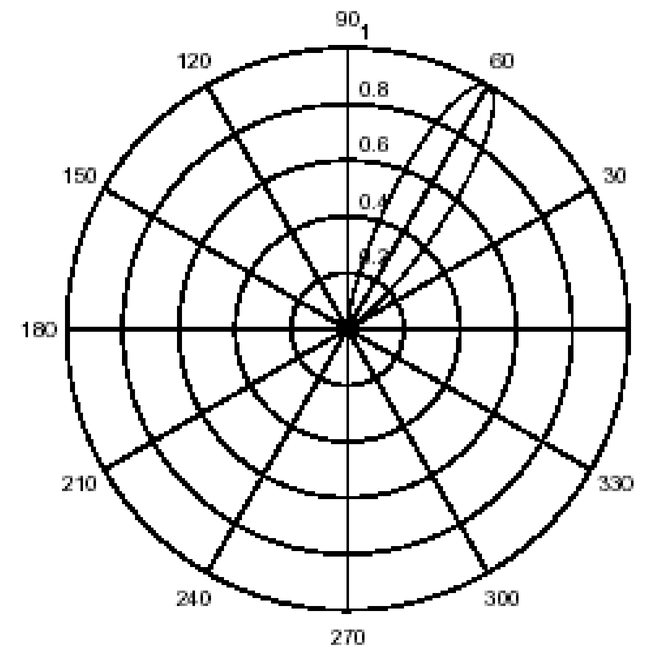



$$I = \begin{cases} I_l & \text{if } \mathbf{V} = \mathbf{R} \\ 0 & \text{otherwise} \end{cases}$$

- For a near-perfect reflector, you might expect the highlight to fall off quickly with increasing angle $\varpi$.
- Also known as:
  - **"rough specular" reflection**
  - **"directional diffuse" reflection**
  - **"glossy" reflection**

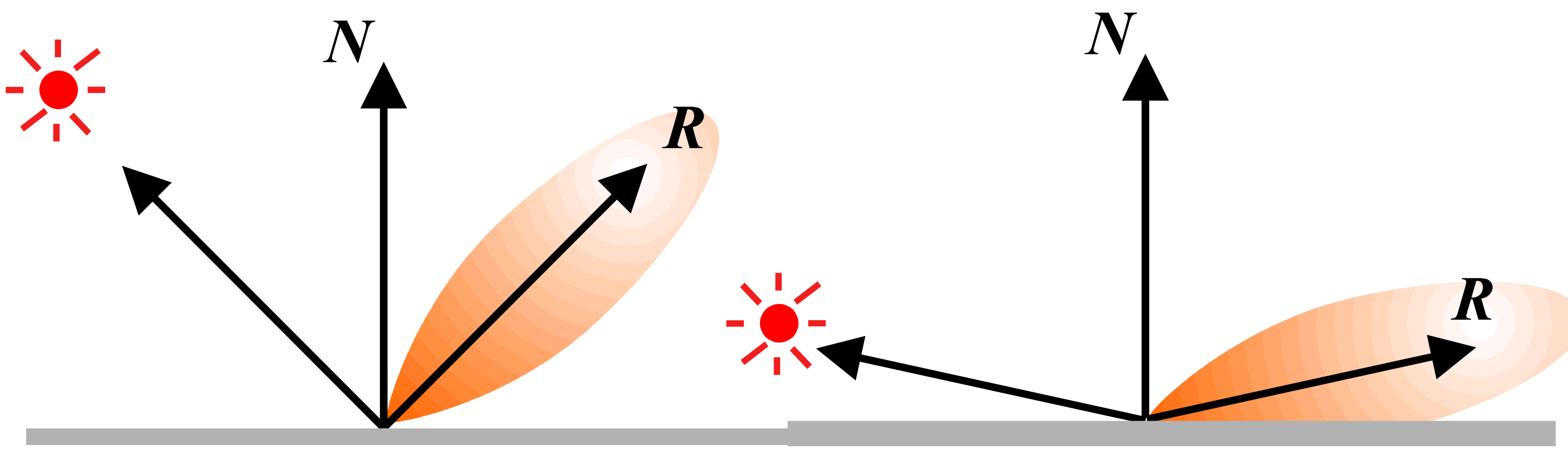
# Specular Reflection Derivation




- One way to get this effect is to take ($\boldsymbol{R} \cdot \boldsymbol{V}$), raised to a power $n_s$.
- As $n_s$ gets larger,
  - the dropoff becomes {more,less} gradual
  - gives a {larger,smaller} highlight
  - simulates a {more,less} shiny surface

# Specular Reflection



# Iteration Three

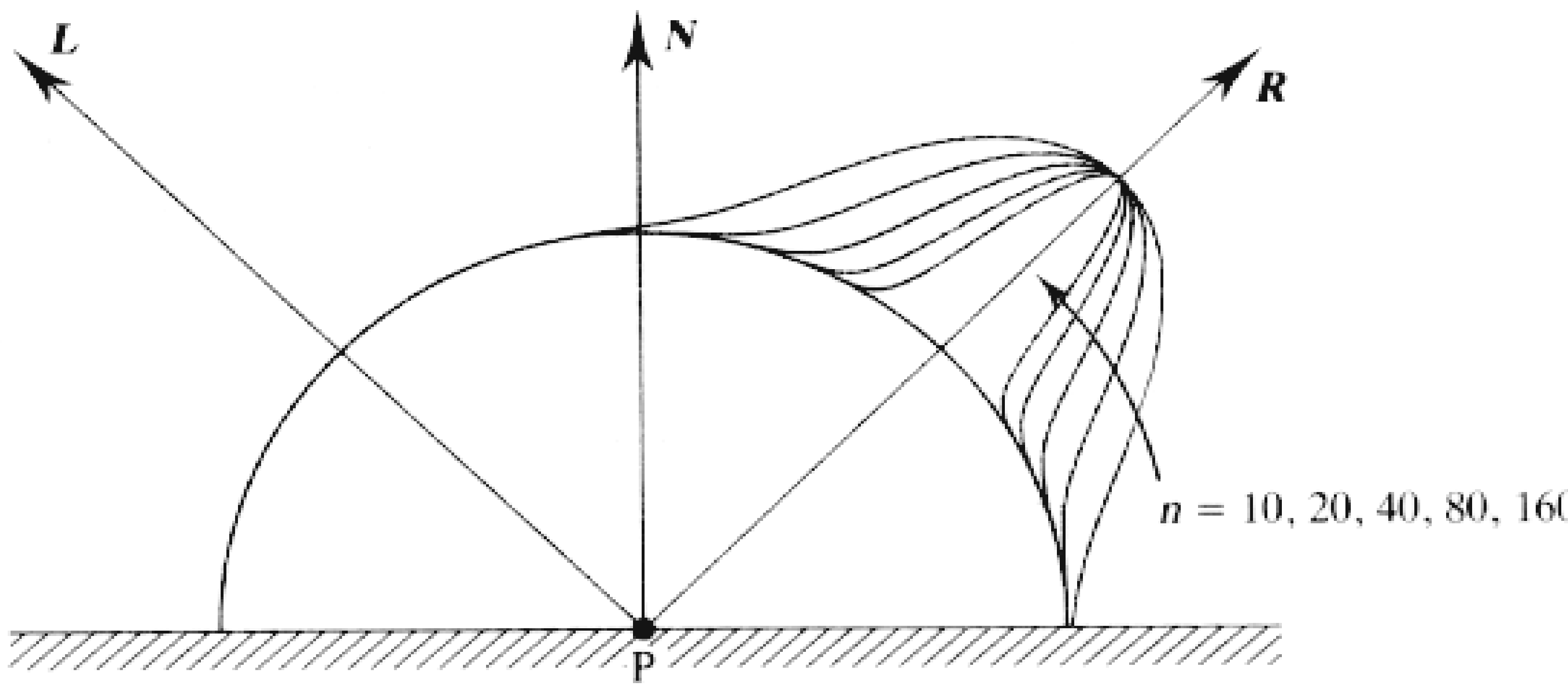
- Since light is additive, we can handle multiple lights by taking the sum over every light.
- Our equation is now:

$$I = k_e + k_a I_a + \sum_i f(d_i) I_{li} \left[ k_d (\mathbf{N} \cdot \mathbf{L}_i)_+ + k_s (\mathbf{V} \cdot \mathbf{R})_+^{n_s} \right]$$

- This is the **Phong illumination model**.
- Which quantities are spatial vectors?
- Which are RGB triples?
- Which are scalars?

# Diffuse + Specular Reflection

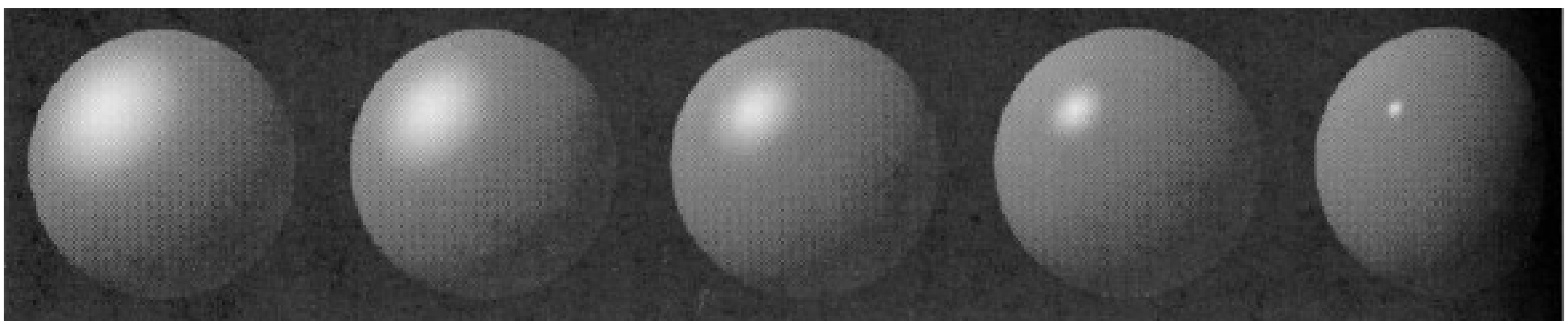
- **<u>Diffuse + Specular</u>**




- You can control the ratio of diffuse to specular by adjusting $\boldsymbol{k_s}$ and $\boldsymbol{k_d}$
- Contribution due to multiple light sources are simply added together

# Specular Examples

- Effect on varying $n_s$.



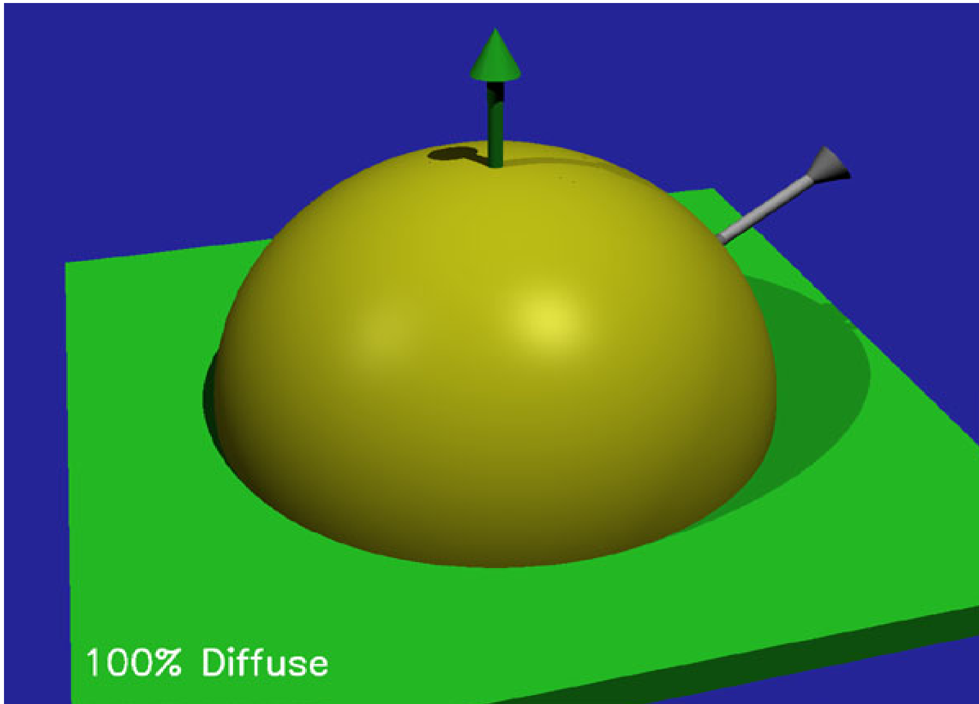
# Choosing the Parameters?

- How would I model…
  - polished copper?
  - blue plastic?
  - lunar dust?

# Choosing the Parameters

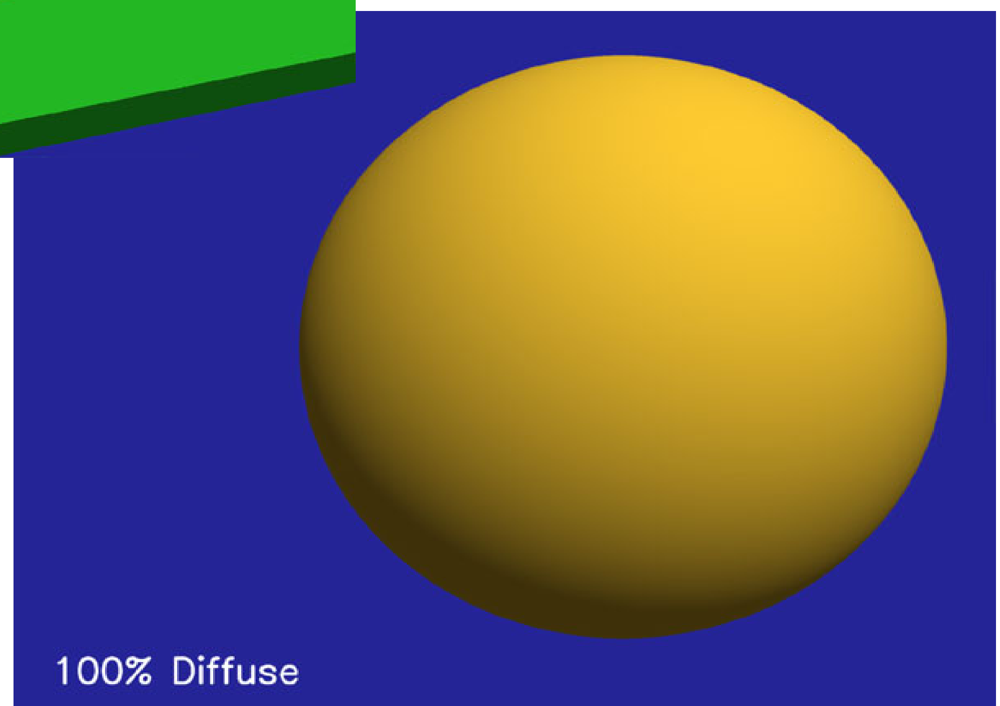
- Ns in the range [0,100]
- Try Ka +Kd + Ks <= 1
- Use a small Ka (~0.1)

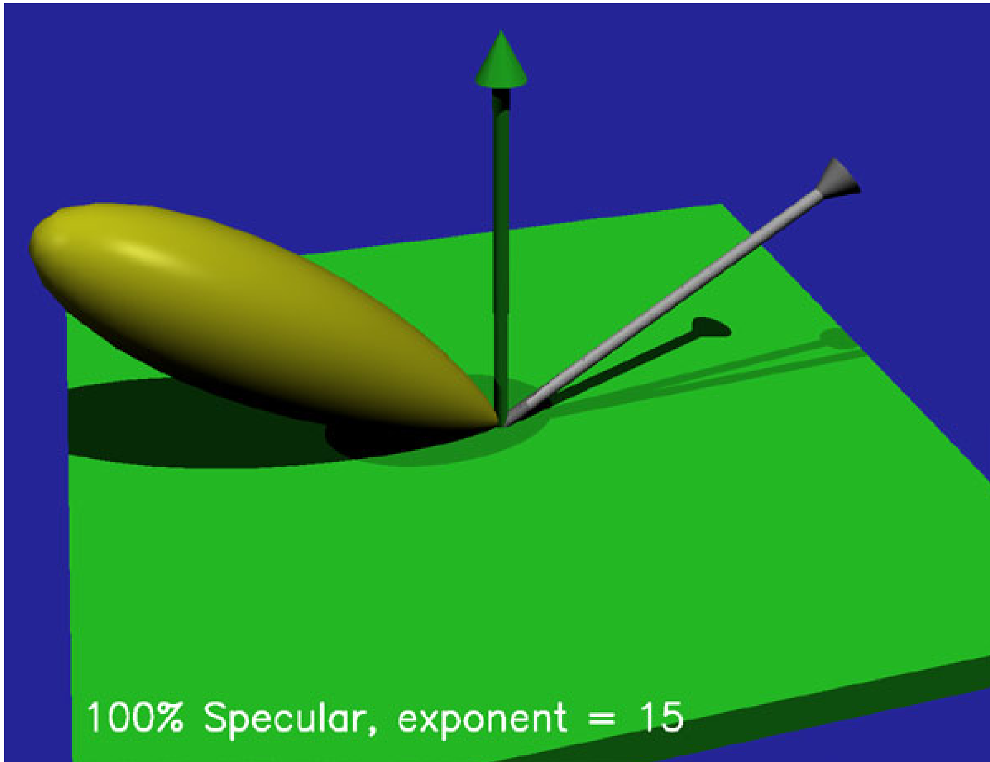
| | Ns | Kd | Ks |
|---|---|---|---|
| Metal | Large | Small,<br>color of metal | Large,<br>color of metal |
| Plastic | Medium | Medium,<br>color of plastic | Medium,<br>white |
| Planet | 0 | Varying | 0 |




3D plot of reflected intensity

Appearance of a
diffuse (dull) sphere

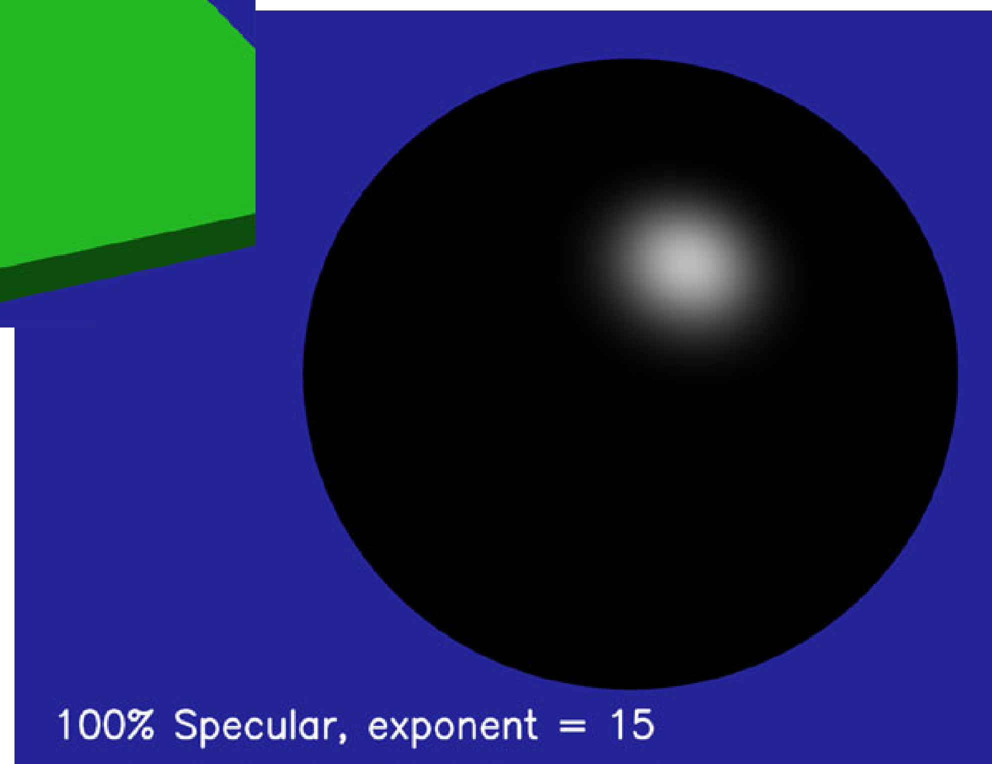


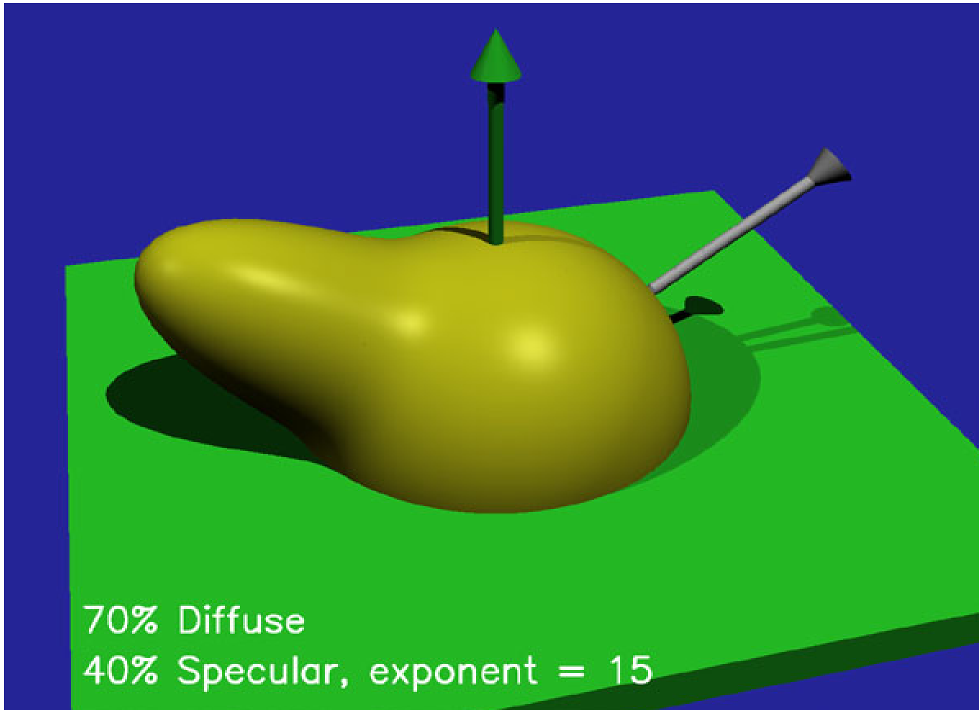


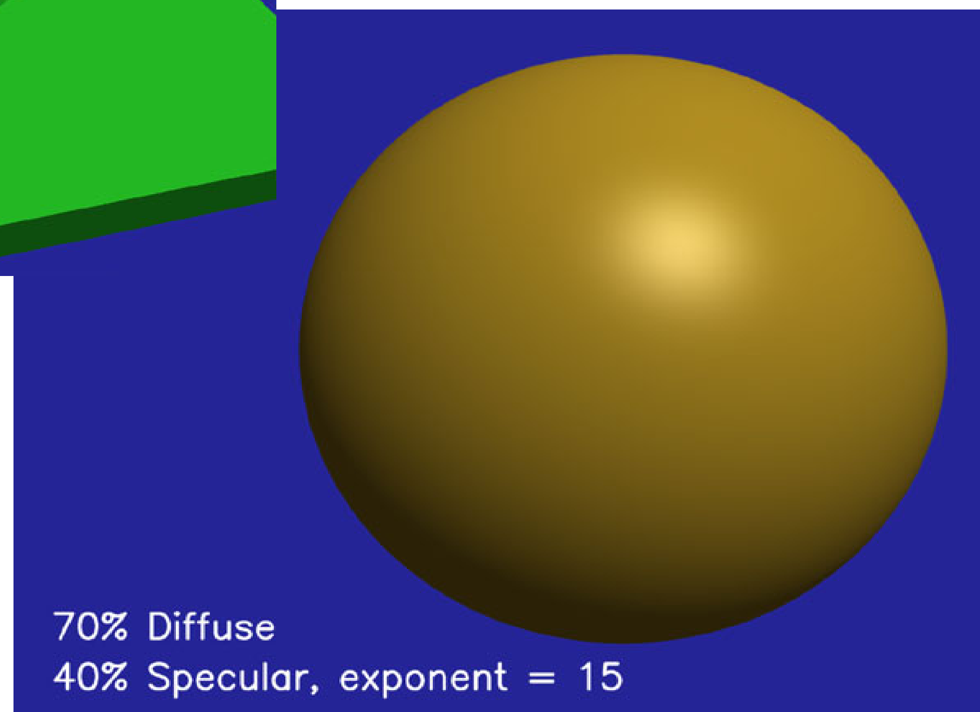

**3D plot of reflected intensity**

***n* = *15***

**Appearance of a specular sphere**



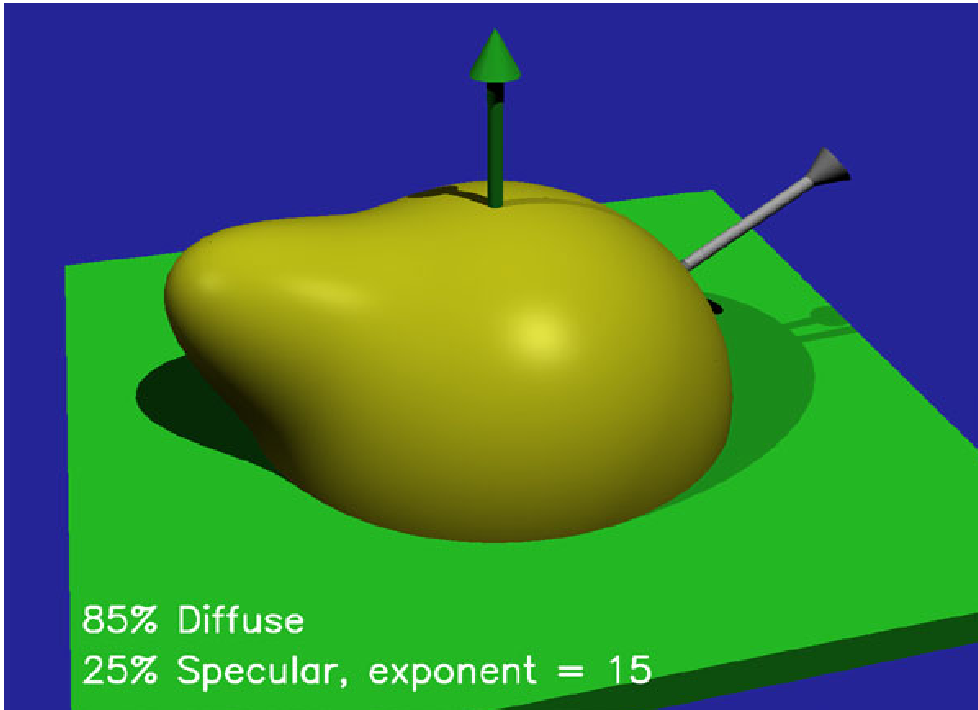


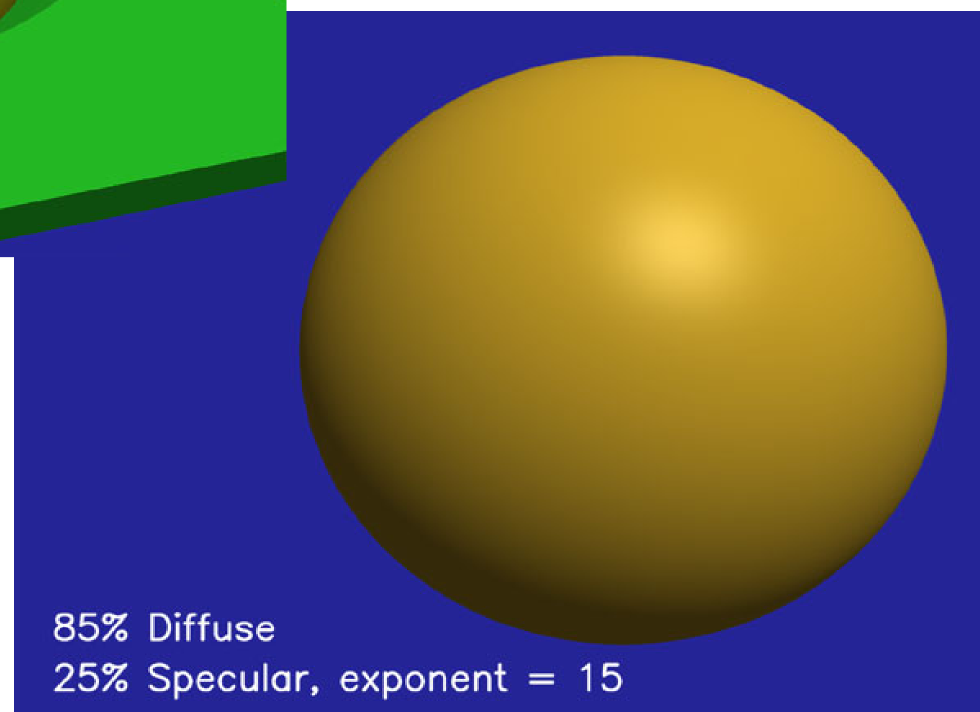

$k_d = 0.7$

$k_s = 0.4$

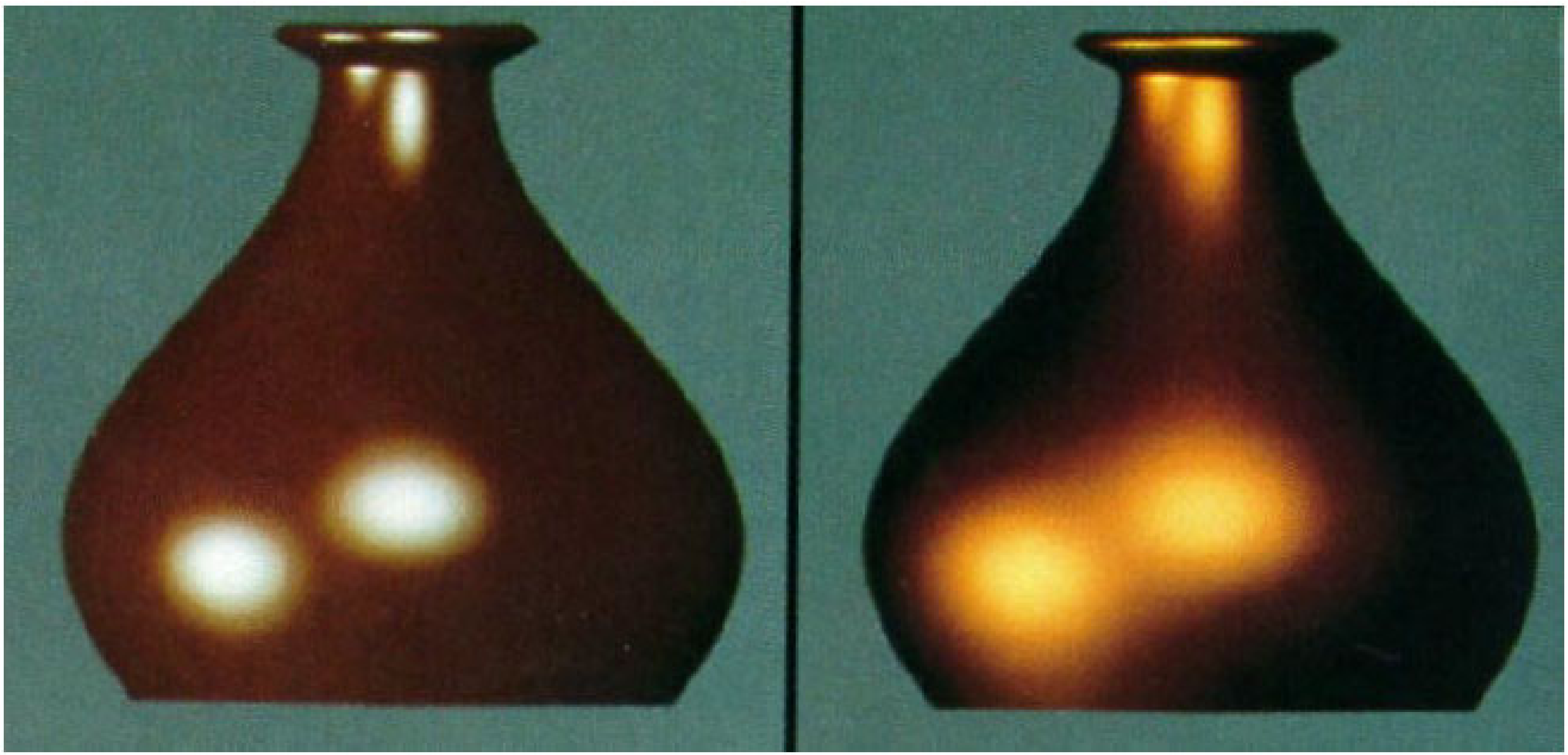
$n = 15$




$k_d = 0.85$
$k_s = 0.25$
$n = 15$



# Choosing the Parameters



# Summary

- The most important thing to take away from this lecture is the final equation for the Phong model.
  - What is the physical meaning of each variable?
  - How are the terms computed?
  - What effect does each term contribute to the image?
  - What does varying the parameters do?