Modeling Light

- Based on, but not the same as, real world lighting
  - Food for thought: why not?
- Thus, whatever you do with light in computer graphics, especially in real time, is ultimately an approximation
- Beyond some core principles (see below), modeling light is a pretty wide-open game, grounded in physics but in some ways ultimately modulated by aesthetics and practicality

Core Calculations

- A lighting model computes the light that hits a polygon, then computes how that polygon reflects this light
- Multiple light sources on a polygon are cumulative: i.e., their RGB values add up, clamped at 1.0
- A material absorbs or reflects light, based on its own color(s) and other properties: this reflection is equivalent to multiplying the light's and material's RGB values (note how having a 0.0–1.0 range is particularly helpful here)

From the Fixed Function Model

The former fixed function OpenGL light model modeled light as three main components:

- Ambient—Light that is so scattered as to appear to be coming from all directions and going in all directions
- Diffuse—Light coming from either a specific direction or a point in space
- Specular—Light that is reflected back in a focused direction; affects the perception of "shininess"

Setting Up a Lit Scene

- Define your objects so that they capture the data that affect how things get lit
  - Light sources: Colors, positions, directions
  - Material settings: Colors, other properties
- Use these settings to perform color calculations within your shaders
- An initial approach would be to assign lit color per vertex, then having OpenGL interpolate the rest

The New "Normal"

- One of the most important geometric aspects of many lighting approaches is the normal vector: that is, the "direction" that a polygon is facing, expressed as (duh) a vector
- The importance of this value makes intuitive sense—see sunrise, noon, and sunset
- Because we care mainly about direction when dealing with normal, they are generally worked with as unit vectors (i.e., lengths equal to 1)

Other Key Issues

Many other issues can get involved, all of which translate into additional shader variables, logic, and computation:

- Attenuation—Does brightness decrease as a function of distance, and if so, by how much?
- Physical properties—"Shininess" for specular reflection is one thing, but there can be many more
- Local vs. global lighting—For simplicity, we calculate lighting per object or vertex; in reality, lit objects also affect each other