




## Lighting

Dave Shreiner



## Lighting Principles

- Lighting simulates how objects reflect light
  - material composition of object
  - light's color and position
  - global lighting parameters
    - ambient light
    - two sided lighting
  - available in both color index and RGBA mode

## How OpenGL Simulates Lights


- Phong lighting model
  - Computed at vertices
- Lighting contributors
  - Surface material properties
  - Light properties
  - Lighting model properties



## Differences between OpenGL and Phong Lighting



## Surface Normals



- Normals define how a surface reflects light
 

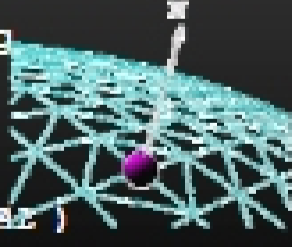

```
glNormal3f( x, y, z )
```

  - Current normal is used to compute vertex's color
  - Use *unit* normals for proper lighting
    - scaling affects a normal's length

```
glEnable( GL_NORMALIZE )
```

or

```
glEnable( GL_RESCALE_NORMAL )
```

## Material Properties

- Define the surface properties of a primitive
 

```
glMaterialfv( face, property, value );
```

GL_DIFFUSE	Base color
GL_SPECULAR	Highlight Color
GL_AMBIENT	Low-light Color
GL_EMISSION	Glow Color
GL_SHININESS	Surface Smoothness

  - separate materials for front and back



### Light Properties

```
glLightfv( light, property, value );
```

- *light* specifies which light
  - multiple lights, starting with `GL_LIGHT0`  
`glGetIntegerv( GL_MAX_LIGHTS, &n );`
- *properties*
  - colors
  - position and type
  - attenuation

### Light Sources (cont.)

- Light color properties
  - `GL_AMBIENT`
  - `GL_DIFFUSE`
  - `GL_SPECULAR`

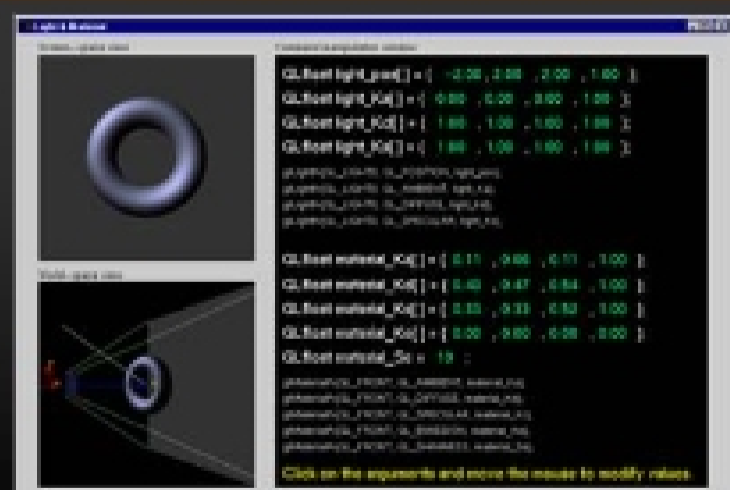
### Types of Lights

- OpenGL supports two types of Lights
  - Local (Point) light sources
  - Infinite (Directional) light sources
- Type of light controlled by *w* coordinate
  - $w = 0$  Infinite Light directed along  $(x \ y \ z)$
  - $w \neq 0$  Local Light positioned at  $(\frac{x}{w} \ \frac{y}{w} \ \frac{z}{w})$

### Turning on the Lights

- Flip each light's switch  
`glEnable( GL_LIGHTn );`
- Turn on the power  
`glEnable( GL_LIGHTING );`

### Light Material Tutorial



```
GLfloat light_pos[] = { -2.0, 2.0, 2.0, 1.0 };
GLfloat light_ka[] = { 0.0, 0.0, 0.0, 1.0 };
GLfloat light_kd[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat light_ks[] = { 1.0, 1.0, 1.0, 1.0 };

glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glEnable(GL_LIGHT1);
glEnable(GL_LIGHT2);
glEnable(GL_LIGHT3);

GLfloat material_ka[] = { 0.1, 0.0, 0.1, 1.0 };
GLfloat material_kd[] = { 0.4, 0.4, 0.4, 1.0 };
GLfloat material_ks[] = { 0.0, 0.0, 0.0, 1.0 };
GLfloat material_ks2[] = { 0.0, 0.0, 0.0, 0.0 };
GLfloat material_sh = 15 ;

glMaterialfv(GL_FRONT, GL_AMBIENT, material_ka);
glMaterialfv(GL_FRONT, GL_DIFFUSE, material_kd);
glMaterialfv(GL_FRONT, GL_SPECULAR, material_ks);
glMaterialfv(GL_BACK, GL_SPECULAR, material_ks2);
glMaterialfv(GL_FRONT, GL_SHININESS, material_sh);

Click on the equations and move the mouse to modify values.
```

### Controlling a Light's Position

- Modelview matrix affects a light's position
  - Different effects based on when position is specified
    - eye coordinates
    - world coordinates
    - model coordinates
  - Push and pop matrices to uniquely control a light's position

### Light Position Tutorial

```

GLfloat pos[] = { 1.0, 1.0, 1.0, 0.0 };
glLightfv( GL_LIGHT0, GL_POSITION, pos );
    
```

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

### Advanced Lighting Features

- **Spotlights**
  - localize lighting affects
    - `GL_SPOT_DIRECTION`
    - `GL_SPOT_CUTOFF`
    - `GL_SPOT_EXPONENT`

### Light Material Tutorial

```

GLfloat mat[] = { 0.5, 0.5, 0.5, 1.0 };
glMaterialfv( GL_FRONT, GL_SPECULAR, mat );
    
```

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

### Advanced Lighting Features

- **Light attenuation**
  - decrease light intensity with distance
    - `GL_CONSTANT_ATTENUATION`
    - `GL_LINEAR_ATTENUATION`
    - `GL_QUADRATIC_ATTENUATION`

$$f_i = \frac{1}{k_c + k_l d + k_q d^2}$$

### Light Model Properties

```

glLightModelfv( property, value );
    
```

- **Enabling two sided lighting**  
`GL_LIGHT_MODEL_TWO_SIDE`
- **Global ambient color**  
`GL_LIGHT_MODEL_AMBIENT`
- **Local viewer mode**  
`GL_LIGHT_MODEL_LOCAL_VIEWER`
- **Separate specular color**  
`GL_LIGHT_MODEL_COLOR_CONTROL`

### Ambient Light Sources

A scene lit only with an ambient light source:

- Light Position  
Not Important
- Viewer Position  
Not Important
- Surface Angle  
Not Important