

# Lighting in OpenGL

Excerpted from  
An Interactive Introduction to  
OpenGL Programming

Dave Shreiner

Ed Angel

Vicki Shreiner





# Lighting

## Vicki Shreiner

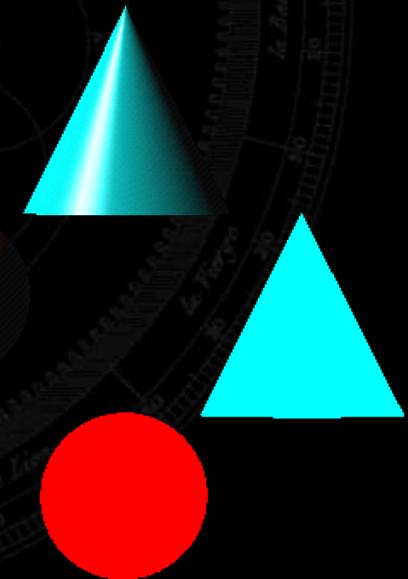


EXPLORE INTERACTION  
AND DIGITAL IMAGES

# 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 *natriail ជីវិតនាយកសារ  
ពេទ័រក្រុង*
- available in both color index and RGBA mode



# How OpenGL Simulates Lights

## Phong lighting model

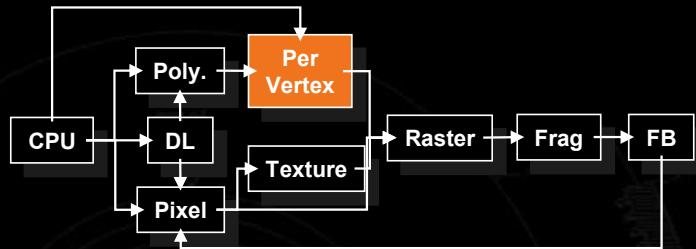
- Computed at vertices

## Lighting contributors

- Surface material properties ผิวฝาผนัง
- Light properties
- Lighting model properties



# Surface Normals



Normals define how a surface reflects light

`glNormal3f( x, y, z )`

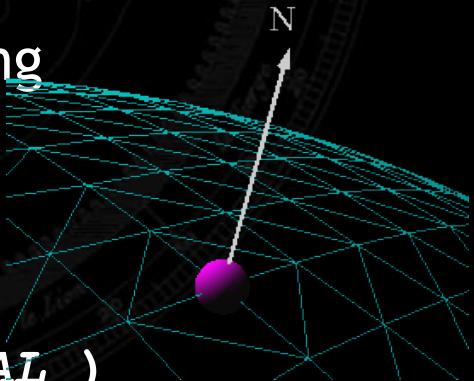
- Current normal is used to compute vertex's color
- Use *unit* normals for proper lighting
  - scaling affects a normal's length

`glEnable( GL_NORMALIZE )`

or

`glEnable( GL_RESCALE_NORMAL )`

↳ ສະແດງ unit normalize ໂດຍ ມີຄວາມທິດທະນາ  
ຫຼັກ



# Material Properties

ດែលវាមិនជូនទេ

GL\_FRONT

GL\_BACK

GL\_FRONT\_AND\_BACK

ជាមួយលើ ព័ត៌មាន

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
scalar → GL_SHININESS	Surface Smoothness

scalar

- separate materials for front and back



# Light Properties

គុណវត្ថិលេខា

ពាណិជ្ជកម្មនៃលើកដែលត្រូវបានបង្កើត

`glLightfv( light, property, value )`

- *light* specifies which light

ផ្លូវ `GL_LIGHT`

- multiple lights, starting with `GL_LIGHT0`

`n = glGetIntegerv( GL_MAX_LIGHTS)`

- *properties*

↑  
រាយការណ៍ដែលត្រូវបានបង្កើត

- colors / *ambient, diffuse, specular*

- position and type

- attenuation

←  
ការបង្កើតលើកដែលត្រូវបានបង្កើត



# Light Sources (cont.)

## Light color properties

- **GL\_AMBIENT**
- **GL\_DIFFUSE**
- **GL\_SPECULAR**



# Types of Lights

Light properties  
↳ GL\_POSITION

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 ເປີ ຈຳ ໄນ

Flip each light's switch

`glEnable( GL_LIGHTn )` ລັບລົງຈວາວ

Turn on the power

`glEnable( GL_LIGHTING )` ລັບກຸ່ມາ

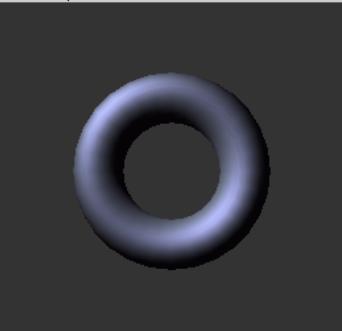
ິຈາ ສົກສົງ ມີ ອົບ ດີເລີດ ໄນຈວາວ



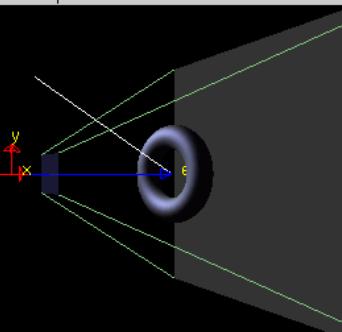
# Light Material Tutorial

**Light & Material**

Screen-space view



World-space view



Command manipulation window

```
GLfloat light_pos[] = { -2.00 , 2.00 , 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[] = { 0.43 , 0.47 , 0.54 , 1.00 };
GLfloat material_Ks[] = { 0.33 , 0.33 , 0.52 , 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.



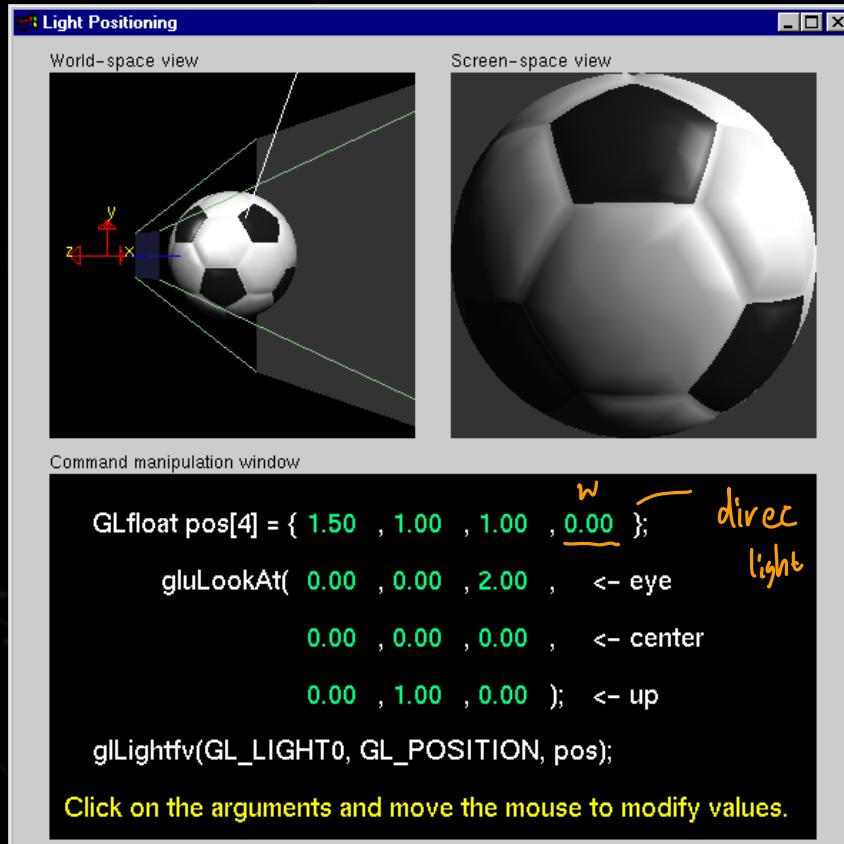
# Controlling a Light's Position

## Modelview matrix affects a light's position

- Different effects based on when position is specified
  - eye coordinates ក្នុងវិនិច្ឆ័យ តាមពេលណា  
សម្រាប់ការគិតថយក (ឱ្យអាចគិតឡើង)
  - world coordinates ឱ្យក្នុងរៀបចំ  
សម្រាប់ការគិតថយក
  - model coordinates ឱ្យក្នុងរៀបចំ
- Push and pop matrices to uniquely control a light's position



# Light Position Tutorial

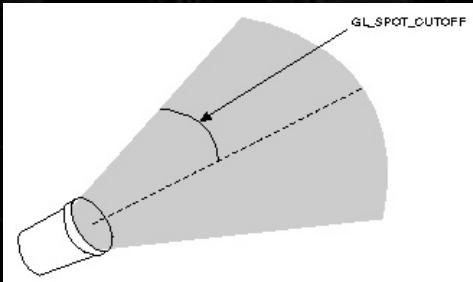


# Advanced Lighting Features

## Spotlights

- localize lighting affects

- *GL\_SPOT\_DIRECTION* ក្រុងលាងនៃអំពី
- *GL\_SPOT\_CUTOFF* ក្រុងលាងនៃចាត់បន្ទាន់
- *GL\_SPOT\_EXPONENT* ចំណាំរាយតាមលក្ខណៈ



# Advanced Lighting Features

## Light attenuation

- decrease light intensity with distance
  - ***GL\_CONSTANT\_ATTENUATION***  $k_c$
  - ***GL\_LINEAR\_ATTENUATION***  $k_l$
  - ***GL\_QUADRATIC\_ATTENUATION***  $k_q$

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

surround  $\Rightarrow 1$       sun & screen  $\Rightarrow 1$



# 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`



# Tips for Better Lighting

Recall lighting computed only at vertices

- model tessellation heavily affects lighting results
  - better results but more geometry to process

Use a single infinite light for fastest lighting

- minimal computation per vertex



# On-Line Resources

- <http://www.opengl.org>
  - start here; up to date specification and lots of sample code
- `news:comp.graphics.api.opengl`
- <http://www.sgi.com/software/opengl>
- <http://www.mesa3d.org/>
  - Brian Paul's Mesa 3D
- <http://www.cs.utah.edu/~narobins/opengl.html>
  - very special thanks to Nate Robins for the OpenGL Tutors
  - source code for tutors available here!



# Books

**OpenGL Programming Guide, 3<sup>rd</sup> Edition**

**OpenGL Reference Manual, 3<sup>rd</sup> Edition**

**OpenGL Programming for the X Window System**

- includes many GLUT examples

**Interactive Computer Graphics: A top-down approach with OpenGL, 2<sup>nd</sup> Edition**



# Authors

Dave Shreiner

[shreiner@sgi.com](mailto:shreiner@sgi.com)

Ed Angel

[angel@cs.unm.edu](mailto:angel@cs.unm.edu)

Vicki Shreiner

[vshreiner@sgi.com](mailto:vshreiner@sgi.com)

