
CS770/870 Fall 2008

OpenGL Lighting

Related material in Hill and Kelley: Sections 8.2-8.4

00:19

CS770/870 Fall 2008 Bergeron/Plumlee

1

Preview

- How to add lighting to an OpenGL program
- How to enable the depth buffer
- Putting it into a Scene Graph

00:19

CS770/870 Fall 2008 Bergeron/Plumlee

2

How to Add OpenGL Lighting

- Step 1: Tell GL you want to use lighting
`glEnable(GL_LIGHTING);`
- Step 2: Turn on a light!
`glEnable(GL_LIGHT0);`
- Step 3: Specify light parameters
`glLightfv(GL_LIGHT0, GL_..., ...);`
- Step 4: Specify mat. properties for vertices
`glMaterialfv(GL_FRONT, GL_..., ...);`
- Step 5: Specify normals for vertices
`glNormal3f(x, y, z);`

00:19

CS770/870 Fall 2008 Bergeron/Plumlee

3

Key glLightfv() Parameters

Parameter Name	Default Value	Meaning
GL_AMBIENT	(0.0, 0.0, 0.0, 1.0)	ambient RGBA intensity of light
GL_DIFFUSE	(1.0, 1.0, 1.0, 1.0)	diffuse RGBA intensity of light
GL_SPECULAR	(1.0, 1.0, 1.0, 1.0)	specular RGBA intensity of light
GL_POSITION	(0.0, 0.0, 1.0, 0.0)	(x, y, z, w) position of light

Table 5-1 : Default Values for pname Parameter of glLight(), from <http://www.glprogramming.com/red/chapter05.html>*

00:19

CS770/870 Fall 2008 Bergeron/Plumlee

4

Key glMaterialfv() Parameters

Parameter Name	Default Value	Meaning
GL_AMBIENT	(0.2, 0.2, 0.2, 1.0)	ambient color of material
GL_DIFFUSE	(0.8, 0.8, 0.8, 1.0)	diffuse color of material
GL_AMBIENT_AND_DIFFUSE		ambient and diffuse color of material
GL_SPECULAR	(0.0, 0.0, 0.0, 1.0)	specular color of material
GL_SHININESS	0.0	specular exponent
GL_EMISSION	(0.0, 0.0, 0.0, 1.0)	emissive color of material

Table 5-3 : Default Values for pname Parameter of glMaterial*(), from <http://www.glprogramming.com/red/chapter05.html>

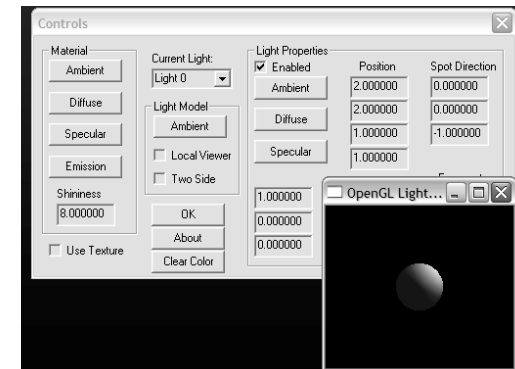
00:19

CS770/870 Fall 2008 Bergeron/Plumlee

5

Lighting: Putting It All Together

- Example at <http://graphicscodingprojects.blogspot.com/2008/10/opengl-lighting-demo.html>
- Demo



00:19

6

Enabling the Depth Buffer

- Why do we need it?
 - Else: apparent depth depends on order drawn
- One-time initializations items


```
glutInitDisplayMode (GLUT_DEPTH | ... );
glEnable (GL_DEPTH_TEST);
```
- Done once per frame


```
glClear( GL_COLOR_BUFFER_BIT |
          GL_DEPTH_BUFFER_BIT )
```
- Other functions

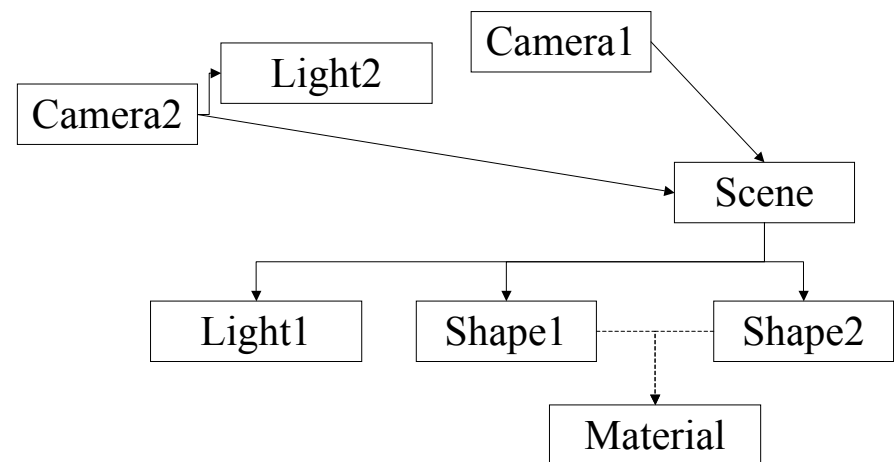

```
glDepthFunc(), glDepthMask(), glDepthRange()...
```

00:19

CS770/870 Fall 2008 Bergeron/Plumlee

7

A Sample Program Instance



00:19

CS770/870 Fall 2008 Bergeron/Plumlee

8

Camera Items

- Bookkeeping (might be in separate Window class, with Camera as a component)
 - `glutInitDisplayMode(...)`
 - `glEnable(GL_DEPTH_TEST)`
 - `glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)`
- Setting up the camera
 - Orthographic or perspective projection?
 - Camera position (`GL_MODELVIEW`)
- Other items: Lights? Spec. Material? Fog?

00:19

CS770/870 Fall 2008 Bergeron/Plumlee

9

Shape Items

- Apply at start of rendering
 - Transform (may be own class)
 - Material properties (may be own class)
- Apply potentially per-vector (`glBegin/End`)
 - `glShadeModel(GL_SMOOTH)` *or*
`glShadeModel(GL_FLAT)`
 - `glNormal3f(x, y, z);`
 - Special material properties

00:19

CS770/870 Fall 2008 Bergeron/Plumlee

10

Light Items

- Lights might “render” similarly to shapes
 - Transform
 - Material-like properties: color, specular, etc.
 - Call to `glLightfv(GL_LIGHT_POSITION ...)`

00:19

CS770/870 Fall 2008 Bergeron/Plumlee

11

Review

- Lighting in an OpenGL program

```
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glLightfv(GL_LIGHT0, GL_..., ...);
glMaterialfv(GL_FRONT, GL_..., ...);
glNormal3f(x, y, z);
glShadeModel(...);
```
- Enabling the depth buffer

```
glutInitDisplayMode( GLUT_DEPTH | ... );
glEnable(GL_DEPTH_TEST);
glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT )
```
- Putting it into a Scene Graph

00:19

CS770/870 Fall 2008 Bergeron/Plumlee

12

Resources

- Some good OpenGL resources

- <http://www.opengl.org/resources/faq/technical/lights.htm>
- <http://www.glprogramming.com/red/chapter05.html>
(Chapter 5 from the Red OpenGL book)

- Source for lighting demo

<http://graphicscodingprojects.blogspot.com/2008/10/opengl-lighting-demo.html>