LIGHTING AND MATERIALS IN OPENGL

HOW OPENGL SIMULATES LIGHTS

- PHONG LIGHTING MODEL
  - COMPUTED AT VERTICES
- LIGHTING CONTRIBUTORS
  - SURFACE MATERIAL PROPERTIES
  - LIGHT PROPERTIES
  - LIGHTING MODEL PROPERTIES

SURFACE NORMALS

- NORMALS DEFINE HOW A SURFACE REFLECTS LIGHT

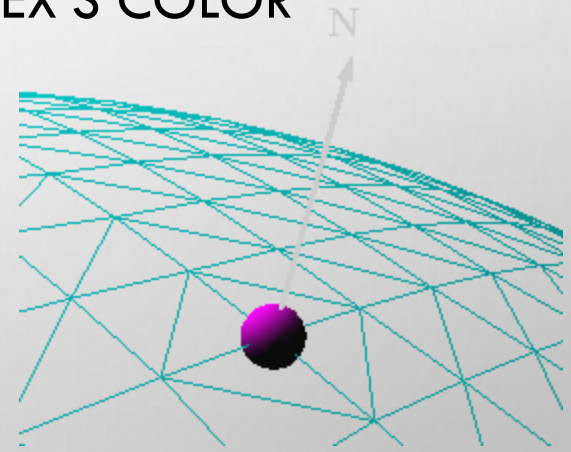
```
GLNORMAL3F( X, Y, Z )
```

  - 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 )
```



MATERIAL PROPERTIES

- DEFINE THE SURFACE PROPERTIES OF A PRIMITIVE

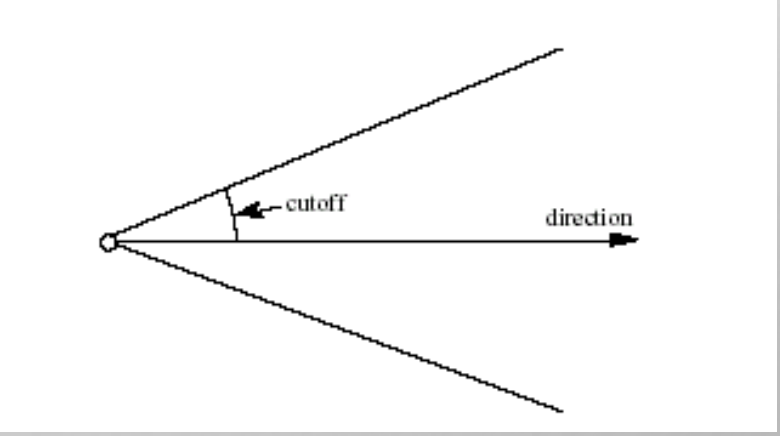
```
GLMATERIALFV( FACE, PROPERTY, VALUE );
```

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

- POSITION OR DIRECTION
- COLOR
- HOW IT IS ATTENUATED (DIMINISHED) OVER DISTANCE
- OMNI-DIRECTIONAL (DEFAULT) OR SPOTLIGHT
  - DIRECTION (3D VECTOR)
  - CUTOFF (0 TO 90)
  - DROPOFF EXPONENT



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*** $\begin{pmatrix} x & y & z \end{pmatrix}$

$w \neq 0$ ***Local Light positioned at*** $\begin{pmatrix} x/w & y/w & z/w \end{pmatrix}$

```
Light_position[]={1,1,1,w}
glLightfv(LIGHT, GL_POSITION, Light_position);
```

TURNING ON THE LIGHTS

- FLIP EACH LIGHT'S SWITCH

```
glEnable( GL_LIGHTN );
```

- TURN ON THE POWER

```
glEnable( GL_LIGHTING );
```

CONTROLLING A LIGHT'S POSITION

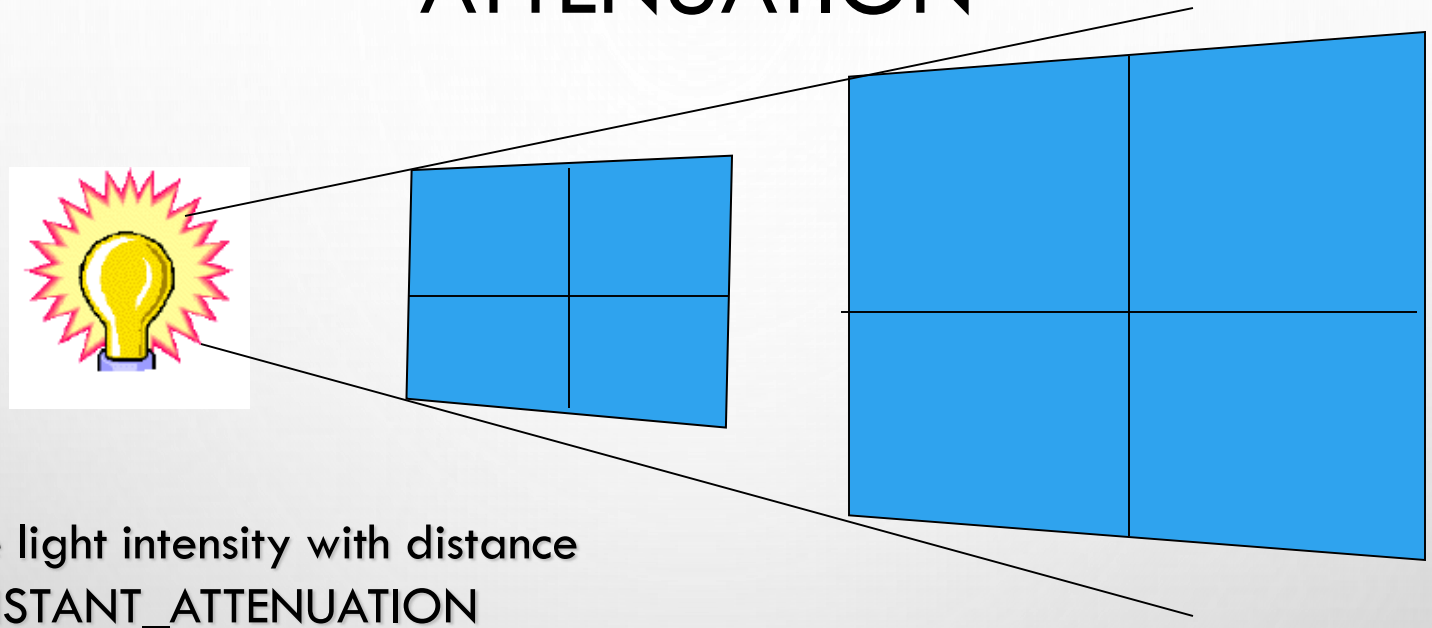
- MODELVIEW MATRIX AFFECTS A LIGHT'S POSITION
  - DIFFERENT EFFECTS BASED ON WHEN POSITION IS SPECIFIED
    - EYE COORDINATES
    - WORLD COORDINATES
    - MODEL COORDINATES
  - PUSH AND POP MATRICES TO UNIQUELY CONTROL A LIGHT'S POSITION

LIGHT MODEL PROPERTIES

- **`GLLIGHTMODELFV( PROPERTY, VALUE );`**
  - PROPERTIES WHICH AREN'T DIRECTLY CONNECTED WITH MATERIALS OR LIGHTS ARE GROUPED INTO *LIGHT MODEL PROPERTIES*. WITH OPENGL 1.2, THERE ARE FOUR PROPERTIES ASSOCIATED WITH THE LIGHTING MODEL:

- ENABLING TWO SIDED LIGHTING
  `GL_LIGHT_MODEL_TWO_SIDE`
- GLOBAL AMBIENT COLOR
  `GL_LIGHT_MODEL_AMBIENT`
- LOCAL VIEWER MODE
  `GL_LIGHT_MODEL_LOCAL_VIEWER`
  LOCAL VIEWER MODE DISABLES AN OPTIMIZATION WHICH PROVIDES FASTER LIGHTING COMPUTATIONS. WITH LOCAL VIEWER MODE ON, YOU GET BETTER LIGHT RESULTS AT A SLIGHT PERFORMANCE PENALTY.
- SEPARATE SPECULAR COLOR
  `GL_LIGHT_MODEL_COLOR_CONTROL`
  SEPARATE SPECULAR COLOR IS A MODE FOR MAINTAINING BETTER SPECULAR HIGHLIGHTS IN CERTAIN TEXTURE MAPPED CONDITIONS. THIS IS A NEW FEATURE FOR OPENGL 1.2.

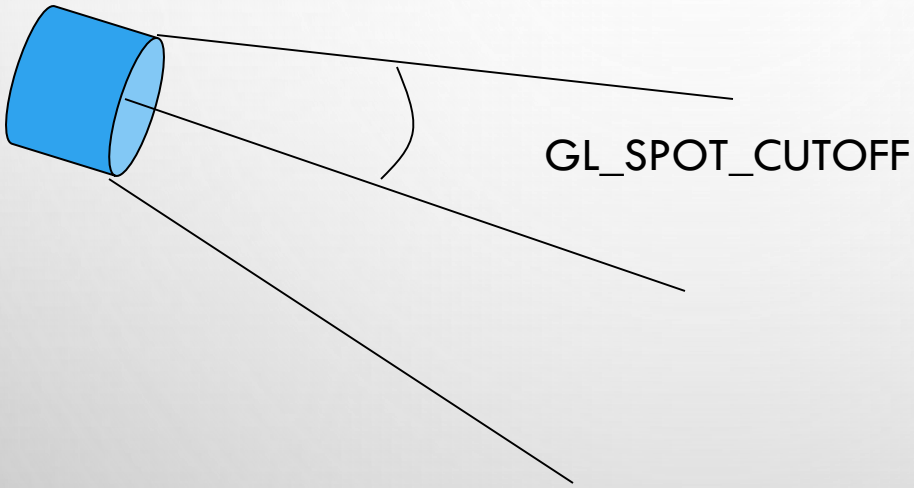
ATTENUATION



Decrease light intensity with distance
GL_CONSTANT_ATTENUATION
GL_LINEAR_ATTENUATION
GL_QUADRATIC_ATTENUATION

$$attenuation\ factor = \frac{1}{k_c + k_j d + k_q d^2}$$

SPOTLIGHT




```
glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 45);
Glfloat spot_direction[]={-1,-1,0}
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, spot_direction);
```

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

LIGHTS IN OPENGL

- **GLENABLE**(GL_LIGHTING)
  - IF ENABLED, USE THE CURRENT LIGHTING PARAMETERS TO COMPUTE THE VERTEX COLOR OR INDEX. OTHERWISE, SIMPLY ASSOCIATE THE CURRENT COLOR OR INDEX WITH EACH VERTEX.
- **GLENABLE**(GL_LIGHT_I)
  - IF ENABLED, INCLUDE LIGHT I IN THE EVALUATION OF THE LIGHTING EQUATION.
- **GLENABLE**(GL_NORMALIZE)
  - IF ENABLED, NORMAL VECTORS SPECIFIED WITH ***GLNORMAL*** ARE SCALED TO UNIT LENGTH AFTER TRANSFORMATION.
  - BEFORE ANY GEOMETRY IS SPECIFIED, WILL AUTOMATICALLY NORMALIZE VECTORS!

GL_LIGHT_MODEL[F,I] (*PNAME*, *PARAM*)

- SET THE LIGHTING MODEL PARAMETERS
- *PNAME*
  - GL_LIGHT_MODEL_AMBIENT
  - GL_LIGHT_MODEL_LOCAL_VIEWER
  - GL_LIGHT_MODEL_TWO_SIDE
- *PARAM* (WITH RESPECT TO *PNAME*)
  - AMBIENT RGBA INTENSITY OF THE ENTIRE SCENE
  - HOW SPECULAR REFLECTION ANGLES ARE COMPUTED. 0 (DEFAULT) VIEW DIRECTION TO BE PARALLEL TO AND IN THE DIRECTION OF THE -Z-AXIS. OTHERWISE, FROM THE ORIGIN OF THE EYE COORDINATE SYSTEM.
  - SPECIFIES WHETHER ONE- (0, FRONT ONLY, DEFAULT) OR TWO-SIDED LIGHTING (NON-ZERO) CALCULATIONS ARE DONE FOR POLYGONS

GL_LIGHT[F,I]**V**(LIGHT, PNAME, *PARAMS)

- *LIGHT* - SPECIFIES A LIGHT: **GL_LIGHT**$_i$
- *PNAME* - SPECIFIES A LIGHT SOURCE PARAMETER FOR *LIGHT*:
  - GL_AMBIENT — AMBIENT INTENSITY, RGBA, DEFAULT (0, 0, 0, 1)
  - GL_DIFFUSE — DIFFUSE INTENSITY, RGBA, DEFAULT (1, 1, 1, 1)
  - GL_SPECULAR — SPECULAR INTENSITY, RGBA, DEFAULT (1, 1, 1, 1)
  - GL_POSITION — LIGHT POSITION, WORLD COORDS, DEFAULT (0, 0, 1, 0, DIRECTIONAL, PARALLEL TO Z-AXIS)
  - GL_SPOT_DIRECTION — EYE COORDS, DEFAULT (0,0,-1)
  - GL_SPOT_EXPONENT — INTENSITY DISTRIBUTION [0, 128], DEFAULT (0)
  - GL_SPOT_CUTOFF — MAXIMUM SPREAD ANGLE [0, 90], DEFAULT 180
  - GL_CONSTANT_ATTENUATION — DEFAULT 1
  - GL_LINEAR_ATTENUATION — DEFAULT 0
  - GL_QUADRATIC_ATTENUATION — DEFAULT 0
- *PARAMS* - SPECIFIES A POINTER TO THE VALUE OR VALUES THAT PARAMETER PNAME OF LIGHT SOURCE LIGHT WILL BE SET TO (SEE COLUMN 2).

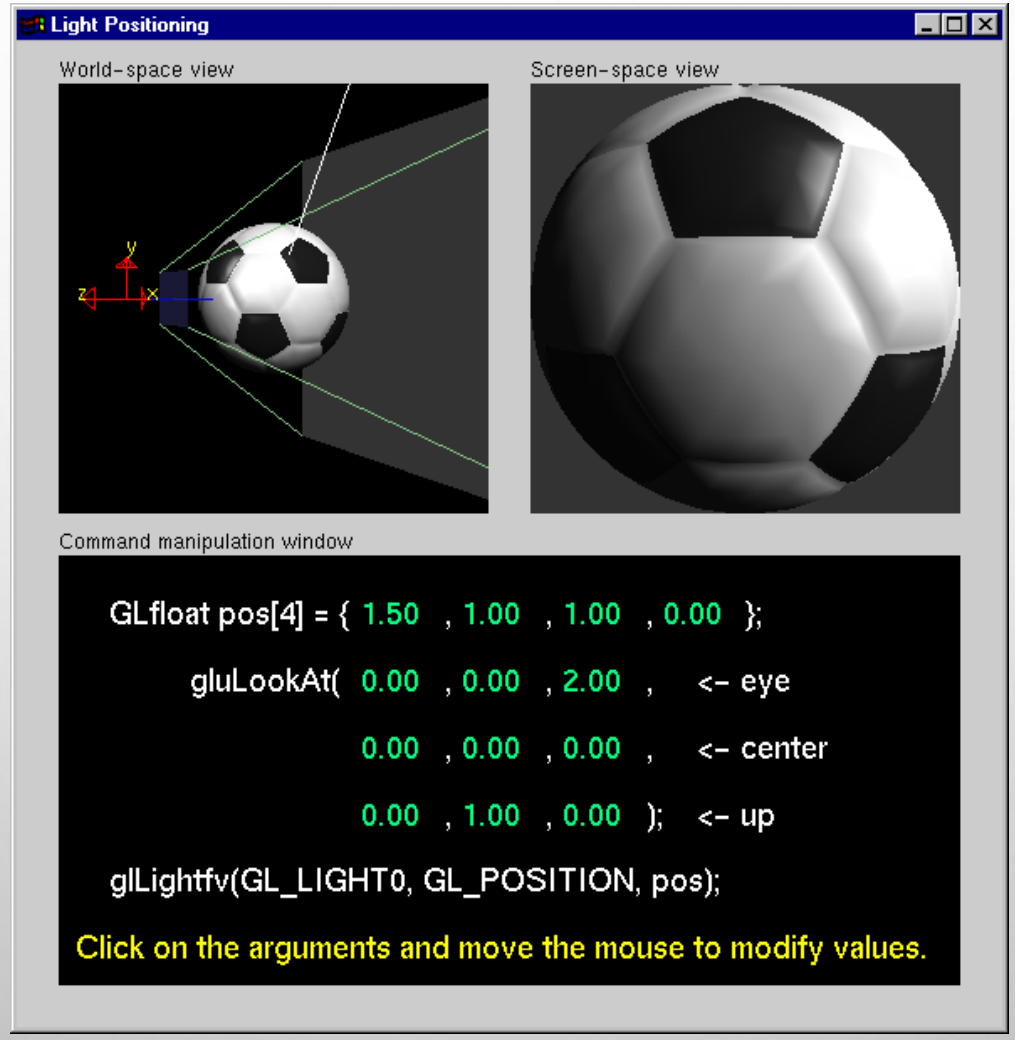
GL_NORMAL3[B,D,F,I,S](NX, NY, NZ)
GL_NORMAL3[B,D,F,I,S]**V**(*V)

- SET THE CURRENT NORMAL VECTOR (FOR A VERTEX)
- SPECIFY THE X, Y, AND Z COORDINATES OF THE NEW CURRENT NORMAL. THE INITIAL VALUE OF THE CURRENT NORMAL IS (0,0,1)

  OR

- SPECIFIES A POINTER TO AN ARRAY OF THREE ELEMENTS: THE X, Y, AND Z COORDINATES OF THE NEW CURRENT NORMAL.

LIGHT POSITION TUTORIAL



MATERIAL COLORS

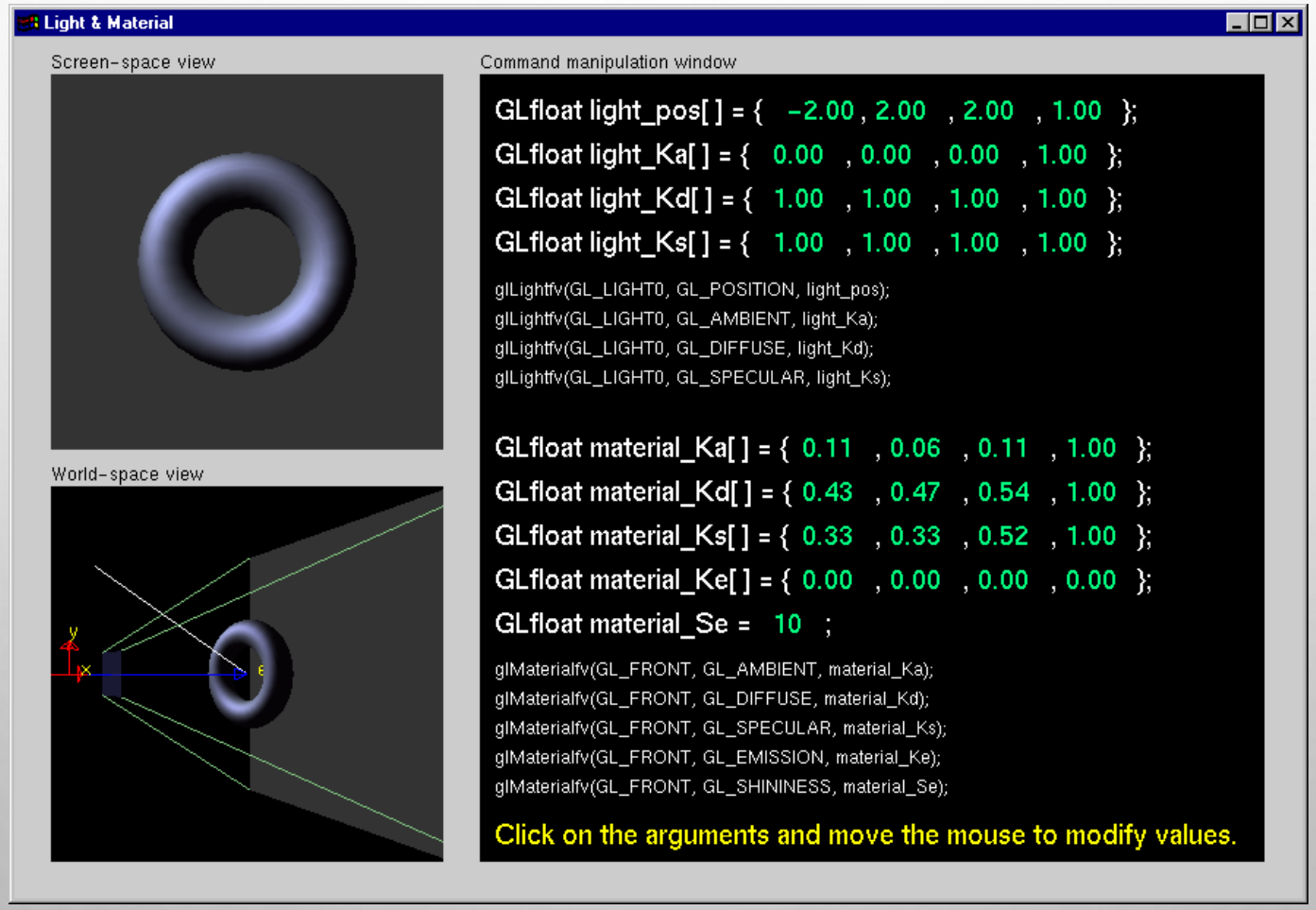
- CHARACTERISTICS OF SURFACES
  - AMBIENT
  - DIFFUSE
  - SPECULAR

- WHAT HAPPEN IF THERE IS A WHITE LIGHT SHINING ON A SHINY RED BALL?

RBG VALUES FOR LIGHT AND MATERIALS

- R=1, G=0.5, B=0
  - REFLECTS ALL INCOMING RED
  - REFLECTS HALF OF INCOMING GREEN
  - REFLECTS NO INCOMING BLUE
- LIGHT(LR, LG, LB),MATERIAL(MR, MG, MB)
  - (LRMR , LGMG, LBMB)
- LIGHT(R1, G1, B1), LIGHT2(R2, G2, B2)
  - (R1+R1, G1+G2, B1+B2)

LIGHT MATERIAL TUTORIAL



MATERIALS IN OPENGL

- **GLMATERIAL**[F,I]**V**(FACE, PNAME, *PARAMS)

  OR

- **GLCOLORMATERIAL**(FACE, MODE) (PREFERRED)
  - ***GLCOLORMATERIAL*** SPECIFIES WHICH MATERIAL PARAMETERS TRACK THE CURRENT COLOR.
  - ***GLENABLE (GL_COLOR_MATERIAL)*** NEEDED
  - ***GLCOLORMATERIAL*** ALLOWS A SUBSET OF MATERIAL PARAMETERS TO BE CHANGED FOR EACH VERTEX USING ONLY THE ***GLCOLOR*** COMMAND, WITHOUT CALLING ***GLMATERIAL***. IF ONLY SUCH A SUBSET OF PARAMETERS IS TO BE SPECIFIED FOR EACH VERTEX, ***GLCOLORMATERIAL*** IS PREFERRED OVER CALLING ***GLMATERIAL***.

GLMATERIAL[F,I]**V**(FACE, PNAME, *PARAMS)

- SPECIFY MATERIAL PARAMETERS FOR THE LIGHTING MODEL
- *FACE*
  - GL_FRONT
  - GL_BACK
  - GL_FRONT_AND_BACK
- *PNAME*

	PARAMS (DEFAULT)
GL_AMBIENT	AMBIENT RGBA REFLECTANCE (0.2, 0.2, 0.2, 1.0)
GL_DIFFUSE	DIFFUSE RGBA REFLECTANCE (0.8, 0.8, 0.8, 1.0)
GL_SPECULAR	SPECULAR RGBA REFLECTANCE (0.0,0.0, 0.0, 1.0)
GL_EMISSION	RGB EMITTED LIGHT INTENSITY (0.0,0.0, 0.0, 1.0)
GL_SHININESS	SPECULAR EXPONENT, RANGE [0,128] (0)
GL_AMBIENT_AND_DIFFUSE	EQUIVALENT TO CALLING GLMATERIAL TWICE
GL_COLOR_INDEXES	COLOR INDICES FOR AMBIENT, DIFFUSE, AND SPECULAR LIGHTING - RGB

GLCOLORMATERIAL(FACE, MODE)

- *FACE*
  - GL_FRONT
  - GL_BACK
  - GL_FRONT_AND_BACK
- *MODE* - SPECIFIES WHICH OF SEVERAL MATERIAL PARAMETERS TRACK THE CURRENT COLOR.
  - GL_EMISSION
  - GL_AMBIENT
  - GL_DIFFUSE
  - GL_SPECULAR
  - GL_AMBIENT_AND_DIFFUSE (DEFAULT)