

Iluminación y sombreado en OpenGL

Antes de empezar ...

- Para habilitar un estado se utiliza la función:
 - `glEnable(GLenum)`
- Para deshabilitar un estado se utiliza la función:
 - `glDisable(GLenum)`
- Para verificar un estado se utiliza la función:
 - `glIsEnabled(GLenum)`

Antes de empezar ...

- Vamos a definir la ***reflectividad*** de los materiales como el porcentaje de luz que pueden reflejar.
- Como vamos a trabajar en un modelo RGB, este porcentaje es independiente para componente (rojo, verde, y azul)
 - Va a ser un valor entre 0.f y 1.f
 - No confundir con un color!!

Modelo de luz en OpenGL

- En el modo RGBA, el color de un vértice es la suma de los siguientes términos:
 - La intensidad de emisión del material
 - La intensidad ambiental global por la reflectividad ambiente del material
 - La contribución de cada luz.

Modelo de luz en OpenGL

- La contribución de la luz es la suma de tres términos:
 - Componente ambiente
 - Componente difuso
 - Componente especular

Modelo de luz en OpenGL

- La contribución del color ambiente se calcula como la multiplicación entre el color ambiente de la luz y la ***reflectividad*** ambiente del material
- La contribución del color difuso se calcula como la multiplicación entre el color difuso de la luz, la ***reflectividad*** difusa del material, y el producto interno entre la normal en el vértice y el vector **NORMALIZADO** que va desde el vértice hacia la luz.

Modelo de luz en OpenGL

- La contribución del color especular se calcula como la multiplicación entre el color especular de la luz, la ***reflectividad*** especular del material, el producto interno entre los vectores **NORMALIZADOS** "vértice hacia la luz" y "vértice hacia el ojo", y todo esto elevado a la potencia de brillo especular del material.

Modelo de luz en OpenGL

- Los tres componentes de color de luz, son atenuados por igual según la distancia del vértice a la luz, la dirección de la luz, exponente de brillo especular, y ángulo de corte de la luz (aletas y conos).
- Cuando un producto interno tiene un resultado negativo, se cambia ese valor por cero.

Modelo de luz en OpenGL

- El componente *alpha* del color resultante, toma el valor del canal alpha del componente difuso en el material.
- **Nota:** existe otro tipo de iluminación (indexación de colores), pero no lo vamos a ver en este curso.

Funciones a utilizar

- `glNormal3[bdfis](x, y, z)`
- `glShadeModel(GLenum)`
- `glLight[if]{v}(light, pname, param)`
- `glMaterial[if]{v}(face, pname, param)`

glNormal

- Especifica la normal en el punto que estamos definiendo (vector no necesariamente normalizado).
 - El valor inicial es el vector unidad $(0, 0, 1)$.
- Para habilitar/deshabilitar la normalización de los vectores, se utiliza el valor **GL_NORMALIZE**.
 - La normalización esta deshabilitada por defecto.

glShadeModel

- La especificación del tipo de sombreado, se hace mediante la invocación del método
 - `glShadeModel(GLenum)`
- Los posibles valores que podemos especificar son:
 - `GL_FLAT`: para un sombreado constante
 - `GL_SMOOTH`: para un sombreado de Gouraud

`glLight[if]{v}`

- Especifica los atributos de una de las luces del sistema. Se asegura el soporte de ocho luces, pero pueden ser más según la implementación.
- Para habilitar/deshabilitar una luz en particular, se utiliza el parámetro `GL_LIGHTi`, donde
 - $0 < i < \text{GL_MAX_LIGHTS}$
 - Se cumple que $\text{GL_LIGHT}i = \text{GL_LIGHT}0 + i$
- Para habilitar/deshabilitar los cálculos de la iluminación, se utiliza el parámetro `GL_LIGHTING`.

`glLight[if]{v}`

- EL primer parámetro indica la luz. Se utiliza cualquier valor del tipo `GL_LIGHTi`, donde
 - `0 < i < GL_MAX_LIGHTS`
- El segundo parámetro indica el atributo al que le queremos cambiar el valor. Puede tener cualquiera de los siguiente valores (con los correspondiente valores del tercer parámetro):

`glLight[if]{v}`

- Dependiendo de si se va a especificar un único valor (`glLight[if]`) o un vector de valores (`glLight[if]v`), se define el conjunto de valores posibles a ser especificados en el segundo parámetro.

glLight[if] {v}

- `glLight[if]`:
 - `GL_SPOT_EXPONENT`
 - `GL_SPOT_CUTOFF`
 - `GL_CONSTANT_ATTENUATION`
 - `GL_LINEAR_ATTENUATION`
 - `GL_QUADRATIC_ATTENUATION`

glLight[if]{v}

- `glLight[if]v:`
 - `GL_AMBIENT`
 - `GL_DIFFUSE`
 - `GL_SPECULAR`
 - `GL_POSITION`
 - `GL_SPOT_CUTOFF`
 - `GL_SPOT_DIRECTION`
 - `GL_SPOT_EXPONENT`
 - `GL_CONSTANT_ATTENUATION`
 - `GL_LINEAR_ATTENUATION`
 - `GL_QUADRATIC_ATTENUATION`

`glLight[if]{v}`

- **GL_ambient**: cuatro valores especifican el **RGBA** ambiente de la luz.
 - El valor inicial para todas las luces es $(0, 0, 0, 1)$
- **GL_diffuse**: cuatro valores especifican el **RGBA** difuso de la luz.
 - El valor inicial para **GL_LIGHT0** es $(1, 1, 1, 1)$, para el resto es $(0, 0, 0, 0)$

`glLight[if] {v}`

- **GL_SPECULAR**: cuatro valores especifican el **RGBA** especular de la luz.
 - El valor inicial para **GL_LIGHT0** es $(1, 1, 1, 1)$, para el resto es $(0, 0, 0, 0)$
- **GL_POSITION**: cuatro valores especifican las coordenadas de la luz.
 - Si el cuarto valor es 0 (cero), significa que se está definiendo una luz direccional.
 - Si tiene otro valor, se está definiendo una luz posicional.

`glLight[if]{v}`

- **GL_SPOT_DIRECTION**: tres valores especifican la dirección de la luz.
 - Solo tiene interés cuando **GL_SPOT_CUTOFF** no es 180 (valor inicial).
 - La dirección inicial es $(0, 0, -1)$
- **GL_SPOT_EXPONENT**: un único valor indica la distribución de la intensidad de la luz a medida que aumenta el ángulo con respecto al foco central (dirección) de la misma.
 - Cualquier valor en el intervalo $[0, 128]$ es permitido.

`glLight[if]{v}`

- **GL_CONSTANT_ATTENUATION**: un único valor indica el componente constante de la ecuación de atenuación de la luz.
- **GL_LINEAR_ATTENUATION**: un único valor indica el componente lineal de la ecuación de atenuación de la luz.
- **GL_QUADRATIC_ATTENUATION**: un único valor indica el componente cuadrático de la ecuación de la atenuación de la luz.

`glLight[if]{v}`

- El valor inicial para los últimos tres atributos es:
 - `GL_CONSTANT_ATTENUATION:` 1
 - `GL_LINEAR_ATTENUATION:` 0
 - `GL_QUADRATIC_ATTENUATION:` 0
- Lo cual indica que no hay atenuación de la luz.

glMaterial[if]{v}

```
glMaterial[if]{v} ( face, pname, param )
```

- Especifica los atributos del material actual.
- Esta información es utilizada para el cálculo de la iluminación.
- Para un mismo polígono, hay dos grupos de parámetros de materiales:
 - **FRONT**: cara anterior
 - **BACK**: cara posterior

`glMaterial[if]{v}`

- El primer parámetro puede tener los valores:
 - `GL_FRONT`: material asignado a la cara anterior
 - `GL_BACK`: material asignado a la cara posterior
 - `GL_FRONT_AND_BACK`: material asignado a ambas caras.
- Dependiendo de si se va a especificar un único valor (`glMaterial[if]`) o un vector de valores (`glMaterial[if]v`), se define el conjunto de valores posibles a ser especificados en el segundo parámetro.

glMaterial[if]{v}

- `glMaterial[if]:`
 - `GL_SHININESS`
- `glMaterial[if]V:`
 - `GL_AMBIENT`
 - `GL_DIFFUSE`
 - `GL_SPECULAR`
 - `GL_EMISSION`
 - `GL_SHININESS`
 - `GL_AMBIENT_AND_DIFFUSE`

glMaterial[if]{v}

- **GL_ambient**: cuatro valores especifican la reflectividad ambiente del material.
- **GL_diffuse**: cuatro valores especifican la reflectividad difusa del material.
- **GL_specular**: cuatro valores especifican la reflectividad especular del material.

`glmMaterial [if] {v}`

- **GL_EMISSION**: cuatro valores especifican el color emitido por el material. Se lo utiliza para modelar objetos como lámparas de superficie (emisores de luz no puntuales), pero su valor no afecta al cálculo de iluminación de los otros objetos en la escena.
- **GL_SHININESS**: un único parámetro indica el exponente de brillo especular del material.

glMaterial[if]{v}

- `GL_ambient_and_diffuse`: equivalente a llamar `glMaterial(GL_ambient)` y luego `glMaterial(GL_diffuse)` con los mismos parámetros.