

Iluminação em OpenGL

André Tavares da Silva

andre.silva@udesc.br

Iluminação em OpenGL

- Assume fontes pontuais de luz
 - **Omnidirecionais**: todas as direções.
Ex. : sol (direcional), luz incandescente (pontual);
 - **Spot** : um feixe de luz.
Exemplo: lanterna, luminária de mesa, holofote
- Interações de luz com superfície modeladas em componentes (modelo de *Phong*):
 - Emissão
 - Ambiente
 - Difusa
 - Especular

Iluminação em OpenGL

- Suporte a efeitos atmosféricos como
 - *Fog*
 - Atenuação
- Modelo de iluminação é calculada apenas nos vértices das superfícies
 - Cor dos demais pixels é interpolada linearmente (sombreamento *Gouraud*)

Fontes de Luz

- Para ligar uma fonte: **glEnable (source) ;**
 - **source** é uma constante cujo nome é **GL_LIGHT_i**, começando com **GL_LIGHT0**
 - Quantas? Pelo menos 8, mas para ter certeza:
 - **glGetIntegerv(GL_MAX_LIGHTS, &n) ;**
- Não esquecer de ligar o cálculo de cores pelo modelo de iluminação
 - **glEnable (GL_LIGHTING) ;**

Fontes de Luz

- Para configurar as propriedades de cada fonte:
`glLightfv(source, property, value);`
 - **Property** é uma constante designando:
 - Coeficientes de cor usados no modelo de iluminação:
 - **`GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR`**
 - Geometria da fonte
 - **`GL_POSITION, GL_SPOT_DIRECTION, GL_SPOT_CUTOFF, GL_SPOT_EXPONENT`**
 - Coeficientes de atenuação
 - **`GL_CONSTANT_ATTENUATION, GL_LINEAR_ATTENUATION, GL_QUADRATIC_ATTENUATION`**

Exemplo

```
GLfloat light0_ambient[] = {0.0, 0.1, 0.0, 1.0};  
GLfloat light0_diffuse[] = {0.0, 0.0, 1.0, 1.0};  
GLfloat light0_specular[] = {1.0, 1.0, 1.0, 1.0};  
GLfloat light0_position[] = {1.0, 2.0, 3.0, 1.0};  
glLightfv(GL_LIGHT0, GL_POSITION, light0_position);  
glLightfv(GL_LIGHT0, GL_AMBIENT, light0_ambient);  
glLightfv(GL_LIGHT0, GL_DIFFUSE, light0_diffuse);  
glLightfv(GL_LIGHT0, GL_SPECULAR, light0_specular);  
glEnable(GL_LIGHT0);  
glEnable(GL_LIGHTING);
```

Fontes de Luz (relembrando...)

- Para configurar as propriedades de cada fonte:
`glLightfv(source, property, value);`
 - **Property** é uma constante designando:
 - Coeficientes de cor usados no modelo de iluminação:
 - **`GL_AMBIENT, GL_DIFFUSE, GL_SPECULAR`**
 - Geometria da fonte
 - **`GL_POSITION, GL_SPOT_DIRECTION, GL_SPOT_CUTOFF, GL_SPOT_EXPONENT`**
 - Coeficientes de atenuação
 - **`GL_CONSTANT_ATTENUATION, GL_LINEAR_ATTENUATION, GL_QUADRATIC_ATTENUATION`**

Como criar uma luz direcional?

Exemplo

```
GLfloat light0_ambient[] = {0.0, 0.1, 0.0, 1.0};  
GLfloat light0_diffuse[] = {0.0, 0.0, 1.0, 1.0};  
GLfloat light0_specular[] = {1.0, 1.0, 1.0, 1.0};  
GLfloat light0_position[] = {1.0, 2.0, 3.0, 0.0}; ←  
glLightfv(GL_LIGHT0, GL_POSITION, light0_position);  
glLightfv(GL_LIGHT0, GL_AMBIENT, light0_ambient);  
glLightfv(GL_LIGHT0, GL_DIFFUSE, light0_diffuse);  
glLightfv(GL_LIGHT0, GL_SPECULAR, light0_specular);  
glEnable(GL_LIGHT0);  
glEnable(GL_LIGHTING);
```

Isso significa que esse vetor agora será interpretado como uma direção, e não mais como uma posição.

Propriedades de Material

- Especificados por
`glMaterialfv` (*face*, *property*, *value*)
 - ***Face*** define quais lados da superfície se quer configurar:
 - `GL_FRONT`, `GL_BACK`, `GL_FRONT_AND_BACK`
 - ***Property*** a propriedade do modelo de iluminação:
 - `GL_AMBIENT`, `GL_DIFFUSE`, `GL_SPECULAR`,
`GL_EMISSION`, `GL_SHININESS`

Exemplo

```
GLfloat mat_ambient[] = {0.2, 0.2, 0.2, 1.0};  
GLfloat mat_diffuse[] = {0.7, 0.0, 0.0, 1.0};  
GLfloat mat_specular[] = {1.0, 1.0, 1.0, 1.0};  
GLfloat mat_shininess[] = {5.0};  
glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);  
glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);  
glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);  
glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
```

Valores **default** OpenGL

- Fonte de luz: `GL_LIGHT0`
 - Luz difusa e especular branca!
 - Não existe componente ambiente
 - posição: infinito $(0,0,1,0)$ // $Z+$
 - atenuação constante: 1.0
- Material: (Reflexão)
 - Reflexão de 80% da luz difusa e 20% da luz de ambiente, 0% de especular, 0% de emissiva;
 - brilho=0.0;

Modelo de Sombreamento (*shade*)

- Especificados por
`glShadeModel (GLenum mode) ;`
 - *mode* estabelece o modo de colorização:
 - `GL_FLAT`, `GL_SMOOTH`

`GL_FLAT` – cor da face não varia

`GL_SMOOTH` – cor da face é calculada pela interpolação dos vértices (Gouraud)

Modelo de Sombreamento (*shade*)

- Especificados por
`glShadeModel (GLenum mode) ;`
 - *mode* estabelece o modo de colorização:
 - `GL_FLAT`, `GL_SMOOTH`

`GL_FLAT` – cor da face não varia

`GL_SMOOTH` – cor da face é calculada pela interpolação dos vértices (Gouraud)

Phong?

Exercício

- O primeiro passo é ler com atenção os slides que falam sobre iluminação;
- O segundo passo é compilar e executar o exemplo;
- Veja o que acontece quando desabilitamos a iluminação;
(**glEnable(GL_LIGHTING)** ;)
- Veja o que acontece quando desabilitamos a “luz 0”;
(**glEnable(GL_LIGHT0)** ;)
- Transforme a luz pontual em uma luz direcional;
- Veja o que acontece se mudarmos o modelo de sombreamento para *flat shading*;
- Troque a esfera pelo “Utah *teapot*” (`glutSolidTeapot`).

Exercício

- Insira uma luz do tipo *spot* de cor amarela apontada para o objeto;
- Mude a distância da luz *spot* ao objeto;
- Coloque dois retângulos lado a lado;
- Faça com que ambos fiquem a um certo ângulo entre eles (como uma folha dobrada);
- Altere a normal dos vértices de forma que seja a média da normal das duas faces.