



## Iluminação

*Definição de luzes, materiais e normais*



# OpenGL - Normais

- Para se computar a iluminação é necessário definir uma normal por vértice.
- A normal é um vector perpendicular à superfície que se pretende iluminar.

```
glBegin(GL_TRIANGLE);  
    glNormal3f(0,1,0);  
    glVertex3f(0,0,0);  
    glNormal3f(0,1,0);  
    glVertex3f(0,0,1);  
    glNormal3f(0,1,0);  
    glVertex3f(1,0,0);  
glEnd();
```

No caso de se utilizar a mesma normal para todos os vértices escreve-se:

```
glBegin(GL_TRIANGLE);  
    glNormal3f(0,1,0);  
    glVertex3f(0,0,0);  
    glVertex3f(0,0,1);  
    glVertex3f(1,0,0);  
glEnd();
```



# OpenGL - Normais com VBOs

---

- Criação dos VBOs
  - Passo 1 a) Activar Buffers

```
glEnableClientState(GL_VERTEX_ARRAY);  
glEnableClientState(GL_NORMAL_ARRAY);
```



# OpenGL - Normais com VBOs

- Criação dos VBOs

- Passo 1 b - Alocar e Preencher os arrays de vértices e normais

```
// array de vértices
float *vertexB;
// preencher array com os valores dos vértices

// array de normais
float *normalB;
// preencher array com os valores das normais dos vértices
...
```

- Passo 1 c (opcional) - Alocar e Preencher um array com os índices

```
unsigned int *indices;
```

```
...
```



# OpenGL - Normais com VBOs

- Criação dos VBOs

- Passo 1 d : Gerar VBOs

```
GLuint buffers[2];  
// arrays de vértices, normais, etc...  
float *vertexB, *normalB;  
...  
glGenBuffers(2, buffers);  
glBindBuffer(GL_ARRAY_BUFFER, buffers[0]);  
glBufferData(GL_ARRAY_BUFFER, arraySize, vertexB,  
             GL_STATIC_DRAW);  
  
glBindBuffer(GL_ARRAY_BUFFER, buffers[1]);  
glBufferData(GL_ARRAY_BUFFER, arraySize, normalB,  
             GL_STATIC_DRAW);
```



# OpenGL - Normais com VBOs

---

- Desenhar com VBOs
  - Passo 2 a - Semântica
    - Indicar para cada buffer qual a sua utilização e composição

```
glBindBuffer(GL_ARRAY_BUFFER, buffers[0]);  
glVertexAttribPointer(3, GL_FLOAT, 0, 0);
```

```
glBindBuffer(GL_ARRAY_BUFFER, buffers[1]);  
glNormalPointer(GL_FLOAT, 0, 0);
```



# OpenGL - Normais com VBOs

---

- Desenhar com VBOs
  - Passo 2 b: Instrução de desenho
    - Desenho recorrendo a uma lista de índices

```
glDrawElements(GL_TRIANGLES, count ,GL_UNSIGNED_INT, indices);
```

- Desenho sem lista de índices

```
glDrawArrays(GL_TRIANGLES, first, count);
```

Nota: count é o número de vértices, ou de índices, a desenhar



# OpenGL - Materiais

- Atribuir materiais:

```
glMaterialfv(GL_FRONT, componente, array);  
glMaterialf(GL_FRONT, GL_SHININESS, valor);
```

0..128

## Componente:

```
GL_DIFFUSE  
GL_AMBIENT  
GL_SPECULAR  
GL_EMISSION  
GL_AMBIENT_AND_DIFFUSE
```

- Exemplo: definir a cor difusa a vermelho

```
float vermelho[4] = {0.8f, 0.2f, 0.2f, 1.0f};  
glMaterialfv(GL_FRONT, GL_DIFFUSE, vermelho);
```





# OpenGL - Iluminação

---

- Definir propriedades da luz

```
glLight{if}(GL_LIGHTi, param, valor1, valor2, ...);  
glLight{if}v(GL_LIGHTi, param, array_valores);
```

- Nota 1: A posição de uma luz fixa no mundo deve ser definida após a instrução `gluLookAt`.
- Nota 2: As cores associadas à luz devem ser definidas na inicialização



# OpenGL - Iluminação

- Definir uma luz direccional

```
GLfloat amb[3] = {0.2, 0.2, 0.2};  
GLfloat diff[3] = {1.0, 1.0, 1.0};  
GLfloat pos[4] = {0.0, 0.0, 1.0, 0.0};
```

0.0 indica que a luz é  
direccional

```
// posição da luz
```

```
glLightfv(GL_LIGHT0, GL_POSITION, pos);
```

```
// cores da luz
```

```
glLightfv(GL_LIGHT0, GL_AMBIENT, amb);
```

```
glLightfv(GL_LIGHT0, GL_DIFFUSE, diff);
```

A posição indica a  
direcção de onde  
vem a luz



# OpenGL - Iluminação

- Ligar/desligar luzes individuais (por omissão estão desligadas)

```
glEnable(GL_LIGHTi); // i = 0..7  
glDisable(GL_LIGHTi);
```

- Ligar/desligar o quadro (por omissão está desligado)

```
glEnable(GL_LIGHTING);  
glDisable(GL_LIGHTING);
```

- **Nota:** As instruções acima indicadas devem ser definidas na inicialização.



# Exercício

---

- Definir normais apropriadas para os vértices do cilindro