

# OpenGL

## Iluminação



Renato Rodrigues Oliveira da Silva  
Outubro 2013

# Sumário

---

## # Iluminação

- ▣ Fontes de luz
- ▣ Modelos de reflexão
- ▣ Vetores normais
- ▣ Materiais
- ▣ Tonalização *Flat* e *Gourand*

# Iluminação – Fontes de Luz

- # Modelo de iluminação define a natureza de uma fonte de luz e sua interação com a cena
- # A OpenGL permite adicionar até 8 fontes de luz na cena
- # Para habilitar o uso de iluminação
  - glEnable(GL\_LIGHTING);
  - glEnable(GL\_LIGHT0); //0 a 7

# Iluminação – Fontes de Luz

## # Propriedades de uma fonte de luz:

- ▣ `glLightfv(GLenum light, GLenum pname, TYPE *param)`
- ▣ **light** – Identificador da fonte de luz (GL\_LIGHT0 a GL\_LIGHT7)
- ▣ **pname** – característica (GL\_AMBIENT, GL\_DIFFUSE, GL\_SPECULAR, GL\_POSITION, ...)
- ▣ **param** – valor

# Iluminação – Modelo de iluminação

# Para definir propriedades do modelo de iluminação:

▣ void glLightModelfv(GLenum pname,  
TYPE \*param);

▣ **pname** – propriedade do modelo  
(GL\_LIGHT\_MODEL\_AMBIENT,  
GL\_LIGHT\_MODEL\_TWO\_SIDE, etc.)

▣ **param** – valor

# Exemplo

- # Baixar e compilar o arquivo
  - **03-01-Exemplo3DComIluminacao.c**

# Iluminação

## Modelos de Reflexão

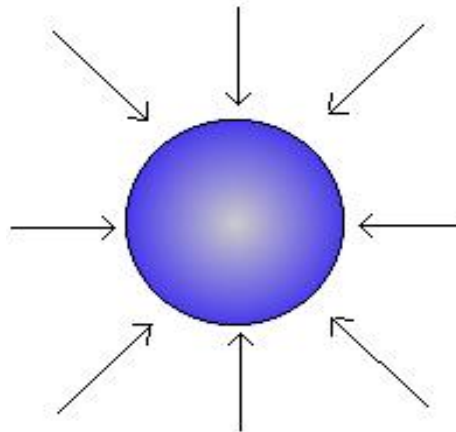
- # Descrevem a interação dos raios de luz com uma superfície
- # Considera as propriedades da superfície e a natureza da fonte de luz incidente
  - Cor do objeto
  - Cor da fonte de luz
  - Posição da fonte de luz, etc.
- # Modelos utilizados: reflexão ambiente, difusa e especular

# Iluminação

## Modelos de Reflexão

### # Reflexão Ambiente

- Considera-se a existência de uma fonte de luz não direcional
- Origina-se da interação e da reflexão dos raios de luz com todas as superfícies da cena



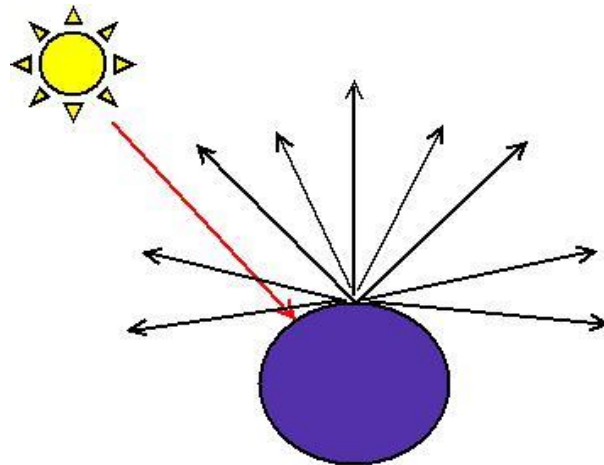


# Iluminação

## Modelos de Reflexão

### # Reflexão Difusa

- ▣ Os objetos absorvem e refletem, em todas as direções, parte da luz incidente
- ▣ Depende da cor do material
- ▣ Cria o efeito de *dégradé* nos objetos

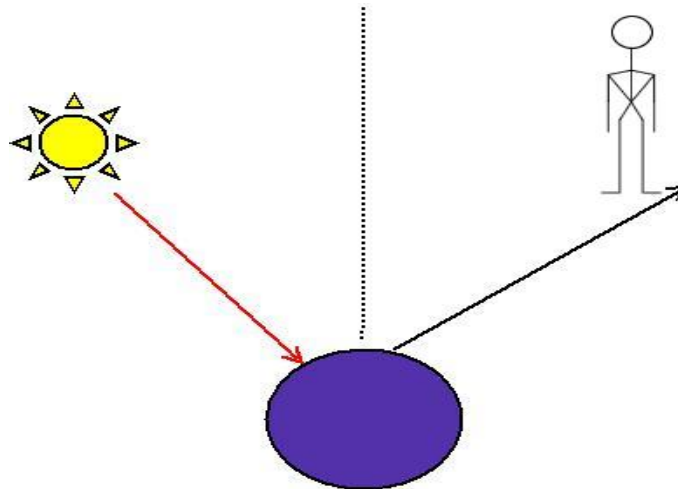


# Iluminação

## Modelos de Reflexão

### # Reflexão Especular

- Responsável pela geração do “ponto de brilho” dos objetos
- Depende da cor do objeto, posição da fonte de luz e do observador



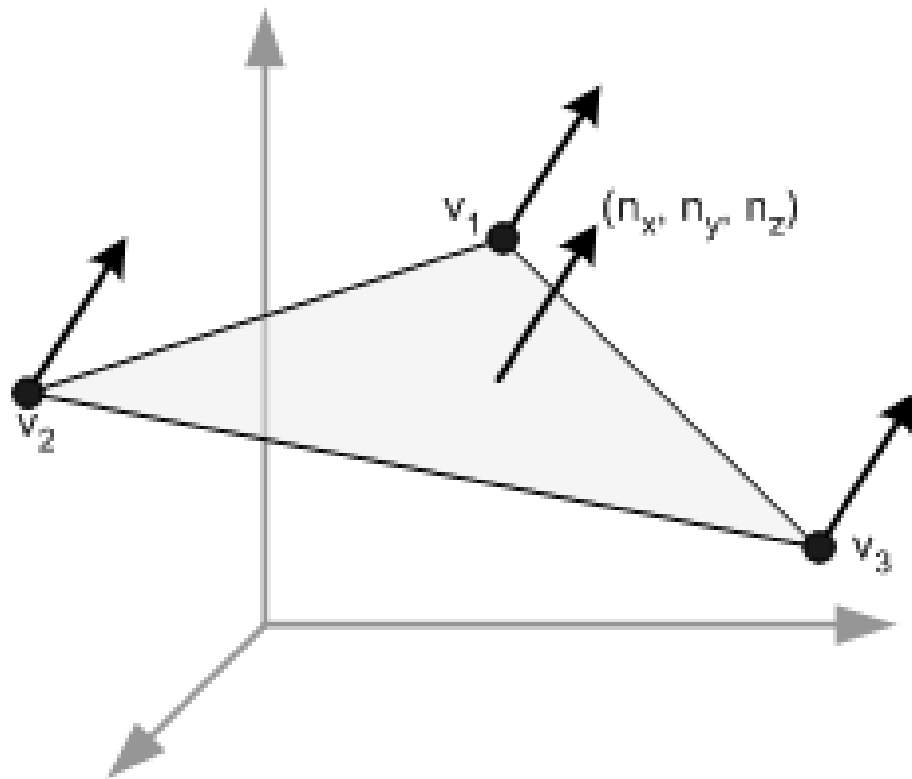
# Exemplo

- # Baixar e compilar o arquivo
  - ▣ **03-01-Exemplo3DComIluminacao.c**
  - ▣ (alterar parâmetros da fonte de luz)

# Vetores Normais

- # Um vetor normal a uma face é perpendicular a essa face.
- # São importantes para definir o realismo de uma cena
  - Quais são as faces visíveis?
  - Qual a cor de cada ponto da face?
- # Alguns objetos pré-definidos possuem as normais de cada face. Para outros é necessário computar.

# Vetores Normais



**Vetor normal a uma face**

# Vetores Normais

# Procedimento para computar o vetor normal:

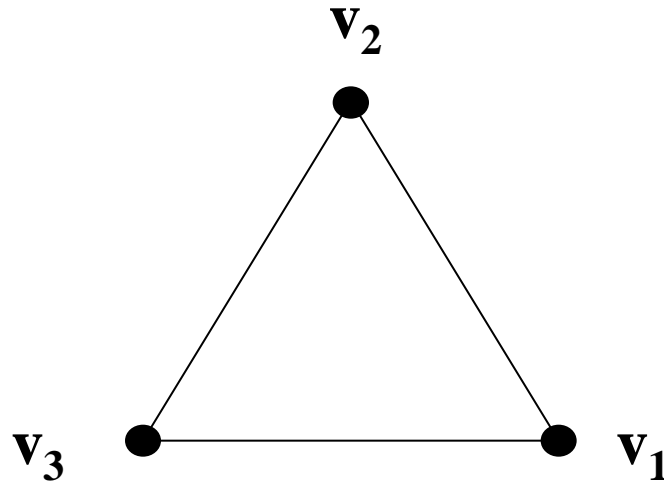
1. Calcular dois vetores ( $v_1$  e  $v_2$ ) a partir dos vértices da face
2. Dividir cada vetor pelo seu módulo, para que seja unitário
3. Calcular o produto vetorial entre os vetores unitários  $v_1$  e  $v_2$
4. O vetor resultante é o normal da face

# Vetores Normais

- # Em OpenGL o vetor normal deve ser informado antes dos vértices que compõem a face
  - ▣ **void glNormal3f(GLfloat nx, ny, nz)**
  - ▣ **nx, ny e nz** – Componentes do vetor normal nos eixos cartesianos

# Vetores Normais

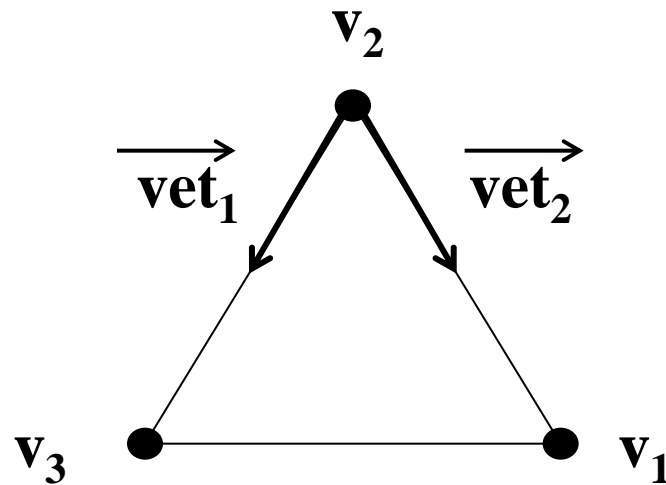
# Exemplo: Normal de um triângulo





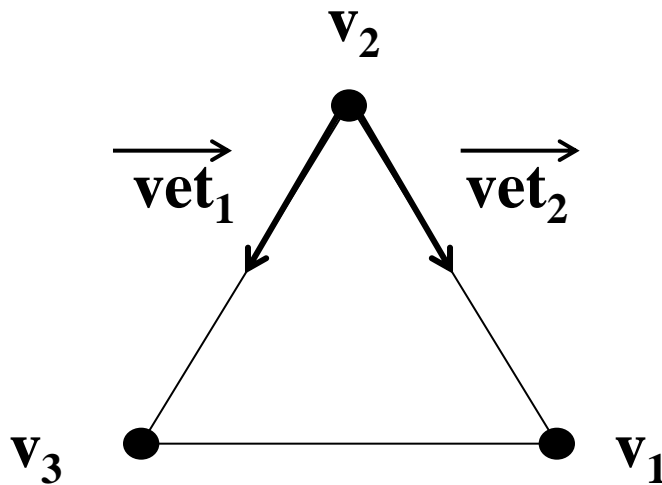
# Vetores Normais

# Exemplo:



# Vetores Normais

# Exemplo:



$$|\vec{v}| = \sqrt{x^2 + y^2 + z^2}$$

$$\vec{vet}_1 = \frac{\mathbf{v}_3 - \mathbf{v}_2}{|\mathbf{v}_3 - \mathbf{v}_2|}$$

$$\vec{vet}_2 = \frac{\mathbf{v}_1 - \mathbf{v}_2}{|\mathbf{v}_1 - \mathbf{v}_2|}$$

# Vetores Normais

# Exemplo:

$$\vec{n}_x = \vec{vet1}_y * \vec{vet2}_z - \vec{vet1}_z * \vec{vet2}_y$$

$$\vec{n}_y = \vec{vet1}_z * \vec{vet2}_x - \vec{vet1}_x * \vec{vet2}_z$$

$$\vec{n}_z = \vec{vet1}_x * \vec{vet2}_y - \vec{vet1}_y * \vec{vet2}_x$$

# Vetores Normais

# Exemplo:

- **03-02-CuboNormaisRGB.c**
- Interação
  - **A/a:** Aumenta/diminui luz Ambiente
  - **D/d:** Aumenta/diminui luz Difusa

# Materiais

- # Os objetos são compostos por materiais com diferentes propriedades que podem refletir ou emitir luz
- # A cor final da superfície é definida de acordo com as propriedades do material e da(s) fonte(s) de luz
- # É possível definir, para cada material, a componente especular, difusa, ambiente, etc.

# Materiais

- # Para definir propriedades do material:
  - ▣ void glMaterialfv(GLenum face, GLenum pname, TYPE \*param)
  - ▣ **face** – GL\_FRONT, GL\_BACK, GL\_FRONT\_AND\_BACK
  - ▣ **pname** – GL\_AMBIENT, GL\_DIFFUSE, GL\_AMBIENT\_AND\_DIFFUSE, GL\_SPECULAR, GL\_SHININESS, GL\_EMISSION, GL\_COLOR\_INDEXES
  - ▣ **param** – valor do parâmetro

# Materiais



- # Uma vez definidos os parâmetros do material, eles serão válidos para todos os objetos
- # Para criar objetos com diferentes materiais é necessário redefinir os parâmetros do material para cada objeto

# Materiais

# Exemplo:

■ **03-02-Exemplo3DComMateriais.c**

# Interação:

- **+ / -** : Aumenta ou diminui a concentração do brilho
- **e**: Habilita/Desabilita a emissão de luz
- **R/r, G/g, B/b** : Aumenta/Diminui a emissão de luz do componente RGB



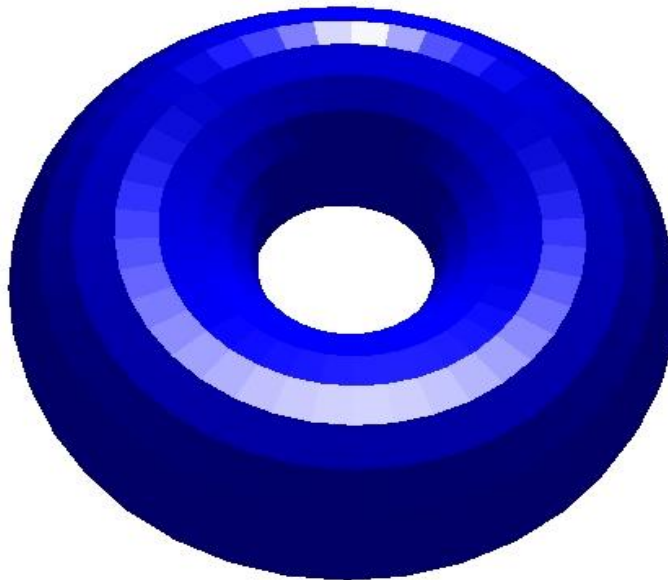
# Modelos de Tonalização

- # Determina a cor de cada ponto que compõe a superfície de um objeto
- # Define a variação de cor ao longo das faces dos objetos da cena
- # Depende das propriedades da superfície e da iluminação
- # A OpenGL suporta os modelos Flat e Gourand

# Modelos de Tonalização

## # Flat Shading

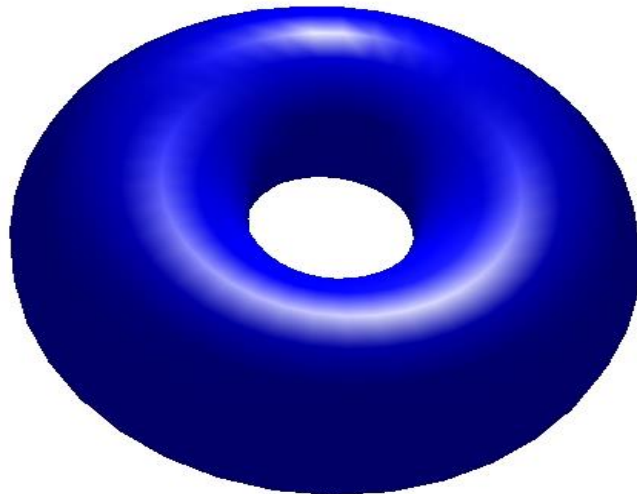
- É o modelo mais simples (+ rápido)
- Cada face tem apenas um valor de intensidade



# Modelos de Tonalização

## # Gourand Shading

- Exibe uma transição suave entre as faces que compõem os objetos
- Interpola as intensidades das cores ao longo de cada face



# Modelos de Tonalização

- # Para definir o modelo de tonalização
  - ▣ void glShadeModel (GLenum mode)
  - ▣ **mode** – GL\_FLAT (Flat Shading) ou GL\_SMOOTH (Gourand Shading)

# Exemplo

- # Baixar e compilar o arquivo
  - ▣ **03-01-Exemplo3DComIluminacao.c**
  - ▣ (alterar modelo de tonalização)

# Bibliografia

---

- # Cohen, M. , Manssour, I. H, OpenGL – Uma Abordagem Prática e Objetiva, 2006, Novatec Editora