

# WebGL: Sombreamento e Iluminação

Prof. Gilzamir Gomes

# Objetivo

- Aprender sobre fontes de luz, normais e materiais
- Aprender a diferença entre sombreamento e iluminação
- Uso de métodos de sombreamento de Goraud e de Phong
- Uso dos modelos de iluminação lambertianos e de Phong
- Definir o uso de *uniforms*, *attributes* e *varyings*

# Iluminação de Cena

- Objetos são vistos porque refletem luz dependendo de:
  - Posição da fonte de luz e distância relativa para a fonte de luz,
  - Orientação de sua superfície (vetor normal)
  - Material

## Scene Lighting

Light Source

Camera

Materials

Different materials reflect different amounts of light.

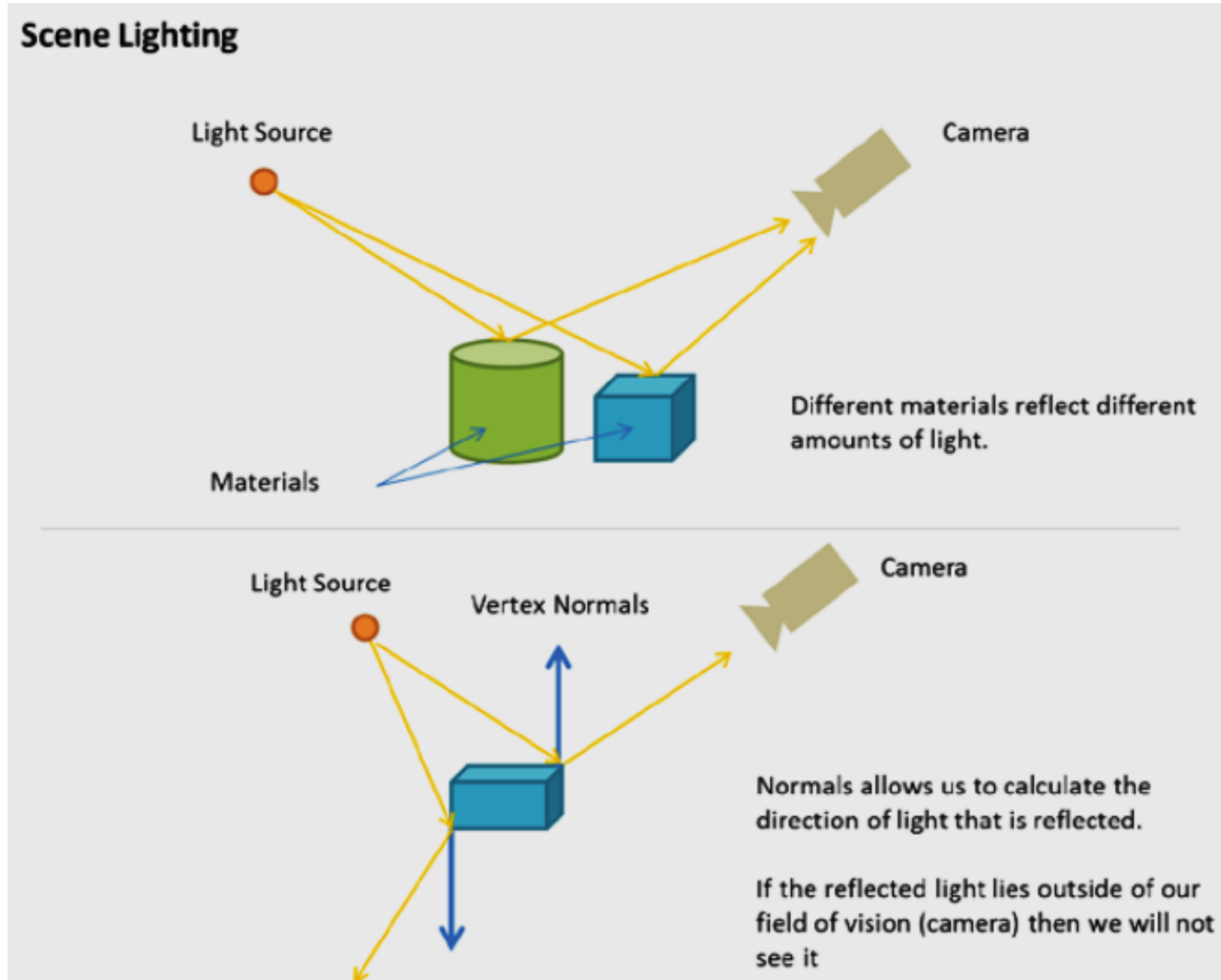
Light Source

Vertex Normals

Camera

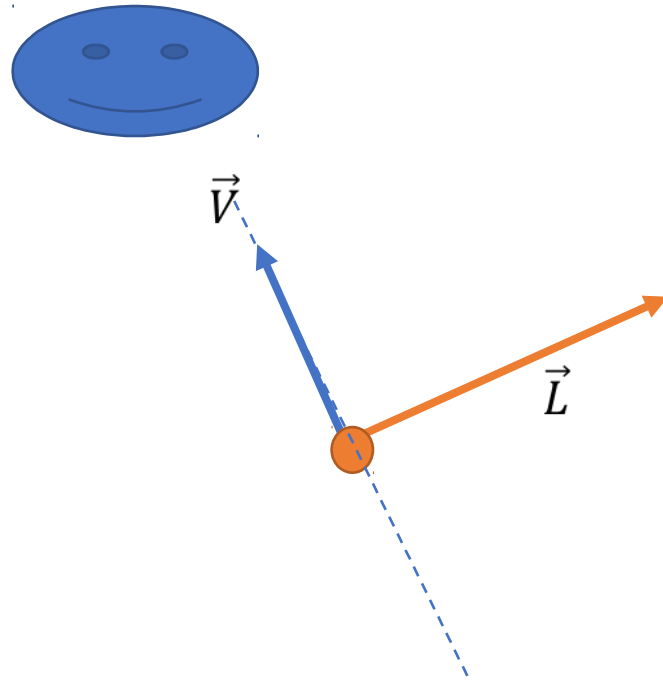
Normals allows us to calculate the direction of light that is reflected.

If the reflected light lies outside of our field of vision (camera) then we will not see it



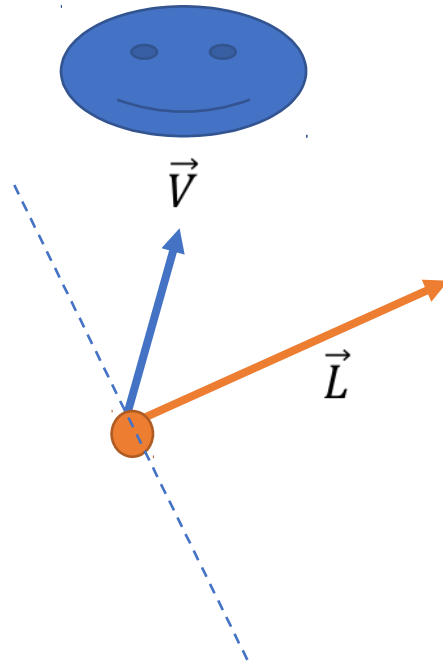
# Cálculo de Visibilidade

$$\vec{L} \cdot \vec{V} = 0$$



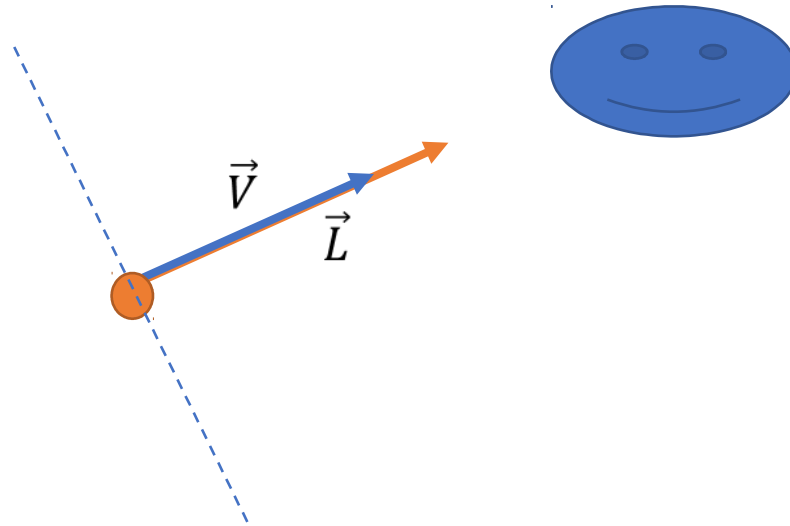
# Cálculo de Visibilidade

•  $0 < \vec{L} \cdot \vec{V} < 1$



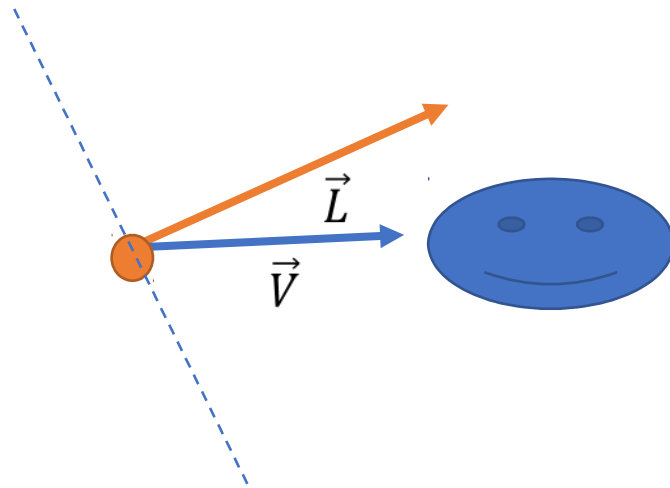
# Cálculo de Visibilidade

$$\vec{L} \cdot \vec{V} = 1$$



# Cálculo de Visibilidade

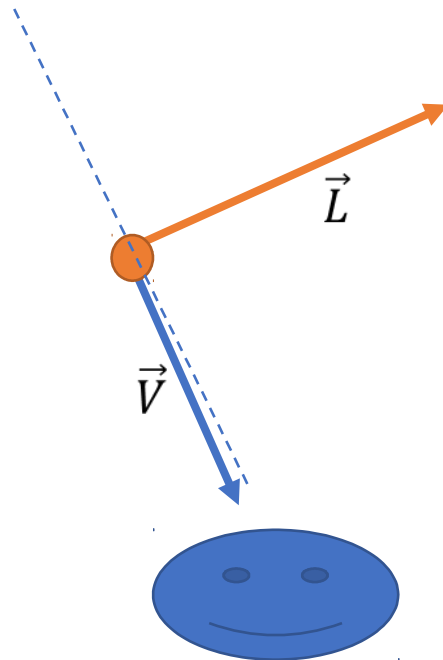
•  $0 < \vec{L} \cdot \vec{V} < 1$





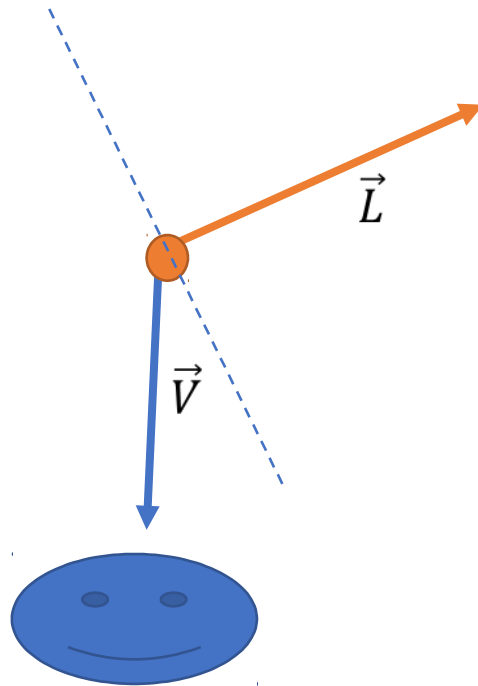
# Cálculo de Visibilidade

$$\vec{L} \cdot \vec{V} = 0$$



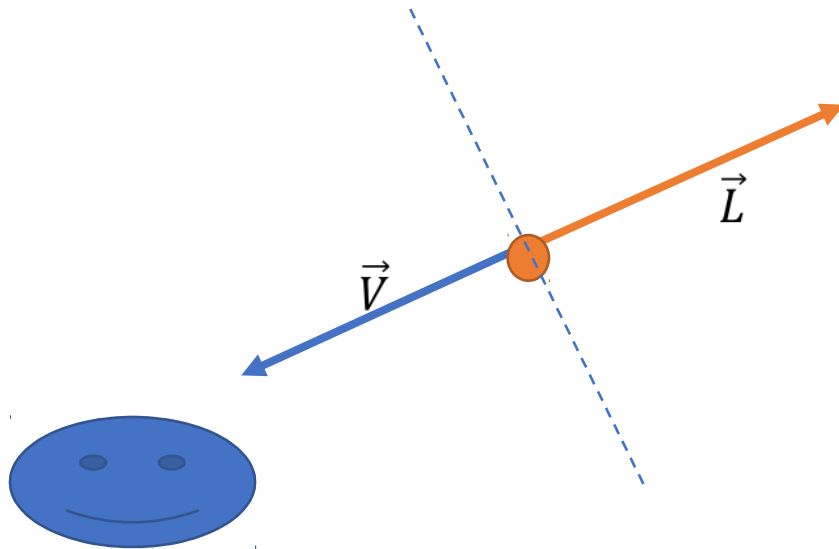
# Cálculo de Visibilidade

- $-1 < \vec{L} \cdot \vec{V} < 0$



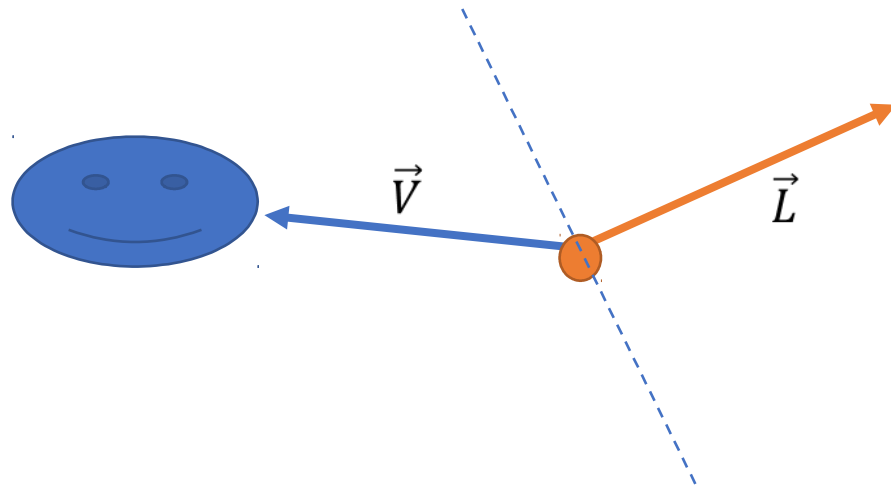
# Cálculo de Visibilidade

$$\vec{L} \cdot \vec{V} = -1$$



# Cálculo de Visibilidade

$$-1 < \vec{L} \cdot \vec{V} < 0$$

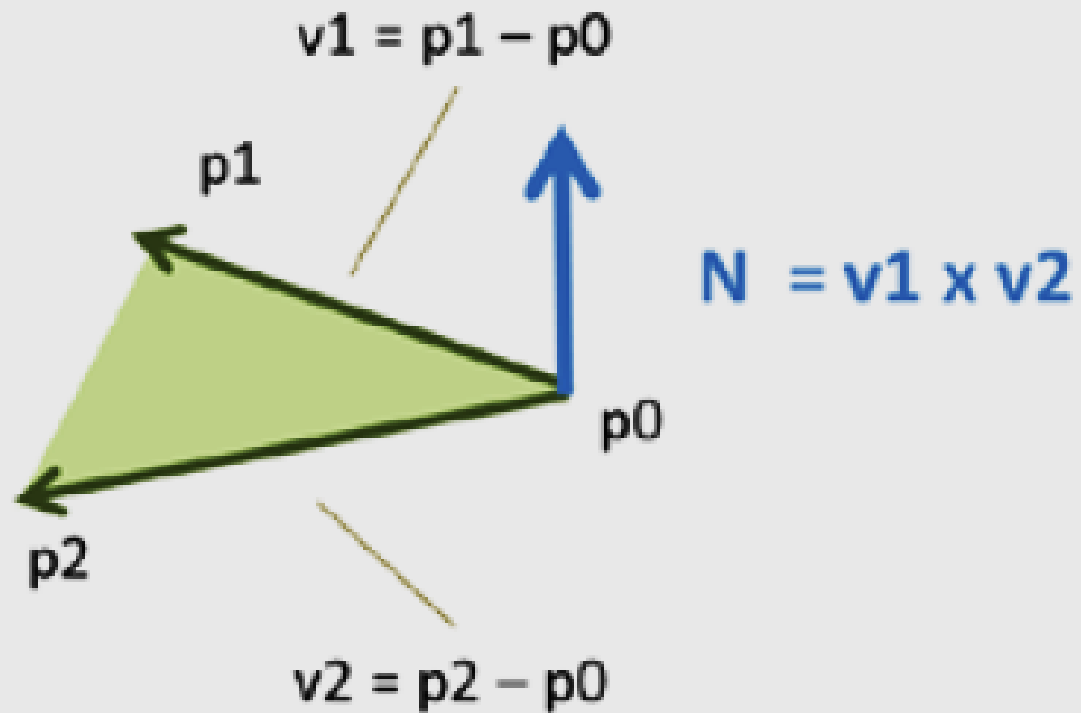


# Normais

- São vetores perpendiculares à superfície que queremos iluminar
- Representam a orientação da superfície
  - Necessários para modelar a interação entre a fonte de luz e o objeto
- Cada vértice de um objeto tem um vetor normal associado
- Utilizamos produto vetorial para calcularmos o vetor normal

# Calculando o vetor normal

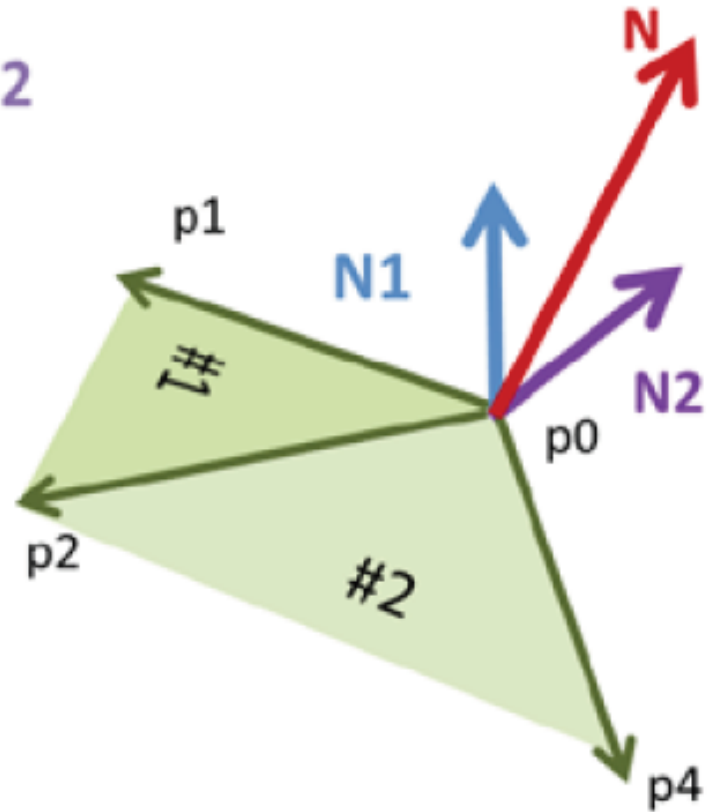
Calculating the normals



# Calculando o vetor normal (vértices compartilhados)

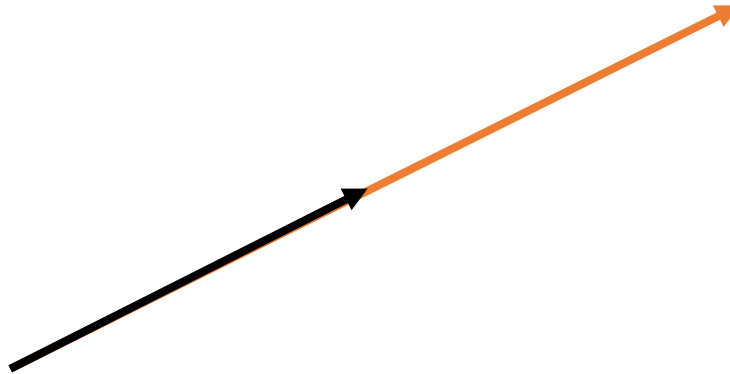
Updating normals for shared vertices

$$\mathbf{N} = \mathbf{N1} + \mathbf{N2}$$



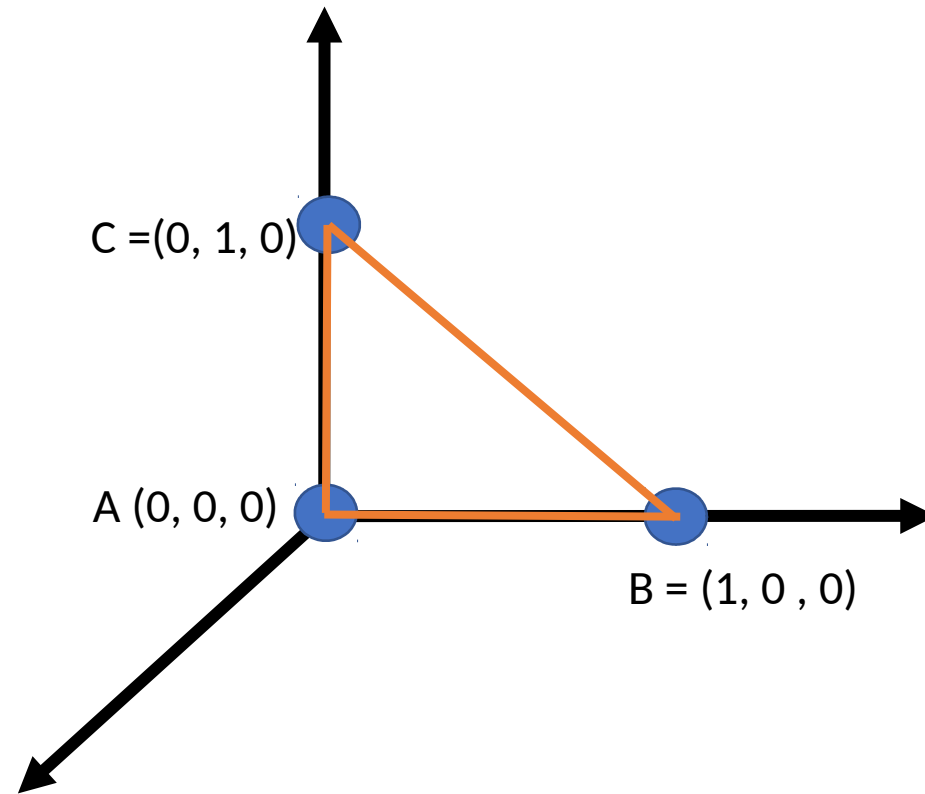
# Normalizando vetores

- $\vec{v} = \frac{\vec{v}}{||\vec{v}||}$





# Exemplo de cálculo da normal de um objeto

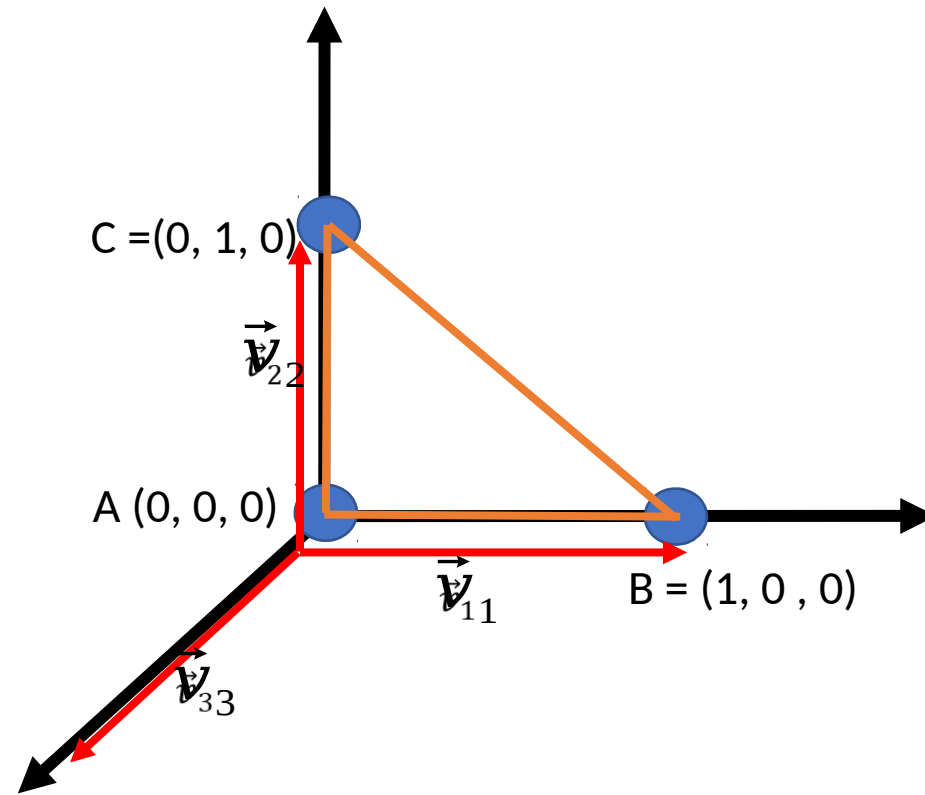


# Exemplo de cálculo da normal de um objeto

$$\vec{v}_1 = B - A$$
$$= [x_1, y_1, z_1]$$

$$\vec{v}_2 = C - A$$
$$= [x_2, y_2, z_2]$$

$$\vec{v}_1 \times \vec{v}_2 = \vec{v}_3$$
$$\vec{N} = \frac{\vec{v}_3}{\|\vec{v}_3\|}$$

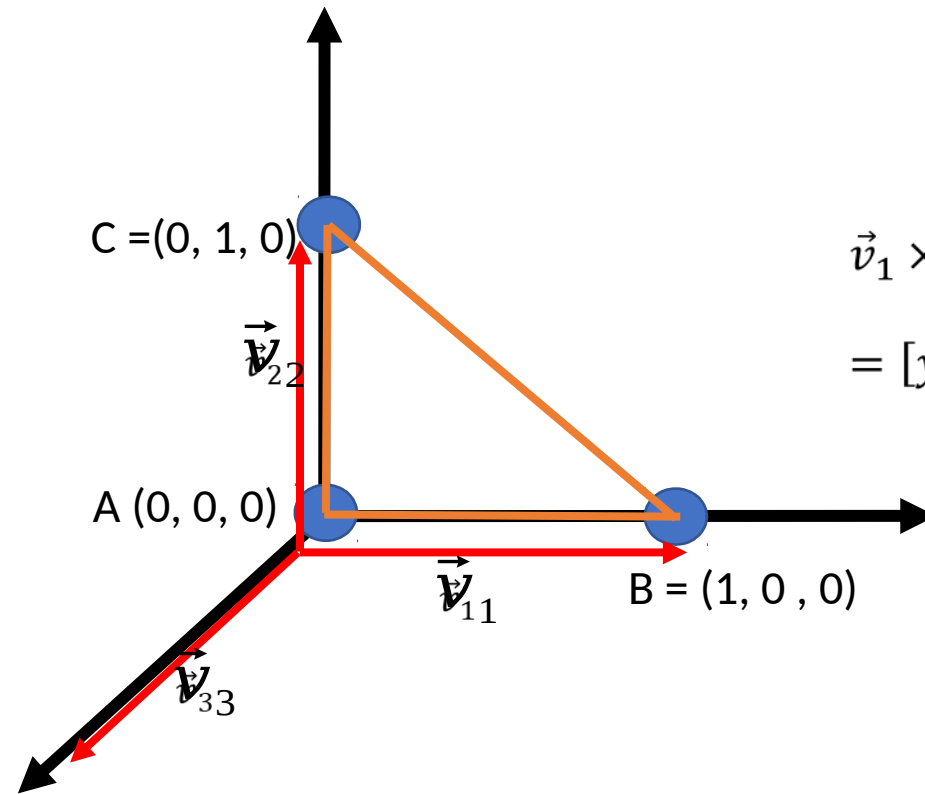


# Exemplo de cálculo da normal de um objeto

$$\vec{v}_1 = B - A \\ = [x_1, y_1, z_1]$$

$$\vec{v}_2 = C - A \\ = [x_2, y_2, z_2]$$

$$\vec{v}_1 \times \vec{v}_2 = \vec{v}_3 \\ \vec{N} = \frac{\vec{v}_3}{||\vec{v}_3||}$$



$$\vec{v}_1 \times \vec{v}_2 = \begin{vmatrix} i & j & k \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix}$$

$$\vec{v}_1 \times \vec{v}_2 = \begin{vmatrix} i & j & k & i & j \\ x_1 & y_1 & z_1 & x_1 & y_1 \\ x_2 & y_2 & z_2 & x_2 & y_2 \end{vmatrix} \\ = [y_1z_2 - y_2z_1, z_1x_2 - x_1z_2, x_1y_2 - x_2y_1]$$

$$\vec{v}_1 \times \vec{v}_2 = (0, 0, 1)$$

# Fonte de Luz

- Posicional
  - Modelada por um ponto no espaço
- Direcional
  - Modelada por um vetor (normalizado) que indica sua orientação

# Fonte de luz ambiente

- Geralmente, as cenas possuem uma quantidade de luz dissipada, constante, sem origem definida. Em computação gráfica criamos um componente de luz constante desse tipo chamada de luz ambiente.
- Assim, se um vértice reflete a cor vermelha (representada por  $[1.0, 0.0, 0.0]$ ) e há apenas luz ambiente em uma cena sendo que a cor da luz ambiente é  $[0.2, 0.2, 0.2]$ . Qual a cor refletida do vértice?
  - 20% do vermelho, ou seja,  $[0.2, 0.0, 0.0]$

# Outras fontes de luz

- Em uma cena, podemos ter uma fonte de luz pontual, cuja luz se espalha no ambiente em todas as direções. Nesse caso, a luz refletida pelos vértices depende apenas da posição da fonte de luz e da normal calculada no vértice. A intensidade da luz no vértice pode ser inversamente proporcional à distância à fonte de luz.
- Também podemos ter uma fonte de luz direcional, tal que a luz segue em uma direção definida e, portanto, esse tipo de fonte de luz é definida apenas por uma direção. Matematicamente, por um vetor.

# Materiais

- Definem como a superfície dos objetos interagem com a luz que chega até a superfície.
- Diferentes tipos de materiais podem ser modelados: superfícies rugosas, lisas, foscas, etc.

# Materiais

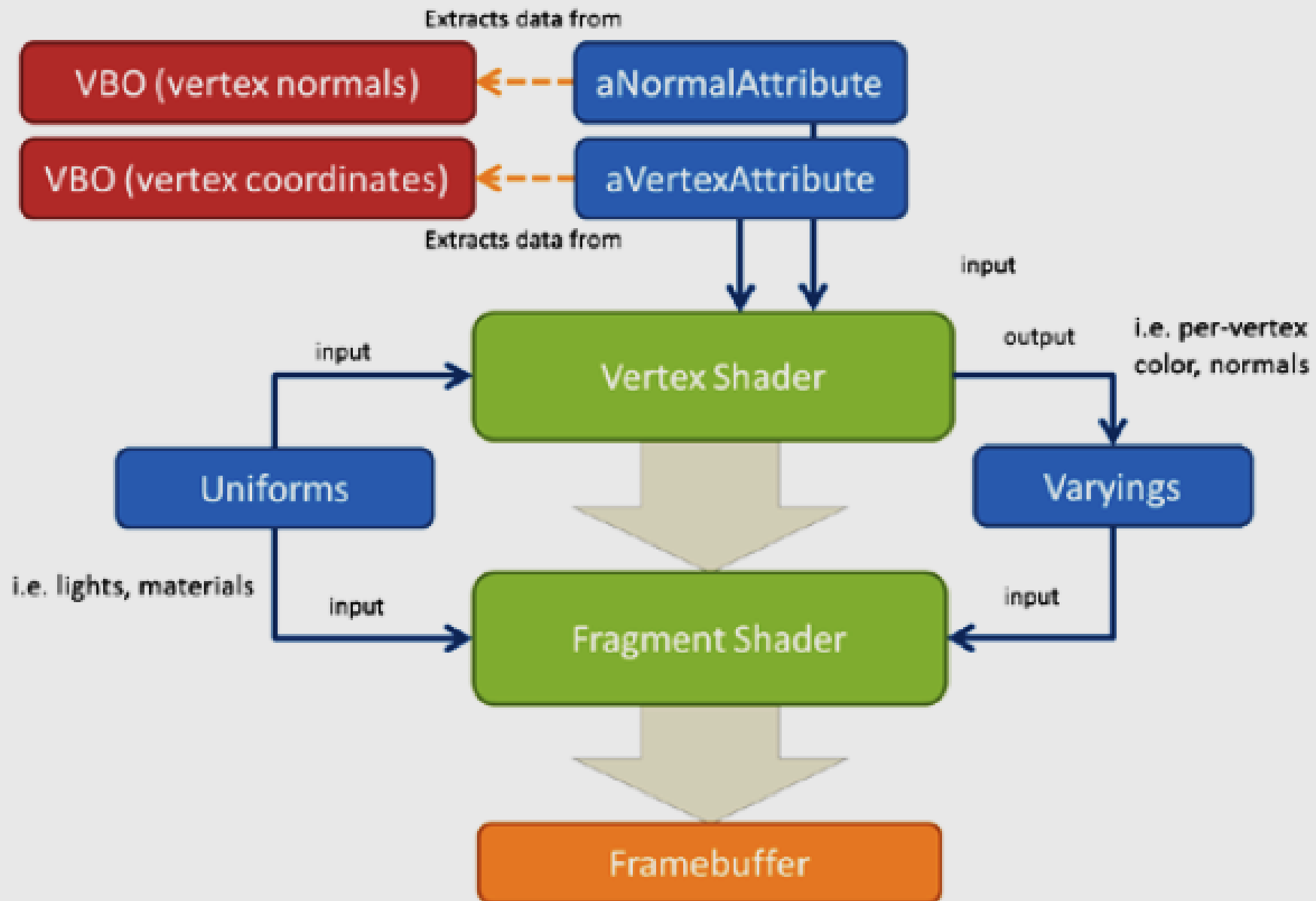
- Vários parâmetros
  - Cor
  - Textura
  - Seja  $I_a$  a intensidade da fonte de luz ambiente e  $C$  a cor do objeto.
  - A cor refletida pelo objeto, levando-se em conta apenas a cor ambiente, pode ser obtida por:
    - $R(i) = C(i) * I_a(i)$ , onde  $i \in \{r, g, b\}$



# Passando as informações necessárias para WebGL

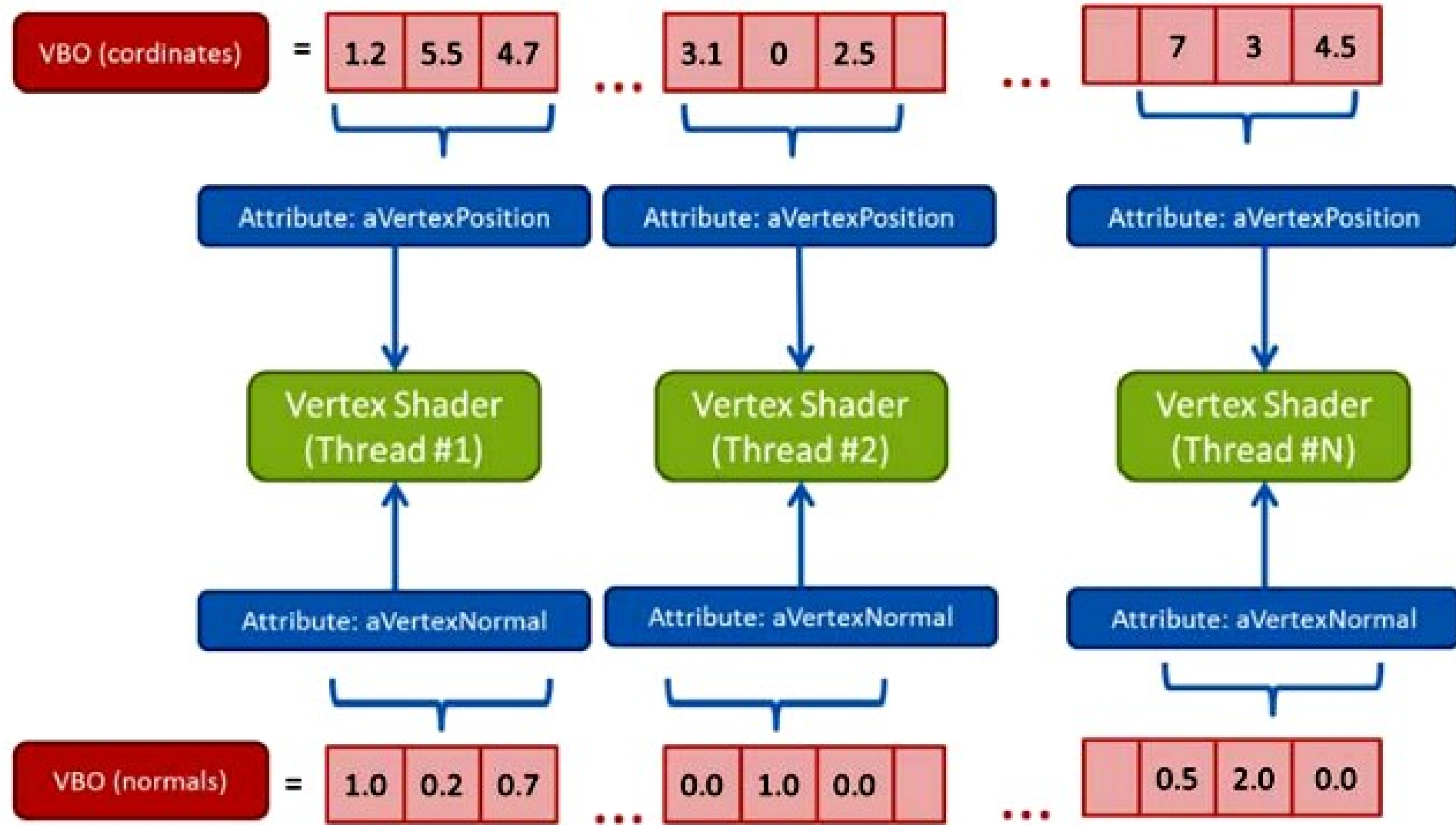
- Para cada vértice, temos que passar a normal desse vértice, portanto, é preciso criarmos um ARRAY BUFFER para passarmos dados de normais.

## WebGL Rendering Pipeline Revisited



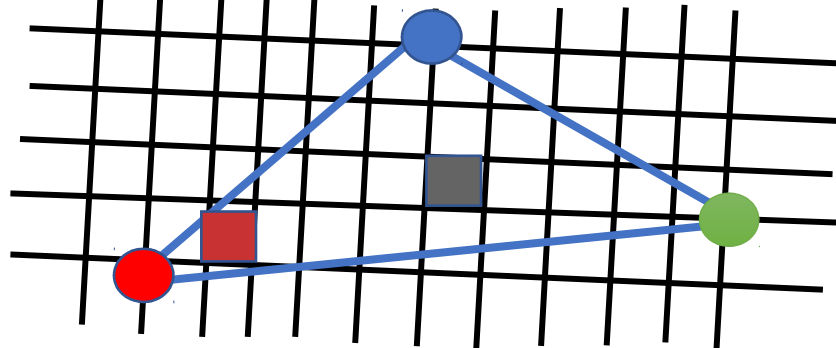
## Parallel processing in the Vertex Shader

The number of threads depends on the local GPU capabilities



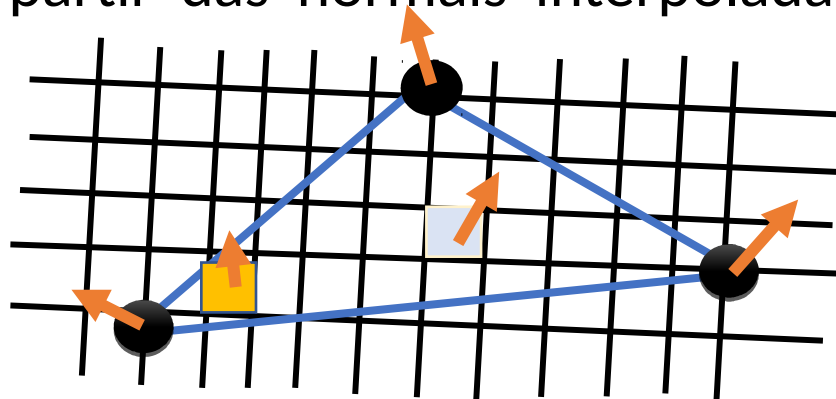
# Método de Sombreamento e Modelo de Reflexão da Luz

- Sombreamento: tipo de interpolação que é realizada para se obter a cor final para cada fragmento (pixel) na cena.
- O tipo de sombreamento define aonde a cor final será calculada: no *vertex shader* ou no *fragment shader*.
  - Sombreamento de Goraud: aplica-se o modelo de iluminação em cada vértice, obtendo a cor em cada vértice. A cor do pixel dentro de um triângulo é a interpolação das cores encontradas nos vértices.



# Método de Sombreamento e Modelo de Reflexão da Luz

- Sombreamento: tipo de interpolação que é realizada para se obter a cor final para cada fragmento (pixel) na cena.
- O tipo de sombreamento define aonde a cor final será calculada: no *vertex shader* ou no *fragment shader*.
  - Sombreamento de Phong: o modelo de iluminação é aplicado em cada fragmento a partir das normais interpoladas pelas normais calculadas nos vértices.



# Método de Sombreamento e Modelo de Reflexão da Luz

- Modelo de iluminação: determina como normais, materiais e luzes são combinadas para produzirem as cores finais.
- A equação para modelos de iluminação se baseia em princípios físicos de reflexão da luz.
- Modelos de iluminação = modelos de reflexão.

# Interpolação de Goraud

- Calcula a cor final por vértices, portando, em WebGL, deve ser feita no *vertex shader*.
- A cor de cada ponto (fragmento) dentro de um triângulo é interpolada pelas cores obtidas nos vértices.
- Em WebGL, a interpolação é feita de forma automática por meio de ***varyings***. Os valores calculados no *vertex shader* são interpolados e o resultado é passado, via *varying* para o *fragmente shader*.
- Uma variável do tipo *varying* deve ser declarada com o mesmo nome nos *shaders* de *vértice* e de *fragmento*.

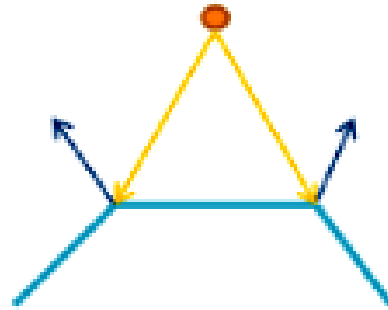
# Interpolação de Phong

- A método de Phong calcula a cor final no *fragment shader*. Ou seja, o cálculo da cor é feita por ponto (ou fragmento) no interior do triângulo.
- Cada fragmento tem sua própria normal, que é resultado da interpolação da normal nos vértices de um triângulo.



## Shading/Interpolation Methods

Gouraud



Vertex Shader

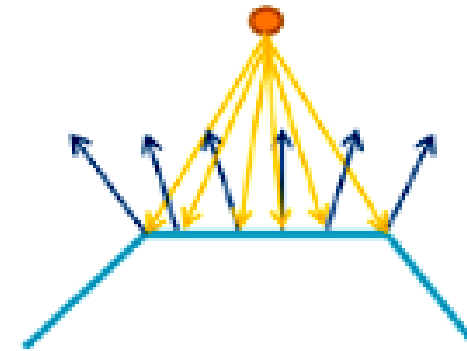
Computes the final color for the **vertex** using the vertex normal. Passes the calculated color to the fragment shader in a varying variable

Varying color

Fragment Shader

Assigns the color for the fragment using the interpolated varying color

Phong



Passes the vertex normal to the fragment shader in a varying variable

Varying normal

Computes the final color for the **fragment** using the respective interpolated normal

The interpolation of varyings is a feature of the pipeline. No programming is required.

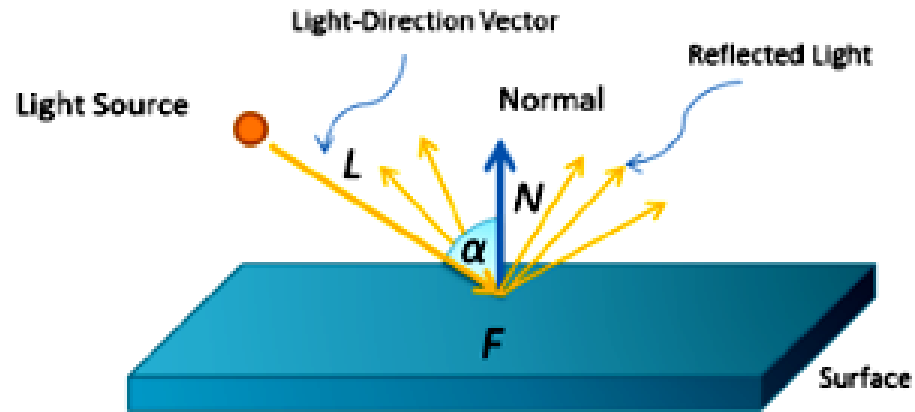
# Modelos de Reflexão da Luz

- Mas como calcular a cor final, independente de ser no vértice ou no fragmento?

# Modelos de Reflexão da Luz

- Modelo de Reflexão Lambertiano: usado como modelo de reflexão difusa.
- Um raio de luz incidente é refletido em muitos ângulos diferentes, em vez de apenas em um único ângulo, como a reflexão especular.
- Baseado na lei de emissão do cosseno ou lei de emissão de Lambert.
- Nomeada após Johann Heinrich Lambert, de seu livro *Photometria*, publicado em 1760.

## Lambertian Reflectance



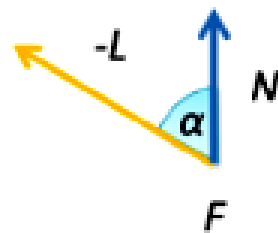
Final Diffuse Color

$$F = C_l C_m (-L \cdot N)$$

Light Diffuse Color

Material Diffuse Color

### Final diffuse color calculation for fragment F



$$-L \cdot N = |-L||N| \cos \alpha$$

If  $L$  and  $N$  are normalized then:

$$-L \cdot N = \cos \alpha$$

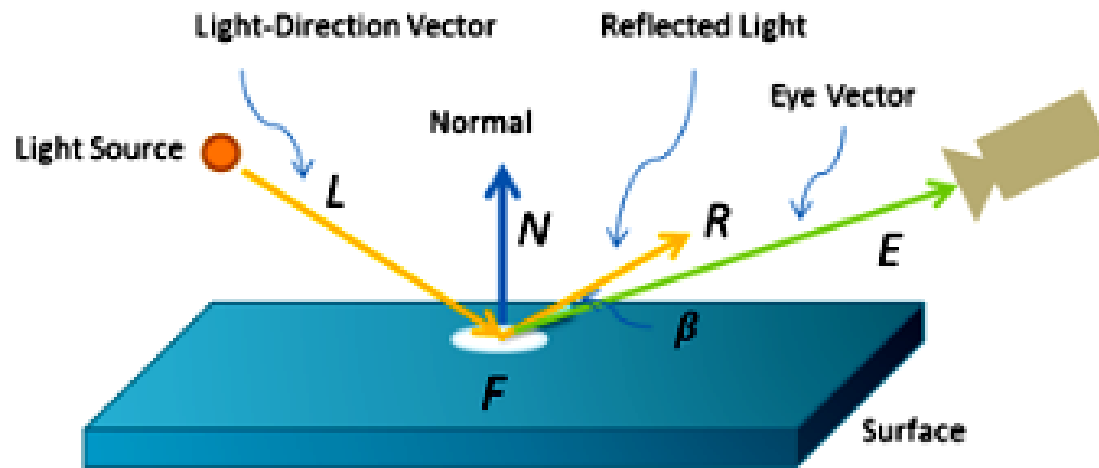
$$F = C_l C_m \cos \alpha$$

A Lambertian surface reflects light in many directions

# Modelos de Reflexão de Phong

- O modelo de reflexão de Phong descreve a forma como uma superfície reflete a luz como a soma de três tipos de componentes: ambiente, difusa e especular.
- Foi desenvolvida por Phong (Bui Tuong Phong), publicado em sua tese de doutoramento em 1973.

## Specular Reflection



Final Specular Color

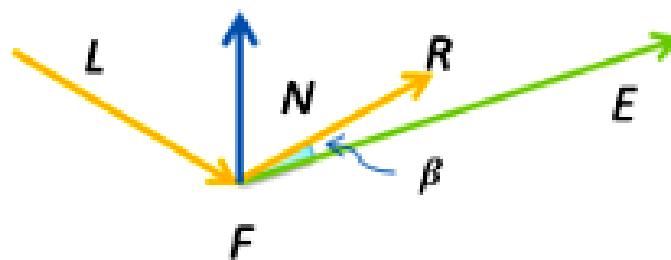
Material Shininess

$$F_s = C_l C_m (R \cdot E)^n$$

Light Specular Color

Material Specular Color

### Final specular color calculation for fragment F



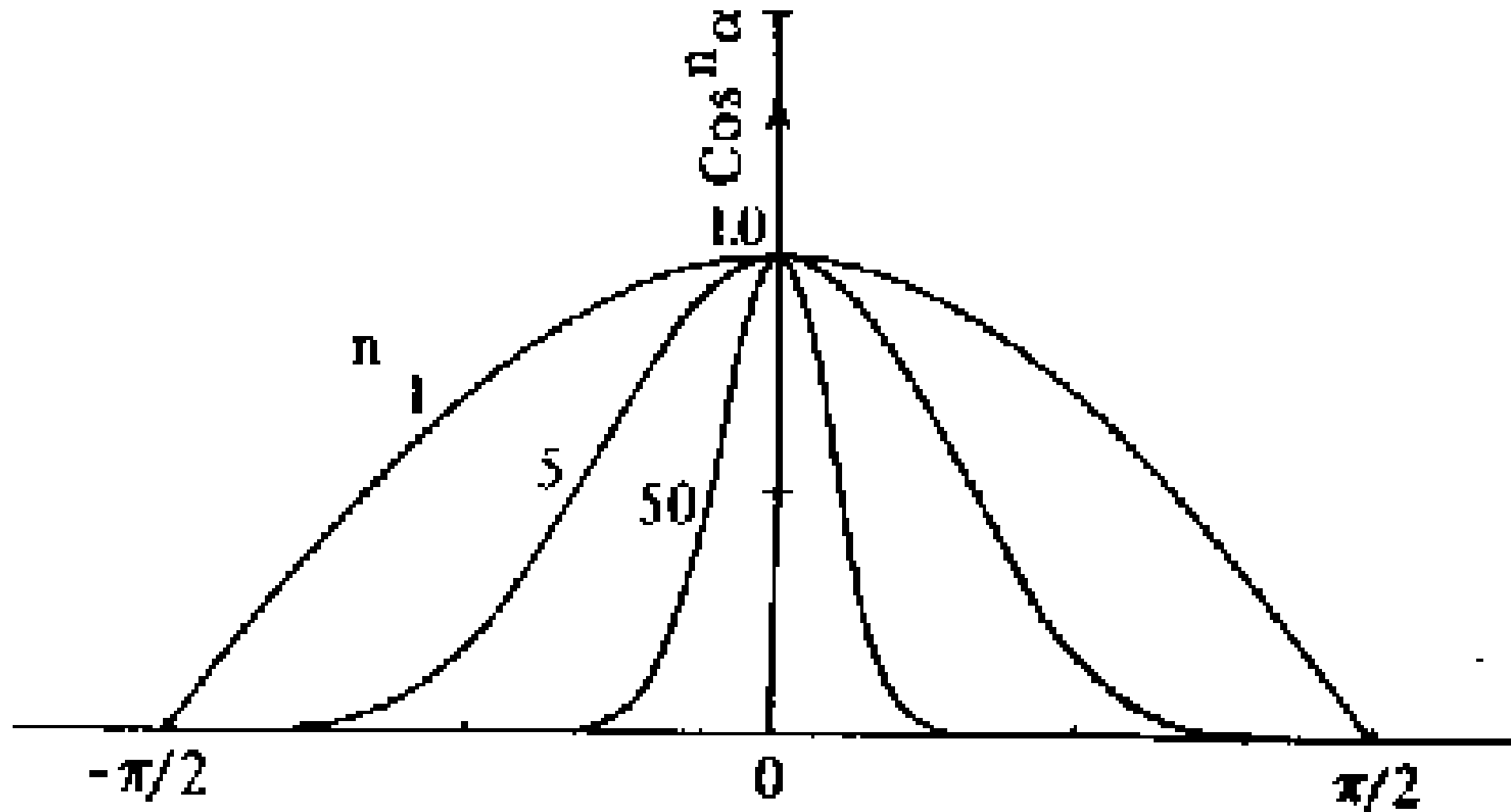
$$R \cdot E = |R||E| \cos \beta$$

If  $R$  and  $E$  are normalized then:

$$R \cdot E = \cos \beta$$

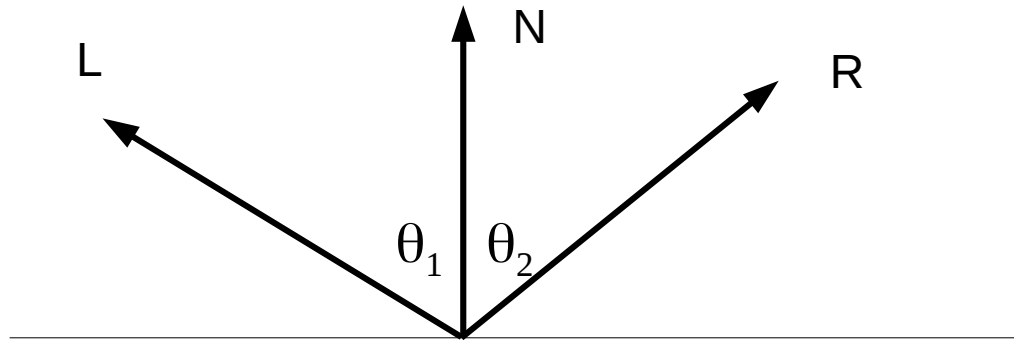
$$F = C_l C_m \cos^n \beta$$

The specular reflection reaches its maximum when  $R$  and  $E$  have the same direction.



# Modelos de Reflexão de Phong

- Calculando o raio de reflexão

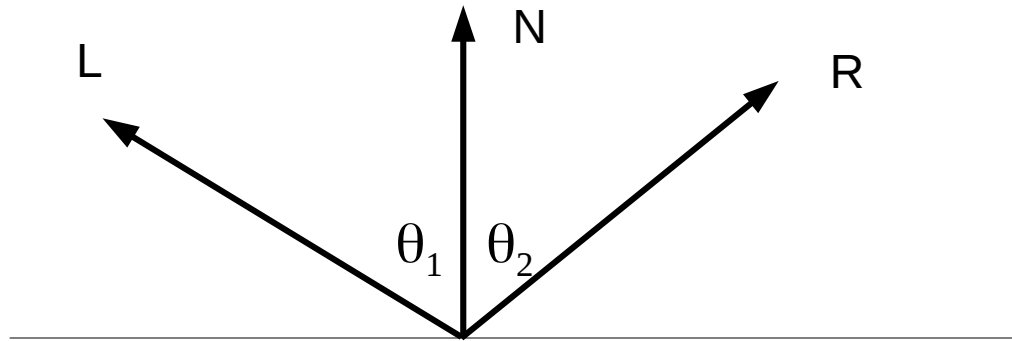


$R = ?$



# Modelos de Reflexão de Phong

- Calculando o raio de reflexão

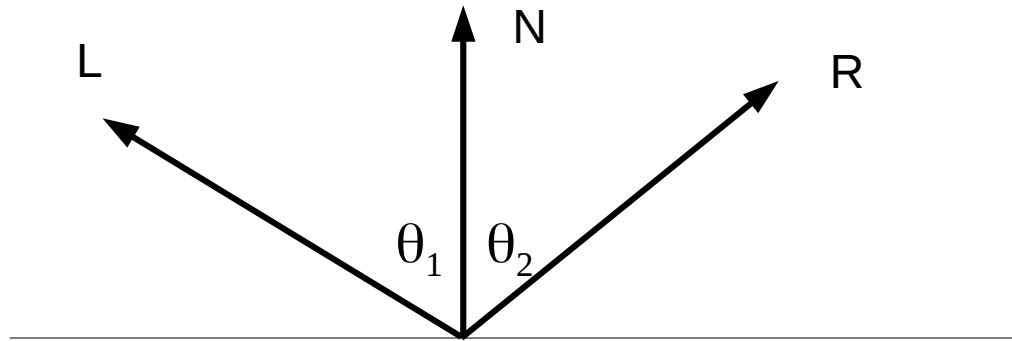


$R = ?$

$$\theta_1 = \theta_2 = \theta$$

# Modelos de Reflexão de Phong

- Calculando o raio de reflexão



$R = ?$

$$\theta_1 = \theta_2 = \theta$$

$$\cos(2\theta) = \frac{L \cdot R}{\|R\| \cdot \|L\|}$$

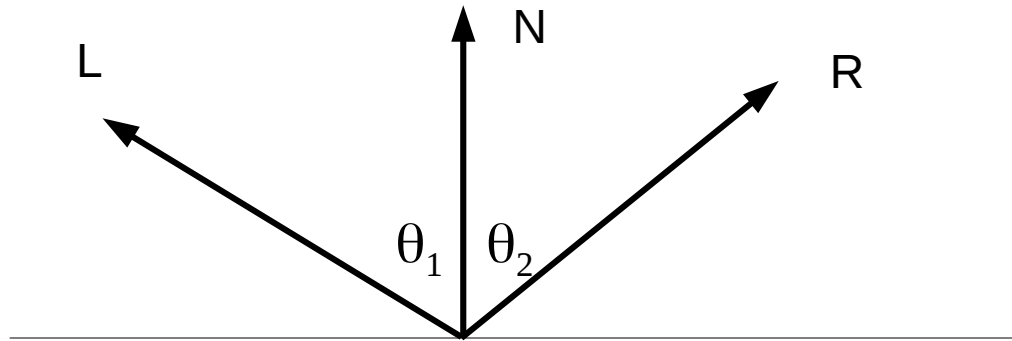
$$\|R\| = \|L\| = 1$$

$$\cos(2\theta) = L \cdot R$$

$$\cos(2\theta) = 2 \cdot \cos(\theta)^2 - 1$$

# Modelos de Reflexão de Phong

- Calculando o raio de reflexão



$R = ?$

$$\theta_1 = \theta_2 = \theta$$

$$\cos(2\theta) = \frac{L \cdot R}{\|R\| \cdot \|L\|}$$

$$\|R\| = \|L\| = 1$$

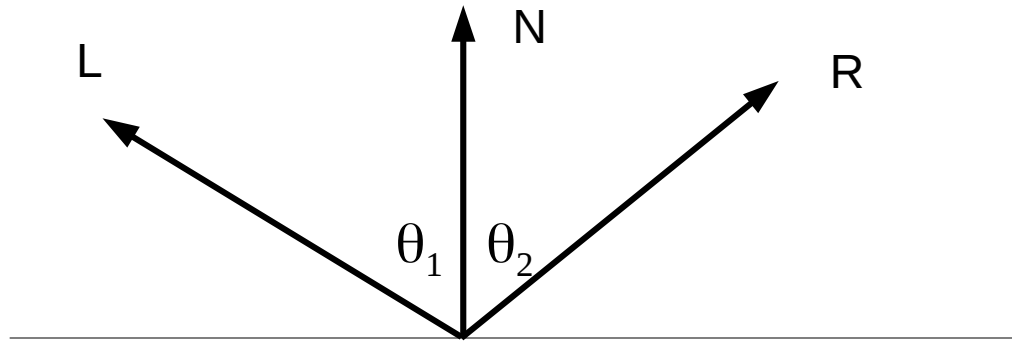
$$\cos(2\theta) = L \cdot R$$

$$\cos(2\theta) = 2 \cdot \cos(\theta)^2 - 1$$

$$L \cdot N = \cos(\theta)$$

# Modelos de Reflexão de Phong

- Calculando o raio de reflexão



$$R = ?$$

$$\theta_1 = \theta_2 = \theta$$

$$\cos(2\theta) = \frac{L \cdot R}{\|R\| \cdot \|L\|}$$

$$\|R\| = \|L\| = 1$$

$$\cos(2\theta) = L \cdot R$$

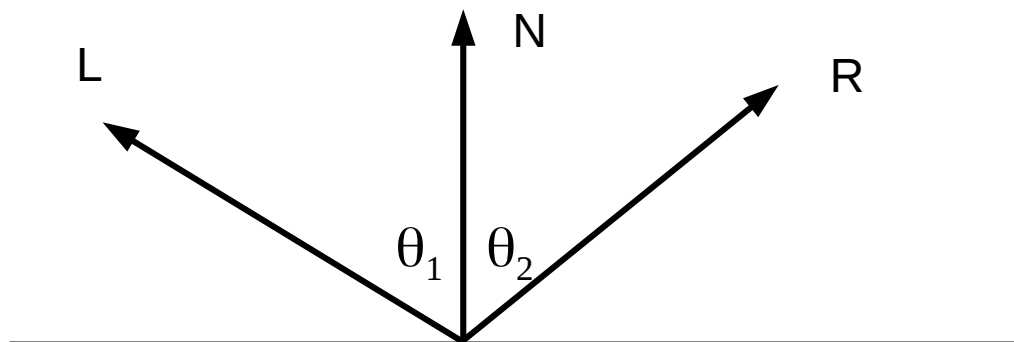
$$\cos(2\theta) = 2 \cdot \cos(\theta)^2 - 1$$

$$L \cdot N = \cos(\theta)$$

$$L \cdot R = 2 \cdot (L \cdot N) - 1$$

# Modelos de Reflexão de Phong

- Calculando o raio de reflexão



$$R = ?$$

$$\theta_1 = \theta_2 = \theta$$

$$\cos(2\theta) = \frac{L \cdot R}{\|R\| \cdot \|L\|}$$

$$\|R\| = \|L\| = 1$$

$$\cos(2\theta) = L \cdot R$$

$$\cos(2\theta) = 2 \cdot \cos(\theta)^2 - 1$$

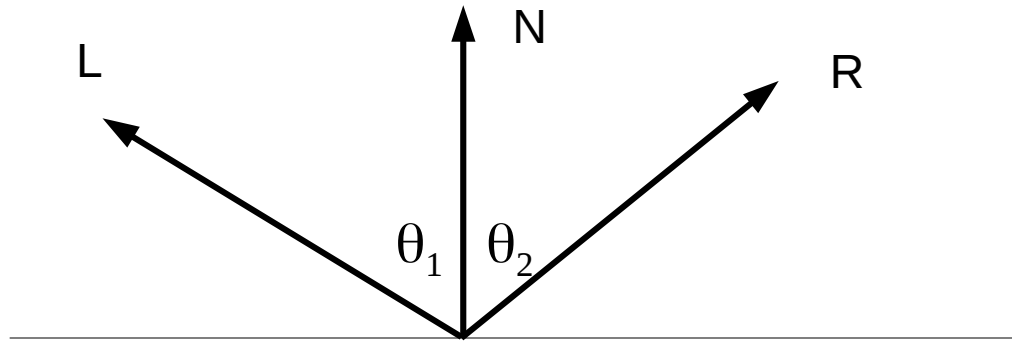
$$L \cdot N = \cos(\theta)$$

$$L \cdot R = 2 \cdot (L \cdot N) - 1$$

$$L \cdot L = 1$$

# Modelos de Reflexão de Phong

- Calculando o raio de reflexão



$$R = ?$$

$$\theta_1 = \theta_2 = \theta$$

$$\cos(2\theta) = \frac{L \cdot R}{\|R\| \cdot \|L\|}$$

$$\|R\| = \|L\| = 1$$

$$\cos(2\theta) = L \cdot R$$

$$\cos(2\theta) = 2 \cdot \cos(\theta)^2 - 1$$

$$L \cdot N = \cos(\theta)$$

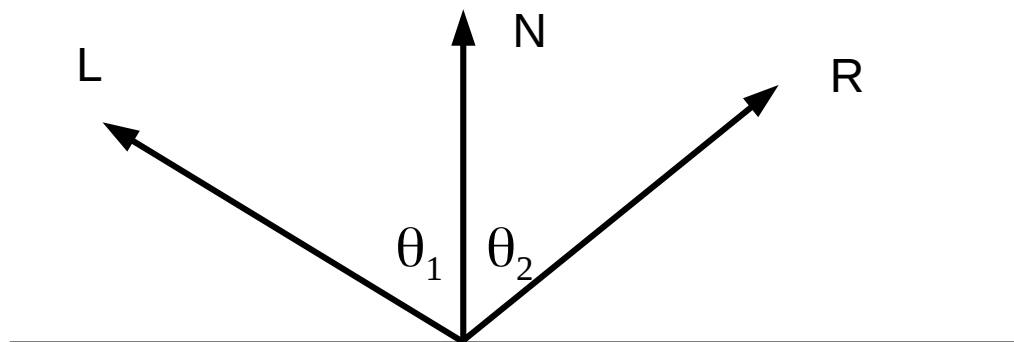
$$L \cdot R = 2 \cdot (L \cdot N) - 1$$

$$L \cdot L = 1$$

$$L \cdot R = 2 \cdot (L \cdot N) - L \cdot L$$

# Modelos de Reflexão de Phong

- Calculando o raio de reflexão



$$R = ?$$

$$\theta_1 = \theta_2 = \theta$$

$$\cos(2\theta) = \frac{L \cdot R}{\|R\| \cdot \|L\|}$$

$$\|R\| = \|L\| = 1$$

$$\cos(2\theta) = L \cdot R$$

$$\cos(2\theta) = 2 \cdot \cos(\theta)^2 - 1$$

$$L \cdot N = \cos(\theta)$$

$$L \cdot R = 2 \cdot (L \cdot N) - 1$$

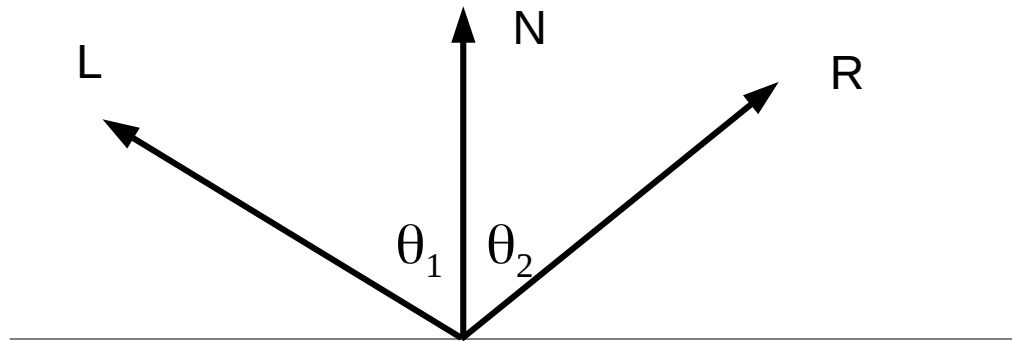
$$L \cdot L = 1$$

$$L \cdot R = 2 \cdot (L \cdot N) - L \cdot L$$

$$L \cdot R = 2 \cdot (L \cdot N) \cdot (L \cdot N) - L \cdot L$$

# Modelos de Reflexão de Phong

- Calculando o raio de reflexão



$$R = ?$$

$$\theta_1 = \theta_2 = \theta$$

$$\cos(2\theta) = \frac{L \cdot R}{\|R\| \cdot \|L\|}$$

$$\|R\| = \|L\| = 1$$

$$\cos(2\theta) = L \cdot R$$

$$\cos(2\theta) = 2 \cdot \cos(\theta)^2 - 1$$

$$L \cdot N = \cos(\theta)$$

$$L \cdot R = 2 \cdot (L \cdot N) - 1$$

$$L \cdot L = 1$$

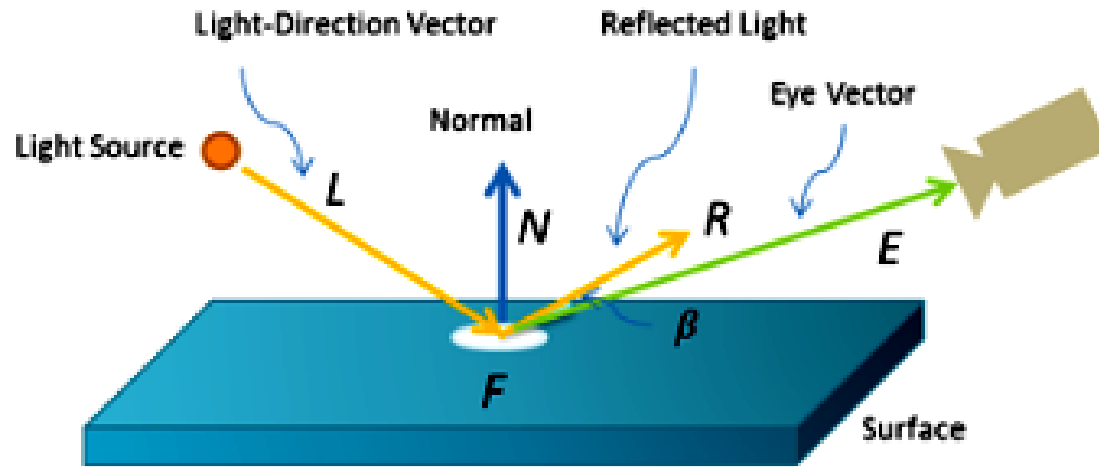
$$L \cdot R = 2 \cdot (L \cdot N) - L \cdot L$$

$$L \cdot R = 2 \cdot (L \cdot N) \cdot (L \cdot N) - L \cdot L$$

$$R = 2 \cdot (L \cdot N) \cdot N - L$$



## Specular Reflection



Final Specular Color

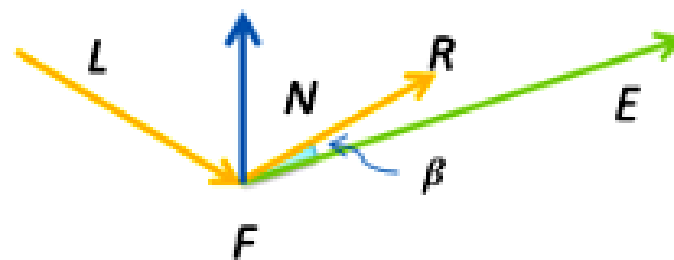
Material Shininess

$$F_s = C_l C_m (R \cdot E)^n$$

Light Specular Color

Material Specular Color

### Final specular color calculation for fragment F



$$R \cdot E = |R||E| \cos \beta$$

If  $R$  and  $E$  are normalized then:

$$R \cdot E = \cos \beta$$

$$F = C_l C_m \cos^n \beta$$

The specular reflection reaches its maximum when  $R$  and  $E$  have the same direction.

# Modelos de Reflexão de Phong

- Quando um objeto sofre transformação, a normal também não se altera?



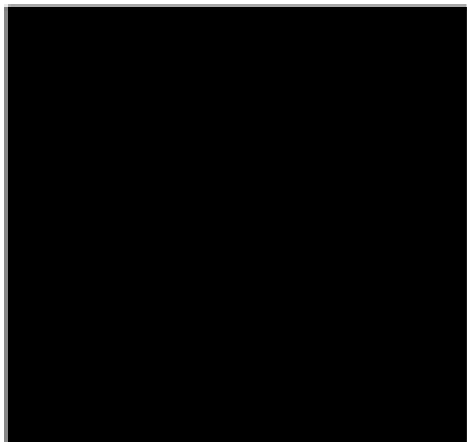
# Modelos de Reflexão de Phong

- Sabemos que depois da transformação, a normal deve continuar perpendicular à superfície no ponto dado
- A matriz de transformação da normal é a inversa da matriz modelview:
  - $N \cdot S = 0$
  - $S' = M \cdot S$
  - $N' \cdot S' = 0$
  - $N' = K \cdot N$
  - $(K N) \cdot (M S) = 0$
  - $(K N)^t (MS) = 0$  (multiplicação matricial)
  - $N^t K^t M S = 0$
  - $N^t (K^t M) S = 0$
  - Como  $N^t S = 0$ ,  $K^t M = I \Rightarrow K = (M^{-1})^T$

# Phong Reflection Model

Reflected color is the result of combining three types of light-object interactions:

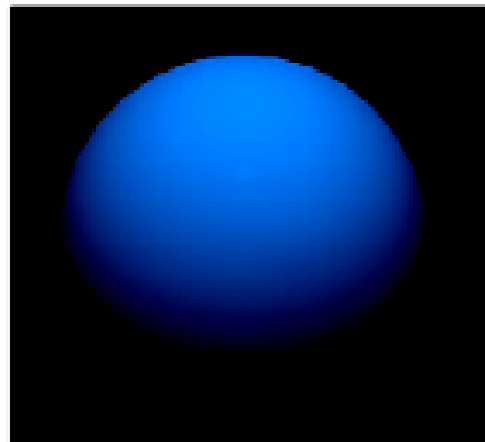
Ambient



Amount of light present *everywhere* in the scene. Independent from any light source

+

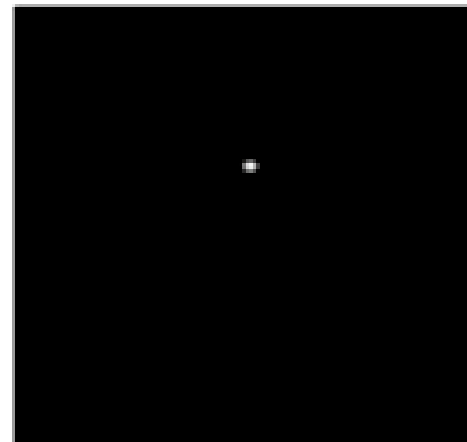
Diffuse



The incident light is reflected in *many directions*. It can be modelled by a Lambertian surface.

+

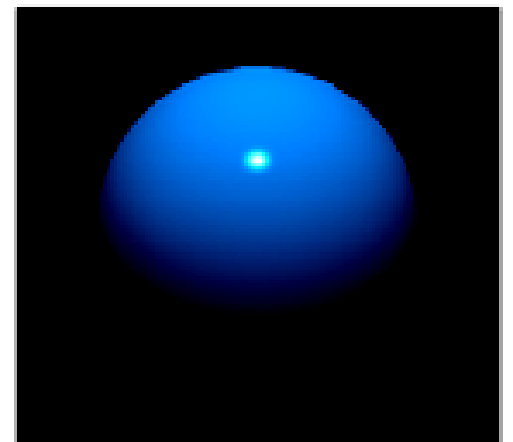
Specular



Mirror-like reflection. The direction of the incoming light and the direction of the reflected outgoing light make *the same angle with respect to the surface normal*.

=

Phong  
Reflection



# Cor final

- Cor ambiente, difusa e especular são somadas para obtenção do resultado final
- $Cor = R + F + F_s$