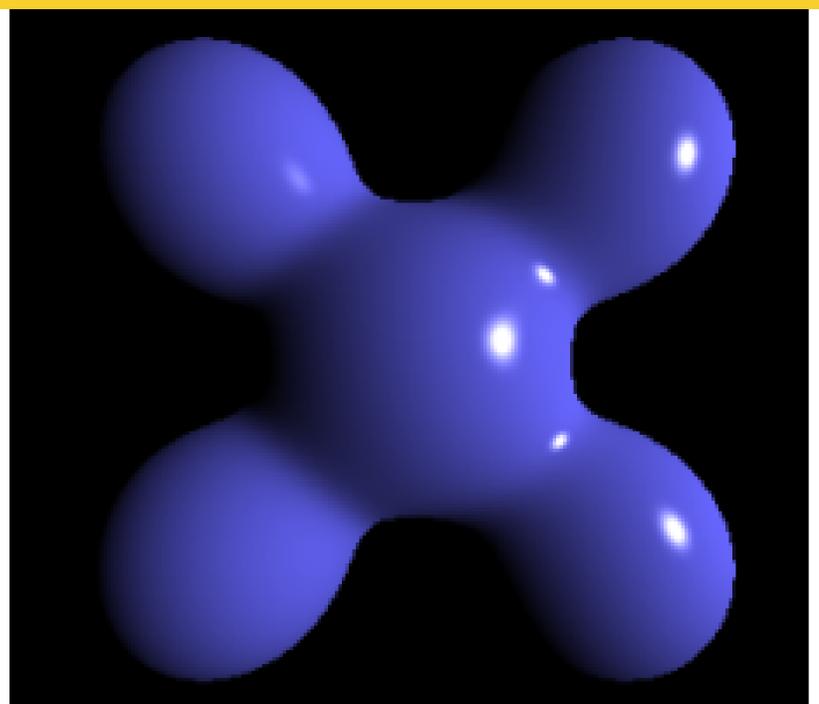
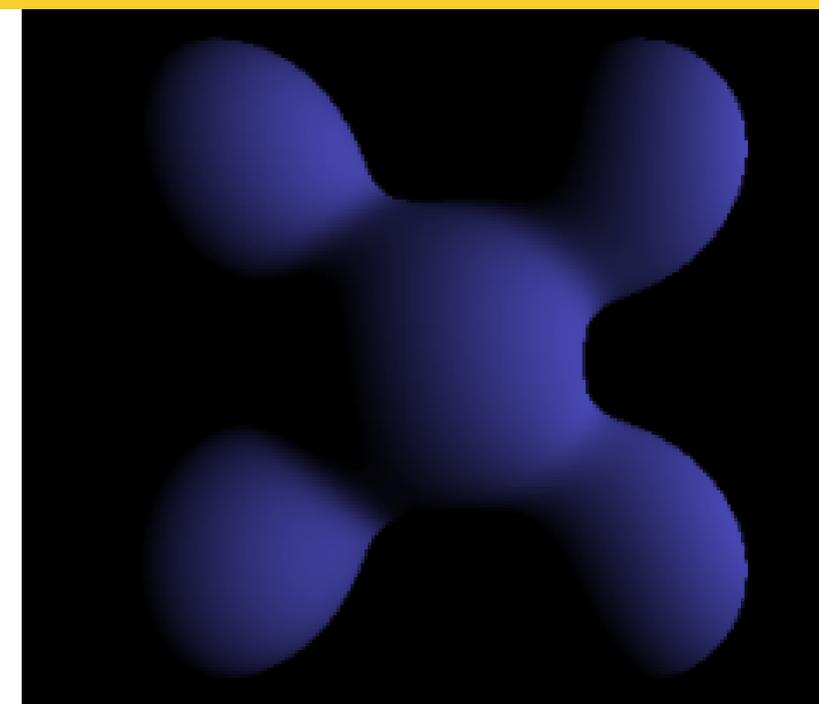




AUXILIAR 6

# ILUMINACIÓN

CC3501





# Agenda

- Iluminación Local vs Global
- Tipos de fuente de luz
- ¿Qué le están haciendo a mis colores?
- Modelo de Iluminación de Phong
- Modelos de Sombreado o Shading
  - Plano
  - Gouraud o Per-Vertex
  - Phong o Per-Pixel
- Ejemplos

# Iluminación Local versus Global

¿De dónde vienen mis fotones?

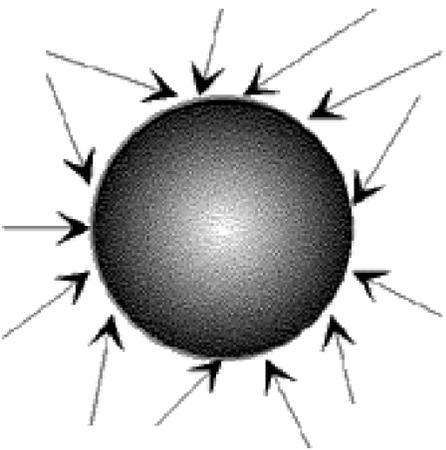
La **iluminación local** o **directa** concierne cómo las superficies y materiales reaccionan a la luz **proveniente de fuentes de luz** como el sol o luces puntuales.

La **iluminación global** o **indirecta** concierne cómo las superficies reciben la **luz reflejada** por otras superficies.

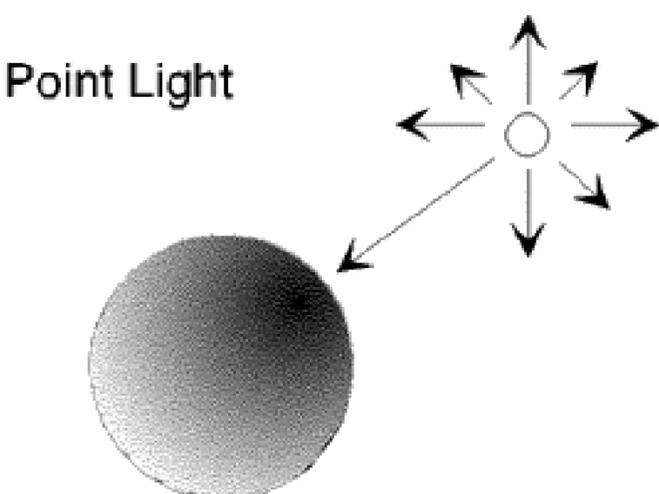
En esta parte del curso nos preocuparemos por la **iluminación directa**.



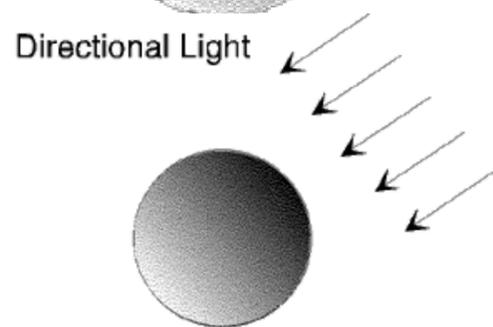
Ambient Light



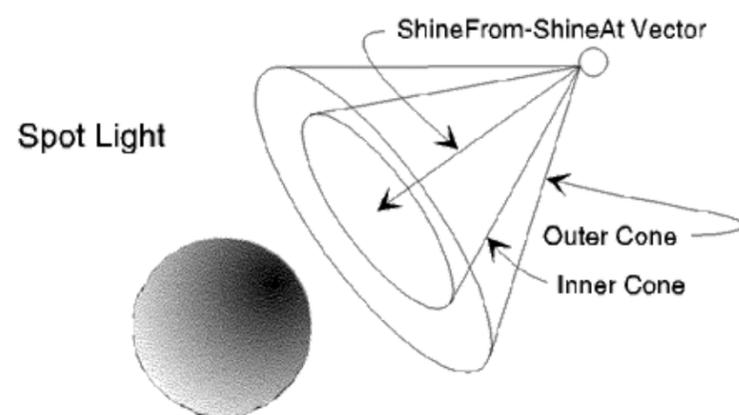
Point Light



Directional Light



Spot Light



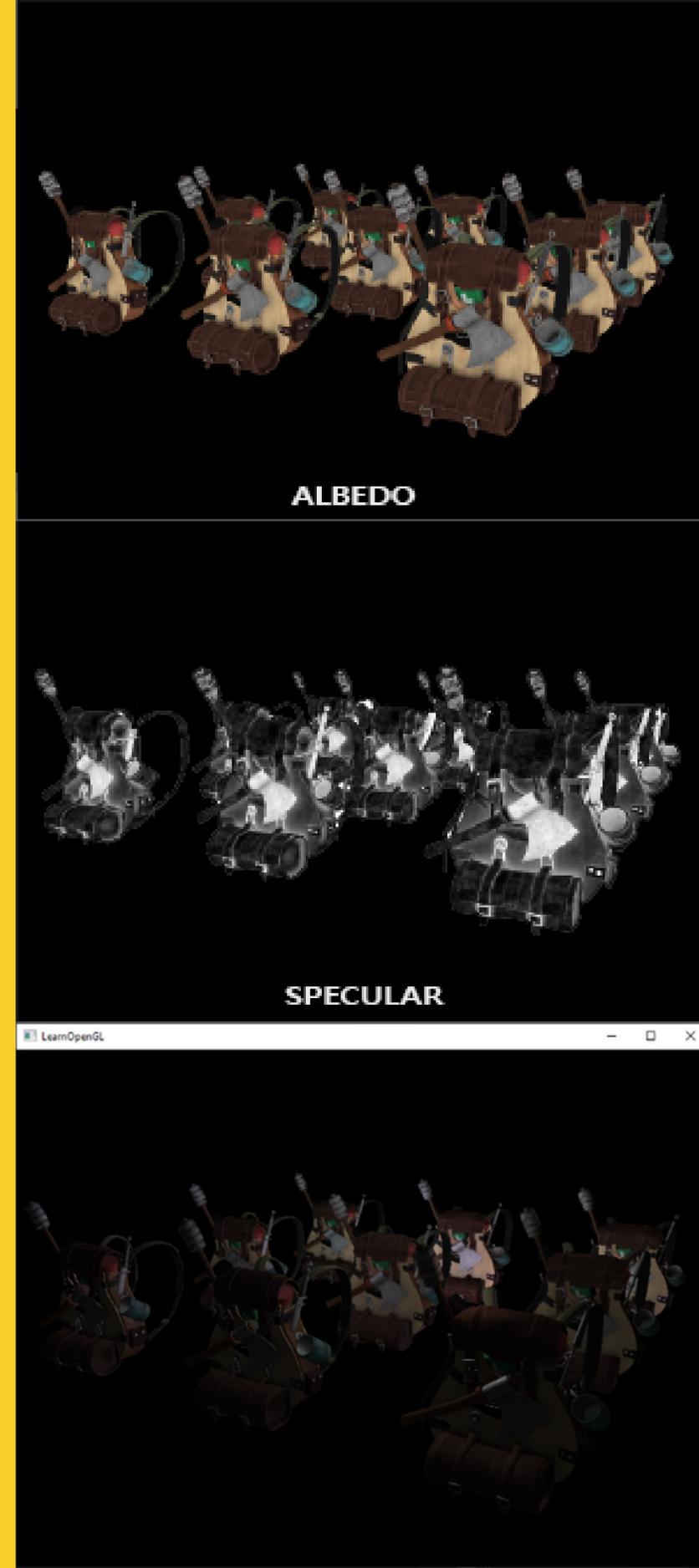
# Tipos de fuentes de luz

- **Luz Ambiental:** Funciona como "luz de relleno". Dicta cómo se van a ver las superficies en oscuridad total.
- **Luz Puntual:** Es una fuente de luz que emite fotones en todas direcciones desde un punto en el espacio. Como una ampolleta.
- **Luz Direccional:** Es una fuente de luz que emite fotones paralelos en una dirección específica. Como el sol o la luna.
- **Luz Spotlight:** Es una fuente de luz que emite fotones en un cono de un ángulo determinado. Como una linterna o un reflector.

# ¿Qué le están haciendo a mis colores?

Mucho de la iluminación se basa en manipular los colores base de las superficies por medio de los shaders.

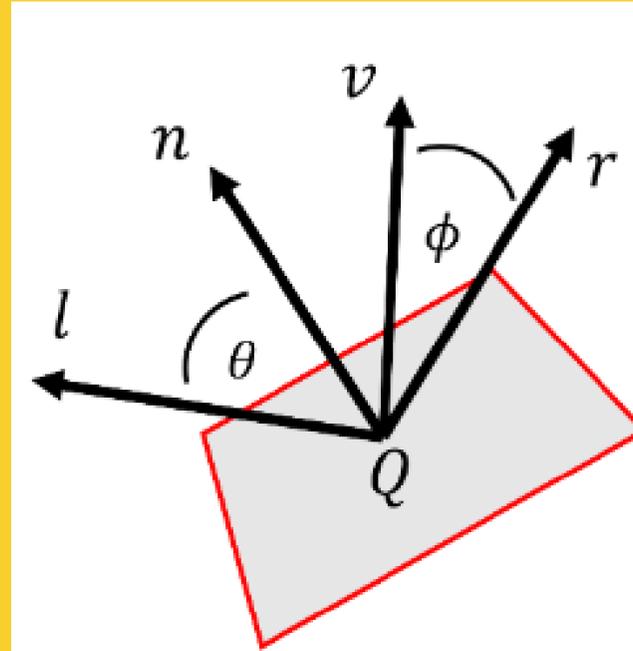
Como cada color tiene 3 componentes, se pueden sumar o multiplicar por separado.



# Modelo de Iluminación de Phong



- $\mathcal{L}_a$ ,  $\mathcal{L}_d$  y  $\mathcal{L}_s$  son las componentes de la **fuentes de luz**.
- $\mathcal{K}_a$ ,  $\mathcal{K}_d$  y  $\mathcal{K}_s$  son los coeficientes de reflexión **del material** para cada componente.
- $d = \|P - Q\|$  es la **distancia** del punto  $Q$  en la superficie hasta  $P$  en la fuente de luz.
- $k_c$ ,  $k_l$  y  $k_q$  son los **coeficientes de atenuación** constante, lineal y cuadrático respectivamente para la **fuentes de luz puntual**.
- $\alpha$  es el coeficiente de brillo.



$$\mathcal{I} = \mathcal{K}_a \mathcal{L}_a + \frac{1}{k_c + k_l d + k_q d^2} (\mathcal{K}_d \mathcal{L}_d (l \cdot n) + \mathcal{K}_s \mathcal{L}_s (v \cdot r)^\alpha)$$

$$d = \|Q - P\|$$

## Importante

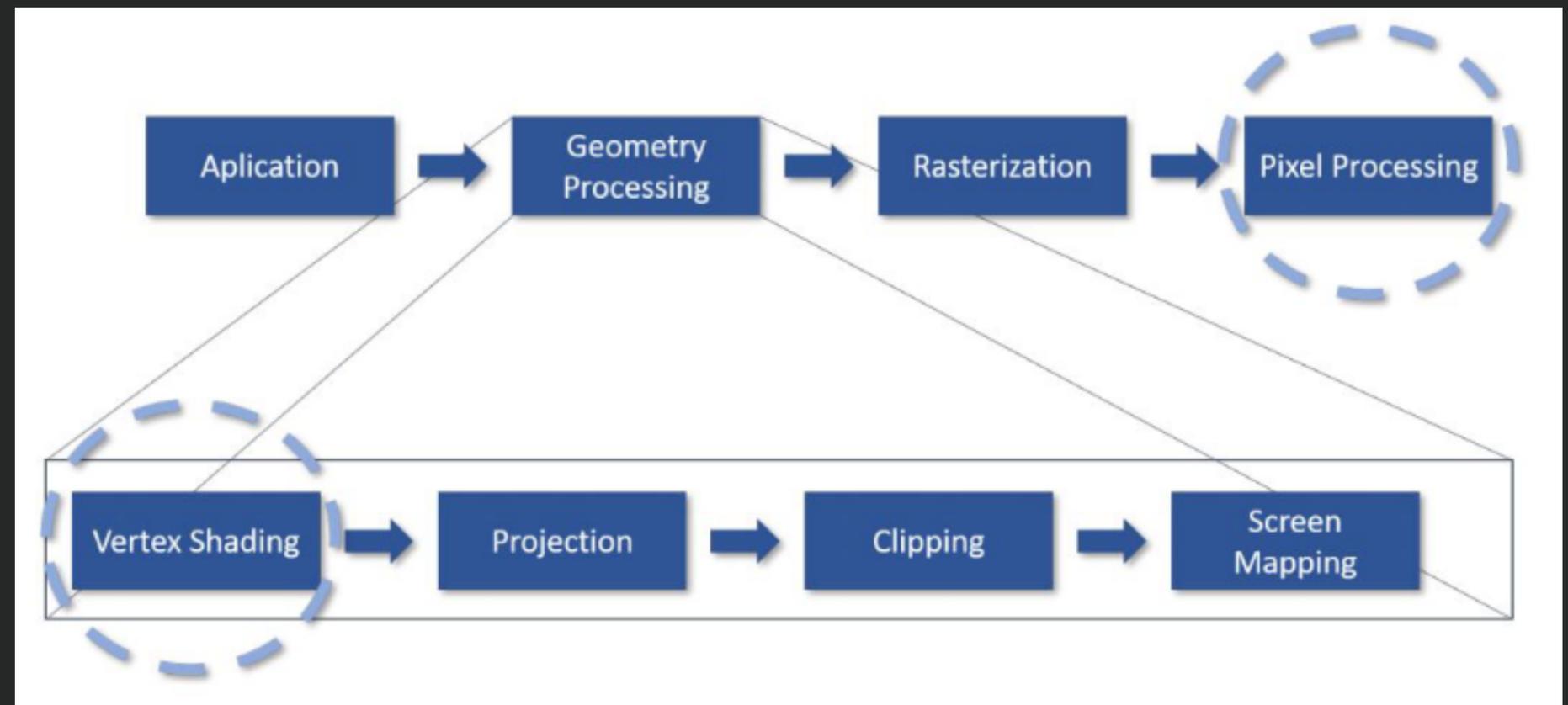
- $\mathcal{L}_a$ ,  $\mathcal{L}_d$ ,  $\mathcal{L}_s$ ,  $\mathcal{K}_a$ ,  $\mathcal{K}_d$  y  $\mathcal{K}_s$  se aplican para cada independientemente para R, G y B
- Luego,  $\mathcal{L}_a$ ,  $\mathcal{L}_d$ ,  $\mathcal{L}_s$ ,  $\mathcal{K}_a$ ,  $\mathcal{K}_d$  y  $\mathcal{K}_s$  son representadas compactamente como colores



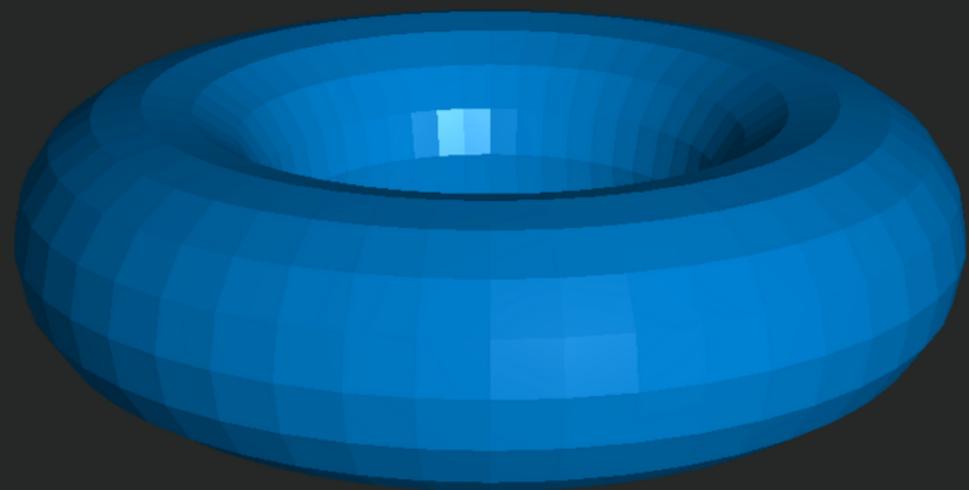
# Modelos de Sombreado

Muy bonitas las matemáticas y todo pero, ¿cómo se usan?

La iluminación se puede calcular en dos puntos particulares del pipeline de renderizado:



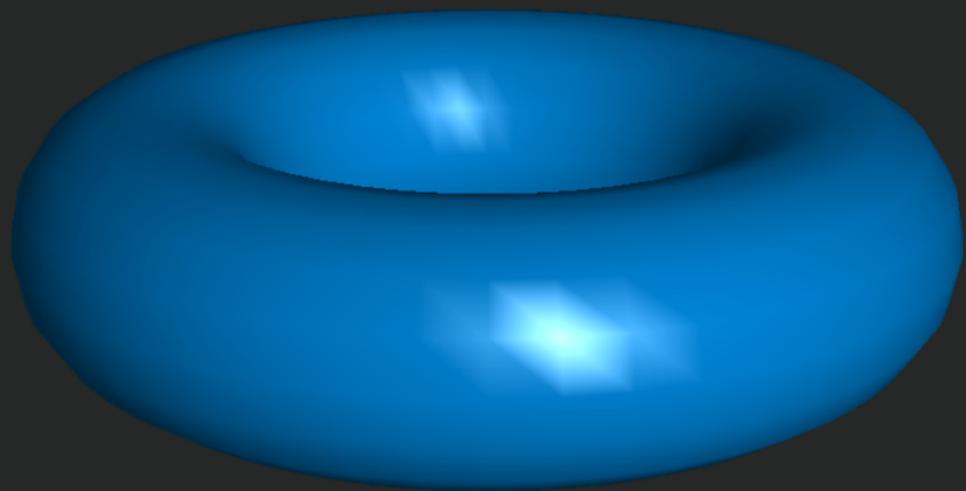
# Flat Shading



Para **cada vértice** (o cara), **calcula un único color** y lo **asigna a la cara entera** sin interpolar, de manera similar al método del vecino más cercano.

Este procedimiento tiene lugar en el **Vertex Shader**.

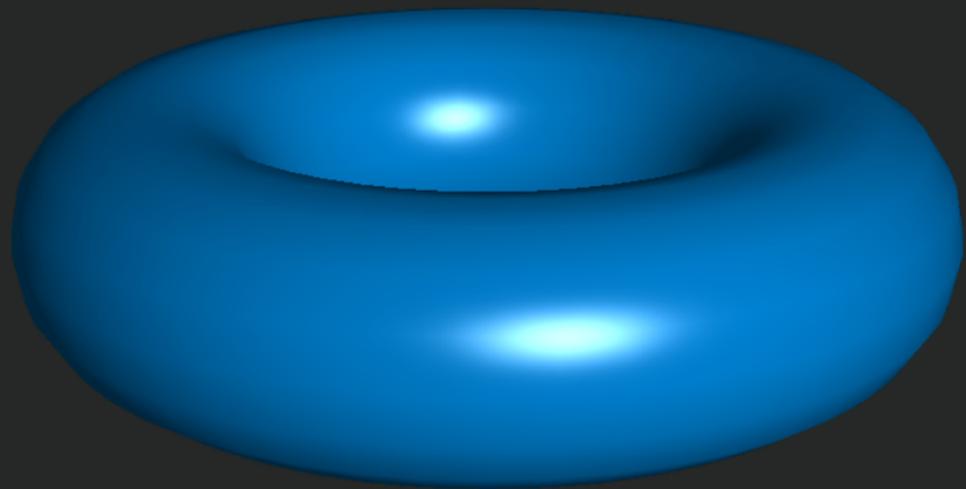
# Gouraud o Per-Vertex Shading



Para **cada vértice**, **calcula un único color** y luego **interpola linealmente** entre los valores de cada vértice en el triángulo para entregar el color de cada píxel asociado a él.

Este procedimiento también tiene lugar en el **Vertex Shader**.

# Phong o Per-Pixel Shading



Tip para la vida™:

!!!!NO CONFUNDIR CON EL MODELO DE  
ILUMINACIÓN DE PHONG!!!!!!!

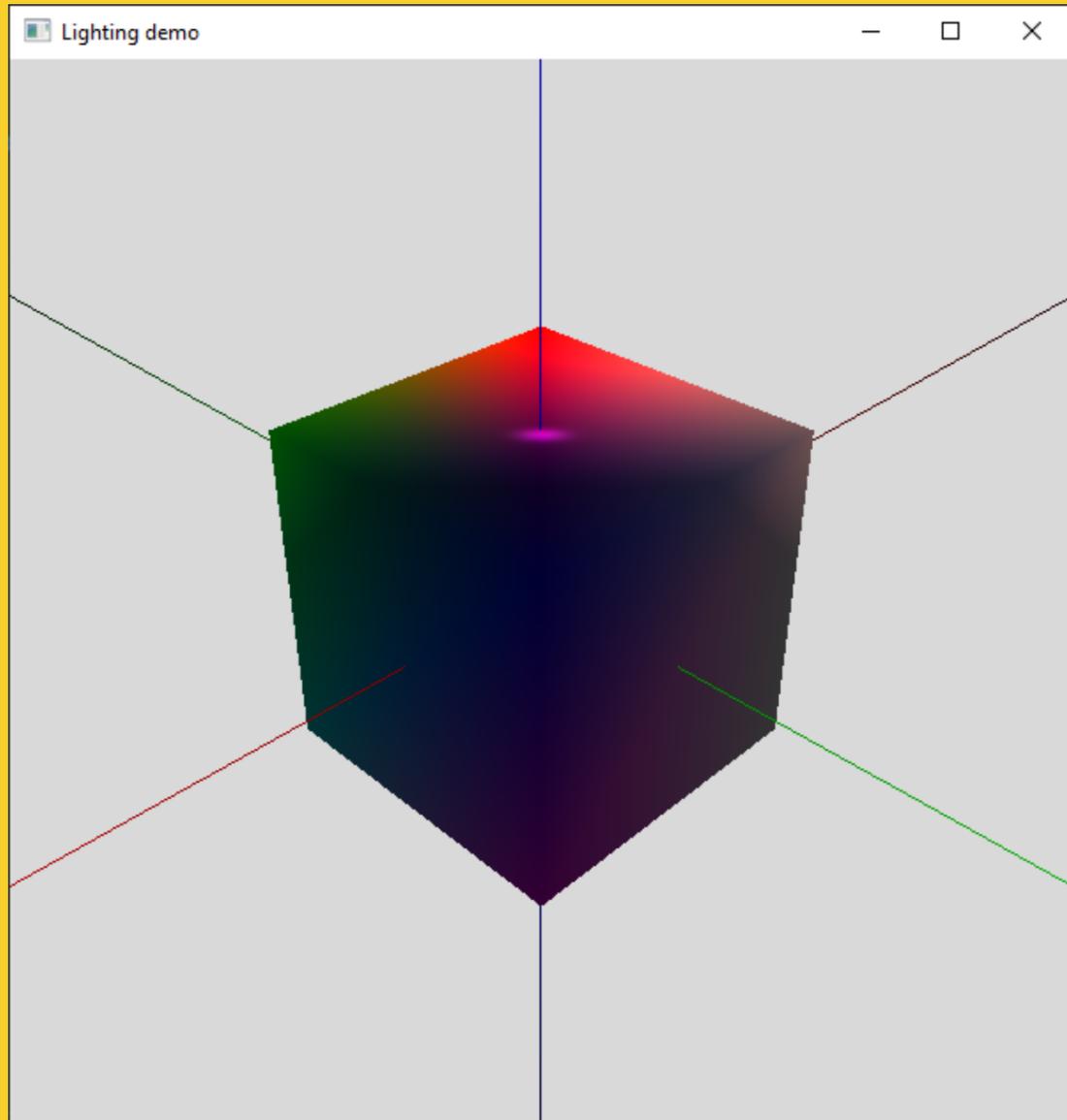
Para **cada vértice**, **interpola el vector normal** a partir de los triángulos que lo comparten.

Luego, para **cada píxel**, **se utiliza este vector normal** suavemente interpolado **para calcular la iluminación** final, que resulta en superficies más suaves.

Este procedimiento tiene una parte en el **Vertex Shader** y otra en el **Fragment Shader**.

# Ejemplos

`examples/ex_lighting.py`



`examples/ex_lighting_texture2.py`

