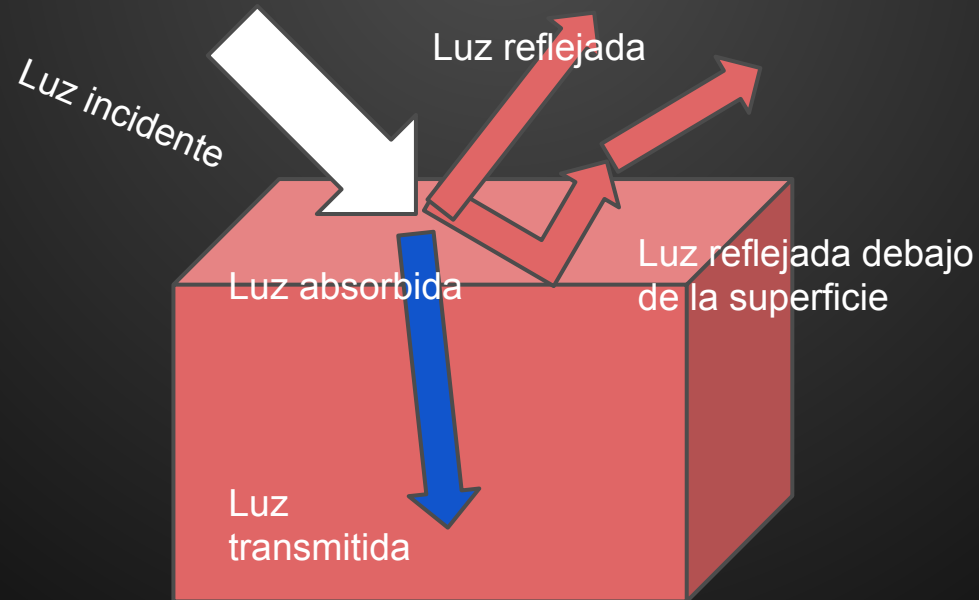


# Iluminación

[mleguen@ucsp.edu.pe](mailto:mleguen@ucsp.edu.pe)

# Iluminación y materiales

- Reflejo a la superficie
- Transmisión, refracción
- Absorción
- Reflexión debajo la superficie

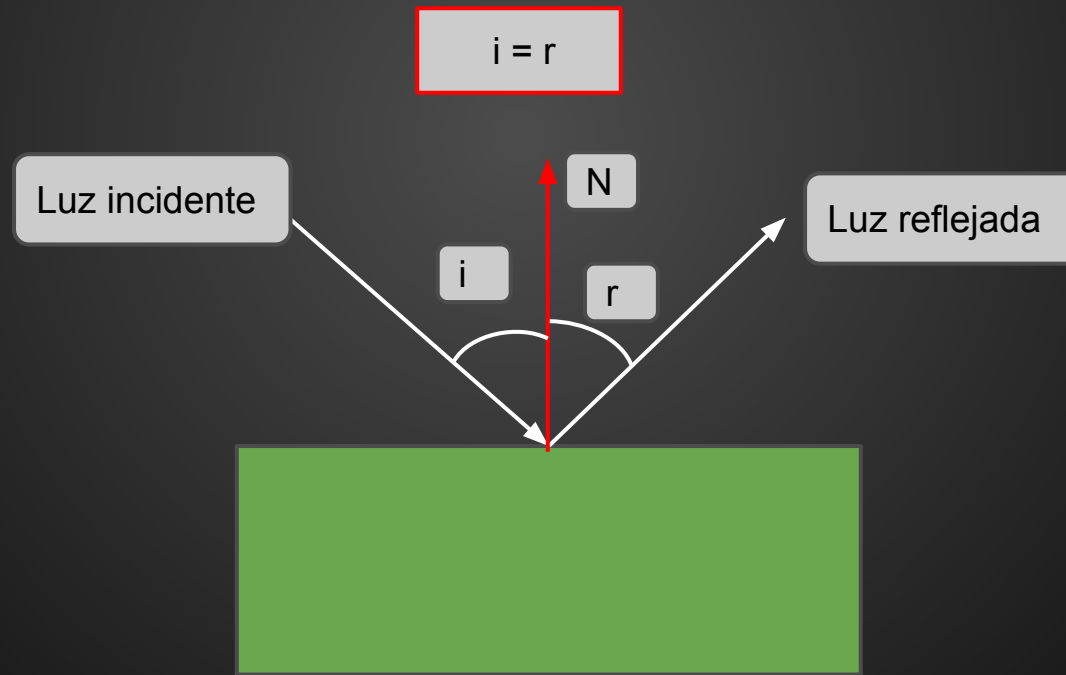


# Reflexión de la luz

- La luz se propaga en línea recta y todo objeto opaco es un obstáculo. Como el sonido la luz puede rebotar sobre la superficie. Este fenómeno se llama : **reflexión.**
- La mayor parte de la luz que llega hasta nuestros ojos es reflejada

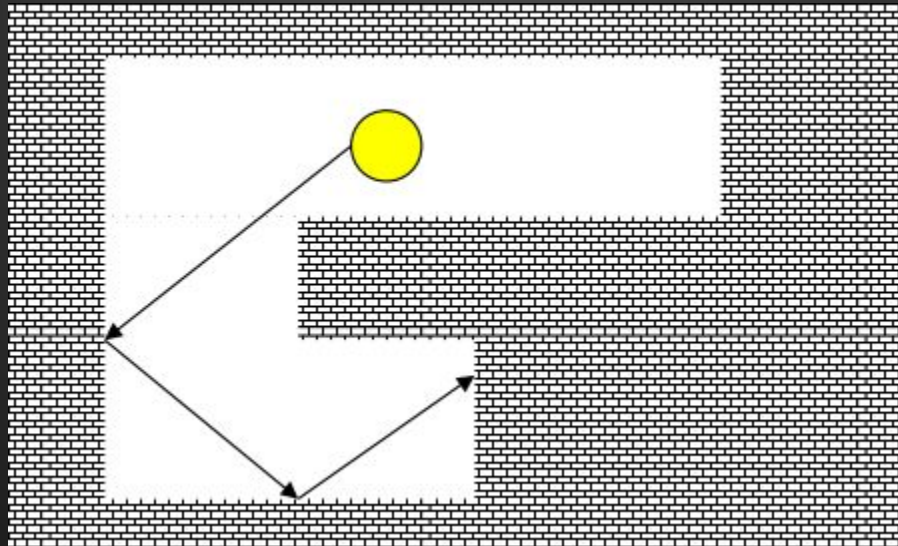
# Reflexión de la luz

- Según la ley de Descarte - Snell, la luz incidente en un punto de una superficie con un ángulo relativo la normal del objeto es reflejada con el mismo ángulo



# Reflexión de la luz

La luz puede ser reflejada sucesivamente por varias superficies y de tal modo llegar hasta zonas que no son directamente iluminadas por la luz



# Reflexión de la luz

En los algoritmos de síntesis de imagen si los rebotes de la luz no son tomados en cuenta las zonas no directamente iluminadas por luz aparecerán en negro.

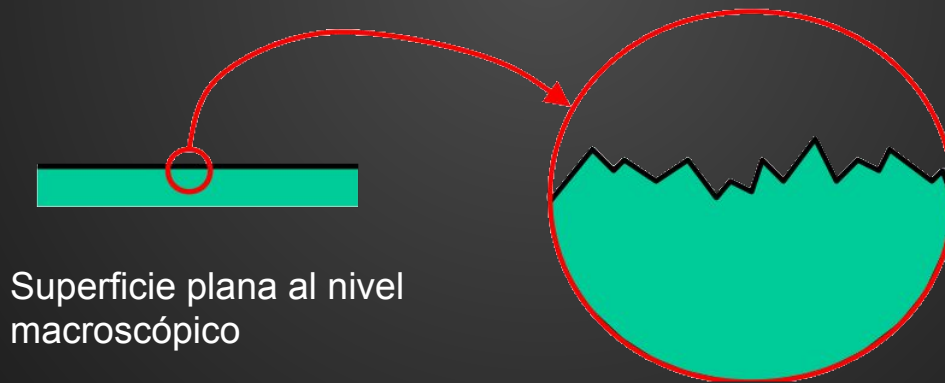


En síntesis de imágenes, este mecanismo puede ser simulado con diferentes métodos como la radiosidad, el path tracing etc.

Con OpenGL este sistema es simulado con un componente “ambiente” de la luz y de los materiales

# Reflexión de la luz en la realidad

- La luz incidente solo es reflejada como lo acabamos de ver en caso de espejos perfectos.
- En la realidad, las superficies tienen defectos a un nivel microscópico.

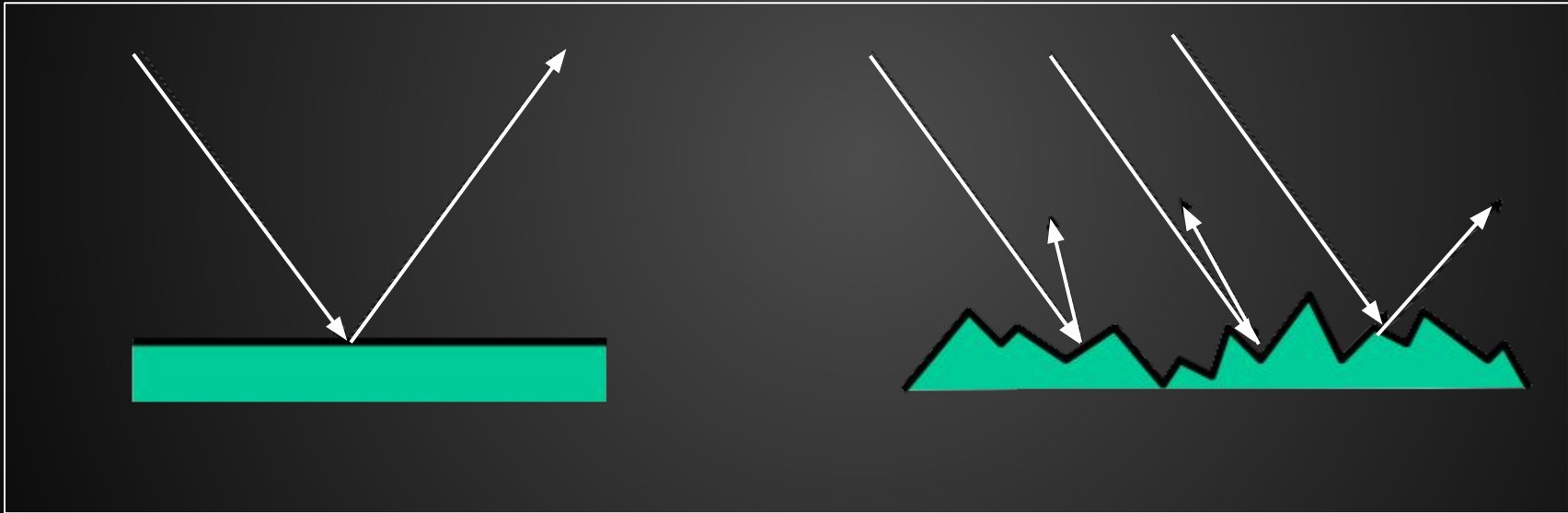


Superficie plana al nivel  
macroscópico

Superficie vista a una escala microscópica

# Reflexión de la luz en la realidad

- La luz no es reflejada en una sola dirección sino en un conjunto de direcciones.

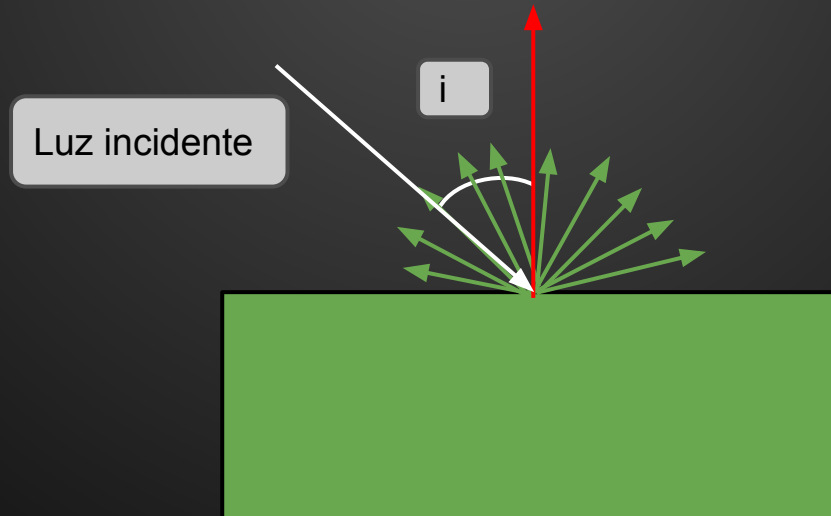


- Estas direcciones son divididas según una componente especular y una componente difusa.



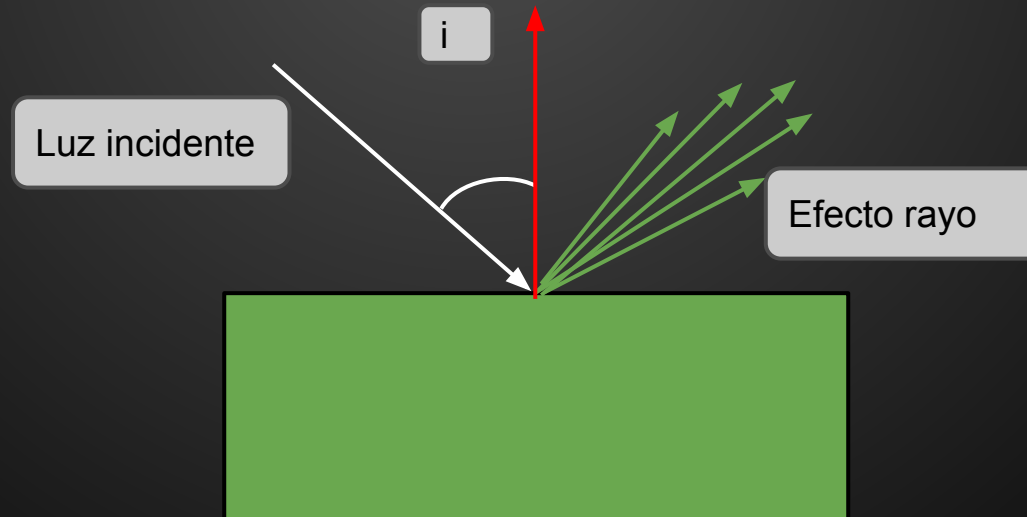
# Reflexión difusa

- La luz incidente es reflejada en todas las direcciones
- La intensidad reflejada depende solamente del ángulo  $i$  entre la dirección de la luz y la normal
- La intensidad difusa es maximal cuando el ángulo  $i = 0$  (la fuente de luz es perpendicular al objeto)
- Es nula por una iluminación con un ángulo  $i = 90^\circ$



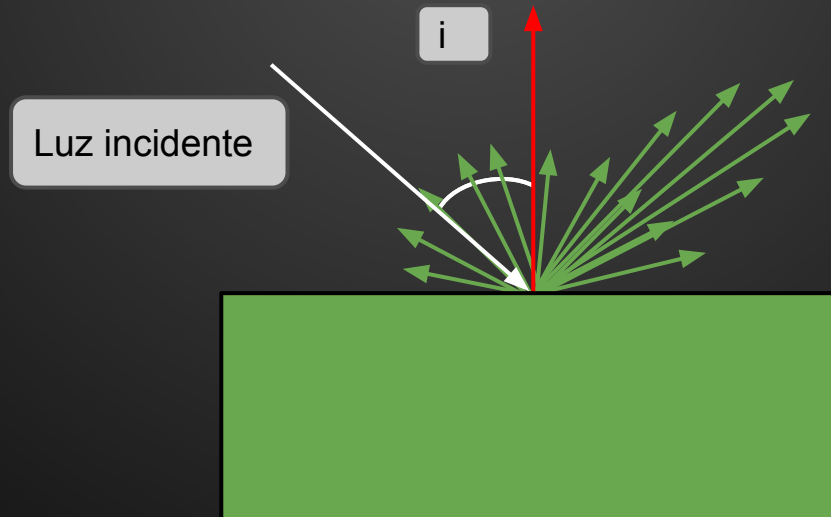
# Reflexión especular

- La luz es principalmente reflejada en la dirección de la reflexión perfecta
- La intensidad de la luz reflejada disminuye cuando se aleja de esta dirección perfecta



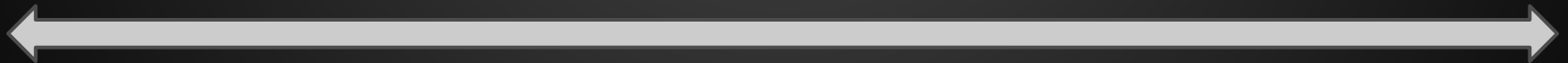
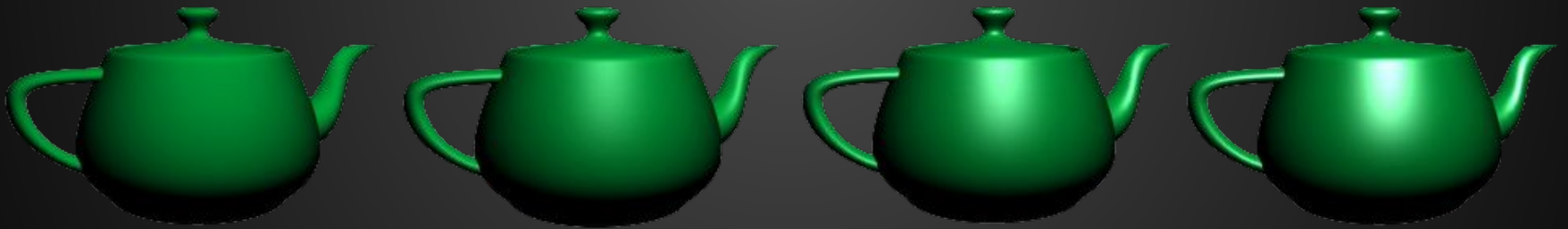
# Reflexión final

- En la realidad, la luz reflejada está compuesta por la suma de la reflexión difusa y especular
- Esta función final que da las direcciones de reflexión de la luz relativa a la luz incidente se llama BRDF (Bidirectional reflectance distribution function)



# Reflexión

La proporción de reflexión difusa y especular depende del material. Materiales de reflexión difusa ( papel, tela, ...) o de reflexión especular (metal, vidrio, etc).



Diffuso

especular

Mate

Brillante

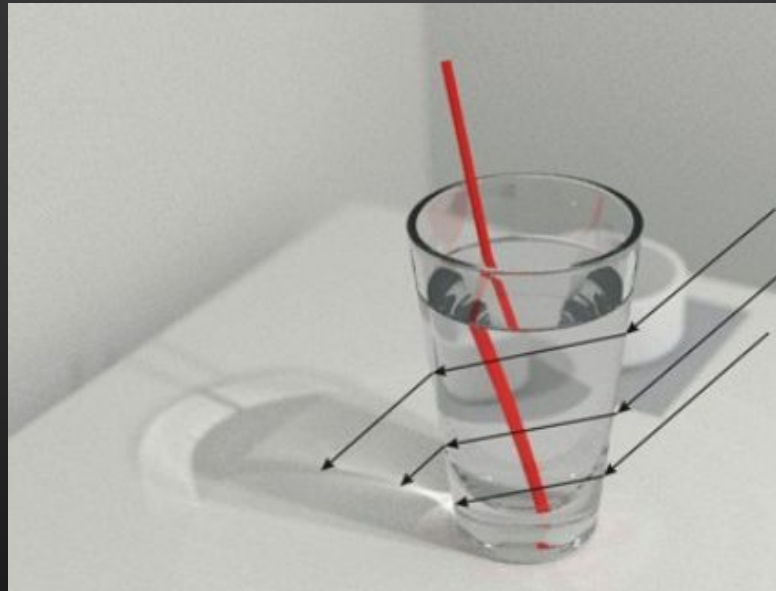
# Reflexión

- BRDF
  - Phong
  - Lambert
  - Blinn Ward
  - Cook-Torrance



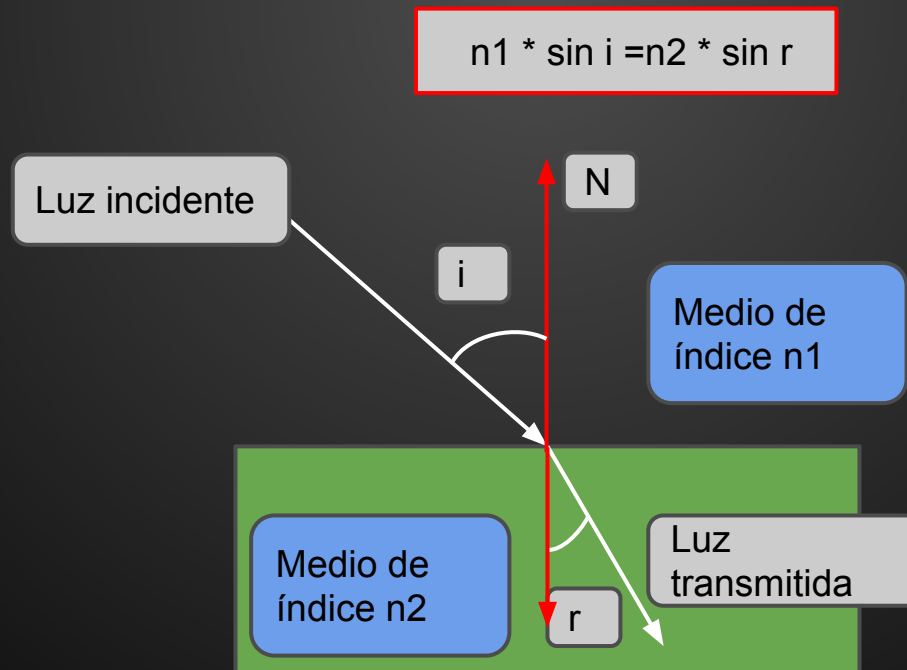
# Refracción de la luz

- En un medio uniforme la luz se transmite a una velocidad constante pero cuando pasa (transmisión) de un medio transparente a otro medio transparente(aire -> agua ) su velocidad cambia
- Este cambio de velocidad provoca un cambio de dirección en la luz.
- Hablamos de refracción cuando la luz es transmitida



# Refracción de la luz

El ángulo de refracción es relativo al ángulo de incidencia. Descartes - Snell :



# Refracción de la luz

- Índice de refracción
  - Este índice corresponde a la relación de la velocidad de la luz en el vacío entre la velocidad de la luz en el medio.
  - Más el índice de refracción es alto más lento es la velocidad de luz
- Ejemplo :
  - Velocidad de la luz en el vacío:  $V_1 = 300\,000 \text{ km/s}$
  - Velocidad de la luz en el diamante:  $V_2 = 120\,000 \text{ km/s}$
  - Índice de refracción del diamante:  $V_1/V_2 = 2.42$



# Algunos índices de refracción

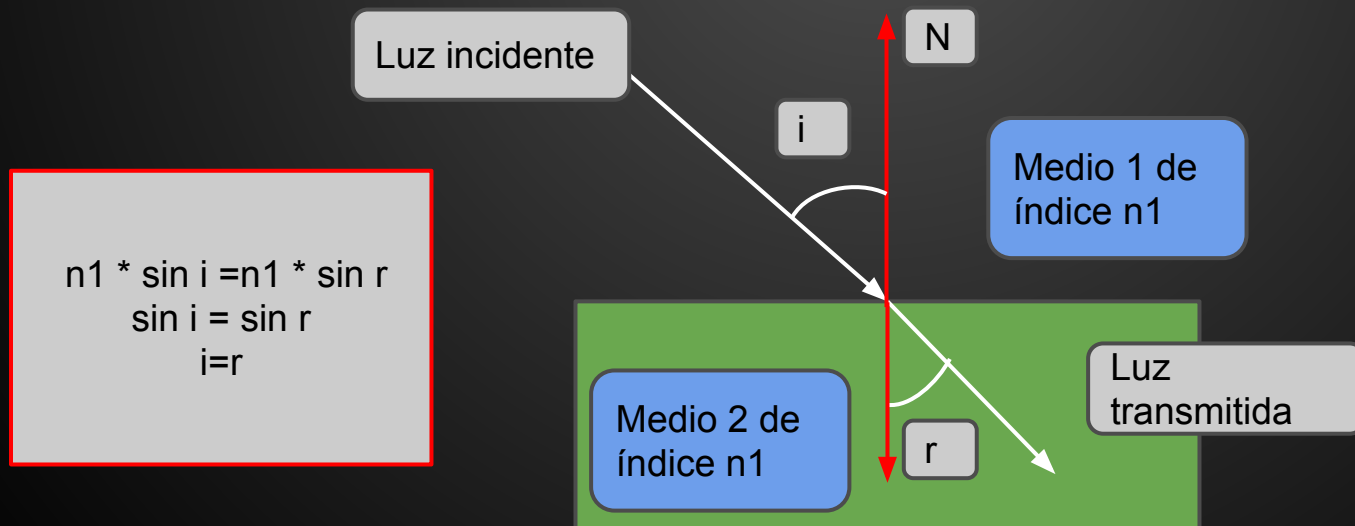
Medio	Indice
Vacío	1
Aire	1,00029
Agua	1.33
Vidrio Plástico	1,51
Diamante	2.42

# Capacidad de refracción

- La capacidad de refracción de un medio transparente es obtenida por su índice de refracción. Un medio con un índice  $n_1$  tiene mayor capacidad de refracción que un medio de índice  $n_2$  si  $n_1 > n_2$
- Cuando la luz es transmitida existen 3 casos posibles :
  - La luz se encuentra con un medio de igual índice
  - La luz se encuentra con un medio de mayor índice
  - La luz se encuentra con un medio de menor índice

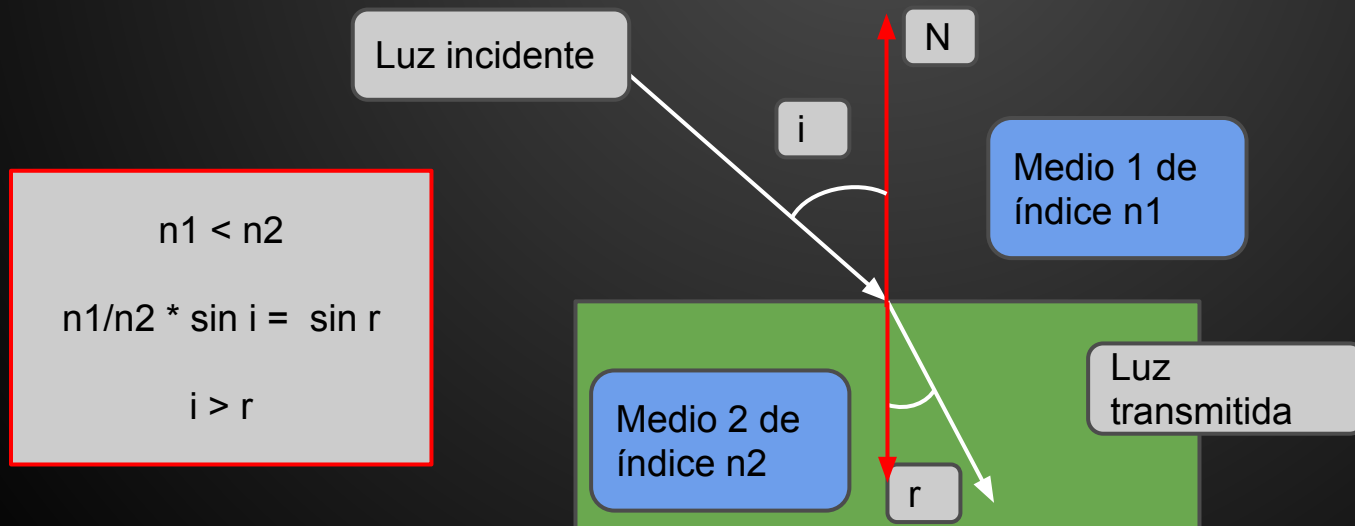
# Refracción - medio igual índice

- La luz se encuentra con un medio de igual índice
- Una parte de la luz es transmitida al medio 2 con cualquier ángulo  $i$  incidente



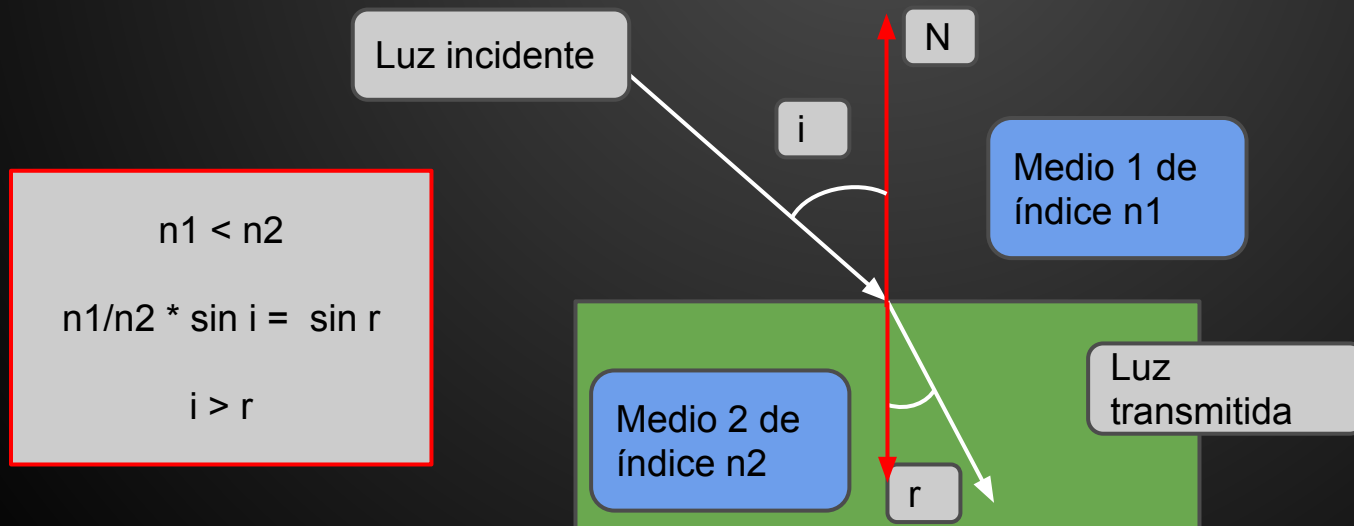
# Refracción - medio mayor índice

- La luz se encuentra con un medio de mayor índice
- Una parte de la luz es transmitida al medio 2 con cualquier ángulo  $i$  incidente



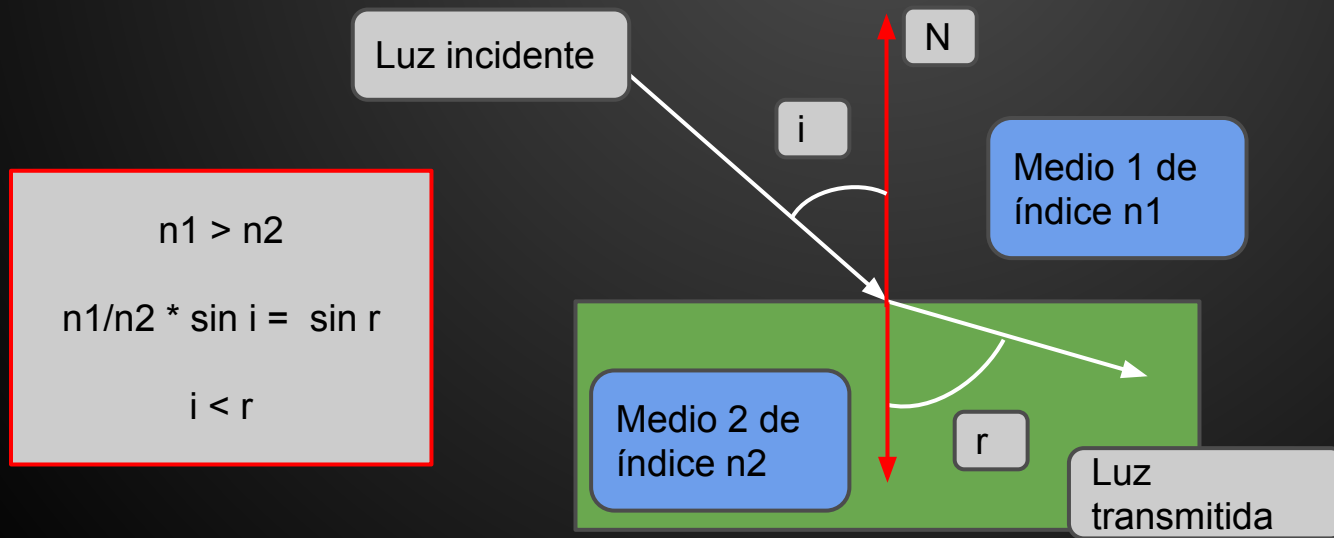
# Refracción - medio mayor índice

- Ejemplo : pasar del aire al agua
  - Un sorbete parecerá roto
  - Los objetos en agua parecen más cerca a un observador fuera del agua



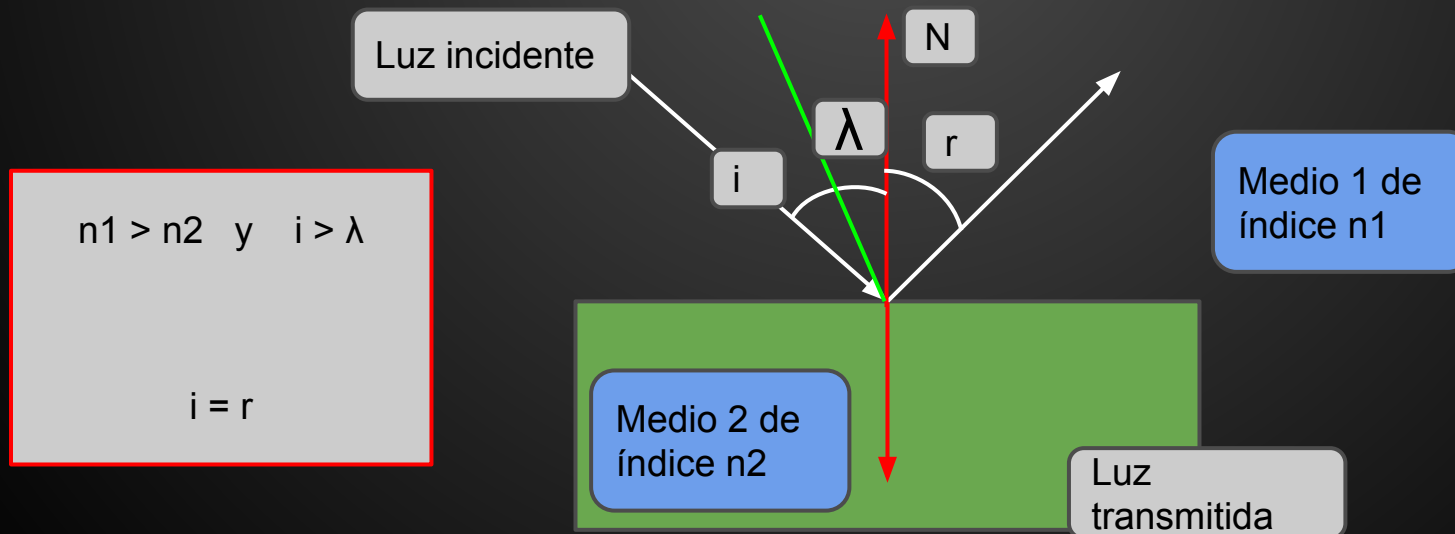
# Refracción - medio mayor índice

- La luz se encuentra con un medio de menor índice
- la luz refractada se aleja de la normal N



# Refracción - medio mayor índice

- Cuando  $r=90^\circ \Rightarrow i = \lambda$  de tal modo que  $\sin \lambda = n_2/n_1 * \sin(90)$
- $\lambda$  se llama el ángulo límite
- Si  $i > \lambda$  no existe ninguna luz refractada, hay una reflexión total

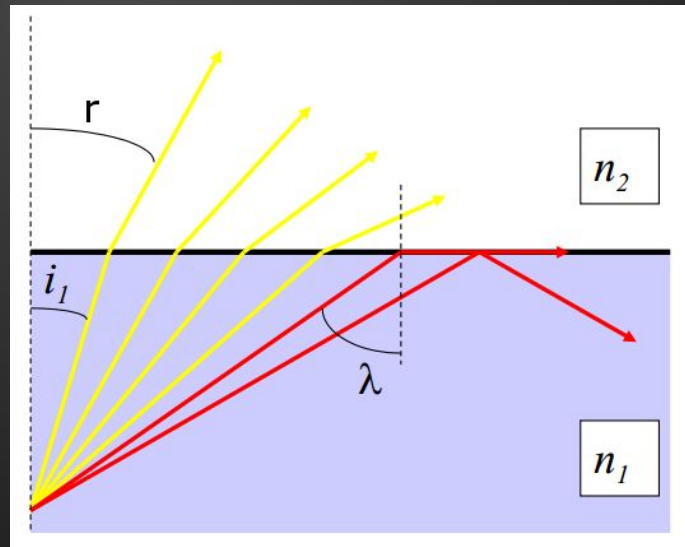


# Refracción - medio mayor índice

- Cuando  $r=90^\circ \Rightarrow i_1 = \lambda$  de tal modo que  $\sin \lambda = n_2/n_1$
- $\lambda$  se llama el ángulo límite
- Si  $i > \lambda$  no existe ninguna luz refractada, hay una reflexión total

$$n_1 > n_2 \quad \text{y} \quad i > \lambda$$

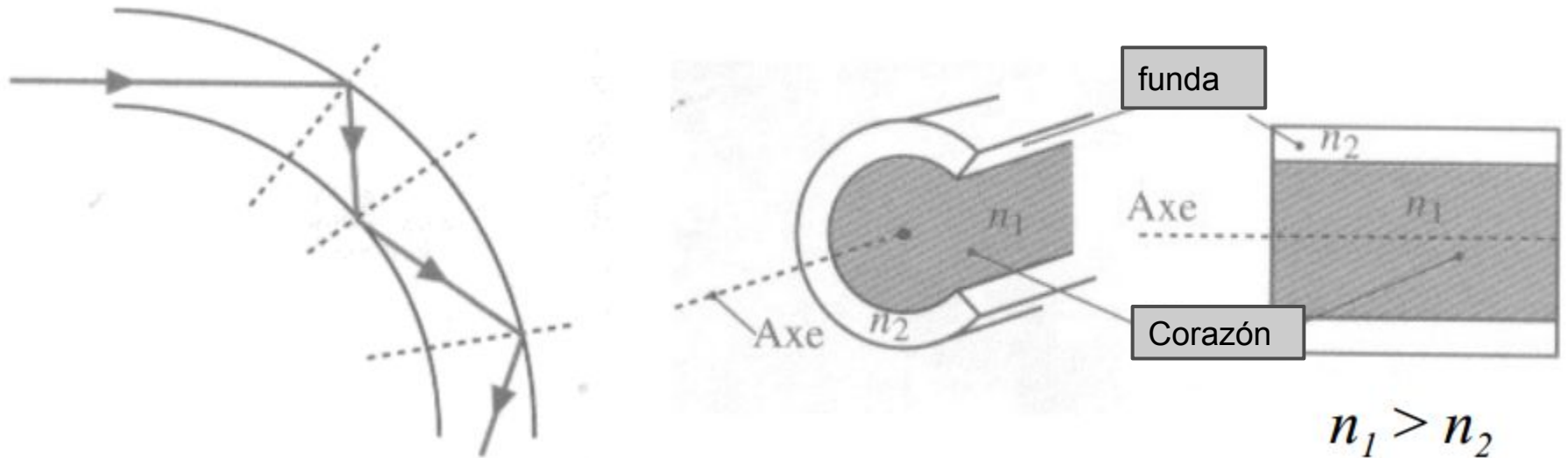
$$i = r$$





# Reflexión total - Aplicación

- Las fibras ópticas conducen la luz con reflexiones totales sucesivas. Son constituidas de un corazón  $n_1$  y de una funda  $n_2$  un índice menor a  $n_1$ .



# Absorción

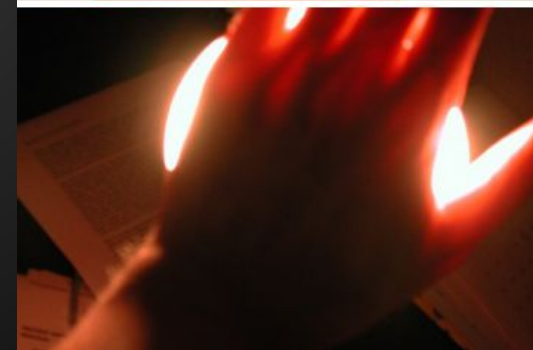
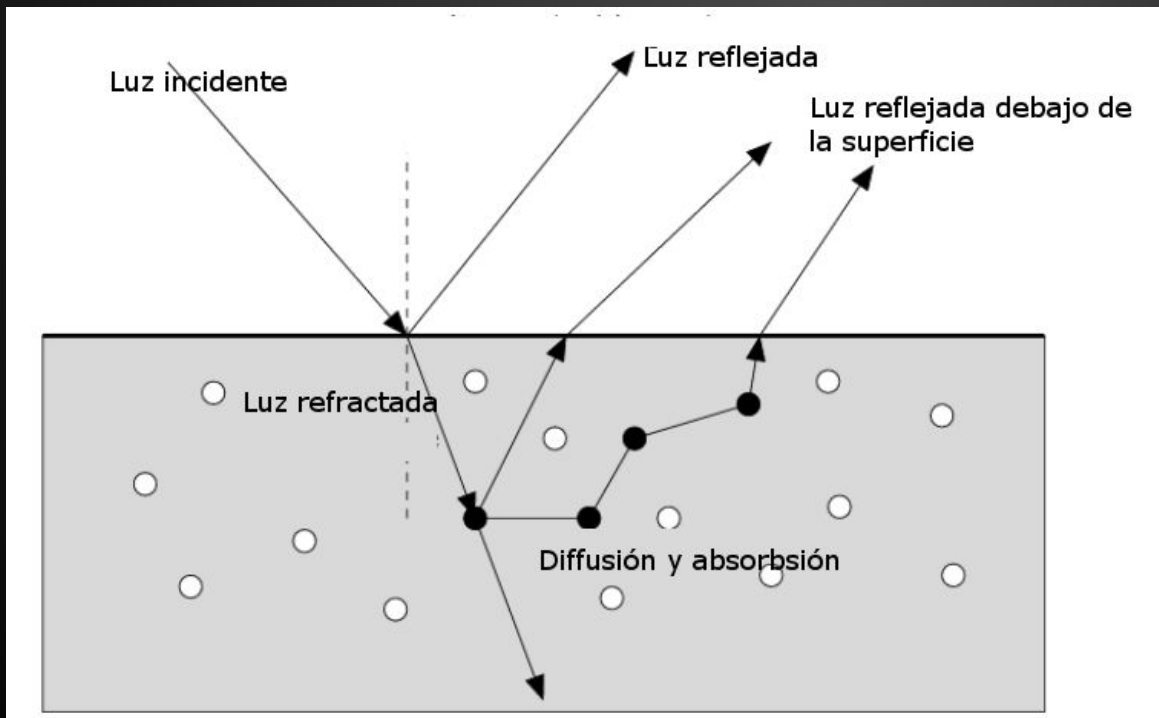
La parte de la luz incidente a una superficie que no es reflejada o refractada es absorbida.

La absorción corresponde a la captura de los fotones por los electrones de los átomos del material de objeto.

- Calentamiento del material
- Producción eléctrica

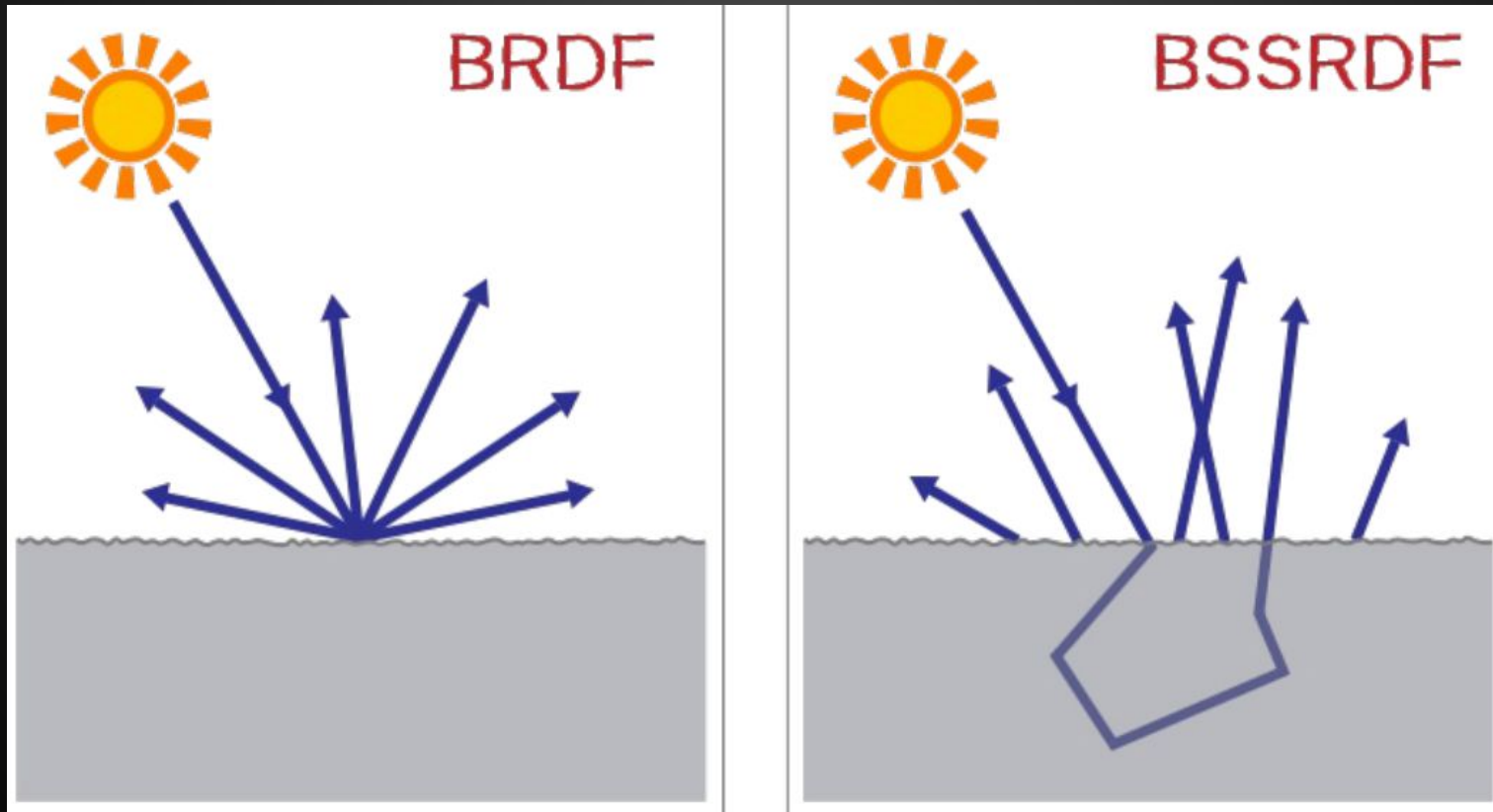
# Reflexión debajo debajo de la superficie

- Subsurface scattering
  - En caso de materiales translúcidos
  - La luz transmitida al material puede ser remitida por partículas hasta eventualmente ser emitida hasta la superficie

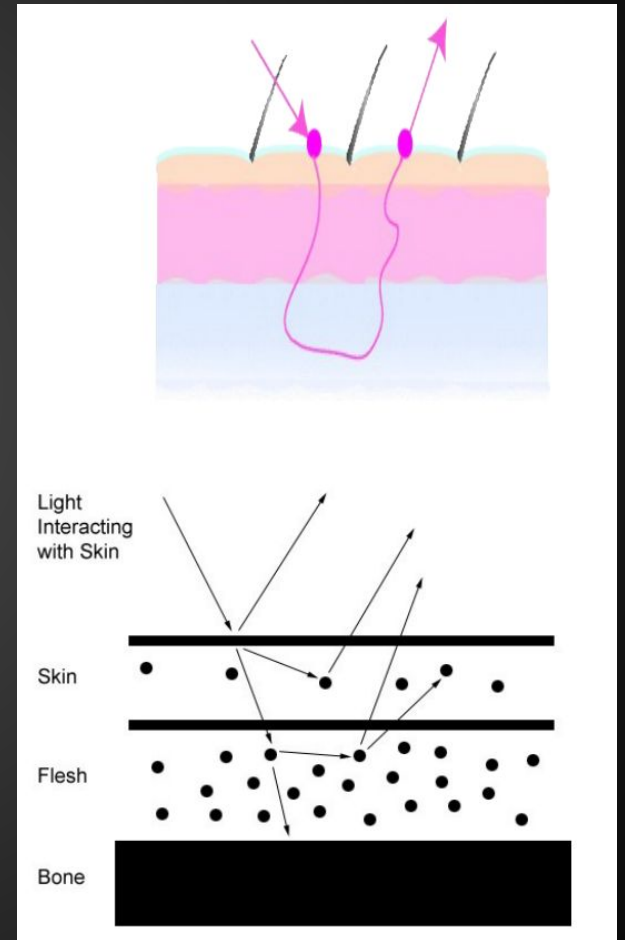


# Reflexión debajo debajo de la superficie

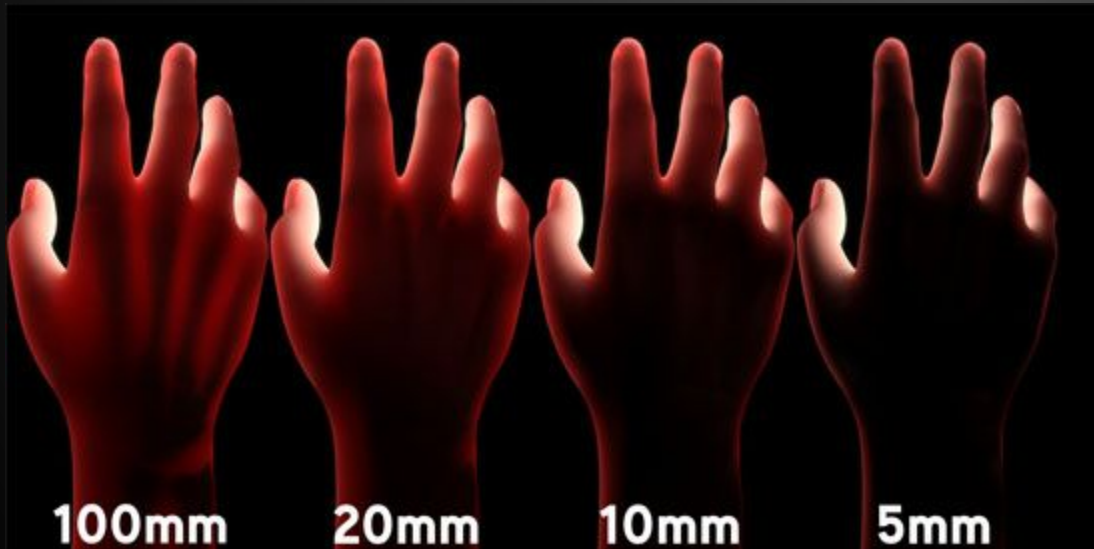
La función que permite conocer las direcciones de las reflexiones después de pasar debajo de la superficies relativo a la luz incidente se llama BSSRDF (Bidirectional surface scattering reflectance distribution function)



# Reflexión debajo debajo de la superficie



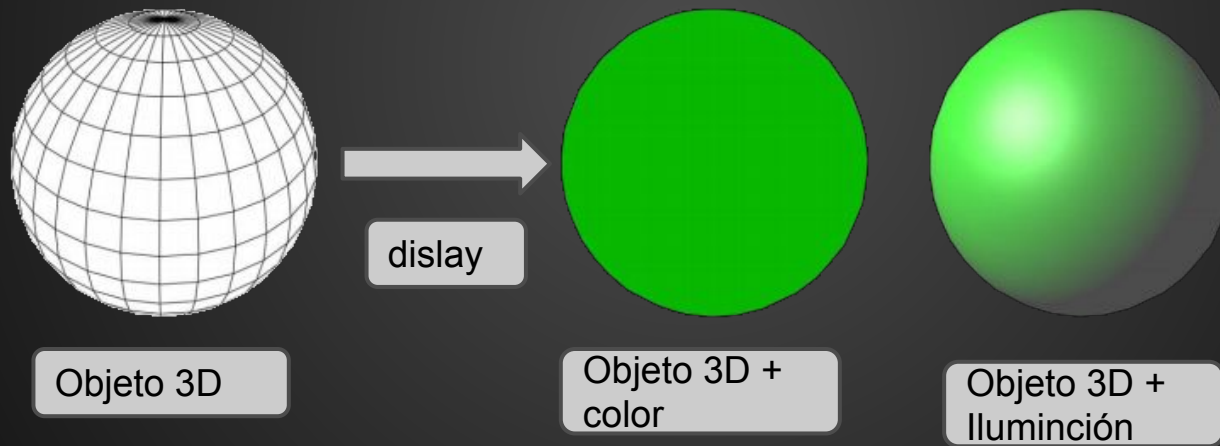
# Reflexión debajo debajo de la superficie





# Iluminación - OpenGL

Visualizar un objeto sin tomar en cuenta la luz no resulta realista.



Es necesario tomar en cuenta las interacciones de la luz con las superficies de los objetos => “Shading”

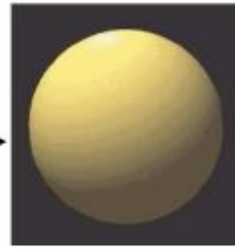
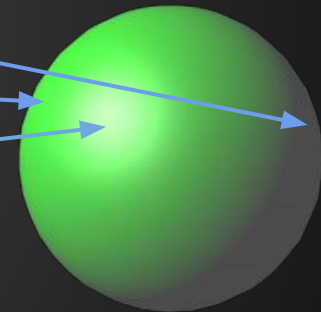
# Iluminación - Phong

Vemos la teoría del modelo de Phong, muy utilizado por sistemas de síntesis de imágenes como OpenGL. Los objetos son vistos porque reflejan la luz. En todo los puntos del objeto el color es dado por un suma :

reflexión ambiente  $I_a$

reflexión difusa  $I_d$

reflexión especular  $I_s$



Difusa + Especular + ambiente

Imagen final

Color del pixel :  $I = I_a + I_d + I_s$

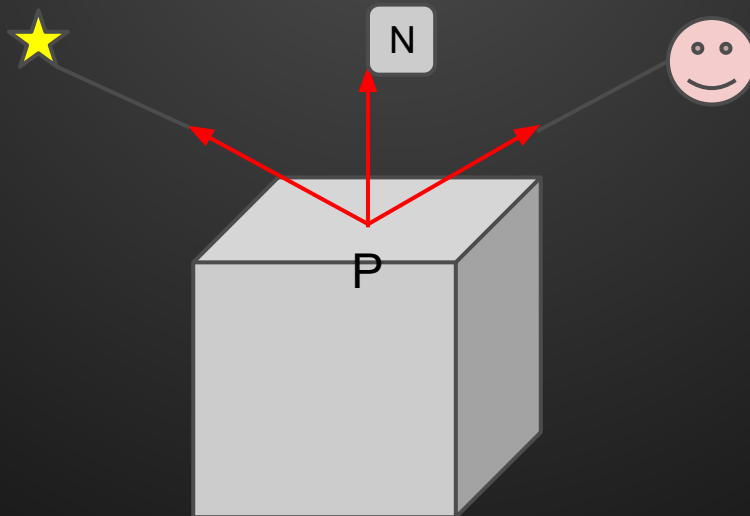


# Iluminación - Phong

- Calcular 3 veces esta suma para obtener los tres componentes R,G,B del color reflejado por el objeto
- Para calcular el color reflejado  $I$  en un pixel necesitamos
  - $I_{sa}, I_{sd}, I_{ss}$  : colores (RGB) ambiente, difusa, especular de la luz
  - $K_a, K_d, K_s$  : colores (RGB) ambiente, difusa, especular del material del objeto

# Ingredientes necesarios al cálculo de la iluminación

- Calcular la iluminación en un punto  $P$  de una superficie necesita
  - el vector normal a la superficie
  - el vector de dirección de la visualización
  - el vector de dirección de la luz



# Reflexión ambiente

El color ambiente de un objeto solo depende del coeficiente de reflexión ambiente  $K_a$  del objeto. no de su posición relativa a la luz.

$$I_a = I_{sa} + K_a$$

$I_a$  = Intensidad de la luz ambiente reflejada

$I_{sa}$  = intensidad de la luz ambiente incidente

$K_a$  = coeficiente de reflexión ambiente del objeto

# Reflexión ambiente

$K_a$  es un color RGB como  $I_a$  e  $I_{sa}$ , en el ejemplo debajo hacemos variar  $K_a$  de  $(R=0, G=0, B=0)$  hasta  $(R=1, G=0, B=0)$



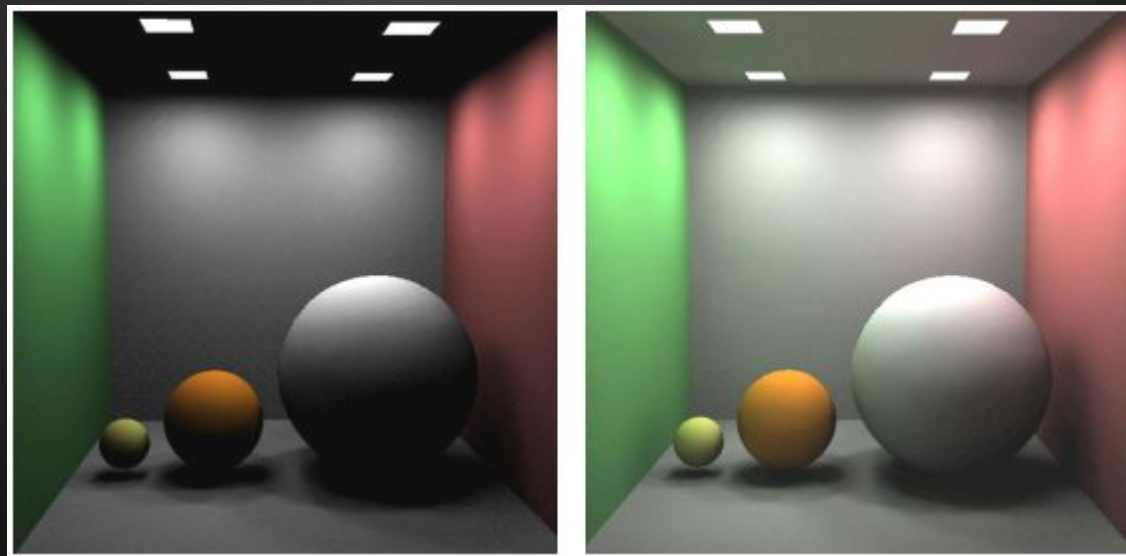
Hacemos varias  $K_a$



# Reflexión ambiente

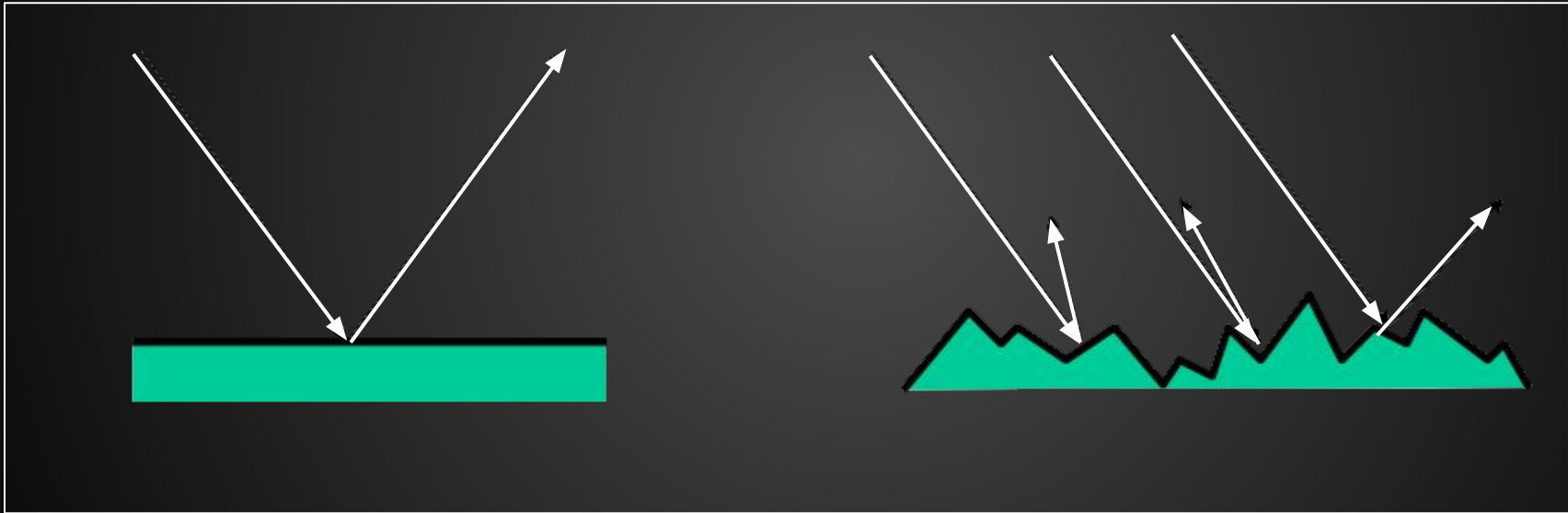
Modelo primitivo :

- No tiene sentido físico
- Permite maquillar problemas de la iluminación local. Problema resuelto gracias a la radiosidad (Iluminación global).



# Reflexión de la luz

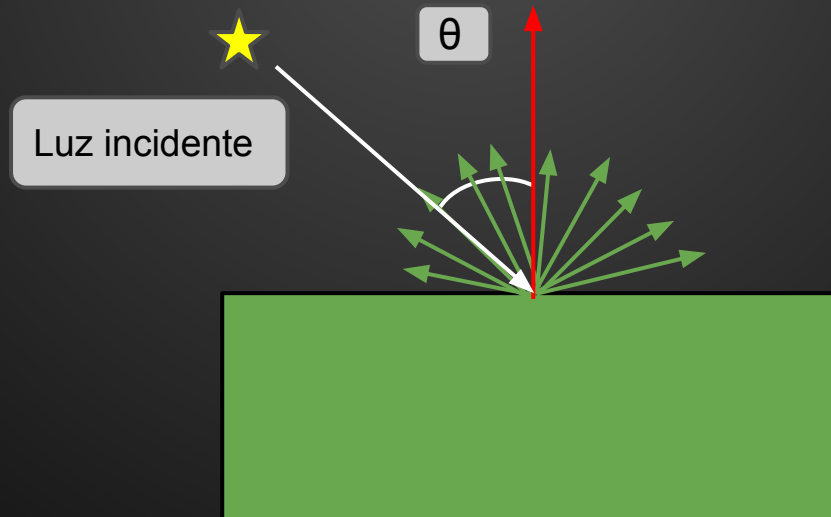
- La luz no es reflejada en una sola dirección sino en un conjunto de direcciones.



- Estas direcciones son divididas según un componente especular y un componente difusa.

# Reflexión difusa

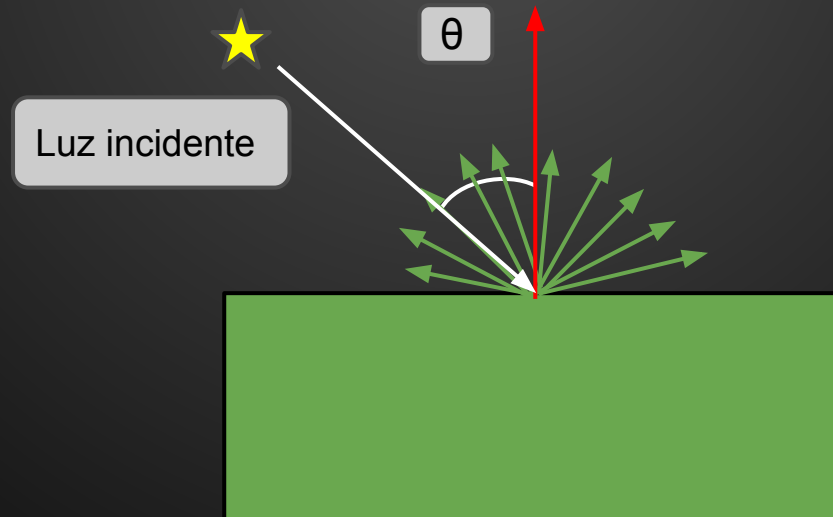
- Materiales mates
- La luz es reflejada en todas las direcciones (reflexión isótropa)
- El color del objeto es independiente de la posición del observador
- Solo depende del ángulo  $\theta$  entre la dirección de la luz y la normal a la superficie, y de coeficiente  $K_d$  de objeto (ley de Lambert)



# Reflexión difusa

$$I_d = I_{sd} \cdot K_d \cdot \cos \theta$$

- $I_d$  : intensidad luminosa de la luz difusa reflejada
- $I_{sd}$  : intensidad de la luz difusa
- $K_d$  : coeficiente de reflexión del material
- $\theta$  : ángulo entre la luz y la normal

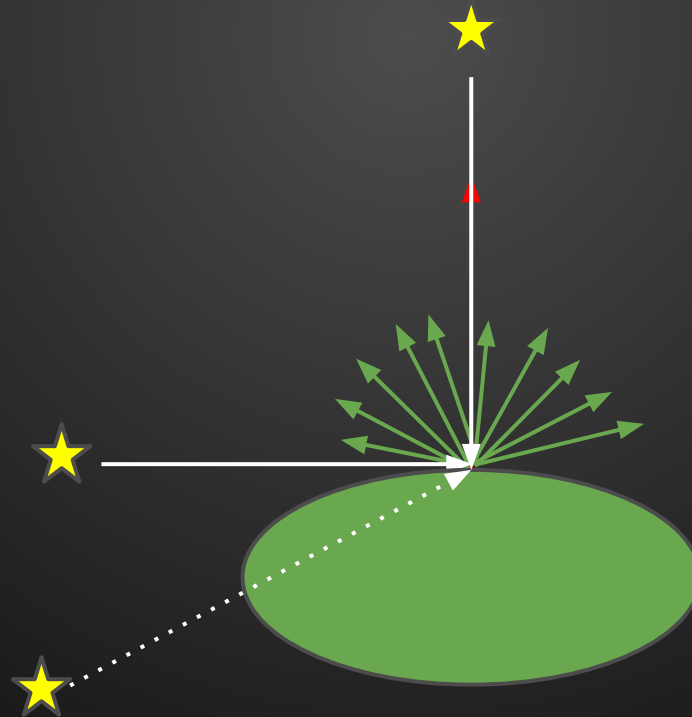




# Reflexión difusa

$$I_d = I_{sd} \cdot K_d \cdot \cos \theta$$

- La intensidad lumino es maximal para  $\theta=0$  (vertical a la superficie)
- Es nula para un ángulo  $\theta=90$
- Si  $\theta > 90^\circ$  el punto no es visible por la luz



# Reflexión difusa

$$I_d = I_{sd} \cdot K_d \cdot \cos \theta$$

- $k_d$  es un color RGB (idem que  $I_d$  y  $i_{sd}$ ). En el ejemplo debajo hacemos variar  $K_d$  de  $(R=0, G=0, B=0)$  hasta  $(R=1, G=0.5, B=0)$

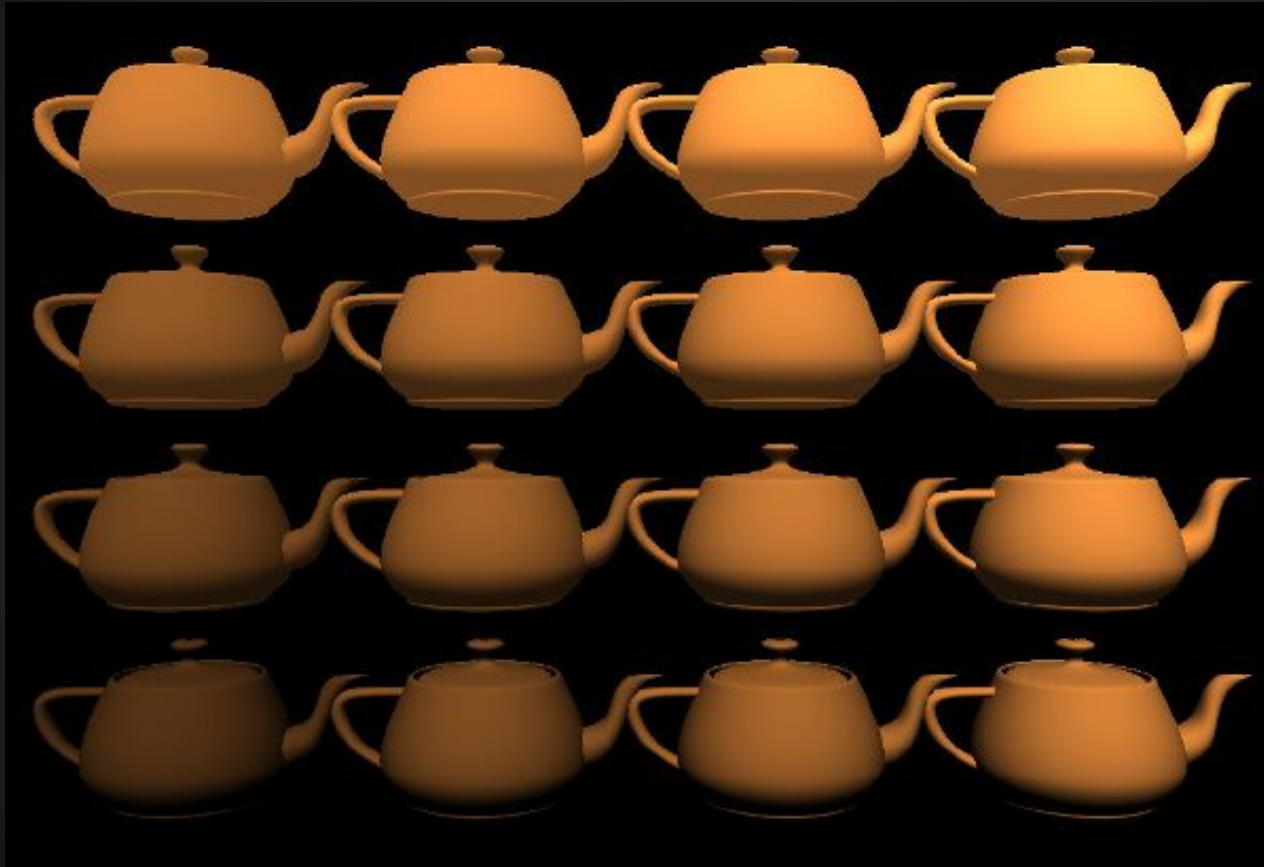


Hacemos varias  $K_d$



# Reflexión difusa + Reflexión Ambiente

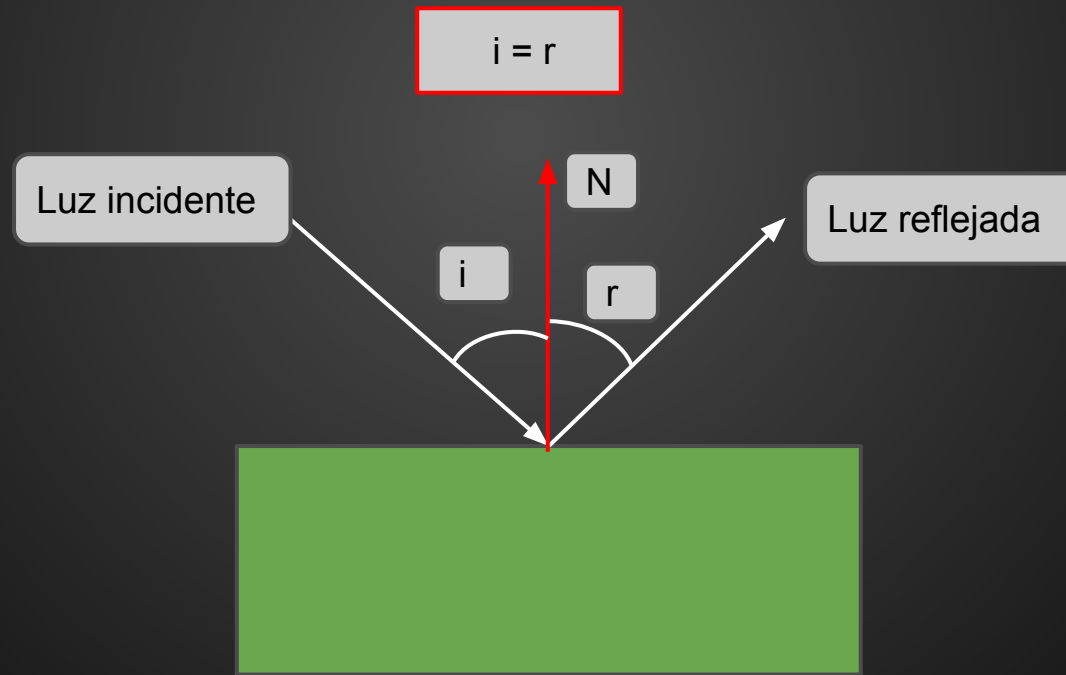
Hacemos varias  $K_a$



Hacemos varias  $K_d$

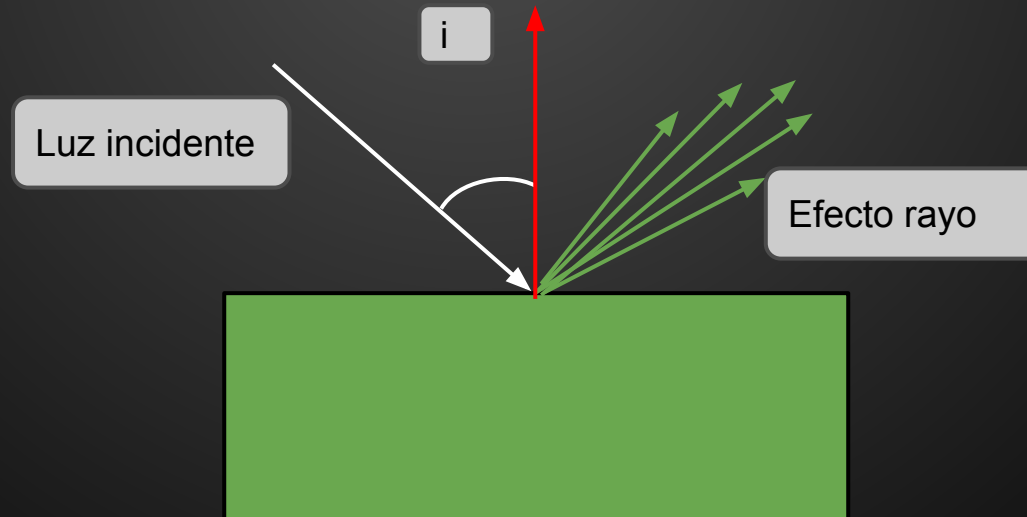
# Reflexión especular

- Permite obtener reflejos
- Espejo perfecto -> ley de Descarte



# Reflexión especular

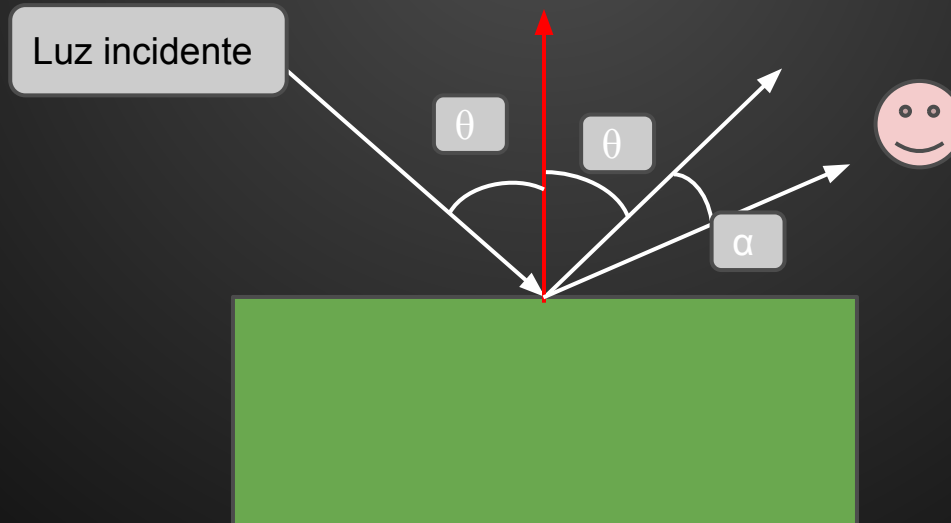
- Espejo imperfecto
- La luz es reflejada principalmente en la dirección perfecta
- La intensidad disminuye cuando no alejamos de esta dirección



# Reflexión especular

$$I_s = I_{ss} \cdot K_s \cdot \cos(\alpha)^n$$

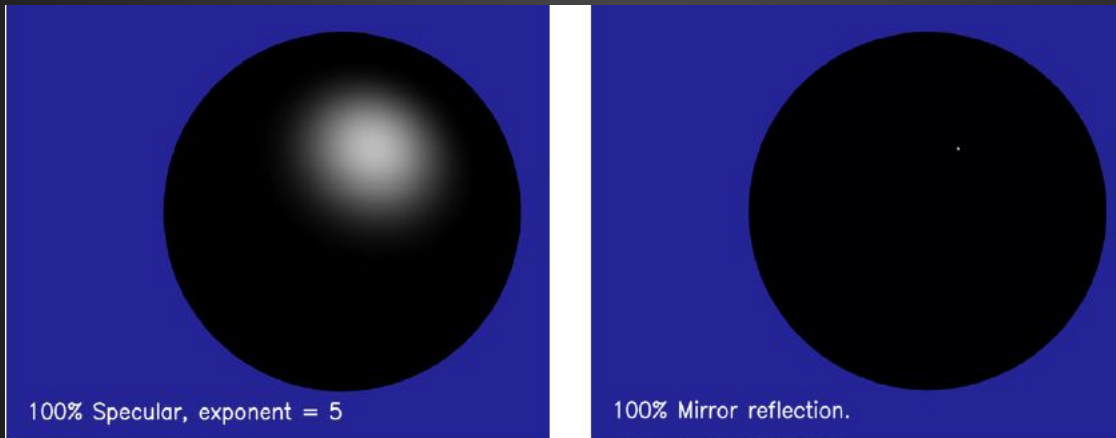
- $I_s$  : intensidad luminosa de la luz especular reflejada
- $I_{ss}$  : intensidad de la luz especular
- $K_s$  : coeficiente de reflexión especular del material
- $\alpha$  : angulo entre la dirección reflejada y de la cámara
- $n$  : coeficiente de brillo



# Reflexión especular

$$I_s = I_{ss} \cdot K_s \cdot \cos(\alpha)^n$$

- Más  $n$  es grande más estamos cerca de un espejo perfecto
- Pero no obtendremos un objeto con reflexión de su entorno, El tamaño del reflejo especular será más pequeño

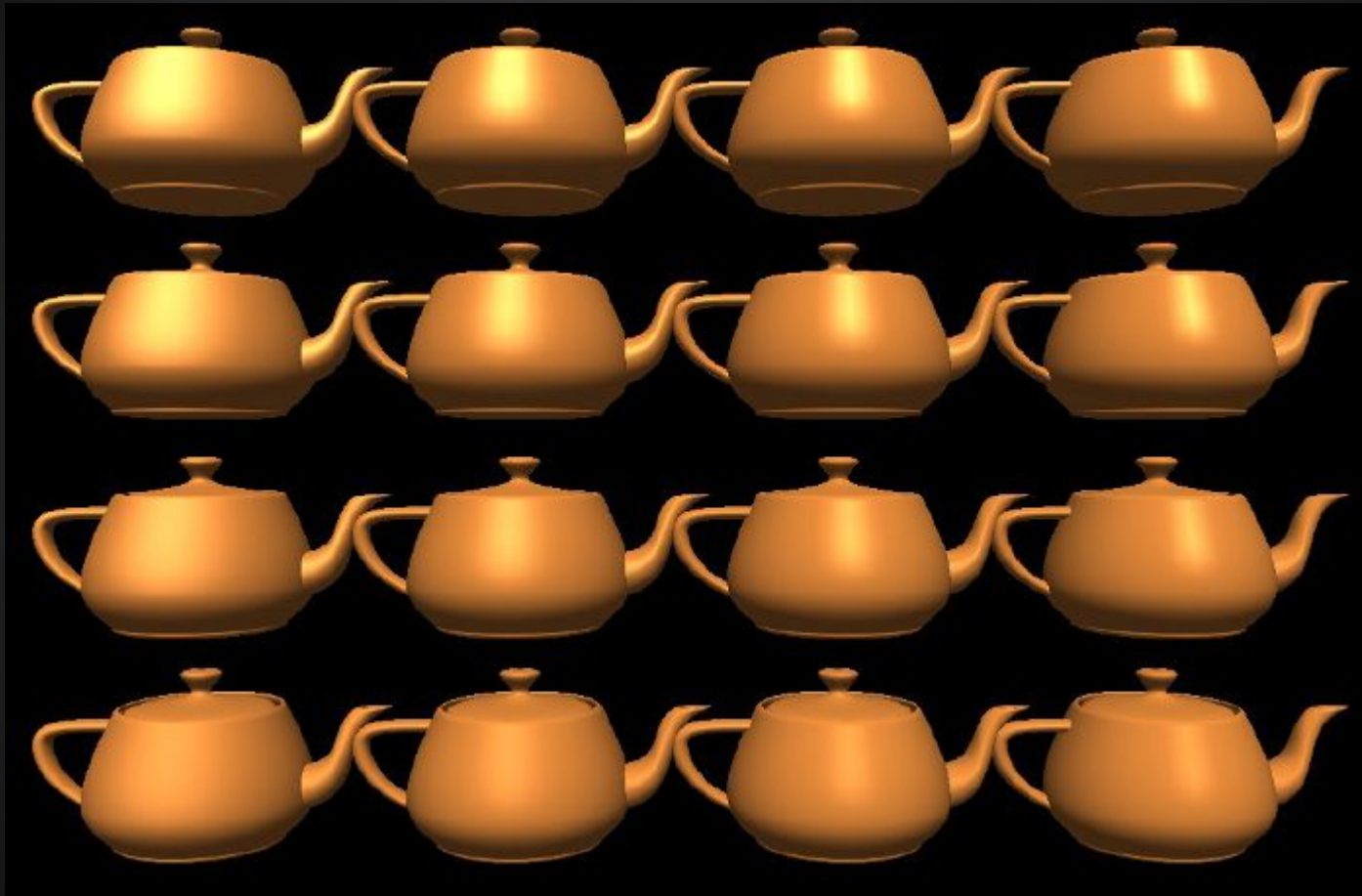


Para obtener un objeto que refleja su entorno necesitamos usar otras técnicas : environment mapping o shaders



# Reflexión difusa + Reflexión Ambiente

Hacemos varias  $K_s$

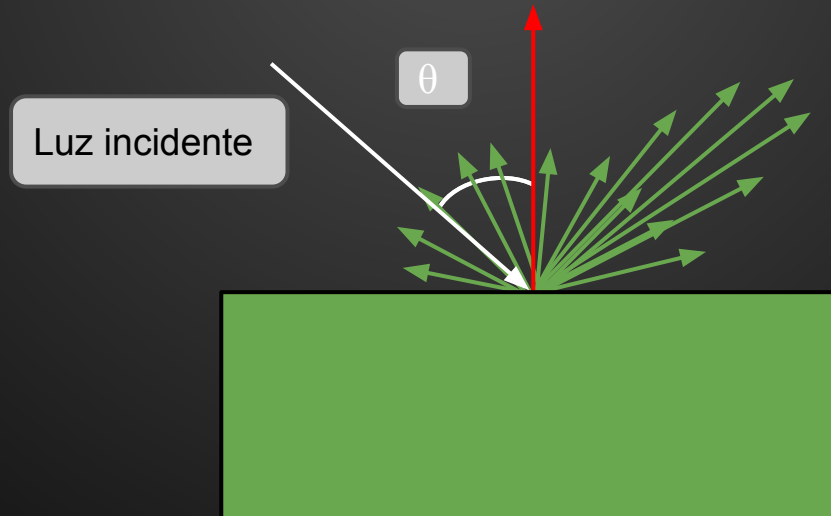


Hacemos varias  $n$



# Modelo de Phong final

- En la realidad la luz reflejada por una superficie es compuesta por la suma de la reflexión difusa y especular
- La proporción de reflexión difusa y especular dependen de material. Algunos son más difusos (papel, madera, tela) que especular (vidrio, diamante, metal).



# Modelo de Phong final

Es la suma de las reflexiones ambiente, difusa y especular :

$$I = I_a + I_d + I_s$$

$$I = I_a + K_a + I_{sd} \cdot K_d \cdot \cos \theta + I_{ss} \cdot K_s \cdot \cos(\alpha)^n$$

Si tenemos varias fuentes de luz : suma de las intensidades

modelo utilizado por OpenGL y DirectX

# Cálculos final del color

- Sumamos la intensidad luminosa de cada componente del color
- En el sistema RGB, agregamos las intensidad Rojo , Verde y Azul
- Definimos por cada una de estas 3 componentes
  - las características de las fuentes de luz
    - $I_{saR}, I_{saG}, I_{saB}$  ;  $I_{sdR}, I_{sdG}, I_{sdB}$  ;  $I_{ssR}, I_{ssG}, I_{ssB}$  ;
  - las características de los materiales
    - $K_{aR}, K_{aG}, K_{aB}$  ;  $k_{dR}, K_{dG}, K_{dB}$  ;  $K_{sR}, K_{sG}, K_{sB}$  ;

# Cálculos final del color

- Las intensidades luminosa por cada una de los 3 componentes se obtienen así :

$$I_r = I_{saR} + K_{aR} + I_{sdR} \cdot K_{dR} \cdot \cos \theta + I_{ssR} \cdot K_{sR} \cdot \cos(\alpha)^n$$

$$I_g = I_{saG} + K_{aG} + I_{sdG} \cdot K_{dG} \cdot \cos \theta + I_{ssG} \cdot K_{sG} \cdot \cos(\alpha)^n$$

$$I_b = I_{saB} + K_{aB} + I_{sdB} \cdot K_{dB} \cdot \cos \theta + I_{ssB} \cdot K_{sB} \cdot \cos(\alpha)^n$$

Color final de  
objeto

Color ambiente  
de la luz

Color difuso de  
la luz

Color ambiente  
del objeto

Color difuso del  
objeto

Color especular  
de la luz

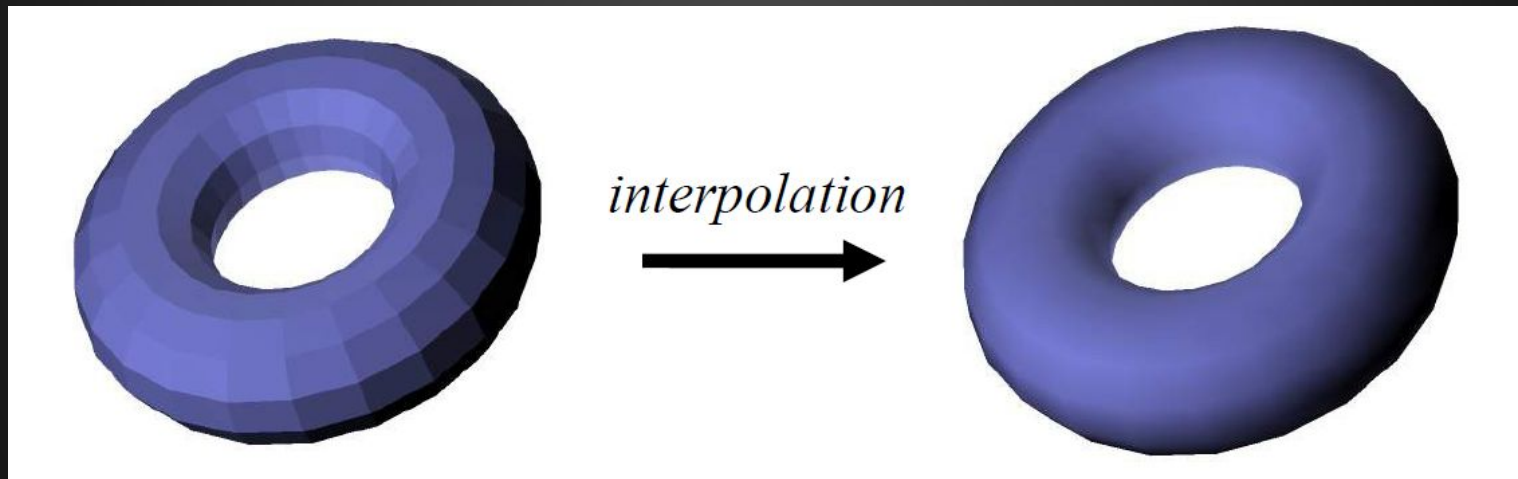
Color especular  
del objeto

# Modelo de Phong

- Ventaja del modelo de Phong
  - Muy práctico (simple, resultados interesantes)
  - Muy rápido
- Desventaja
  - No sentido físico
  - No relación con el material físico : rugosidad etc...
- Existen modelos de iluminación más realistas pero más complejos (Cook-Torrance)
- **Pipeline programable : podemos programar nuestro modelo de iluminación (shaders)**

# Utilización de la iluminación sobre un objeto 3D

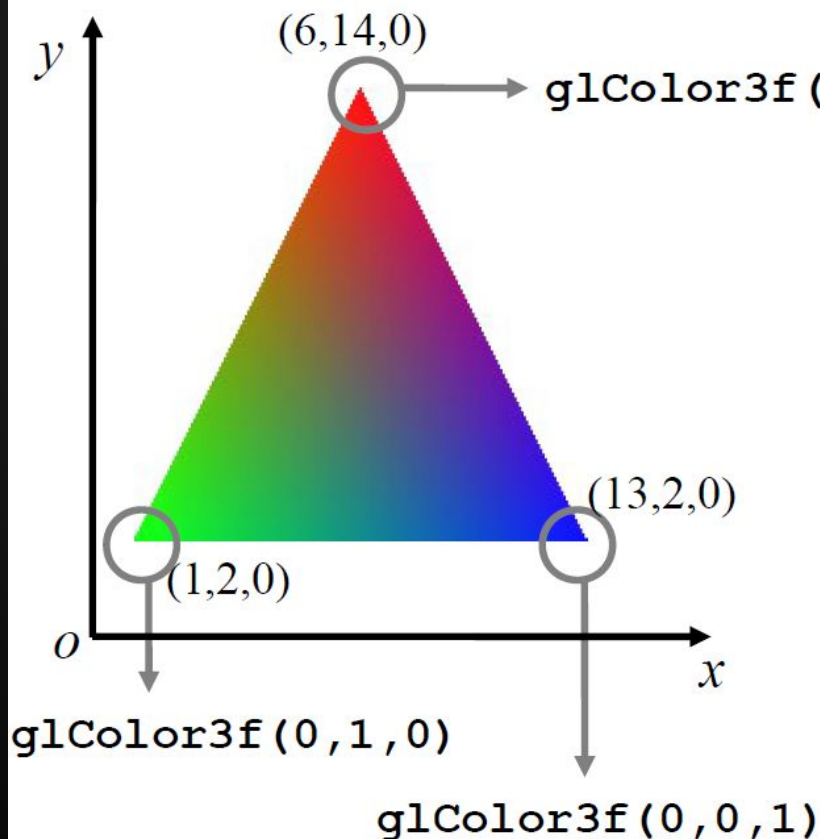
- Problema, calculamos la misma iluminación por cada polígono del objeto
  - Flat shading



- Solución : calcular la iluminación por cada vértice del objeto para interpolar el color entre cada vertice.

# Utilización de la iluminación sobre un objeto 3D

- OpenGL interpola los colores de cada vértice del triángulo para obtener el color de los demás puntos del triángulo



```
glBegin(GL_TRIANGLES);  
    glColor3f (1, 0, 0);  
    glVertex3f (6, 14, 0);  
    glColor3f (0, 1, 0);  
    glVertex3f (1, 2, 0);  
    glColor3f (0, 0, 1);  
    glVertex3f (13, 2, 0);  
glEnd();
```

# Utilización de la iluminación sobre un objeto 3D

- Interpolación de la iluminación
  - Objetivo : Calcular un color por cada punto visible del objeto que se visualiza.
    - Alisamiento de Phong : interpolación de normales
    - Alisamiento de Gouraud : interpolación de colores
- No hay que confundir la iluminación de Phong con el Alisamiento de Phong



# Utilización de la iluminación sobre un objeto 3D

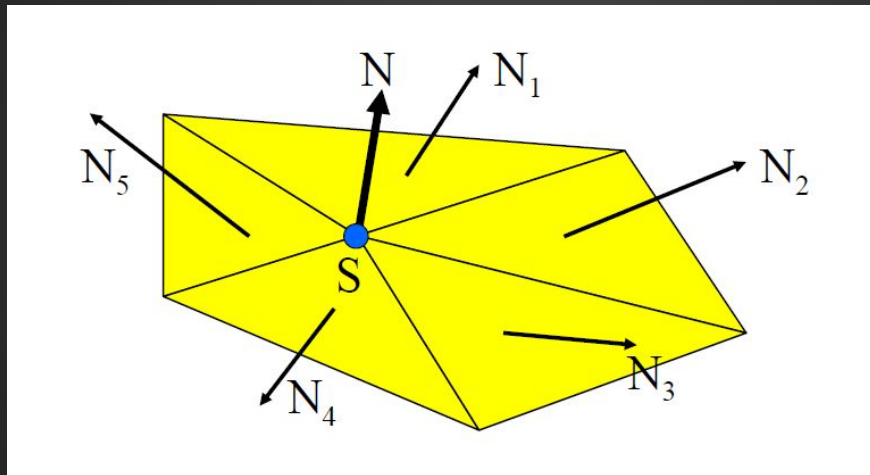
- Interpolación de Gouraud
  - Calcular para cada vértice un color gracias a un modelo de iluminación ( Phong, ...)
  - Interpoliar los colores de los vértices para calcular el color de cada pixel de polígono .

# Utilización de la iluminación sobre un objeto 3D

- Interpolación de Gouraud
  - Para calcular un color en un vértice, necesitamos una normal en este punto.
    - Si la superficie es analíticamente conocida (esfera, cilindro, cono, ...) podemos calcular la normal simplemente en este punto.
    - Qué hacemos si nuestro objeto es una malla poligonal ?

# Utilización de la iluminación sobre un objeto 3D

- La normal a un vértice de un polígono = promedio de las normales a todos los polígonos que comparten este vértice.

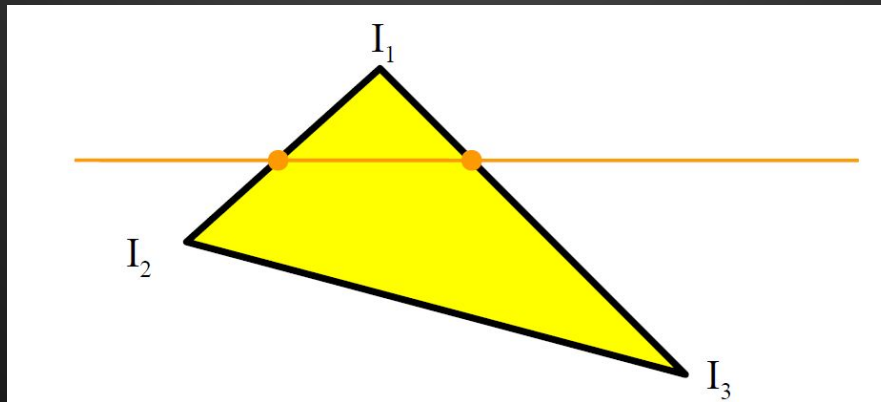


Normal en el vertice S:

$$N = (N_1 + N_2 + N_3 + N_4 + N_5) / 5$$

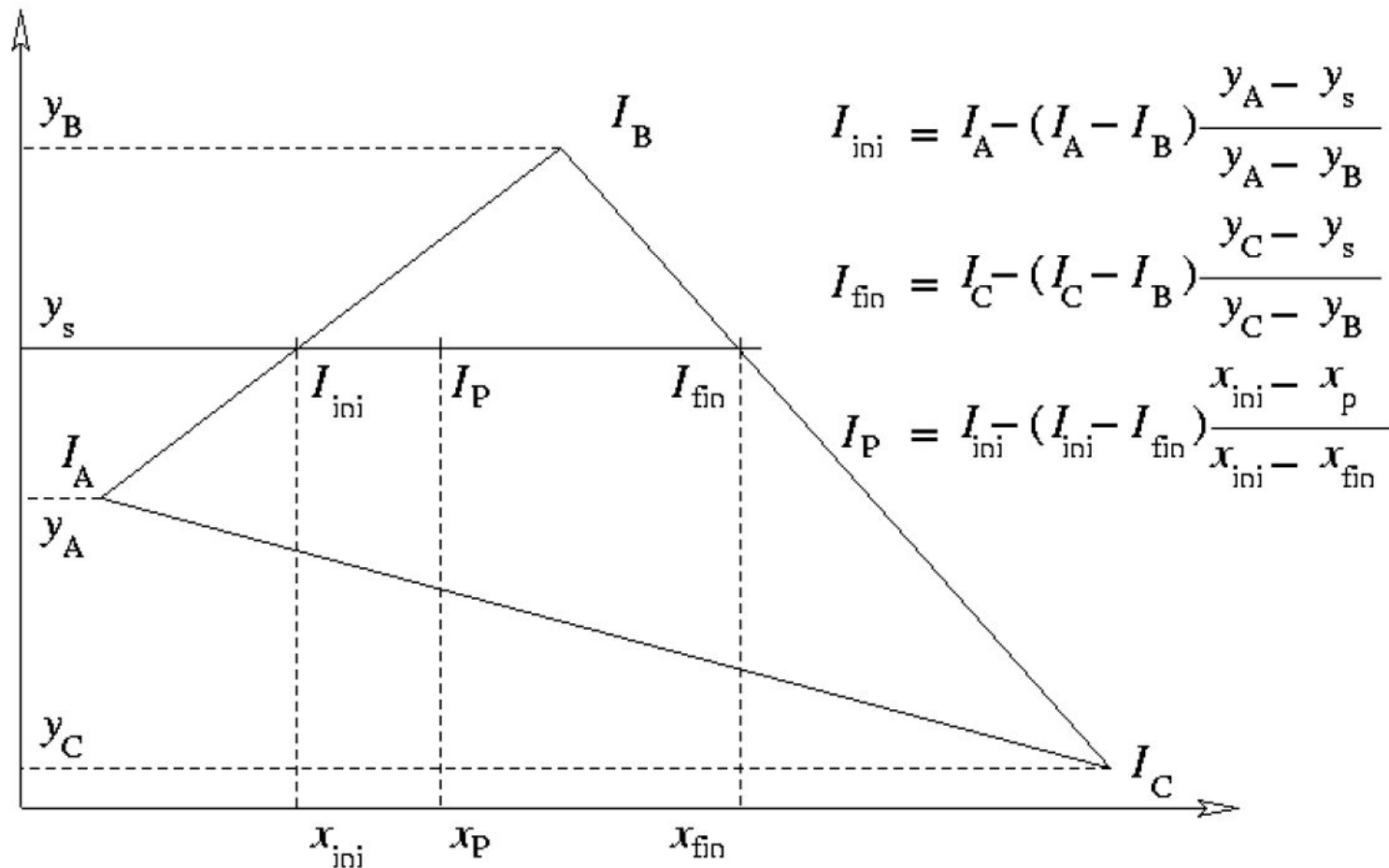
# Utilización de la iluminación sobre un objeto 3D

- Una vez que tenemos una normal para cada vértice de un polígono :
  - Calcular un color para cada vértice (con un modelo de iluminación)
  - Sobre una arista interpolar el color entre dos vértices
  - Sobre una scanline del polígono interpolar el color entre dos aristas.



# Utilización de la iluminación sobre un objeto 3D

Cálculos de la interpolación de Gouraud

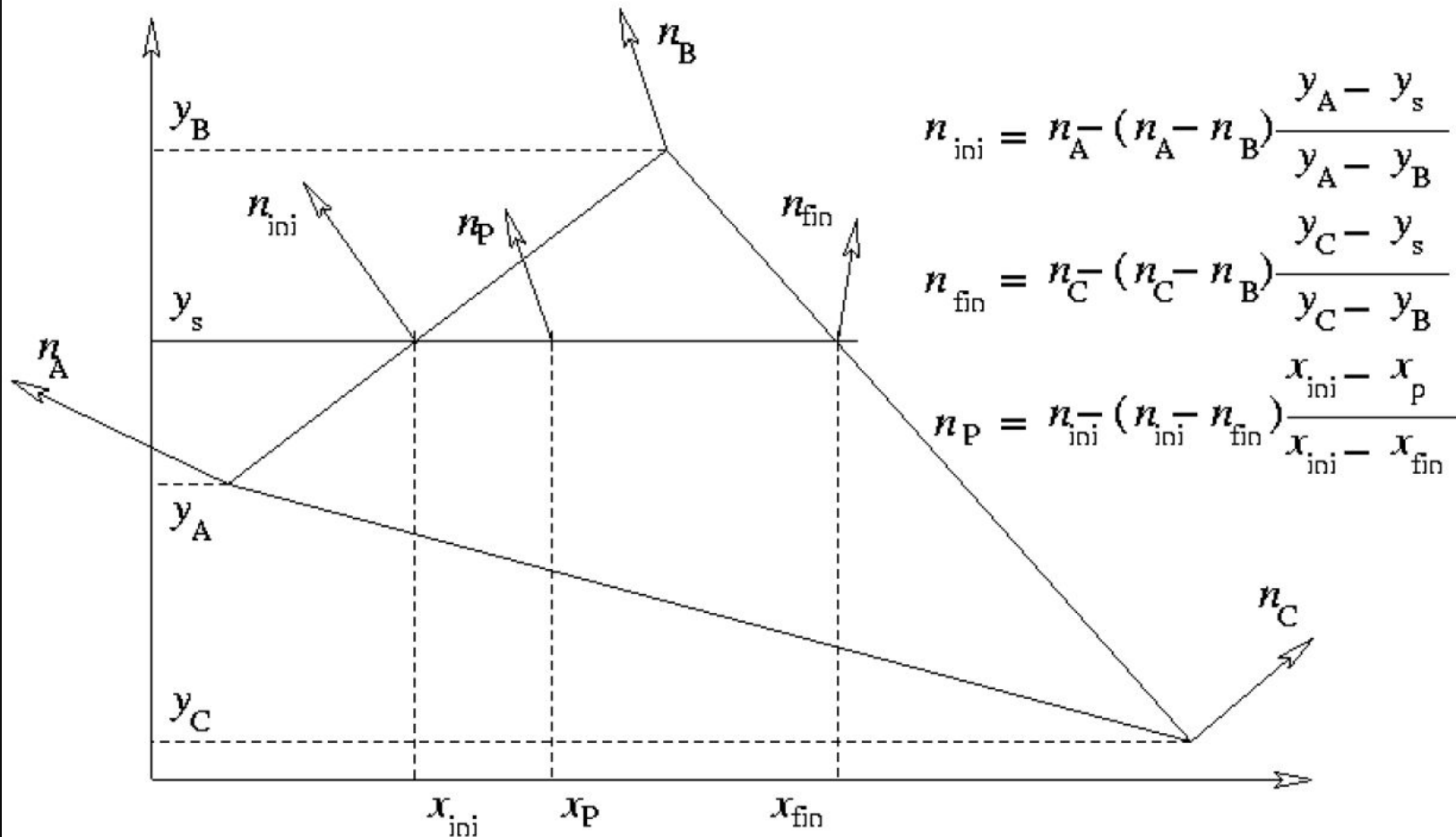


# Utilización de la iluminación sobre un objeto 3D

- Interpolación de Phong
  - Al lugar de interpolar los colores interpolamos las normales sobre las aristas entre los vértices
  - Sobre un scanline interpolamos las normales entre dos aristas
  - Calculamos para cada pixel el modelo de iluminación

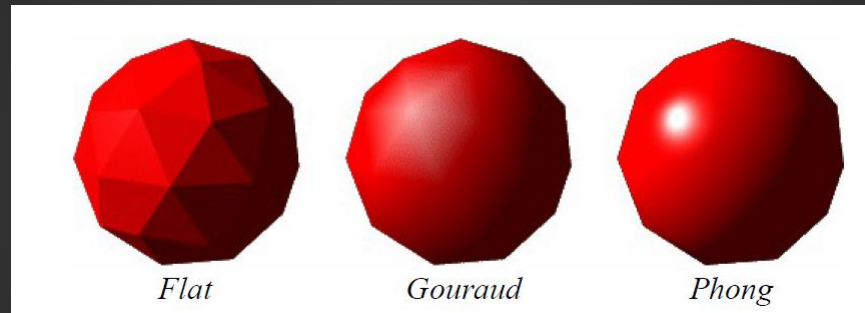
# Utilización de la iluminación sobre un objeto 3D

## Cálculos de la interpolación de Phong

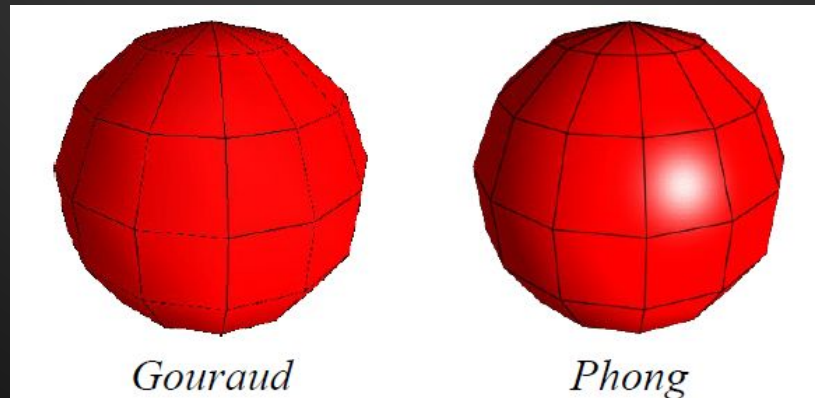


# Utilización de la iluminación sobre un objeto 3D

Alisamiento de phong más lento que gouraud (calcula la iluminación en cada pixel y no en cada vértice ) pero mucho es mucho más realista.



Phong permite calcular reflejos especular dentro de un polígono lo cual es imposible con Gouraud



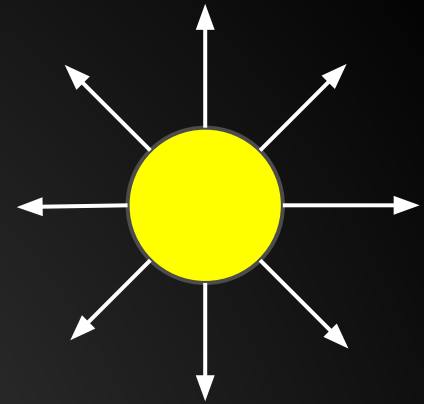


# Fuente de luz en OpenGL

- Todas las fuentes luminosas son definidas gracias a 3 vectores de 4 componentes cada uno RGBA (Alpha)
- Luz ambiente (valor por defecto  $\langle 0,0,0,1 \rangle$ )
- Luz difusa (valor por defecto  $\langle 1,1,1,1 \rangle$ )
- Luz especular (valor por defecto  $\langle 1,1,1,1 \rangle$ )

3 tipos de fuentes luminosas son soportadas

# Fuente de luz en OpenGL



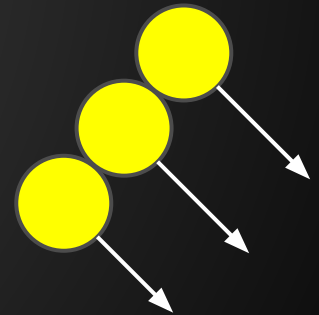
## 1. Fuente puntual

La luz viene de un punto específico coordenadas homogéneas de un punto

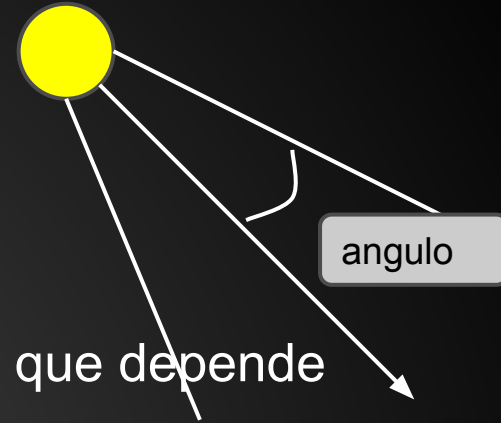
```
GLfloat position[] = {5.0,17.0,9.0,1.0};  
glLightfv(GL_LIGHT0, GL_POSITION, position);
```

## 2. Fuente direccional

```
GLfloat direction[] = {8.0,2.0,14.0,0.0};  
glLightfv(GL_LIGHT0, GL_POSITION, direction);
```



# Fuente de luz en OpenGL



## 3. Spot

La luz viene de un punto específico con una intensidad que depende de su dirección

- Position : posición de la fuente luminosa

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

- Direction : dirección de eje central de la luz

```
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, direction);
```

- Angle : ancho del rayo

```
glLightfv(GL_LIGHT0, GL_SPOT_CUTOFF, 45.0);
```

- Potencia : Disminución de la luz al acercarse a los conos

```
glLightfv(GL_LIGHT0, GL_SPOT_EXPONENT, 1.0);
```

# Fuente de luz en OpenGL

- Utilización :

// Valores de colores : luz roja

```
GLfloat Light0Amb[4] = {0.5f, 0.0f, 0.0f, 1.0f};
```

```
GLfloat Light0Dif[4] = {1.0f, 0.0f, 0.2f, 1.0f};
```

```
GLfloat Light0Spec[4] = {0.2f, 0.2f, 0.2f, 1.0f};
```

// Valores de posición : luz puntual

```
GLfloat Light0Pos[4] = {0.0f, 20.0f, 0.0f, 1.0f};
```

# Fuente de luz en OpenGL

- Utilización :

```
// los parámetros de colores de la luz 0
glLightfv(GL_LIGHT0, GL_AMBIENT, Light0Amb);
glLightfv(GL_LIGHT0, GL_DIFFUSE, Light0Dif);
glLightfv(GL_LIGHT0, GL_SPECULAR, Light0Spec);
// la posición de la luz 0
glLightfv(GL_LIGHT0, GL_POSITION, Light0Pos);
// Activación de la luz
glEnable(GL_LIGHTING);
//Activación de la luz 0
glEnable(GL_LIGHT0);
```

# Fuente de luz en OpenGL

- Observación :

La cantidad de luces en OpenGL es limitada, es minimo de 8, este valor es definido en la constante : `GL_MAX_LIGHTS`

`glEnable(GL_LIGHTING)` permite de autorizar el uso de la iluminación

`glEnable(GL_LIGHT0)` permite de activar la primera luz `GL_LIGHT0`

Podemos desactivar la iluminación

- General : `glDisable(GL_LIGHTING)`
- De una fuente de luz en particular : `glDisable(GL_LIGHT0)`

# Materiales en OpenGL

- Todos los materiales son definidos gracias a 3 vectores de 4 componentes cada uno RGBA (Alpha)
- coeficiente de reflexión ambiente (valor por defecto  $\langle 0,0,0,1 \rangle$ )
- coeficiente de reflexión difusa (valor por defecto  $\langle 1,1,1,1 \rangle$ )
- coeficiente de reflexión especular (valor por defecto  $\langle 1,1,1,1 \rangle$ )

Así que el coeficiente de brillo  $n$  del  $\cos(\alpha)^n$  de la reflexión especular

# Materiales en OpenGL

Utilización :

```
GLfloat MatAmbient[4] = {0.0f, 0.5f, 0.0f, 1.0f};
```

```
GLfloat MatDiffuse[4] = {0.0f, 0.8f, 0.0f, 1.0f};
```

```
GLfloat MatSpecular[4] = {0.2f, 0.2f, 0.2f, 1.0f};
```

```
GLfloat MatShininess[] = { 5.0F };
```

```
glMaterialfv(GL_FRONT, GL_AMBIENT , MatAmbient);
```

```
glMaterialfv(GL_FRONT, GL_DIFFUSE , MatDiffuse);
```

```
glMaterialfv(GL_FRONT, GL_SPECULAR, MatSpecular);
```

```
glMaterialfv(GL_FRONT, GL_SHININESS, MatShininess);
```



# Neblina

OpenGL permite fácilmente la creación del efecto de neblina



Sin neblina

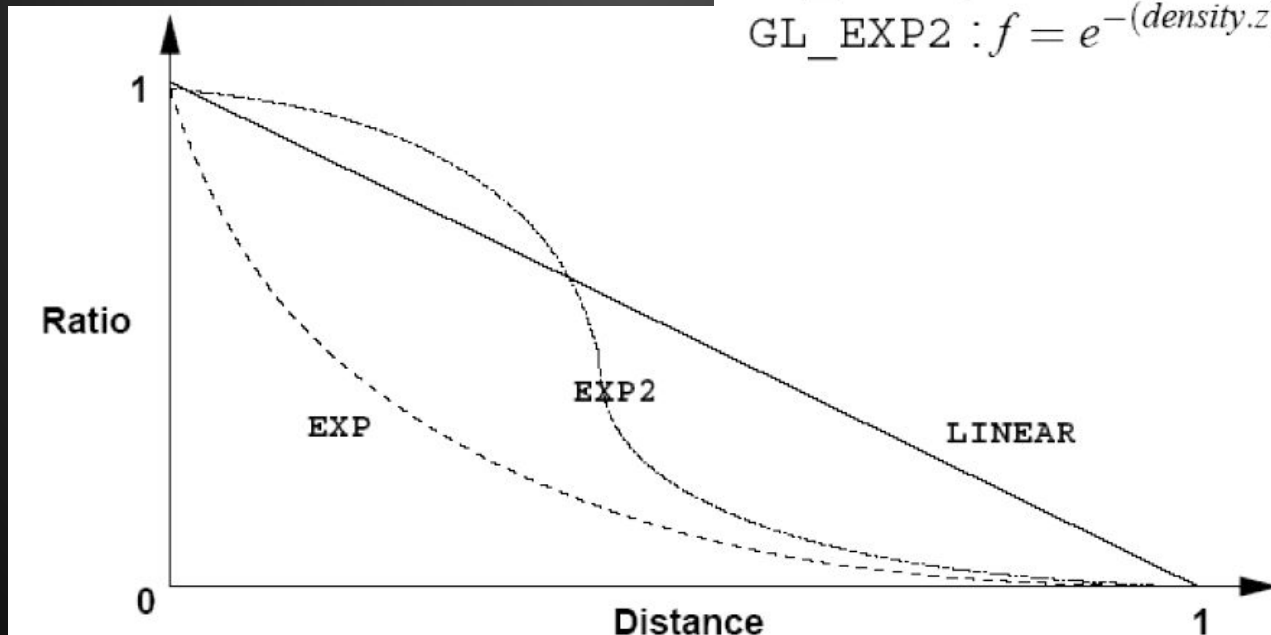
Con neblina

# Neblina

El color de la neblina puede ser mezclada a los objetos en función a 3 métodos

- Atenuación lineal
- Atenuación exponencial
- Atenuación exponencial 2

$$\begin{aligned} \text{GL\_LINEAR} : f &= \frac{\text{end} - z}{\text{end} - \text{start}} \\ \text{GL\_EXP} : f &= e^{-(\text{density} \cdot z)} \\ \text{GL\_EXP2} : f &= e^{-(\text{density} \cdot z)^2} \end{aligned}$$



# Neblina

Distancia de inicio y fin de la neblina. Entre estos dos valores, el color de la neblina es mezclado a los colores de los objetos linealmente en función a la distancia.

```
GLfloat fogColor[4]= {0.4f,0.4f,0.4f,0.0f};  
glFogf(GL_FOG_MODE, GL_LINEAR);  
glFogf(GL_FOG_START, 100); // défaut : 0.0f  
glFogf(GL_FOG_END, 800); // défaut : 1.0f  
glFogfv(GL_FOG_COLOR, fogColor);  
glEnable(GL_FOG);
```

# Neblina

Densidad de la neblina. El color de la neblina es mezclado a los colores de los objetos de manera exponencial en función a la distancia.

```
GLfloat fogColor[4]= {0.4f,0.4f,0.4f,0.0f};  
glFogf(GL_FOG_MODE, GL_EXP); // ou GL_EXP2  
glFogf(GL_FOG_DENSITY, 2.0f); // défaut : 1.0f  
glFogfv(GL_FOG_COLOR, fogColor);  
glEnable(GL_FOG);
```

# Neblina

- Reproducir fenómeno natural
- Creación de ambiente (misterioso, ...)



This place hasn't changed at all  
in three years...

# Neblina

- Reproducir fenómeno natural
- Creación de ambiente (misterioso, ...)





# Neblina

- Reproducir fenómeno natural
- Creación de ambiente (misterioso, ...)

