

Tema 4: Modelos de Iluminación y Sombreado

José Ribelles

Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I

SIU020 - Síntesis de Imagen y Animación

Contenido

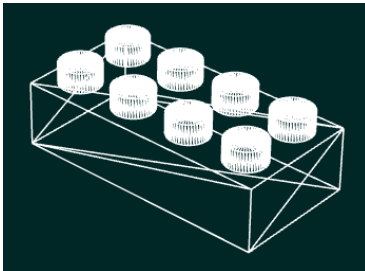
- 1 **Introducción**
- 2 **Modelo de Iluminación de Phong**
 - Luz Ambiente
 - Reflexión Difusa
 - Reflexión Especular
 - Materiales
 - El modelo de Phong
- 3 **Tipos de Fuentes de Luz**
- 4 **Modelos de Sombreado**
- 5 **Casos prácticos**
 - Sombreado Cómico
 - Niebla

Hoy veremos...

- 1 **Introducción**
- 2 Modelo de Iluminación de Phong
- 3 Tipos de Fuentes de Luz
- 4 Modelos de Sombreado
- 5 Casos prácticos

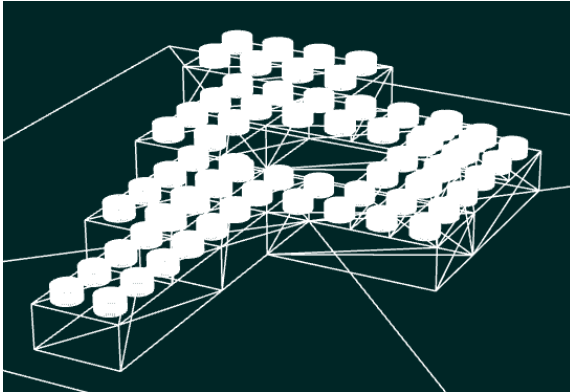
Introducción

Ya sabemos cómo crear y visualizar un modelo como este:



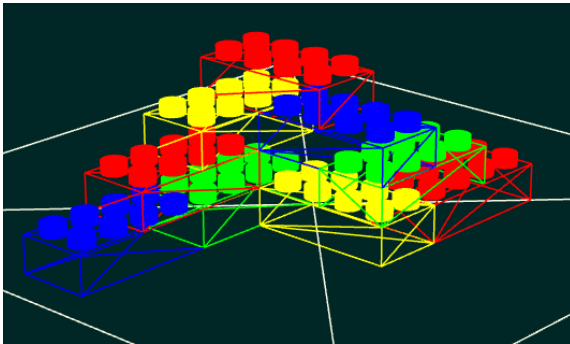
Introducción

Y a partir de dicho modelo sabemos también cómo crear y visualizar una escena como esta:



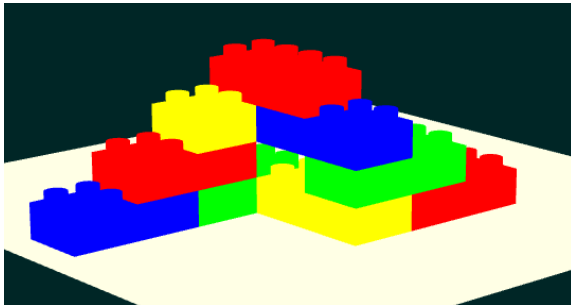
Introducción

E incluso añadir colores:



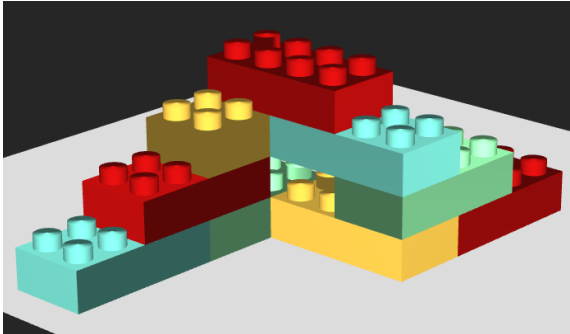
Introducción

Y pintar superficies en lugar de solo líneas:



Introducción

En este tema vamos a conocer cómo mejorar la calidad visual añadiendo iluminación:



Introducción

¿Qué trabajo conlleva obtener dicho resultado?

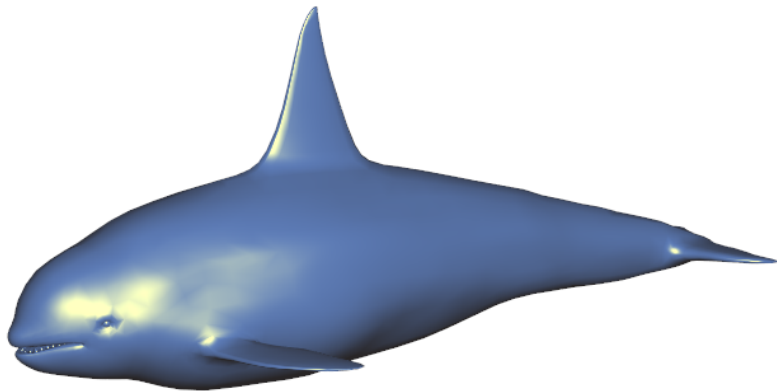
- Incorporar un modelo de iluminación.
 - Propiedades y características de la fuente de luz.
 - Propiedades de material.
- Incorporar un modelo de sombreado.
- Y para cada vértice necesitamos añadir el atributo de la Normal.

Hoy veremos...

- 1 Introducción
- 2 **Modelo de Iluminación de Phong**
 - Luz Ambiente
 - Reflexión Difusa
 - Reflexión Especular
 - Materiales
 - El modelo de Phong
- 3 Tipos de Fuentes de Luz
- 4 Modelos de Sombreado
- 5 Casos prácticos

Modelo de Iluminación de Phong

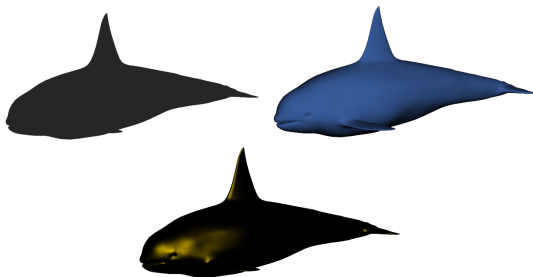
"In trying to improve the quality of the synthetic images, we do not expect to be able to display the object exactly as it would appear in reality, with texture, overcast shadows, etc. We hope only to display an image that approximates the real object closely enough to provide a certain degree of realism." – Bui Tuong Phong, 1975

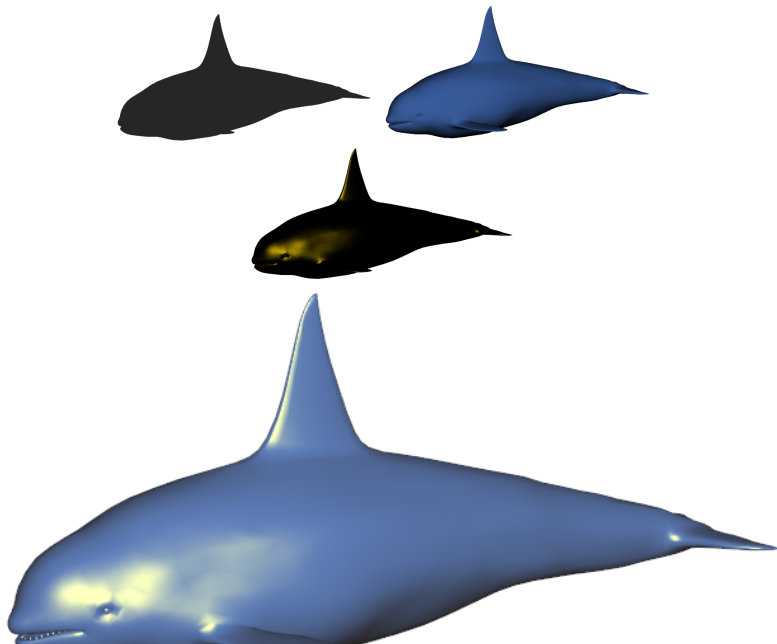


Modelo de Iluminación de Phong

Este modelo tiene en cuenta los tres siguientes aspectos

- **Luz ambiente:** luz que proporciona iluminación uniforme a lo largo de la escena.
- **Reflexión difusa:** luz reflejada por la superficie en todas las direcciones.
- **Reflexión especular:** luz reflejada por la superficie en una sola dirección o en un rango de ángulos muy cercano al ángulo de reflexión perfecta.



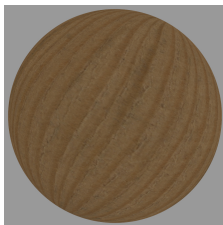


Luz Ambiente

Descripción

- La luz ambiente I_a que se observa en cualquier punto de una superficie es siempre la misma.
- Parte de la luz que llega a un objeto es absorbida por este, y parte es reflejada, y se modela con el coeficiente k_a , $0 \leq k_a \leq 1$.

$$I_a = k_a L_a \quad (1)$$

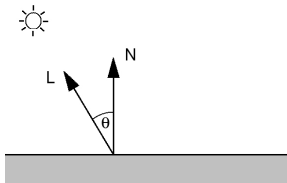
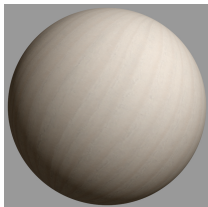


Reflexión Difusa

Descripción

- Es característica de superficies rugosas, mates, sin brillo.
- Se modela fácilmente con la Ley de Lambert. Así, el brillo observado en un punto depende sólo del ángulo θ , $0 \leq \theta \leq 90$, entre la dirección a la fuente de luz L y la normal N en dicho punto.
- L y N son vectores unitarios y k_d , $0 \leq k_d \leq 1$, representa la parte de luz difusa reflejada por la superficie.

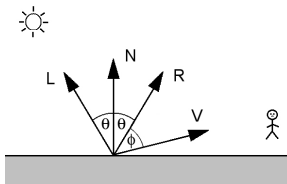
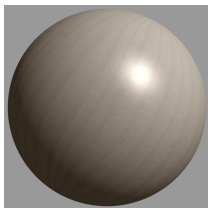
$$I_d = k_d L_d \cos \theta = k_d L_d (L \cdot N) \quad (2)$$



Reflexión Especular

Descripción

- Es propia de superficies brillantes, pulidas, y responsable de los brillos que suelen observarse.
- El color del brillo suele ser diferente del color de la superficie y muy parecido al color de la fuente de luz.
- Phong propone que la luz que llega al observador depende únicamente del ángulo Φ entre el vector de reflexión perfecta R y el vector dirección del observador V .



Reflexión Especular

Descripción

- Si R y V son vectores unitarios, k_s , $0 \leq k_s \leq 1$, representa la parte de luz especular reflejada por la superficie y α modela el brillo característico del material de la superficie.

$$I_s = k_s L_s \cos^\alpha \Phi = k_s L_s (R \cdot V)^\alpha \quad (3)$$

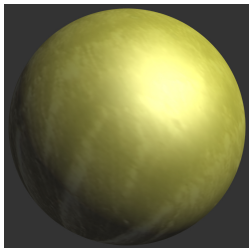
$$R = 2N(N \cdot L) - L \quad (4)$$

- Señalar que GLSL proporciona la función *reflect* que implementa el cálculo del vector R :
 - $R = \text{reflect}(I, N)$;donde I es el vector incidente, es decir, $I = -L$.

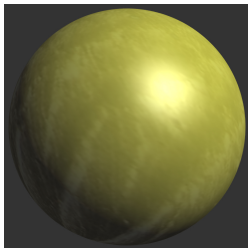
Reflexión Especular

El valor de α

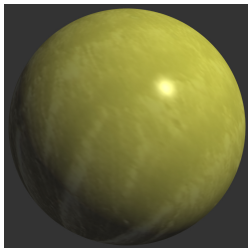
- Un valor igual a 1 modela un brillo grande.
- Valores mucho mayores, por ejemplo entre 100 y 500, modelan brillos más pequeños propios de materiales, por ejemplo, metálicos.



(a) $\alpha = 3$



(b) $\alpha = 10$

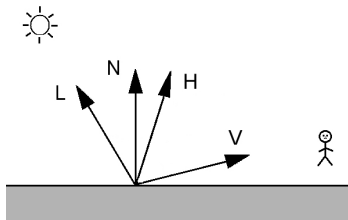


(c) $\alpha = 100$

Posible optimización

El vector intermedio H

$$H = L + V \quad (5)$$
$$I_s = k_s L_s (N \cdot H)^\alpha$$

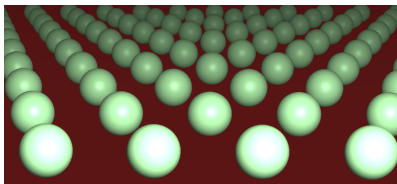


Atenuación

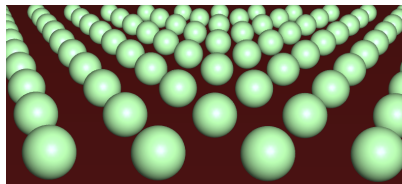
Atenuación

- Al viajar desde su fuente de origen hasta el objeto situado a una distancia d .
- Los coeficientes a , b y c son constantes características de la fuente de luz.

$$\frac{1}{a + bd + cd^2} (k_d L_d (L \cdot N) + k_s L_s (R \cdot V)^\alpha) \quad (6)$$



(d) Con atenuación



(e) Sin atenuación

Materiales

En el modelo de Phong

- Se tiene en cuenta las propiedades del material del objeto al calcular la iluminación y así proporcionar mayor realismo.
- En concreto son cuatro:
 - ambiente k_a ,
 - difusa k_d ,
 - especular k_s
 - y brillo α .
- Por ejemplo:

$$\text{Esmeralda: } k_a[] = \{ 0.022, 0.17, 0.02, 1.0 \}$$

$$k_d[] = \{ 0.08, 0.61, 0.08, 1.0 \}$$

$$k_s[] = \{ 0.63, 0.73, 0.63, 1.0 \}$$

$$\alpha[] = \{ 0.6 \}$$

Materiales

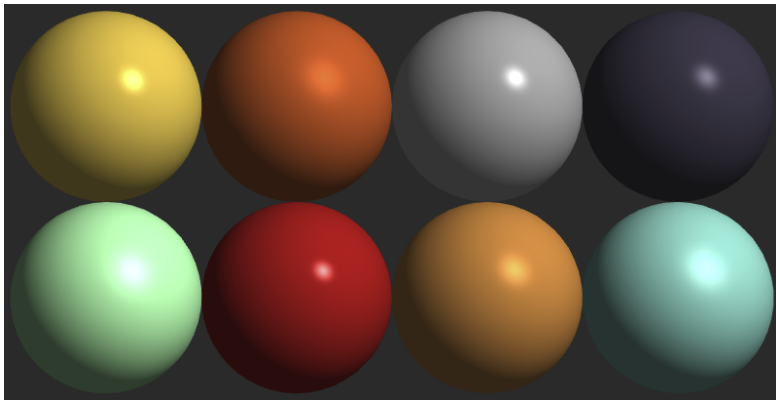


Figura: Ejemplos de materiales, de izquierda a derecha y de arriba a abajo: Oro, Cobre, Plata, Obsidiana, Jade, Rubí, Bronce, Turquesa.

El modelo de Phong

En resumen

$$I = k_a L_a + \frac{1}{a + bd + cd^2} (k_d L_d (L \cdot N) + k_s L_s (R \cdot V)^\alpha) \quad (7)$$

Listado 1: Función que implementa para una fuente de luz el modelo de iluminación de Phong sin incluir el factor de atenuación

```
// Ka, Kd, Ks, alpha, La, Ld y Ls son variables uniformes
// N, L y V se asumen normalizados
vec4 phong (vec3 N, vec3 L, vec3 V) {

    vec4 ambient = Ka * La;
    vec4 diffuse = vec4(0.0);
    vec4 specular = vec4(0.0);
    float NdotL = dot(N,L);
    if (NdotL > 0.0)
    {
        vec3 R = reflect(-L,N);
        float RdotV_n = pow(max(0.0, dot(R,V)), alpha);
        diffuse = NdotL * (Ld * Kd);
        specular = RdotV_n * (Ls * Ks);
    }
    return (ambient + diffuse + specular);
}
```

Ejercicios

- 1 Piensa cómo debería ser el modelo de Phong si hubiera más de una fuente de luz. Razona la respuesta.
- 2 En el modelo de iluminación de Phong, ¿en qué situaciones el producto escalar entre N y L resulta siempre un valor negativo?
- 3 ¿Qué problema piensas que puede aparecer al utilizar el modelo de Phong si en la escena hay objetos abiertos (como un vaso por ejemplo)?

Hoy veremos...

- 1 Introducción
- 2 Modelo de Iluminación de Phong
- 3 Tipos de Fuentes de Luz**
- 4 Modelos de Sombreado
- 5 Casos prácticos

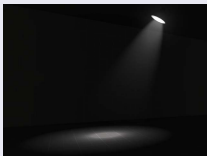
Tipos de Fuentes de Luz

Tipos

- Posicional: la fuente emite luz en todas las direcciones desde un punto dado, muy parecido a como por ejemplo ilumina una bombilla.
- Direccional: la fuente está ubicada en el infinito, todos los rayos de luz son paralelos y viajan en la misma dirección. En este caso el vector L en el modelo de iluminación de Phong es constante.

Foco de luz

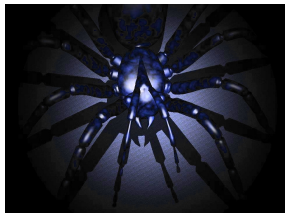
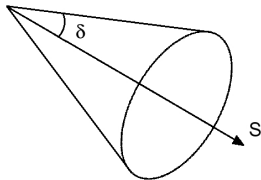
- Restringir los efectos de una fuente de luz posicional a un área limitada de la escena.



Foco de Luz

Cálculo

- Requiere una dirección S y un ángulo δ para la forma del cono.
- Un fragmento es iluminado si el ángulo entre el vector L y el vector S es menor que el ángulo δ .
- Atenuación: se calcula mediante el coseno del ángulo entre los vectores L y S elevado a un exponente. Cuanto mayor sea este exponente, mayor la concentración de luz alrededor del eje del cono.



Hoy veremos...

- 1 Introducción
- 2 Modelo de Iluminación de Phong
- 3 Tipos de Fuentes de Luz
- 4 Modelos de Sombreado**
- 5 Casos prácticos

Modelos de Sombreado

Recuerda ...

«Un modelo de iluminación determina el color de la superficie en un punto. Un modelo de sombreado utiliza un modelo de iluminación y especifica cuándo usarlo.»

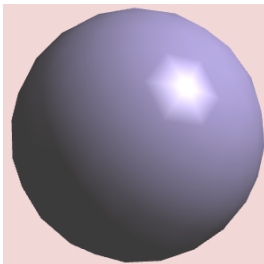
Métodos

- Plano: el modelo de iluminación se aplica una sola vez y su resultado se aplica a toda la superficie del polígono. Este método requiere la normal de cada polígono.
- Gouraud: el modelo de iluminación se aplica en cada vértice del polígono y los resultados se interpolan sobre su superficie. Este método requiere la normal en cada uno de los vértices del polígono.
- Phong: el modelo de iluminación se aplica para cada fragmento. Este método requiere la normal en el fragmento, que se puede obtener por interpolación de las normales de los vértices.

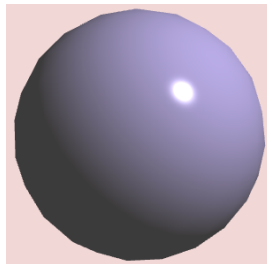
Ejemplos de sombreado



(a) Plano

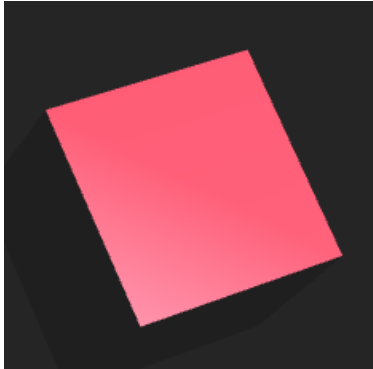


(b) Gouraud

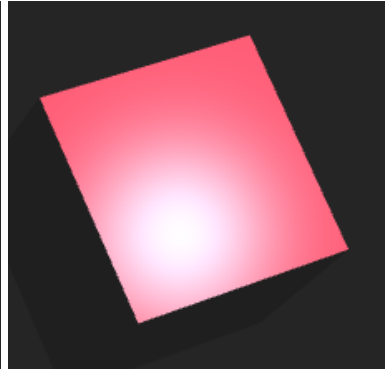


(c) Phong

Ejemplos de sombreado



(d) Gouraud



(e) Phong

Shader para sombreado de Gouraud

```
#version 300 es // Vertex Shader -----
uniform mat4  projectionMatrix , modelViewMatrix;
uniform mat3  normalMatrix;
uniform vec4  Ka, Kd, Ks;           // material
uniform float alfa;
uniform vec4  Lp, La, Ld, Ls;     // fuente de luz
in   vec3  vertexPosition , vertexNormal;
out  vec4  myColor;

void main()
{
    vec4 ecPosition = modelViewMatrix * vec4(vertexPosition ,1.0);
    vec3 N          = normalize(normalMatrix * vertexNormal);
    vec3 L          = normalize(vec3(Lp - ecPosition));
    vec3 V          = normalize(vec3(-ecPosition));

    myColor = phong(N, L, V);
    gl_Position = projectionMatrix * ecPosition;
}

#version 300 es // Fragment Shader -----
precision mediump float;
in   vec4 myColor;
out  vec4 fragmentColor;

void main()
{
    fragmentColor = myColor;
}
```


Shader para sombreado de Phong

```
#version 300 es // Vertex Shader -----
uniform mat4  projectionMatrix , modelViewMatrix;
uniform mat3  normalMatrix;
in           vec3  vertexPosition , vertexNormal;
out          vec3  Position , N;

void main()
{
    vec4  ecPosition = modelViewMatrix * vec4(vertexPosition ,1.0);
    Position = vec3(ecPosition);
    N        = normalize (normalMatrix * vertexNormal);

    gl_Position = projectionMatrix * ecPosition;
}

#version 300 es // Fragment Shader -----
uniform vec4  Ka, Kd, Ks;           // material
uniform float alfa;
uniform vec4  La, Ld, Ls, Lp;      // fuente de luz
in           vec3  Position , N;
out          vec4  fragmentColor;

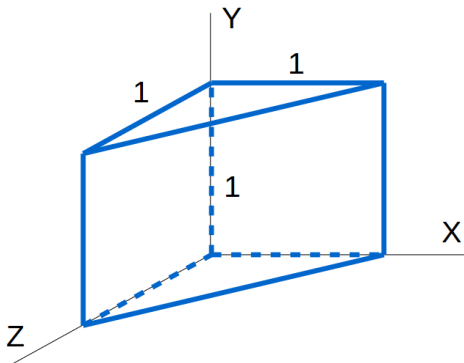
void main() {
    n = normalize(N);
    L = vec3 (normalize(Lp - Position));
    V = normalize (vec3(-Position));
    fragmentColor = phong (n, L, V);
}
```

Ejercicios

- 1 Observa en los listados anteriores el cálculo de los vectores N , L y V . ¿En qué situación se están calculando? ¿por qué?
- 2 A la hora de visualizar un triángulo de manera independiente, ¿cuántas veces se aplica el modelo de iluminación si se utiliza el modelo de sombreado de Gouraud?
- 3 Modifica el Shader que implementa el sombreado de Gouraud para utilizar una fuente de luz de tipo direccional.

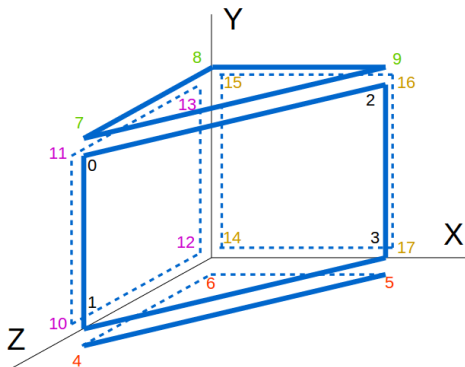
Ejercicio

Describe el prisma de la figura incluyendo para cada vértice dos atributos: posición y normal. Utiliza la estructura de vértices compartidos.



Procede por ejemplo así:

Separa las caras y enumera los vértices, y entonces averigua la normal de cada cara. Ahora ya puedes crear la lista de vértices y la de índices.



Hoy veremos...

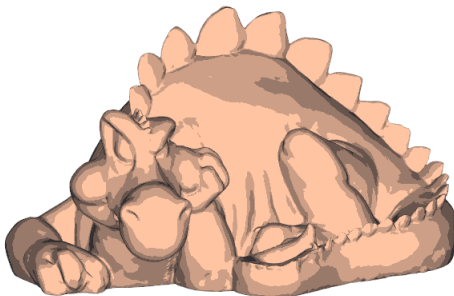
- 1 Introducción
- 2 Modelo de Iluminación de Phong
- 3 Tipos de Fuentes de Luz
- 4 Modelos de Sombreado
- 5 Casos prácticos
 - Sombreado Cómico
 - Niebla

Casos prácticos

Sombreado Cómic

Simula el aspecto típico en cómics haciendo que la componente difusa se reduzca a solo un número de posible valores (*levels*).

```
diffuse = (ceil (NdotL * levels) / levels) * (Light.Ld * Material.Kd);
```



Casos prácticos

Niebla

Mezcla el color del fragmento con el de la niebla y utiliza la distancia del fragmento a la cámara para decidir cómo repartir el peso:

- A mayor distancia, más peso para el color de la niebla.
- A menor distancia, más peso para el color del fragmento.

En cualquier caso, hay distancias de corte mínimo y máximo. Solo si el fragmento está entre ambas distancias habrá que calcular su color definitivo mediante interpolación (lineal vs exponencial).

