

Iluminación y sombreado 1

Modelo de iluminación Phong

El modelo de iluminación de Phong tiene en cuenta los tres aspectos siguientes:

- **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
- [WebGL - Phong Shading](#)

Modelo de iluminación Phong

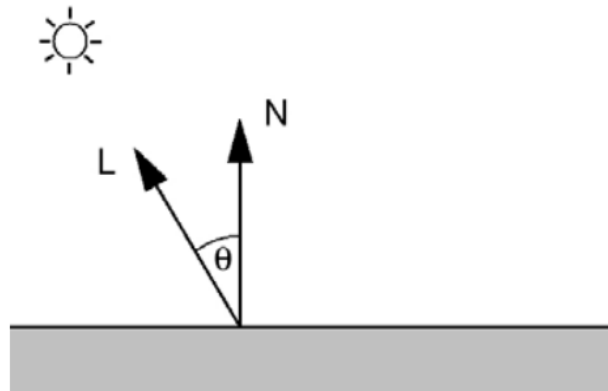
Luz ambiente:

- Es constante para toda la escena. La iluminación que se observa en cualquier punto de la superficie de un objeto debida a la luz ambiente sea siempre la misma: L_a
- Depende también del material del objeto: $0 \leq K_a \leq 1$
- $I_a = L_a * K_a$

Modelo de iluminación Phong

Reflexión difusa:

- La iluminación observada en un punto de la superficie de un objeto depende del ángulo θ , $0 \leq \theta \leq 90$, entre la dirección a la fuente de luz L y la normal N de la superficie en dicho punto
- Depende también del material del objeto: $0 \leq K_d \leq 1$



- $I_d = K_d * I_d(L \cdot N)$

Modelo de iluminación Phong

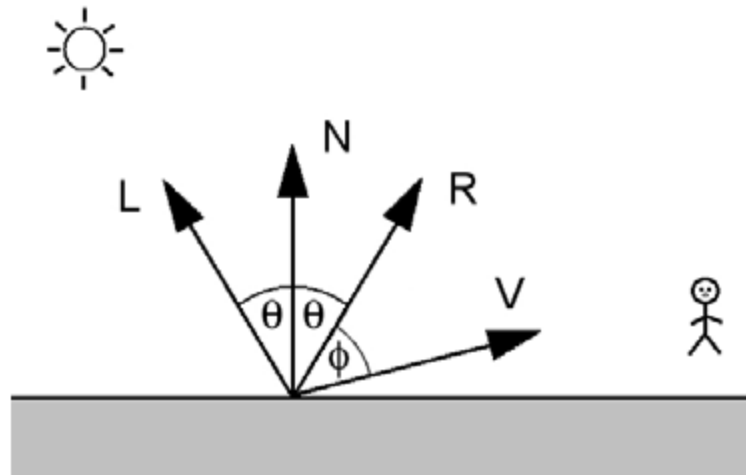
Reflexión especular:

- Este tipo de reflexión es propia de superficies brillantes, pulidas, y es responsable de los brillos que suelen observarse en esos tipos de superficies. Su principal característica es que este tipo de reflexión depende de la posición del observador
- Reflexion especular = Mat super * Cant Luz * (Reflex Perf . Dir Observador) elevado al tamaño del brillo

Modelo de iluminación Phong

Reflexión especular:

- Reflexion especular = Mat super * Cant Luz * (Reflex Perf . Dir Observador) elevado al tamaño del brillo



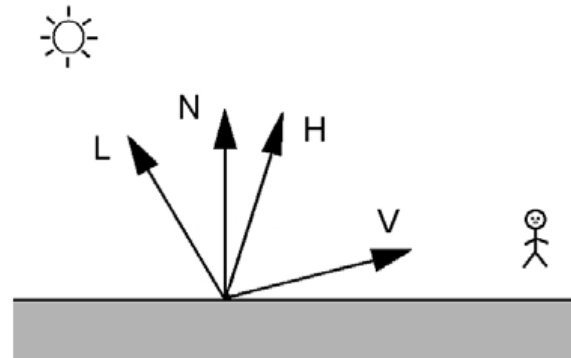
Modelo de iluminación Phong

Reflexión especular (versión pa pobres):

- Como calcular el vector R es caro computacionalmente, se propone calcular H en su lugar que es $L + V$.

$$H = L + V$$

$$I_s = k_s L_s (N \cdot H)^\alpha$$



Modelo de iluminación

Phong

Materiales

Esmeralda	k_a : [0.022, 0.17, 0.02] k_d : [0.08, 0.61, 0.08] k_s : [0.63, 0.73, 0.63] α : [0.6]
Obsidiana	k_a : [0.05, 0.05, 0.07] k_d : [0.18, 0.17, 0.23] k_s : [0.33, 0.33, 0.35] α : [0.30]
Rubí	k_a : [0.18, 0.01, 0.01] k_d : [0.61, 0.04, 0.04] k_s : [0.73, 0.63, 0.63] α : [0.60]
Bronce	k_a : [0.21, 0.13, 0.05] k_d : [0.71, 0.43, 0.18] k_s : [0.39, 0.27, 0.17] α : [0.20]
Oro	k_a : [0.25, 0.20, 0.07] k_d : [0.75, 0.61, 0.23] k_s : [0.63, 0.56, 0.37] α : [0.40]
Plástico	k_a : [0.0, 0.0, 0.0] k_d : [0.55, 0.55, 0.55] k_s : [0.70, 0.70, 0.70] α : [0.25]

Jade	k_a : [0.14, 0.22, 0.16] k_d : [0.54, 0.89, 0.63] k_s : [0.32, 0.32, 0.32] α : [0.10]
Perla	k_a : [0.25, 0.21, 0.21] k_d : [1.0, 0.83, 0.83] k_s : [0.30, 0.30, 0.30] α : [0.09]
Turquesa	k_a : [0.10, 0.19, 0.17] k_d : [0.39, 0.74, 0.69] k_s : [0.29, 0.31, 0.31] α : [0.10]
Cobre	k_a : [0.19, 0.07, 0.02] k_d : [0.71, 0.27, 0.08] k_s : [0.26, 0.14, 0.09] α : [0.10]
Plata	k_a : [0.20, 0.20, 0.20] k_d : [0.51, 0.51, 0.51] k_s : [0.51, 0.51, 0.51] α : [0.40]
Goma	k_a : [0.05, 0.05, 0.05] k_d : [0.50, 0.50, 0.50] k_s : [0.70, 0.70, 0.70] α : [0.08]

Modelo de iluminación Phong

Fuente de luz

```
program.LaIndex      = gl.getUniformLocation(program, "Light.La");  
program.LdIndex      = gl.getUniformLocation(program, "Light.Ld");  
program.LsIndex      = gl.getUniformLocation(program, "Light.Ls");  
program.PositionIndex = gl.getUniformLocation(program, "Light.Position");
```

```
gl.uniform3f(program.LaIndex,      1.0, 1.0, 1.0);  
gl.uniform3f(program.LdIndex,      1.0, 1.0, 1.0);  
gl.uniform3f(program.LsIndex,      1.0, 1.0, 1.0);  
gl.uniform3f(program.PositionIndex, 10.0, 10.0, 0.0);
```

Modelos de Sombreado

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. Dados un polígono y un modelo de iluminación, hay tres métodos para determinar el color de cada fragmento:

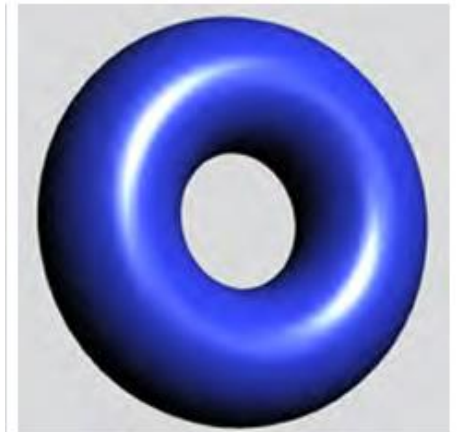
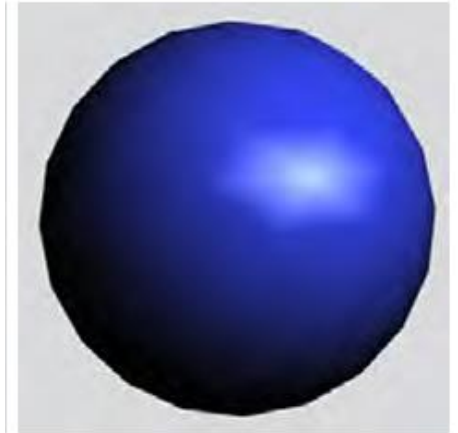
Modelos de Sombreado

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. Para obtener este tipo de sombreado en WebGL 2.0 hay que utilizar el calificador flat con la variable que almacene el color (tanto en el shader de vértices como en el de fragmentos) para evitar que precisamente se realice la interpolación del color por fragmento.



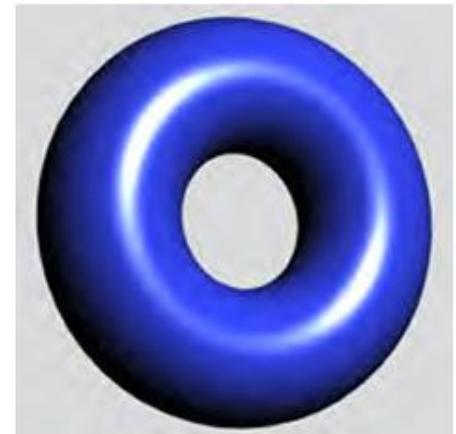
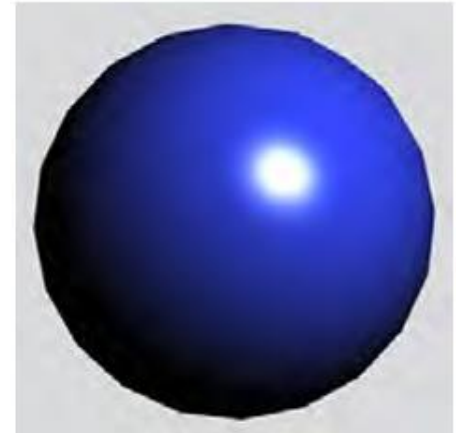
Modelos de Sombreado

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.



Modelos de Sombreado

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.



Modelo de iluminación Phong

Ejercicios:

1. Descarga phong_basics, ejecuta el html para ver qué se ve y comprueba dónde se ha añadido lo siguiente:
 - Método de iluminación phong.
 - Las normales de los vértices del modelo.
 - Cálculo e inicialización de la matriz de la normal (se usa para cada modelo).
 - Nueva función de dibujo que añade el nuevo atributo.
 - Obtención de referencia de materiales, el material usado y se establece su valor.

Modelo de iluminación Phong

Ejercicios:

2. ¿Qué cambios harías para implementar la optimización del vector intermedio? Piensa primero dónde implementar los cambios, luego crea un archivo a parte para implementarlos y ver que funciona como esperabas.

Modelo de iluminación Phong

Ejercicios:

3. ¿Qué tipo de sombreado se está aplicando y dónde?

Modelo de iluminación Phong

Ejercicios:

4. Implementa en otro archivo el método de sombreado Phong y en otro el método de sombrado Plano.