

Iluminación. Introducción.

`<a.soróa@ehu.eus>`

EHU

Iluminación

- En este tema vamos a hablar de la iluminación de la escena 3D.
 - luces.
 - materiales.
- Algoritmo de sombreado (*shading algorithm*):
 - Color de los objetos.
 - Reflejos luminosos.
 - Transparencia.
 - Texturas.
 - Sombras.

Iluminación

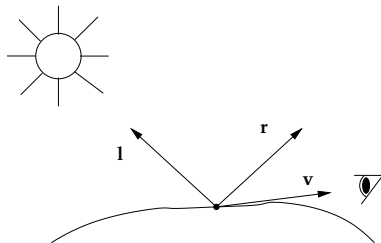
- Elementos necesarios para la iluminación:
 - Modelo óptico de la luz.
 - Geometría de los objetos.
 - Material asignado a cada objeto.
 - Luces.

Modelo óptico de la luz

- Simplificación matemática de las complejas inter-relaciones entre luz y objetos.
- Se distinguen dos modelos:
 - Modelo de iluminación
 - Cómo se refleja la luz en las superficies para producir lo que vemos como color.
 - Modelo de sombreado
 - Proceso de alterar el color de cada cara de los objetos 3D.

Modelo de iluminación

- Modela las complejas interacciones entre la luz y los objetos.
- Cómo reaccionan los objetos a la luz.

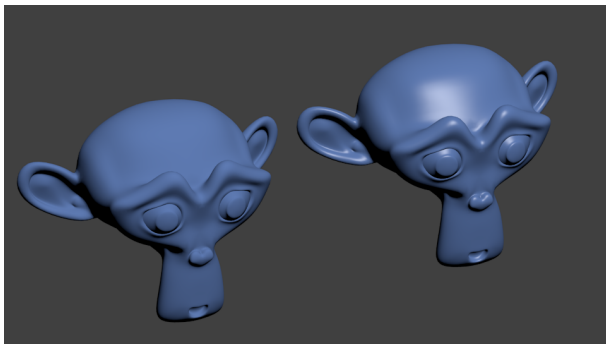


Modelo de iluminación

- Puede ser muy complicado:
 - la luz incidente puede venir de una fuente de luz o de reflejos de otros objetos.
 - a su vez, la superficie iluminada refleja luz a otros objetos.
 - multitud de interacciones entre la luz y la superficie.

Modelos básicos

- Dos modelos básicos:
 - Modelo de *Lambert*
 - Modelo de *Phong*
- La principal diferencia es que el modelo de *Phong* contiene reflejos especulares.



Lambert

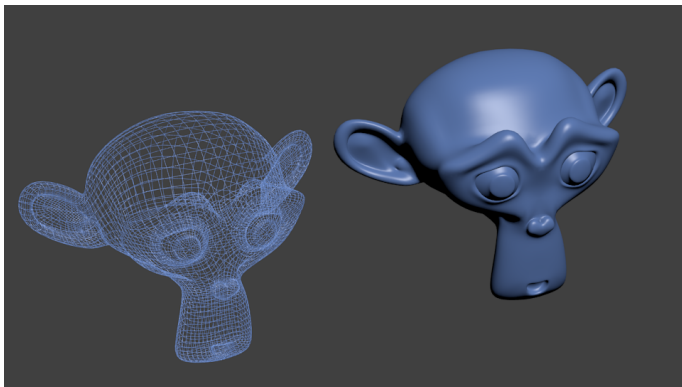
Phong

Modelo de sombreado

- Modelo poligonal.
- Los objetos son aproximaciones formadas por primitivas:
 - vértices, aristas, caras.
- El modelo de sombreado asigna colores a cada *pixel*.

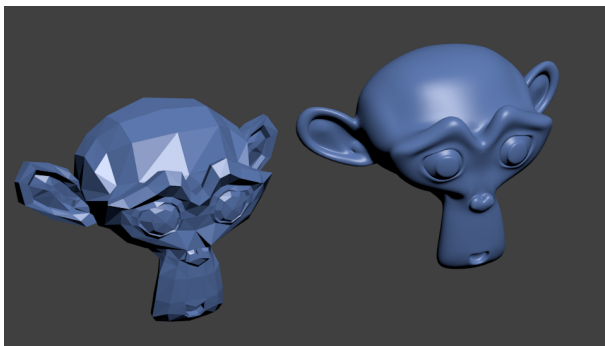
Modelo de sombreado

- Permite abstraer el modelo poligonal



Tipos básicos

- Dos tipos básicos de sombreado:
 - *Flat shading*: se interpola cada cara por separado.
 - *Smooth shading*: se interpola entre caras.

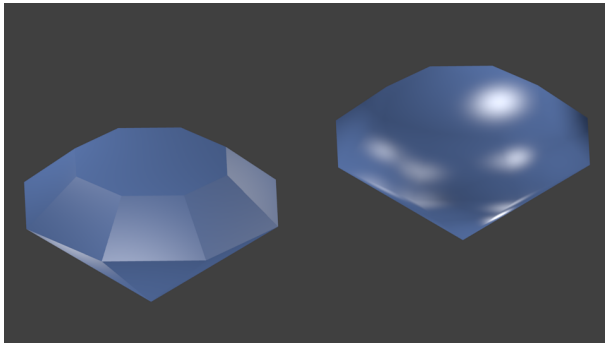


Flat

Smooth

Tipos básicos

- Dos tipos básicos de sombreado:
 - *Flat shading*: se interpola cada cara por separado.
 - *Smooth shading*: se interpola entre caras.



Flat

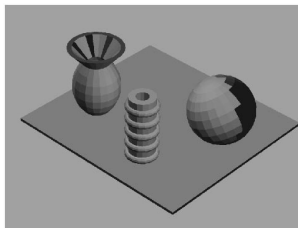
Smooth

También se distingue:

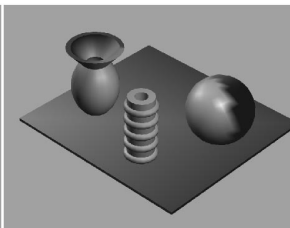
Dos modelos comunes:

- *Per vertex shading (Gouraud shading)*: los cálculos se hacen para cada vértice. Se interpolan los colores de los vértices.
- *Per fragment shading (Phong shading)*: Los cálculos se hacen para cada fragmento (*pixel*). Se interpolan las normales de los vértices.

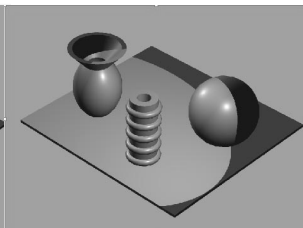
Flat, per vertex, per fragment



Flat



Per vertex (Gouraud)



Per pixel (Phong)

(Imágenes obtenidas de "Real time rendering")

Material

- Cada objeto tiene asignado un material.
- Diferente comportamiento con respecto a la luz.
- El material define también:
 - el modelo de iluminación.
 - el modelo de sombreado.

El objeto `THREE.Material`

- Se define los parámetros de material.
- Color del material: atributo `color`
- Modelo de sombreado: atributo `shading`
 - `THREE.FlatShading`, `THREE.SmoothShading`
- Modelo de alambre: atributo `wireframe`
- Diferentes tipos de materiales implementan diferentes modelos de iluminación:
 - `THREE.MeshBasicMaterial`, `THREE.MeshLambertMaterial`, `THREE.MeshPhongMaterial`, ...

Tipos de materiales

- `THREE.MeshBasicMaterial`: Material básico, asigna un color uniforme a todo el objeto.
- `THREE.MeshLambertMaterial`: Material de tipo Lambert.
- `THREE.MeshPhongMaterial`: Material de tipo Phong.
- Por ejemplo:

```
// color: rojo
// modelo de iluminacion: phong
// modelo de sombreado: flat
var material = new THREE.MeshPhongMaterial(
    { color: 0xFF0000
      shading: THREE.FlatShading });
```


THREE.Mesh

- Objetos básico en `Three.js`
- Dos parámetros fundamentales:
 - geometría (tipo derivado de `THREE.Geometry`)
 - material (tipo derivado de `THREE.Material`)
- Por ejemplo:

```
var geometry = new THREE.BoxGeometry(1, 1, 1); // Cubo de 1x1x1
var material = new THREE.MeshLambertMaterial(
    { color: 0xFF0000 } ); // Material de color rojo
var cube = new THREE.Mesh (geometry, material); // Crea el objeto
scene.add(cube); // insertar el objeto en la escena
```