

Materiales, Luces y sombras

MSC J. Fco. Jafet P. L.

Proyección de sombras (shadow casting)

Shadow Casting es una técnica que nos permite aproximarnos a la manera en la que vemos las proyecciones de sombras en el mundo real al emularlo en una escena 3D.

La proyección de sombras es un recurso eficaz para lograr escenas más realistas.

Proyectando sombras

Proyectar sombras en Three.js implica considerar la configuración de tres aspectos:

- El procesador que hace el cálculo
- Los objetos que recibe luces y sombras
- Las luces que proyectan sombras

Configurar el procesador

El procesador (Renderer) es el que va a calcular las posiciones sombras de la escena 3D. La proyección de sombras es un proceso bastante caro. Sólo WebGLRenderer lo soporta.

Utiliza Shadow mapping, una técnica específica para WebGL. El proceso es realizado directamente en la GPU.

Por defecto la proyección de sombras esta desactivada, para activarla debemos especificarlo de la siguiente manera:

```
renderizador.shadowMapEnabled = true;
```

Configurar los objetos

Existen dos parámetros utilizados para controlar la forma en que los objetos 3D interactúan con las luces y las sombras.

Si un objeto no permite pasar la luz estableceremos el parámetro **.castShadow** a true para que genere la sombra correspondiente.

Si se supone que un objeto recibirá sombras estableceremos el parámetro **.receiveShadow** a true para permitirlo.

Ambos parámetros tienen falso como valor por defecto.

```
objeto.castShadow = true;  
objeto.receiveShadow = false;
```

Configurar las luces

Existen en Three.js dos tipos de luces capaces de proyectar sombras:

Luces Direccionales (`THREE.DirectionalLight`)

Luces Puntuales (`THREE.SpotLight`)



Cuando los rayos de luz son paralelos tenemos una **luz direccional**. Podemos comprenderla pensando en ella como una fuente de luz muy lejos de nosotros (por ejemplo, el sol).

Cuando los rayos de luz parece provenir de un solo punto, y se extienden hacia el exterior en una dirección en forma de cono, hablamos de una **luz puntual**.



Para activar la proyección de sombras derivadas de una luz solo tenemos que indicarlo mediante el parámetro adecuado:

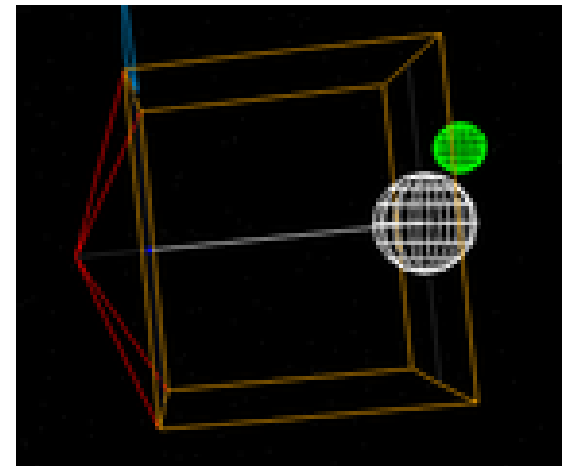
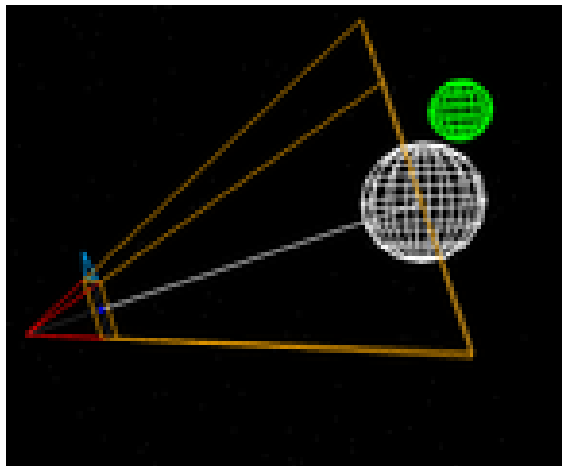
```
luz.castShadow = true;
```


Se puede ajustar la oscuridad sombra, es decir su opacidad. El parámetro **shadowDarkness** recibe un valor entre 0 y 1 para tal efecto. 0 significa que no hay sombra, 1 significa sombra negra pura.

Por ejemplo, podemos especificar media sombra así:

```
luz.shadowDarkness = 0.5;
```

Cuando se proyectan sombras con una luz direccional, es necesario trabajar con una cámara con proyección ortográfica. La proyección en perspectiva es la forma de ver las cosas en el mundo real. Por lo que nos parece más natural que la proyección ortográfica.



Texturas

MSC J. Fco. Jafet P. L.

Poner la textura en el material de un objeto 3D en Three.js implica considerar tres aspectos:

- La escena debe trabajar con luces y sombras.
- Se debe tener algún tipo de animación.
- La aplicación debe estar alojada en un servidor web.

Animación simple

Se utiliza la función:

`window.requestAnimationFrame()`

Animación simple

`window.requestAnimationFrame` informa al navegador que quieres realizar una animación y solicita que el navegador programe el repintado de la ventana para el próximo ciclo de animación. El método acepta como argumento una función a la que llamar antes de efectuar el repintado.

Animación simple

Sintaxis:

```
window.requestAnimationFrame( funciónDeAnimación );
```

Animación simple

Parámetros

funciónDeAnimación

Parámetro que especifica la función a la cual llamar llegado el momento de actualizar tu animación para el próximo repintado.

Animación simple

Debes llamar a este método cuando estés preparado para actualizar tu animación en la pantalla para pedir que se programe el repintado.

Esto puede suceder hasta 60 veces por segundo en pestañas en primer plano, pero se puede ver reducido a velocidades inferiores en pestañas en segundo plano.

El método indicado como `funciónDeAnimación` recibe un único argumento que indica el tiempo en el que está programado que se ejecute el ciclo de animación.

Animación simple

Nota: Si no quieres que tu animación se detenga, debes asegurarte de llamar a su vez a `requestAnimationFrame()` desde tu *funciónDeAnimación*.

Animación simple

Ejemplo:

```
animar();
```

```
function animar() {
```

```
    //Realizar aquí la animación
```

```
    requestAnimationFrame(animar);
```

```
    renderizador.render(escena, camara);
```

```
}
```

Cambiando el material de un objeto 3D para que acepte una textura

```
var material= new THREE.MeshBasicMaterial({color: 0xff0000});
```



```
var material=new THREE.MeshBasicMaterial({ map : THREE.ImageUtils.loadTexture('t.jpg') });
```