



Universidad Tecnológica de la Mixteca

Ingeniería en Computación

Grupo: 502

Programación Web I

Inicialización three.js

En este documento aprenderemos cómo trabajar con gráficos 3D usando la librería three.js

Three es una librería eficiente y ligera para generar y animar gráficos en 3D desde el navegador.

Para poder empezar a utilizarla, se debe importar la librería:

```
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  </head>
  <body>
    <script type="importmap">
      {
        "imports": {
          "three": "../build/three.module.js",
          "three/addons/": "../jsm/"
        }
      }
    </script>
    <script type="module">
      import * as THREE from 'three';
    </script>
  </body>
</html>
```

Crear una escena

Una escena es el escenario que se pretende visualizar y se le puede agregar un fondo.

```
const scene = new THREE.Scene();
scene.background = new THREE.Color(0xBFDD1E5)
```

Una escena a su vez puede incluir la cámara, dibujos 3D, luces, etc. Para que algo pueda visualizarse (renderizarse) deberá estar contenido en la escena.

La cámara permite indicar la parte que se va a renderizar de la escena, es decir, sólo la parte que abarca la cámara se podrá visualizar.

```
const camera = new THREE.PerspectiveCamera(30, window.innerWidth / window.innerHeight, 0.1, 1000);
camera.position.set(0,90,150)
camera.lookAt(new THREE.Vector3(0,0,0))
```

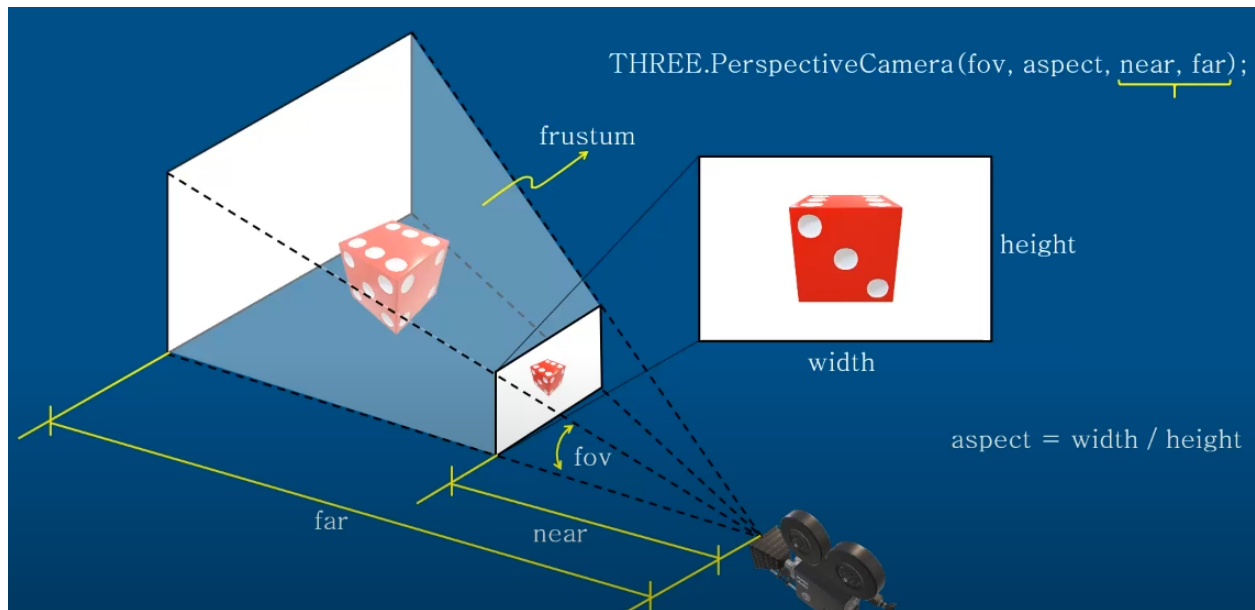
La función PerspectiveCamera recibe 4 parámetros:

Fov: campo de vista.

Aspect: relación entre el ancho y el alto.

Near: Distancia de la cámara al plano más cercano.

Far: Distancia de la cámara al plano más alejado.



Dibujando un cubo

Para dibujar un cubo se utilizará un mesh o malla, la cuál tiene 2 partes, la geometría y el material. La geometría define la forma que va a tener la malla (vértices, aristas y posición) y el material indica el color que tendrá.

```
const anchoCubo = 1;
const altoCubo = 1;
const profundidadCubo = 1;
const geometry = new THREE.BoxGeometry(anchoCubo, altoCubo, profundidadCubo);
const material = new THREE.MeshPhongMaterial({ color: 0x44aa88 });
const cube = new THREE.Mesh(geometry, material);
scene.add(cube);
```

```
renderer.render(scene, camera);
```

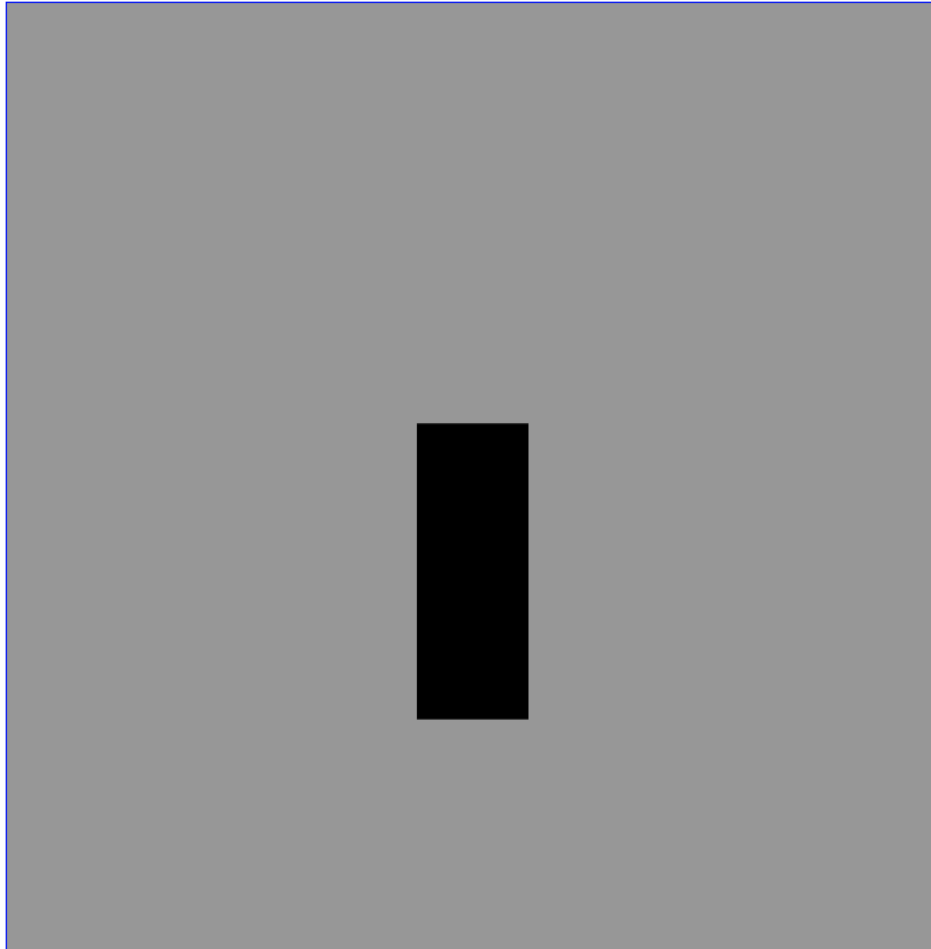


Fig. 2: Dibujando un cubo

El cubo se puede ver negro, esto es porque no se ha puesto ninguna luz.

```
const color = 0xffffff;  
const intensity = 1;  
const light = new THREE.DirectionalLight(color, intensity);  
light.position.set(-1, 2, 4);  
scene.add(light);
```

Se crea una luz de color blanco con intensidad 1 y de tipo direccional, inicialmente se posiciona en la coordenada 0,0,0, pero esto quedaría dentro del cubo y seguiría viéndose de color negro. Para poder visualizar el color del material que se asignó, es necesario mover la luz.

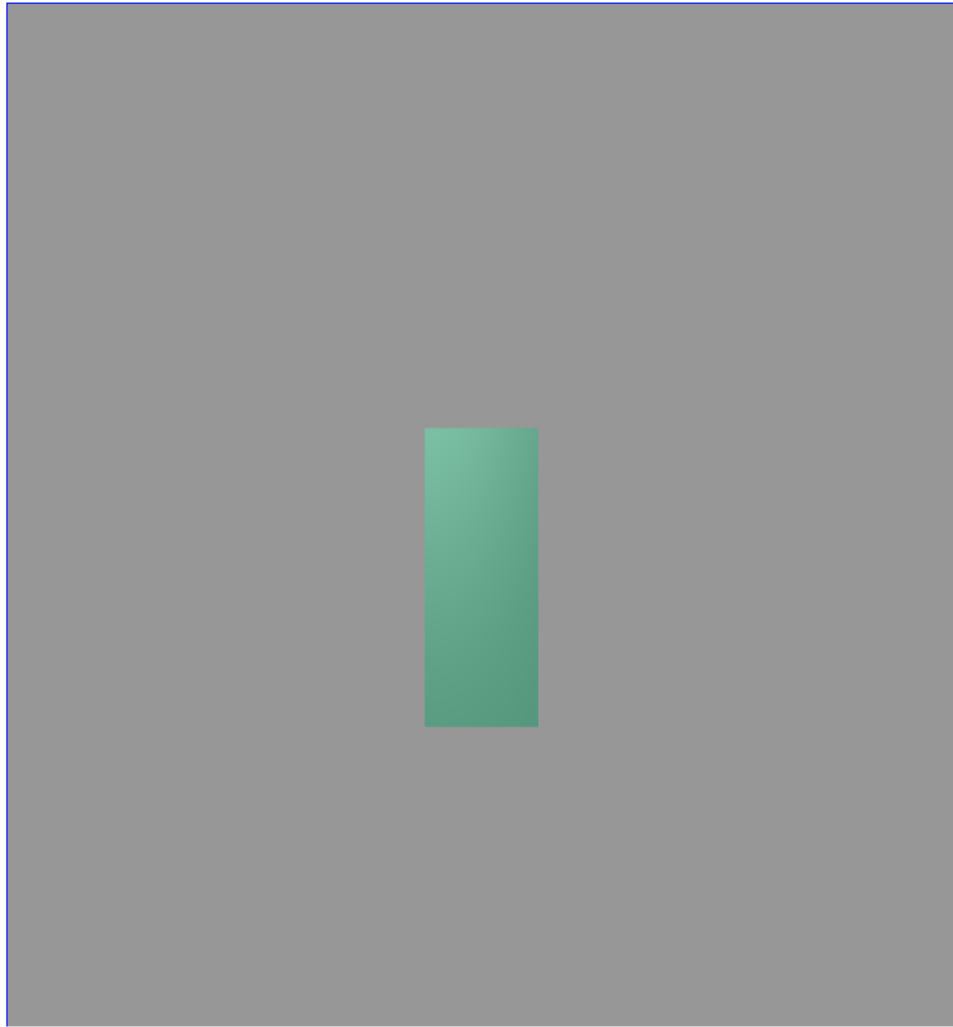


Fig. 3: Dibujando un cubo con luz

Finalmente se rota el cubo.

```
cube.rotation.x = 35;  
cube.rotation.y = 45;
```

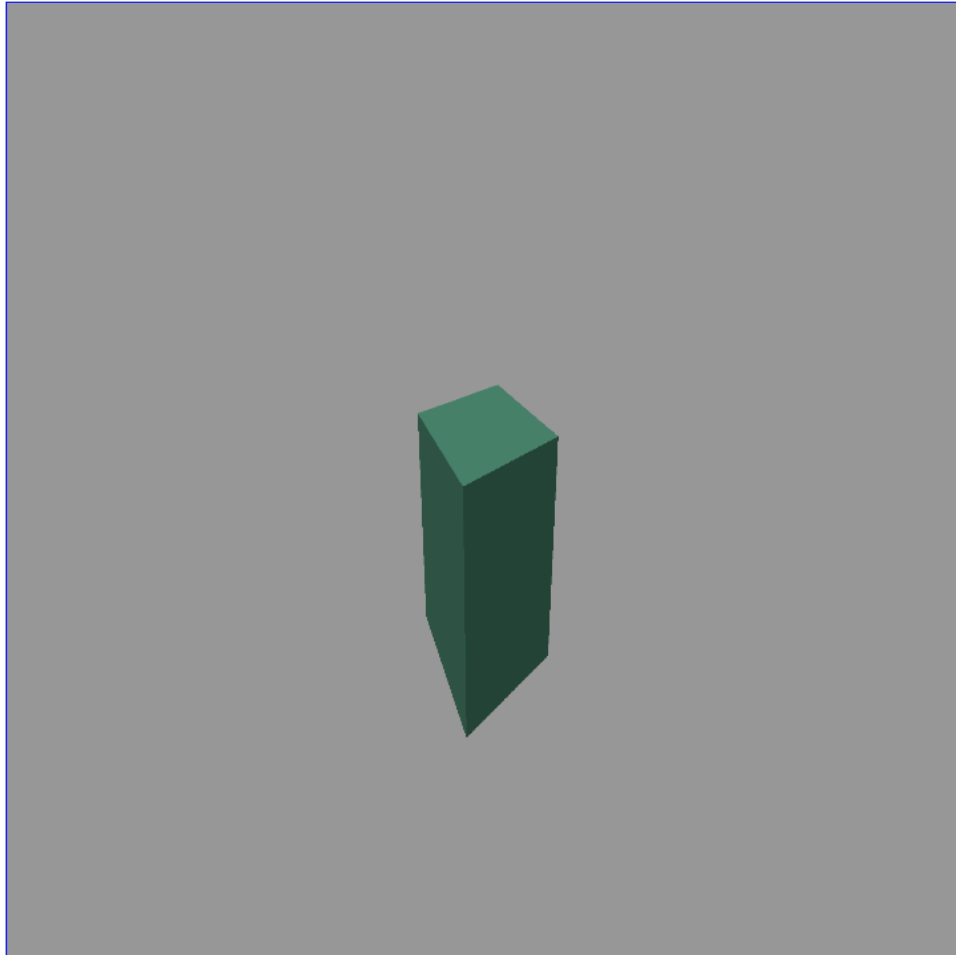


Fig. 4: Dibujando un cubo con luz y rotado

Para poder quitar los picos de la figura:

```
const renderer = new THREE.WebGLRenderer({ canvas, antialias: true });
```

Dibujando una esfera:

```
var esferaGeometry = new THREE.SphereGeometry(1, 20, 20);  
var esferaMaterial = new THREE.MeshBasicMaterial({  
  color: 0x7777FF,  
  wireframe: false
```

```
});  
var esfera = new THREE.Mesh(esferaGeometry, esferaMaterial);  
esfera.position.set(1, 1, 1);  
scene.add(esfera);
```

Agregando otro cubo:

```
const cube2 = new THREE.Mesh(geometry, material);  
scene.add(cube2);  
cube2.rotation.x = 35;  
cube2.rotation.y = 45;  
cube2.position.set(3, 1, 1);
```

Cambiando de material a la esfera:

```
var esferaMaterial = new THREE.MeshLambertMaterial({  
    color: 0xffff00,  
    wireframe: false  
});
```

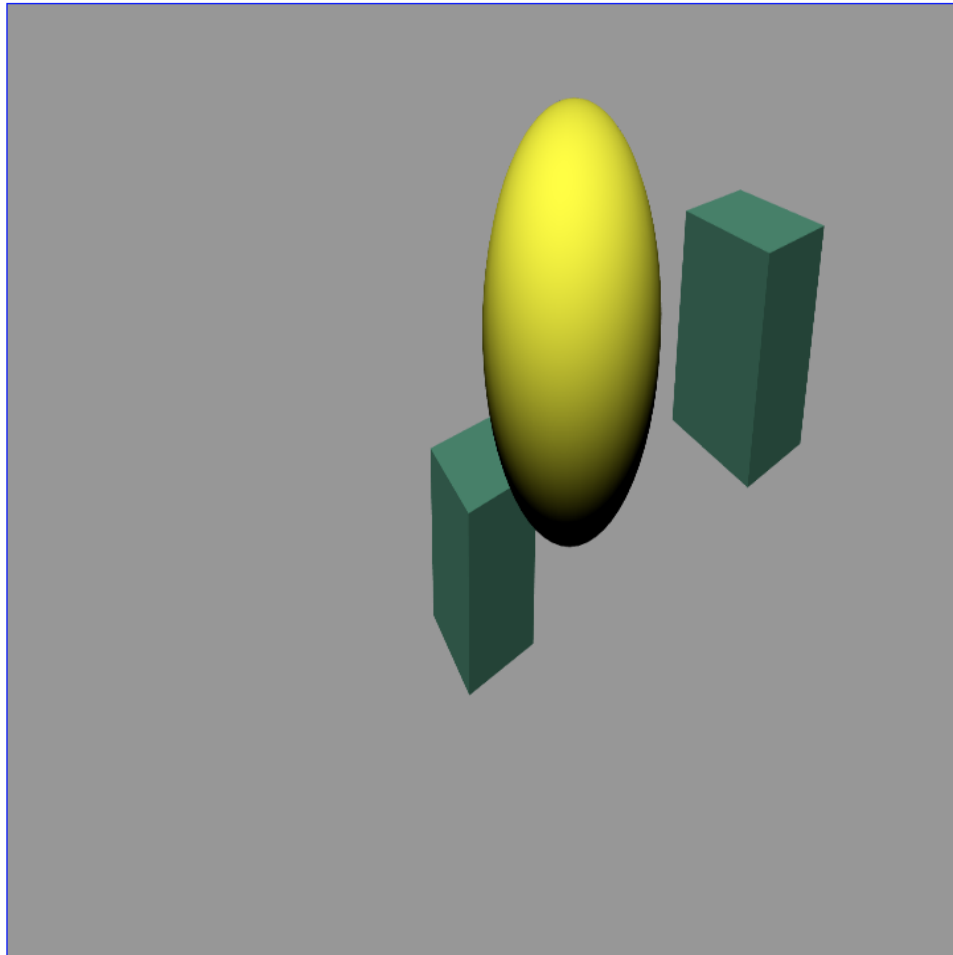


Fig. 5: Dibujando dos cubos y una esfera.

Dibujando un cilindro:

```
const geometry3 = new THREE.CylinderGeometry( 1, 2, 1, 15 );
const material3 = new THREE.MeshLambertMaterial( {color: 0xffff00} );
const cylinder = new THREE.Mesh( geometry3, material3 );
cylinder.position.set(3, -2, -6);
scene.add( cylinder );
```

CylinderGeometry: El primer parámetro es el radio de arriba, el segundo parámetro es el radio de abajo, el tercer parámetro es la altura y el último parámetro es el número de segmentos radiales.

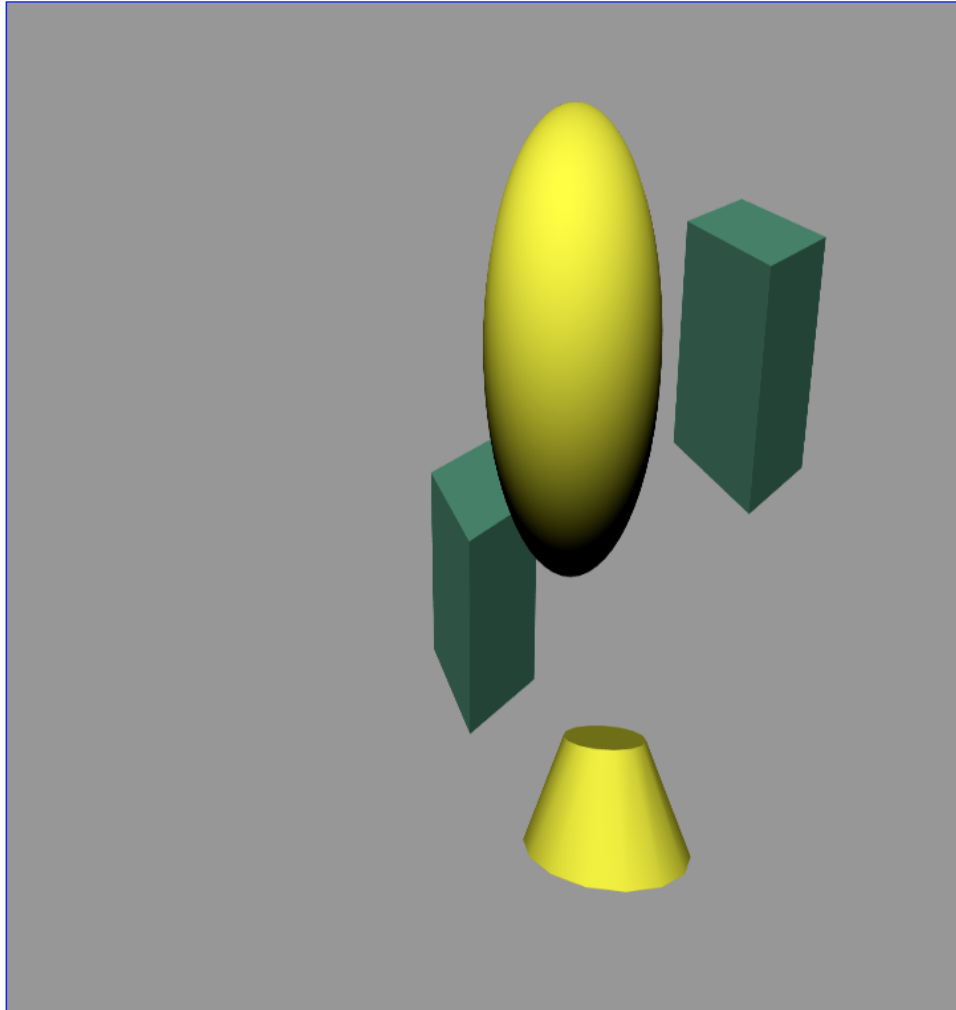


Fig. 6: Dibujando dos cubos, una esfera y un cilindro.

Controlando la escena

```
npm install three-orbitcontrols
```

```
npm i --save-dev @types/three`
```

Permite a la cámara girar alrededor de un objetivo.

```
const controls = new OrbitControls(camera, canvas)
renderer.render(scene, camera);
function animate()
{
  requestAnimationFrame(animate)
  controls.update();

  renderer.render(scene, camera)
}
animate()
```

Agregando Ejes

```
const axesHelper = new THREE.AxesHelper( 20 );
scene.add( axesHelper );
```

Agregando Ejes a un objeto

```
var esferaEjes = new THREE.AxesHelper(20);
esfera.add(esferaEjes);
```