

1. Introducción

El código utiliza Three.js para crear una simulación del sistema solar, permitiendo visualizar los planetas y sus lunas en órbita alrededor del sol. La escena está animada, con controles de cámara que permiten explorar el entorno de manera interactiva.

2. Configuración e Inicialización

El código comienza creando la escena, la cámara y el renderizador. También se crea un elemento DOM para visualizar información sobre la escena.

```
function init() {  
  info = document.createElement("div");  
  info.style.position = "absolute";  
  info.style.top = "30px";  
  info.style.width = "100%";  
  info.style.textAlign = "center";  
  info.style.color = "#fff";  
  info.style.fontWeight = "bold";  
  info.style.backgroundColor = "transparent";  
  info.style.zIndex = "1";  
  info.style.fontFamily = "Monospace";  
  info.innerHTML = "Solar system";  
  document.body.appendChild(info);  
}
```

3. Configuración de la Cámara

Se definen dos cámaras: cámara y cámara2. La primera se utiliza para la visualización principal, mientras que la segunda se emplea para el modo en primera persona.

```
camera = new THREE.PerspectiveCamera(  
  75,  
  window.innerWidth / window.innerHeight,  
  0.1,  
  1000  
);  
camera.position.set(0, 0, 20);  
  
camera2 = new THREE.PerspectiveCamera(  
  75,  
  window.innerWidth / window.innerHeight,  
  3,  
  1000  
);  
camera2.position.set(0, 0, 50);
```

4. Iluminación

La escena está iluminada por una luz ambiental y una luz direccional.

```
const Lamb = new THREE.AmbientLight(0xffffff, 0.75);  
scene.add(Lamb);  
  
const Ldir = new THREE.DirectionalLight(0xcccccc, 1)  
Ldir.position.set(5, 5, 5)  
scene.add(Ldir)  
Ldir.castShadow = true
```

5. Textura

El código carga varias texturas para el sol y los planetas utilizando THREE.TextureLoader().

```
//textures sun  
const text_sun = new THREE.TextureLoader().load("https://cdn.glitch.global/f122d42a-c38c-45cd-98bb-68f28cc345a8/8k_sun.jp  
const sun_bump = new THREE.TextureLoader().load("https://cdn.glitch.global/f122d42a-c38c-45cd-98bb-68f28cc345a8/sun_bump.  
const sun_light = new THREE.TextureLoader().load("https://cdn.glitch.global/f122d42a-c38c-45cd-98bb-68f28cc345a8/sunmap.j  
  
//textures mercury  
const text_mercury = new THREE.TextureLoader().load("https://cdn.glitch.global/f122d42a-c38c-45cd-98bb-68f28cc345a8/8k_me  
const mercury_bump = new THREE.TextureLoader().load("https://cdn.glitch.global/f122d42a-c38c-45cd-98bb-68f28cc345a8/mercu  
  
//textures venus  
const text_venus = new THREE.TextureLoader().load("https://cdn.glitch.global/f122d42a-c38c-45cd-98bb-68f28cc345a8/8k_venu  
const venus_bump = new THREE.TextureLoader().load("https://cdn.glitch.global/f122d42a-c38c-45cd-98bb-68f28cc345a8/venusbu  
const venus_atm = new THREE.TextureLoader().load("https://cdn.glitch.global/f122d42a-c38c-45cd-98bb-68f28cc345a8/4k_venus
```

6. Creación de los Objetos

Se crean funciones para generar estrellas, planetas y lunas. Cada función acepta parámetros como tamaños, colores y texturas.

```
function Luna(padre, radio, dist, vel, col, f1, f2, texture = undefined) {
  let geometry = new THREE.SphereBufferGeometry(radio, 32, 32);
  let material = new THREE.MeshPhongMaterial({ color: col });
```

7. Ciclo de Animación

El ciclo de animación actualiza continuamente la posición de los planetas y las lunas en función del tiempo, creando un efecto de órbita.

```
function animationLoop() {
  timestamp = (Date.now() - t0) * accglobal;

  requestAnimationFrame(animationLoop);

  //Modifica rotación de todos los objetos en Planetas
  Planetas.forEach(function (planeta) {
    planeta.position.x =
      Math.cos(timestamp * planeta.userData.speed) * planeta.userData.dist * planeta.userData.f1;
    planeta.position.z =
      Math.sin(timestamp * planeta.userData.speed) * planeta.userData.dist * planeta.userData.f2;

    planeta.rotation.y += 0.01;
  });
```

8. Controles

El código utiliza controles en primera persona para permitir al usuario navegar a través de la escena.

```
// Inizializza i controlli
controls = new FirstPersonControls(camera2, renderer.domElement);
controls.movementSpeed = 5; // Velocità di movimento
controls.lookSpeed = 0.1; // Velocità di rotazione della camera
controls.lookVertical = true; // Abilita la rotazione verticale
controls.autoForward = false; // Disabilita il movimento automatico in avanti
controls.activeLook = false; // Abilita il movimento con il mouse
```

9. Botón

Gestión del evento clic del ratón para mostrar u ocultar el pop-up con las instrucciones.

```
//Manejador de evento del botón de ratón
document.getElementById("instr").addEventListener("click",
  function(){
    if (popup.style.display === 'none' || popup.style.display === '') {
      popup.style.display = 'block'; // Mostra il pop-up
    } else {
      popup.style.display = 'none'; // Nascondi il pop-up
    }
  });
```