

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title> 🌌 3D Universe Explorer</title>
  <style>
    body, html {
      margin: 0;
      padding: 0;
      overflow: hidden;
      background: black;
      font-family: Arial, sans-serif;
      color: white;
    }
    #info-panel {
      position: absolute;
      top: 10px;
      left: 10px;
      background: rgba(0,0,0,0.6);
      padding: 15px;
      border-radius: 10px;
      max-width: 320px;
    }
    h2 { margin: 0 0 10px; }
    canvas { display: block; }
  </style>
</head>
<body>
  <div id="info-panel">
    <h2> 🌍 3D Universe Explorer</h2>
    <p>Explore the Solar System in 3D.<br>
      Use mouse/touch to orbit, zoom, and pan.</p>
  </div>

  <!-- Three.js -->
  <script src="https://cdn.jsdelivr.net/npm/three@0.160.0/build/three.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/three@0.160.0/examples/js/controls/OrbitControls.js">
</script>

  <script>
    const scene = new THREE.Scene();

```

```
const camera = new THREE.PerspectiveCamera(60, window.innerWidth/window.innerHeight,
0.1, 2000);
camera.position.set(0, 50, 150);

const renderer = new THREE.WebGLRenderer({ antialias: true });
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);

const controls = new THREE.OrbitControls(camera, renderer.domElement);
controls.enableDamping = true;

// Lights
const pointLight = new THREE.PointLight(0xffffff, 2, 1000);
pointLight.position.set(0, 0, 0);
scene.add(pointLight);
scene.add(new THREE.AmbientLight(0x333333));

// Sun
const sun = new THREE.Mesh(
  new THREE.SphereGeometry(10, 64, 64),
  new THREE.MeshBasicMaterial({ color: 0xffcc00 })
);
scene.add(sun);

// Planet textures
const loader = new THREE.TextureLoader();
const textures = {
  mercury: loader.load("https://raw.githubusercontent.com/raj-patra/planet-
textures/main/mercury.jpg"),
  venus: loader.load("https://raw.githubusercontent.com/raj-patra/planet-
textures/main/venus.jpg"),
  earth: loader.load("https://raw.githubusercontent.com/raj-patra/planet-
textures/main/earth.jpg"),
  mars: loader.load("https://raw.githubusercontent.com/raj-patra/planet-
textures/main/mars.jpg"),
  jupiter: loader.load("https://raw.githubusercontent.com/raj-patra/planet-
textures/main/jupiter.jpg"),
  saturn: loader.load("https://raw.githubusercontent.com/raj-patra/planet-
textures/main/saturn.jpg"),
  uranus: loader.load("https://raw.githubusercontent.com/raj-patra/planet-
textures/main/uranus.jpg"),
  neptune: loader.load("https://raw.githubusercontent.com/raj-patra/planet-
textures/main/neptune.jpg"),
```

```
    moon: loader.load("https://raw.githubusercontent.com/raj-patra/planet-  
textures/main/moon.jpg"),  
    stars:  
loader.load("https://raw.githubusercontent.com/mrdoob/three.js/master/examples/textures/gala  
xy_starfield.png")  
};
```

```
// Starfield
```

```
const starField = new THREE.Mesh(  
    new THREE.SphereGeometry(1000, 64, 64),  
    new THREE.MeshBasicMaterial({ map: textures.stars, side: THREE.BackSide })  
);  
scene.add(starField);
```

```
// Planet factory
```

```
function createPlanet(size, texture, distance, speed) {  
    const planet = new THREE.Mesh(  
        new THREE.SphereGeometry(size, 32, 32),  
        new THREE.MeshStandardMaterial({ map: texture })  
    );  
    planet.userData = { distance, angle: Math.random()*Math.PI*2, speed };  
    scene.add(planet);  
    return planet;  
}
```

```
// Moons
```

```
function createMoon(size, texture, distance, speed, parent) {  
    const moon = new THREE.Mesh(  
        new THREE.SphereGeometry(size, 32, 32),  
        new THREE.MeshStandardMaterial({ map: texture })  
    );  
    moon.userData = { distance, angle: Math.random()*Math.PI*2, speed, parent };  
    scene.add(moon);  
    return moon;  
}
```

```
// Planets
```

```
const planets = {  
    mercury: createPlanet(1, textures.mercury, 15, 0.02),  
    venus: createPlanet(1.5, textures.venus, 22, 0.015),  
    earth: createPlanet(2, textures.earth, 30, 0.01),  
    mars: createPlanet(1.2, textures.mars, 38, 0.008),  
    jupiter: createPlanet(5, textures.jupiter, 50, 0.006),  
    saturn: createPlanet(4.5, textures.saturn, 65, 0.005),
```

```

    uranus: createPlanet(3.5, textures.uranus, 80, 0.004),
    neptune: createPlanet(3.4, textures.neptune, 95, 0.003)
};

// Saturn's rings
const ringGeometry = new THREE.RingGeometry(6, 8, 64);
const ringMaterial = new THREE.MeshStandardMaterial({
    color: 0xd2b48c, side: THREE.DoubleSide, transparent: true, opacity: 0.7
});
const saturnRings = new THREE.Mesh(ringGeometry, ringMaterial);
saturnRings.rotation.x = Math.PI/2;
planets.saturn.add(saturnRings);

// Moons
const moons = [
    createMoon(0.5, textures.moon, 3, 0.05, planets.earth), // Earth's moon
    createMoon(0.8, textures.moon, 8, 0.03, planets.jupiter), // simple Jupiter moon
];

// Animate
function animate() {
    requestAnimationFrame(animate);

    sun.rotation.y += 0.002;

    for (let key in planets) {
        const planet = planets[key];
        planet.userData.angle += planet.userData.speed;
        planet.position.set(
            planet.userData.distance * Math.cos(planet.userData.angle),
            0,
            planet.userData.distance * Math.sin(planet.userData.angle)
        );
        planet.rotation.y += 0.01;
    }

    moons.forEach(moon => {
        moon.userData.angle += moon.userData.speed;
        const parent = moon.userData.parent;
        moon.position.set(
            parent.position.x + moon.userData.distance * Math.cos(moon.userData.angle),
            0,
            parent.position.z + moon.userData.distance * Math.sin(moon.userData.angle)
        );
    });
}

```

```
});
```

```
controls.update();
```

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

```
}
```

```
animate();
```

```
// Resize
```

```
window.addEventListener("resize", () => {
```

```
    camera.aspect = window.innerWidth / window.innerHeight;
```

```
    camera.updateProjectionMatrix();
```

```
    renderer.setSize(window.innerWidth, window.innerHeight);
```

```
});
```

```
</script>
```

```
</body>
```

```
</html>
```