<https://jsbin.com/rogecin/edit?js,output>

**CSS**:

#myCanvas {

width: 100% !important;

height: auto !important;

background-image: radial-gradient(#3d4040 , #564786);

}

**HTML**:

<html>

<head>

<title>Earthandmoon</title>

</head>

<body>

<canvas id="myCanvas"></canvas>

<script src="https://cdnjs.cloudflare.com/ajax/libs/three.js/109/three.min.js"></script>

<script src="https://unpkg.com/three@0.85.0/examples/js/controls/OrbitControls.js"></script>

<script src="https://unpkg.com/three@0.85.0/examples/js/controls/TransformControls.js"></script>

<script type="text/javascript"></script>

<canvas id="myCanvas"></canvas>

</body>

</html>

**JavaScript**:

window.addEventListener( 'load', init );

function init() {

let width = 580

, height = 410

, rot = 0;

/\*\*

\* render

\*\*/

const renderer = new THREE.WebGLRenderer( {

canvas: document.querySelector( '#myCanvas' ),

alpha: true

} );

renderer.setPixelRatio( window.devicePixelRatio );

renderer.setSize( width, height );

renderer.shadowMap.enabled = true;

const scene = new THREE.Scene();

const clock = new THREE.Clock();

/\*\*

\* camera

\*\*/

const camera = new THREE.PerspectiveCamera( 45, width / height, 0.1, 10000 );

camera.position.set( 400, 350, 5000 );

/\*\*

\* controls

\*\*/

const controls = new THREE.OrbitControls(camera, document.body);

/\*\*

\* earth

\*\*/

const e\_Geometry = new THREE.SphereGeometry(150, 60, 60 );

const e\_texture = new THREE.TextureLoader().load('https://upload.wikimedia.org/wikipedia/commons/c/cf/WorldMap-A\_non-Frame.png');

const e\_materials = new THREE.MeshStandardMaterial( { color: 0xaaaaaa, specular: 0x292929, shininess: 15, map:e\_texture } );

const earth = new THREE.Mesh(e\_Geometry, e\_materials );

scene.add(earth);

earth.receiveShadow = true;

earth.castShadow = true;

/\*\*

\* moon

\*\*/

const m\_Geometry = new THREE.SphereGeometry( 40, 60, 60 );

const m\_texture = new THREE.TextureLoader().load('https://upload.wikimedia.org/wikipedia/commons/d/db/Moonmap\_from\_clementine\_data.png');

const m\_materials = new THREE.MeshStandardMaterial( { color: 0xaaaaaa, specular: 0x292929, shininess: 12, map:m\_texture } );

const moon = new THREE.Mesh( m\_Geometry, m\_materials );

scene.add(moon);

/\*\*

\* fragment

\*\*/

createStarField();

function createStarField() {

const geometry = new THREE.SphereBufferGeometry(4, 3, 4),

size = 1;

for (let i = 0; i < 2000; i++) {

const material = new THREE.MeshPhongMaterial({

color: 0xffffff,

flatShading: true

})

/\* const mesh = new THREE.Mesh(geometry, material)

mesh.position.set(size \* Math.random() - 0.5, size \* Math.random() - 0.5, size \* Math.random() - 0.5).normalize()

mesh.position.multiplyScalar(Math.random() \* 800)

mesh.rotation.set(Math.random() \* 2, Math.random() \* 2, Math.random() \* 2)

mesh.scale.x = mesh.scale.y = mesh.scale.z = Math.random() \* 2

scene.add(mesh)\*/

}

}

/\*\*

\* light

\*\*/

var light = new THREE.AmbientLight( 0x404040 )

var light = new THREE.DirectionalLight( 0xffffff, 1, 100)

light.position.set(100, 1, 100)

light.castShadow = true;

light.shadowDarkness = 20;

light.shadowCameraVisible=true;

scene.add( light )

animation();

var a = 70;

var b = 0;

var c = 0.3 \* Math.PI / 100;

function animation() {

rot += .5;

renderer.render( scene, camera );

/\*\*

\* camera

\*\*/

const radian = ( rot \* Math.PI ) / 1000;

camera.position.x = 1000 \* Math.sin( radian );

camera.position.z = 1000 \* Math.cos( radian );

/\*\*

\* moon

\*\*/

const m\_radian = ( rot \* Math.PI ) / 100;

moon.position.x = 300 \* Math.sin( m\_radian );

moon.position.y = 0;

moon.position.z = 300 \* Math.cos( m\_radian );

/\*\*

\* earth

\*\*/

earth.rotation.x = 70;

earth.position.y = 0;

earth.rotation.z = 0.3 \* ( Math.PI / 100 );

camera.lookAt( new THREE.Vector3( 0, 0, 0 ) );

requestAnimationFrame( animation );

}

}