

Project describe: StormEye

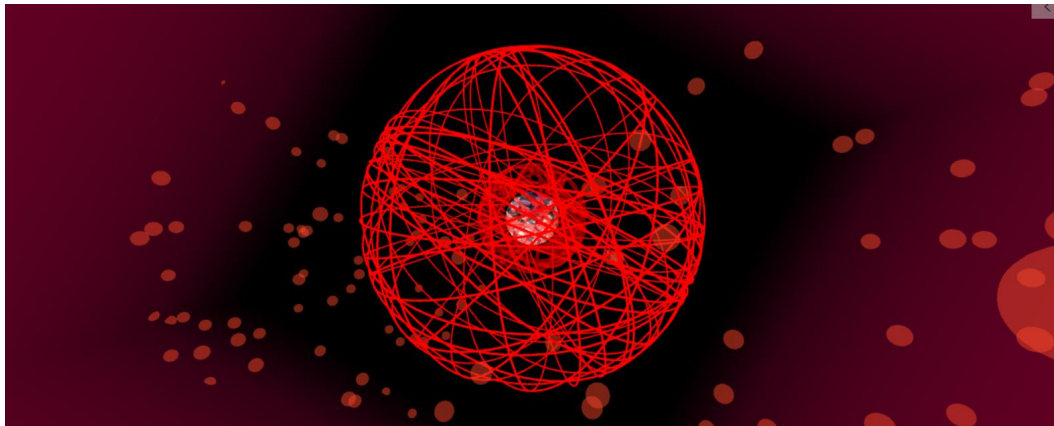
MIMIC Link: <https://mimicproject.com/code/edb00f8f-6bc9-4576-b16e-72fb734d7b45>

Github link : <https://github.com/zxyzxz/StormEye-Project>

Project inspiration:

The project inspired by a map adventure which I played, "Light Encounter", in the clouds kingdom, searching for lost souls around the world. Eventually taking the collected souls to the end of the map, the Garden of Eden, liberates the souls and returns them to the heavens. I used this as inspiration for my project, so I used a red and black gradient sky box to render the atmosphere, with a rainbow coloured ball in the middle to symbolise the souls in the sky, I added the sound of wind and the random spheres to represent the rocks and hurricanes in the Garden of Eden. The rubble hurricane and the lost souls are the symbol of the confrontation between light and darkness.

Project Show:



Coding part:

1. Sky box:

I first tried to use a sky box because I wanted to render a specific scene atmosphere, so I tried to create a special sky box by myself. I made a sky box with a red and black gradient using gradients in photoShop and then exported the image.

```
//skyBox!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
let materialArray = [];
```

```
let texture_ft = new THREE.TextureLoader().load( 'middle-2.png');
```

```
let texture_bk = new THREE.TextureLoader().load( 'middle-2.png');
```

```
let texture_up = new THREE.TextureLoader().load( 'up-21.png');
```

```
let texture_dn = new THREE.TextureLoader().load( 'down-2.png');
```

```

let texture_rt = new THREE.TextureLoader().load( 'middle-2.png');

let texture_lf = new THREE.TextureLoader().load( 'middle-2.png');


materialArray.push(new THREE.MeshBasicMaterial( { map: texture_ft }));

materialArray.push(new THREE.MeshBasicMaterial( { map: texture_bk }));

materialArray.push(new THREE.MeshBasicMaterial( { map: texture_up }));

materialArray.push(new THREE.MeshBasicMaterial( { map: texture_dn }));

materialArray.push(new THREE.MeshBasicMaterial( { map: texture_rt }));

materialArray.push(new THREE.MeshBasicMaterial( { map: texture_lf }));


for (let i = 0; i < 6; i++)

    materialArray[i].side = THREE.BackSide;

let skyboxGeo = new THREE.BoxGeometry( 1500,1500, 1500);

let skybox = new THREE.Mesh( skyboxGeo, materialArray );

scene.add( skybox );

animate();

```

By creating an array to store the material of the skybox, apply the created image as a texture to the inside of the cube. The pixel size of the sky image is 2048 x 2048.

2. Set the scene, set the camera, set the renderer, OrbitControls

```

//scene + camera and position

var scene = new THREE.Scene();

var camera = new THREE.PerspectiveCamera(55,window.innerWidth/window.innerHeight,45,30000);

camera.position.set(0,-20,-800);

camera.lookAt( scene.position );


//renderer:

var renderer = new THREE.WebGLRenderer({antialias:true});

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

```

```
document.body.appendChild(renderer.domElement);

window.addEventListener( 'resize', onWindowResize );

//controls:

let controls = new THREE.OrbitControls(camera);

controls.addEventListener('change', renderer);

controls.minDistance = 500;

controls.maxDistance = 1500;
```

3.Adding .wav sound

```
var audio = document.createElement("audio");

audio.src = "wind.wav";

//audio.src = "wind bell.wav";

audio.play();
```

Adding wind sounds to render the atmosphere.

4.Adding ambientlight and pointlighting

```
const ambient = new THREE.AmbientLight("#ff0000",10);

scene.add(ambient);

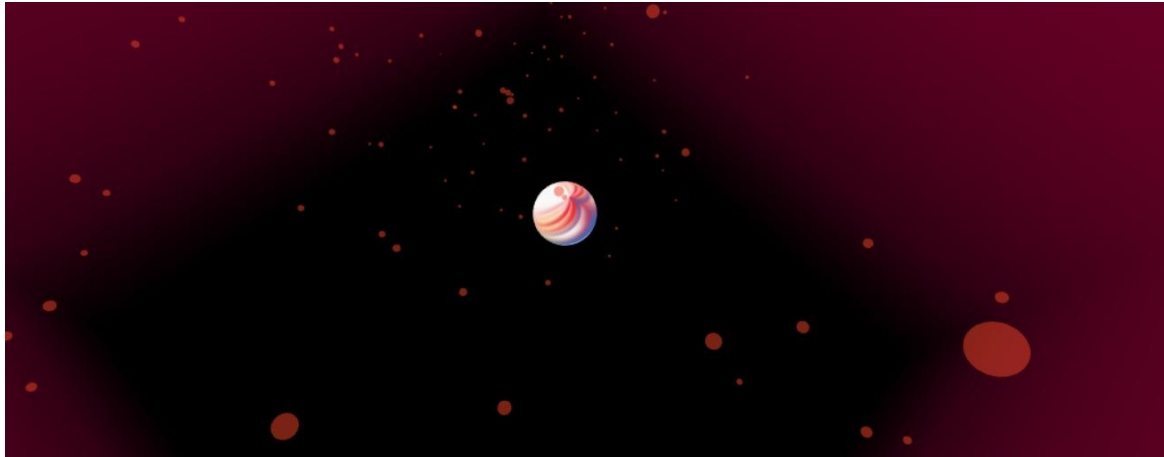
const pointLight = new THREE.PointLight("#ff0000", 10,100);

scene.add(pointLight);

pointLight.position.set(-800,0,0);
```

Add lights and set light colour, intensity and light position. And don't forget to add light to the scene.

5.Adding the geometry:colorful ball



```
//colorful ball
```

```
var geometry = new THREE.SphereGeometry( 50, 320, 160 );
```

```
var myTextureLoader = new THREE.TextureLoader();
```

```
var myTexture = myTextureLoader.load('background.jpg');
```

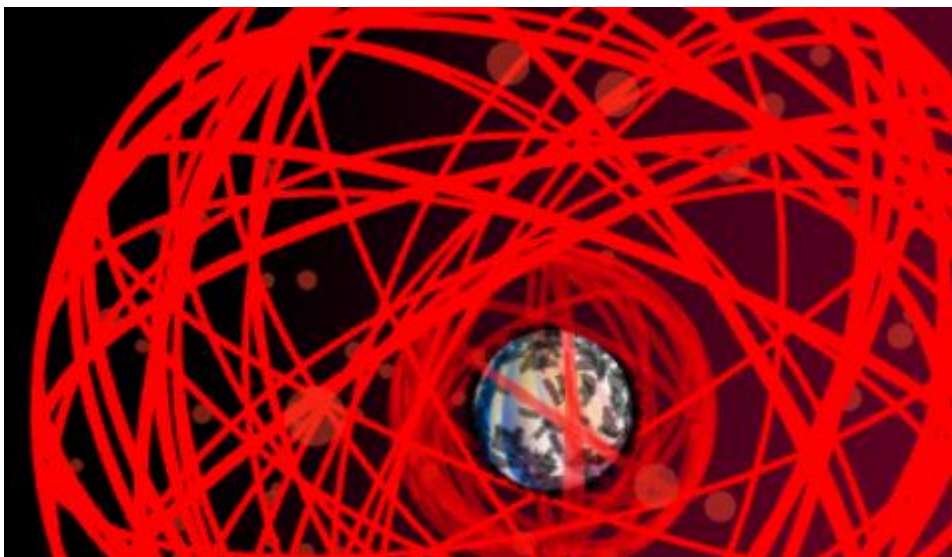
```
var geometryMaterial = new THREE.MeshBasicMaterial({map: myTexture});
```

```
var mesh = new THREE.Mesh(geometry, geometryMaterial);
```

```
scene.add(mesh);
```

Set the geometry parameters, load the material and texture maps and add them to the scene.

6.Adding random circles



```
// random Torus
```

```

var TorusGeometry = new THREE.TorusGeometry( 100, 3, 16, 100 );;

var TorusMaterial = new THREE.MeshBasicMaterial({

    color: "#ff0000",

    opacity: 0.5,

    transparent: true,

});

for(var i=0; i<20; i++) {

    var Torus = new THREE.Mesh(TorusGeometry,TorusMaterial);

    Torus.rotation.x = Math.round((Math.random() * 800));

    Torus.rotation.y = Math.round((Math.random() * -800));

    Torus.rotation.z = Math.round((Math.random() * 800));

    scene.add(Torus);

}

// random Torus2

var Torus2Geometry = new THREE.TorusGeometry( 300, 3, 16, 100 );;

var Torus2Material = new THREE.MeshLambertMaterial({

    color: "#ff0000",

    //emissive:"#000000"

    opacity: 0.2,

    //transparent: true,

});

for(var i=0; i<30; i++) {

    var Torus2 = new THREE.Mesh(Torus2Geometry,Torus2Material);

    Torus2.rotation.x = Math.round((Math.random() * 800));

    Torus2.rotation.y = Math.round((Math.random() * -800));

    Torus2.rotation.z = Math.round((Math.random() * 800));

    scene.add(Torus2);

}

```

Set the circle geometry parameters with a colour of #ff0000 and a transparency of 0.5 and use a for loop to define the number of circles and create a random rotation.

Set up a larger circle geometry.

7. Increase the effect of rubble — random sphere geometry



```
// random SphereGeometry

var SphereGeometry = new THREE.SphereGeometry( Math.round((Math.random() * 25-5)), 32, 15);

var SphereMaterial = new THREE.MeshBasicMaterial({

    color: "#F34527",

    opacity: 0.5,

    transparent: true,

});

for(var i=0; i<80; i++) {

    var Sphere = new THREE.Mesh(SphereGeometry,SphereMaterial);

    Sphere.position.x = Math.round((Math.random() *800));

    Sphere.position.y = Math.round((Math.random() *800));

    Sphere.position.z = Math.round((Math.random() *800));

    scene.add(Sphere);

}
```

Actually I wanted to use a texture, but the grey meteorite texture didn't show up very well, so I switched to an abstract effect.