Lesego Nkadimeng

COS3712 Assessment

This document contains all of the source code used I this current version of the solar

Note the Web 3D graphics Library Threejs was used to create this project

HTML File

```
!DOCTYPE html>

<html lang="en" >

<head>

       <meta charset="UTF-8">

      <title>Solar System</title>

      <link rel="stylesheet" href="./style.css">

</head>

<body>

 <script src='https://cdnjs.cloudflare.com/ajax/libs/three.js/105/three.min.js'></script>

<script src='https://s3-us-west-2.amazonaws.com/s.cdpn.io/2666677/OrbitControls.js'></script>


   <script  src="./script.js"></script>

</body>

</html>
```

CSS file

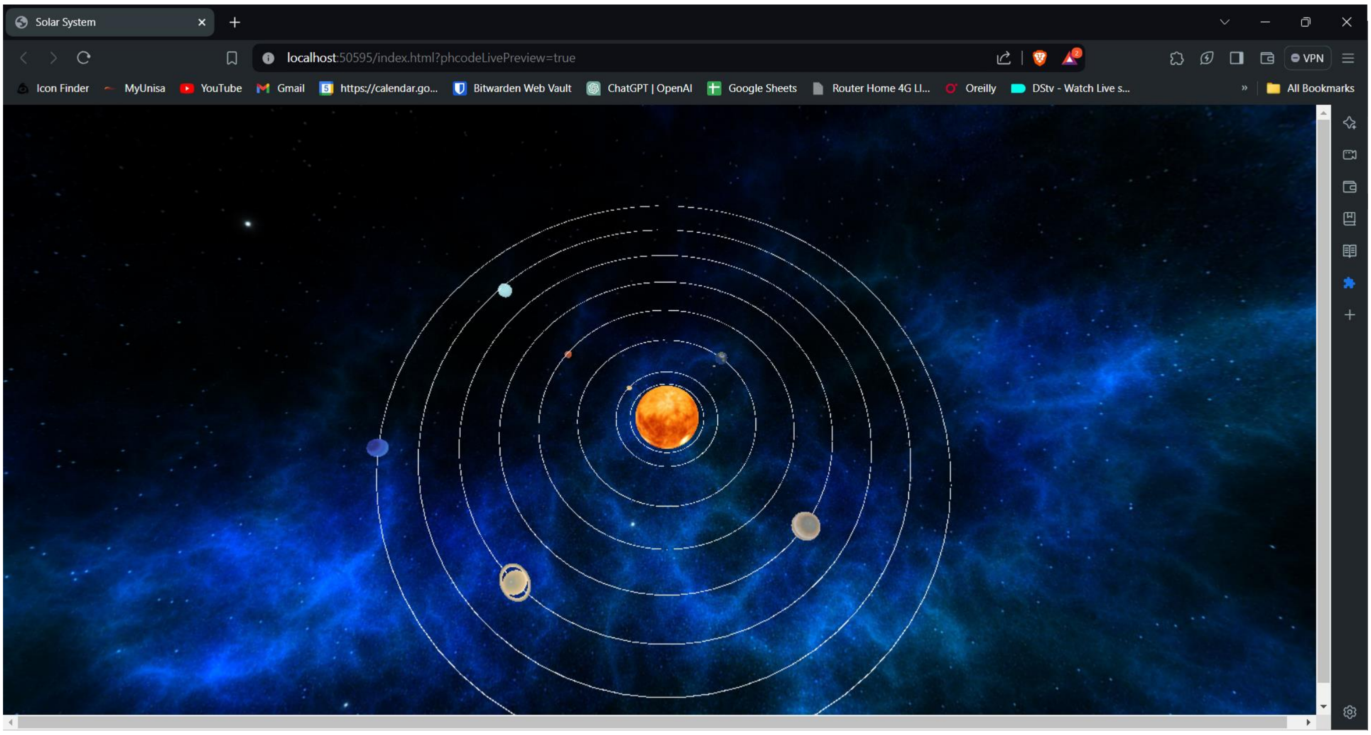
```
body{

  padding: 0;

  margin: 0;

}
```

Javascript file

```
var scene, camera, renderer, mesh;


init();
function init(){

  scene = new THREE.Scene();

  scene.background = new THREE.Color('grey');

camera = new THREE.PerspectiveCamera( 70, window.innerWidth /
window.innerHeight, 0.1, 1000 );

camera.position.set(0, 50, 500);


  const ambient = new THREE.HemisphereLight(0xffffbb, 0x080820);

  scene.add(ambient);


const light = new THREE.DirectionalLight(0xFFFFFF, 1);

  light.position.set( 1, 10, 6);

  scene.add(light);

   let orbitRot= 1.6;


  renderer = new THREE.WebGLRenderer();

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

  document.body.appendChild( renderer.domElement );


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

  controls.target.set(0,0,0);

  controls.update();


  const assetPath = 'https://s3-us-west-
2.amazonaws.com/s.cdpn.io/2666677/';


  const alpha = new THREE.TextureLoader().load(`${assetPath}dots.jpg`);
  const tex = new THREE.TextureLoader().load(`${assetPath}bricks-diffuse3.png`);


  const cubemap = new THREE.CubeTextureLoader()
      .setPath( `${assetPath}skybox1_` )
      .load( [
            'px.jpg',
            'nx.jpg',
            'py.jpg',
            'ny.jpg',
            'pz.jpg',
            'nz.jpg'
      ] );

  scene.background = cubemap;
    //Textures
    const textureloader = new THREE.TextureLoader();
    const suntexture=textureloader.load('./textures/sun.jpg');
    const earthtexture=textureloader.load('./textures/earth.jpg');
    const marstexture=textureloader.load('./textures/mars.jpg');
    const jupitertexture=textureloader.load('./textures/jupiter.jpg');
    const mecurytexture=textureloader.load('./textures/mercury.jpg');
    const venustexture=textureloader.load('./textures/venus.jpg');
    const saturntexture=textureloader.load('./textures/saturn.jpg');
    //const saturnRingstexture=textureloader.load('./saturnringalpha.png');
    const uranustexture=textureloader.load('./textures/uranus.jpg');
    const neptunetexture=textureloader.load('./textures/neptune.jpg');
    const earthMoontexture=textureloader.load('./textures/earthmoon.jpg');
```

```
const material = new THREE.MeshStandardMaterial({color:0xffff00,
metalness:0.95, roughness:0.01, envMap: cubemap});


    // orbit gemetry setup

     //const OrbitGeometry = new THREE.TorusGeometry(40,0.25,16,64,
Math.PI*2);

    const orbitMaterial= new THREE.MeshBasicMaterial( { color: 0xFFFFFF }
);


    //Add meshes here

    //the sun

    const geometry = new THREE.SphereGeometry( 25, 32, 16 );

    const sunMaterial = new THREE.MeshBasicMaterial( {map: suntexture } );

    sun = new THREE.Mesh( geometry, sunMaterial);

    scene.add( sun );

    sun.position.y =0;


    //Mecury

    const mecuryGeo = new THREE.SphereGeometry(1,32,160);

    const mecuryMat = new THREE.MeshBasicMaterial( { map: mecurytexture }
);

    mecury =new THREE.Mesh(mecuryGeo,mecuryMat);

    scene.add(mecury);

    mecury.position.set(29,0,0);

     const mecuryOrbitGeometry = new THREE.TorusGeometry(29,0.25,16,64,
Math.PI*2);

    mecuryOrbirt = new THREE.Mesh(mecuryOrbitGeometry,orbitMaterial);

    scene.add(mecuryOrbirt);

     mecuryOrbirt.rotation.set(orbitRot,0,0);


    //Venus

    const venusGeometry = new THREE.SphereGeometry(2,32,160);

    const venusMaterial = new THREE.MeshBasicMaterial( { map: venustexture
} );

    venus = new THREE.Mesh(venusGeometry,venusMaterial);
    venus.position.x= 40;
    scene.add(venus);
    const venusOrbitGeometry = new THREE.TorusGeometry(40,0.25,16,64,
Math.PI*2);
    venusOrbirt = new THREE.Mesh(venusOrbitGeometry,orbitMaterial);
    scene.add(venusOrbirt);
    venusOrbirt.rotation.set(orbitRot,0,0);

    //Earth
    const earthGeometry = new THREE.SphereGeometry(5,32,160);
    const earthMaterial = new THREE.MeshBasicMaterial( { map: earthtexture
} );
    earth = new THREE.Mesh(earthGeometry,earthMaterial);
    earth.position.set(0,0,70);
   // scene.add(earth);
    const earthOrbitGeometry = new THREE.TorusGeometry(70,0.25,16,64,
Math.PI*2);
    earthOrbirt = new THREE.Mesh(earthOrbitGeometry,orbitMaterial);
    scene.add(earthOrbirt);
    earthOrbirt.rotation.set(orbitRot,0,0);
    //moon
    moonGeo = new THREE.SphereGeometry(1,32,160);
    moonMaterial = new THREE.MeshBasicMaterial({map : earthMoontexture} );
    earthmoon = new THREE.Mesh(moonGeo,moonMaterial);
    earthgroup = new THREE.Object3D();
    earthmoon.position.set(0,0,60);

   earthgroup.add(earthmoon,earth);
    scene.add(earthgroup);

    //Mars
    const marsGeometry = new THREE.SphereGeometry(3,32,160);
    const marsMaterial = new THREE.MeshBasicMaterial( { map: marstexture }
);
    mars = new THREE.Mesh(marsGeometry,marsMaterial);
    mars.position.set(100,0,-10);
    scene.add(mars);
    const marsOrbitGeometry = new THREE.TorusGeometry(100,0.25,16,128,
Math.PI*2);
    marsOrbirt = new THREE.Mesh(marsOrbitGeometry,orbitMaterial);
    scene.add(marsOrbirt);
    marsOrbirt.rotation.set(orbitRot,0,0);

    //jupiter
    const jupiterGeometry = new THREE.SphereGeometry(10,32,160);
    const jupiterMaterial = new THREE.MeshBasicMaterial( { map:
jupitertexture } );
    jupiter = new THREE.Mesh(jupiterGeometry,jupiterMaterial);
    jupiter.position.set(-130,0,0);
    scene.add(jupiter);
    const jupiterOrbitGeometry = new THREE.TorusGeometry(130,0.25,16,64,
Math.PI*2);
   jupiterOrbirt = new THREE.Mesh(jupiterOrbitGeometry,orbitMaterial);
    scene.add(jupiterOrbirt);
    jupiterOrbirt.rotation.set(orbitRot,0,0);

    //saturn
    const saturnGeometry = new THREE.SphereGeometry(8,32,160);
    const saturnMaterial = new THREE.MeshBasicMaterial( {  map:
saturntexture } );
    saturn = new THREE.Mesh(saturnGeometry,saturnMaterial);
    saturn.position.set(0,0,-160);
    scene.add(saturn);
    const saturnOrbitGeometry = new THREE.TorusGeometry(160,0.25,16,64,
Math.PI*2);
    saturnOrbirt = new THREE.Mesh(saturnOrbitGeometry,orbitMaterial);
    scene.add(saturnOrbirt);
    saturnOrbirt.rotation.set(orbitRot,0,0);

    //Saturns Rings
    const saturnRingsGeo = new THREE.TorusGeometry(12,1.3,2,64, Math.PI*2);
    const saturnRingsMaterial = new THREE.MeshBasicMaterial( { map:
saturntexture } );
    saturnRings = new THREE.Mesh(saturnRingsGeo,saturnRingsMaterial);
    saturnRings.position.set(0,0,-160);
    saturnRings.rotation.set(1,0,0);
    scene.add(saturnRings);

    //uranus
    const uranusGeometry = new THREE.SphereGeometry(6,32,160);
    const uranusMaterial = new THREE.MeshBasicMaterial( { map:
uranustexture } );
    uranus = new THREE.Mesh(uranusGeometry,uranusMaterial);
   uranus.position.set(190,0,0);
    scene.add(uranus);
    const uranusOrbitGeometry = new THREE.TorusGeometry(190,0.25,16,64,
Math.PI*2);
    uranusOrbirt = new THREE.Mesh(uranusOrbitGeometry,orbitMaterial);
    scene.add(uranusOrbirt);
    uranusOrbirt.rotation.set(orbitRot,0,0);

    //Neptune
    const neptuneGeometry = new THREE.SphereGeometry(7,32,160);
    const neptuneMaterial = new THREE.MeshBasicMaterial( { map:
neptunetexture} );
    neptune = new THREE.Mesh(neptuneGeometry,neptuneMaterial);
   neptune.position.set(150,3,-162);
    scene.add(neptune);
    const neptuneOrbitGeometry = new THREE.TorusGeometry(220,0.25,16,64,
Math.PI*2);
    neptuneOrbirt = new THREE.Mesh(neptuneOrbitGeometry,orbitMaterial);
    scene.add(neptuneOrbirt);
    neptuneOrbirt.rotation.set(orbitRot,0,0);
    planetObj = new THREE.Object3D();
    planetObj.add(
        earthgroup,
        mecury,
        venus,
        neptune,
        mars,
        jupiter,
        saturn,
        saturnRings,
        uranus,
    );
   /* planetObj.add();
    planetObj.add();
    planetObj.add(mars);
    planetObj.add(neptune);
    planetObj.add(jupiter);
    planetObj.add(saturn);
    planetObj.add(saturnRings);
    planetObj.add(uranus);*/
    scene.add(planetObj);

  //  camera.lookAt(new vector3(190,0,0));
  window.addEventListener( 'resize', resize, false);

  update();
}
function animate(){
}renderer.setAnimationLoop(animate);

function update(){
  requestAnimationFrame( update );
      renderer.render( scene, camera );
     earth.rotation.y += 0.005;
    sun.rotation.y += 0.001;
   planetObj.rotateY (0.0005);
   // mars.rotateY(0.002);
  if (mesh!==undefined){
    earth.rotation.y += 0.01;
    mesh.rotation.y -= 0.01;
  }
}
function resize(){
  camera.aspect = window.innerWidth / window.innerHeight;
  camera.updateProjectionMatrix();
  renderer.setSize( window.innerWidth, window.innerHeight );
}
```

Current render


Solar System
localhost:50595/index.html?phcodeLivePreview=true
VPN
Icon Finder
MyUnisa
YouTube
Gmail
https://calendar.go...
Bitwarden Web Vault
ChatGPT | OpenAI
Google Sheets
Router Home 4G LI...
Oreilly
DStv - Watch Live s...
All Bookmarks