



Компютърна графика

Курсов проект на тема:

„3D Визуализация на Земята с Three.js”

Изготвил:

Николай Николов; гр.45; Фак.№ 121221019

1.Резюме

В проекта се демонстрира 3D визуализация на Земята в уеб страница чрез библиотеката Three.js, която ни дава възможност за лесен и достъпен начин при създаване на 3D сцени.

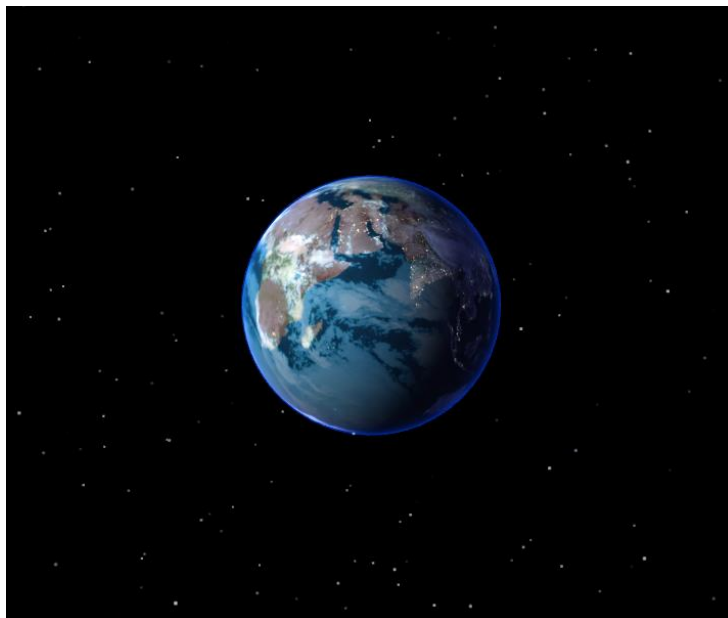
2.Въведение

За да бъде възможно изобразяването на каквото и да било чрез Three.js са необходими три основни неща: сцена, камера и рендър. Библиотеката Three.js улеснява създаването на 3D сцени в браузъра като дава възможност за използване на инструменти за създаване на сложни 3D обекти, предлага възможности за използване на различни геометрии, материали и текстури.

В проекта е показано как в уеб страница е създадено триизмерно изображение на планетата Земя като на заден фон са добавени звезди, има и насочена светлина имитираща слънчевата за осветяване на сцената, за управление на сцената се използва мишката.

3.Основна част

При стартиране на проекта в уеб браузъра се зарежда следния прозорец



Фиг.1

3.1.Реализиране на сцената:

3.1.1Задаване на височина и дължина на прозореца, сцена, камера и рендър

```
const w = window.innerWidth;
const h = window.innerHeight;
const scene = new THREE.Scene();
const camera = new THREE.PerspectiveCamera(75, w / h, 0.1, 1000);
camera.position.z = 5;
const renderer = new THREE.WebGLRenderer({ antialias: true });
renderer.setSize(w, h);
document.body.appendChild(renderer.domElement);
// THREE.ColorManagement.enabled = true;
renderer.toneMapping = THREE.ACESFilmicToneMapping;
renderer.outputColorSpace = THREE.LinearSRGBColorSpace;
```

3.1.2. Група, която задава всички текстури и компоненти необходими за визуализирането на планетата

```
const earthGroup = new THREE.Group();
earthGroup.rotation.z = -23.4 * Math.PI / 180;
scene.add(earthGroup);
new OrbitControls(camera, renderer.domElement);
const detail = 12;
const loader = new THREE.TextureLoader();
const geometry = new THREE.IcosahedronGeometry(1, detail);
const material = new THREE.MeshPhongMaterial({
  map: loader.load("./textures/00_earthmap1k.jpg"),
  specularMap: loader.load("./textures/02_earthspec1k.jpg"),
  bumpMap: loader.load("./textures/01_earthbump1k.jpg"),
  bumpScale: 0.04,
});
// material.map.colorSpace = THREE.SRGBColorSpace;
const earthMesh = new THREE.Mesh(geometry, material);
earthGroup.add(earthMesh);

const lightsMat = new THREE.MeshBasicMaterial({
  map: loader.load("./textures/03_earthlights1k.jpg"),
  blending: THREE.AdditiveBlending,
});
const lightsMesh = new THREE.Mesh(geometry, lightsMat);
earthGroup.add(lightsMesh);
```

```

const cloudsMat = new THREE.MeshStandardMaterial({
  map: loader.load("./textures/04_earthcloudmap.jpg"),
  transparent: true,
  opacity: 0.8,
  blending: THREE.AdditiveBlending,
  alphaMap: loader.load('./textures/05_earthcloudmaptrans.jpg'),
  // alphaTest: 0.3,
});
const cloudsMesh = new THREE.Mesh(geometry, cloudsMat);
cloudsMesh.scale.setScalar(1.003);
earthGroup.add(cloudsMesh);

const fresnelMat = getFresnelMat();
const glowMesh = new THREE.Mesh(geometry, fresnelMat);
glowMesh.scale.setScalar(1.01);
earthGroup.add(glowMesh);

const stars = getStarfield({numStars: 2000});
scene.add(stars);

const sunLight = new THREE.DirectionalLight(0xffffffff, 2.0);
sunLight.position.set(-2, 0.5, 1.5);
scene.add(sunLight);

```

3.3. Функции отговарящи за анимацията и промяна на разделителната способност на камерата.

```

function animate() {
  requestAnimationFrame(animate);

  earthMesh.rotation.y += 0.002;
  lightsMesh.rotation.y += 0.002;
  cloudsMesh.rotation.y += 0.0023;
  glowMesh.rotation.y += 0.002;
  stars.rotation.y -= 0.0002;
  renderer.render(scene, camera);
}

animate();

function handleWindowResize () {
  camera.aspect = window.innerWidth / window.innerHeight;
  camera.updateProjectionMatrix();
  renderer.setSize(window.innerWidth, window.innerHeight);
}

```

2. Функция GetStarfield.

3.2.1 Функцията getStarfield, генерира 3D звездно поле с помощта на Three.js. Звездите се поставят произволно върху повърхността на сфера и се изобразяват като точки с текстура, приложена към всяка от тях.



Фиг.2

3.2.2 Функцията randomSpherePoint генерира произволна точка върху повърхността на сфера (в 3D пространство), използвайки сферични координати.

```
function randomSpherePoint() {  
  const radius = Math.random() * 25 + 25;  
  const u = Math.random();  
  const v = Math.random();  
  const theta = 2 * Math.PI * u;  
  const phi = Math.acos(2 * v - 1);  
  let x = radius * Math.sin(phi) * Math.cos(theta);  
  let y = radius * Math.sin(phi) * Math.sin(theta);  
  let z = radius * Math.cos(phi);  
  
  return {  
    pos: new THREE.Vector3(x, y, z),  
    hue: 0.6,  
    minDist: radius,  
  };  
};
```

3.2.3 Този цикъл създава звездите чрез извикване `randomSpherePoint()` на всяка звезда, извлича позицията и цвета на звездата.

```
const verts = [];  
const colors = [];  
const positions = [];  
let col;  
for (let i = 0; i < numStars; i += 1) {  
  let p = randomSpherePoint();  
  const { pos, hue } = p;  
  positions.push(p);  
  col = new THREE.Color().setHSL(hue, 0.2, Math.random());  
  verts.push(pos.x, pos.y, pos.z);  
  colors.push(col.r, col.g, col.b);  
}
```

4. Заключение

В този проект успешно е създадено 3D представяне на планетата Земя, включващо реалистични текстури, осветление и интерактивни елементи. Проектът демонстрира как да се комбинират различни функции на Three.js, включително геометрия, материали и контрол на камерата. Използвайки различни текстури за повърхността на Земята, облаците и нощните светлини се симулира добро визуално изображение на планетата Земя, като по този начин се дава пример за това как Three.js може да се използва за създаване на, интерактивни 3D модели.

5. Източници

1. <https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>
2. <https://discourse.threejs.org/t/want-to-change-background-color-of-star-animation/29492>
3. <https://www.youtube.com/watch?v=XPhAR1YdD6o>
4. <https://www.youtube.com/watch?v=vuA3qG15Jfl>
5. https://www.youtube.com/watch?v=vLz2Rk1r_gQ

Съдържание

Резюме:	2
Въведение:	2
Основна част:	2
Заключение:	6