

浙江大学计算机图形学

课程作业报告（2024-2025 年秋冬）

姓名：Wu Yizheng 学号：324060007 专业：Computer Science

一、 实验实现的功能简述及运行说明

The program completed is a WebGL-based solar system simulation built with Three.js. Earth, Moon, Sun, Jupiter, Mars, Venus, Saturn and Mercury are created. I achieved the following main features and functions in the program:

- View and move around the solar system using OrbitControls.
- Move the camera dynamically using keyboard inputs (W, A, S, D, Arrow keys).
- Turn rotation/orbit animations on and off using the E key.
- Reset the camera position using the R key.
- Simulate planetary rotations and orbital movements. Show orbital tracks with O key.
- Use realistic textures for planets.

The program runs in a browser and uses an HTML `<canvas>` element to render the 3D scene.

二、 作业的开发与运行环境

The program has the following specifications that need to be installed or imported:

- **Three.js** for 3D graphics and rendering.
- **OrbitControls** for interactive camera controls.
- **Node.js** modules, including `gl-matrix`, to help with mathematical
- **Babel** for ES6+ JavaScript compatibility.
- The program is designed to work with textures stored locally.

Ensure the project has:

- Proper textures in the `textures/` directory (e.g., `stars.jpg`, `sun2.jpeg`, `earth2.jpeg`).
- Dependencies installed via `npm`
- Use `npx webpack` to build, then use `npx webpack serve` to serve the site and access it on a local host url provided by webpack.

三、 系统或算法的基本思路、原理、及流程或步骤等

The solar system simulation has the following principles:

- A **scene** (`THREE.Scene`) is created to hold all objects.
- A **camera** (`THREE.PerspectiveCamera`) observes the scene.
- A **renderer** (`THREE.WebGLRenderer`) draws the scene onto the canvas.
- **OrbitControls** enables interactive exploration of the scene.

Steps in the program:

1. Load necessary textures (Sun, Earth, Moon...etc).
2. Create planets using `THREE.SphereGeometry` and assign textures.
3. Create Lights and shadows to add to the scene.
4. Group planets to simulate relationships (e.g., Earth and Moon) using `THREE.Object3D()`.
 1. Each planet has its own group: to allow Moon to orbit around Earth, Moon group is added to Earth group
5. Render the scene and animate orbits using `requestAnimationFrame`.

四、 具体如何实现，例如关键（伪）代码、主要用到函数与算法等

This is an example of how the orbital controls are set up.

```
controls = new OrbitControls(camera, renderer.domElement);
controls.enableDamping = true; // Smooth camera movements
controls.minDistance = 10; // Limits zoom-in
controls.maxDistance = 100; // Limits zoom-out
```

Below is an example of how planets are initialized and added to planet groups with earth as an example. A planet has two main components, geometry and texture. For geometry, `sphereGeometry` from Three.js library was used, and for the material, `Phong Material` with texture was loaded such that the planets can have realistic reactions to light.

```
const earthGeometry = new THREE.SphereGeometry(1, 32, 32);
const earthMaterial = new THREE.MeshPhongMaterial({map:
textureLoader.load("./textures/earth2.jpeg")});
earth = new THREE.Mesh(earthGeometry, earthMaterial);
earthGroup = new THREE.Object3D(); // Grouping Earth for easy
rotation/orbit
earthGroup.add(earth);
```

Ambient light, point light to mimic sunlight are used. Below is the code snippet for sunlight. `castShadow` function is enabled to make the rendering more realistic.

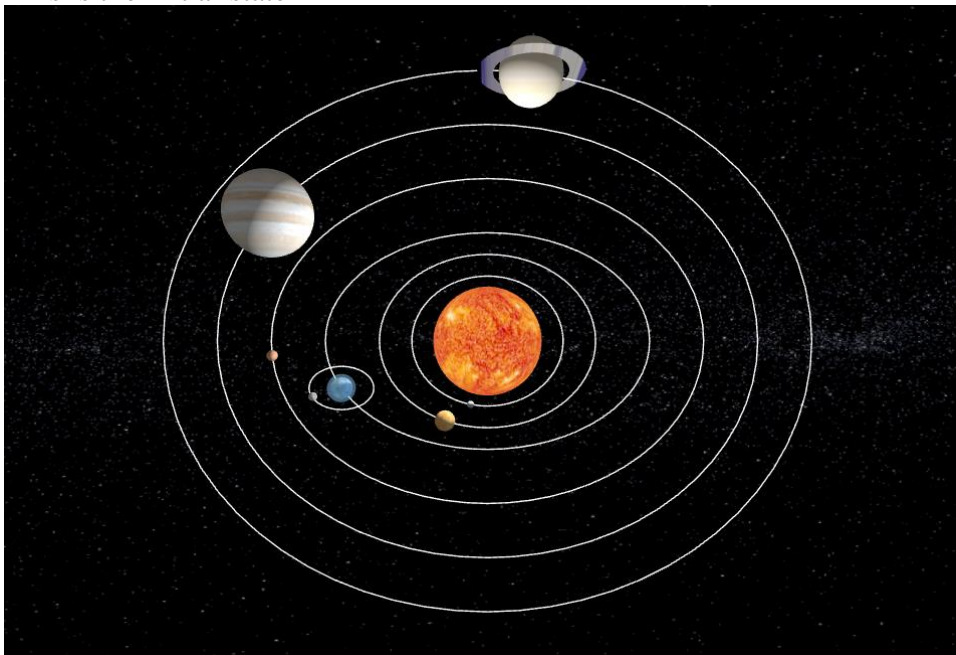
```
// Point Light Mimics sunlight's effect
const pointLight = new THREE.PointLight(0xffffff, 20, 200, 0.5);
pointLight.position.set(0, 0, 0);
pointLight.castShadow = true;
scene.add(ambientLight, pointLight);
```

In order to mimic the elliptical orbits of the planets, planet groups' positions are set using semi-major and semi-minor axis and an increase in the orbit angle. Below is an example of Earth's orbit.

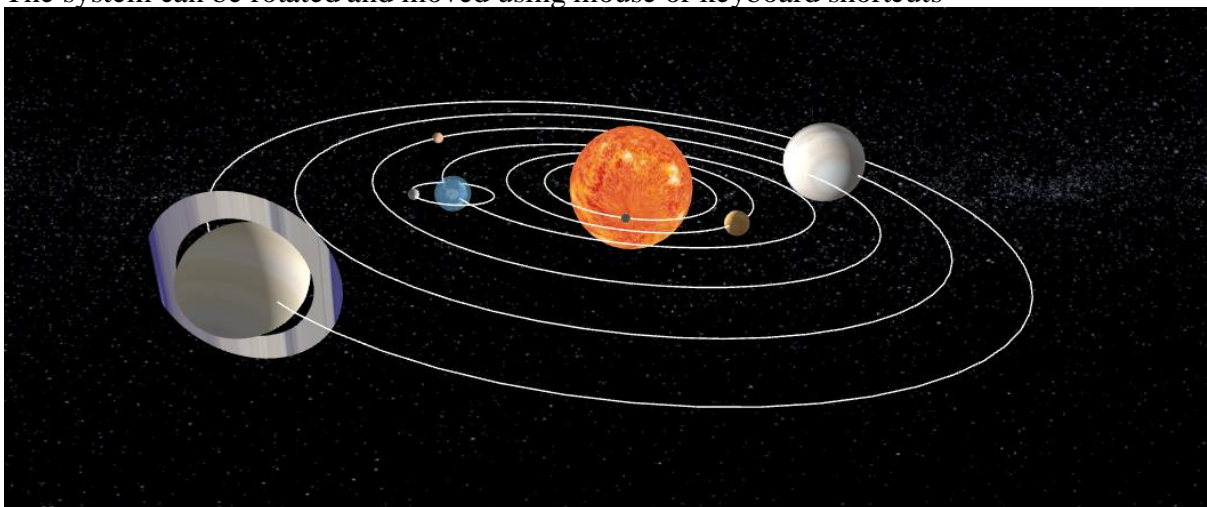
```
// Elliptical orbits
angleEarth += 0.01;
earthGroup.position.set(
    15 * Math.cos(angleEarth), // Semi-major axis for Earth
    10 * Math.sin(angleEarth), // Semi-minor axis for Earth
    0
);
```

五、 实验结果与分析

This is the initial state



The system can be rotated and moved using mouse or keyboard shortcuts



The orbit lines can be turned offed using keyboard “o”



六、 参考文献

<https://threejs.org/docs/#api/en/>