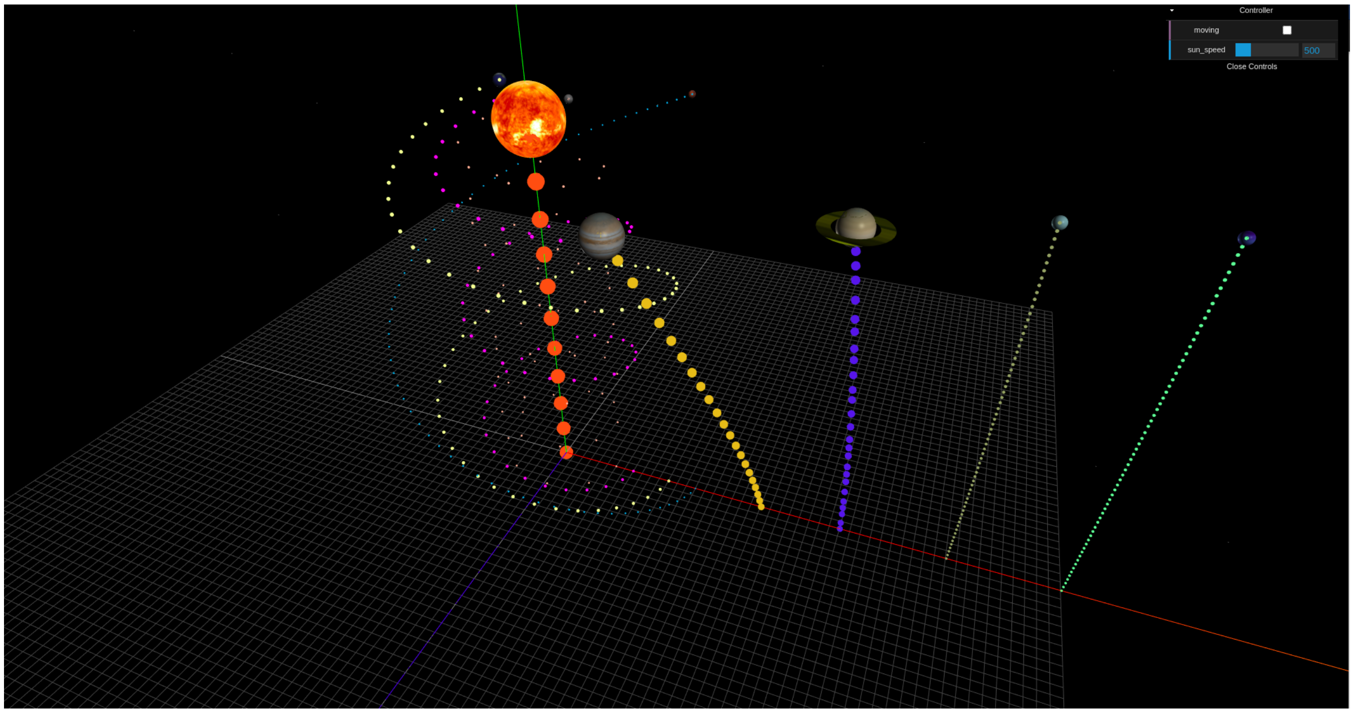
Solar System in Motion

tiffanyxqz and haoyu2
{xiaoqian.zhang001},{haoyu.wang001}@umb.edu
University of Massachusetts Boston



moving

sun_speed 500

Figure 1: Dotted spiral lines following each moving planets and the Sun

ABSTRACT

This project is meant to show the motion trace of the solar systems planets when they are moving along with the Sun.

KEYWORDS

WebGL, Visualization, Motion Trace

ACM Reference Format:
tiffanyxqz and haoyu2. 2020. Solar System in Motion. In *CS460: Computer Graphics at UMass Boston, Fall 2020.* Boston, MA, USA, 3 pages. https://CS460.org

1 INTRODUCTION

Demo: https://haoyu2.github.io/cs460student/final/build/
Source code: https://github.com/Haoyu2/cs460student/tree/master/final

This project demonstrated a moving solar system with traces following each celestial body which helps audience to gain a perspective of motion in different coordinate system.
We prototypes each of the 8 planets and the Sun and then add motion and motion trace to every one. We also add a interactive control of making them move or stay still as well as the speed of the moving Sun.

2 RELATED WORK

We use the libaray XTK [2] and Three.js [1] for our model and Snowpack or our project builder.
The planets' texture I used the images and material from James EdMichaud's work Solar System.

3 METHOD

- First, we create the three.js environment for showing any object which is **World.js** in our code.
- Second, we write a function **addSolarMeshes** which add all solar planets and the Sun into the scene which is created in the World.js.

- Third, we write a function **addMotions** which takes all the planets and the Sun as argument and then add motions and motions trace to all of them.
- Fourth, we add a interactive control in **gui.js** which controls the moving of all the object with a default value to be false which makes every object to be still when we launch the webpage. And we add a speed control of the moving Sun.

3.1 Implementation

Please tell the reader how you implemented the project. You can include code snippets that you want to highlight. Don't include the whole code.

This is the code for World environment.

```
export class World {
    constructor(container) {
        this.renderer = createRenderer(container);
        this.camera = createCamera(container);
        this.scene = createScene();
        addLights(this.scene);

        this.updatables = [];
        this.clock = new THREE.Clock();

     addOrbitControls(this.camera, this.renderer, this.updatables);

    }

    render() {
        this.renderer.render(this.scene, this.camera)
    };

    start() {
        this.renderer.setAnimationLoop(() => {
                const delta = this.clock.getDelta();
                this.updatables.forEach((obj) => {
                    obj.update(delta);
             this.renderer.render(this.scene, this.camera);
                });
            }
        )
    }

    stop() {
        this.renderer.setAnimationLoop(null);
    }

}
```

This is the code for creating meshes:

```
const geometry = new THREE.SphereBufferGeometry(r, 32, 32);
    const material = name == 'sun' ?
        new THREE.MeshMatcapMaterial({
            color: 0xffffff,
        map: new THREE.TextureLoader().load(config.texture)
        })
        :
        new THREE.MeshPhongMaterial({
            color: 0xffffff,
            specular: 0x050505,
            shininess: 100,
        map: new THREE.TextureLoader().load(config.texture)
        });
    const mesh = new THREE.Mesh(geometry, material);
```

This is the code for adding motions:

```
export const addMotions = (meshes, configs) => {
   const colors = ['#FF6633', '#FFB399', '#FF33FF', '#FFFF99', '#00B
     '#E6B333', '#3366E6', '#999966', '#99FF99', '#6666FF'];
    let i = 0;
    for (let [name, mesh] of Object.entries(meshes))
      addMotion(name, mesh, configs[name], colors[i++]);

}
```

3.2 Milestones

How did you structure the development?

3.2.1 Milestone 1. We thought about doing some work to practive transformations. And then the idea of solar system in motion came to mind, because though all planets rotating around the Sun, but the Sun also moves itself.

3.2.2 Milestone 2. We tried to build the rotating solar system in the Sun's coordinate system and then move the Sun's coordinate system in the scene's system and then transform all the positions of the planets in Sun's system into the scene's global system to make traces which are observed from the scene's global system.

We tried the transform from world to local and local to world of both the scene and planets. It gets complicated and messed up. And it seems that we can not finish this project.

3.2.3 Milestone 3. Then Xiaoqian suggests that we can just first make this work by simply getting the motion traces in the scene's global coordinate system without the transformation. And then we made this work by simply add a motion to the Sun's y axis.

3.3 Challenges

Describe the challenges you faced.

- Challenge 1: how to efficiently show the trace. We can use line and we can use some sphere as dots. Finally we chose to use sphere as dots and we only move the last dots on the trace to update its position to the planet's newest position when the current position to the previous position is large enough.
- Challenge 2: We want to make the camera move along the motion. But when we did this and stop the motion, we found that we cannot use the camera's orbit control to observe the scene. So we decouple the motion of the camera with the camera's orbit control in the animation.

4 RESULTS

We have the motion traces showed up in our scene as in this image 5. Also we can make it still and the Sun moving in different speed. Also if you exam the real scene, each object is also self rotating.

Or you could add tables (see Table 1 - maybe with some timings?).

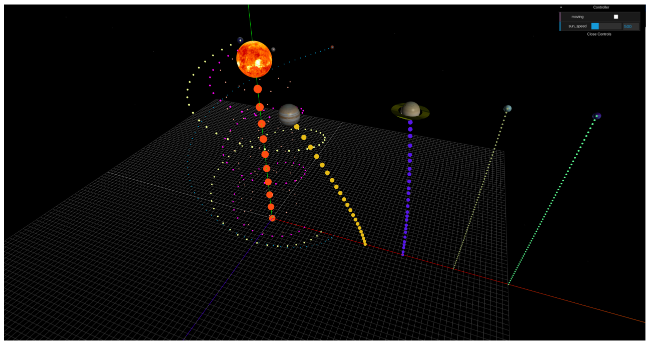


Figure 2: An example image.

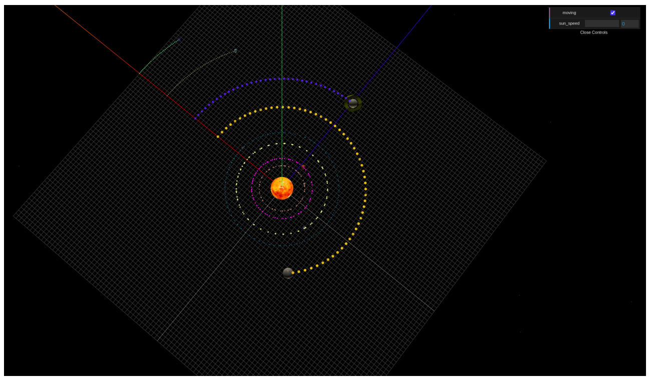


Figure 3: An example image.

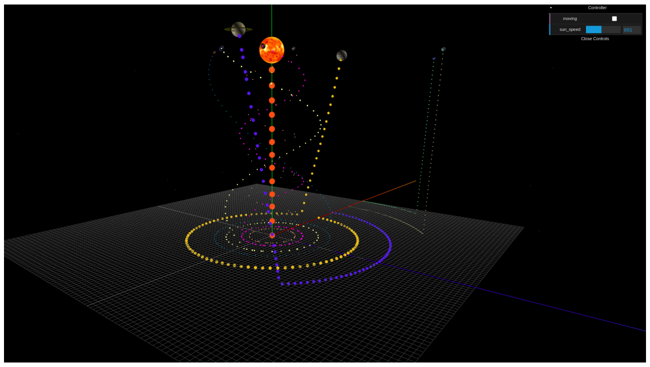


Figure 4: An example image.

Table 1: Some example table

Device	Performance
iPhone	60 FPS
Android	60 FPS
Old Macbook	10 FPS

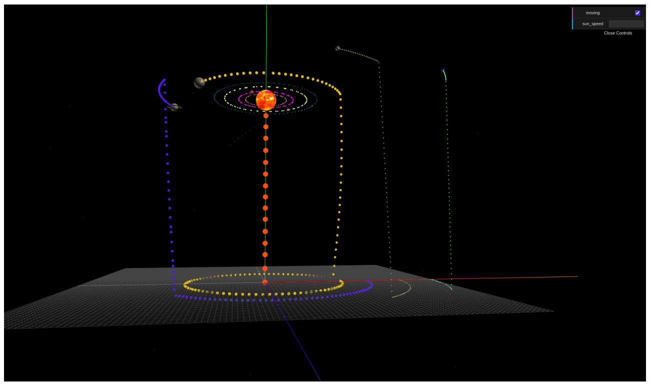


Figure 5: An example image.

5 CONCLUSIONS

We show the solar system in motion in a bigger view by adding traces to each moving object. This scene is interactive in the sense that we can start or move the planets and we can also control the speed of the moving Sun.

Demo: https://haoyu2.github.io/cs460student/final/build/ Source code: https://github.com/Haoyu2/cs460student/tree/master/final

Your references are loaded in BibTex from references.bib!

REFERENCES

[1] Ricardo Cabello et al. 2010. Three.js. *URL: https://github. com/mrdoob/three.js* (2010).
[2] Daniel Haehn, Nicolas Rannou, Banu Ahtam, P. Ellen Grant, and Rudolph Pienaar. 2012. Neuroimaging in the Browser using the X Toolkit. *Frontiers in Neuroinformatics* (2012).