Repo link: https://github.com/lukeboyden/-CS559-Spring24-P3-lukeboyden.git

**How to View and Interact with the Solar System Simulation**

**Viewing Instructions:**

1. Open gr-01-01.html in the for\_students folder to view the simulation.
2. Use your mouse to rotate around the space by clicking and dragging.
3. Zoom in and out using the scroll wheel to get a closer view of any planet.
4. Click on any planet to display detailed information about it in the top-right corner of the screen.

**Interaction Details:**

* Rotation: Click and drag anywhere in the space to look around from different angles.
* Zoom: Scroll up to zoom in and down to zoom out, focusing on different parts of the solar system.
* Information Display: Click on planets to view details such as composition, orbit, and history in an information card on the screen.
* Time Slider: Adjust the time slider to control the speed of the planetary orbits. This feature allows you to speed up or slow down time, providing a dynamic view of how planets move around the sun. Observing changes over time can help you understand orbital mechanics and the relative motion of celestial bodies.

**Description of Implementation:**

**Key Components of the Simulation:**

* Three.js Library: Utilizes Three.js for 3D rendering.
* OrbitControls.js: Allows user-friendly camera control for rotating and zooming the view.
* Planetary Data: Contains predefined properties for each planet like distance from the Sun, texture, and orbital speed, which are crucial for realistic rendering and movement.
* Lighting: Includes ambient light for overall illumination and a point light from the Sun to mimic sunlight in the solar system.

**Simulation Implementation:**

1. Environment Setup: Initialize the scene, camera, and renderer. Add ambient lighting to simulate the light from the sun.
2. Create Planets: Define each planet's properties including texture, orbit, and size. Note: Due to the vast nature of the solar system, it is practically impossible to implement a 1-to-1 realistically scaled solar system and have the program remain usable. Thus, certain measurements have been scaled for usability-purposes; refraining from doing so as much as possible.
3. Animate Orbits: Calculate the position of each planet in its orbit over time using Kepler's laws. This involves updating the position based on orbital speed and distance from the Sun.
4. Interactive Elements: Implement click event listeners to enable users to click on planets and display information about them dynamically.
5. Rendering Loop: Continuously update and render the scene to reflect changes, such as planetary movements and user interactions.

**Educational value:**

The project implements a simulation of Kepler's laws to accurately animate the elliptical orbits of the planets, providing an educational insight into the dynamics of planetary motion. Students can observe and interact with the simulation as it unfolds in real time, making it a dynamic learning tool. Using interactive 3D elements this project provides an interactive and engaging educational tool that teachers can leverage to give their students an engaging method of learning about the makeup and dynamism of our solar system.