Orrery Model Web App:

Create an Orrery Web App that Displays Near-Earth Objects

Event
2024 NASA Space Apps Challenge

Difficulty
Intermediate
Advanced

Subjects
Astrophysics | Games | Planets & Moons | Software | Space Exploration

Since a mechanical model of the solar system was presented to Charles Boyle, 4th Earl of Orrery, in 1713, such models have been referred to as orreries. The first orreries were physical models, but today we can use numerous tools to create virtual orreries that have many more features than their ancient mechanical counterparts. Your challenge is to create an interactive orrery web app that is embedded in a webpage and displays celestial bodies such as planets, Near-Earth Asteroids, Near-Earth Comets, and Potentially Hazardous Asteroids.

Challenge Explanation:

You're being asked to create an interactive web application that functions as an orrery—a digital model of the solar system. The key twist is that this orrery will not only display planets but also track and display Near-Earth Objects (NEOs) such as asteroids, comets, and potentially hazardous asteroids that could pose a threat to Earth. The app should be embedded within a webpage and offer an engaging, interactive experience for users.

Orrery Model:

Tech Stack Needed:

- **Frontend:**
  - **HTML/CSS**: Structure and style the web page.
  - **JavaScript/TypeScript**: Implement interactive features and real-time updates.
  - **React.js**: Build reusable UI components and manage the app's state.
  - **Three.js**: Render 3D models and animations of celestial bodies in the browser.
  - **D3.js**: Visualize data related to the orbits and trajectories of NEOs.
  - **Tailwind CSS**: Use for responsive design and pre-styled components.
- **Backend:**
  - **Django**: Handle server-side logic, API requests, data processing and building the server and handling routes.
  - **MongoDB**: Store data related to celestial bodies, NEOs, and user interactions.
  - **RESTful APIs**: Integrate with external APIs (e.g., NASA's API) to fetch real-time data about NEOs.
  - **WebSockets**: Implement real-time communication between the server and client for live updates.

- **Data Sources:**
  - **NASA Open APIs**: Fetch real-time data about planets and Near-Earth Objects.
  - **SPICE Toolkit**: For precise astronomical data and trajectory calculations.
- **DevOps/Hosting:**
  - **Docker**: Containerize the application for easier deployment.
  - **Heroku/Netlify/Vercel**: Host the web application.
  - **Git/GitHub**: Version control and collaboration.
- **Optional Enhancements:**
  - **WebGL**: For advanced 3D rendering and performance optimization.
  - **Redux**: If using React, to manage complex state across the application.
  - **GraphQL**: For more efficient data fetching and manipulation if needed.
- **3D Modeling:**
  - **Creating Custom 3D Models**: If you want to create detailed, custom 3D models of planets, asteroids, or other celestial bodies, Blender can be used to design and export these models into formats that can be used with Three.js or WebGL.
  - **Advanced Animations**: Blender can also be used to create complex animations that you could import into your web app. However, simpler animations can be handled directly within Three.js or via CSS and JavaScript.

Brainstorm of what we can do.

The Web Application:

- **The Frontend:**
  - 3d model (blender).
  - Web app (NextJS + ThreeJs).
  - UI/UX.
- **The Backend:**
  - Data gathering Api (NEOs).
  - LLM Api.
  - Users system (time based rank) → Statistics UX
- **The Shenanigans:**
  - TTS (LLM api).
  - Large scale testing.

Tasks:

Task	Name
3d Model	Mason
Frontend	SiZiF + Z3ln + Mason
UI/UX	Z3ln + Mason
Backend	Ktlr + 3b3ziz + lilbaba
LLM Api	lilbaba + 3b3ziz
Presentation	Mason (training)
Testing (ON site)	ALL (pr training)
Documentation	Ktlr

Roles:

Name	Role
3b3ziz	Backend & LLM Api
Ktlr	Backend & Documentation
SiZiF	Frontend & connecting the backend
Z3ln	Frontend & UI/UX
lilbaba	Backend & LLM Api
Mason	Fuck my life

Timeline:

- Learning Tech Stack
- Starting Work *(1st September)*
- 1st Proto-type *(15th September)*
- The Hackathon (Thur & Fri October)

Needs:

1. مشترك (6 مخارج)
2. حاجات ننام عليها (اغطية عين + مخدة صغيرة)
3. اقلام صبورة + حاجة نمسح بيها الصبورة
4. مناديل
5. كتشب
6. ازايز ماية
7. ترموس (كل واحد يجي بحاجة لو عاوز)
8. شرابات تقيلة (عشان نقعد بيها هناك عشان مش هنمشي حافيين ولا هنقعد اليومين بالجزم)

DDay:

9pm Day 1:

- Hackathon opening & Orientation.

12pm Day 1:

- Chitchats and Breakfast.
- Starting to wrap work up and testing it.
- Designing the presentation (Mason)

~6pm Day 1:

- Lunch.
- Lunch break.

8pm Day 1:

- Continue work on the project.

12am Day 2:

- Presentation Training.

1~2am Day 2:

- Public presentation of the project (on site marketing to get statistics and public opinions).

3am Day 2:

- Late Night Walk & Chitchats.

Fajr Day 2:

- Pray.
- Sleep.

12pm Day 2:

- Friday prayer.
- Omar Waleed’s (lilbaba) arrival to the AUC.

1pm Day 2:

- Rehearse the presentation.

3~4pm Day 2:

- More Public presentation.

~6pm Day 2:

- Lunch.
- Lunch break.

8~10pm Day 2:

- Hackathon closing & Coronation (إن شاء الله).