

# UniCS Challenge

## Project name

Solar System Visualiser

## Elevator pitch

Because visualising the Solar System should not only be accessible and coherent but also to scale and accurate.

## Project Story

### \* About the project

**Be sure to write what inspired you, what you learned, how you built your project, and the challenges you faced. Format your story in Markdown.**

### ## Inspiration

A game called Universe Sandbox is the inspiration for this project (Hence why we are using its music in our website). It is a game in which you can spawn celestial bodies and simulate them on the basis of various conditions. We wanted to use a similar concept for our project where we simulate the solar system with its intricate properties at as much accuracy as we can manage, while also making it accessible from a simple web browser. The scope was a sense of realism , while keeping it fun.

Shoutout to the big bang for creating the solar system, he's a real one.

### ## What it does

It is a simulator that helps the user immerse themselves in the solar system. The user initially spawns looking at the Sun. From there, it is the user's choice on where they want to traverse and which celestial body they want to learn more about. The website is very intuitive to use and the exploratory aspect of it keeps its users hooked.

### ## How we built it

The first step was to put on paper a rough design of the ideas we had in our mind at the beginning. Throughout the development, a notebook of ideas was maintained in which we documented our features, ideas, and tasks to be implemented. Tasks were assigned on-the-go, but rough roles were set from the start. One member would build the skeleton of the program, the physics, movements, mechanics. Another member would research, design and implement the visual aesthetics of the program. Another person would provide background research and brainstorming, testing, debugging, feature implementations, and someone would add more details such as realism of scales, data, space, ui interaction and

information for the educational panels. Based on skills and difficulty of tasks, members would pick up new features and tasks from the notebook to follow through, with constant feedback, testing and brainstorming from the entire team, taking notes and improving, following a more agile approach. Throughout the project development, a repository on GitHub was maintained for version control and backup security.

## **## Challenges we ran into**

- Had to research websites in order to obtain 3D images for our project's assets folder.
- Our program crashed when we tried hosting it on the server, but we soon fixed the problem.
- Incompatibility of assets or software across different operating systems and integration difficulties were a few of the more frequent setbacks we met, until we found the proper libraries, asset sources, and source control for proper sharing. Integrating 3D models into three.js proved to be difficult.
- Data corruption was another issue, as the constant movement of files across devices, local and cloud, was prone to cause errors, which we fortunately dealt with.

## **## Accomplishments that we're proud of**

The team communication was delightful. Task assignments, brainstorming, discussion on progress and project direction was flawless. The team was motivated and willing to finish the event with a project to be proud of, which is something we all felt we accomplished at the end. The accomplishment was measurable, as we managed to finish and implement most of the features we wrote down from the start. We maintained a realistic view of what direction we wanted to follow, while being aware of possible "could-be" future expansions that were limited by time, and it didn't disappoint.

## **## What we learned**

- Displaying 3D computer graphics using three.js
- Understanding and implementing orbits and other convoluted physics concepts
- How to optimize thousands of objects
- Sooo many moons
- Camera manipulations are way harder than we ever imagined
- JavaScript is a fake language designed to torment developers
- Javascript...

## **## What's next for Solar System Visualiser**

- First-person controller (player spaceship able to interact with planets)
- More information about celestial bodies
- Universe editor
- GPU-shader-based particle systems
- More celestial object types (asteroid, satellites, rockets, etc)
- Advanced realism (timelines, future predictions, orbital mechanics)
- Double-check simulation data (we eyeballed it :b)
- Better UI

**\* Built with**

**What languages, frameworks, platforms, cloud services, databases, APIs, or other technologies did you use?**

- Node
- three.js
- Docker
- Git

**"Try it out" links**

**Add links where people can try your project or see your code.**

<https://studenthack.karaolidis.com> (Allow autoplay for music :D)  
<https://git.karaolidis.com/karaolidis/studenthack2024-project>

**Provide details on how your project aligns with the outlined challenges of UNICS:**

**If you answered "Yes" to any of the above questions, could you provide details on how your project aligns with the outlined challenges of the respective society?**

Our project is model of our solar system, with a sense of realism, qualifying for the space topic. It also included most of the solar system's major entities and information about them.

## **UNI CS Brief**

**Develop an educational app or website to teach primary school aged kids about the solar system.**

- **Think about interactivity, and how it can be engaging for the target users.**
- **Think about the interface, and making it intuitive for children.**
- **Think about what your resource can do, that others can't. Be creative!**