Solarmax 3D

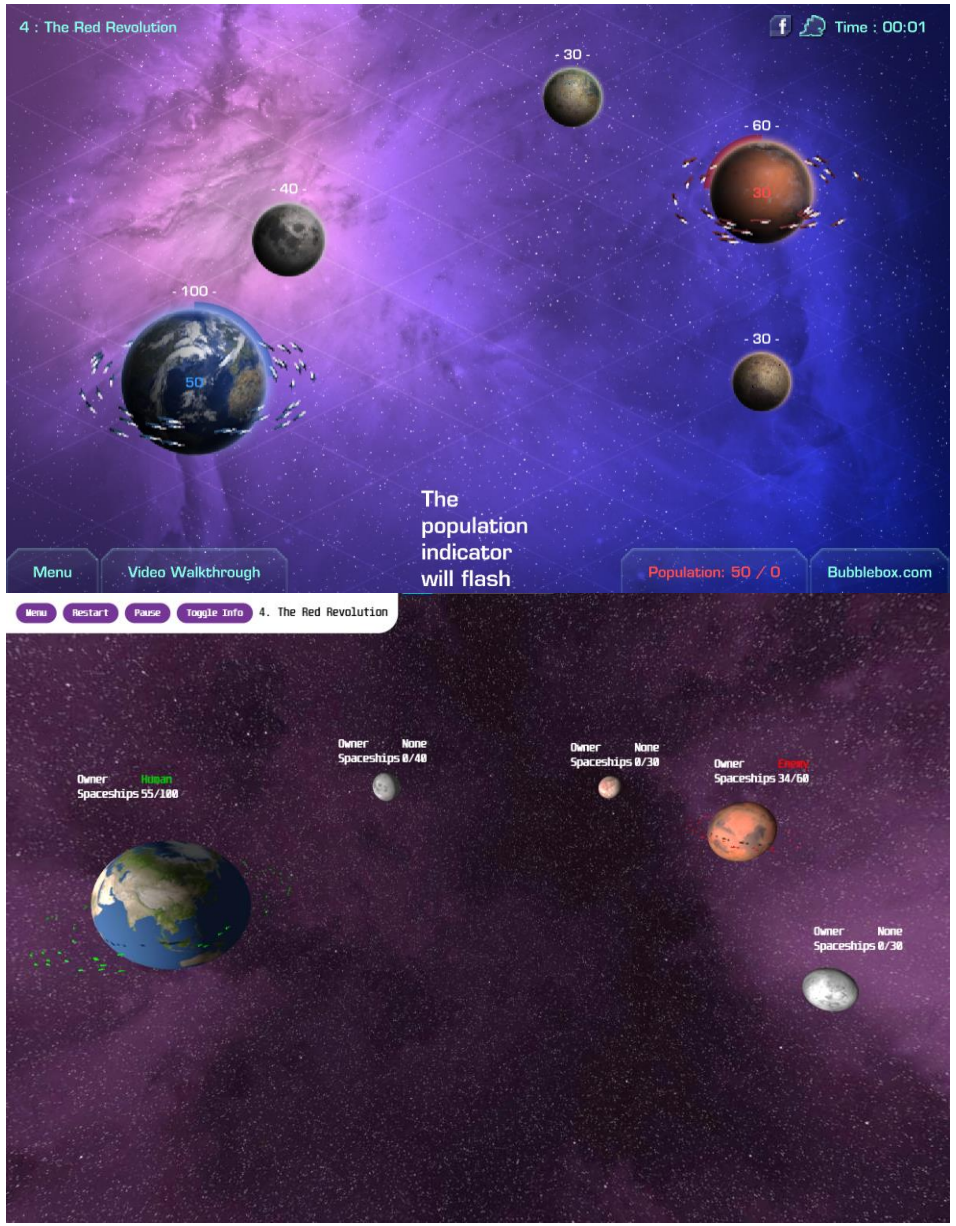
João Fonseca (103154)

Introduction to Computer Graphics – 2022/2023 – Project

Introduction

- Inspired on **Solarmax (2D)**
- Space-themed strategy game
- Players have a fleet of spaceships
- <u>Goal: colonize all planets in the level</u>

- Created using basic Three.js



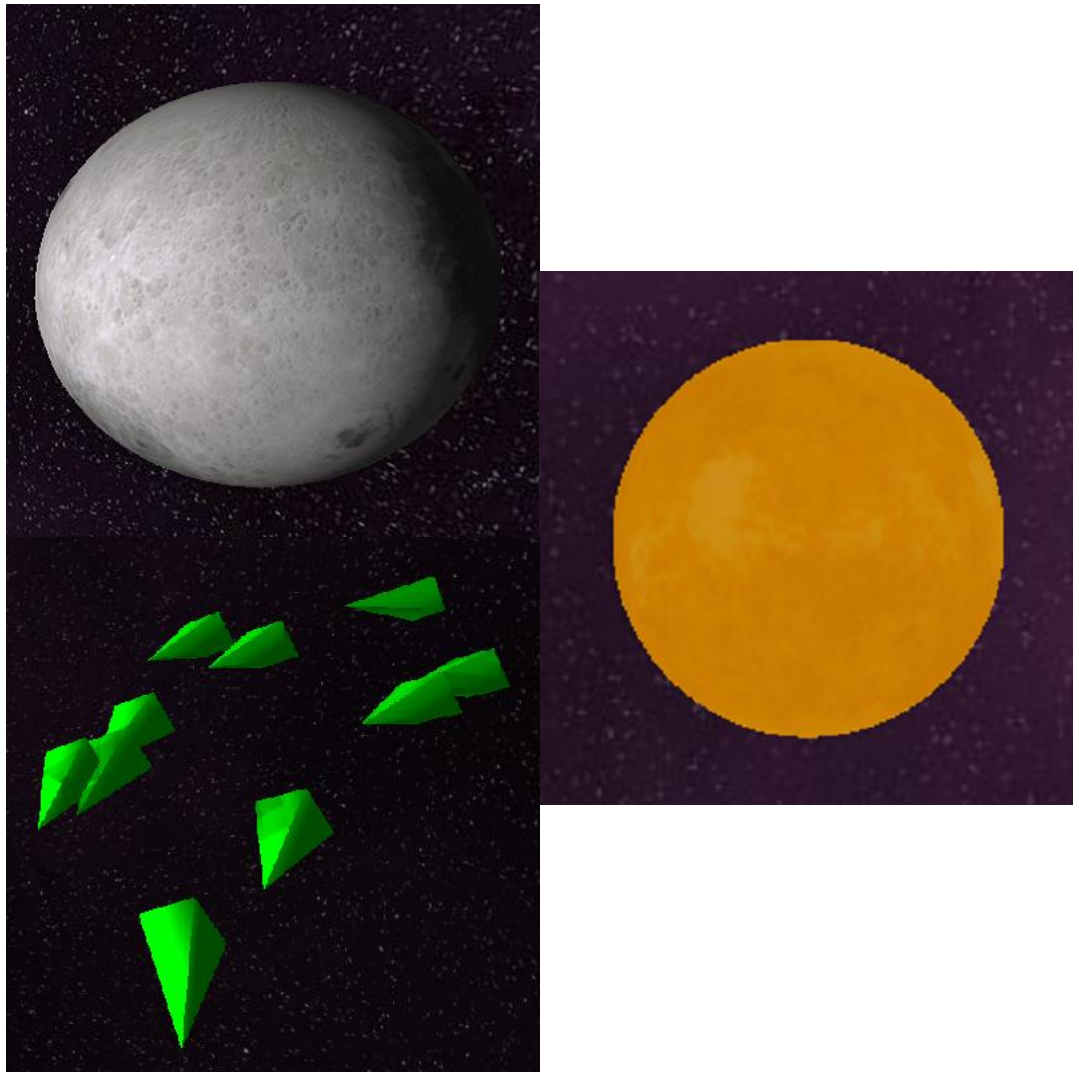
Deployment

https://joaompfonseca.github.io/icg-project/

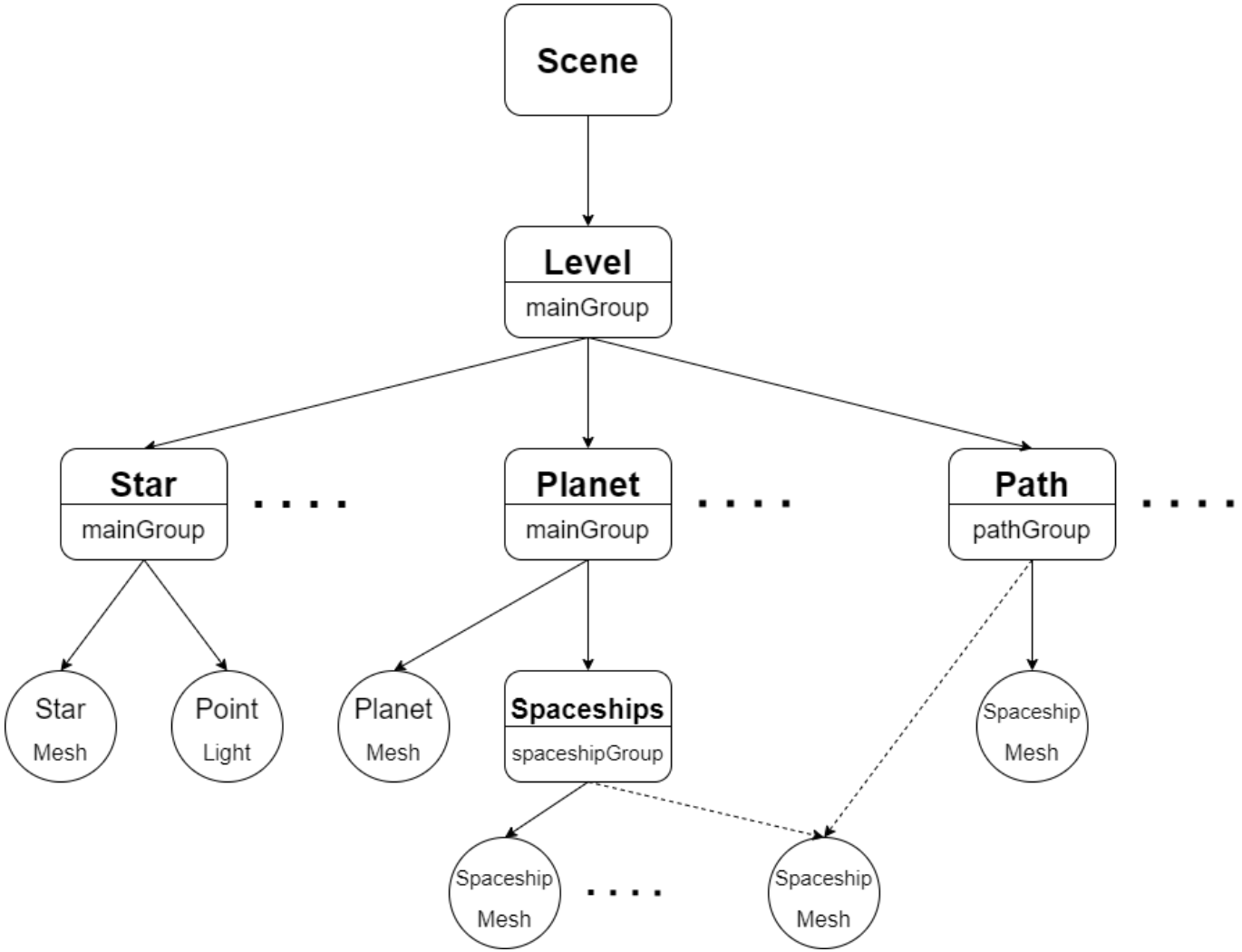
Models

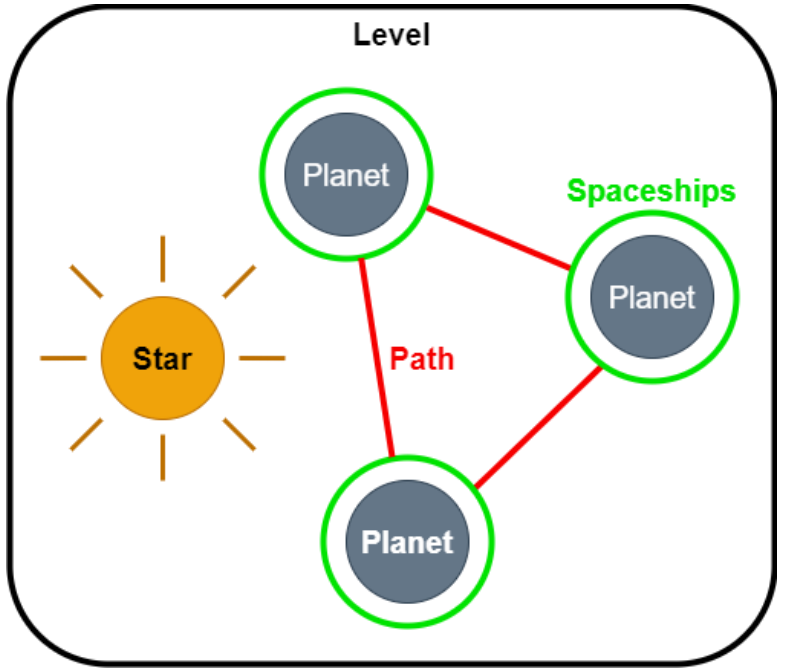
- Planets & Stars
  - SphereGeometry
  - MeshPhongMaterial (+ custom texture)

- Spaceships
  - ConeGeometry
  - MeshPhongMaterial (+ emissive of intensity 0.2)



Scene Graph


Scene
Level
mainGroup
Star
mainGroup
Planet
mainGroup
Path
pathGroup
Star
Mesh
Point
Light
Planet
Mesh
Spaceships
spaceshipGroup
Spaceship
Mesh
Spaceship
Mesh
Spaceship
Mesh


Level
Planet
Spaceships
Planet
Star
Path
Planet

Animation

- Planets & Stars
  - Fixed tilt (around Z)
  - Rotation on own axis (around Y)
- Spaceships
  - Fixed rotation (around X)
  - Orbit around planets / rotation of group (around Y)
- Paths
  - Spacehips points to destination planet
  - Spaceship moves towards planet
- Entities have animate function that is called on each frame

```
this.mesh.rotateZ(tilt);
this.mesh.rotation.y += this.rotationSpeed;
```

Ball#animate

```
this.geometry.rotateX(Math.PI / 2);
this.spaceshipGroup.rotation.y += this.orbitSpeed;
```

Spaceship#animate

```
this.spaceship.lookAt(this.targetBall.position);
const displacement = this.direction.clone().multiplyScalar(this.spaceship.speed);
this.spaceship.mesh.position.add(displacement);
```

Path#animate

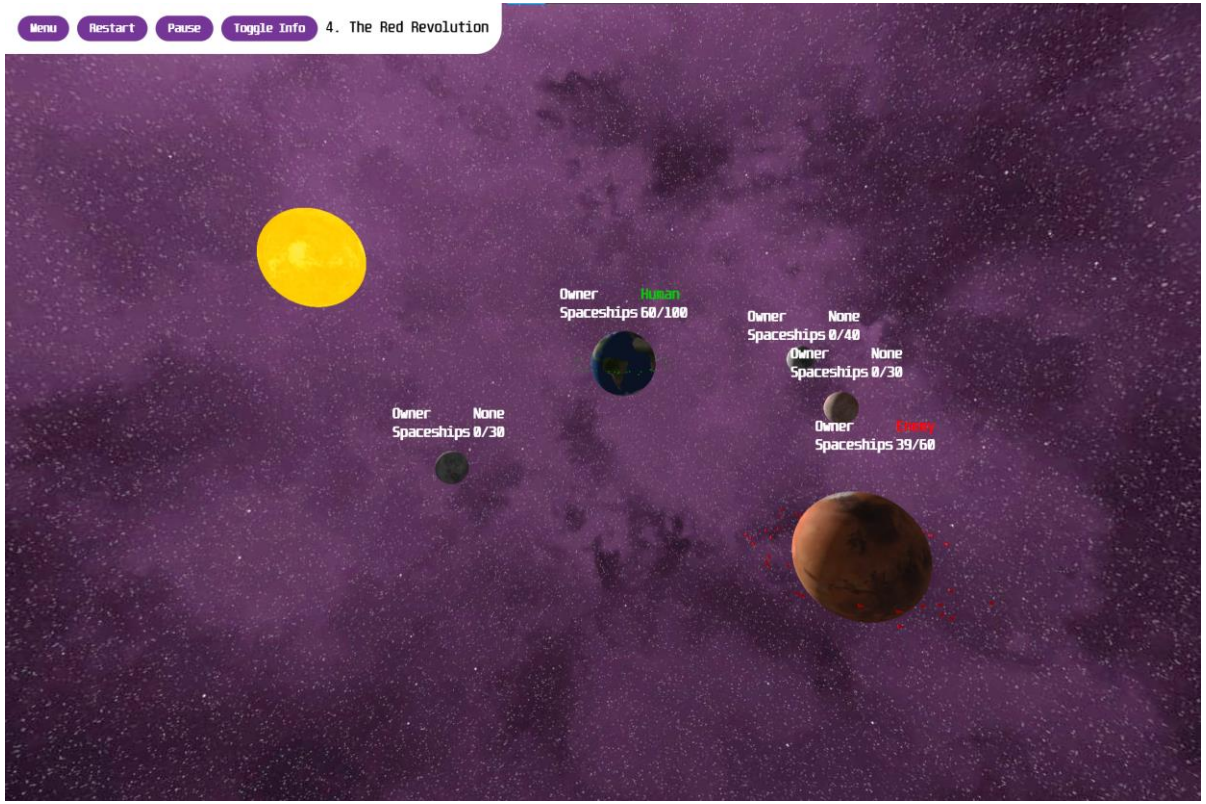
Illumination

- Ambient lighting is present in the environment
  - AmbientLight
  - White light
  - Intensity of 0.1

```
const ambientLight = new THREE.AmbientLight(0xffffff, 0.1);
```

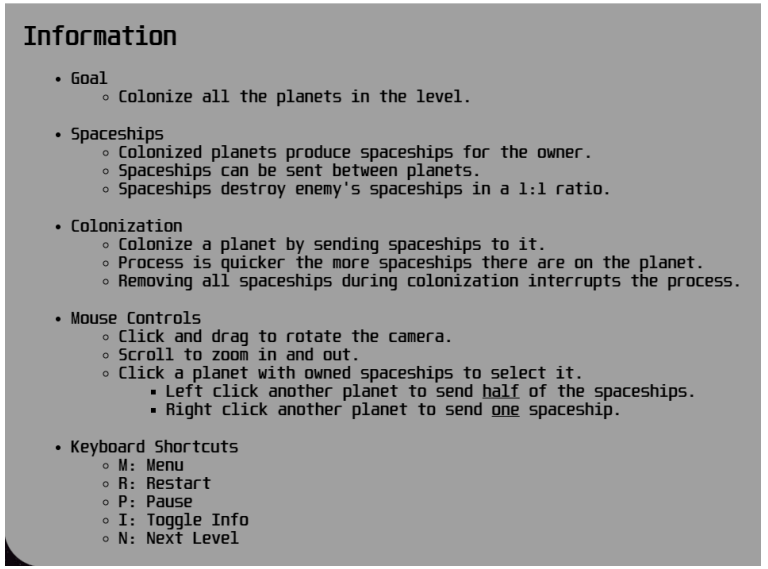
- Star emits light to the entire level
  - PointLight
  - White light
  - Intensity of 0.1
  - Distance of 10000

```
const light = new THREE.PointLight(0xffffff, 1, 10000);
```



- All objects receive and cast shadows between eachother

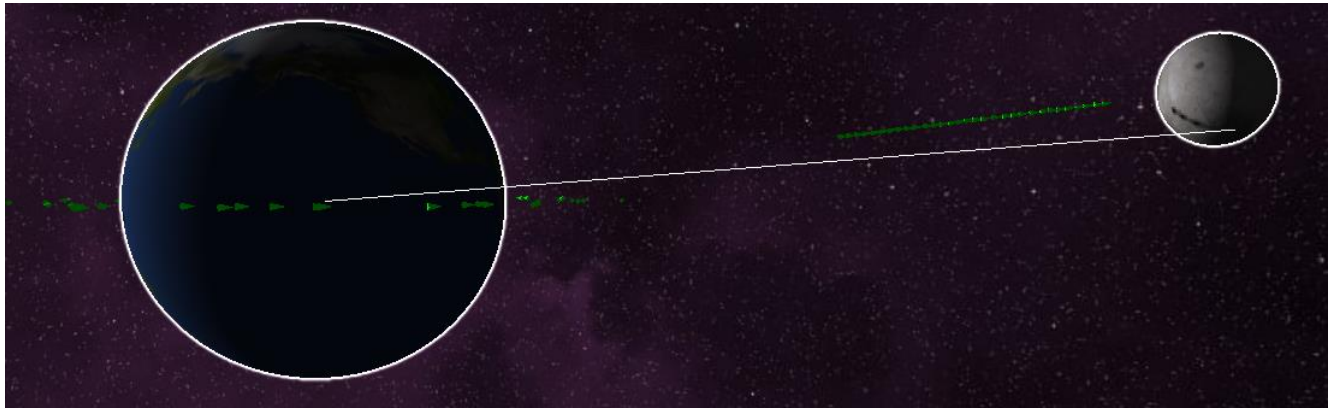
User Interaction



- UI Elements
  - Main menu
  - Navbar buttons
  - Labels above planets
  - Colonization progress
  - Information panel

- Mouse controls
  - OrbitControls: rotation, zoom
  - Raycaster & OutlinePass: interaction with planets

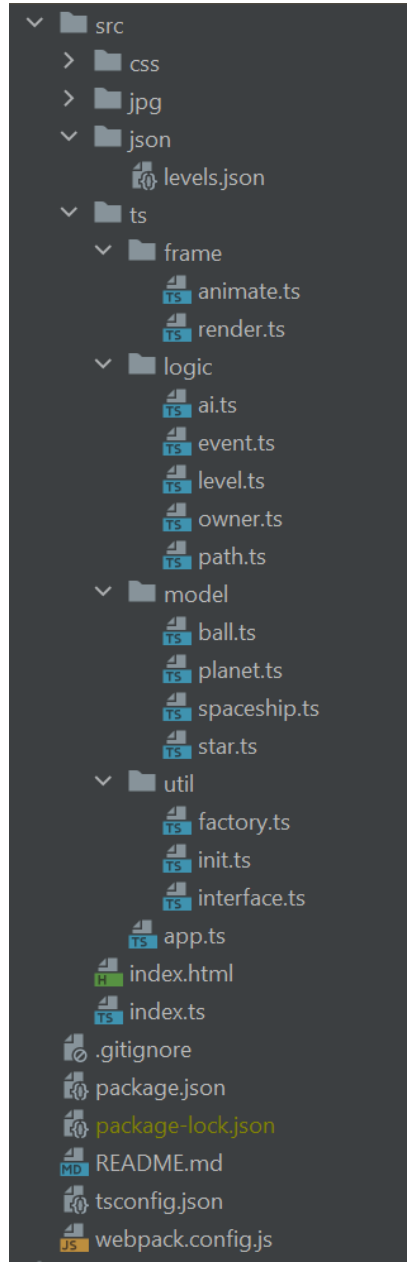
- Keyboard shortcuts
  - M: Menu
  - R: Restart
  - P: Pause
  - I: Toggle Info
  - N: Next Level



Development

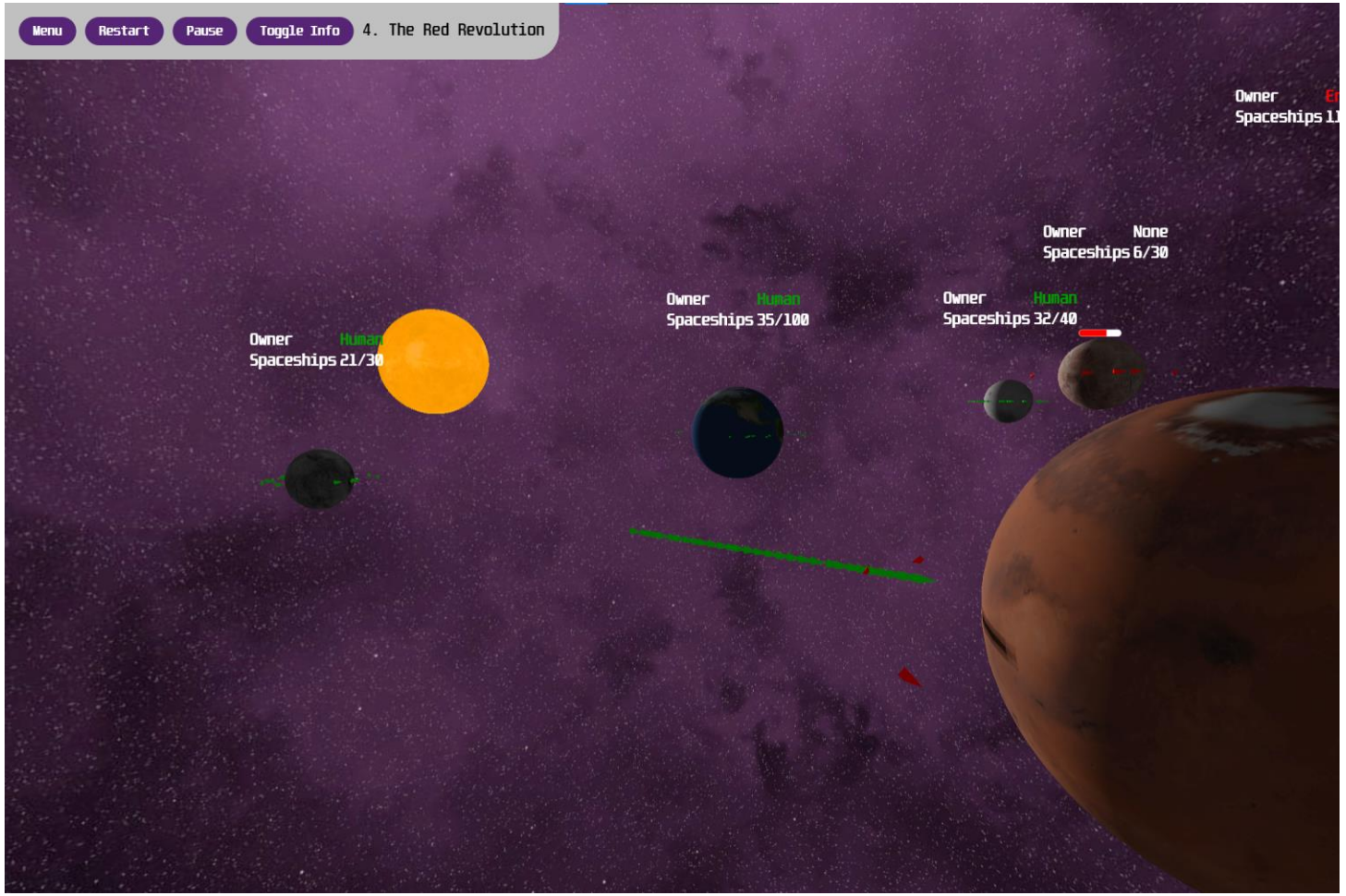
- Coded in **TypeScript** (transpiled to JavaScript)
  - Strong static typing – variables, function parameters and return values
  - ECMAScript support (ES6) – modern JavaScript features
  - Can compile to any JavaScript version
- Follows some OOP practices
  - Classes
  - Design patterns - factory
- Bundled with **Webpack**
  - Optimization of the code
  - **bundle.js ~ 499 KB**
- Deployed in **GitHub Pages**

- <u>Problems: creating UI elements (menu, navbar) using basic HTML, CSS and JS</u>



Conclusions

- Understanding of 3D graphics
- Practical experience with Three.js
- Future applications



References

- Examples from ICG practical exercises
  - https://elearning.ua.pt/
- Three.js Documentation
  - https://threejs.org/docs/
- Solar System Textures
  - https://solarsystemscope.com/textures/
- Tutorial: Hosting a Webpack Project with GH-Pages
  - https://learnhowtoprogram.com/intermediate-javascript/team-week/hosting-a-webpack-project-with-gh-pages
- Project repository on GitHub
  - https://github.com/joaompfonseca/icg-project
- Deployment on GitHub Pages
  - https://joaompfonseca.github.io/icg-project/