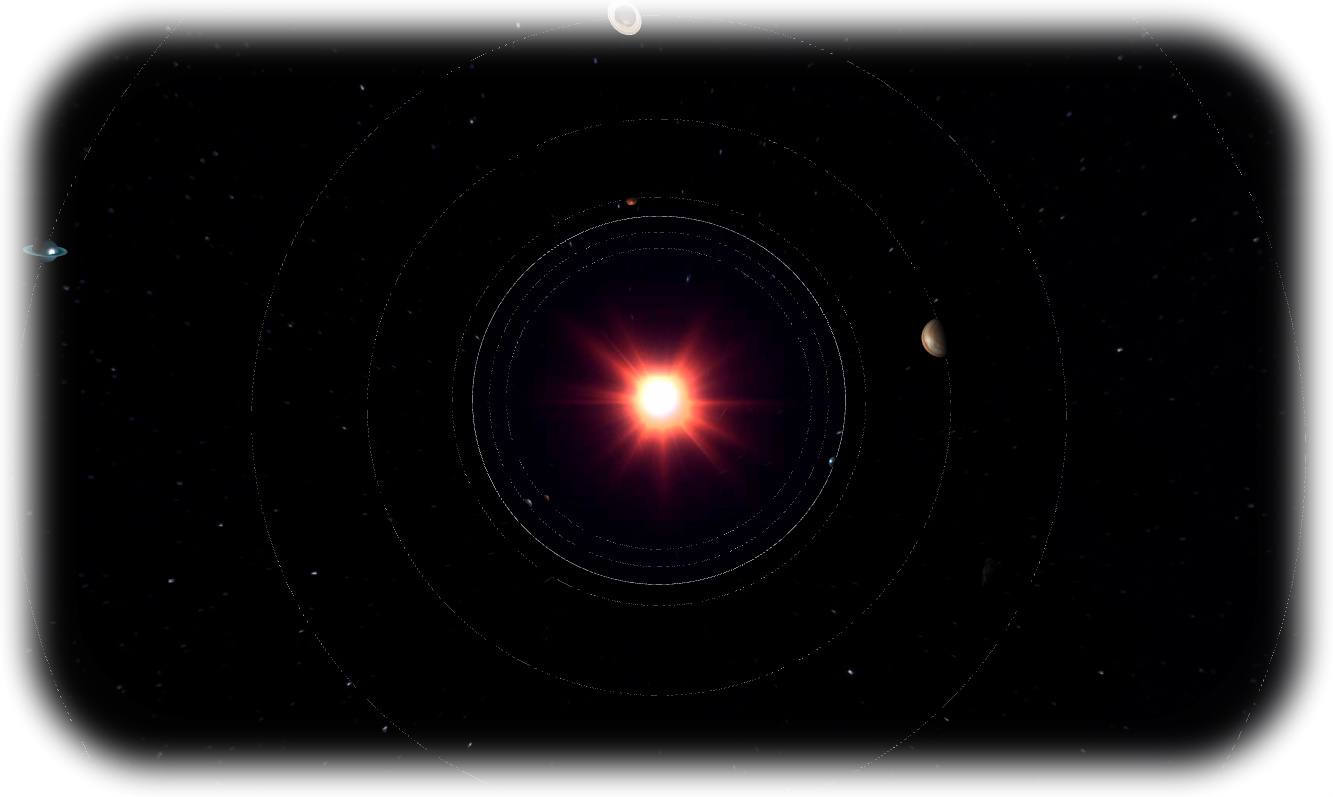
Interactive Solar System Project - Interactive Graphics

By Manuel Mandatori, Hossein Hashemi, Carlo Gabrielli



**Table of content**

1. Introduction
2. User Manual
   1. Views and Controls
   2. Other Interactions
3. Environment, libraries, and tools description
4. Code break-down
5. **Introduction**  
     
   Our project is a 3D reconstruction of our Solar System and is characterized by realistic fully textured planets that follow their own orbit reproduction around the sun.

The planets are are structured by different features:

|  |  |
| --- | --- |
| **1.** **High resolution 4K textures:** | Most planets in our virtual solar system are characterized by 4k textures, unfortunately not those too far from Earth, due to the inability to take such detailed pictures of far away planets. |
| **2.** **Bump textures:** | Same as for the 4k textures, bump textures are available for most planets but not all, to showcase inbalances in the terrain and add to the realism. |
| **3. Dynamic shadows:** | Planets are affected by the sunlight and will display the shadows created by the latter as they circle around the sun. |
| **3.** **Atmospheres and clouds:** | Clouds and atmospehere reproductions are available for the system’s planets that actually have one, rotating around the planet, but not around the same axis of rotation of the planet itself. |
| **4.** **Rings and respective textures:** | Our project features the rings around Jupiter and Saturn as well as their respective textures. |
| **5.** **Realistic planet sizes:** | The planet sizes reflect the real size of the original planets, with some exceptions, to ensure that all planets are visible in our system, no matter how small, while still reflecting a good level of realism. |
| **6.** **Realistic planet distance:** | Each planet’s orbit route reflects the real distance each original planet has from the Sun and from each other. |
| **7.** **Realistic periods of revolution:** | Each planet follows realistic periods of revolution with different orbit velocities to better represent the original system. |
| **8. Moving spaceships:** | The system is also enriched by two fully textured moving spaceships that orbit around the system around their own trajectory. |

1. **User Manual**

The project includes two different views of the solar system, as well as various types of interactions available for the user to utilize. This section will help understand the underlying differences between the views and the various ways of interacting with both.

* 1. Views and Controls

Our project includes 3 different scenes, all of which are interactable and provide different views of the system with different controls:

|  |
| --- |
| System View  C:\Users\Carlo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\5d581d36cf3afcc9717d8c3a65f1a2b4.png |
| The entire system is viewable, with camera lock on the sun. All planets are animated and fill follow their orbit around the sun. The sun acts as a point light, radiating light in every direction and reflecting on the planets. The background is composed by a sphere that inglobes the whole system and is characterized by a texture that resembles space and stars. |
|  |

|  |
| --- |
| Earth View |
| The focus and the camera lock are shifted toplanet Earth, to allow the user to get a closer look. This allows us to showcase most of what we talked about in the introduction and that is more difficult to see from a distance. The main features of this view are the: the atmosphere, clouds, and bump textures that become more apparent and more detailed. The controls remain the same as for the previous view. |
|  |

|  |
| --- |
| Spaceship View  C:\Users\Carlo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\d412b2aea1cdd1d54cb8d34aaba10b74.png |
| Very similar to the System view, with all planets in the system moving along their own orbit, but with a 3rd person view locked onto one of the spaceships, that can be moved and driven around the system as the user pleases. The spaceship is capable of moving in every direction through the ability of various translations and rotations. |
| C:\Users\Carlo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\keyboard_layout_en-US.PNG |

* 1. Other interactions

Other than the controls and interactions we already covered to be able to easily control both scenes, a couple of controls were added to facilitate the user’s experience while using our solar system.

|  |  |
| --- | --- |
|  | |
| In the top-right hand corner of both scenes a “Controls” tab is present, and offers 5 different interactions: | |
| **1. earthView** | A checkbox used to be able to switch to the *Earth view* without having to manually input the file location, a simple click will suffice to switch to the other view automatically. |
| **2. spaceshipView:** | Similarly, a checkbox used to switch to the *3rd person view* of the spaceship. |
| **3. solarSystemView:** | The checkbox used to switch to the *Solar System view.* |
| **4. extraLight:** | A second type of checkbox only available in the **System View,** allows the user to turn on a secondary light (other than the sun) to shed more light on the planets, which can sometimes be difficult to see due to their small size. |
| **5. Close Controls:** | Allows the user to hide or show the two checkboxes at his own discretion. |

1. **Environment, tools, and libraries**

The environment we decided to work with is **Three.js,**  a cross-browser JavaScript library/API used to create and display animated 3D computer graphics in a web browser. This allows us to create our project animations using the JavaScript language and the WebGL that Three.js uses.

To further improve our work we took inspiration from the **“**[**jsOrrery javascript solar system simulator**](http://mgvez.github.io/jsorrery/)**”** for the visual aspects

To allow the project to correctly function, we utilised some libraries that helped us with a few things, particularly with the controls, a list of these follows:

|  |  |
| --- | --- |
| Libraries | |
| * **Detector.js** | This is used for general rendering. |
| * **Dat.gui** | This is used for the Controls tab, and allows us to use the checkboxes. |
| * **Obj.loader.js** | This is used to load the spaceship models in both the Solar System and Spaceship views. |
| * **MTLLoader.js** | This is used to load the spaceship textures. |
| * **THREEx.KeyboardState.js** | This is used for the spaceship’s controls, and can be modified to change keybindings. |
| * **THREEx.FullScreen.js** | This is used to load the project in fullscreen mode if the user chooses to do so. |
| * **THREEx.WindowResize.js** | This is used to automatically detect the screen size and resize the window accordingly, when using fullscreen. |
| * **TrackballControls.js** | This is used to control mouse movements and rotations for the Earth and Solar System views. |

4. Environment, tools, and libraries

**Env**: Three.js + Apache server

**Libraries:**

Detector.js for rendering;

Dat.gui, for options tab;

Obj.loader.js to load spaceships;

MTLLoader.js to load spaceships textures;

THREEx.KeyboardState.js for the spaceship’s controls;

THREEx.FullScreen.js to load the view in fullscreen;

THREEx.WindowResize.js to resize the window;  
TrackballControls.js to control mouse movements;

**TOOLS**: JSOrrey javascript solar system simulator

**SetPlanets.js**

**CreatePlanets.js**

**SolarSystem.js**

**Index2.html**

**Index.html**

**SolarEarth.js**

**INDEX2**

**59-78 -> loadModel function**

Here we attach a texture to an object, settings the scale and position. MovingCube is the name of the spaceship.

**141-152**

To set the lens flare of the sun

**190-223 -> Update function**

Here we manage the movements and rotation from the keyboard controls

**CreatePlanet.js**

Here we create the various planets with a THREE.SphereGeometry and setting a THREE.MeshPhongMaterial (to better reflect the light). Then we load the textures

**SetPlanets.js**

Here we add the rotations, starting points, clouds and a TorusGeometry in order to move the planet around itself and the sun.

• The accompanying document should be both a technical presentation and a user manual and contain:

– Description of the environment used (basic WebGL or other) Three.js

– List of all the libraries, tools and models used in the project but not developed by the team

– Description of all the technical aspects of the project

– Description of the implemented interactions

some general information that we can describe for professor, in base we use three.js library , also for made spaceship we use 2 JS library for load the model that name of file is MTLloader.js and OBJloader.js and for error detection we use detector.js library.

the main of our project it's in the solarsystem.js file , that we made all solar system with Planet's ,spaceship's and all action's.  
  
  
1- Introduction to the project and description of it's functionalities

2- Description of Three.js and list of libraries, etc.

3- Code break-down and explanation

4- User Manual