Interactive Graphics - Final Project
Solar System

Gianmarco Cariggi 1698481
Marco Costa 1691388

10th July 2019

1 Introduction

For the final project of the course of Interactive Graphics we decided to make a simulator of our solar system, trying to reproduce it in the most faithful and realistic way possible, using of course assumptions and approximations to simplify its implementation.

1.1 Requirements

- Hierarchical models

- Lights

- Textures

- User interaction

- Animation

1.2 Environment

The project is developed in WebGL (Web Graphics Library). It's a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser without the use of plug-ins. It is fully integrated with other web standards, allowing GPU-accelerated usage of physics and image processing and effects as part of the web page canvas.
Click here to check if your browser supports WebGL.

1.3 Libraries not developed by the team

- **Three Js** (rev. 107dev): a cross-browser JavaScript library and Application Programming Interface (API) used to create and display animated 3D computer graphics in a web browser using *WebGL*.

- **Materialize** (v. 1.0.0): a design language that combines the classic principles of successful design along with innovation and technology.

- **jQuery** (v. 3.4.1): it takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code. It also simplifies a lot of the complicated things from *JavaScript*, like *AJAX* calls and *DOM* manipulation.

1.4 Assumptions and approximations

As is well known, the planets follow an elliptical orbit (a "flattened" circumference), and the sun occupies one of the two focuses. Rotating along this orbit, we have the 4 seasons (also due to the inclination of the axis). For simplicity of implementation and positioning of the various planets, we have approximated the orbit with a circumference, and the sun occupies centre.

Another important fact is that the orbit of all the planets varies continuously (for example, the orbit of the earth is not the same from one year to the next), this is because there is the gravitational influence of all the celestial bodies present in the universe (and in particular in the solar system, being closer). In the solar system, besides the sun, Jupiter has an important influence, being the biggest planet. In our simulator we do not take into account all this, the planets revolve around the sun with the same orbit (we can say that the sun is the only body that influences the gravitation of the planets).

The last detail is the actual position of the planets. Probably they will not be located in the real position, because in our simulator this position depends on the date, and in JavaScript the function **Date.getTime()** returns the milliseconds passed from January 1, 1970 to the moment in which the function is called. In theory we should have the milliseconds passed from the birth of the solar system to today, or set a certain initial position so as to match the real position of each planet with that in the simulator. But this is very complex, so we preferred to concentrate on other things. It must be noted however that every planet makes a turn around the sun in the real time of revolution and makes a turn around its axis (inclined with respect to the equatorial plane) in the real time of rotation (for example the earth takes exactly one year to make the revolution around the sun and 24 hours to make a turn on itself).
Here it's possible to see the real position of the planets.

2   Usage

The graphic interface allows to client to interact with our projects. First of all there is a fast loader page meanwhile the browser loads all the data.

At the beginning camera is focused on Sun and the *follow-planet* flag is enabled, then you can only move camera around the sun. In order to move in the solar system, you have to disable it through the main menu.

In meanwhile a background music starts in order to create a pleasant atmosphere and to take a good travel around the solar system. If you don't like it, you can disable through the specific button at bottom right of the window.

Since you disable the *follow-planet*, now you can move.

Have a good trip!

2.1   Mouse Controls

The mouse is very important to fully enjoy this experience, so we explain the available mouse controls below:

- **Left Click**: Normally you click button or elements in the menu. If menu is opened and click on the solar system, it will close. If you click and drag the camera orbits around the target point.

- **Double Left Click**: If it's done over a celestial objects (except *Asteroid Belt*), it permits to select and focus the camera on it.

- **Right Click**: If the *follow-planet* flag is disabled, you can move around the environment clicking and dragging.

- **Scroll Wheel**: As usual, the scroll wheel permits to zoom in or zoom out on camera's target.

- **Move on the object**: Moving over the planets, a pop-up window appears where there are all planet's physical data.

All the commands are possible through the using of *OrbitControls* of *Three JS*.

2.2   Menu Options

The menu offers to you a lot of options to custom the animation and the aspect of solar system. It can be opened or closed simply clicking on the relative button in the top right of the screen. So let to describe all the options:

- **Play / Pause Animation**: Play or pause all animations of the solar system. (Default: Play)

- **Rotation**: Play or pause only the rotation on the axis of planet. (Default: Play)

- **Revolution**: Play or pause only the revolution orbit. (Default: Play)

- **Trajectories**: Show / Hide the orbit trajectories. (Default: ON)
- **Inclined orbit**: Set / Unset the real inclination of the planet's orbit. (Default: ON)
- **Date**: Set a specific date to see the state of solar. system.
- **Time**: Set a specific time to see the state of solar system.
- **Speed**: Set the speed of time flowing (and then the animation). (Default: 1.0x)
- **Far**: Set the far parameter of camera. (Default: 10000).
- **Camera**: Set the camera target planet (as double click on a planet). (Default: Sun)
- **Follow**: Camera follows or not the target planet. (Default: ON)
- **Rotate Camera**: Camera orbits around the target planet. (Default: OFF)
- **Earth clouds**: Show / Hide the Earth's clouds. (Default: OFF)
- **Asteroid Belt**: Show / Hide the asteroid belt. (Default: OFF)
- **Ambient light**: Turn on / off the ambient light. (Default: OFF)
- **Sun light**: Turn on / off the Sun light. (Default: ON)
- **Sun glow**: Turn on / off the Sun glow. (Default: ON)
- **Sun light intensity**: Set the sun light intensity (Default: 1.5)
- **Wallpaper**: Choose the background.
- **Track**: Choose the background music.
- **Volume**: Set the volume of track. (Default: 0.3)

3   Implementation

Appendices

A   Sources

- Wikipedia - for data and descriptions of the celestial objects.
- Threex.planets.js - for loading earth clouds texture and have a ring geometry with a radius texture
- Planet pixel emporium - for textures