Montagnes, Luzzatto

CSE104 – Computer Programming: Project

We created the interactive game 'Discovery', where the player can explore the Solar System.

When we started looking for an idea for this project, our first ambition was to think about something that could potentially be useful in the daily life, such as a website dedicated to activities organized on Ecole Polytechnique's campus, or a sort of application meant to help with some basic task. However, then, we thought: how about creating something esthetically stunning, maybe less useful on a daily basis, but which would make one realize how beautiful is the universe surrounding us?

That is why we created the game *Discovery*. The goal of the game is to explore the Solar System, which was recreated in three dimensions, taking, roughly but realistically enough into account the main proportions defining the system.

The way one can play is detailed in the menu section of our page, which briefly presents the game and describes the rules:

- "Choose the shuttle which will lead you to the end of the world";
- "Move the arrows to move the spaceship. Move the mouse to look around you";
- "Press the *spacebar* to pause the game. Press the *tab* to open the menu";

The menu section was realized with basic HTML and CSS tools: the menu on the top bar was created with boxes inside a list, which references (hrefs) to the other possible pages (from the 'Home', one can open the 'Rules', an 'About' section and, obviously, the access to the game). The pointer is a small tribute to the recent, first image of a black hole, using a simple JS function seen during the tutorials.

The game itself is realized in a three dimensions simulation, performed with *Three.js*[1], a JavaScript library and, essentially, an Application Programming Interface (API), to draw and render objects in 3D inside a web browser. It is complementary to the Web-based Graphics Library (WebGL)[3], an HTML graphic library. To learn the basics of the APIs, we used, between others, the site of our professor D. Rohmer[2].

When starting the game, the user has to choose both the background he wants to use (one can explore in a 'Night mode', more realistic, which recreates the space environment with low lights and the various galaxies in the background, and a 'Day mode', more esthetical, which allows the players to discover the Solar system in a bright and science-fiction-like model) and the shuttle he wants to pilot. The rotating choice menu was created with the Three's library Collada loader, that we took entirely on internet, leaving the credits.

The backgrounds are created and rendered in 3D thanks to CubeTextureLoader[5], a Three.js class to load a cube texture, which would enclose the whole game later. We then load pictures inside it, and the library renders it in the full window (using the setPixelRatio and the setSize methods of the Three rendering, ensuring this way that the planets keep fixed ratios even when the window sizes change).

Then, the game's core is made of the planets, which are realized with a simple, reiterated structure: we created a new object with Three, using the SphereGeometry class[6], then we set the position of the object in the absolute frame of reference, where (0,0,0) is the point where the journey starts. Consequently, we modified the textures of the planets by downloading them from a specific website[7]. Finally, we make it appear in the virtual space.

The idea of the game is to use the FirstPersonControls JS file, hosted in a repository on GitHub[8] to move inside the virtual space, here, in particular, using the arrows.

Finally, we define the camera, which is connected to the spaceship chosen in the previous menu, we scale it with respect to the space's dimensions, and we finally define the movement speed and the look speed, which allow us to give birth to the Scene[9], as a class in Three.

The animation is realized using concepts learned during the CSE102 – Computer Programming class, with the notion of refresh date and tracking times (realized with the help of Three.Clock()[10]), rendering after each request of animation.

Finally, we implemented new features such as the in-game menu, which allows to come back to the Home menu, to restart at the origin point, or even to change the movement speed of the shuttle a slider[11,12].

- [1]Source of the library Three.js: https://threejs.org
- [2]Three and WebGL introductory courses: https://imagecomputing.net/damien.rohmer/
- [3]Resources useful to render WebGL: https://threejs.org/docs/#api/en/renderers/WebGLRenderer
- [4]Collada loader: https://threejs.org/examples/webgl_loader_collada.html
- [5]Cube texture loader: https://threejs.org/docs/#api/en/loaders/CubeTextureLoader
- [6]Sphere geometry class: https://www.solarsystemscope.com/textures/
- [7]Solar system textures: https://www.solarsystemscope.com/textures/
- [8]First-person controls: https://threejs.org/docs/#api/en/geometries/SphereGeometry
- [9]Three's scene: https://threejs.org/docs/#api/en/scenes/Scene
- [10]Three's clock: https://threejs.org/docs/#api/en/core/Clock
- [11]Drop-down menu: https://www.w3schools.com/howto/howto_css_dropdown.asp
- [12]Range slider: tutorial on JS and https://www.w3schools.com/howto/howto_js_rangeslider.asp