Final project Interactive Graphics

Master’s degree in Artificial Intelligence and Robotics

Course: Interactive Graphics

Professor: Marco Schaerf

Author: Davide Palleschi – 1697259

SAPIENZA
UNIVERSITÀ DI ROMA

La Sapienza University - Roma

Academic year: 2020-2021

Sommario

1. Three.js, Tween.js 3
2. Environment And Entities 4
   2.1 Map 4
   2.2 Character and enemies 5
3. Interface and Configuration 6
4. Levels 7
5. Gameplay 8
6. Compatible Browsers 8

1. Three.js, Tween.js

In order to do the project, I used the following libraries:

1. Three.js
2. Tween.js

Three.js is an open-source library that is based on WebGL. Three.js contains a lot of features such as:

- Scene that allows me to add/remove the 3D objects.
- Cameras (perspective and orthographic).
- Lights and shadows.
- Materials: Lambert, Phong, etc.…
- Objects: geometry, meshes, etc.…
- Geometry primitives: cubes, planes, spheres, cylinders.
- Loaders: in order to load my model and textures.

In my project I have built the environment and the character by using cubes, cylinders and sphere.

Tween.js library is a simple interpolation library useful for basic animation like the one in this project.

I used this library for animate the camera when a level starts and ends, animate the character using the yoyo option and animate the enemy cubes.

2. Environment And Entities

2.1 Map

For the creation of the map, I used the following objects:

- a box to which I applied a color texture and a normalized texture to create the lawn
- a tree model from the internet
- a bench model still taken from the web

In detail:

The tree:

I take the tree model from the website CGTrader, I used the .obj file because they are simple to manage. I created and added 289 of these trees to the scene and then moved them above and below the floor to create the different levels of the game

The bench:
I take the bench from the same website of the tree, I added some of these to the scene to make it richer.

The result:

2.2 Character and enemies

As for the creation of the character I decided to create it locally and not to import it from the internet, this above all to maintain acceptable performance having already added many models for the creation of the levels and to be able to control it more easily.

The character is a hierarchical model composed of 26 parts, I also decided to add a sphere to each connection so as not to see gaps between the different parts, also it was useful to use these 'joints' as they can be used in animations to make them more understandable.

For the creation of the enemies, I decided to use simple boxes that emit light , this once again to try to maintain high performance.

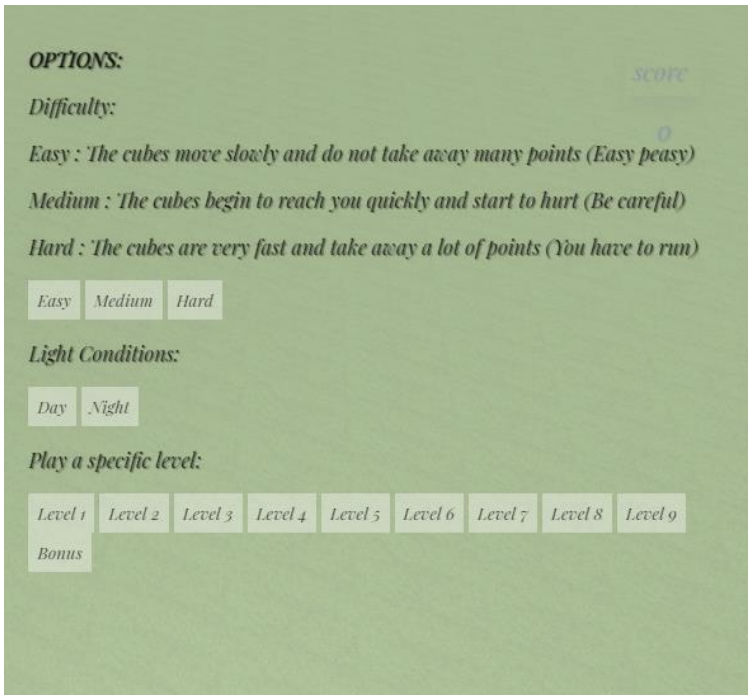
For the same reason I finally decided to create only 3 instead of 7 initially present.

3. Interface and Configuration

For the interface of the game and the configurations I decided not to create a special page but to insert buttons and boxes in the game itself, by doing so it is possible to see the variations in real time on the game.

At the beginning of the game, we will find the button to start playing, the button to access the options and the button to read the instructions of the game.

During the game there is also a button to activate and deactivate the music and a box where you can see the current score.



The options box (where you can change the difficulty of the game, the lighting conditions and the level to be faced) is present both at the beginning of the game and at the end of each level.

Difficulty:

To vary the difficulty of the game I decided to use the enemy cubes, increasing the difficulty these become faster and therefore will reach the character more easily and take away more points. The score increases as you run but if the total score at the end of the level is negative you will lose the level.

Light Conditions:

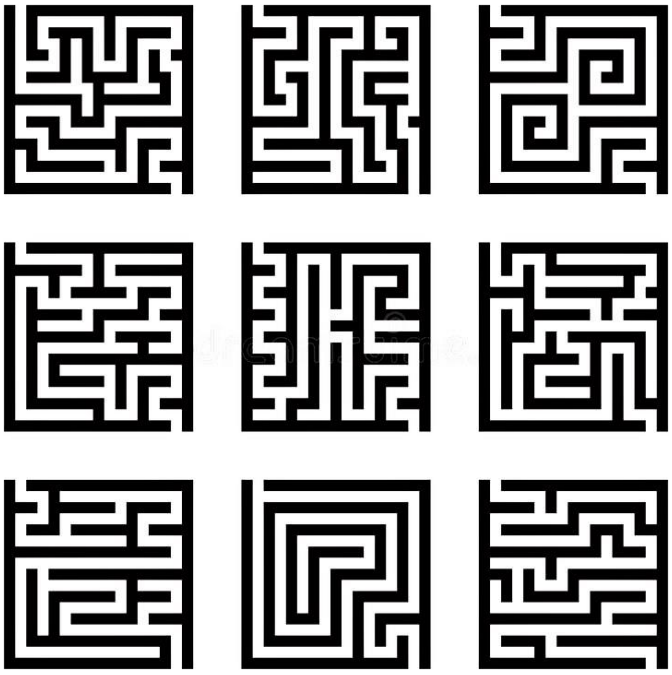
1) Day: we have a directional light to simulate the sun, an ambient light and a hemisphere light.
2) Night: we have only a directional light with a white color to simulate the moon.

The map in day light conditions

The map in night light conditions

4. Levels

The game consists of 9 levels, all with the same difficulty. The passage from one level to another is carried out by lowering the trees below the ground.



This image contains the levels present in the game, the only difference is in the entrance of the maze which in the game has been closed to not allow the player to bypass the level.

Beyond these levels I have decided to add a bonus level in which the player can move freely around the map and in which the trees form the word “THE END”.

5. Gameplay

Regarding the game mechanics I decided not to use a physics engine as the only thing I needed was basic collision detection. I therefore decided to simply calculate the distance between the character and the elements of interest.

The commands to control the character are:

W: Move forward

A: Move leftward

S: Move backward

D: Move rightward

6. Compatible Browsers

I tested the game on the main existing browsers.

I noticed that the best for the performance on the Windows OS is Edge followed by Chrome and lastly Firefox on which the game tends to lag.

While for the MacOS Chrome seems to be more performing than Safari.