Final project Interactive graphics

Edoardo Montecchiani **Luca Maurici** **Lorenzo Mattia**

1. Introduction

The project we have developed is an FPS game in which the player is in a fantasy city and its goal is to kill all the enemies, ninjas trained to murder him. In the map is present also a spider boss to make things more difficult. In order to play, you just need to click the mouse to shot, use WASD to move and the space bar to jump; more details about other commands will be given in the game's home page.

2. Libraries and code aspects

The project is based on the WebGL library and using some support libraries such as:

- Three.js is the main library we have used that allow us to raise the abstraction level with respect to the previous projects;
- Three.GLTFLoader.js used to load external GLTF model;
- Tween.js used to realize the animations of the entities in our project;
- Cannon.js that is the physics engine.

All the code is organized in different files and classes in order to handle it better:

- main.js is the most important file in which all the game aspects are initialized, in particular: it saves all the settings parameters, creates the map and handles the winning and the defeat
- EntityManager is the class responsible for the creation and handling of all the game entities. With the word entity we mean the union between the mesh (graphic part of the entity), the body (physic part of the entity) and a basic controller.
- BulletManager is the class responsible for the creation and handling of the bullets.
- Factories (in particular we have the Boss and the Character factories) are responsible for the creation of the mesh
- Controllers, in particular we have the BossAI, BasicAI to handle the artificial intelligence of the boss and the enemies, and Character and Input controllers which handle the player movement.
- ScoreManager, which interacts with the GUI to update the score (number of kills) and the lives of the player.

3. Map and entities

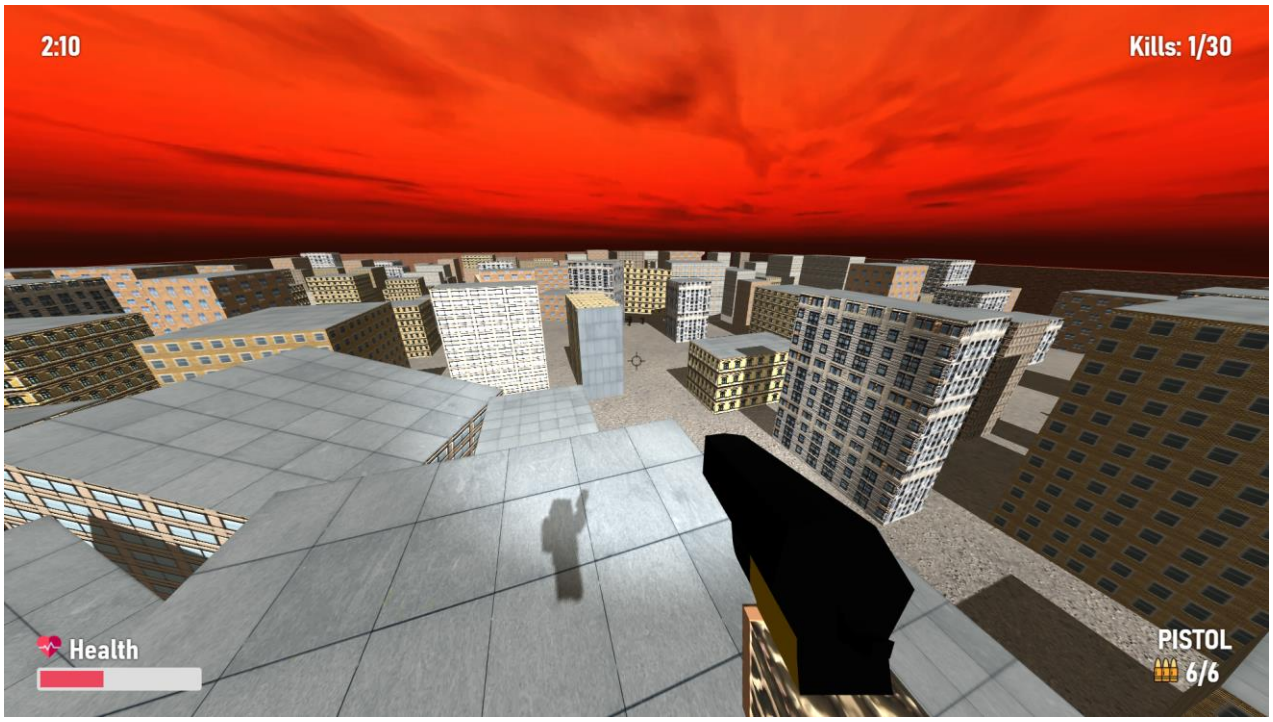
3.1 Map

The map is composed of a plane, representing the floor, some random dimensions obstacles (parallelepipeds) spawned according to a simple generation algorithm, and some walls around the game arena. In addition, for aesthetic purposes, we added a skybox that can assume different textures, depending on the time of day set.

In order to create a new map at each game, we use a generation algorithm that spawns enemies and obstacles (the parallelepipeds) in such a way that no overlaps can occur.

Each time a new game object is generated, we randomly generate its coordinates and, in the case of an obstacle, its extension; and we check whether it overlaps with any object already generated. If it is safe, the object is generated, otherwise other coordinates and extensions are generated.

At the end of this procedure, we obtain a correctly generated and unique map.



The image depicts the map with textures and sunset. Some buildings are falling down because the boss is chasing us and he is able to move them. The HUD can also be seen in this photo. Camera view is set to first person mode.

3.2 Models

The only external models we have used are for the characters weapons and have been downloaded from *sketchfab*.

AK-47:

High number of munitions and shot rate.

Pistol:

Few ammos available and medium shot rate.

RPG:

Only one ammo available, very slow, but it can move and fall parallelepipeds.

Sniper:

It does not have many ammos, but they are very fast. Low shot rate.

3.3 Entities

In the game are present three different kind of entities: the player, the enemies and the boss. They are all hierarchical models and in particular the first two share the same structure (humanoid appearance).

Player:

It is a very simple hierarchical model composed of a head, a body, two arms and two legs. Its movements are controlled by keyboard input. It can use all the four weapons shown above.

Enemy:

It shares the same hierarchical structure of the player (same factory in the code). Its movements are controlled by a simple algorithm: after being triggered it moves towards the player until a certain distance and from there starts to shot to the player while moving randomly laterally. Each enemy has only one weapon and is not allowed to use the RPG.

Boss:

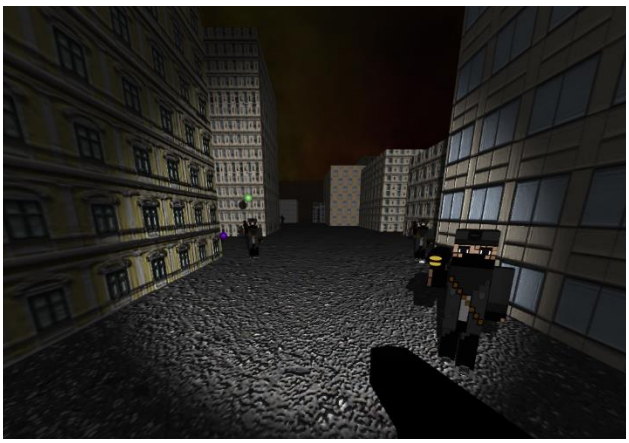
It has a hierarchical structure consisting of a head, a neck, a body and 8 legs divided into upper and lower parts. Once triggered the spider will chase the player by moving everything in between, including the parallelepipeds. Once hit, the player will jump backwards and lose a life. Killing the spider will require several hits.

4. Technical Aspects

4.1 Light and shadow

The light factory class manages the generation of lights according to the chosen time of day. All times using a more or less powerful HemisphereLight of different colours instead of an ambient light as it gives a more pleasant effect. For daytime and sunset the sun in the skybox is simulated with a directional light coming from the same direction, which will generate a shadow on each object in the scene. In addition there is a light simulating a torch in the weapon, which will move with the

4.1 Detail of light and shadow during sunset



4.2 Detail of the torch at night

player and in the direction of the main camera. This torch, especially at night, is very realistic and will also generate additional amazing shadows.

4.2 Texture

Every object in the scene has a texture. Before starting the game all the textures are loaded and then assigned to the object or part of the hierarchical model to which they belong. Each texture is given a ″fake″ three-dimensionality with the help of normal maps (The normal maps were generated using *this* tool). For very large objects such as floors, walls and parallelepipeds it was decided to use seamless textures so that the texture could be repeated according to the size of the object. In particular, how many times to repeat the textures of the parallelepipeds is decided in dynamic time according to their randomly chosen size.

4.3 Animations

All the animations we have realized are made using the Tween.js library. In particular, three are the animations present in the game:

- Player and enemies walk: consisting of the alternating rotation of the legs and the swinging of the left arm (the one not carrying the weapon) to give the appearance of a real walk.
- Boss walk: this has been the more complex animation we have made. It can be subdivided into two different behaviors, that of the front and hind legs, which make a wide angle, arriving almost parallel to the body, so as to carry most of the load of the movement; that of the central legs, which make a smaller angle, with the intention of supporting the movement of the other legs, with the feeling of giving greater stability to the movement.
- Boss death: a simple animation intended to give the impression that the spider, repeatedly hit by the player's bullets, no longer has the strength to stand and then collapses to the ground dying. The animation consists of extending all legs almost parallel to the ground and simultaneously lowering the y-axis position of the spider's body. After a short time lying in this position, it disappears below the ground.

4.4 Audio

For the game audio we used and edited some free short records of sounds for several types of actions and events.

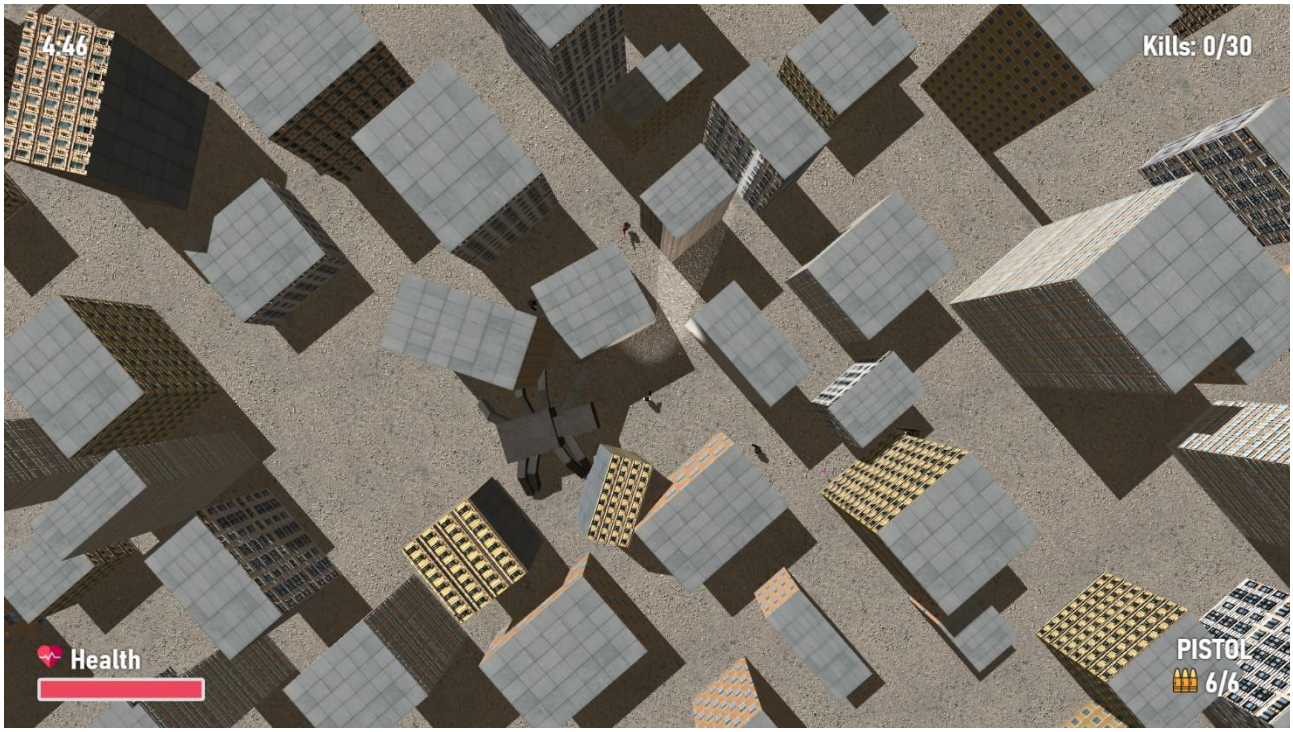
In particular, there is an audio track associated to each of the following events.

- Gunshot (different for pistol, ak-47, sniper, RPG)
- Reloading (different for pistol, ak-47, sniper, RPG)
- Walk and run (the same audio track but speeded up)
- Jump
- Getting injured (15 different audio variants)
- Kill an enemy
- Kill the boss
- Injure the boss
- Trigger the boss (it starts a music to increase pathos)
- Boss attack

- Game victory
- Game defeat

4.5 Camera views

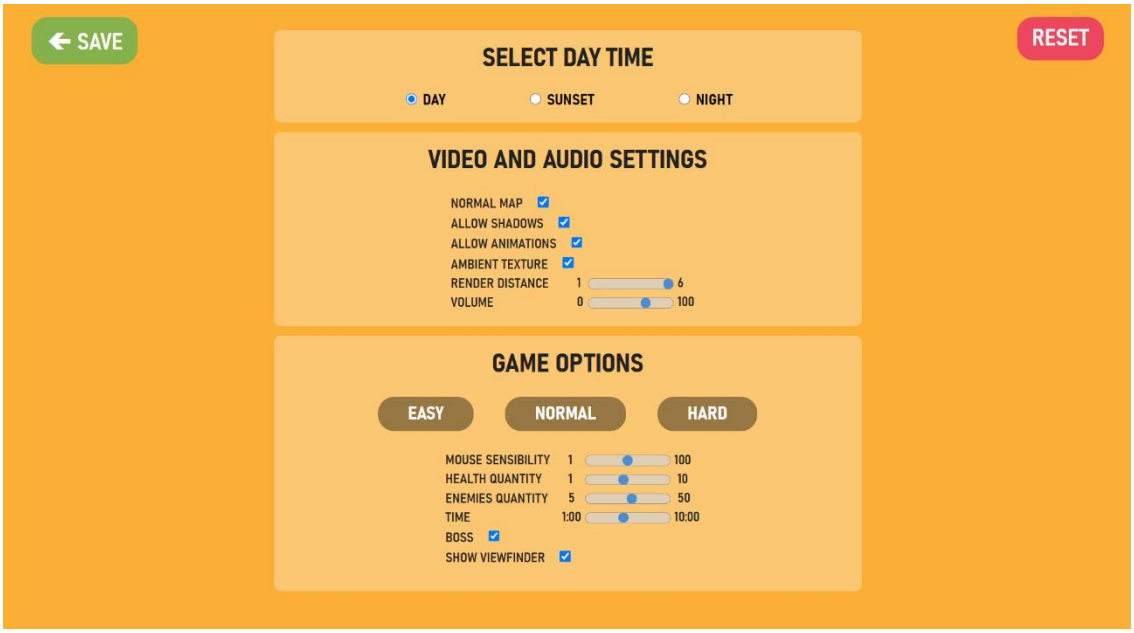
Tree different types of camera views can be used: first person, back and top view. In particular, the top view is a lot useful to find enemies behind buildings.



4.3 Top view camera

4.4 Back view camera

4.6 Game settings



In the first box it is possible to select the day time, this will change the skybox texture and also the lights and shadows appearance.

In video and audio settings it is possible to control some game feature aimed at improving the visual and auditory experience. Indeed, for everyone that does not have a dedicated GPU is recommended to disable some options. The settings you will find in this section are:

- Normal map: a checkbox allowing the user to activate/deactivate the presence of normal map in textures.
- Allow shadows: a checkbox allowing the user to activate/deactivate the presence of shadows.
- Allow animations: a checkbox allowing the user to activate/deactivate animations.
- Ambient texture: a checkbox allowing the user to activate/deactivate the presence of textures. In their absence all the objects (exception made for entities) will assume different random colors.
- Render distance: a slider consisting of six different levels, each one corresponds to an increment/decrease of 50 (meters) in the distance rendered. The minimum level allows the camera to render until a distance of 50 (meters) starting from the player position (for aesthetic reasons we preferred to not allow to decrease it more); the maximum level instead renders until a distance of 450 (meters), because it allows the user to see all the map angle to angle.
- Volume

In the game options section, it is possible to control some game aspects:

- It is possible to choose the difficulty of the game, this will set the game option to predefined values that however the user can change at will.
- Mouse sensibility
- Health quantity
- Enemies quantity
- Time, game duration

- Boss, a checkbox that allows the user to control the presence of the boss in the match (in easy modality it is disabled by default)
- Show viewfinder, a checkbox that controls the presence of the viewfinder (in hard modality it is disabled by default)

4.7 User interface

The UI is designed to be friendly and as simple as possible. To tell the user that something is clickable, buttons change appearance when the cursor is over them. Colours used in the whole game are more or less always the same. Main menu, HUD and game over page are responsive to different screen sizes. For the settings' page, we use the vertical scrolling bar.

During the game essential information are displayed:

- an animated health bar;
- ammo information;
- time remaining;
- enemies killed.