Final Project Interactive Graphics
"OPOLI's Drone Adventures"

Giacomo Colizzi Coin 1794538

Lorenzo Diaco 1796658

Matteo Germano 1807599

July 2021

Contents

1 Introduction

The game we have designed and implemented is a one player game. it is set in a futuristic city and the player controls directly a drone by moving it in the 3D environment. The drone can naturally move forward and backward, to the left and to the right and it can rise and descend. The goal consists in the collection of as many coins as possible before the depletion of the three lives of the drone. A life is lost when the drone hits a wall or when the fuel bar drop to the left red zone. During the game it is possible to collect also fuel tanks, to fill the fuel bar, stars, which make possible a greater collision on a wall or on the ground without the loss of one life and hearts, used in order to refill a life of the drone.

2 Libraries used

For a good realization of the project we have used two important libraries, three.js and tween.js. Three.js is 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. It allows the creation of graphical processing unit (GPU)-accelerated 3D animations using the JavaScript language as part of a website without relying on proprietary browser plugins. Tween.js is a JavaScript tweening engine for easy animations, incorporating optimised Robert Penner's equations. It supports tweening of both numeric object properties and CSS style properties. The API is simple but very powerful, making it easy to create complex tweens by chaining commands.

We have used ammo.js as phisical engine, to make more realistic and accurate some behaviors in the game. Ammo.js is a direct port of the Bullet physics engine to JavaScript, using Emscripten. In particular we handle collisions with this powerful library. It is very important for the correct implementation of the project because, if the drone collides with something, it must lose a life. Moreover also the collection of coins, hearts and fuel tanks could be seen as a collision but it must be handled in another way (the drone have not to lose a life in this case).

3 User Interface

As we can see in figure 1, the game starts with this interface. The player can choose the color of the drone and the color of the propellers between 5 different colors. The game has three levels of difficulty. The change of difficulty modify the speed of depletion of the fuel bar. The player can also choose the time of the day between day, afternoon and night and the drone type between the two visible in Figure 1. The change of day time makes possible to use three different lights in the game scene. The models of the two drones are analyzed in a successive section of the report.

The start button lets the game begin.

The game interface is very simple. In the upper right some notifies appear to inform the player about relevant events in the game. To handle these simple messages we have used the library Noty, a Dependency-free notification library. In the lower right part we set the fuel indicator while in the lower left part of the screen there are all the other relevant buttons and informations. between these ones we can see the current Frames per second, the time of play in seconds, the current number of lives of the drone (counted with hearts), the current level of difficulty of the game and last but not least the number of coins collected by the drone. Under these information we have a button for the various sounds of the game, an indicator for the current weather and a button that shows

the commands of the drone. In figure 5 we have a preview of this page. We decide to use not too many buttons to control the drone, bur as we can see later in the report, the drone behavior is very realistic. Near commands there is also the respective rotations of the drone (pitch for forward and backward, roll for the clockwise rotations and yaw for the left and right translations).

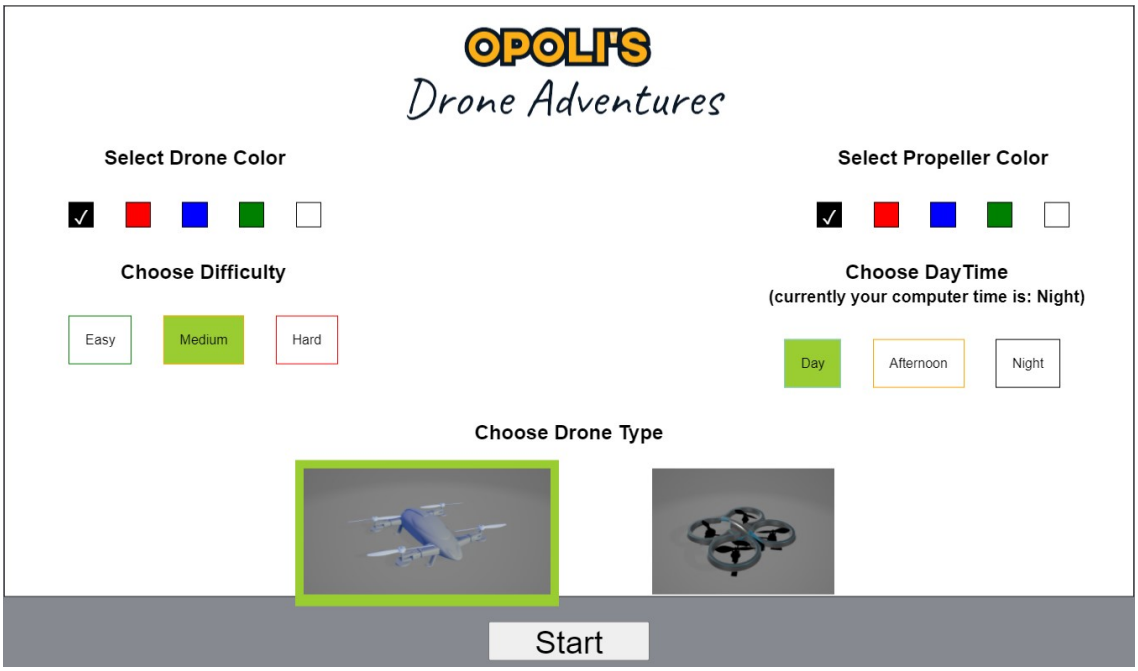

Figure 1: Starting game interface

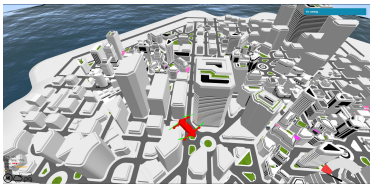

Figure 2: Day

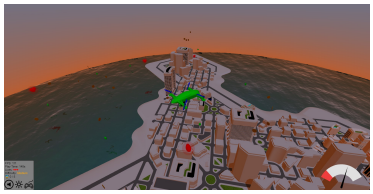

Figure 3: Afternoon

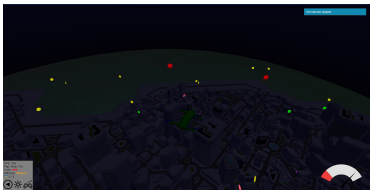

Figure 4: Night

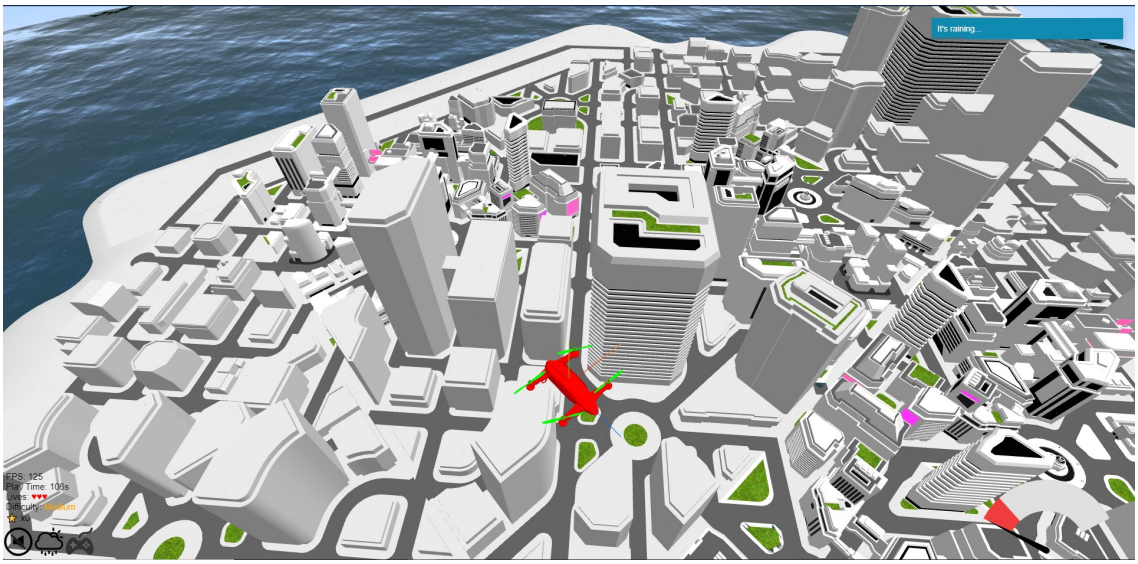

Figure 5: The flight of the drone and the HUD

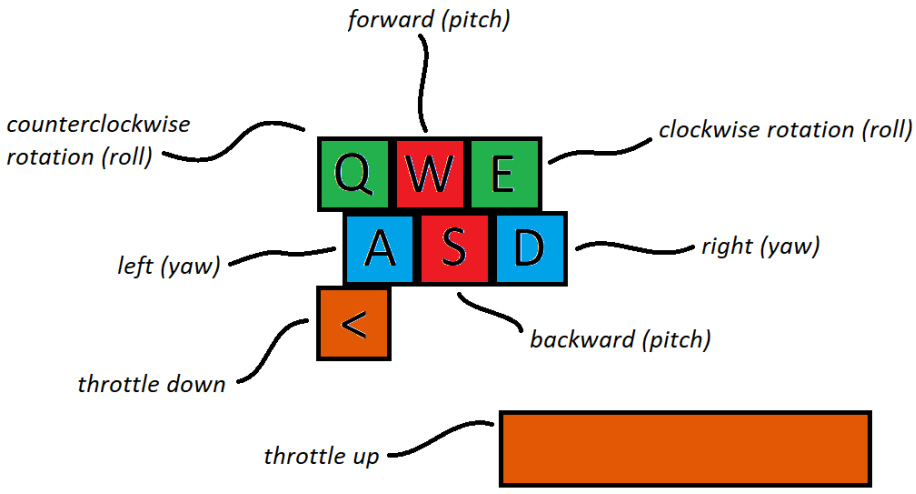

Figure 6: Controls of the movement of the drone

4 City

This futuristic city is the main environment of the game and the drone is free to move around the map. Sometimes and totally random rains can start to fall down. Around the city we have created the sea using textures, to have a more realistic ambient. Textures for the grass are been used for some places into the city. The bumpmap textures are fully visible in Figure 10.

4.1 Model

The model of the city is taken from Sketchfab and it is composed by a total of 284.5k triangles and 182.3k vertices. We have modified the original model to fill some holes present in the city model.

5 Drones

We have used two different models for two different drones in the game. Both drones are hierachical models. they are very complex and composed by a lot of objects. In our case we wanted to access the four propellers of the drone to animate them. These accelerate and decelerate when the drone moves up or down. Tween.js was very useful for the most realistic implementation of the movement. thanks to this library the animation are very fluid and realistic.

5.1 Models

The model of the first drone is taken from Turbosquid and it is composed by a total of 4148 polygons and 4558 vertices. The model of the second drone is taken from Sketchfab and it is composed by 27.6k triangles and 14.3k vertices.

6 Sounds and effects

If the player clicks the key U sounds start in the game. Sounds include music in background, propeller's drone movement sound, death and win sound, consumables collecting sound, rain sound and finally sound of hit again a wall or the floor without the loss of one life (soft collision).

7 Gameplay

After the selection of the color of the body of the drone and of the color of propellers, the difficulty level and the click of the button start, the game begins. The drone is on the floor at the very beginning. it has to ascend in order to start collecting some coins. If the player see that the drone is going up with too much speed, the button "<" can be pressed to slow the movement of the propellers in order to start a descent. Opposite commands (such as forward and backward, left and right) are mutually exclusive and clicking them together doesn't change the direction and the rotation of the drone while clicking together non mutally exclusive commands combine different kinds of movements and rotations. The player can see many kind of objects through the map. Coins must be collected to win the game, in particular 3 coins for the easy mode, 5 coins for the medium mode and 10 coins for the hard mode. We have modified the rotation axis of the coin to reach the current result visible in the game. Fuel tanks must be collected in order to keep the fuel

bar filled in order to keep flying. If the bar is totally empty the drone will fall down and it looses a life. Moreover a life is lost also when a collision with a wall happens. The last kind of objects in the game are hearts. They can be collect to restore one life. Every collectable item falls down from the sky and they can stop themselves at random altitudes in a random point in the plain parallel to the floor of the map. If it rains or it is night a spotlight appears lightning the objects in front of the drone. It is realized by using tween.js to reproduce a more realistic behavior. Figure 11 shows the spotlight.

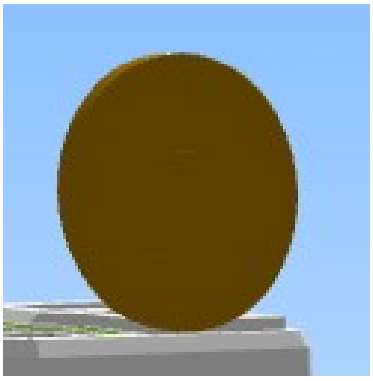

Figure 7: A coin

Figure 8: An heart

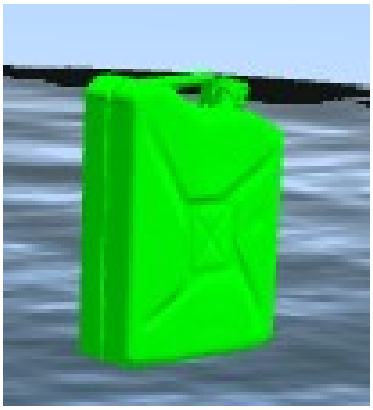

Figure 9: A fuel tank

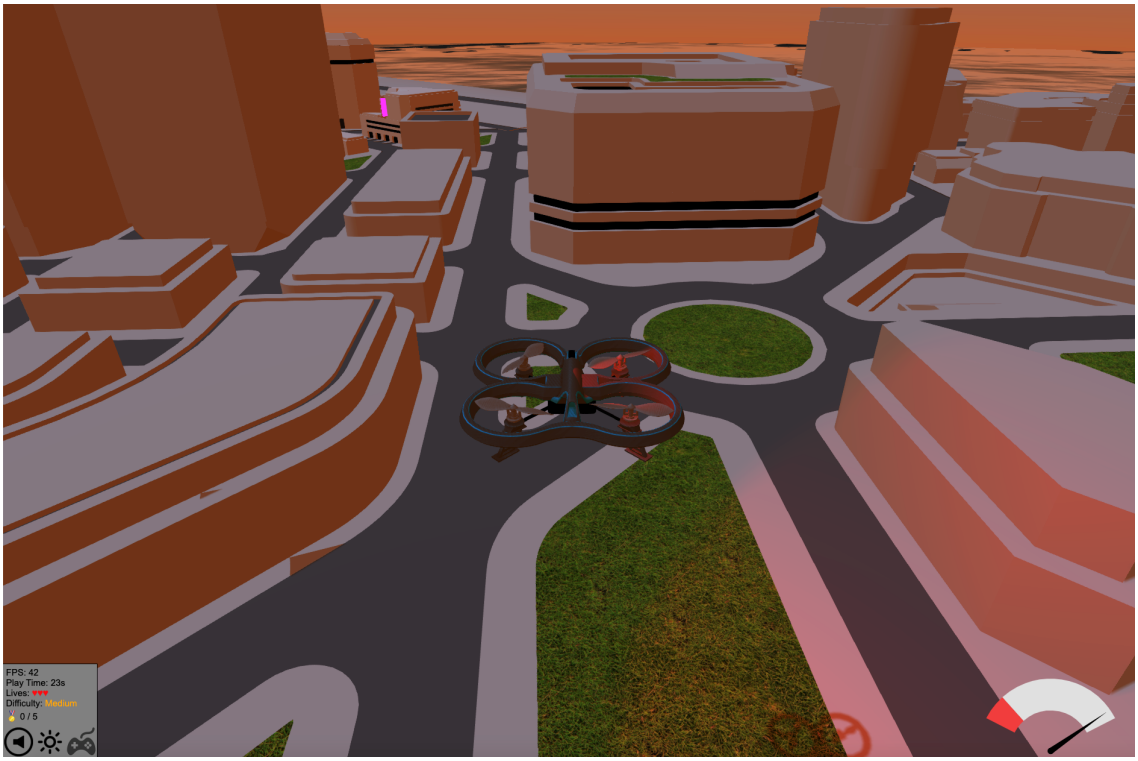

Figure 10: Grass bumpmap textures

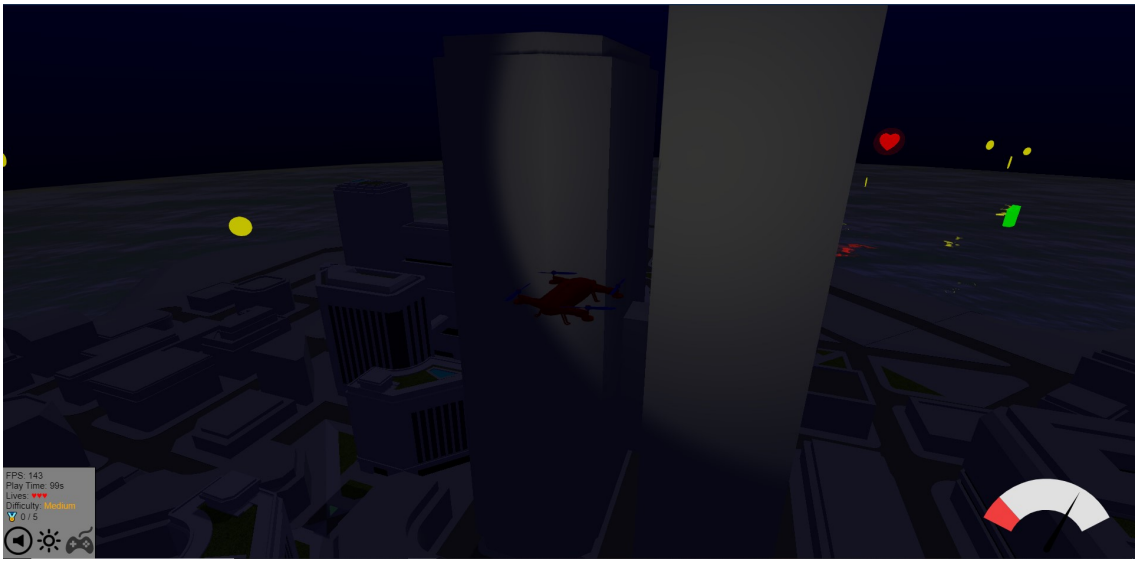

Figure 11: Spotlight of the drone

8 Conclusions

We have tried since the very beginning of the project to realize the most accurate animation for the drone. For this reason it could be not so easy the perfect control of the drone. But with some practice it can be perfectly governed and controlled.

9 Credits

Complete list of models used

- Drone 1: https://www.turbosquid.com/3d-models/qimmiq-transformer-drone-hybrid-c4d-free/1128800
- Drone 2: https://sketchfab.com/3d-models/rc-quadcopter-e03710ac63a1437ab51723b5c04ad50b
- City: https://sketchfab.com/3d-models/city-grid-block-3488e40ceca846bb9023f894a749c398
- Coin: https://sketchfab.com/3d-models/bitcoin-ff17dab54b044d789d05ae7c8dfb9808
- Heart: https://sketchfab.com/3d-models/heart-in-love-f39ce19b92e246268f4c501b72ea7d0e
- Fuel tank: https://sketchfab.com/3d-models/fuel-tank-midpoly-painted-c82eec8c84c04f10a707d2b3f7e0b727