The Working Dog
Final project report

Gabriele Iannola 1849732

Contents

1 Intro **1**

2 The game **1**
2.1 Lore 1
2.2 User interface 2
2.3 Controls and keybinds 4

3 Implementing the game **4**
3.1 The coding enviroment 4
3.2 The dog model 5
3.3 The city and other models 6
3.4 Camera, lights and post-processing 7
3.5 User interaction 7

4 Conclusive thoughts and further improvements **8**

5 Credits **8**

1 Intro

The Working Dog is a project developed as final project for the Interactive Graphics course. In this report, I am going to illustrate the game and discuss about some of its aspects, such as **interactive controls**, **models** and the **game environment**.

2 The game

2.1 Lore

With the discovery of Mars, a team of scientists is conducting a series of tests to develop a new parcel delivery system suitable for the dangerous Martian atmosphere. To do this, they have built R8-DG, a robot dog capable of efficiently

transporting parcels, which will have to pass a test in a purpose-built test environment. The goal is to **lead R8T-DG to the end of the experiment**, delivering the **packages** before his **battery** runs out.

2.2 User interface

When opening the game, you can choose between **three levels of test difficulty**, changing the number of packages to be delivered and the battery life of R8T-DG. The difficulty is set to EASY by default. In the initial interface, you can also see the **START button** and a number of **graphical controls** on the right-hand side, which I will discuss later.

At the start of the level, we can immediately notice two counters at the bottom of the interface: on the left, we have the **ENERGY LEVE**L, which will steadily decrease during the game, while on the right we have the number of **packages currently delivered**. If the energy level reaches zero, the game will end in defeat. The **arrow** above the dog will help you reach your destination.

2.3 Controls and keybinds

3 Implementing the game

3.1 The coding enviroment

For this project, I mainly used **Three.js**, a JavaScript library facilitating the use of WebGL coding, to which I combined **dat.GUI**, a simple library for implementing interactive controls and **Tween.js**, a library used to improve animation quality. I have also used **HTML and CSS to create the game GUI** and the various indicators.

3.2 The dog model

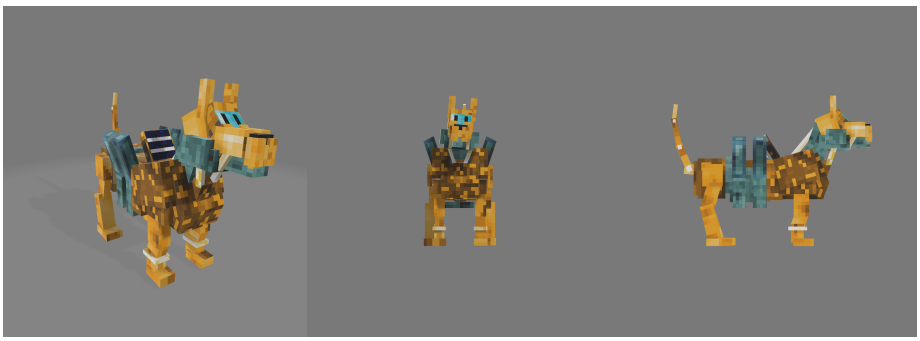


Figure 1: R8T-DG model from different view points

One of the requirements of the project was to include a **hierarchical model** that would allow complex animations to be created (using JavaScript). The dog model was imported from **Sketchfab** (see credits at the bottom of the document) in **GLTF format**, and was subsequently modified using Blender.

The structure of the nodes together with their rotation axes was modified in order to redefine the required hierarchical structure, and the basic animation was removed. The changes concern in particular the dog's legs and part of the head, which were necessary to define the new animations.

Using Tween.js, I have created two animations for the dog: a **running animation** and an **idle animation**, used respectively when the dog is or is not in motion.

A texture file is also associated with the model, which I had not changed. Nevertheless, I made changes to the model material, converting it to a **Mesh-Phong Material**, in order to simulate the metallic appearance of the robot's body.

3.3 The city and other models

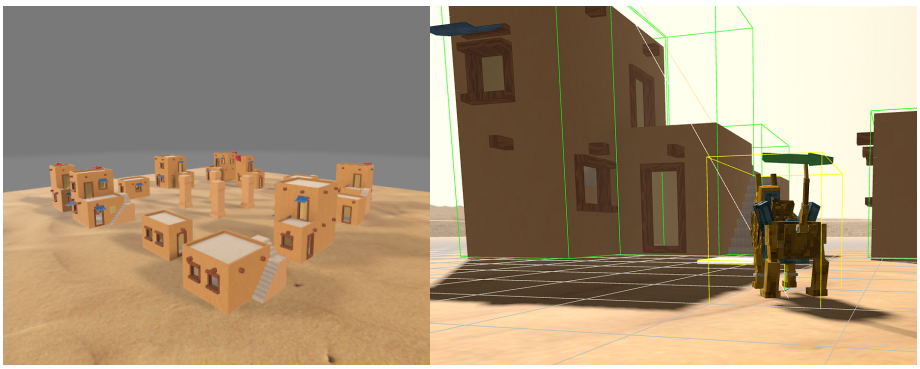
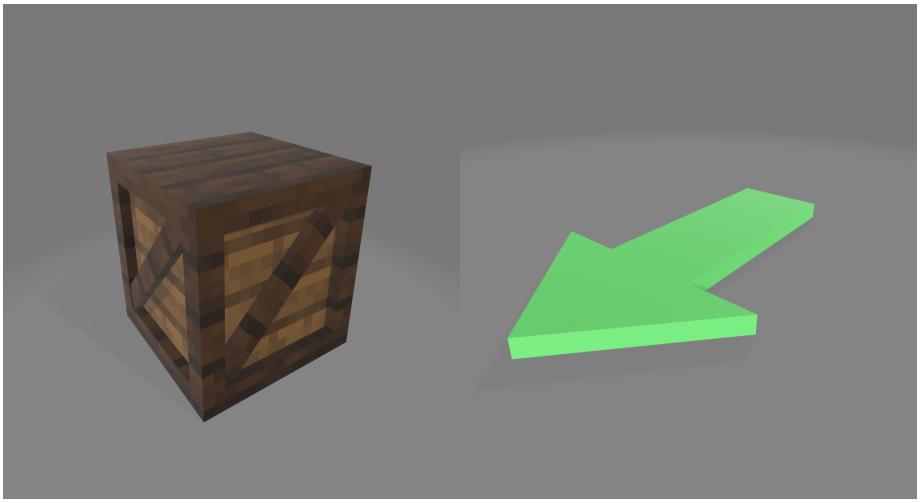


Figure 2: On the left, a view of the city. On the right, the hitbox system

As with the other models, the game environment consists of a **GLTF file imported from Sketchfab**. The city is built using one node for each of the **twelve houses**, each rendered as a **single mesh**, along with an additional node for the **terrain**. Although the city model did not undergo any particular changes, the use of single meshes for the houses led to some **problems with the collision system** between the dog and the buildings. As **no physics engine was used**, collisions were coded using Three's **Bounding Boxes**, which meant that more complex hitbox had to be created. The city model also includes **three textures**, two for the colours (of the sand and houses) and one for the buildings normals.

Two models (also imported) were used for the **box** and the **arrow**. No notable changes were made.



3.4 Camera, lights and post-processing

For the project, I have decided to implement a **third-person user interface**, which is quite traditional in several games. To do this, the **camera was locked** onto the dog's reference system, keeping the character always in the centre of the scene, but leaving the freedom to rotate the camera around the dog.

Regarding the lighting system, the scene comprises an **Ambient Light**, which is a dim light used mainly for shading, and a **Directional Light**, which in addition to being the main source of light is also responsible for the **shadows generated by the dog and the buildings**. In THREE it is indeed possible, starting from certain types of light sources, to have almost **complete control** over shadows by specifying which meshes cast and/or receive shadows. Furthermore, by exploiting a certain type of camera, which in THREE is called the **Directional Light Shadow Camera**, it has been chosen to render shadows only when the dog is close to the buildings that generate them, in order to lighten the data load.

Finally yet importantly, still regarding the post-processing part, I added an **exponential fog** which, unlike the linear fog, provides a more realistic effect.

3.5 User interaction

After introducing models, animations and lighting, the last requirement of the project was to provide **controls with which the user could interact** with the game. We have already mentioned the game controls and the possibility of choosing the difficulty, but I would now like to elaborate on the **side control interface**, implemented with 'dat.GUI'. From this interface, it is possible to control:

- **Ambient and Directional light**: you can control its intensity, colour and, for the latter only, the position of the source

- **Shadows**: it is possible to deactivate them and modify certain controls of the Light Shadow Camera (near, far and bias)
- **Audio**: you can control the audio volume of the effects and music
- **Other**: you can control fog density and disable light reflections on the dog model

4 Conclusive thoughts and further improvements

Working on this project was an opportunity for me to not only gain a better understanding of 3D graphics tools such as Three.js, but also the chance to see a video game idea come to life.

Below are some possible improvements (outside of the given requests):

- **Original assets**: replace imported models and audio with customised assets
- **Graphic quality**: improve lighting and post-processing as far as possible
- **Gameplay quality**: add challenges to the game, to make the gameplay more appealing to the user

In spite of everything, I believe that the project met the minimum requirements as best as possible.

5 Credits

- "The Last Dog" (https://skfb.ly/6Uwoy) by Thelta is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).
- "low poly desert village" (https://skfb.ly/6ZIAx) by cubemelongames is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).
- "Wooden Crate" (https://skfb.ly/6YTsu) by FishyBusiness is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).
- "Arrow V1" (https://skfb.ly/6X7qU) by mobileEY is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).
- Skybox: Perseverance Mars Panorama by NASA/JPL-Caltech/ASU/MSSS (https://mars.nasa.gov/resources/25640)
- Battery icons: https://pixabay.com/vectors/batteries-landing-icons-set-1379208/
- Keypad icons created by Freepik - Flaticon
- Crate icons created by lapiyee – Flaticon

- OST: "Sands of Mystery" Trey VanZandt (YouTube.com/c/treyvanzandt)
  Licensed under Creative Commons: By Attribution 3.0 License
- https://freesound.org/s/518850/
- https://freesound.org/s/561495/
- https://freesound.org/s/531512/
- https://freesound.org/s/523422/
- https://freesound.org/s/487436/
- https://freesound.org/s/533925/
- https://freesound.org/s/459449/