

Final Project
CROSS THE ROAD
Interactive Graphics 2021/2022

Nunziato Crisafulli 1696498

September 25, 2022

Contents

1	Scope	2
2	Environment	2
2.1	Other Libraries	2
3	Complex Hierarchical Models	2
3.1	Car	2
3.2	Truck and Tractor	2
3.3	Vehicles Texture	2
4	Other Models	2
4.1	startLawn and finishLawn	2
4.2	Street	2
4.3	Bush	3
4.4	Tree	3
4.5	StreetLamp	3
4.6	Rabbit	3
5	Lights	4
6	Animations	4
6.1	Vehicles	4
6.2	Rabbit	4
6.2.1	Movement Animation	4
6.2.2	Crash Animation	4
7	Alert	4
8	Commands	5

1 Scope

The scope of this project is to reproduce in a custom way the famous game **Crossy Road**. In this project there is one playable character (a rabbit) that needs to jump his way through the traffic to the other side of the street.

2 Environment

To develop this project I used the javascript library **three.js**, in particular it was used to create the ambient and the 3d objects involved. To animate these objects I used a tweening engine called **tween.js**.

2.1 Other Libraries

Another library used is **GLTFLoader**. It was used to load a well formed 3d model of a rabbit. More details later.

3 Complex Hierarchical Models

3.1 Car

Hierarchical model of the car is made up of **centralBlock**, **frontBlock**, **BackBlock**. The root node is **centralBlock**, the other two are both children of the root node. **frontBlock** has 4 children: 2 **frontWheels** and 2 **frontLights**. In the same way **backBlock** has 4 children: 2 **backWheels** and 2 **tailLights**. Blocks and frontLights/backLights are a **THREE.BoxBufferGeometry**, wheels are a **THREE.CircleGeometry**

3.2 Truck and Tractor

Hierarchical models of truck and tractor are made up of **frontBlock** and **BackBlock**. The root node is **frontBlock**, **BackBlock** is a child of the root node. **frontBlock** has 4 children: 2 **frontWheels** and 2 **frontLights**. In the same way **backBlock** has 4 children: 2 **backWheels** and 2 **tailLights**. Blocks and frontLights/backLights are a **THREE.BoxBufferGeometry**, wheels are a **THREE.CircleGeometry**

3.3 Vehicles Texture

Car, Truck and Tractor have 3 types of texture: **color**, **normal** and **specular** and then a color is applied. The color is chosen randomly in order to have always different vehicles. The choice of three textures is due to have a more realistic effects of the lights. More details later.

4 Other Models

4.1 startLawn and finishLawn

These two are a simple **THREE.BoxBufferGeometry** with a color texture applied. One is used as start position, the other as finish line to win the game.

4.2 Street

a simple **THREE.BoxBufferGeometry** with a color texture applied to make it look like a Road.

4.3 Bush

a simple **THREE.DodecahedronBufferGeometry** with a color texture applied to make it look like a real bush.

4.4 Tree

It is made of two nodes. a root **Trunk**, **THREE.BoxBufferGeometry**, node with a color texture applied, a child **Bush** node to make it look like a head of a tree.

4.5 StreetLamp

It is made of two nodes. a root **lampTrunk**, **THREE.BoxBufferGeometry**, node with a color texture applied, a child node, another **THREE.BoxBufferGeometry**, that represents the head of the street lamp where the light is located.

4.6 Rabbit

The rabbit used as character is a very complex model. It was downloaded from <https://sketchfab.com> and imported using **GLTFLoader**. Once imported it is used as any other 3d object in the scene.

5 Lights

Lights have two settings. The game start with a normal ambient light **THREE.AmbientLight(0xfffff, 1.2)**. The user can switch to a night setting clicking on **LIGHT SWITCH** button. This setting has 2 **THREE.SpotLight(0xfffff, 1)**, one for each **streetLamp**. Along with these two lights there are 4 spotlights for each vehicle, 2 white frontLights and 2 red tailLights with less intensity. The user can switch between the two at runtime without restarting the game. Spotlights usually point at (0, 0, 0). In the case of vehicles spotlights point at an invisible target positioned far away from the scene, one far left, one far right in order to have the frontlights always pointing at the direction of travel of the vehicle, instead the taillights always point in the opposite direction. The use of normal and specular textures applied using **THREE.MeshPhongMaterial** make the effect of the light on the passing vehicles more realistic.

6 Animations

As I said before to perform animation **tweenjs** was used.

6.1 Vehicles

Vehicles simply moves along the X axis from left to right or viceversa. So for them one simple tween is used.

6.2 Rabbit

6.2.1 Movement Animation

Rabbit can jump in three direction: **forward, left and right**. There is no coming back. All three animation are a combination of several tweens:

1. First the rabbit will rotate in the direction of the jump with one tween;
2. Then front legs and backs legs will slightly rotate clockwise using one tween for each leg, so four tweens
3. Then the rabbit will make in parallel a movement along x (if it jumps left or right) or along y axis (if it jumps forward) and a movement along z axis with one tween (to go up in the air).
4. Subsequent tweens are used to make the rabbit go down while restoring the position of the legs.

(1) is performed alone, (2) and (3) are performed in parallel, also subsequent tweens are executed in parallel. To make the jump look more realistic I used an easing function on (3) animation using **TWEEN.Easing.Exponential.Out**. On the inverse tween I used instead **TWEEN.Easing.Exponential.In** to make it feel like there is the presence of gravity.

6.2.2 Crash Animation

If the rabbit will collide with a vehicle (the space they occupy is overlapping) it will jump very high while spreading the legs to reproduce the consequences of an impact.

7 Alert

You win if the rabbit reach the other side of the street and you get **"WIN!" alert** on the browser, otherwise if there is a crash you get **"CRASH!" alert**. In both cases the game will restart, the rabbit comes back in starting position and new vehicles will spawn.

8 Commands

- **[W]** Jump forward
- **[D]** Jump right
- **[S]** Jump left