Final Project - Interactive Graphics
Kick Off

Domenico Tersigni 1817502
Alessia Orchi 1792993

17 July 2022

1 Libraries used

We have designed a 3D soccer game and we used two main libraries:

- **Three.js**: it is a 3D library that tries to make easy to get 3D content on a webpage. Three.js is often confused with WebGL since more often than not, but not always, three.js uses WebGL to draw 3D. WebGL is a very low-level system that only draws points, lines, and triangles. To do anything useful with WebGL generally requires quite a bit of code and that is where three.js comes in. It handles stuff like scenes, lights, shadows, materials, textures, 3d math, all things that you'd have to write yourself if you were to use WebGL directly.
- **Tween.js**: it is a simple tweening library for use in JavaScript. The API is simple but very powerful, making it easy to create complex tweens by chaining commands.

2 Game description

The game take place in an outside environment where are represented a soccer field, many trees and a road. User plays against computer, he/she controls a robot that first has to complete an obstacle race, then when he/she reaches the penalty area, where there is a ball, can kick and the goalkeeper can make a save. The goal of the game is to reach an higher score than the goalkeeper, rules are simple:

- **Obstacles**: when the player hits an obstacle he lost one point.
- **Goal**: when the player score a goal he/she obtains one point and the goalkeeper obtains zero points.
- **Save**: when the goalkeeper make a save he/she obtains one point and the player obtains zero points.

3 UI interactions

Kick Off starts with this interface:

Figure 1: Interface

There are two models and each of them performs a different animation, the robot on the right side performs a Jumping Jack and the other is doing warm up created using Tween.js. In this page using a dropdown menu the user can:

- **Uniform color 1**: change color of player's uniform using the Uniform Player button.
- **Uniform color 1**: change color of keeper's uniform using the Uniform Keeper button.
- **Time of day**: change the time of the day between three possible times: day,sunset and night; according to this choice, lights change their color.
- **Mode** : change the difficulty between two levels: beginner and legend, if he/she selects beginner there is an hemisphere light and all the obstacles are clearly visible. if he/she selects legend mode there is no hemisphere light, but there are two spotlighs, one of these is set on the goalkeeper and the other follows the player, (in order to implement this we use target property of **THREE.Spotlight**). In the legend mode also appears the ceiling lamp and even it follows the player but the obstacles aren't clearly visible.

In the bottom of the page there is the Kick Off button that starts the game and the rules of the game. To pass the modality selected in this page to the game, we exploited the sessionStorage, in which we write and retrieve the attribute:

- Mode
- Color of the keeper uniform
- Color of the player uniform
- Time of the Day

4 Models

In this section are described the model used to create the game.

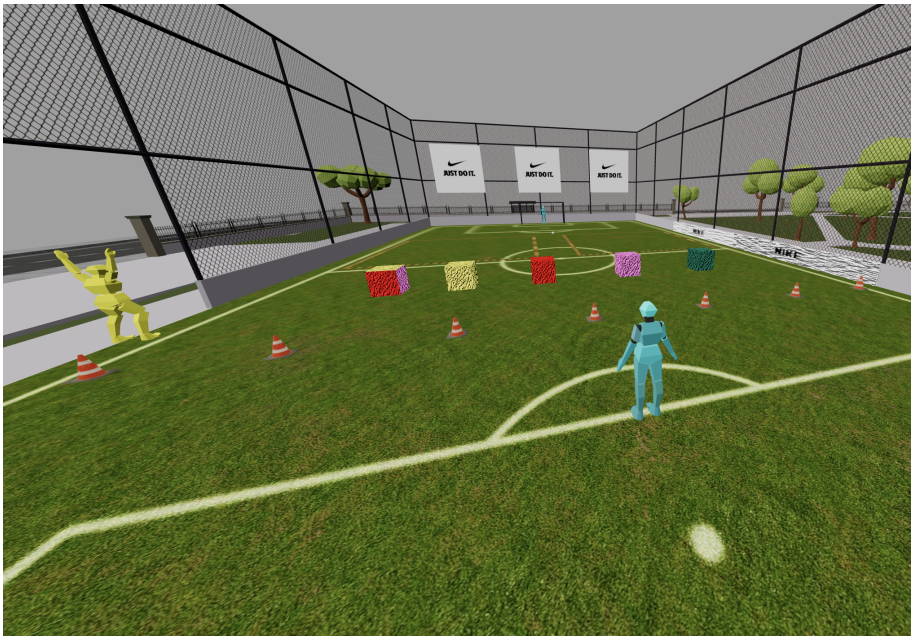

Figure 2: Models

4.1 Enviroment

The Enviroment, which is composed by the field, many trees and a road, is a 3D model that we have imported from Sketchfab, it's seem very complex but it has only 14.8k vertices. [7] We have modified the original model by adding some objects inside the soccer field, in particular:

- **Billboards**: we have put three thin cubes on the fencing net and we have applied to them a **texture** that shows the famous slogan 'JUST DO IT'.
- **Sponsor 1**: on the right side of the field there are two sponsor stadium banners, they are two thin cubes and on one of them we have applied a **normal texture**.
- **Sponsor 2**: on the second sponsor we have applied a **bump texture**.

4.2 Robot

The player, the goalkeeper, the player who is doing warm up on the left side of the field and the two models who are in the menu page, are all the same 3D model that is taken from sketchfab, it's low-poly, in fact it has only 510 vertices.
[5] This model is a quite complex hierarchical model and we have implemented different animations based on this model in order to completely exploit his structure. To create smooth animation we extract the nodes of the hierarchical model from the gltf model and using tween we made the animation.

4.3 Ball

The ball is a simply 3D model taken from sketchfab, it has 482 vertices and on it there is a normal texture not implemented by us. [6]

4.4 Ceiling Lamp

The ceiling lamp is a simply 3D model taken from Sketchfab, it doesn't emit real light, but in order to give this effect we added a spotlight whose position is the same of the ceiling lamp.[3]

4.5 Car

It's a low-poly model of a Lamborghini Aventador taken from Sketchfab[2]. We have implemented a simple animation and through it the car moves on the road.

4.6 Obstacles

There are three different types of obstacles in the game, and for all of them we have done **collisions detection**.

4.6.1 Cones

First obstacles are cones and the player should go ahead in the field without hitting them, if he/she hits a cone he/she lost two points and the game restart. Cones are simply 3D models imported from Sketchfab.[1]

4.6.2 Cubes

After the cones, the player should go through some cubes, we have added five different cubes and on each of them we have applied a **bump texture** in order to give different aspect at these cubes according to the light changes. One of these cube has a **different texture on each face**. As before If the player hits a cube he/she lost two points and the game restart.

4.6.3 Caution

After the cubes, there are two areas of the field where the access is prohibited by two "not-cross-squares" that are two simply models imported from Sketchfab.[4]

5 Keyboard

We have added two event listeners, one for the event keyup and one for the event keydown and by using different flag we handle incorrect behavior. In particular the keys that we manage are:

- **A**: to run.
- **K**: to kick when we are in front of ball.
- **Up arrow**: forward direction.
- **Left arrow** : to turn left.
- **Right arrow**: to turn right.

6 Lights

Our scene contains different types of lights:

- **Hemisphere Light**: It's a light source positioned directly above the scene, with color fading from the sky color to the ground color. We use it in order to capture all the shades of the sunset and of the nights.
- **Fixed Spotlight**: When the mode is legend appears a spotlight on top of the goal-keeper, with correct position,angle and intensity.
- **Mobile Spotlight**: When the mode is legend appears also a spotlight on top of the player, and this light follows player movements, we correctly set position,angle and intensity.

In both spotlight we also set to true the **castShadow** property and the meshes in the scene have the property **receiveShadow** set to true in order to have shadows.

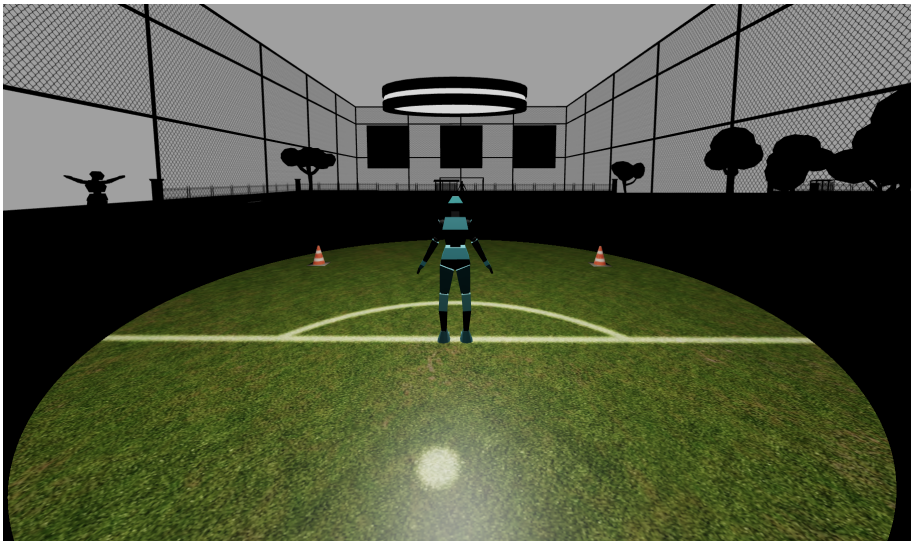

Figure 3: Legend Light

7 Texture

We use different types of textures:

- **Color texture**
- **Normal texture**: we have applied it on one of the two sponsor stadium banners. We drawn the cube with **THREE.BoxGeometry** and we used as material **THREE.meshPhongMaterial**, then we set **normalMap** and **normalScale** properties of this material.
- **Bump texture**: we have applied it in the other sponsor stadium banner, and on all cubes obstacles. We drawn these cubes with **THREE.BoxGeometry** and we used as material **THREE.meshPhongMaterial**, then we set **bumpMap** and **bumpScale** properties of this material.

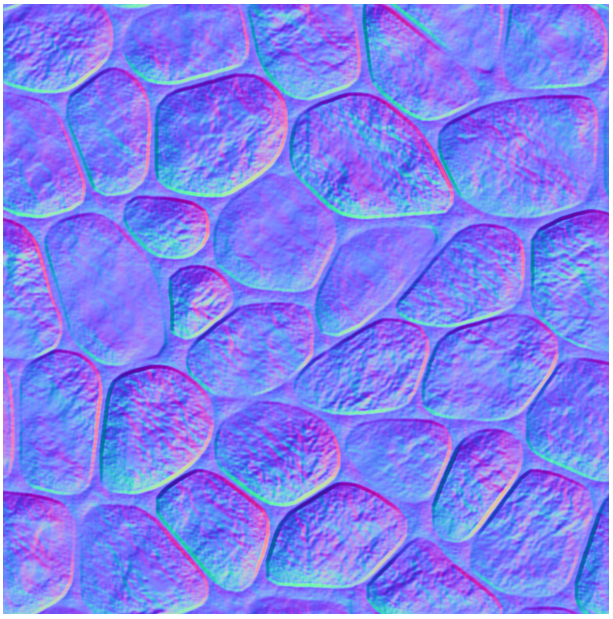

Figure 4: Normal Map

8 OrbitControls

We have used OrbitControls in order to move the camera using the mouse and to make the came follow the player movement, it's always possibile to look back and forward by dragging the mouse.

9 Animations

Animations are the core of this game and using Tween.js we have realized different quite complex animations:

- **Jumping Jack**
- **Warm up**
- **Run**: it's a complex animation that exploits the whole hierarchical structure of the robot.
- **Kick**: there are 4 different versions of this animation, he/she can shot top right or top left or bottom right or botton left;
- **Save**: there are 2 different versions of this animation that are related to the 4 possibile versions of the kick animation.

- **Car**: it's a simple animation performed by the car moving on the road.

We use a random function in order to pick one of the possible combinations of kick and save animations.

10 Gui

We used a simple GUI to show the actual scoring of the system:

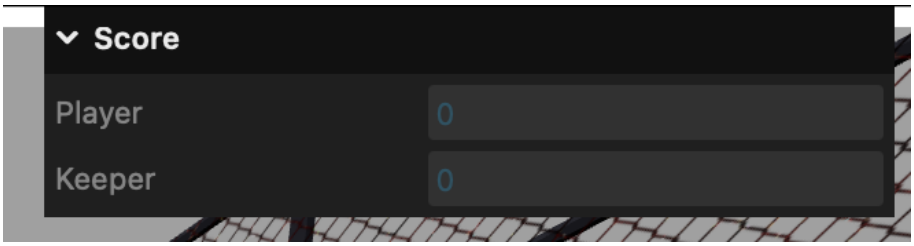

Figure 5: GUI

It retrieves and update the score exploiting the session storage and is updated of every time that:

- **T**he player scores a goal
- **T**he player hits an obstacle.
- **T**he keeper saves a goal

References

[1] https://sketchfab.com/3d-models/street-cone-c469cc01064742b4a730d51526458c78

[2] https://sketchfab.com/3d-models/lowpoly-lamborghini-aventador-fa870a26ccb8401cb5c9809349592eeb

[3] https://sketchfab.com/3d-models/ceiling-light-3e65cebe43b144ca802101d482b9c193

[4] https://sketchfab.com/3d-models/caution-band-empale-altoahite-3fea0193a8e24ea7a51a376e08879cc8

[5] https://sketchfab.com/3d-models/mia-rigged-33d906720b004702bccc664c26e9342f

[6] https://sketchfab.com/3d-models/football-72a3f17f6d0245ef9b047ff813c6f647

[7] https://sketchfab.com/3d-models/football-field-60331aee1f2e4c7fb545c464855bb378