# **Interactive Graphics**

# **Final Project**

Immagine che contiene disegnando, cibo

Descrizione generata automaticamente

2019/2020

Mauro Ficorella

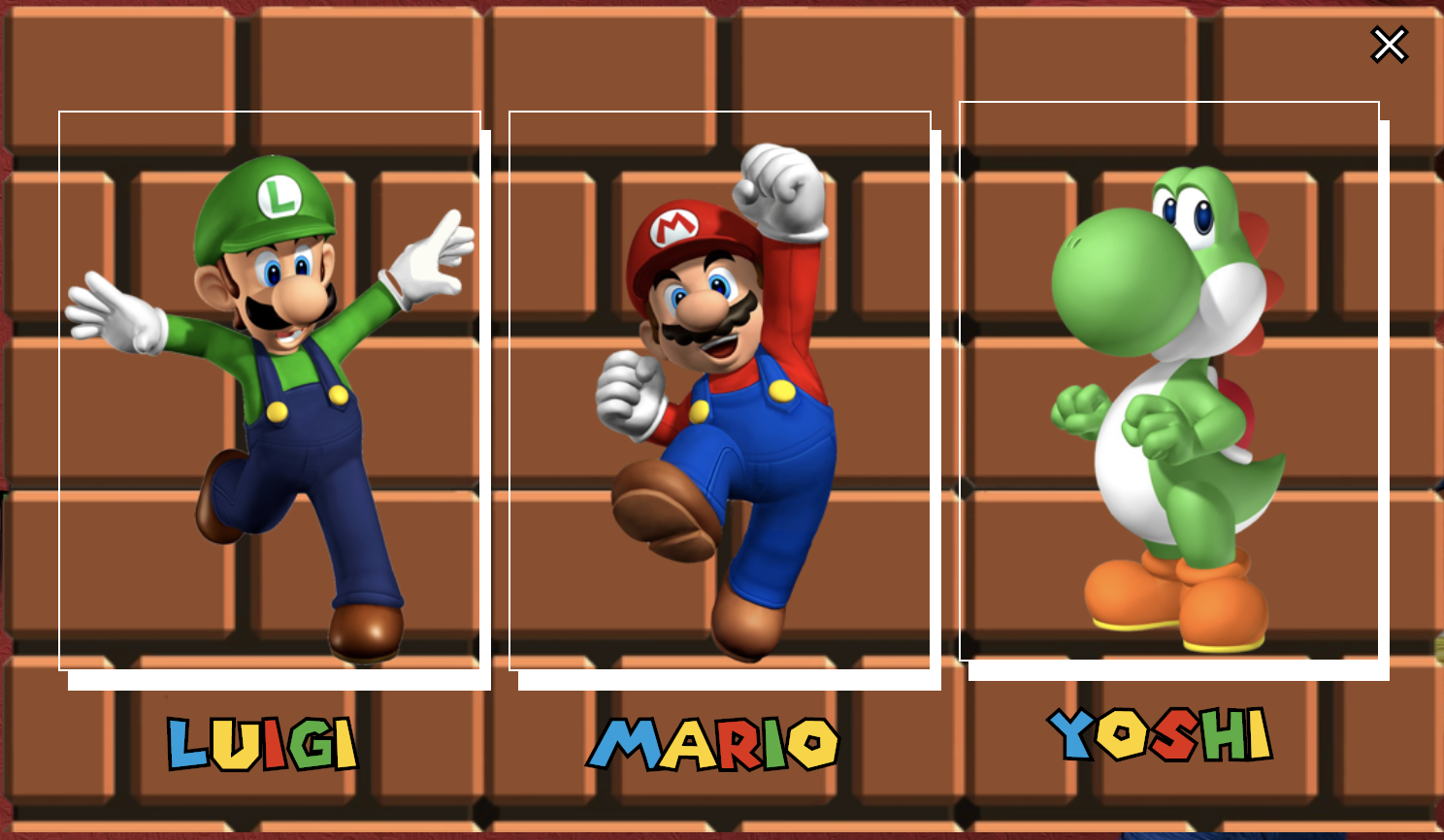
Valentina Sisti

Martina Turbessi

**Introduction**

We have chosen the famous Super Mario platform game as the main theme of the final project. Precisely, we recreated Super Mario Bros’ first level.

The main goal of this game is to finish the level collecting as many coins as possible without losing life against enemies (goombas).

At the beginning of the game, the player can choose between three characters: Mario, Luigi and Yoshi.

During the level, the player can interact with some rewards, like coins or power ups; more deeply, each question block contains those items and character can earn them by jumping against them (hitting with head) from below.

In the level are also shown enemies that are the well known goombas, taken from the original game: if the character collides with them, he loses life, and, if the character lose all his lives, the game is over; otherwise, if the character has earned more than one life, if he collides with goombas, he restart from the beginning of the level without losing coins. Another possibility is that the character collides with goombas jumping on them: in this case the goomba will die.

When the character reaches the end of the level, more precisely in front of castle’s door, the player wins the game and is shown a window containing how many coins he has collected during the level.

The project has been developed using Three.js as the main library, Tween.js for the animations and Physijs for collision detection.

**Scene**

**//METTERE FOTO BACKGROUND SENZA ORBIT CONTROLS**

The environment of the game includes a main platform on which the character walks, a background image that repeats itself during the whole level; we used a perspective camera placed in a way that shows character in a profile view; then, in order to follow character’s movements, we used the “lookAt” function inside the animate function to update its parameters at runtime with character’s position.

The entire scene is illuminated by an ambient light and a directional light that follows character’s movements. All these lights are implemented using Three.js library, more precisely in the following way:

const d = 100;

const color = 0xffffff;

const intensity = 1;

dirLight = new THREE.DirectionalLight(color, intensity, 100);

dirLight.position.set(0, 100, -620);

dirLight.castShadow = true;

dirLight.shadow.mapSize.width = 512;

dirLight.shadow.mapSize.height = 512;

dirLight.shadow.camera.near = 0.5;

dirLight.shadow.camera.far = 500;

dirLight.shadow.camera.fov = 50;

dirLight.shadow.bias = 0.0039;

dirLight.shadow.camera.left = -d;

dirLight.shadow.camera.right = d;

dirLight.shadow.camera.top = d;

dirLight.shadow.camera.bottom = -d;

scene.add(dirLight);

ambientLight = new THREE.AmbientLight(color, intensity);

scene.add(ambientLight);