

Final Project – Interactive Graphics

Stefano Servillo – 1845313

ROBOT ROAD

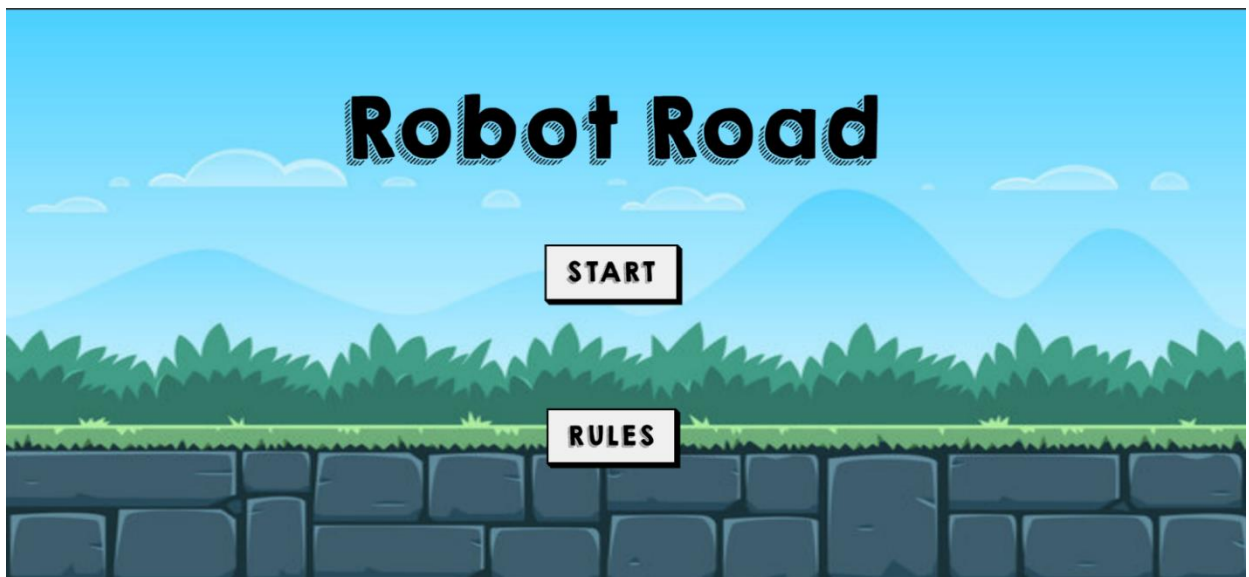


1. Introduction

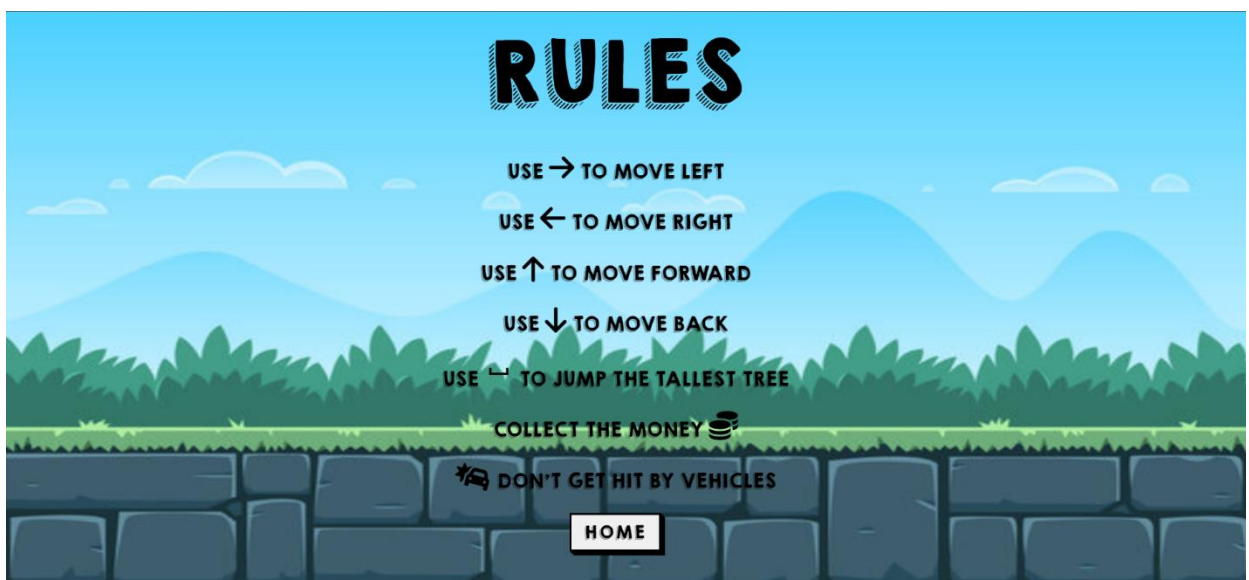
Robot Road is an infinite game in which the player can move the character in all directions along various roads. His goal will be to collect as many coins as possible without getting hit by any vehicle.

2. Robot Road

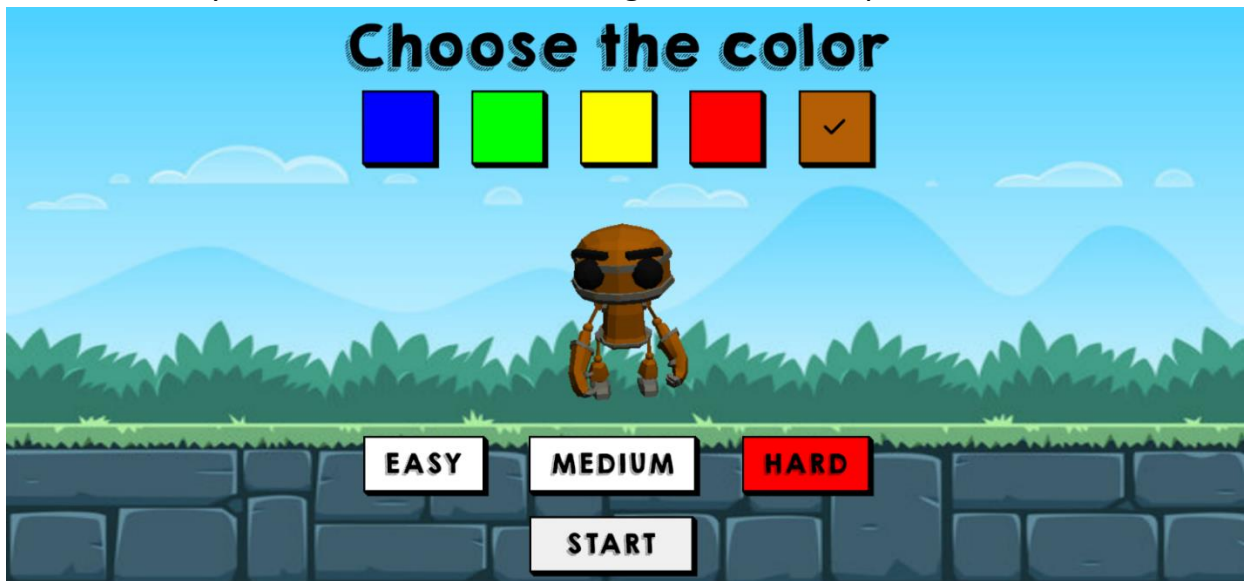
The start page allows the user to press two single buttons, one that will show him all the information before starting to play, and the other to continue in the beginning of the game.



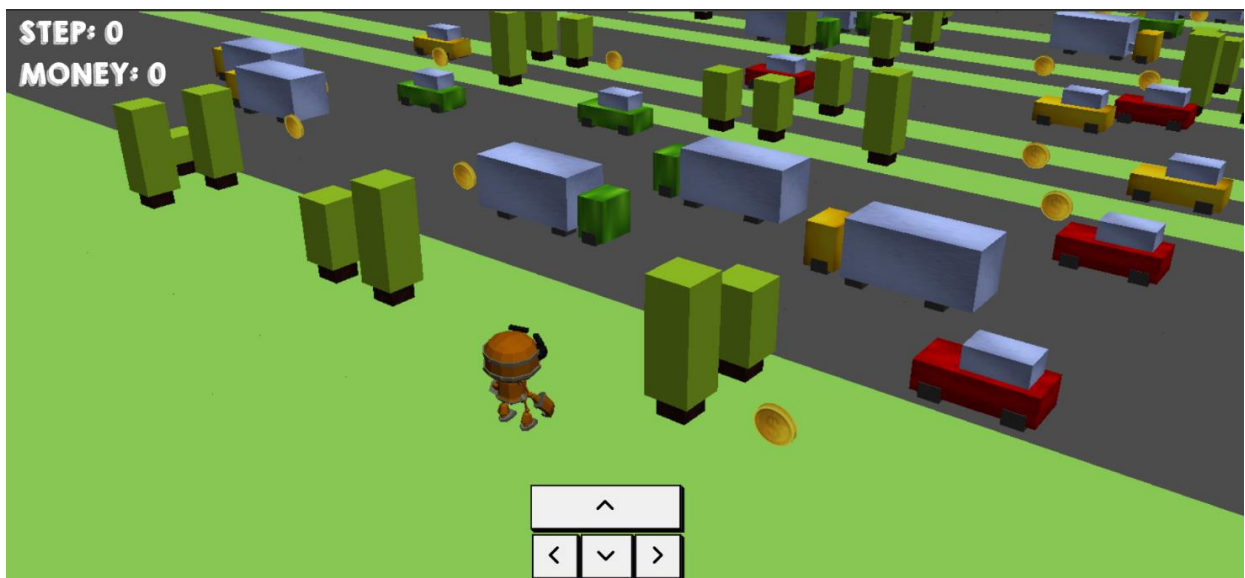
If the player presses on the **RULES** button he will see the commands, what not to do if you do not want to lose and what is the goal of the game.



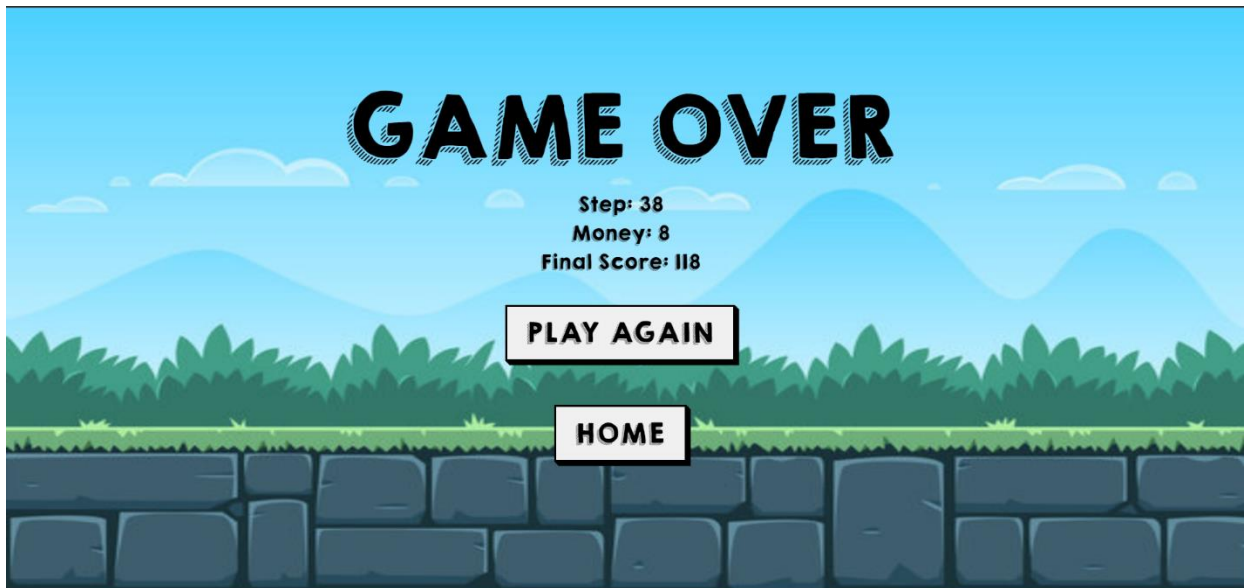
When the player has pressed **START**, an intermediate page will be shown in which he will be able to change the colour of the character and the difficulty of the game, a difficulty that will change the speed of vehicles.



Once the setting has been chosen, the player can start playing. The page will show the robot in its initial position and through the buttons on the keyboard, but also those on the screen, the user can move the character, who will perform different animations based on the movement chosen. In the game there are also coins to collect, disturbing elements such as trees and vehicles, elements to be avoided.



If the character is hit by a vehicle, the player will have lost and will be taken to the final screen where he can see his final score and decide whether to return to the home or replay with the same settings chosen at the beginning.



3. Code

The code is organized in three files:

- **main.js;**
- **main.css;**
- **Index.html**

In the main.js file there is the function Scene() that initialize all the parameters for the scene, like camera, lights, background and all variable necessary for the application. In this file are define vary classes to create objects.

- The Object class was used to make the various objects of the game:

```
class Object {  
  constructor() {}  
  
  > static Wheel() { ...  
  }  
  
  > static Car(flag) { ...  
  }  
  
  > static Tree() { ...  
  }  
  
  > static Money() { ...  
  }  
  
  > static Truck(flag) { ...  
  }  
  
  > static Road() { ...  
  }  
  
  > static Grass() { ...  
  }  
}
```

- The Lane class was used to make the several types of lanes of the floor:

```

class Lane {
  constructor(index) {
    const type = Math.floor(Math.random() * laneTypes.length);
    this.type = index <= 0 ? 'grass' : laneTypes[type];
    var position = 7-(index+8);
    switch (this.type) {
      case 'grass': { ...
    }
    case 'car': { ...
    }
    case 'truck': { ...
    }
    case 'forest': { ...
    }
  }
}

```

- The Animation class takes care of running the animation of the character:

```

class Animator {
  constructor() {}
  > static preprocessing(type, plus){ ...
  }
  > static shoulder(type,plus){ ...
  }
  > static jump(front, height,rotation){ ...
  }
  > static jumpSide(type){ ...
  }
  > static idle(){ ...
  }
  > static collect(){ ...
  }
  > static money(){ ...
  }
}

```

To realize the infinite game, when the character moves, also the camera moves. When the character moves forward, a new line is added at the end and at the same time the first line is deleted from the scene and stored in a variable to be able to re-insert it in case the character returns backwards. The animation of the objects is handled by the animation() function. The collision() function is responsible to check if the character is hit by some objects and do different things depending on the type of object.

4. Models

4.1 Robot

The model with the hierarchical structure is the main character, the robot, taken from THREE.JS, composed by many and many bones and skinned mesh.

4.2 Other elements

All other elements were made by me using the three.js functions.

4.2.1 Trees

The tree was divided into two parts, the crown, and the trunk. Both are parallelepipeds with different width and height. The crown is associated with a light green colour and its height can change randomly, while the trunk always remains the same and a texture has been applied.

```
const crown = new THREE.Mesh(  
  new THREE.BoxBufferGeometry(0.6,height,0.6),  
  new THREE.MeshPhongMaterial({  
    color: 0x7aa21d,  
    flatShading: true})  
);
```

```
const trunk = new THREE.Mesh(  
  new THREE.BoxBufferGeometry(0.4,0.6,0.4),  
  new THREE.MeshPhongMaterial({  
    color: 0x4d2926,  
    flatShading: true,  
    map: trunk_texture})  
);
```



Trunk texture

3.2.2 Vehicles

In the game we have two distinct types of vehicles: cars and trucks.

The cars are divided into two parts, the cabin, which includes the upper part and the main part, which is the actual machine. Both are parallelepipeds of varied sizes to which two different textures have been applied, that of the main part changes according to the colour of the car, which can be red, yellow, or green, while a silver texture has been associated to the cabin.

```
const main = new THREE.Mesh(  
  new THREE.BoxBufferGeometry(0.8,0.3,0.5),  
  new THREE.MeshPhongMaterial({color: 0xb4c6fc, flatShading:true, map: car_texture})  
);
```

```
const cabin = new THREE.Mesh(  
  new THREE.BoxBufferGeometry(0.5,0.7,0.7),  
  [  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } ),  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } ),  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } ),  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } ),  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } ),  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } )  
  ]  
);
```

The trucks, on the other hand, have been divided into three parts, we always have the cabin, then the cargo and finally a base to be able to connect the cabin with the cargo. All are always parallelepipeds of varied sizes to which textures have

been applied, For the cabin always the texture based on the colour, while for the other two a silver texture.

```
const base = new THREE.Mesh(  
  new THREE.BoxBufferGeometry(2.15,0.2,0.2),  
  new THREE.MeshPhongMaterial({color: 0xb4c6fc, flatShading: true, map: car_texture})  
);
```

```
const cabin = new THREE.Mesh(  
  new THREE.BoxBufferGeometry( 1.5, 0.4, 0.7 ),  
  [  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } ),  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } ),  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } ),  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } ),  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } ),  
    new THREE.MeshPhongMaterial( { color, flatShading: true, map: texture } )  
  ]  
);
```

```
const cargo = new THREE.Mesh(  
  new THREE.BoxBufferGeometry(2,0.9,0.9),  
  new THREE.MeshPhongMaterial({color: 0xb4c6fc, flatShading: true, map: car_texture})  
);
```

Both vehicles have wheels, made like cubes. The cars have 4, 2 rear and 2 front, while the trucks have 6, 2 front, 2 rear and 2 intermediate.

```
const wheel = new THREE.Mesh(  
  new THREE.BoxBufferGeometry(0.275,0.275,0.275),  
  new THREE.MeshPhongMaterial({ color: 0x333333, flatShading: true})  
);
```



Silver texture



green texture



red texture



yellow texture

3.2.3 Coins

The coins were made as cylinders with a small height to give the impression of the thickness of the coin. A texture was applied to apply the dollar symbol on them.

```
const money = new THREE.Mesh(  
  new THREE.CylinderGeometry( 0.25, 0.25, 0.08, 64 ),  
  new THREE.MeshBasicMaterial({  
    color: 0xbdb638,  
    map: money_texture  
  })  
);
```



Coin texture

3.2.4 Lines

Finally, the various lines are divided into two distinct types: roads and grass. Both are long and thin parallelepipeds of distinct colours, grey for the road and green for the grass.

```
const geometry = new THREE.BoxBufferGeometry(75,0.1,1);  
const material = new THREE.MeshPhongMaterial({color: 0x444444});  
const road = new THREE.Mesh(geometry, material);
```

```
const geometry = new THREE.BoxBufferGeometry(75,0.1,1);  
const material = new THREE.MeshPhongMaterial({color: 0x78b14b});  
const grass = new THREE.Mesh(geometry, material);
```

5. Animations

Tween.js was used to make the animations.

5.1 Robot

The character can perform six different animations depending on the chosen movement and one periodic.

MOVE FORWARD: The first animation allows the character to move forward, the movement that will be performed is a leap forward. The character will take the charge to perform the jump, a movement including bending of the knees and opening of the arms, will perform the jump and finally will land a square forward, simulating the landing always bending the knees and raising the arms.

MOVE LEFT AND RIGHT: To move left and right, the character will perform a simple rotation in the chosen direction. The charge and the stop of the jump is always present.

MOVE BACKWARD: For the backward movement, the character will perform a backward somersault.

JUMP THE TALLEST TREE: There is a further possible movement, this can be done when the character meets the tree with the highest crown, allowing the character to jump it, jumping three lines forward. Everything will be possible only if there is no tree at the destination point. When possible, the button to use will be enabled.

PERIODIC: An animation has also been inserted every 10 seconds in which the robot head will be rotated.

COLLECTION: Finally, if the character collects a coin, a pirouette will be performed on the spot.

5.2 Other elements

For the other objects, the trees are fixed, the vehicles have a simple animation including the movement along the lines, created by modifying the z-value of the vehicle's position. Since the playing field has been limited, the various vehicles, once the game limits have been exceeded, disappear, and then reappear at the beginning of the line to repeat their movement. With a probability of 50%, vehicles can start from both the right and the left.

As for the coins, on the other hand, they have a simple up and down animation.

6. Lights

Two different lights were applied, one directional and one ambient. The ambient light illuminates all the streets, while the directional light serves to better illuminate the character.



Game with Ambient Light



Game without Ambient Light

7. Browsers

The application has been tested on the following browsers:

- Google Chrome 104.0.5112.102
- Opera 89.0.4447.83
- Safari 15.6 (iPhone XR)

You can also use the application on mobile phone.