

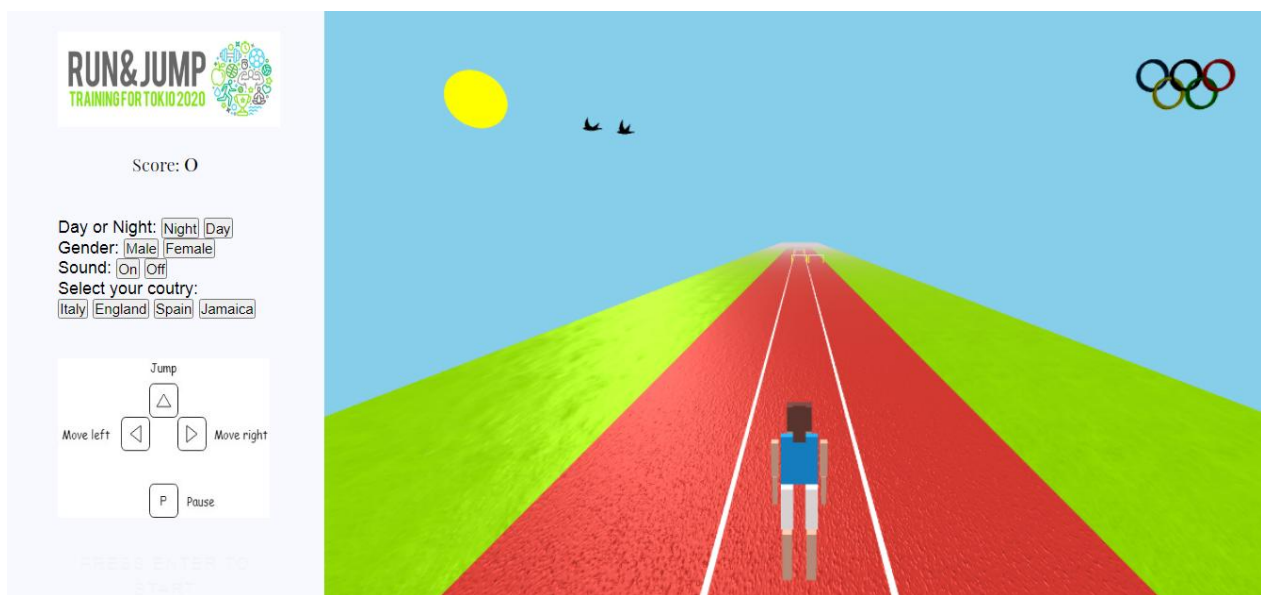
# INTERACTIVE GRAPHICS – FINAL PROJECT

Guiso Carlotta, 1801584

## Project idea

The project consists of a simple game where the player controls the position of a moving athlete who is training to win the next Olympics and has the goal of jumping obstacles. The athlete runs continuously, only stopping once it collides with an obstacle. The game's difficulty, consist in increasing the number of obstacles, increase over time.

You can play here: <https://sapienzainteractivegraphicscourse.github.io/final-project-cg/>



## Framework

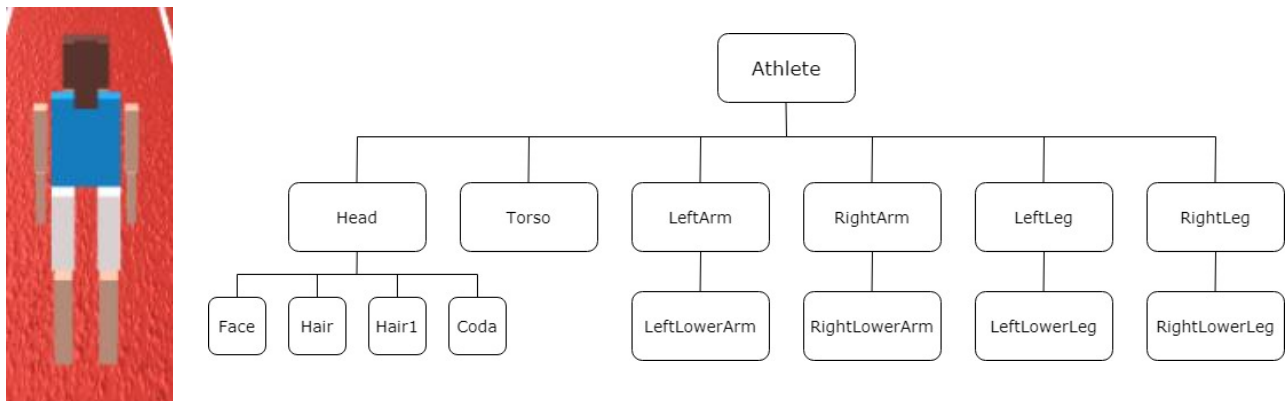
The entire application is built with the framework Three.js, a high level JavaScript library that allow to build 3D content on webpages using WebGL.

## Models

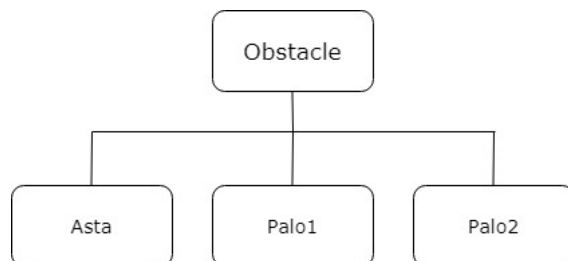
There are five models in the scene: two hierarchical models that are the athlete model, obstacle model and three imported models that are the olympic logo model, birds model and stars model.

### Athlete

It is the most important model, and is composed by different parts, each of them is a basic structure. The structure of the model is the following:



### Obstacle



### Imported models



Where the bird model appear only in the “Day” mode; star model appear only in the “Night” model and olympic logo model that at the start of the game it rotates on itself.

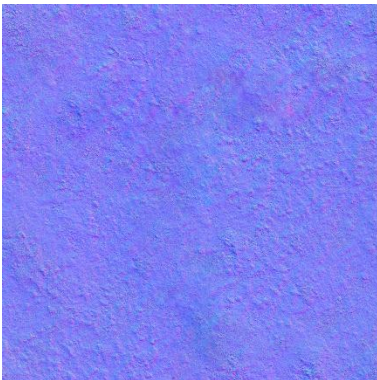
## Lights and Textures

### Lights

There are a hemisphere light that simulate an ambient environment light and a directional light that simulate a sun/moon light.

### Textures

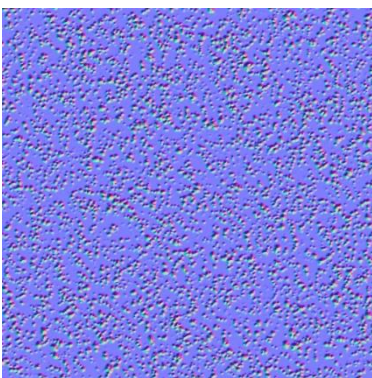
Textures are used in the project in order to obtain a realistic surface. In this case are used a normal texture to simulate a ground surface and a track surface. These normal textures are light responsive so they give the feeling of a 3D surface.



Ground normal texture



Final result



Track normal texture



Final result

## Animation

The mechanisms of the game allow for three types of movements:

- Two sideways, left and right
- One upward, jump

The animation of the athlete is done using a periodic function that create a sine wave with control over period/frequency, phase, amplitude and offset. This function is applied to each part of the athlete body and simulates the running mode. The function is the following:

```
function RunMode(frequency, minimum, maximum, phase, time) {  
  var t = time * frequency * Math.PI * 2; // get phase at time  
  t += phase * Math.PI / 180; // add the phase offset  
  var v = Math.sin(t); // get the value at the calculated position in the cycle  
  var amplitude = 0.5 * (maximum - minimum);  
  var offset = amplitude * v;  
  v += offset;  
  var average = (minimum + maximum) / 2;  
  return v + average;  
}
```



Running



Jumping

To manage the collision detected were built invisible boxes around the athlete and obstacles. The boolean function check if a specific obstacle is colliding with the athlete or not given the coordinate space.

```
/*Ritorna true se l'atleta tocca l'ostacolo*/  
function collisionsDetected() {  
  var aMinX = atleta.position.x - 5;  
  var aMaxX = atleta.position.x + 5;  
  var aMinY = atleta.position.y - 10;  
  var aMaxY = atleta.position.y + 10;  
  var aMinZ = atleta.position.z - 2;  
  var aMaxZ = atleta.position.z + 2;  
  for (var i = 0; i < objects.length; i++) {  
    if (objects[i].collides(aMinX, aMaxX, aMinY, aMaxY, aMinZ, aMaxZ)) {  
      return true;  
    }  
  }  
  return false;  
}
```

```
this.collides = function(minX, maxX, minY, maxY, minZ, maxZ) {  
  var oMinX = self.mesh.position.x - 200;  
  var oMaxX = self.mesh.position.x + 200;  
  var oMinY = self.mesh.position.y;  
  var oMaxY = self.mesh.position.y + 500;  
  var oMinZ = self.mesh.position.z - 200;  
  var oMaxZ = self.mesh.position.z + 200;  
  return oMinX <= maxX && oMaxX >= minX && oMinY <= maxY  
    && oMaxY >= minY && oMinZ <= maxZ && oMaxZ >= minZ;  
}
```

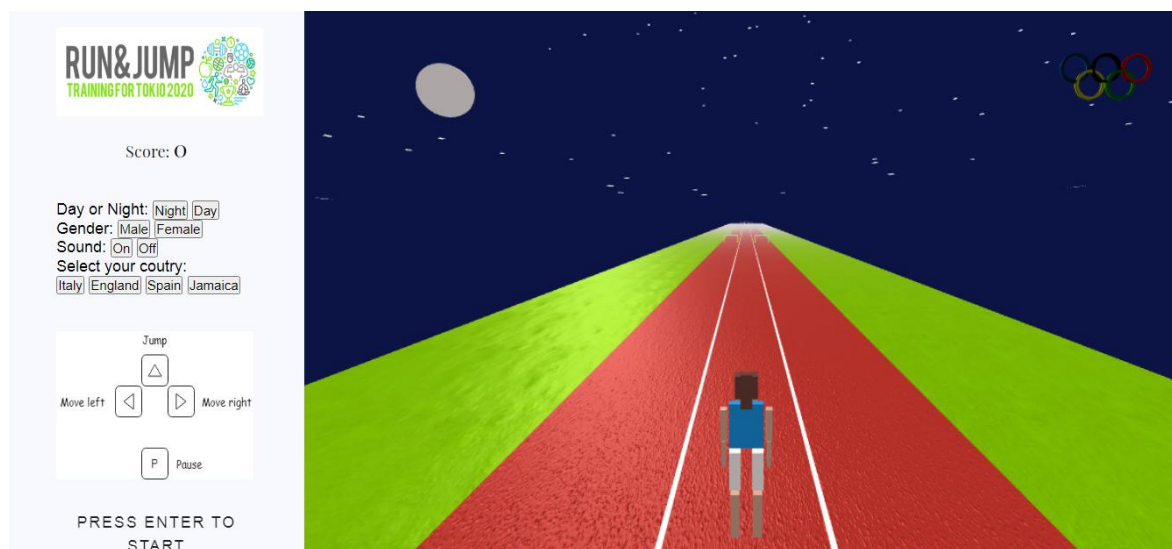
## User interaction

As soon as the initial page loads, the user can do different things like:

- Chose a Day or Night mode
- Chose the gender, female or male
- Play or pause a song
- Chose its country
- Start the game

When the game starts, the user can still use these buttons and also pause the game. When there is a collision the user can press enter to try again.

**Note:** the song takes a few seconds to load.



Night mode



Male



Female, Italy



England



Spain



Jamaica

## Credits

The models are available on the following links:

- Olympic logo: <https://sketchfab.com/3d-models/olympic-games-symbol-materials-30df0825b324422b974484f1fe444b62>
- Stars: <https://sketchfab.com/3d-models/extracted-minecraft-java-editions-stars-c8868b7ffc4a473eb5cd9203a59e3650>
- Birds: <https://sketchfab.com/3d-models/birds-3a9bb97be78944f9bffc23fb25c2154e>

All the other models not here mentioned were custom made.

Sound: <https://www.bensound.com/royalty-free-music/track/creative-minds>

For RunMode function the following link has been studied:

<https://riptutorial.com/javascript/example/10173/periodic-functions-using-math-sin>