

Racing Game

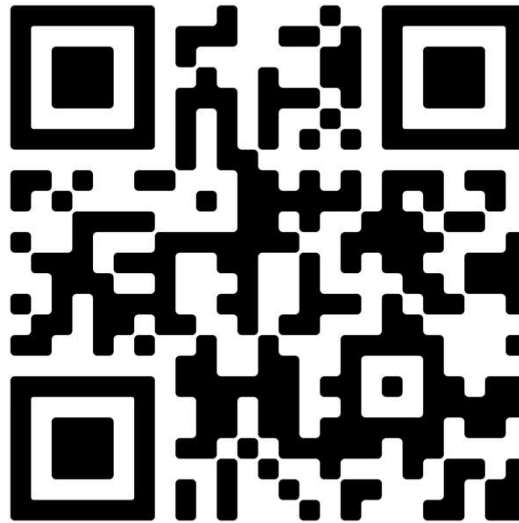
Diogo Paiva+ 103183

Introduction to Computer Graphics – 2022/2023 – Project

Introduction

- 2-Player Racing Game
- Each player controls a car
- Left car with WASD
- Right car with Arrows
- Created using Three.js and Ammo.js

Deployment



<https://diogopaiva21.github.io/icgproject/>

Models

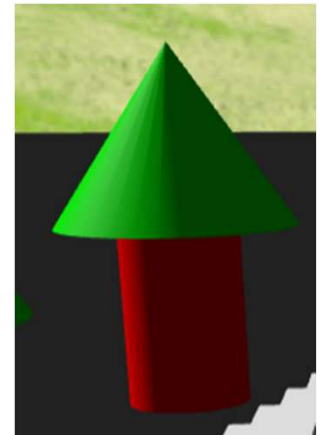
- **Cars**

- BoxGeometry
- CylinderGeometry
- MeshPhongMaterial



- **Trees**

- CylinderGeometry
- ConeGeometry
- MeshPhongMaterial



- **Walls**

- BoxGeometry
- MeshPhongMaterial



- **Arrow**

- CylinderGeometry
- MeshBasicMaterial



Animation

- **Cars**

- The cars move depending on the keys pressed

- **Camera**

- Moves with the car

```
const view = views[0];
const camera = view.camera;
chassisMesh.add(camera);
// Camera more close to the car
camera.position.set(0, 5, -20);
// Camera look at the car
const controls = view.controls;
controls.target = new THREE.Vector3(p.x(), p.y(), p.z());
controls.update();
```

```
if (actions.acceleration) {
    if (speed < -1)
        breakingForce = maxBreakingForce;
    else engineForce = maxEngineForce;
}
if (actions.braking) {
    if (speed > 1)
        breakingForce = maxBreakingForce;
    else engineForce = -maxEngineForce / 2;
}
if (actions.left) {
    if (vehicleSteering < steeringClamp)
        vehicleSteering += steeringIncrement;
}
else {
    if (actions.right) {
        if (vehicleSteering > -steeringClamp)
            vehicleSteering -= steeringIncrement;
    }
    else {
        if (vehicleSteering < -steeringIncrement)
            vehicleSteering += steeringIncrement;
        else {
            if (vehicleSteering > steeringIncrement)
                vehicleSteering -= steeringIncrement;
            else {
                vehicleSteering = 0;
            }
        }
    }
}
}
```

Illumination

- Ambient Light is present in the environment
 - **AmbientLight**
 - Soft White Light
 - Intensity = 1
- Directional Light to simulate sun
 - **DirectionalLight**
 - White Light
 - Intensity = 1
- All objects, except the arrows, receive and cast shadows

```
var ambientLight = new THREE.AmbientLight( 0x404040 );  
scene.add( ambientLight );  
  
var dirLight = new THREE.DirectionalLight( 0xffffff, 1 );  
dirLight.position.set( 10, 10, 5 );  
scene.add( dirLight );
```

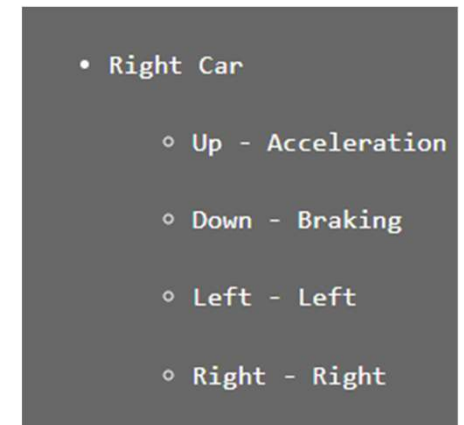
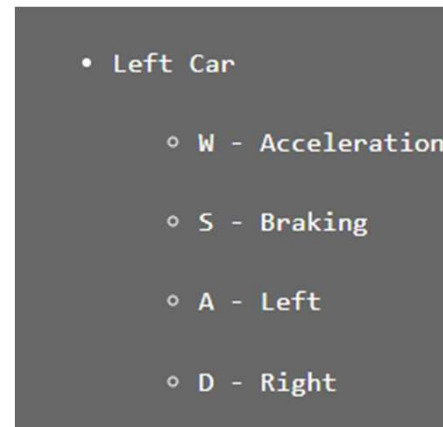
User Interaction

- **UI Elements**

- Information Panels
- Time of lap for each car
- Velocity of each car

- **Keyboard actions**

- **W** – Acceleration Left Car
- **S** – Braking Left Car
- **A** – Left Turning Left Car
- **D** – Right Turning Left Car
- **Up** – Acceleration Right Car
- **Down** – Braking Right Car
- **Left** – Left Turning Right Car
- **Right** – Right Turning Right Car
- **H** – Toggle Info Panels

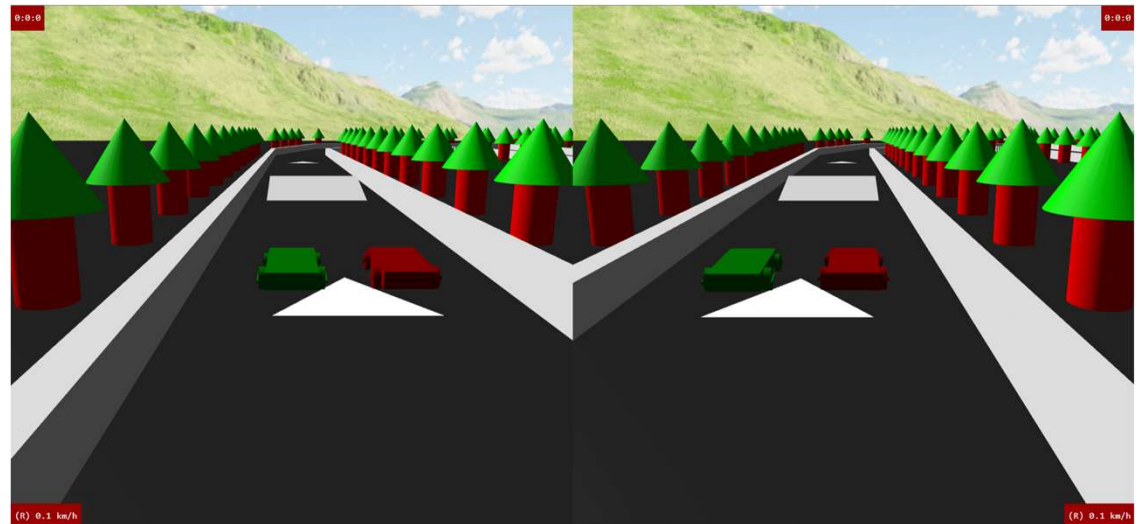


Development

- Coded in **Javascript**
- Only 1 file divided with multiple functions
- Deployed in **Github Pages**
- Problems: Working with ammo.js, creating UI elements, and too heavy with imported models

Conclusions

- Understanding 3D Graphics and Three.js
- Future Applications



- Future Work: Improve Car Model and Add Checkpoints

References

- Three.js Documentation
 - <https://threejs.org/>
- Ammo Example
 - https://rawcdn.githack.com/kripken/ammo.js/99d0ec0b1e26d7ccc13e013caba8e8a5c98d953b/examples/webgl_demo_vehicle/index.html
- Examples from ICG classes
 - <https://elearning.ua.pt/?redirect=0>
- Repository
 - <https://github.com/DiogoPaiva21/icgproject>
- Deployment
 - <https://diogopaiva21.github.io/icgproject/>