

## Work Journal:

Each task required careful implementation and testing to ensure smooth functionality and optimal performance. From creating the basic mechanics of car movement and jump methods to adding visual effects, obstacles, shooting mechanics, and enhancing the UI, every step demanded attention to detail and rigorous development. Despite the time investment, these updates were essential in refining the overall gameplay experience and delivering a polished final product.

My first step in this project was to register for the Unity Student plan. In this plan, I completed three missions to learn about Unity.

**Mission one**, which took approximately **2 hours**, this mission guided me from first install of the Unity editor to creating my first Unity projects to play and share with others.

**Mission two**, which took about **21 hours**, teaches what Unity is and how it's used while creating simple 2D and 3D real-time experiences from scratch.

**Mission three**, which took approximately **2 hours**, Learn how Unity fits in with the broader creative industry of real-time development.

So, I can say it took about **25 hours** just to learn how to develop a game using Unity engine. After this preparing I started to develop my game step by step, every two week I tried to push a new Update to my GitHub repository.

Here more details about each one:

**Update 1-** this update took **15 hours** (6 hours just for creating basic mechanics) because my understanding of Unity's physics system and scripting wasn't so good . In this update:

Created the basic mechanics of car movement and jumping using the wheel collider built in Unity.

Implemented the car physics component using the rigid body component to simulate physical movement affected by gravity.

Designed a camera script to follow the car with an offset in the x and y-axis while staying on the same z-axis.

Set up a basic terrain and created a layer called "Ground" to enable the car to recognize the ground surface using Raycast and prevent unlimited jumping in the air.

**Update 2-** this update took **13 hours**. In this update:

Limited the car rotation to a range of 45 and -45 degrees in the y-axis Euler angles, allowing the car to move forward across the road.

Implemented auto-rotation of the car towards the direction of the road when no input is given or when reaching the boundaries on the sides.

Added point lights to the car, as well as point and directional lights to the environment for improved visuals.

Utilized the particle system component to emit particles from the exhaust based on the car's speed.

Used the Trail Renderer component to visualize the car's path.

Added boundaries by clamping the z-rotation of the car to prevent the player from moving off the road.

**Update 3-** This update took **9 hours**. In this update:

Introduced obstacles to the road by creating a wall script that detects collisions with the car. This is achieved using the `OnTriggerEnter` event triggered by the "Player" tag.

Upon collision, the wall is destroyed by adding a dynamic Rigidbody component to the wall's mesh components, causing it to explode.

Included a machine gun component on the car (currently non-functional).

Implemented post-processing effects to enhance the visual scene.

Added a Particle System component to visualize the dead zone area.

**Update 4-** This update took **12 hours**. In this update:

Created a gun shooting mechanic script, utilizing `Transform.Translate` to move the gun in the desired direction.

Set the shooting action to be triggered by pressing the "F" key.

Instantiated a bullet every 0.5 seconds and added a Trail Renderer component to visualize its trajectory.

Enabled the `onTriggerEnter` event to destroy the car when it enters the dead zone or collides with walls.

**Update 5-** This update took **22 hours**; 8 hours just to fix the camera perspective. In this update:

Adjusted the camera mechanic to a top-down perspective.

Developed a shooter that fires balls from the air using the Rigidbody component. The shooter detects the player using Physics.OverlapSphere and aims directly at the player with a randomized ratio.

Implemented a shield box that activates a shield on the car when collected using the OnTriggerEnter event. The shield remains active for a random duration and is deactivated using the Invoke method in Unity.

Set the shield within the obstacles script to protect the player from any obstacles in the scene.

**Last Update-** This update took **25 hours**. In this update:

Implemented final touches on the user interface (UI) by incorporating the main menu script and linking the methods to UI buttons.

Created two additional levels within the scene, each featuring different roads and crossings.

Enhanced the walls by introducing a health system, requiring multiple bullets to destroy them.

finish and design the read.me file.

creating a first release for my game in the Github, which allow people to download it and play.