

University of Zagreb – Faculty of Electrical  
Engineering and Computing

# **Development of a Car Game Mobile Application**

Max Henrotin

## **Project Report**

**Course:** Project C + E

**Mentor:** Tomislav Jagušt

**Semester:** Winter 2024/2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Game Concept</b>	<b>2</b>
<b>3</b>	<b>Development Timeline</b>	<b>3</b>
3.1	Week 1 (18h of work) - Setup and Initial Implementation . . . . .	3
3.2	Week 2 (16h of work) - Physics and Camera Setup . . . . .	3
3.3	Week 3 (18h of work) - Car Behavior Optimization and Game Structure .	3
3.4	Week 4 (9h of work) - Game Logic . . . . .	4
3.5	Week 5 (18h of work) - Drifting mechanics . . . . .	4
3.6	Week 6 (27h of work) - Design and Tilemap Integration . . . . .	4
3.7	Week 7 (17h of work) - Prefabs and Level Design . . . . .	5
3.8	Week 8 (18h of work) - Game Structure and Menus . . . . .	5
3.9	Week 9 (11h of work) - UI Enhancements . . . . .	5
3.10	Week 10 (20h of work) - Data Saving and Medals . . . . .	6
3.11	Week 11 (18h of work) - Medals Display and Animations . . . . .	6
3.12	Week 12 (10h of work) - Build Challenges . . . . .	6
3.13	Week 13 (10h of work) - Continued IOS Build Struggles . . . . .	6
3.14	Week 14 (10h of work) - Persistence in Testing . . . . .	6
3.15	Week 15 (41h of work) - Android Build and Release, Binary Data Storage and Personal Best Ghost . . . . .	7
3.16	Week 16 (15h of work) - Drifting Warning and Data Security . . . . .	7
3.17	Week 17 (10h of work) - Report Writing . . . . .	7
<b>4</b>	<b>Conclusion</b>	<b>7</b>
<b>5</b>	<b>Project Resources</b>	<b>8</b>
<b>6</b>	<b>Acknowledgments</b>	<b>8</b>

# 1 Introduction

This report documents the development process of a minimalist racing game created as part of the course projects C and E. The game is designed to be skill-based, where the player controls a car by sliding their finger on a touchpad to navigate through a track and reach the finish line as quickly as possible. The project was developed using Unity and C#, with a focus on creating a smooth and engaging gameplay experience.

## 2 Game Concept

The game is inspired by the mechanics of *Trackmania*, where the player's skill is the primary factor in achieving fast times. The core mechanics include:

- **Control Scheme:** The player slides their finger on a touchpad at the bottom of the screen. The car turns more sharply as the finger moves toward the edges of the pad but slows down simultaneously. If the finger is in the middle, the car accelerates in a straight line.
- **Sliding Mechanic:** If the player's finger moves past a sliding threshold (marked by darker zones on the pad), the car enters a slide, loses steering control, and slows down until the finger is out of those bounds.
- **Dynamic Sliding Threshold:** The sliding threshold adjusts based on the car's speed. At higher speeds, the threshold moves closer to the center of the pad, making it harder to maintain control.

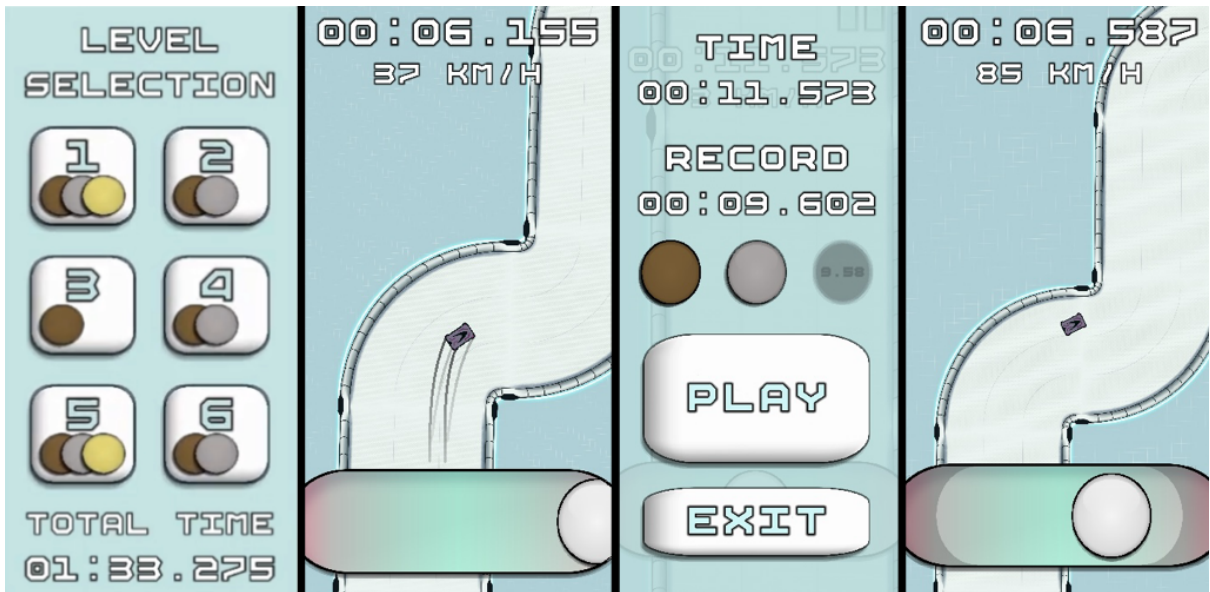


Figure 1: Game overview

### 3 Development Timeline

The development process spanned 17 weeks, with each week dedicated to specific tasks and milestones. Below is a summary of the work completed each week.

#### 3.1 Week 1 (18h of work) - Setup and Initial Implementation

In the initial stages of my project, I focused on researching and selecting the best software tools to use and how to proceed efficiently. After choosing the necessary tools, I set up the development environment by installing UnityHub, Unity 2022.3.3f1, Unity Remote 5, iOS Builder for Unity, Microsoft Visual Studio Community 2022, and several Unity extensions, packages, and libraries, ensuring everything was functioning properly. I then watched my first Unity tutorials, exploring the Unity environment to understand how it operates, how to write code within it, and how to interact with its various components using Visual Studio. I also began the project by implementing a button to switch between interfaces and a slider to move an object within the scene. These steps helped lay a solid foundation for the development of the game.

**Video illustration:**

[https://drive.google.com/file/d/1ir\\_yfcb-g76nJ1PDXZiBGdtnNaiiSXFN/view?usp=drive\\_link](https://drive.google.com/file/d/1ir_yfcb-g76nJ1PDXZiBGdtnNaiiSXFN/view?usp=drive_link)

#### 3.2 Week 2 (16h of work) - Physics and Camera Setup

I created the first sprites in Procreate and began placing them in the scene. Then, I focused on implementing basic car physics using a custom script, while trying to understand how everything connects. I also explored the camera and canvas concepts. After discovering the Rigidbody component, I realized it would be more suitable for my car object than my previous approach. Tutorials helped clarify that the Canvas is only for buttons and menus, not for objects like the car or background, and I learned how to manage layer levels and Z positions. I then added a camera that follows the car and integrated the Rigidbody component, though it caused some loss in previous behaviors.

**Video illustration:**

[https://drive.google.com/file/d/1JzoVJ-kdL7TZZwr9k9mea0lhxFxWb0cN/view?usp=drive\\_link](https://drive.google.com/file/d/1JzoVJ-kdL7TZZwr9k9mea0lhxFxWb0cN/view?usp=drive_link)

#### 3.3 Week 3 (18h of work) - Car Behavior Optimization and Game Structure

This week, I focused on refining the car's physics and camera behavior. I implemented ground friction to restore realistic car movement and fine-tuned various parameters to achieve optimal performance. Additionally, I optimized the car's turning mechanics to

prevent unrealistic behavior, such as infinite rotation. A significant breakthrough was resolving a laggy camera issue by extrapolating the car's position in the Rigidbody component, which led to smoother car movement. I also added a speed counter, a chronometer, and a finish line with associated logic, improving the game's structure and modularity. Finally, I introduced a rock object to test colliders and ensure proper interaction with the car.

**Video illustration:**

[https://drive.google.com/file/d/1IkKFRu5MEbCtGEgpQnnwL5E3DQquY4TP/view?usp=drive\\_link](https://drive.google.com/file/d/1IkKFRu5MEbCtGEgpQnnwL5E3DQquY4TP/view?usp=drive_link)

### **3.4 Week 4 (9h of work) - Game Logic**

I began by allowing the car to move backward slowly to prevent it from getting stuck, which required rethinking the CarMovement script. I then added walls to create the first real track and implemented a restart mechanic integrated into a new GameSupervisor script. A countdown was introduced before the chronometer starts, and the game's architecture was cleaned up for better control and modularity. Debugging the restart mechanic revealed issues with UI layering and code structure, which were resolved to ensure smooth functionality.

**Video illustration:**

[https://drive.google.com/file/d/16hjViVzSUBAd3wapto5b8wyME04Z1NGs/view?usp=drive\\_link](https://drive.google.com/file/d/16hjViVzSUBAd3wapto5b8wyME04Z1NGs/view?usp=drive_link)

### **3.5 Week 5 (18h of work) - Drifting mechanics**

I improved the slider's acceleration response, making the car accelerate more and brake only at the extremes of the slider. A bug causing awkward backward turning after collisions was fixed by increasing the car's angular drag. Diagonal collision issues were addressed by adding a physics material with adjusted bounciness and friction. I also experimented with separating front and rear wheel slipperiness but settled on moving the car's center of mass forward as an easier solution. Finally, I implemented a drifting mechanic and added visual skid marks to enhance the game's realism. Those slides were triggered by the addition of a drifting threshold proportional to the car's speed.

**Video illustration:**

[https://drive.google.com/file/d/14ehEyK56-yTq1u48-wwHI9P5a9yVVVTa/view?usp=drive\\_link](https://drive.google.com/file/d/14ehEyK56-yTq1u48-wwHI9P5a9yVVVTa/view?usp=drive_link)

### **3.6 Week 6 (27h of work) - Design and Tilemap Integration**

This week, I dedicated significant time to learning color theory, game design, and asset creation. I chose a futuristic, minimalist, and arcade-style aesthetic for the game. Using

Procreate, I designed the game's map with a tile-based system for easy track creation. I integrated the tilemap system in Unity, complete with Tilemap Colliders, enabling infinite track variations. Additionally, I experimented with camera rotation and resolved rendering bugs introduced by the new map design.

**Video illustration:**

[https://drive.google.com/file/d/1FgVq7v3YXPavLFDhBHW309d9VPiz45IC/view?usp=drive\\_link](https://drive.google.com/file/d/1FgVq7v3YXPavLFDhBHW309d9VPiz45IC/view?usp=drive_link)

### **3.7 Week 7 (17h of work) - Prefabs and Level Design**

I resolved a collision bug at tile boundaries by implementing multiple capsule colliders for the car. The slider and car designs were updated. I fine-tuned gameplay parameters and addressed resolution issues by modifying the canvas behavior to adapt to different screen sizes. The slider handle was improved to prevent it from going out of bounds, and a Mask component was added to the slider for better visual rendering.

**Video illustration:**

[https://drive.google.com/file/d/1paMNb8C48D9IESJicmVHR\\_TslRo1BCMz/view?usp=drive\\_link](https://drive.google.com/file/d/1paMNb8C48D9IESJicmVHR_TslRo1BCMz/view?usp=drive_link)

### **3.8 Week 8 (18h of work) - Game Structure and Menus**

I improved the design of the car, walls, and finish line while refining gameplay mechanics for a better player learning curve. All game elements were converted into prefabs, enabling easy level creation. A home menu with level selection buttons and an in-game pause menu were implemented. Each level was given its own scene, and the game's structure was organized for better modularity.

**Video illustration:**

[https://drive.google.com/file/d/1B\\_t9VGeSSzpV9LUHFrp6RFisH\\_Zcz-V6/view?usp=drive\\_link](https://drive.google.com/file/d/1B_t9VGeSSzpV9LUHFrp6RFisH_Zcz-V6/view?usp=drive_link)

### **3.9 Week 9 (11h of work) - UI Enhancements**

I improved functionality, UX, and maintainability. The game now pauses automatically when the menu opens. Key events are centralized in GameSupervisor, including a finish line popup for better flow. UI enhancements include a fade-in menu, real-time race time display, new fonts, redesigned buttons, and an updated color scheme. I also fixed a pause bug by properly resetting Time.timeScale. Additionally, I created a button prefab to streamline future updates and ensure consistency.

**Video illustration:**

[https://drive.google.com/file/d/1h-y3-\\_JUORsZAK\\_4qbNkMQvKgN\\_ntclU/view?usp=drive\\_link](https://drive.google.com/file/d/1h-y3-_JUORsZAK_4qbNkMQvKgN_ntclU/view?usp=drive_link)

### **3.10 Week 10 (20h of work) - Data Saving and Medals**

I added outlines and shadows to buttons and imported new fonts to better suit the game's aesthetic. Player progress, including personal best times, is now saved using PlayerPrefs, and a total time score is displayed on the home screen. The slider and handler visuals were improved, and the camera now zooms out as the car speeds up. A public static script was created to store global constants, and medals were introduced to reward players for achieving specific times.

**Video illustration:**

[https://drive.google.com/file/d/1UCRt5X4tMBHYuGCVtHnxkJ7RAsCVw91a/view?usp=drive\\_link](https://drive.google.com/file/d/1UCRt5X4tMBHYuGCVtHnxkJ7RAsCVw91a/view?usp=drive_link)

### **3.11 Week 11 (18h of work) - Medals Display and Animations**

Medals were turned into prefabs, and an interface was created to display earned medals and their required times. I attempted to modify the drifting zone system but settled on improving the existing gauges. I played the game extensively to set competitive yet achievable medal times. Animations were added to enhance the medal reward display, though working with Unity's Animator proved challenging.

**Video illustration:**

[https://drive.google.com/file/d/1bryt0DTCidUXZ7vH\\_1U2K1\\_x78Uf\\_nNz/view?usp=drive\\_link](https://drive.google.com/file/d/1bryt0DTCidUXZ7vH_1U2K1_x78Uf_nNz/view?usp=drive_link)

### **3.12 Week 12 (10h of work) - Build Challenges**

I attempted to build the game using Xcode for iOS testing but faced numerous issues due to the necessity of owning a Mac. After trying various solutions, including virtual machines and remote Mac services, I was unable to resolve the problems in time. This setback highlighted the need for better access to macOS for future development.

### **3.13 Week 13 (10h of work) - Continued IOS Build Struggles**

Despite efforts to use a virtual Mac through Ultra Host, I encountered further issues with macOS compatibility and iPhone connectivity. These challenges significantly delayed the testing process and reduced my motivation.

### **3.14 Week 14 (10h of work) - Persistence in Testing**

I continued trying to upload the game to a phone but made no significant progress. Exhausting all available options, I decided to pivot to Android testing for the time being.

### **3.15 Week 15 (41h of work) - Android Build and Release, Binary Data Storage and Personal Best Ghost**

I successfully installed and tested the game on an Android tablet, resolving SDK issues along the way to prepare it for release on the Google Play Store. After completing the prelaunch phase, I shared a link for early testing with friends, who provided helpful feedback. I also uploaded the project to GitHub for version control, ensuring the safety of my work.

To improve the game's data management, I transitioned from using PlayerPrefs to binary writing in a .dat file for storing more complex data locally. This new approach allowed me to store the player's best game positions and implement a personal best ghost, adding a layer of motivation for players to beat their own records. I faced challenges with serializing Vector3 for the ghost data but resolved this by implementing a static SaveManager class.

#### **Video illustration:**

[https://drive.google.com/file/d/1pqTmERHyddsv5ZYqvbe0GbcaiI88KPQu/view?usp=drive\\_link](https://drive.google.com/file/d/1pqTmERHyddsv5ZYqvbe0GbcaiI88KPQu/view?usp=drive_link)

### **3.16 Week 16 (15h of work) - Drifting Warning and Data Security**

Graphics of the map were optimized, and new color warnings have been added to the slider to warn when the player is getting too close to the sliding limits. I added a method to encrypt the data before saving it locally to prevent tampering. A migration method was also implemented to handle later updates to the save format. The ghost system was further refined, and the code was optimized for performance.

#### **Video illustration:**

[https://drive.google.com/file/d/1hDz4vxqCojxdVkRYATG53F\\_qtvwcZims/view?usp=drive\\_link](https://drive.google.com/file/d/1hDz4vxqCojxdVkRYATG53F_qtvwcZims/view?usp=drive_link)

### **3.17 Week 17 (10h of work) - Report Writing**

This week was dedicated to compiling and documenting the entire development process in this report.

## **4 Conclusion**

The development of this minimalist racing game was a challenging yet rewarding experience. The project involved learning and applying various concepts in game design, problem resolution, and user interface development. Despite the challenges faced during deployment, the game was successfully published on the Google Play Store.



In the future of this project, I would like to add sound effects, improve the car dynamics and graphics to create a more enjoyable experience, and most importantly, implement a global leaderboard system connected to a server.

## 5 Project Resources

- The source code is available as open source on GitHub: <https://github.com/Maxhencerotin/GamaxTrack>.
- The game can be downloaded from the app store by searching for the name: *Gamax Track*.
- For further details or questions regarding the work progress, you can contact me at [max.henrotin@gmail.com](mailto:max.henrotin@gmail.com).

## 6 Acknowledgments

I would like to thank my instructor Tomislav Jagušć for his support and feedback throughout the development process. Special thanks to my friends who tested the game and provided valuable insights.