

Name of Student : Harshad Patel

Enrolment Number : 12002040701057

Name of Student : Hitendra Patel

Enrolment Number : 12002040701066

Name of Student : Hunaid Siamwala

Enrolment Number : 12002040701067

Guided By

Prof. Mihir Rajyaguru

MINI PROJECT (102040601)

A.Y. 2022-23 EVEN TERM

ACKNOWLEDGEMENT

On this great occasion of accomplishment of our project on **GO CAR**, we would like to sincerely express our gratitude to Prof. Mihir Rajyaguru, who has been supported through the completion of this project.

Finally, as one of the team members, I would like to appreciate all my group members for their support and coordination, I hope we will achieve more in our future endeavours.

Abstract

Game designing is an interactive and creative part of modern IT culture. This has motivated various developers to create interesting games. Thus, we aim to develop an entertaining and fun game based on our childhood favour where we are focusing on developing a 2D racing car game, using a process based upon agile development, an evolutionary development method. Our game is a single player game which provides multiplayer functionality as well. The game will consist of various road maps with power boost challenges and speed timing to get user more addictive and entertain. Since it is a racing genre game, the multiplayer gaming will focus on races with friends and let the users compete to be the best. Apart from this even time depended on mode enables players to track and beat their own high scores. The platform of the game is developed using Unity 2D game engine which is multiplatform software building for games and main coding is in C#. Hence whole game development process will cover implementation of real-time graphics, physics engine, network support, as well as sound effects and background music. Thus, we will try to create a game which is extremely entertaining with great graphics to make it visually appealing.

Table of Content

	Chapter	Page No.
1	Introduction	1
	1.1 Problem Statement	1
	1.2 Project summary and introduction	1
	1.3 Aim and objective of project	2
2	System analysis	2
	2.1 Motivation	2
	2.2 Brief Literature survey	2
3	Design: analysis, design methodology	5
	3.1 H/W and S/W requirement	5
	3.2 Program/module specification	5
	3.3 Timeline chart	6
	3.4 UML diagrams	8
4	Implementation	9
	4.1 System flow	9
	4.2 Module specification (Screenshot of working module)	10
	4.3 Results	11
	Conclusion and Future work	12
	References	13

List of Tables

Table No.	Title	Page No.
1.	Inclusive and Exclusive Criteria	4

List of Figures

Figure No.	Name	Page No.
1.	Timeline Chart-1	6
2.	Timeline Chart-2	7
3.	UML Diagram	8
4.	System Flow	9

1. INTRODUCTION

1.1 PROBLEM STATEMENT

- A racing car game for PC using Unity.

1.2 PROJECT SUMMARY AND INTRODUCTION

- Summary
 - A car game created in Unity-a cross platform game engine developed by Unity Technologies best for beginners.
- Introduction
 - Introducing a new Racing Game where we will use Unity input system and create controls for the car by using player inputs with keyboard controls. Developing software applications is a time-consuming process, and with time-consuming processes come high costs. To address this issue, several software development methodologies, agile software development, have become widely used by software developers. One of the software development methodologies is the evolutionary software method, which allows the project to evolve through different stages of the project and this approach worked on our project where we choose to develop a 2D graphic computer game. Some requirements for the computer game development to 2D environment like 2D graphics, graphical effects, multi-platform. We decided to develop this game in Unity 3D platform which gives better working these requirements. The game is a single player type in racing genre. A racing game is one in which the player generally races other vehicles to secure first position. Unlike the racing genre, we feature only the main player car which must beat the time set by AI. To develop 2D game we have need to develop 2D gaming objects, environment and UI, we should use Adobe tools. Adobe tools are also helpful for gaming effects for better graphics. As unity is a multiplatform engine, we decided to develop a game for desktop (windows platform). So, our final product will be running on windows.

1.3 AIM AND OBJECTIVE OF PROJECT

➤ Aim:

- To make a racing car game.

➤ Objective:

- To show a small glimpse of the basics of Unity Game.

2. SYSTEM ANALYSIS

2.1 MOTIVATION

- The idea that gave us a high boost to jump in the world of gaming came while watching advanced sci-fi movies.

2.2 BRIEF LITERATURE SURVEY

- There has been substantial research work that focuses on developing AI-controlled components of game systems, which can approximate or emulate human game playing styles. These components are often referred to as “bots.” The motivation for developing the “bots” is that a human player’s enjoyment in the gaming experience will be higher if he or she can be led to believe that the opponents in the game are other human players. An example of this type of game is the popular multiplayer first person shooter game Counterstrike, in which the objective is to eliminate all players on the enemy or opponent team. Similar AI-controlled components or “bots” also appear in the early video game of Pac Man; the computer-controlled ghosts in the maze can move towards the player-controlled characters because the former incorporated some path finding algorithms. Other AI algorithms that were adopted by games include finite state machines, fuzzy state machines, and decision trees. Decision trees are used to represent the decision-making process involved in games like Checkers or Chess. A decision tree can specify several possible game states given a current state and support a search process for a goal state, where the human player is defeated provided the AI-controlled game can identify the path to the goal state more efficiently than the human player.

Table 1. Inclusion and exclusion criteria.

Inclusion criteria	Exclusion criteria
Involve AI as primary component	No clear data collection method
Involve game play in the design	Only introduction without evaluation
For educational purposes	For target groups with special needs

3. DESIGN: ANALYSIS, DESIGN METHODOLOGY

3.1 H/W AND S/W REQUIREMENT

- H/W requirement
 - Requirements for developing AR applications with Unity boils down to 3 Things:
 - Your laptop or Desktop Computer
 - An SSD Hard Drive (with lots of Free Memory)
- S/W requirement
 - Unity
 - VS code

3.2 PROGRAM/MODULE SPECIFICATION

1. Game Objectives: The objective of the game is to complete each track as quickly as possible while avoiding obstacles and competing with other racers.
2. User Interface: The game's user interface will consist of a main menu, level select screen, and in-game HUD. The main menu will allow players to start the game and exit the game. The level select screen will allow players to choose which track they want to race on. The in-game HUD will display the player's lap time, and position.
3. Game Mechanics: The player will control their car using the keyboard or gamepad, with the ability to accelerate, brake, turn left and right, and drift. The game will feature several tracks with varying layouts and difficulty levels.
4. Audio: The game will feature a soundtrack and sound effects for the car's engine, crashes, and in-game events.
5. Visuals: The game will have realistic 3D graphics, with detailed car models and environments. The camera will follow the player's car and dynamically adjust to provide the best view of the action.
6. Game Flow: The game will have a series of levels with increasing difficulty, and players must complete each level within a certain time limit. Players can earn points based on their finishing position and time, which can be used to upgrade their car's performance or

unlock new tracks. The game will also feature an online leader board for players to compete for the best times.

7. Platforms: The game will be developed using Unity and will be compatible with Windows.

3.3 TIMELINE CHART

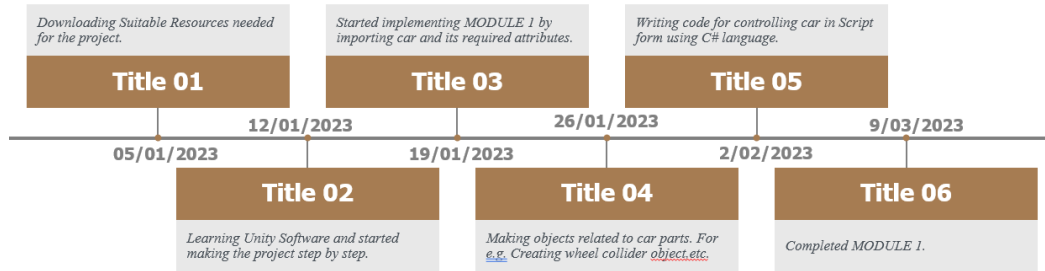


Fig.1 Timeline Chart-1

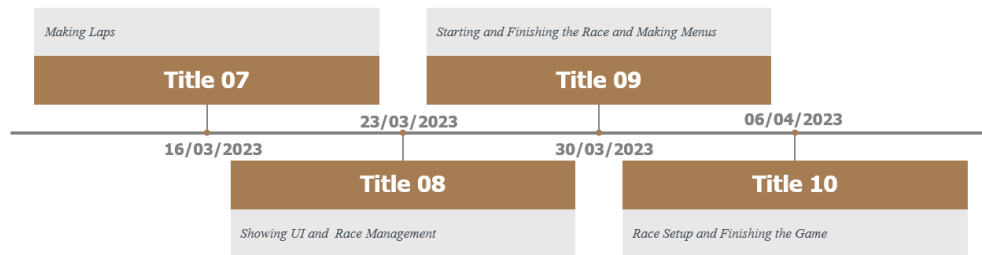


Fig.2 Timeline Chart-2

3.4 UML DIAGRAMS

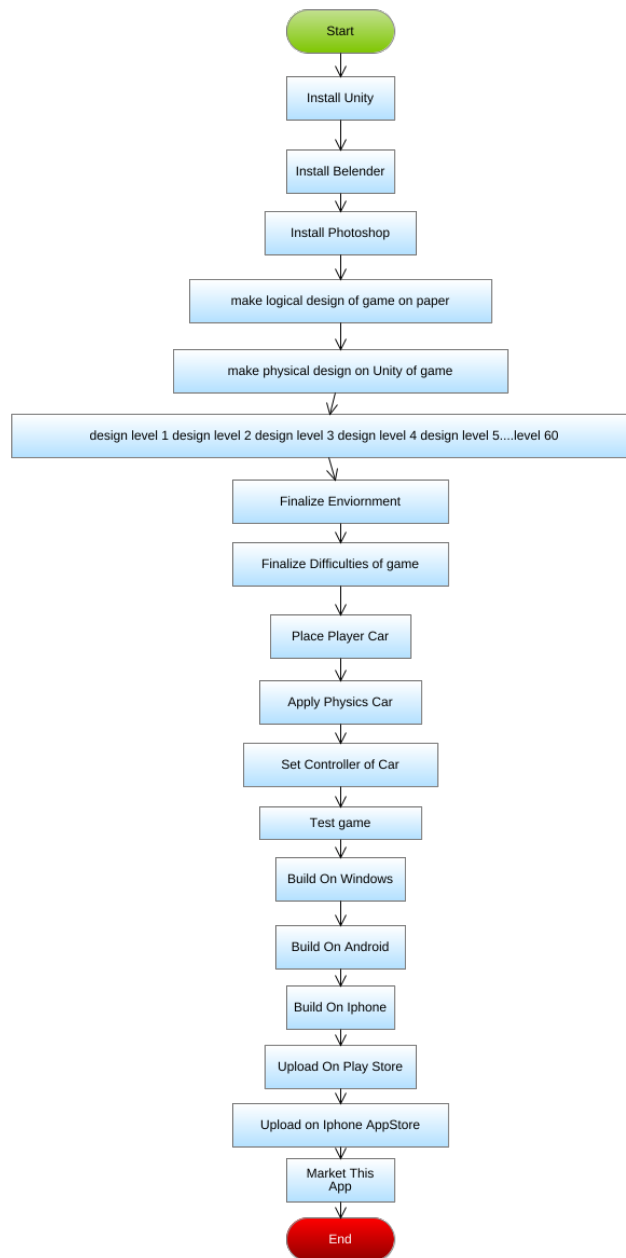


Fig.3 UML Diagrams

4. IMPLEMENTATION

4.1 SYSTEM FLOW

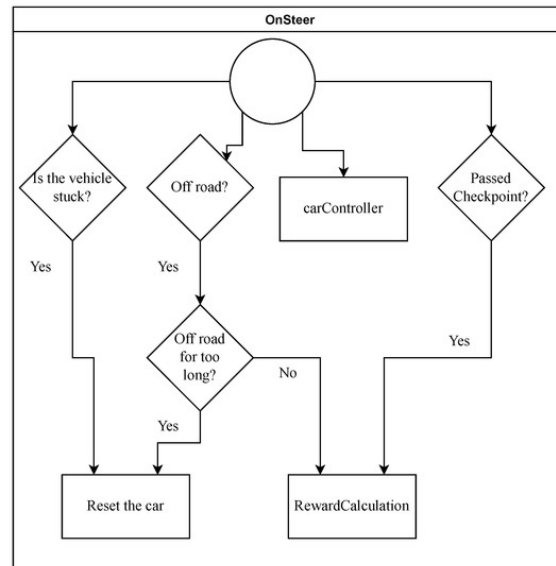
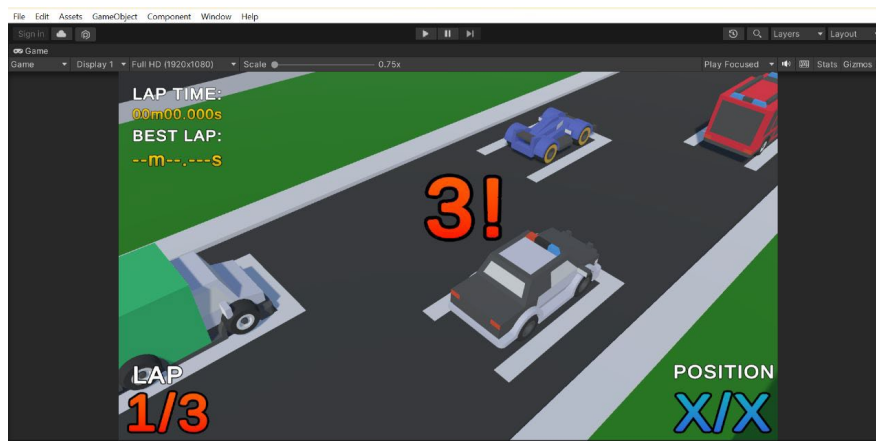


Fig.4 System Flow

4.2 MODULE SPECIFICATION (SCREENSHOTS OF WORKING MODULE)





4.3 RESULTS

1. Physics-based gameplay: A racing car game made using Unity can be made more immersive by incorporating realistic physics-based gameplay. The car's movements, acceleration, and handling should feel natural and responsive.
2. Multiple game modes: Including different game modes such as time trials, races against other players or AI, or drifting challenges can make the game more engaging.
3. Variety of tracks: The game can include a variety of tracks with different themes and layouts, each with its unique challenges and obstacles.
4. Customizable cars: Allowing players to customize their cars with different paints, decals, and upgrades can add a sense of ownership and progression to the game.
5. Good audio and visuals: Including good-quality sound effects and a soundtrack, as well as detailed 3D graphics and environments, can make the game more immersive and engaging.

Overall, a racing car game made using Unity can be successful if it provides engaging gameplay, a variety of content, and a polished user experience.

CONCLUSION AND FUTURE WORK

Racer was tested by over twenty users, who all reported that they thoroughly enjoyed the game. It was observed that there was an approximately equal number of players who were able to win the race against the game-controlled cars. This suggests that not only did the game provide an entertaining gaming experience, but it also provided a reasonably engaging and challenging gameplay. In general, it can be concluded that the Unity platform supported efficient development of the race car game. The Unity platform supports implementing the race car's search for a path on the racetrack with its components of the way-point system, the physics engine, and vector calculation functions, all of which are not available if the implementation was done using traditional AI search techniques. With these Unity components, the developer was able to implement the race car's search for a path on the track with less effort and more efficiently, and the developed race car can successfully mimic human driving behavior.

REFERENCES

1. K. Salen and E. Zimmerman, *Rules of Play: Game Design Fundamentals*, vol. 1, MIT Press, 2004.
2. Y. Liu, T. Alexandrova, and T. Nakajima, “Gamifying Intelligent Environments,” <http://www.dcl.cs.waseda.ac.jp/~yefeng/yefeng/pubs/2011/ubimui11/yefeng.pdf>.
3. L. C. Wood and T. Reiners, “Gamification in logistics and supply chain education: extending active learning,” in *Proceedings of the IADIS International Conference on Internet Technologies & Society (ITS '12)*, P. Kommers, T. Issa, and P. Isaias, Eds., pp. 101–108, IADIS Press, Perth, Australia, November 2012.
4. M. Prensky, “Digital natives, digital immigrants part 1,” *On the Horizon*, vol. 9, no. 5, pp. 1–6, 2001.
5. A. L. Samuel, “Some studies in machine learning using the game of checkers. II—recent progress,” *IBM Journal of Research and Development*, vol. 11, no. 6, pp. 601–617, 1967.
6. D. A. Waterman, “Generalization learning techniques for automating the learning of heuristics,” *Artificial Intelligence*, vol. 1, no. 1-2, pp. 121–170, 1970.