

Project Title

ASCII Car Racer: A Console-Based Racing Game

Group Members

- Rayyan Asif (24K-0993)
- Akbar Younsi (24K-0947)
- Ali Hasnain (24K-0696)

Submission Date

[Insert Date]

1. Executive Summary

Overview:

ASCII Car Racer is a console-based racing game where players control a car represented using ASCII characters. The player must dodge obstacles as they navigate a scrolling track. The game challenges the player's reflexes by increasing speed and complexity as the game progresses.

Key Findings:

The project successfully demonstrates real-time input handling, obstacle generation, collision detection, and score tracking in a modular OOP-based C++ application. It also lays a foundation for a future GUI-based version using frameworks like SDL2 or SFML.

2. Introduction

Background:

Games are a powerful medium for applying OOP concepts. This project uses a retro-style ASCII racing game to demonstrate encapsulation, inheritance, polymorphism, and abstraction in C++. The topic is both fun and technically challenging, making it ideal for an OOP Lab course.

Project Objectives:

- Create an interactive ASCII-based racing game in C++.
- Implement keyboard controls to maneuver the car.

- Dynamically generate obstacles and increase game difficulty over time.
 - Detect collisions and manage game states.
 - Record and display scores.
 - Explore a future transition to GUI using C++ frameworks.
-

3. Project Description

Scope:

Included:

- Real-time player control using keyboard.
- Randomized obstacle generation and collision detection.
- Score tracking and speed progression.
- Use of multiple lanes and power-ups.

Excluded:

- Full GUI-based implementation (planned as future upgrade).
- Sound or advanced graphics.

Technical Overview:

- **Language:** C++
 - **Compiler/IDE:** Visual Studio / VS Code
 - **Potential GUI Frameworks:** SDL2, SFML, Qt (for future development)
-

4. Methodology

Approach:

- Weekly meetings for design, development, and testing
- Task delegation based on specialization (e.g., UI, logic, classes)
- Agile-inspired cycle: design → implement → test → review

Roles and Responsibilities:

- **Rayyan Asif:** Core game logic, class architecture, and main loop
 - **Akbar Younsi:** Obstacle mechanics, collision detection, and scoring
 - **Ali Hasnain:** Player controls, UI design, and power-up integration
-

5. Project Implementation

Design and Structure:

The project uses a modular class-based design:

- Car class: Handles player control
- Obstacle class: Manages obstacle behavior
- Game class: Central controller managing game loop, collisions, and UI

Functionalities Developed:

- ASCII track rendering with walls and lanes
- Car movement (A/D keys)
- Random obstacle generation
- Collision detection
- Speed/difficulty scaling
- Score tracking and game-over screen
- Power-up mechanisms (e.g., shields, speed boost)

Challenges Faced:

- Managing real-time input without GUI
 - Ensuring smooth difficulty scaling
 - Preventing overlapping of obstacles and power-ups
 - Balancing gameplay for fun and challenge
-

6. Results

Project Outcomes:

Successfully implemented a fully playable ASCII car racing game using OOP principles. The game dynamically adjusts difficulty and offers a basic score system.

Screenshots and Illustrations:

(Insert relevant screenshots or ASCII mockups here showing track, car, and obstacles)

Testing and Validation:

- Manual testing of car movement, collision detection, and score updates
 - Stress testing with multiple obstacles at high speeds
 - Validation of power-up mechanics under various conditions
-

7. Conclusion**Summary of Findings:**

This project not only met its technical goals but also served as an excellent learning opportunity in applying OOP. Key concepts like encapsulation, polymorphism, and dynamic memory were reinforced through hands-on development.

Final Remarks:

ASCII Car Racer proved that even simple console-based applications can be engaging and educational. Future improvements include a GUI version and leaderboard system. The experience gained is directly applicable to more advanced software and game development using C++.