

# Requirements for Design Project: Car Racing Game

## Table of Contents

1. Introduction	2
a. Purpose	2
b. Scope	2
2. Functional requirements	2
a. Race Simulation	2
b. Speed & Time Calculation	2
c. User Interface	3
d. Winner Declaration	3
e. Data Display	3
3. Data	3
a. Input & Output	3
b. Storage	4
4. Usability	4
a. UI Elements	4
b. UI Considerations	5
c. Error Prevention & User Guidance	5
5. Reliability & Availability	5
a. Error Handling & Feedback	5
b. Data Backup	5
6. Performance	5
a. Responsiveness	5
b. Compatibility & Optimization	5

## **1. Introduction**

### **a. Purpose**

The purpose of this document is to define the functional requirements and system design for a simple car racing game prototype. The game will simulate a race where multiple cars travel through different locations (named A, B, C, D), tracking speed, time, and distance. The winner is the car that completes its assigned path in the shortest time.

### **b. Scope**

This project focuses on designing a car racing simulation with animated movement, location-based travel, and randomized race conditions such as distance, tire wear down, and engine's fuel efficiency. The game will visually display the race progress, car details (speed, stops, time taken), and declare a winner based on performance.

## **2. Functional requirements**

### **a. Race Simulation**

- i. The game will feature multiple cars racing simultaneously.
- ii. Cars will move between predefined stops on a route.
- iii. Movement should be animated or visually represented.

### **b. Speed & Time Calculation**

- i. Each car will have a unique speed based on randomized factors.
- ii. The total time taken by each car is calculated dynamically.
- iii. Factors such as tire wear and engine efficiency will impact speed.

### c. User Interface

- i. A graphical user interface (GUI) will display the race in progress.
- ii. Users can start a race and observe real-time progress.
- iii. Each car's status, including its position, speed, and remaining distance, will be displayed.

### d. Winner Declaration

- i. The system determines the winner based on which car finishes first.
- ii. Once the race ends, a summary screen displays the results.

### e. Data Display

- i. The application will provide information such as:
  - Car names and colors
  - Routes taken (A-B-C-D, or B-A-D-C, etc. )
  - Speed changes due to engine boost or tire wear
  - Total time taken per car
  - The final winner

## 3. Data

### a. Input & Output

- i. Inputs:
  - User starts the race through the GUI.
  - Randomized values determine each car's speed and conditions.
- ii. Outputs:
  - Display of car movements and position updates.
  - Final race results with timers, and winner.

## b. Storage

- i. The system will store race details temporarily during execution.
- ii. No permanent data storage is required for this prototype.

# 4. Usability

## a. UI Elements

- i. Welcome Screen:
  - Users can select the number of cars (minimum: 2, maximum: 12 to avoid overcrowding).
  - Users can customize car attributes, such as:
    - Speed variations (randomized or user-defined max speed).
    - Car durability (engine & tire conditions) may affect performance.
    - Starting and ending locations for each car.
- ii. Race Display:
  - A racetrack visualization with car movement.
  - Each car's current location, speed, and race progress.
  - A live leaderboard showing time elapsed and race positioning.
- iii. Winner Display:
  - A screen displaying the winning car and its total time.
  - A button to restart the race or modify car settings.

### **b. UI Considerations**

- i. The interface will be minimalistic but informative, displaying race progress clearly.

### **c. Error Prevention & User Guidance**

- i. Clear instructions on how the race functions.
- ii. Prevent unintended user interactions during race execution.

## **5. Reliability & Availability**

### **a. Error Handling & Feedback**

- i. The system should handle potential errors such as cars getting stuck at a location.
- ii. If an issue arises, the GUI should provide feedback.

### **b. Data Backup**

- i. Since this is a simulation, no long-term data storage is needed.

## **6. Performance**

### **a. Responsiveness**

- i. The game should run smoothly with no major lag during race execution.
- ii. The UI elements should scale proportionally to the screen size.

### **b. Compatibility & Optimization**

- i. The program will be optimized to run seamlessly in any desktop environment such as Windows, macOS, and Linux.