Aalto University

Project Plan

Prepared by

Kevin Cabras — kevin.cabras@aalto.fi

Petteri Suonpää — petteri.suonpaa@aalto.fi

Michael Chen — michael.chen@aalto.fi

Hanjemma Jeong — hanjemma.jeong@aalto.fi

October 4, 2024

TABLE OF CONTENTS

1. Scope of Work

Features and Functionalities

- **Basic Gameplay**: The core gameplay will be a top-down driving game allowing the player to control toy cars. Simple driving physics will be implemented for fun, accessible controls. Camera control will be implemented to follow the player.
- **Multiple (If there is time) Tracks**: Tracks will be loaded from external files, enabling diversity in game environments.
- **Game Objects**: Interactive game objects (oil spills, jams, boosts, etc.) will impact gameplay dynamics, adding variety and challenge.
- **Additional Gameplay Features**:
    - **Terrain Handling**: Cars will behave differently on various terrains (e.g., asphalt, ice, grass).
    - **Vehicle Variety**: Players can choose between different vehicle types (boats, hovercrafts, helicopters).
    - **Sound Effects**: Audio elements like engine sounds, crashes, and environmental sounds will enhance immersion.
    - **Split-Screen Mode**: Allows two players to compete on the same screen.
    - **Ability System**: Optional weapons like missiles, traps that can affect opponents or speed boost.

Usage and Mechanics

- Players will use a game controller or keyboard for car control. Each car's speed, ability usage, and response to terrain will differ.
- The game objective is to race to the finish line on various tracks, avoiding obstacles and using boosts for speed advantages.

2. High-Level Structure of the Software

Main Modules

- **Game Engine**: Controls physics, and game logic.
- **Track Management**: Handles track loading, track assets, and obstacles.
- **Player Module**: Manages player input and character stats.
- **EnemyAI Module**: Manages the enemy AI that races against the player and other enemies.
- **Physics Module**: Calculates vehicle movement, collisions, and responses to terrain.
- **Sound Module**: Manages sound effects and background music.
- **UI Module**: Provides menu navigation, game HUD.

Main Classes (Initial Outline)

- Game: Manages the main game loop, including game state and scene switching.
- Player: Represents each player, with attributes for position, velocity, weight, and input handling.
- Enemy: Represents each enemy, with attributes for position, velocity, and automatically follows the track and disrupts other enemies and the player.
- Car: Subclass of Player, with properties like handling and speed depending on vehicle type.
- Track: Stores track layout, including road, off-road zones, and obstacles.
- Obstacle: Represents gameplay-affecting items (oil spills, jams, boosts).
- SoundManager: Loads and plays sound effects.
- UIManager: Handles user interface elements, split-screen logic, and score display.
- CameraController: Handles the camera that follows the player. In other words what the player sees on the screen.

This flow captures the interactions and dynamic responses of a vehicle's movement in a top-down driving game.

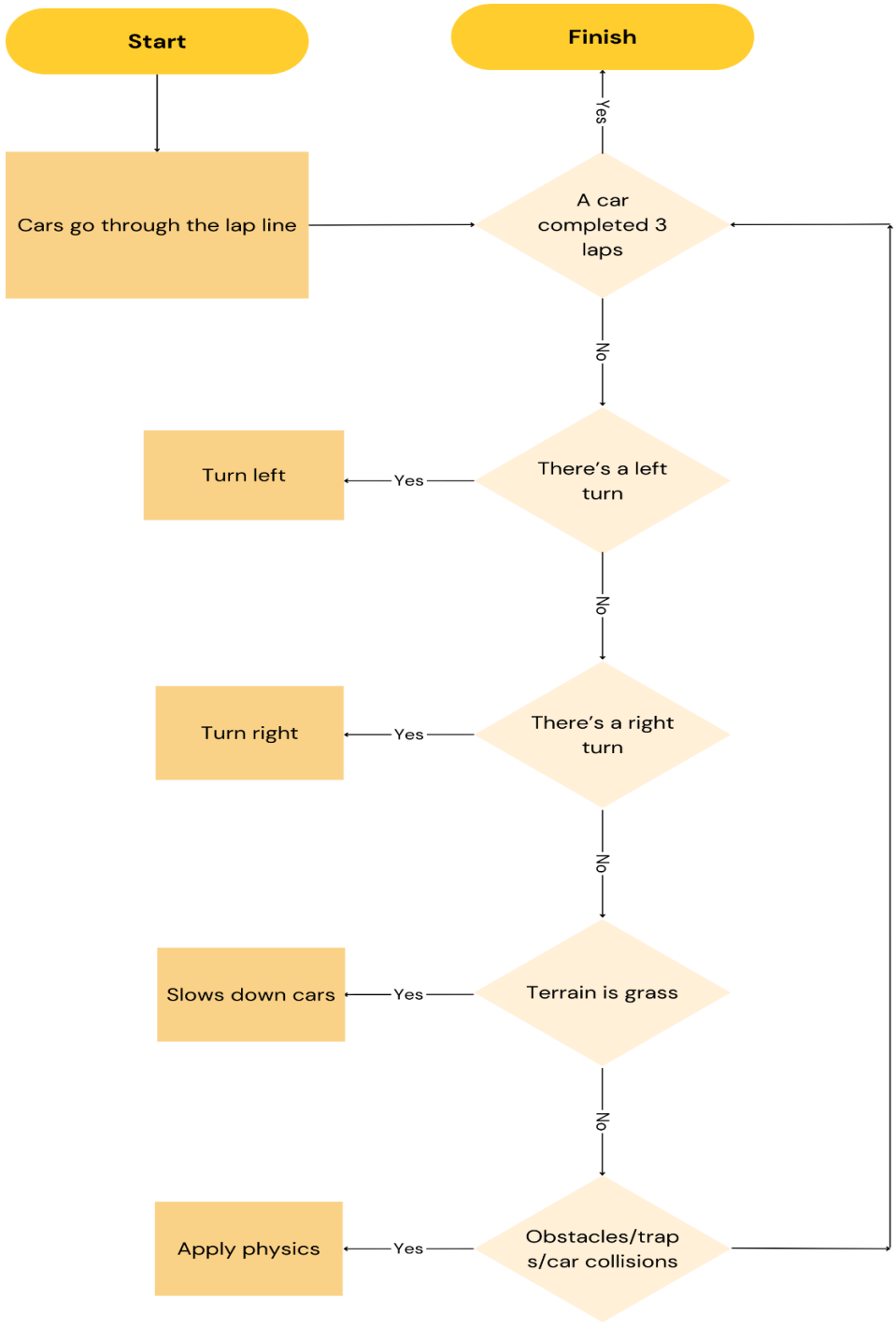

Figure 1: Flowchart of Vehicle Movement Decision Process .

3. External Libraries

- **SFML (Simple and Fast Multimedia Library)**: For rendering graphics, playing audio, and handling inputs. It's suitable for 2D games.
    - https://www.sfml-dev.org/
- **Box2D**: For realistic physics calculations, specifically for vehicle dynamics and collisions.
    - https://box2d.org/
- **IMGUI-SFML**: For simple, efficient GUI elements in menus and in-game HUD.
    - https://github.com/SFML/imgui-sfml

4. Sprint Plans

Sprint 1: Basic Project Planning

- **Duration**: 18.10 - 1.11
- **Goals**:
    - Completing the project plan document
    - Setting up development environment
    - Setting project schedule
    - Define main class structure
- **Progress Tracking**:
    - Weekly review meeting (Wednesday)

Sprint 2: Initial Implementation

- Duration: 1.11. - 15.11
- **Goals**:
    - Set up a basic game loop and initial UI with the main menu.
    - Implement player controls and car physics (basic driving and collisions).
    - Integrate SFML and Box2D libraries, focusing on core movement and collision functionality.
    - Design and load the track files.
    - Implement different terrain types and integrate handling physics for each.
    - Begin incorporating game objects like oil spills and boosts.
- **Progress Tracking**:
    - Weekly review meeting (Wednesday)
    - Update document according to progress and changes

Sprint 3: Enhanced Gameplay Features

- Duration: 15.11. - 29.11
- **Goals**:
    - Working enemy AI that races against the player
    - Enhance vehicle movements and controls
    - Introduce sound effects for cars, obstacles, and ambient noise.
    - UI development
    - Implement the additional vehicle types (boats, helicopters).
    - Ability system
- **Progress Tracking**:
    - Weekly review meeting (Wednesday)
    - Update document according to progress and changes

Sprint 4: Polish and Finalize

- Duration: 29.11. - 13.12
- **Goals**:
    - Bug fixing and testing
    - Add finishing touches, such as ability systems and any additional features agreed upon.
    - Finalize UI and gameplay flow, ensuring all elements are cohesive and debugged.
    - Finalize documentation
- **Progress Tracking**:
    - Weekly review meeting (Wednesday)
    - Sprint-end meeting to review, receive feedback, and finalize project.