

# Vehicle Simulation Game - Design Document

## Game Overview

This game is a vehicle simulation where the player controls a car using keyboard inputs. The game includes multiple camera perspectives, allowing the player to switch between different viewpoints. The main objective is to navigate the environment while managing movement mechanics.

## 1. Object-Oriented Programming (OOP) Implementation

### A. Inheritance (Parent-Child Classes)

- **Base Class: MonoBehaviour** (Handles Unity behavior)
  - **PlayerController** → Controls vehicle movement
  - **CameraSwitcher** → Allows switching between different camera perspectives
  - **FollowPlayer** → Enables a camera to follow the player from a fixed offset

### B. Polymorphism (Method Overriding & Overloading)

- **Update()** method:
  - **PlayerController.FixedUpdate()** → Handles movement logic
  - **FollowPlayer.LateUpdate()** → Updates camera position relative to the player
  - **CameraSwitcher.Update()** → Detects input for camera switching
- **SwitchCameras()** method:
  - **CameraSwitcher.SwitchCameras()** → Toggles between two different cameras

### C. Encapsulation (Getters & Setters)

- **Private fields with controlled access:**
  - **private bool isCamera1Active;** → Manages active camera state
  - **private float horizontalInput;** → Captures player input for movement
  - **private float forwardInput;** → Captures player input for acceleration

### D. Abstraction (High-Level Methods)

- **SwitchCameras()** → Handles camera toggling logic
- **MovePlayer()** → Encapsulates movement mechanics within PlayerController
- **UpdateCameraPosition()** → Manages smooth following of the player in FollowPlayer

## 2. Game Mechanics

- **Player Controls:**
  - WASD or arrow keys to control the vehicle
  - C key to switch between different camera perspectives

- **Vehicle Movement:**
  - Accelerate, decelerate, and turn
- **Camera Views:**
  - Follow camera and alternate viewpoint controlled by the player

### 3. GitHub Repository & Version Control Strategy

- **Main Branch** → Stable version
- **Feature Branches:**
  - feature-movement → Implements player movement mechanics
  - feature-camera → Implements camera switching and following logic
- **Commits:** Each feature will have multiple commits with clear commit messages

### 4. Tools & Technologies

- **Game Engine:** Unity
  - **Programming Language:** C#
  - **Version Control:** Git & GitHub
  - **Project Management:** Trello (for tracking tasks)
-