Game Design Document: Delivery Driver Challenge

Author: Gemini
Date: September 3, 2025
Version: 1.0

1. Executive Summary

1.1 Game Concept

"Delivery Driver Challenge" is a time-attack driving game. The player takes on the role of a delivery driver navigating a bustling, low-poly city. The objective is to pick up packages from designated locations and deliver them to their drop-off points as quickly as possible, all while a countdown timer adds to the pressure.

1.2 Genre

3D Arcade Driving, Time-Attack Simulation

1.3 Target Audience

This game is for beginner to intermediate Unity developers looking for a portfolio-worthy project. It's also for casual players who enjoy simple, satisfying, and fast-paced driving challenges.

1.4 Unique Selling Points (USPs)

- **Fast-Paced, Addictive Core Loop:** Easy to learn, difficult to master. The simple "pick up and deliver" mechanic against a ticking clock is compelling.
- **Vibrant, Low-Poly Art Style:** A clean, stylish, and highly performant aesthetic that is visually appealing without requiring high-end hardware.
- **Satisfying Driving Physics:** Simple, arcade-style controls that feel responsive and fun. Drifting and sharp turns are achievable and feel good.
- **Clear Navigational Aids:** A robust UI, including a minimap and on-screen objective markers, makes navigation intuitive and keeps the player focused on the action.

2. Gameplay

2.1 Core Gameplay Loop

The central loop is designed to be highly repeatable and engaging:

1. **Start:** The player begins with a full timer and is immediately given a "Pickup" location.
2. **Navigate & Pick Up:** The player drives to the pickup point, indicated by a visual marker and the minimap. Driving through the marker collects the "package."
3. **Navigate & Drop Off:** Upon pickup, a "Drop-off" location is immediately assigned. The

player races to this new destination.

4. **Score & Repeat:** Successfully delivering the package adds time back to the clock and awards points. A new "Pickup" location is instantly generated, and the loop repeats.
5. **Game Over:** The game ends when the timer runs out. The final score is displayed.

2.2 Player Controls

The controls will be simple and intuitive for both keyboard and gamepad.

- **Movement:**
  - **Accelerate:** W / Up Arrow / Gamepad Right Trigger
  - **Brake / Reverse:** S / Down Arrow / Gamepad Left Trigger
  - **Steer Left:** A / Left Arrow / Gamepad Left Stick
  - **Steer Right:** D / Right Arrow / Gamepad Left Stick
- **Camera:** The camera will automatically follow the car from a third-person perspective.

2.3 Game Objectives

- **Primary Objective:** Score as many points as possible by completing deliveries before the timer expires.
- **Secondary Objective:** Beat your own high score.

2.4 Mission Structure

- **Packages:** Represented by simple, glowing objects (e.g., cubes or icons).
- **Locations:** Pickup and drop-off points will be randomly selected from a pre-defined list of locations around the city to ensure variety in each playthrough.
- **Interaction:** The player simply drives through the designated zone/object to trigger a pickup or drop-off. No button press is required, keeping the pace fast.

2.5 Scoring & Timer

- **Initial Time:** The game starts with 60 seconds.
- **Time Bonus:** Each successful delivery adds 15 seconds to the timer.
- **Points:** Each successful delivery awards 100 points.
- **Game Over:** When the timer reaches 0.

3. User Interface (UI) & HUD

A clean, non-intrusive UI is crucial.

3.1 Main HUD (Heads-Up Display)

- **Timer:** Displayed prominently at the top-center of the screen (e.g., "TIME: 45").
- **Score:** Displayed in a corner (e.g., top-right, "SCORE: 1200").
- **Objective Indicator:** An on-screen arrow or marker that is always visible and points towards the current objective (pickup or drop-off), helping the player navigate.
- **Objective Text:** A simple text prompt at the top of the screen (e.g., "Pick up the package!" or "Deliver to the destination!").

3.2 Minimap

- **Position:** Bottom-left corner of the screen.
- **Functionality:** A top-down view of a small area around the player.
- **Icons:** The player's car, the current objective location, and major roads will be visible.

3.3 Menus

- **Start Screen:** Shows the game title and a "Play" button.
- **Game Over Screen:** Appears when the timer runs out. It displays "Game Over," the player's final score, and a "Restart" button.

4. Art & Aesthetics

4.1 Visual Style

The game will use a consistent low-poly aesthetic. Colors will be bright and vibrant, with simple materials and lighting to create a clean, stylized world.

4.2 Environment

The "Low Poly Simple City" asset will provide the cityscape, including roads, buildings, and props. The environment should feel dense enough to be interesting but open enough for fun driving.

4.3 Sound Design

- **Car Engine:** A simple, looping engine sound that changes pitch based on acceleration.
- **UI Sounds:** Distinct sounds for successful pickup, successful drop-off, and button clicks.
- **Background Music (BGM)::** An upbeat, energetic, and royalty-free electronic or synth-wave track to drive the action.

5. Technical Details

- **Game Engine:** Unity 6000.2.2f1
- **Render Pipeline:** Universal Render Pipeline (URP)
- **Platform:** PC (Windows/Mac)
- **Input System:** Unity's new Input System package.

6. Project Scope & 2-Day Plan

This GDD is designed to be completed in two focused development sessions.

- **Day 1: Core Mechanics & World Setup**
  1. **Project Setup:** Create the Unity project, import all necessary assets (city, car, input system).
  2. **Scene Assembly:** Build the game level by placing the city and car.
  3. **Player Controller:** Write the CarController script and set up input actions for driving.
  4. **Camera Controller:** Create the third-person follow camera.
  5. **Goal:** By the end of Day 1, have a car that is fully drivable within the city environment.

- **Day 2: Gameplay Loop & UI Polish**
  1. **Gameplay Manager:** Create a script to manage the game state (timer, score, objectives).
  2. **Pickup/Drop-off System:** Implement the logic for spawning objectives and detecting completion.
  3. **UI Implementation:** Build the HUD (Timer, Score), objective markers, and the Start/End game screens.
  4. **Minimap:** Implement the minimap system.
  5. **Audio:** Add background music and sound effects.
  6. **Goal:** By the end of Day 2, the project will be a complete and playable game loop, matching the full vision of this GDD.