Fruit Catcher

Overview

"Fruit Catcher" is a simple 2D game developed in Unity where the player controls a basket to catch falling fruits. The game challenges the player to catch as many fruits as possible while avoiding letting any fruit hit the ground. The game includes basic game mechanics, user interface, multiple sound effects, and a scoring system.

Game Design

Objective:

The primary goal of the player is to catch the falling fruits using the basket. The game ends when a fruit hits the ground, and the player's score is displayed.

Game Mechanics:

- **Basket Movement:** The player can control the basket's movement horizontally using the arrow keys or `A/D` keys. The basket can only move within the range of -9 to 9 on the x-axis.
- **Fruits Falling:** Fruits fall from the top of the screen at random intervals. There are multiple types of fruits, chosen randomly from a list of prefabs.
- **Scoring System:** The player earns 1 point for each fruit caught in the basket. The score is displayed on the screen and increases when the fruit is successfully caught.

Game States:

- **Start Screen:** The game starts with a welcome screen that has a "Play" button. Background music plays continuously from this screen.
- **Main Screen:** If a fruit touches the ground, the game ends, a sound effect is played, and the player is taken to the Start Screen after a brief delay.

User Interface:

- **Start Screen:** A simple screen with a "Play" button that transitions into the gameplay when clicked.
- **In-Game UI:** A score display is shown at the top of the screen, updating in real-time as the player catches fruits.

Sound Management:

The game uses a `SoundManager` to handle all sound effects:

1. **Background Music:** Loops continuously throughout the game, starting from the start screen.
2. **Collect Sound:** Played when a fruit is caught by the basket.
3. **Game Over Sound:** Played when the game ends after a fruit hits the ground.

Key Features:

- **Randomized Fruit Prefabs:** Multiple fruit prefabs are spawned at random, keeping the gameplay fresh.
- **Basket Control Limits:** The player can only move the basket within a defined boundary to prevent going off-screen.
- **Sound Manager:** All sounds are managed through a central script to simplify control and organization.

Challenges Faced:

- **Collision Detection:** Initially, multiple triggers on the same fruit collision were causing the score to increment more than once. This was solved by adding a `isCaught` flag to ensure the fruit is only counted once.
- **Sound Handling:** Playing the sounds directly in the fruit script became cumbersome, so a `SoundManager` was implemented to streamline audio management and reduce redundancy.

Conclusion:

The "Fruit Catcher" game meets all the requirements of the assessment, including a functional UI, random spawning of multiple fruit prefabs, sound effects, and a working scoring system. The project demonstrates core skills in Unity, game mechanics, and user interface design.