Reflective Development Journal

Brona Keevers-Roux

20098883

Into the Orchards

Mobile Game Dev

Phone accessibility

As I don't have access to an android mobile phone I have to use an emulator which isn't very good when it comes to analysing performance as it's effectively running on my pc rather than a mobile phone.

Possibly may be able to borrow a phone to test on a real mobile for the next CA, I was unable to for this one as I have been unwell and contagious for the past 2 weeks.

New systems

Recipes and their ingredients are generated into their respective scroll views on game load using prefabs and scriptable objects.

Scriptable objects used to track tree upgrades and info.

Telemetry logs added.

All game data saves and loads properly.

Fruit is added to the players available resources when they click on the trees.

Fruit can be turned into smoothies to earn coins which can then be used to upgrade orchards. Orchard upgrades can increase the quantity of fruit earned per click.

Performance

Updates are event driven, only triggering when the player clicks on something. There is not much CPU usage from UI so my UI based game runs very smoothly with evidence supporting this in the performance baseline report.

The game averages around 27 batches which very low and indicates excellent draw-call efficiency for mobile targets.

Changes in scope or design

The game idea hasn’t necessarily changed but more so has been expanded upon and gained more depth. The initial design was more vague just mentioning clicking trees to collect fruits, buying upgrades and turning fruits into smoothies. Now the design has been fully fleshed out with precise planning for exactly what I’m going to do and how I’m going to do it.

The UI has been designed with phone screen safe areas in mind, avoiding the camera area. The size of all the elements have been carefully decided with constant referencing to things on my own phone in order to get the scale correct.

Description

Telemetry was introduced for key gameplay events such as recipe selection, smoothie crafting, save, load and upgrade changes. All telemetry is handled through Debug.Log statements.

Performance analysis was conducted using the Unity Profiler connecter to Android studio. Although a real phone would provide more accurate data the emulator still offered enough insight to estimate the performance baseline stats. Logcat was also evaluated to get the average framerates of the CPU, GPU and overall.

Evaluation

Event-driven UI updates were very successful. Since almost all UI changes happen only when the player interacts or when a smoothie finishes crafting, the game avoids the typical heavy Canvas rebuilds that can cause UI performance issues on mobile. As a result, the profiler showed the game running smoothly, with minimal CPU overhead.

The decision to use ScriptableObjects for upgrades, recipes, and ingredients makes the game very easy to expand upon and I will be able to quickly add more trees and recipes to the game in future.

Problem and solution

When I first implemented the recipe toggle for the smoothie machine it wasn’t matching up with the saved recipe from a previous play session. This was due to the way that the deserialization of the saved recipe was being handled. I solved this by just comparing the recipe name rather than the whole recipe scriptable object.

CA3 Goals

Based on the current level of development there is definitely a lot left to do to create a complete and well-balanced game with engaging mechanics. A lot of more fruits need to be added to be able to create more advanced smoothie recipes with each new fruit addition enabling a large number of new recipes to be created.

It is clear from game testing that the blenders will need a big upgrade system of their own to keep up with the fruit income of the player, otherwise the player will be left a large excess of unusable fruits.

- Add more types of fruit
  Oranges, Pears, Lemons, Peaches

- Create more art
  Art for all the new fruit types, Background art, UI art

- Add more upgrades
  Fruit quality has not yet been implemented even though it already exists as an option

- Add automation
  Harvester and harvester speed upgrades to automatically collect fruit for the player

- Add offline resource generation
  Calculate offline resource generation based on amount of time away from the game with a time cap

- Add blender upgrades
  Increase blender speed, increase blender capacity so you can create multiple batches at once. Increase number of blenders available with left and right navigation similar to the trees. Different smoothies can be set to be produced at each blender.

- Add music and SFX
  Menu music, game music, SFX when clicking trees, buttons, upgrades and SFX for when a smoothie finishes crafting.

- Add prestige system
  Extend offline generation time cap, Make upgrades cheaper