**Project** **Rainbow Mountain** **Economy Report**

By Joseph Sprucesef 🎄

This is an economy report for the café game prototype, **Project** **Rainbow Mountain**. This document will showcase:

* Description & Economy Loop.
* Parts of the economy present in the prototype.
* Parts of the economy not included in the prototype, and why.
* Explanation of rules and balancing of economy.
* Testing results and feedback of iteration 1 of the prototype.
* Response to feedback + changes made for iteration 2 of the prototype.
* Final version of prototype.
* Stretch goals.

# Description & Economy Loop:

**Project Rainbow Mountain** features an **economy** that places an emphasis on **constant action** in a **fast-paced hospitality environment**. The game takes place in a popular **café**, with the player acting as the barista making coffee for droves of caffeine addicts.

The **diagram** below page shows the **economic loop** for the **1st iteration** of the project. It places an emphasis on the **ingredients** required to make a **coffee**, and the **money** received for successfully completing an order. This is combined with the **time stress** that comes with **completing orders** **in time** to **pay bills** each **day** and keep the café operational.

The **first iteration** of the **economy prototype** aims to **test** and **balance** these aspects of gameplay to produce a **fast-paced, satisfying experience** for the player.

A diagram of a recipe

Description automatically generatedOriginally it was intended to showcase the time-pressure mechanics and monetary values of buying and selling cups of coffee. Upon receiving **feedback**, the focus for what to showcase in the prototype was pivoted to the **time management** portion of the **economy.**

# Prototype Iteration 1

**Project Rainbow Mountain** is heavily dependent on the **player’s keyboard input** and their **speed of execution.** Considering this, it was decided that developing the prototype in **Unity** would be best, as it would be difficult to **automate** in **Microsoft Excel** or **Machinations** while still delivering the **intended experience**.

While developing the **prototype** in **Unity** using **C# programming**, it was suggested by a teacher to try focusing on one major aspect of the economy at first; either **managing cash** and **purchasing ingredients** or **balancing time** that the player has to **complete orders**.

For **Prototype Iteration 1** it was decided to focus on the **time management** aspect of the **economy**. The gameplay of entering the right combination of inputs to create coffee for customers will remain the same.

Adding in an **ingredient inventory** and **earning money for orders** will be a **stretch goal** for continuing this project.

## Controls

The **player** can press the above buttons on the keyboard in the specified sequence to create a cup of coffee that matches any one of the currently available orders.

Steps to making a coffee:

Step 1: Select a cup size: By pressing Q for Small, W for Medium, and E for Large, the player can select a cup size.

Step 2: Select a coffee type: By pressing J for Cappuccino, W for Latte, or L for Mocha, the player can select a type of coffee to make.

Note: Selecting a cup size or coffee type locks that choice in. The only way to go back is to press Enter, which discards the order if the cup you’re making doesn’t match an order.

Step 3: Add Coffee & Sugar: Add up to 3 shots of coffee and/or teaspoons of sugar.

Step 4: Complete Order: Press Enter to send off a cup of coffee. If it matches a currently available order, it’ll complete that order. If the coffee doesn’t match an order, it will be discarded at no penalty. Either way, you can start making another cup of coffee immediately.

## Economy Values

* Order Recipes: There are currently 4 preset recipes, shown in the next section. They each vary in Size, Type, Sugar and Coffee shot amount.
* Active Order List: Every 15 seconds, a preset recipe is randomly chosen and added to the Active Order list, up to a maximum of 4. This is displayed in the inspector, with UI elements planned for Iteration 2.
* Time Between Orders: This set values determines the time between receiving new orders. **Currently set to 15 seconds.**
* Completed Orders: Upon completing an order, the player’s “Completed Orders” counter will go up.
* Missed Orders: If a player’s active order list is full and the time between orders runs out, the player will miss an order, increasing the counter.
* Orders Per Day: Currently set to 20 orders received before no more orders appear.

## Preset Recipes:

A screenshot of a computer

Description automatically generated

## Testing Results & Feedback

Tester 1:

* Key inputs were difficult to remember at first, but tester 1 showed improvement with each new order completed.
* Was not sure if they could discard coffee after messing up, but after failing several times, they realised that there was no downside to making an invalid cup of coffee. Learnt to discard an order if they messed up, and immediately tried again, which was the intention.
* Number of preset recipes were too small, as tester 1 began to do the same orders and was memorising them quickly.
* At first 15 seconds between each order was good, for learning the basics, but it felt too generous by the end as tester 1 improved in skill.
* Getting feedback in the debug log about what was input was good, but would like to see this in UI.
* 20 orders for 1 day felt like a good amount for a first day.
* Difficulty can be improved by reducing time between orders, and adding more complexity/more steps to orders.
* Ensure that the time between orders isn’t reduced too low, as this can make getting every order impossible.
* Player would like to see on-screen clickable buttons, that also act as on-screen controls for keyboard input.

End of Playtest 1 Score:

A screenshot of a computer

Description automatically generated

Tester 2:

* Tester 2 picked up control scheme quicker than Tester 1, and thought that timer was too generous.
* Tester 2 was completing a single order quickly, and then having to wait ~10 seconds between completing an order and waiting for new ones.
* Tester 2 thought that having multiple orders coming in at once, at an increased frequency would make it more engaging.
* Proposed that rather than having a set number of orders each day, have a set amount of time for each day (5 minutes for example), and have the difficulty scale with the player, increasing in number of orders if player is quicker.
* Some variance in how often orders come in can make it more realistic.
* With only 4 recipes, tester memorised them easily, and was randomly getting the same order over and over.
* Would like to see info on main screen, rather than having info displayed in Inspector and Debug Log.
* Would like to see clickable buttons that show controls also.
* Controls felt good, and were satisfying to memorise.
* In general had fun with it, but felt very relaxed. Would be more engaged if it were more difficult.

A screenshot of a computer

Description automatically generatedEnd of Playtest 2 Score:

## Testing Round 1 Summary:

* Gameplay was fun, but too easy. Requires balancing on active order frequency and amount to make it more engaging.
* Players were able to understand and engage with the game quickly.
* More preset orders should be made, as 4 preset orders are too few, allowing players to see the same ones repeatedly and memorise them.
* Information shown on Debug Log and Inspector should be shown in-game through UI. Specifically, feedback on what has been input by the player, and having active orders displayed.
* Controls should be added on-screen, through buttons that can also be clicked on for mouse accessibility. Having audio feedback for each button press and successful button press can be good as well.
* Communicate to the player that they can discard orders if they mess up by pressing Enter at no downside.

# Prototype Iteration 2

## Focuses for Iteration 2:

* Displaying all information required for the player on-screen, and in build. (Previously in iteration 1, controls were told verbally, and info such as orders were in the Unity Inspector).
* Tweaking economy values to better deliver the intended experience of the project (In particular, time between orders, number of recipes).
* Adding clickable buttons on-screen so the game can be played with mouse or keyboard.
* Adding more recipes to reduce the chance that the player gets the same order multiple times in a row, providing more interesting gameplay.

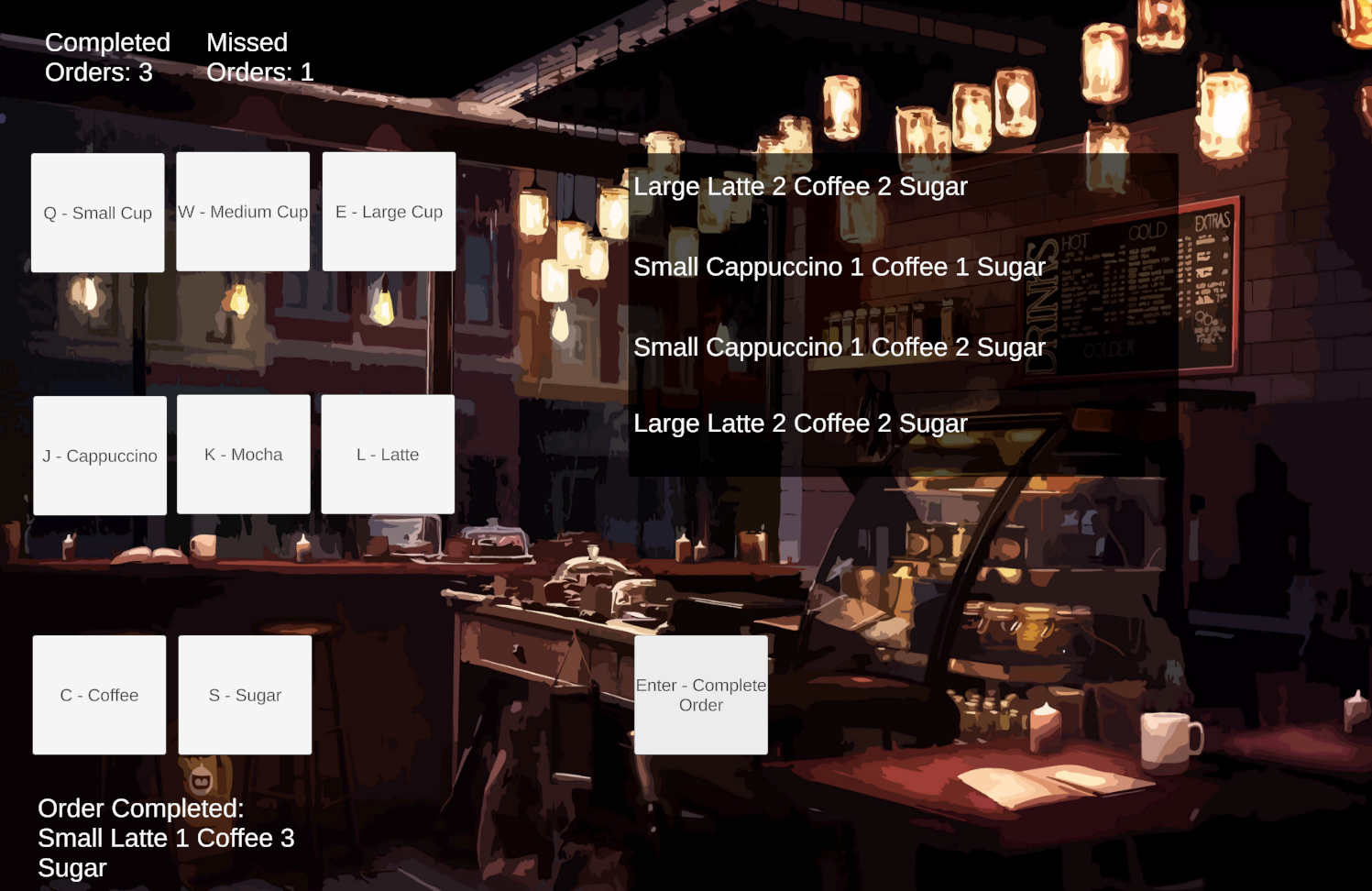
## Changes Made to Economy Values:

* Order Recipes: Increased from 4 preset recipes to 27 to add more variety. For each coffee type, there are 3 small, 3 medium, and 3 large recipes.
* Active Order List: Maximum of 4 orders at a time. A new order activates every 5 seconds, previously 15 seconds. This provides enough time to complete an order in one go, and put pressure on the player if they fail inputs.
* Time Between Orders: This set values determines the time between receiving new orders. **Changed to 5 seconds.**
* Completed Orders: Upon completing an order, the player’s “Completed Orders” counter will go up (Unchanged).
* Missed Orders: If a player’s active order list is full and the time between orders runs out, the player will miss an order, increasing the counter (Unchanged).
* Orders Per Day: Currently set to 20 orders received before no more orders appear (Unchanged).

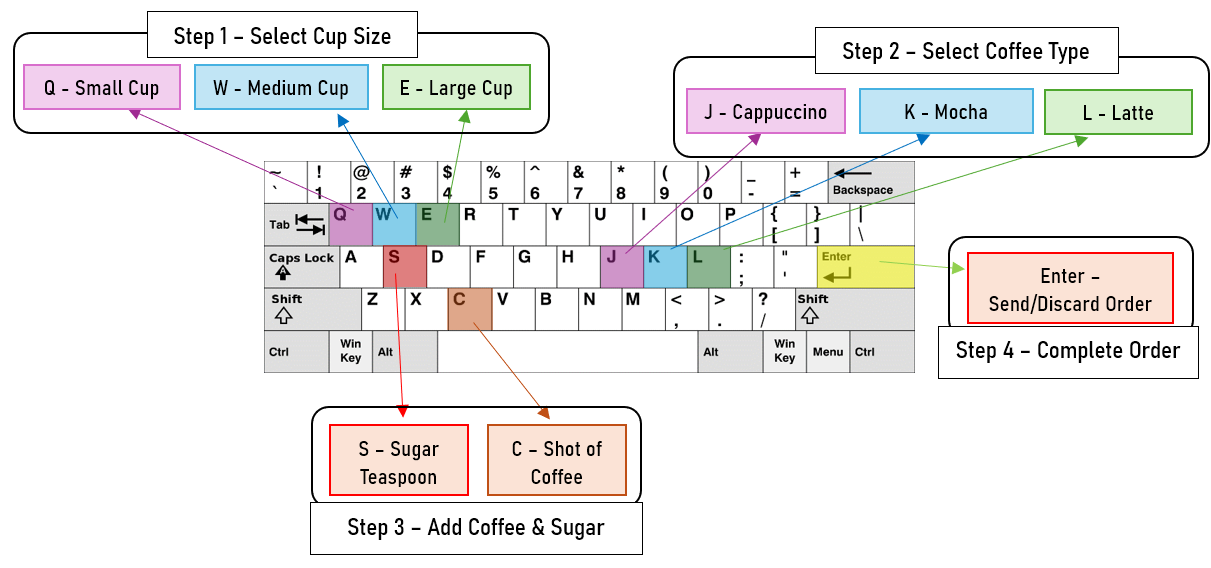
## Iteration 2 Additions

* 23 more recipes, bringing the total recipe count up to 27.
* Implementation of UI with clickable buttons that showcase keyboard controls.
* UI now displays active orders.
* UI now displays last completed order.
* Background image to set the tone of the game.

## New in-game UI in Iteration 2



## Updated Controls for Iteration 2



## Changes Made to Controls:

Swapped binds for Mocha and Latte. Mocha is now K, Latte is now L. This change was made to make selecting coffee type by keyboard slightly more intuitive, as L makes more sense to be set to Latte.

## Overview of Iteration 2

The above changes have considerably brought this project closer to its intended vision. The reduction of time between orders to 5 seconds and the larger variety of recipes in particular bring a sense of urgency to quickly check an order and attempt to input its components correctly.

Showing more information to the player through UI and having a background image helps to convey a clear image to the player on the controls and tone of the game.

Overall, this project was a success, with the objective of delivering an execution-focused, fast-paced café game prototype being achieved.