CS 3110 MS3 Report

Joanna Andrews
Mishita Kaja
Rishika Cherivirala
Jessica Andrews
Jeana Han

Vision

We have a clear vision of what we aim to achieve during our project. Our primary objective is to create a comprehensive garden simulator that will offer an unmatched user experience. Initially, our focus was solely on building a garden and tending to the flowers using features such as feeding, watering, and neglecting the plant, but we have now expanded our vision to include a complete ecosystem with flowers and fruits. Our new plan involves integrating a marketplace and a bakery into the simulator, which will allow users to engage in a more immersive and fulfilling experience. Through the marketplace, users will be able to sell plants and baked goods created using recipes, and, in turn, gain coins that they can use to further care for their plants. This feature will not only add to the game's excitement but also enable players to hone their entrepreneurial skills. With these enhancements, our garden simulator will offer a more engaging and interactive experience that will keep users enthralled for hours.

Summary of progress

Between the milestones MS2 and MS3, we were productive in planning, beginning, and creating the implementation of our project: Garden Gameplay (GG). Now that we had a working garden project with functionality implemented for feeding and viewing plants, we decided to build on the existing features of the garden. First, we introduced new functions in our plant.ml to allow users to perform more actions on the plants. In the plant.ml file, we implemented functions to water, neglect, and modify the price attribute of the plants. We also implemented new functions in garden.ml and plant.ml to apply good and bad events, such as a drought or a surprise visit by benevolent fairies, to the garden. We designed these events to be applied in a random way, so that the user would experience these events at random. These events had effects such as incrementing the hydration level of all plants or killing a random quantity of plants. This introduced an element of nuance and unpredictability to the game. To allow the user to react to the random tragedies, we implemented new attributes for our garden type to represent the level of “luck” and the level of “defense” associated with the garden. When buying a Cactus or Clover plant, the user’s garden would gain defense, and the user’s garden would gain luck by purchasing ladybugs. A higher luck score decreases the user’s chance of experiencing negative tragedies (i.e. drought), while a higher defense score decreases the amount of damage incurred by the user when a tragedy does occur. Additionally, we replaced our existing menu option of “Add a plant” with a store option (“Buy a plant/item”) from which users could purchase plants as well as other items like milk and eggs. We added a new store.ml and store.mli files, and we implemented functions in the store.ml and plant.ml files that would allow for random discounts to be applied to some items within the store each time the user went to purchase an item. We implemented the ability to buy plants from the store and have them added to the garden, as well as the ability to purchase other items and have them added to the inventory. Additionally, we introduced a new compilation unit for crafting recipes from the purchased items, including new recipe.mli and recipe.ml files. We implemented a new menu of recipes, which would display the recipes for the user to select from. We then implemented the functionality within the recipe file for the user to craft recipes and add the finished product to the inventory; if the user does not have enough ingredients to make the recipes, they will be notified, and the recipe will not be made. Additionally, we added a few key design choices to modify the flow of the game. We implemented the ability to allow the user to enter inputs again if they entered an invalid input the first time. We introduced the functionality for the user to return back to the main menu after selecting options in the menu. To make the game more realistic, we implemented the garden to iterate through days and allow the user to perform a certain amount of actions on each day. To create a means of ending the game, we introduced a money attribute that would decrease as the user performed actions. The game ends when the user runs out of money. To restructure the way information was presented to the user, we implemented functionality to allow the user to view/perform actions on certain categories of plants (i.e. Fruits) instead of applying these actions to the entire garden. Additionally, we modified the type of our plant from a variant type into a record type, which included the plant species as a record field. This allowed us to cut out unnecessary repeated code and make our code structure more efficient and readable. Lastly, we implemented test cases and specifications for our store, inventory, plant, garden, store, and recipe compilation units to ensure that our code was working as intended.

Along with the functionality of our code, we also continued exploring using git as a team of 5 members. Through trial and error, the team was able to familiarize themselves with the process of pushing and pulling with many people. We also learned, explored, and completed writing the YAML file and install file in preparation for MS3. Overall, each member of the team made successful contributions to progress the project to our milestone! This project gave our team the opportunity to learn about collaboration, designing and improving compilation units, evaluating and restructuring code to improve efficiency, and writing exhaustive test cases.

Activity Breakdown

- Jessica:
  - Task: Implement the interfaces of plant.ml and garden.ml; Brainstorm current and new features of the game, assess game logic, adjust gameplay as necessary, implement tragedies
  - Delivered: Delivered the files and implemented the interfaces.
  - Hours: 15 hours

- Joanna:
    - Task: Implemented some of the random events, worked on implementing the logic for the user gaining luck/defense. Implemented functionality for applying and displaying discounts in the store.;
    - Delivered: Delivered the files and implemented functions in plant.ml, store.ml, main.ml, and garden.ml
    - Hours: 27 hours
- Mishita:
    - Task: Implemented parts of main.ml, rewrote the type of plant.ml to refactor code; wrote some tests cases for plant.ml; cleaned up store.ml
    - Delivered: I delivered the files and implemented functions in plant.ml and store.ml
    - Hours: 25 hours
- Rishika:
    - Task: Implemented most of the test cases; implemented functionality of recipe.ml; implemented some functionality in inventory.ml and garden.ml
    - Delivered: Delivered the files and implemented the recipe.ml, test_CS3110FinalProject.ml, inventory.ml, garden.ml, main.ml
    - Hours: 25 hours
- Jeana:
    - Task: Implemented parts of main.ml; Created string representations of recipes in recipe.ml; Cleaned up user interface for aesthetic and easy accessibility purpose; Schedules meetings for the demo and check-ins
    - Delivered: Delivered the files and implemented recipe.ml and main.ml
    - Hours: 21 hours

Productivity Analysis

As a team, we were very productive as we were able to accomplish everything that we had planned in our sprints. Between MS2 and MS3, our team made significant progress in planning and implementing our Garden Gameplay project. We started by building on the existing functionality of our garden simulator, which allowed users to feed and view plants. We introduced new functions and plants in our plant.ml file, which enabled users to perform additional actions on the plants, such as watering, neglecting, and modifying the price attribute by conducting certain actions. We also implemented new functions in garden.ml and plant.ml, which applied good and bad events at random, such as a drought or a surprise visit by fairies, to the garden. These events had effects such as incrementing the height of all plants or killing a random quantity of plants, adding an element of nuance and unpredictability to the game. To allow users to react to these events, we implemented new attributes for our garden type to represent the level of “luck” and the level of “defense” associated with the garden. Users could gain luck by purchasing a Cactus or Clover plant, and defense by purchasing ladybugs. We also created an inventory and recipe function for our garden, so users can harvest plants and buy other materials to make recipes and earn money. One thing we were unable to accomplish is making the game end when all plants in your garden die, but we ran into logistical issues and decided to end the game when the user had no money left. Our estimates were accurate because we set deadlines and planned to accomplish specific tasks by a certain date. Overall, our team made significant progress in expanding the functionality of our garden simulator and creating a more immersive and engaging experience for users.