

Team 22 Cobblestone Chi – MS3 Report

Vision

Our vision is to create a mini-replica of Stardew Valley in OCaml called Cameldew Valley. We focused on specifically the crop-planting feature in Stardew Valley and we made it a bit more competitive by adding timer and coins with even a leaderboard system to compare against other players of your score. Between MS2 and MS3 our focus shifted from getting the core functionality finished to finishing up the game and polishing up whatever that needed to be improved. Focusing on the user experience, rather than the logic, helped us improve our game so that it is more accessible and clear to those who were just playing our game for fun.

Summary of progress

Between MS2 and MS3, our team continued to build on the MVC model we created. There were significant improvements in the user experience. First of all, to dive deeper into the visuals, we were able to make more colorful crops and diverse crops instead of the single strawberry crop we had in MS2. We added four separate beautiful screens, consisting of the start screen, pause screen, shop screen, and the leaderboard screen. We kept the functionality of just using keyboard shortcuts to navigate through the screens and interactions, keeping our game minimalistic and friendly. We also added visuals for coins and timer, updating accordingly to the crops collected and time passed. In terms of other quality of life features, we made it so that there is a barrier to some tiles that are unreachable such as the shop where we created a mini barrier around it and the tree where we also created a mini barrier on it. In terms of crop planting, we had much difficulty in aligning the player's movements and player's facing direction with the exact tile they were looking at. Initially, the crop planting was not user friendly, we would plant the crops where the player wasn't looking at directly and the radius of crop planting was too small. However, we were able to significantly make this process smoother through recognizing exactly the position they lie on the screen.

Integrating the database was also a significant milestone of our project. Although integrating sqlite3 made the installation instructions much more complex, having a database allowed us to have more purpose in our game, adding more competitiveness between our players. At the end of the game, the screen displays the top 10 players who collected the most amount of crops/coins. By adding this feature, we hope that our players continue to come back to play the game and strive to be placed in that top 10 list, through finding different strategies and hoping that their luck will allow them to do so.

Activity breakdown

Alex - Game Controller System & Crop Planting/Harvesting Functionality

- Added game controller functionality for planting and harvesting crops (buying from shop, deducting coins, adding seeds, rendering new crops)
- Modified game models to add helpers needed to plant/harvest crops including `board_iterate` and `get_nearest_soil_point`
- Added movement features, fixed crashing bugs and code quality issues
- Recorded the demo video, formatted/prepared repository for submission
- Hours spent: 13

Jiwon - Models & Game Controller Logic & Leaderboard & Database

- Added a leaderboard model to track name, coin, and timestamp
- Made crop planting more smooth and more user friendly
- Added the database `leaderboard.db` with `sqlite3` so that the database is within the program itself
- Worked on timer display and timer control (2 minutes)
- Hours spent: 13

Donathan - Main Loop, Rendering, and Sprite Creation

- Lead the majority of visual design and game polishing
- Built out the coin/clock displays, initial window, background and asset rendering
- Added constraint to player movement to polish the game (not allow player out of bounds)
- Added pause, title, and end screen as well as fixed multiple rendering bugs
- Worked on add more
- Hours spent: 13.5

Jimmy - Game State & System Coordination + Testing

- Focused on state/flow of the game, adding a pause system and connecting it across the game state
- Fixed the crop growth timing bug during pause/unpausing
- Created and iteratively refined the test suite for all models, game state and controller, reaching upward of 80% coverage
- Added and rewrote complex tests as game models and functionality changed, allowing the team to quickly develop new features
- Finished documentation for mli files.
- Hours spent: 13.75

Productivity analysis

As a team, our productivity was consistent and got our project to an end state we were satisfied with. We utilized ideas from Agile methodology, like tackling big segments of functionality in sprints. Many new features were implemented like start screens, leaderboards, pause screens, and a shop screen. Many of these features required intense bug fixing and collaboration to implement successfully, but in the end our productivity was enough to get all these features completed to our own goals satisfaction. Sometimes our productivity was held back by certain features dependent on others still in production, but this entire process helped us understand the importance of teamwork in an expansive project like ours. Overall, we succeeded in accomplishing the major goals of this sprint, and we think our productivity was key in this process.