Team 22 Cobbestone Chi – MS2 Report

Vision

Our current vision for the system we are building is a time-limited farming game in which players must plant, grow, harvest, and purchase crops in order to earn as many “Caml-coins” (in-game currency) as possible before the timer expires. Within this constrained interval, players must make strategic decisions about which crops to invest in, how to manage their inventory, and when to sell their harvests to maximize profit. Since MS1, the most significant addition to our design has been the introduction of a fixed gameplay timer. We realized that without an end condition, the game lacked a clear objective and felt more like an open-ended simulator. Adding a time limit transformed the experience into a challenge, giving players purpose, encouraging optimization, and aligning the game with the resource-management style we wanted.

Summary of progress

Between MS1 and MS2, our team made substantial progress toward building the core gameplay loop of our farming simulator. We expanded the project structure into a clear MVC-style organization, separating logic across model, controller, and view components. Within the model, we implemented foundational modules for crops, the player, the board, and the overall game logic. These modules now support basic game entities such as crop objects with growth properties, a player that can move across the board, and tiles that store crop or soil information. We also introduced the initial game timer concept through the Game State model, which tracks whether the game is actively being played.

On the interaction and rendering side, we added the beginnings of real-time gameplay. Players can now move around the map using keyboard input handled through the new input reader and event system. Pre-seeded crop tiles have been added to the board, and these crops visibly progress through growth stages as time passes. The renderer displays this growth on the screen, enabling us to watch crops mature in real time. Each team member contributed to different aspects of this pipeline: Alex worked on input handling and Raylib event readers, Donathan focused on the main loop, renderer updates, and sprites, Jiwon built model logic and integrated crop growth into the game controller, and Jimmy implemented the game state transitions that determine when the game is active. Together, these updates transformed our project from a static prototype into a working, interactive farming simulator.

Activity breakdown

Alex - Input Handling & Game Controller System

- Implemented the input reader and integrated Raylib event handlers for player movement and interaction.
- Worked on processing keyboard inputs (WASD + action key) and connecting them to the controller layer.
- Participated in discussions on interaction design and tile-based action logic.
- Contributed to debugging early UI responsiveness issues.
- Implemented crop growth logic within the game controller, linking time progression to crop state transitions.
- Hours spent: 7 hours

Jiwon - Models & Game Controller Logic

- Built core model components including crop, player, board, and crop-related functionality.
- Participated in debugging interactions between the model, controller, and rendering layers.
- Managed folder structure and configuration files, including initial crop_data.json.
- Hours spent: 6 hours

Donathan - Main Loop, Rendering, and Sprite Creation

- Developed the project's main loop in main.ml, orchestrating updates to models and calling the renderer.
- Contributed to the renderer, enabling the display of the board, player sprite, and crop growth stages.
- Assisted with crop growth logic, ensuring crops visually progress over time.
- Created all the sprites that are going to be used for the game.
- Hours spent: 7 hours

Jimmy - Game State & System Coordination + Testing

- Implemented the Game State model, including logic for transitioning between playing and not playing states.
- Worked on initial structuring of game flow: when the game begins, updates over time, and conditions for stopping.
- Coordinated integration between controllers and models to ensure state changes trigger properly.
- Assisted with planning, documentation, and discussion of the new timed gameplay objective.
- Hours spent: 6 hours

Productivity analysis

Our productivity between MS1 and MS2 was solid and we learned a lot about estimating workload more realistically. We successfully implemented the core components we planned including the player movement, crop growth, the game state model, and a basic rendering pipeline, but several tasks took longer than anticipated, especially integrating the model, controller, and view layers. Some team members progressed faster in their individual components than in the collaborative integration work, which revealed that our original sprint estimates were a bit more optimistic than we anticipated. We realized that even features that seem small can require substantial coordination when multiple modules must interact smoothly. Overall, we accomplished the major goals of our sprint, but our timeline estimates were off by a meaningful margin, and this experience will help us plan more accurate sprints and allocate responsibilities more effectively for MS3.