CS 2340 — Milestone 3: Project Iteration 2

Market, Basic Farm Interactions, User Stories, and Sequence Diagrams

BACKGROUND: For this iteration of the project, your team will implement a local market for buying and selling, and some initial farm interactions. User stories are simple statements describing some of the game's typical use cases. Sequence diagrams take user stories and provide implementation-specific details on the process.

PURPOSE: This milestone expands upon the work started in M3. Planning is key when designing large systems, and diagrams are very helpful tools for making decisions about how to lay out your code.

TASK: This milestone has two team design deliverables: a testing deliverable and a required feature set for implementation. Other than the requirements outlined below, the details of your implementation are up to you. Your app implementation/functionality will be graded during a demo which will occur the week after this milestone is due.

Design Deliverable 1: User Stories

1. As a team, create 5 unique user stories for your project based on the implementation requirements.
   - You should follow the following format:
     - "As a *[user role]*, I want to *[goal]* so that *[benefit]*."
     - Example: "As an administrator, I want to be able to add developers to projects so that each developer is able to self-assign themselves to issues."
   - Your user stories should focus on the goals of and benefits to the player.

Design Deliverable 2: System Sequence Diagrams

1. Each team member should take one of the user stories created for Design Deliverable 1 and turn it into a UML sequence diagram.
2. For each sequence diagram, ensure that it illustrates the dynamic interactions between the objects and your project's design.
   - For the GUI and other user input, you may represent it with an appropriate "class" name.
     - Examples: CharacterInterface, InventoryScreen

3. Each diagram must contain at least one ALT or LOOP structure, so choose your user stories carefully!
4. Combine all team members' sequence diagrams into a single pdf.

Implementation Requirements

1. Design and implement an **inventory**.
   - Inventory can be implemented either as a separate screen or as a panel on the farm screen.
     - If you choose the separate screen implementation, you must also include a way to return to the farm screen.
   - The inventory should list all the player's crops and seeds. At the start, this will just be the starter seeds that were selected on the initial configuration screen.
   - The Inventory should have a finite capacity.
2. Design and implement a **market**.
   - The market UI should include a listing of the player's inventory. The player should be able to sell items from their inventory.
     - The player should not be able to buy anything which will cause their inventory to exceed its capacity.
     - The player should only be able to buy items if they have enough money.
   - The sell prices for items should be determined by an algorithm that considers a few variables. The exact details of this pricing algorithm are up to you.
     - Example: base_price * (difficulty_multiplier) + random_variance = final_price
   - It's up to you to come up with items to be sold and bought at the market!
   - The market UI should also include a listing of seeds for sale. Purchasing seeds should update the player's inventory.
     - There should be at least three seed types for sale.
   - The buy prices for seeds should be determined by some reasonable algorithm.
     - The exact details of the pricing algorithm are up to you.
     - Different seeds should have different prices.

- The player's money should be displayed on the screen and should be updated after buying or selling.
- There must be a way for the player to return to their farm.

3. Harvesting crops.
   - For this milestone only, the farm plots should start out with a random assortment of plants at different stages of growth. This can be displayed graphically or textually. For a textual example:
     - Plot 1: Seed
     - Plot 2: Seed
     - Plot 3: Mature plant
     - Plot 4: Immature plant
     - Plot 5: Mature plant
   - The player should be able to move ("harvest") mature crops into their inventory. Harvesting a crop should clear the plot that it previously occupied.
   - Only mature plants should be harvestable.
   - There are no requirements for planting or watering for this milestone. The only functionality you are responsible for is moving harvestable crops from a farm plot into the inventory.

Testing Requirements

1. Write **unit tests** to verify the functionality the newly implemented features.
   - There is no code coverage requirement, but you should make sure that your unit tests cover meaningful functionality.
   - *You should have at least 5 unit tests in total.*
2. **Testing Deliverable**: Include with your submission a brief writeup describing your testing process for the milestone. Explain which components were chosen for testing and why. Additionally, explain how your tests verify that the code functions as expected.

Checkstyle

During demo your team will be required to run the checkstyle script (located under files>checkstyle>Java Guide.pdf). This script will give your project a score out of 10 and will

account for 10 points of your M3 final grade. Be sure to run the checkstyle script prior to submission to avoid unforeseen deductions.

Milestone Tagging

Tags are a way of marking a specific commit and are typically used to mark new versions of software. To do this, use "**git tag**" to list tags and "**git tag –a tag_name –m description**". **Important:** To push tags, use "**git push origin –tags".** Pushing changes normally will not push tags to GitHub. You will be required to pull this commit during demo.

Submission Requirements

In addition to your diagrams, ensure that you include a link to your GitHub repository in your submission. Also, ensure that you have added your grading TA(s) as collaborators so that they may view your private repository. **Repositories must be located on the Georgia Tech GitHub and must be set to private!** Points may be deducted if these guidelines are not followed!

CRITERIA: You will be graded according to the rubrics on the final pages of this assignment document. **Groups are required to demo in order to receive credit for the features they have implemented.**