CS 2340 — Milestone 6: Project Iteration 5

Farm Upgrades and SOLID/GRASP Principles

BACKGROUND: For the final iteration of the project, your team will implement some new farm upgrades. Additionally, you will review your code for the OOD principles you have been learning in class.

PURPOSE: Reviewing your implementation of the farming simulation can be a helpful exercise. This assignment allows you to reflect on your implementation choices and identify code smells that may have crept into your code. Identifying what went well and what could have gone better may help you avoid similar problems in the future.

TASK: This milestone has one team design deliverable, one testing deliverable, and a required feature set for implementation. Other than the requirements outlined below, the details of your implementation are up to you. **There is no in-person demo for this milestone.** Your app implementation/functionality will be graded via a video highlighting your work throughout all the major milestones.

Design Deliverable: SOLID and GRASP principles

For this milestone, you will need to show examples of some of the SOLID and GRASP principles in your code. Specifically, you will need to provide concrete examples in your team's code of 3 SOLID principles and 5 GRASP principles (either GRASP "basic" or "advanced").

To get credit for this portion of the assignment, you must take a screenshot of the code for each of your examples. Each principle must be demonstrated in a unique example. When you have compiled your examples of SOLID and GRASP principles from your code, add the pictures to a PDF document. For each picture in your document, label it with the principle it demonstrates and a brief description (3-5 sentences) of how the example fulfills the coding principle. Do not forget to submit a PDF of your OOD principle examples for this assignment.

Implementation Requirements

1. Farm expansion
   - Player should be able to **purchase additional plots**.
   - Price should increase based on current farm size.
   - Purchasing additional farm plots should update the farm UI. New farm plots should start as empty.
2. Farm machinery
   - One farmer can only do so much on their own. You will introduce new farm machinery that can help them with their daily tasks.
   - A farmer should only be able to harvest a certain number of plots per day.
     - Once they have reached the **harvesting maximum** for the day, any further attempts should trigger a popup informing them that they've done all the harvesting that they can for the day.
     - The farmer can purchase a **tractor** at the market which will increase the maximum number of plots that can be harvested per day.
   - A farmer should only be able to water a certain number of times per day.
     - *Once they have reached the **watering maximum** for the day, any further attempts should trigger a popup informing them that they've done all the watering they can for the day.*
     - *The farmer can purchase **irrigation** at the market which will increase the maximum number of times they can water per day.*
3. End game
   - If all the player's crops die and they are out of money, they should be taken to a game over screen.
   - The game over screen should give them the option of starting a new game.

Optional: Extra Credit (15 points)

1. Choose one or more from the following implementation goals to receive up to 10 points of extra credit:
   a. For 10 points, implement a way to save and load your game.

b. For 5 points, implement music or sound into your game.
c. For 5 points, implement some kind of "win" end game. Feel free to get creative with it!

2. Brainstorm ideas for future CS2340 projects for up to 5 points:
    a. Provide a description of your proposed project (~5 sentences).
    b. Additionally, provide a brief description (3- 4 sentences) for 5 iterative milestones and the features they will include. The milestones should include implementation-specific functional requirements. For inspiration, please look at this semester's M3-M6 implementation requirements.

Testing Requirements

1. Write **unit tests** to verify the functionality of 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.*
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.

Video Demo and Competition:

In your video demonstration, you will cover all the implementation requirements from this milestone (M6) and highlight meaningful functionality from previous milestones. Your video should be 3-5 minutes in length. Think of this as an application showcase that you would present to a potential client or investor. The contents of your video must display:

1. All implementation requirements for M6 (Farm expansion, Farm Machinery, End Game, and any Extra Credit implementation goals completed).
2. At least **5** implementation requirements from previous milestones (M3-M5). Some examples of functionality that could be shown are the market, the plant growth cycles, and random events.

Additionally, we will have an optional, extra-credit video demo competition. Your team may opt into the competition and may also choose to remain anonymous. **Please include if you would like to compete and/or remain anonymous in competition in your submission.** For the competition, the class will vote on which projects they like the most to decide the winners. The winning teams will receive 8 points of extra credit on the final exam.

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".

Submission Requirements

In addition to your deliverable, 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!** Checkstyle will not be required for this milestone. Points may be deducted if these guidelines are not followed! All deliverables must be submitted to Canvas prior to the deadline.