Major Course Output Specifications

CCPROG3 Major Course Output
AY 22-23 T1

Game Overview

MyFarm is a farming simulation game, where the player acts as the sole farmer (and manager) of their own farm. The gameplay typically involves the following tasks:

- Buying seeds,
- Preparing land,
- Planting seeds,
- Advancing days (err... watching crops grow), and
- Harvesting crops

There's obviously a lot more to the game, but that's the general idea. There's also no real end goal to this game – as in many simulation games. However, to make things clear, the game can theoretically go on forever and the player can continue to play for as long as they want except in the case when they ~~(1) run out of seeds~~, (2) don't have any active/growing crops, and (3) don't have enough money to buy new seeds. The player also cannot continue if all the tiles in their farm lot contain withered crops (i.e. crops that died due to lack of care). When either situation happens, the game ends and the player is asked if they'd want to start a new game or simply exit the program.

Addendum: It is impossible to "run out of seeds". The original specifications included the idea of an inventory system where seeds could be bought. However, as that was removed, the 3rd game ending condition covers the aspect of "running out of seeds" as not having enough money to buy seeds would entail not being able to continue.

The following sections go into detail about how the game is expected to run while the end-game conditions have yet to be met.

Game on!

On the first day (i.e. day one) of the game, the player finds themself with the following:

- A ten (10) by five (5) farm lot,
- Zero (0) seeds,
- One hundred (100) Objectcoins, and
- Farmer's level zero (0).

The Farming Process

The farm lot starts off with none of the tiles containing any crops and all the tiles being unplowed. Seeds cannot be planted on unplowed tiles, so the player must first use a plowing tool to... plow a tile. Once a tile has been plowed, the player can properly plant a seed in that tile. Upon planting a seed in a plowed tile, the option to water and fertilize the specific tile/crop becomes available to the player. Watering and fertilizing are only available when a tile is plowed. Just like in real life, both water and fertilizer have some effect on a crop's growth. Without properly taking care of a crop, the crop will wither and stay on the tile forever. If the crop is able to successfully grow (i.e. enough time has passed and the minimum requirements of the crop have been met), then an option to harvest the crop should be available to the player. Harvesting a crop that's ready results in the player gaining *n* number of produce

should be available to the player. Harvesting a crop that's ready results in the player gaining a number of produce. Some crops produce only 1 product, while others may produce more. Regardless of the number, all produce from the harvested crop are immediately sold for a specific amount, which is credited to the player's Objectcoin wallet immediately. After harvesting a crop, the crop is removed from the tile and the tile reverts to an unplowed tile. Take note, a successfully grown crop must be harvested in the day it is ready. The player also gains some farming experience from successfully grown crops. Forgetting to harvest a crop (i.e. allowing a day to pass after the crop is ready) will result in the crop withering and the player losing any benefit from the harvest.

And that's the main grind! But as you've probably guessed, there's more to this game.

Oh no, Rocks!

While the farm lot starts off with no crops, there should be rocks that are scattered across the 10x5 lot. Rocks prevent any kind of player action from being performed on a tile (i.e. can't plow, so can't plant) – except for actually removing the rock. To remove the rock, a player must use the pickaxe tool which (*for some game-related reason*) costs 50 Objectcoins to use. Once a rock is removed from a tile, the player gains access to plowing the tile.

At this point, you might be wondering how many rocks are initially on the lot, as well as how they are scattered. Answer: the number of rocks and manner of scatter should be determined via file input. There is no specified format to the file, so feel free to design your own representation of the 10x5 lot and the initial rock scatter. This file should be read on the start of a game and there should be at least 10 rocks and at most 30 rocks. Assume the input file is always in a valid format.

Withering Away …

Whenever a crop withers, it stays on its tile *forever*… unless you have a little bit of money. When a tile has a withered crop, the option to use the shovel tool becomes available. Hence, a player can use the shovel tool to remove a tile's withered crop, but (*for some other game-related reason*) it costs 7 Objectcoins. After removing the crop, the tile reverts to an unplowed tile.

Unlike the pickaxe tool, the shovel tool can be used outside of its intended purpose. If used on an unplowed tile or a tile with a stone, the tile remains as is and the player simply loses Objectcoins. If the shovel tool is used on a tile with a plant, the plant would be removed and the tile reverts to being unplowed. Additionally, the water and/or fertilizer status of the tile (if previously applied) is removed. The player also loses 7 Objectcoins for using the shovel in this scenario.

Farming XP

Every time the player successfully harvests a crop or uses a tool, they gain a specific amount of experience and after gaining enough experience, the player automatically levels up. On the start of the game, the player starts with zero (0) experience. Therefore, the player starts off at level zero (0). Each succeeding level is achieved by moving up 100 experience points. Take the following cases for reference:

- Experience: 0; Level: 0
- Experience: 40; Level: 0
- Experience: 99; Level: 0
- Experience: 100; Level: 1
- Experience: 199; Level: 1
- Experience: 201; Level: 2
- Experience: 500; Level: 5
- Experience: 1000; Level: 10

- ~~Experience: 1000, Level: 10~~

Once a player's level is high enough, they may choose to register themselves for a fee. With registration comes benefits for the player, such as earning more per harvest and spending less for purchases. Please refer to the following table for all levels of registration and their benefits:

Farmer Type	Level Requirement	Bonus Earnings per Produce	Seed Cost Reduction	Water Bonus Limit Increase	Fertilizer Bonus Limit Increase	Registration Fee
Famer (default)	0	+0	-0	+0	+0	n/a
Registered Farmer	5	+1	-1	+0	+0	200
Distinguished Farmer	10	+2	-2	+1	+0	300
Legendary Farmer	15	+4	-3	+2	+1	400

Please note that registration is not automatically awarded upon achieving the minimum level requirement because a fee must be paid. If the player does not have enough Objectcoins to proceed, the registrations imply fails. Hence, registration must be explicitly selected by the player. Additionally, the player may choose not to register at all despite reaching a high level.

Tool Usage

The player's actions have already been described previously, but let's summarize them as tools and properly define the expectations of each. Please refer to the following table:

Tool	Function	Cost of Usage	Experienced Gain from Use
Plow	Converts an unplowed tile to a plowed tile. Can only be performed on an unplowed tile.	0	0.5
Watering Can	Adds to the total number of ~~tiles~~ times a tile/crop has been watered. Can only be performed on a plowed tile with a crop.	0	0.5

Fertilizer	Adds to the total number of ~~tiles~~ times a tile/crop has been applied with fertilizer. Can only be performed on a plowed tile with a crop.	10	4
Pickaxe	Removes a rock from a tile. Can only be applied to tiles with a rock.	50	15
Shovel	Removes a withered plant from a tile. Can be used on any tile/crop with varying effects, as described above.	7	2

* Cost of usage is immediately deducted from the player's Objectcoin wallet if they have enough money and the appropriate conditions are met.

Seed/Crop Information

There are three different types of crops available: Vegetable, Flower, or Fruit Tree. Each of the crops has certain nuances that set them apart from each other. In turn, each individual crop varies in terms of needs – water and fertilizer – to properly grow and become harvestable. Otherwise, they will wither. The list of all seeds available in the game, as well as other important information, can be found in the following table:

Seed Name	Crop Type	Harvest Time in Days	Water Needs (bonus limit)	Fertilizer Needs (bonus limit)	Products Produced	Seed Cost	Base Selling Price per Piece	Experience Yield
Turnip	Root crop	2	1 (2)	0 (1)	1-2	5	6	5
Carrot	Root crop	3	1 (2)	0 (1)	1-2	10	9	7.5
Potato	Root crop	5	3 (4)	1 (2)	1-10	20	3	12.5
Rose	Flower	1	1 (2)	0 (1)	1	5	5	2.5
~~Turnips~~ Tulips	Flower	2	2 (3)	0 (1)	1	10	9	5

Tulips								
Sunflower	Flower	3	2 (3)	1 (2)	1	20	19	7.5
Mango	Fruit tree	10	7 (7)	4 (4)	5 – 15	100	8	25
Apple	Fruit tree	10	7 (7)	5 (5)	10 – 15	200	5	25

Notes:

- *Harvest time* is interpreted in days past. For example, if a crop is planted on day four (4) and harvestable in two (2) days, then it would be ready for harvest on day six (6) from the current day. If the player advances to the next day, any crop that are ready to be harvested will automatically turn into a withered crop. The harvest option should only be available to the player when the crop is harvestable. A player should not be able to harvest a withered crop.
- Successfully harvesting a crop is free of cost and should result in experience. Refer to the *experience yield* in the table above for how much experience is earned per crop.
- *Water/fertilizer needs* refer to the minimum requirements for a crop to become harvestable. If these needs are not met by the crop's harvest day, then the crop automatically turns into a withered crop.
- *Water/fertilizer bonus limit* refers to the maximum times a crop can be watered or added with fertilizer. For example, if one waters a crop ten (10) times but its bonus limit is only 2, then during the computation of the final price, the watered count is capped at two (2). There is no consequence for overwatering or placing more than needed fertilizer (*wow sana all*)
- *Products produced* refers to the capability of a crop to produce *n* number of products. When a single value is specified, the crop should only produce the single value of products upon harvest. When a range is specified, the crop should randomly produce *n* number of products using the min and max specified range. For example, if a crop can produce 1-3 products, then the crop should only be able to randomly produce 1, 2, or 3 products upon harvest.
- *Seed cost* refers to how much the… seed costs to buy. Seeds are bought upon planting, so there's no concept of an inventory of seeds. There is simply a list of seeds that can be bought. If a player has enough money and the tile has been plowed, then planting a seed functions as both the buying of a seed and the placement of the seed into the tile.
- Each product from a harvest has a *base selling price*. For example, if a crop produced 3 products and has a base selling price per piece of 10, then the combined selling price for the harvest is 30.
- When a single seed is planted into a tile, the tile is considered occupied. Hence, only one (1) seed can be planted into a tile at a time. This applies to all crops. However, the *fruit tree* crop has a special planting condition. Aside from needing enough money to buy the seed and having the tile plowed, there should be no tiles next to the selected tile with any objects. This means fruit trees should be planted away at least one (1) tile away from occupied tiles – check immediate sides and diagonals. A tile with a withered plant or a rock is also considered occupied. This condition only applies during the planting of a fruit tree. Once a fruit tree has been planted, root crops and flowers can be planted in tiles next to it. Note that planting a fruit tree on a corner or far sides of a farm lot is not allowed.
- To serve as a guide for computing the final selling price of a harvest, refer to the following:
    - HarvestTotal = ProductsProduced x (BaseSellingPricePerPiece + FarmerTypeEarningBonus)

- HarvestTotal = ProductsProduced x (BaseSellingPricePerPiece + FarmerTypeFarmingBonus)
- WaterBonus = HarvestTotal x 0.2 x (TimesCropWasWatered – 1)
- FertilizerBonus = HarvestTotal x 0.5 x TimesCropAddedFertilizer
- FinalHarvestPrice = HarvestTotal + WaterBonus + FertilizerBonus
- **Notes**:
  - TimesCropWasWatered and TimesCropAddedFertilizer are capped by the crop's respective water/fertilizer bonus limit. Just don't get to note that certain registered farmer types also modify the water/fertilizer bonus limit.
  - The computation assumes the crop's minimum needs are met. Otherwise, the computation should not be made.
  - Because flowers are pretty, all flower crops are given a minor premium. Hence, flower crops have an additional step after the FinalHarvestPrice is computed:
    - FinalHarvestPrice = FinalHarvestPrice * 1.1
  - If a crop is successfully harvested, the player should be informed of how many products were produced and how much they were able to earn in total. If a crop withers for any reason, the players should also be alerted.

Advancing to the Next Day

While this has already been hinted on, the game works on a day system and starts on day 1. The player should be able to advance to the next day when they'd like to. Advancing to the next day should trigger updates to entities throughout the game. For example, if a crop needs 3 days until harvest, then advancing to the next day would mean the crop would need only 2 more days to harvest. Another example would be when a crop is ready for harvest but the player advancing to the next day, the crop should be updated as it should have withered.

```
Player Interaction Design
```

While these specifications outline expectations of the game, you're given the freedom to implement how the player interacts with entities within the game. For example, the player must be able to view the farm lot and the different objects in each tile. You're given the freedom to design how the player can do such for each MCO phase. This also applies to other interactions within the game, such as how the player interacts with tiles (in viewing tile information and in interacting with tiles), how the player views their current experience/level, how the player can register for the next farmer type, or how the player can proceed to the next day. Additionally, the game should be providing information to the user when needed. For example, if they don't have enough money to use the pickaxe, the player should be informed. If a player harvests a crop successfully, they should be informed of the details of the harvest. Your approach to player and game interactions does not need to be innovative, but it should at least be understandable to execute. **Prioritize functionality over form/style**. It is expected that player interactions would be explained in the demo of each MCO phase.

Instructions and Deliverables

Your task is to create an implementation of the farming game described above in Java. As this is a culminating activity, you're expected to apply the different object-oriented concepts learned throughout the term. Create and use methods and classes whenever possible and make sure to use object-oriented concepts properly. No brute force solution.

This project is at most done in groups of 2; however, one may work alone if they wish. A student should not discuss or ask about design or implementation with other individuals, except for the teacher and their groupmate.

Copying other people's work or working in collaboration with other teams is not allowed and is punishable by a

Copying other people's work or working in collaboration with other teams is not allowed and is punishable by a grade of 0.0 for the entire CCPROG3 course. A discipline case may be filed with the Discipline Office. In short, do not risk it; the consequences are not worth the reward and taking a "shortcut" implies one would miss out on the opportunity to learn important foundation concepts.

The above description of the program is the basic requirement. Any additional feature will be left to the creativity of the student. Bonus points would be awarded depending on the additional implemented features. These additional features could include new types of game elements, including new relationships/restrictions among the game elements (on top of the specifications). Depending on the scale of the new feature, additional points will be awarded to the team. However, make sure that all the minimum requirements are met first. If this is not the case, then no additional points will be credited despite the additional features.

We would just like to emphasize that you **DO NOT** need to create and/or stress over assets (e.g. background pictures, item/product pictures, etc.) for the game. Functionality is more important than style for this assessment, so only consider improving the aesthetics of your solution if you've met the specifications. You may use available resources found on the Internet; however, remember to cite where you got the assets. For citations, your group can create a readme file with the citation lists and include the file in the project folder.

MCO Phase 1

For MCO1, you will be required to submit an object-based **UML class diagram** of MyFarm. Your group will also be submitting a declaration of original work signed by everyone in the group. This document may contain all your sources and citations. Note that by the deadline of MCO1, not all the OOP concepts would have been discussed. Hence, expectations for your UML class diagram are adjusted based on the currently discussed topics.

Additionally, a **working object-based prototype** is expected to be submitted. For the prototype, implement the application such there is only one (1) tile and that there is no presence of rocks. The player should be able to plow the tile, plant (and buy) a Turnip seed, water the tile/crop, advance to the next day, and harvest the crop. The crop should wither if it is not harvested on its harvest day or if it isn't watered. There is no need to implement the removal of the withered crop for MCO1. A seed should not be planted on an unplowed tile or on an occupied tile. Buying a turnip or successfully harvesting a crop would result in the decrease or increase of the player's Objectcoin. Simple feedback should be provided to the user, such as how many Turnips were produced from a harvest or if a seed could not be planted. Lastly, no GUI is expected for MCO1, so please implement the prototype such that interaction is made via the terminal/console/command prompt.

Lastly, a **pre-recorded presentation** of the deliverables is expected. This presentation should explain the group's general logic used in designing the UML class diagram and demonstrate their prototype. Keep the presentation short, but ensure to cover vital information, such as how considerations went into modeling entities and how the prototype can simulate a planting to harvesting cycle.

Upload all deliverables to the respective **Canvas** assignments by **9:30 pm of Oct 29 (S), 2022**. As a reminder of university policies, 9:30 pm is the start of the university break on Saturdays. *The deadline has been adjusted by roughly a week to accommodate for the time of the release of the specifications.*

MCO Phase 2

For MCO2, your group will be required to have applied appropriate object-oriented based concepts to design MyFarm. At this phase, the program should have a **graphical user interface** implemented using the **MVC**

architecture. You will also be required to submit your final UML class diagram, declaration of original, Javadoc-

architecture. You will also be required to submit your final UML class diagram, declaration of original, javadoc generated documentation, and all source files (zipped). Again, the submission of files will be via Canvas.

During the MCO2 **MP demo**, it is expected that the program can be compiled successfully and will run. If the program does not run, a grade of 0 will be given. However, a running program with complete features may not necessarily get full credit, as implementation (i.e., code) will still be checked. All members of the group should also be present – whether live or online. Apart from the question and answer, it is possible that a demo problem is given to the group as part of the demo. A student or a group who is not present during the demo or who cannot answer questions regarding the design and implementation of the submitted project convincingly will incur a grade of 0.

Upload all deliverables to the respective **Canvas** assignments by **2:30 pm on Dec 7, 2022 (W)**. As a reminder of university policies, 2:30 pm is the start of the university break on Wednesdays.

Documentation

Do not forget to include internal **documentation** (comments) in your code. At the very least, there should be an introductory comment and a comment before every class and every method. This will be used later to generate the required External Documentation for your course output. You may use an IDE or the command prompt command javadoc to create this documentation, but it must be PROPERLY constructed.

For any questions or concerns, feel free to reply to this discussion thread so that everyone can benefit from the asking and answering of questions.