OBJECTP

Machine Project

Release Date: **July 9, 2018**

Due Date for Phase 1: **July 23, 2018**

Due Date for Phase 2: **August 13, 2018**

MyFarm – Farming Simulator

I. Introduction and Specifications

MyFarm is a farming simulation game, similar to FarmVille. Its gameplay involves different aspects of farm management. This includes buying of seeds, plowing of land, planting, growing, and harvesting. Currency in game revolves around Objectcoin (OC), the newest and most object-oriented cryptocurrency around.

A player starts with an empty lot, and a fixed amount of MyFarm Coins. Players also earn XP (experience points) from performing activities on the farm, such as plowing of land, harvesting or buying of items. As the XP reaches a certain score, player level rises. The higher the player level, the more discount he receives when buying items, and the more coins he earns when selling.

A player may either be registered or not. A registered farmer has three levels – registered farmer, distinguished farmer and honorable farmer. Farmer needs to pay to register. Registered farmers enjoy additional benefits.

Farmer Type	Level Requirement	Earning/buying	Water/fertilizer bonus limits	Harvest time	Registration Fee
Farmer	0	+/- 0	+ 0	- 0%	n/a
Registered Farmer	10	+/- 2	+ 0	- 5%	200
Distinguished Farmer	15	+/- 3	+ 1	- 10%	250
Honorable Farmer	20	+/- 5	+ 2	- 15%	350

A player can earn OC by harvesting the product. This is done by plowing a unit of land, planting seed, and eventually harvesting once enough time has passed.

In order to achieve the different tasks needed in the game, a player has access to certain tools that are immediately available to the player, as described in the following table:

Tool	Function
Watering can	Waters a specific crop; Can dispense an infinite amount of water
Plow	Prepares a specific title for planting; Also removes withered plants (costs 2 OC to remove)
Pickaxe	Used for destroying rocks obstructing tiles
Fertilizer	Not actually a tool, but is bought in finite amounts and fertilizes a specific tile; Cannot be placed on a tile with a plant; Costs 10 OC; At the start of the game, the user originally has 5 units of fertilizer

The player must also be able to see the field for farming. The field has a dimension of 10 units by 5 units. Each cell is called a tile. A tile is originally an unplowed tile, where seeds cannot be planted. Using the plow turns it into a plowed tile, where seeds can be planted. Whether a seed has been planted or a crop has withered, the tile is considered occupied. Once a crop is ready for harvest, a product appears, but is only credited to the user upon selection. After harvest, the plant is removed and the tile reverts to an unplowed tile. If a withered plant had been removed, the tile will revert to an unplowed tile. A majority of the tiles are clear of rocks, but there are a few tiles with rocks where a pickaxe is needed to clear them before plowing can be performed.

The player must also be able to see the different tools and seeds, like an inventory. There should also be an area that shows the status of selected objects. Selecting a tile should show information about the tile. Selecting a tool should show information about a tool and options it can do with it (e.g. selecting the plow brings up the option to plow a tile or remove a withered plant). Selecting a seed should show information about the seed and options whether to plant or to buy more.

Seeds available are classified as either Vegetable, Flower, or Fruit tree. Different crop types have different amount of needs – water or fertilizer. Each crop varies in terms of needed. The list of all seeds available in the game, as well as other important information, can be found in the following table:

SN	**CT**	**HT**	**WN (bl)**	**FN (bl)**	**HC**	**PP**	**SC**	**BP**
Turnip	Vegetable	1	1 (2)	0 (1)	1	1	5	6
Carrot	Vegetable	1.5	1 (2)	0 (1)	1	1-2	10	9
Tomato	Vegetable	2.5	3 (4)	1 (2)	1	1-3	20	15
Potato	Vegetable	5	4 (5)	2 (3)	1	1-6	25	13
Rose	Flower	1	1 (2)	0 (1)	2	1	5	5
Tulip	Flower	1.5	2 (3)	0 (1)	2	1	7	7
Stargazer	Flower	2.5	2 (3)	0 (1)	2	1	10	9
Sunflower	Flower	3.5	2 (3)	1 (2)	2	1	20	19
Mango	Fruit tree	7	7 (7)	4 (4)	3	5 – 10	50	4
Apple	Fruit tree	7	7 (7)	5 (5)	3	7 – 10	55	3.5
Banana	Fruit tree	8	8 (8)	5 (5)	3	10 – 15	60	3.5
Orange	Fruit tree	8	8 (8)	6 (6)	3	13 – 15	65	4.5
Legend: **SN** – seed name**; CT** – crop type**; HT** – time to harvest in minutes**; WN (bl)** – water needed (bonus limit); **FN (bl)** – Fertilizer needed (bonus limit)**; HC** – harvest cost**; PP** – number of products produced**; SC** – seed cost**; BP** – base selling cost per piece produced								
Each seed/crop/withered crop takes up 1 tile; however, fruit trees cannot have other seeds/crops next to them (must be clear of crops 1 tile from the center tile in all directions)								

Once a crop's harvesting time has finished, a product will appear. If harvested within a minute from the time it appeared, the action will produce a number of products *n*, where *n* is the number of products produced stated in the table above. Some crops produce only 1 product, while others may produce more. After which, the products are immediately sold and an amount is credited to the user's OC wallet. The actual selling price *sp* of an individual crop product can be computed as such

$$sp = flb + \ bp + wb + fb + cb,$$

where *flb* is the farmer's level bonus, *bp* is the base price, *wb* is the water bonus, *fb* is the fertilizer bonus, and *cb* is the crop bonus. For *flb*, the actual value varies according to the farmer's level and is independent of the rest of the values. The water bonus is the actual number of times watered multiplied by quarter of the *bp*. Similarly, the fertilizer bonus is the actual number of times fertilizer was applied to the tile of the plant multiplied by half of the *bp*. Both the *wb* and the *fb* should be capped by the bonus limit stated in the table above. As for *cb*, all crops except for flowers have no bonus (*cp* set to 0). For flowers, *cb* is set to 5% of sum of the *bp*, *wb*, and *fb*. Flowers have a bonus because they are pretty. 😊

If a crop's water and fertilizer needs are not met by the time of their harvesting, then the crop withers and does not produce a product. Products disappear if they are left on the crop for more than 1 minute after they appear. The crop also withers in this case. The player must be alerted as to the success or failure of a harvest, how many crop products was produced, and how much profit was made, if any.

A withered crop stays on the tile for a total of amount of time equal to twice the amount of the crop's HT (*time until harvesting*). The player should also have an option to pay 10% of the total amount of SC (*seed cost*) for the removal of a withered crop. This option should be done through the plow tool.

II. Instructions and Deliverables

Your task is to create implementation of the MyFarm – Farming Simulator based on to the description above. You are required to apply the object-oriented concepts learned. You are required to create and use methods and classes whenever possible. Make sure to use object-oriented concepts properly. No brute force solution.

This project is at most **done in groups of 2**. A person may work alone. A person cannot discuss or ask about design or implementation with other persons, with the exception of the teacher and his/her groupmate. 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 OBJECTP course. A discipline case may be filed with the Discipline Office. In short, do not risk it; the consequences are not worth the reward.

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. 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 your own sprites (background pictures, item/product pictures, etc.) for the game. You can just use what is available on the Internet and just include them into your project. If you wish to do so, we would also like to remind you to please cite where you got your sprites.

For MCO1, you will be required to submit a complete **object-based** UML class diagram of MyFarm via **Canvas**. You will also be submitting a declaration of original work signed by everyone in the group. This document may contain all your sources and citations.

For MCO2, your group will be required to have applied appropriate **object-oriented based** concepts to design MyFarm. You are to submit a complete MyFarm program with a graphical user interface following the MVC 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, submission of files will be via **Canvas**.

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 Machine Project. You may use an IDE or the command prompt command javadoc to create this documentation, but it must be PROPERLY constructed.

During the 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. The group should use the machine in the lab for the demo and should know how to generate the bytecode file and to run the said file in the command prompt. Apart from question-answer, it is possible that a demo problem be 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.

III. Grading Breakdown

Your project will be graded as such:

MCO1 (*20% total grade*)		**MCO2 (*20% total grade*)**	
UML Class Diagram	100%	UML Class Diagram (OO)	15%
		Graphical User Interface	20%
		Model View Controller Architecture	15%
		Simulation	40%
		Programming: Individual	10%