Seng201_2020- Farm Stimulator

Project member: Edward Wong Zhan Yip, Yangpin Lin ezw21, yli406

Decision_1: Development of GUI(Edward) and Game Structure(Sam). UML draft is done by both.

- Edward starts to develop a setup frame, Sam will work on game structure.

Decision_2: Development of GUI(Edward)

- The setup frame is done, Sam's prototype of the game structure was barely implemented. Edward help to develop game structure as it's needed for further GUI development.

Decision_3: Development of GUI(Edward), Game structure is done by Edward

- Using Card Layout to switch between panel, and implement GUI_body as panel manager to perform such action. In use with layered pane and multiple panels.

Architecture / Design pattern(Edward) GUI is done(Edward)

- Decided to approach Model – View – Controller design pattern.
- Controller class as Controller to interact with Model to create data for View.
- Game_Profile class as Model to store data for updating View. (Crop plot - model class for farm)
- The multiple panel was implemented in different classes as View to relays user command/action. (Method in controller might directly update View).

Object-Oriented developing style allowed the communication between functional classes to become easily achieved. Game_Profile, Model class which used encapsulation to separate the model from the functional classes which prevent the functional class directly manipulate the data in Model Class. It provide a more flexible and secure layer for the functional class to get and set a variable through the method instead of changing the data directly. Besides, Game_Profile can achieve data sharing between panel classes, by including references of a single Game_Profile object into different panel class. Base_component class contain a frequently used method that update the display of base label such as (Stats, Energy, Inventory) to reduce the redundant code which include in most of the Panel classes .

Panel classes such as Barn, it can relay display and manipulate the data in Game_Profile when action is performed or triggered by user.

Coverage percentage – 52.5%

3.7 % lost because Game_Setup is in another frame.

3.5% lost due to UnitTest as it was not called in GUI_body.

A big source of percentage loss was in Farm -Seeding. It has 6 seed and 4 Crop plot which make a lot of condition not to be covered when GUI_body compile.

UnitTest coverage – 63.7%

UnitTest was assigned to Sam when he can help in the final stage, he cant access Game_Profile data as he is not familiar with the structure. (He did the test with the local variable in separated file).

Edward needs to redo the Unittest a day before due, and the same problem with farm – Seeding.

Also a number of percentage is because there is nothing much to test in a View class.

Thoughts and feedback: (Edward)

As I have gone through from learning Developing GUI, game structure, MVC design pattern, Object-oriented programming, I learned a very huge amount of knowledge and deepen the understanding of java. Doing GUI development was fun and I feel that I did well in this section as a beginner to use Swing,.However, it will be better if the project is Planned in a more Architecture way in the beginning as I spent a lot of time to do modulizing and figure out how to access data and connection between classes.

There are also a few things I wish to improve in my future project:

1. Model Class (Game_Profile) should be broken down into smaller classes may be a Model for each panel/subcomponent. I believe this will make the code more maintainable.
2. I should use more Single inheritance style or Multilevel style as these style form a cleaner view and perspective when we are designing a software. Instead of setting farm type as a variable, I should use inheritance at the first place.
3. Farm seeding condition check was a main drag back for the project as it has too many condition checks. If I have a better knowledge of Inheritance I might figure out a more effective way.
4. Due to the COVID, we cant attend the lab section to meet the tutor. It takes the course difficult to the next level and now only I realize how helpful to have a tutor to guide us. (I mean it!)

Between the first 2weeks of this assignment period, I played the classic Farm game harvest moon to think and understand the algorithm behind making a game, it gives a good perspective on how to design and develop a game software. From 12/05 – 25/05 I spent 10-13hours a day doing this project except for 20/05 where I fall sick. (Nearly 150hrs, not all about coding but also researching MVC and inheritance)

To be honest, more than 95% of the final submission was prepared by me but I have no problem with that as I am in a team. Sam might not achieve much in preparing the final submission, but base on his explanation I choose to believe he also spend a number hours of effort. Be fair, I included his contribution in a individual folder and let the marking team to decide the outcome(I understand a unskilled programmer may bring huge lost to a company, but I also hope he can get a passing mark)

Thoughts and feedback: (YangPin Lin)

This was a challenging project as I consider myself as a programming newbie learning coding for less than 3 months. As instructed in the SENG201 class, a good project management should involve a proper planning with all members agreed and understood on the same objectives, program structures, milestones, a breakdown of short-term goals and an effective reporting and communication format. However, I found that this was particularly difficult to apply in this project based on the state lockdown, skill imbalance and disagreement within the team, and the time constraint of the project.

With all facts listed above, we had to take an emerging approach to accomplish this project. Instead of a top-down design from going through scope agreement, to defining the resources allocation and milestones, we had worked bottom up learning the key skills and tackled problems at the same time. Therefore, I highly value my team member Edward Wong's contribution in this project. Despite my slow adaption, he went ahead and tackled the problems first. As my learning was in progress, he came back and communicated with me about his progress. His forwardness overcame the time constraint and having a challenging team member, and push himself to set up a higher goal to perfect the program. I am in awe of having such an emerging team member.

I have invested 93 hours in this project, however, because some disagreement in working progress and conflict in communication with project planning, most of my investment were not adapted in the final deliverables. With all due respect, we took Edward's design as a preferred approach, because we all understood that time did not allow us to learn and plan for any longer. Thus, I could finally achieve the deadline for this project.

The experience I have learned here is starting as early as possible. I would try to engage the team member to list up all requirements and technical resources before programming, so we all understood each other's capability, responsibility and short-term targets. When we are in progress, we should communicate effectively with each other to identify any goal adjustments, difficulties, skills and other resources required in order to achieve the project objectives.

I understand the Edward as well has invested a tremendous amount of time on this. Most of them are converted in the final deliverables. Therefore, I would not claim these credits as mine. For the final contribution agreement, I would step down and leave Edward to decide.