**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 project 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 classes to become easily achieved. Base\_component class update stats, energy, and inventory and it allows the other classes / all the different panels (Farm, SuperMarket, Barn etc..) to update various data to Game\_Profile. Allow access to the sub-component class(panel) was important and the Object-Oriented developing style was the core pillar that supports the whole project. The inheritance style in this project was mostly Hierarchical (Subclass get value from Fame\_Profile), Multiple(Farm get access from Inventory and Base\_Component), Hybrid(the inheritance of farm have a higher level which is Game\_Profile.)

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% (I didn’t test any getter setter as it appears to be meaningless)

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 several 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. I need to learn more about inheritance and extend to write better code.
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 140hrs I not all about coding but also researching MVC and inheritance) I believe I have contributed at least 95% of the workload for this project.