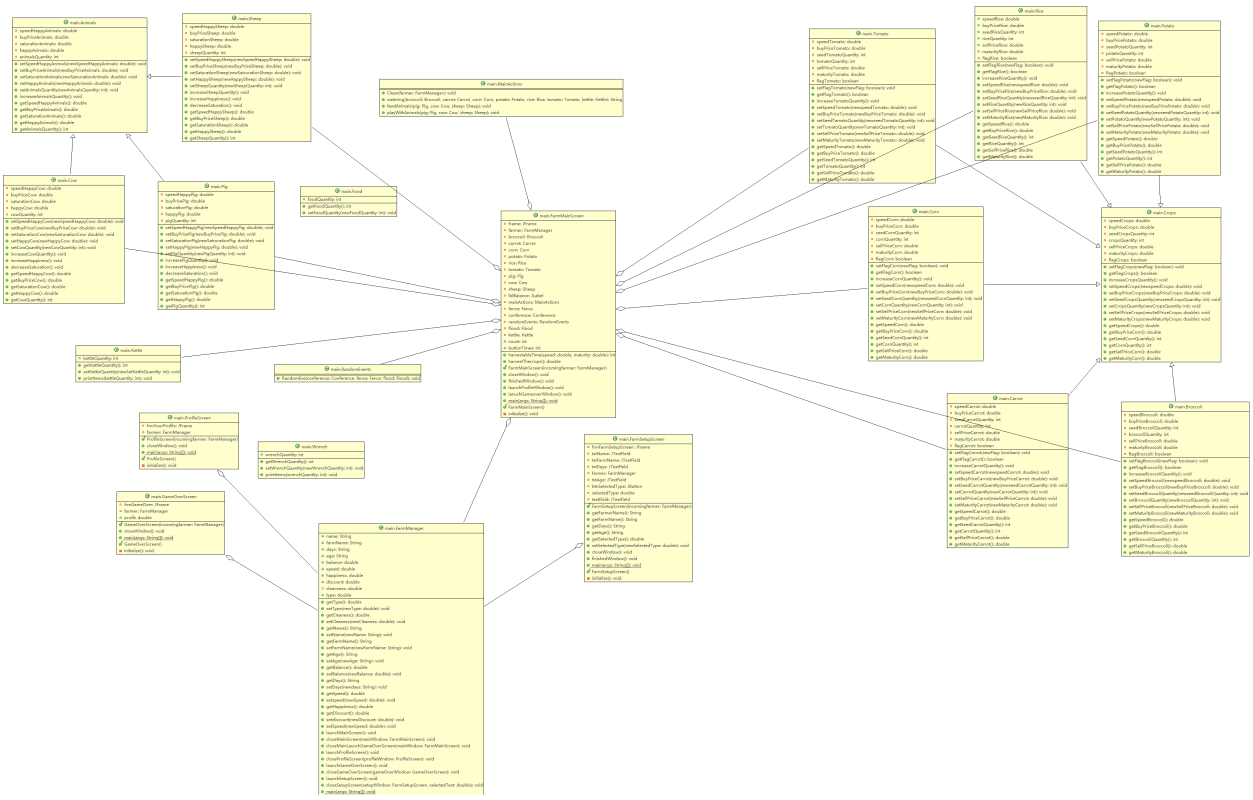
Student name: Yaxian Yu Jiaqi Li
Student Id: 25468326 75766881

1.Structure

In the FarmSimulator project, we had a multiple modular structure. There are some main classes such as crops, animals, main actions. For instance, The crops have some child classes ,broccoli, carrot, corn and so on. It also has some independent ones which are not inherited by any classes. This game has four screens which are FarmSetupScreen, MainScreen, Profile Screen, GameOverScreen. The methods in the FarmManager are for getting the value the player put onto the setupScreen, and then pass the value to the MainScreen. The picture below is the UML of this project.



2.JUnit test

It did help to find what percentage of the code covered by the testing. The unit test coverage is at a low percentage 15%. All of the unit tests are in the test package. We try to test as many methods as we can. However, some classes depend on the other classes which can not be tested properly. We tested some basic getter and setter methods and some other methods such as increaseHappiness, decreaseSaturation. It also helped we find some errors that we did not notice before.

Student name: Yaxian Yu Jiaqi Li
Student Id: 25468326 75766881

3.Feedback

Upon finishing the project, there was a great sense of accomplishment. This project was for both of us, the largest and most complex assignment thus far. This project was particularly interesting because there are a lot of ways you can approach the creation of this game however the end results converge to the same point. This gave an opportunity for us to learn the following lessons:

- The game should establish a clear structure before entering any line of code
- Identify the knowledge gaps early, establish competency for each of the structure we hope to use as well as the structure that we think we will not use.
- Gained understanding and use case for inheritance mechanism for future projects. This can be a key decision made at the start of a project.

4.Review

The key decision to abandon the use of inheritance was that we did not actually know how to use it in the GUI application at first. Although much of the code was completed faster than expected, problems started to appear, and the ratio of time used for fixing the problems outweighed the time used to create new functions. This became a large challenge due to key decisions not being communicated, not knowing the decision was made in hopes to get a head-start on the project and causing multiple duplications of the same functions to be created. Additionally, the knowledge and confidence to use inheritance became apparent well into the completion stages of the project.

The challenges above are the outline of key steps we can take to improve on the next project:

- Plan and establish the structure, required functions, and allocate time for each function.
- Review each function and determine what the key knowledge gaps are; what needs learning and determine if this will increase the efficiency of the code and/or time spent.
-Streamline teamwork by commenting on code, communicating what is completed, what is work in progress, and what is the next task to complete (Use the implementation plan as an outline).

5.Contribution

The contribution of each party is 50% and approximately 80 hours were used to produce Farm Simulator. Jiaqi wrote the command lines, fixed bugs, and add the comments for the classes. Yaxian built the GUI structure and design, fixed the bugs, testing, add comments for the window classes.