Flynn Garthwaite – 26358497

Samuel Burtenshaw – 41654471

## Structure:

The structure of our project changed though out production the most noticeable difference is due to the changes caused by converting the command line to GUI. Initially the command line game revolved around the main class being the hub of the game and each of the other classes were called from main to control their respective parts of the game. In the GUI game the main class was instead used to control each of the screen transitions leaving the functionally of the game to the individual screen classes. The other classes such as Crop, Farm and Animal continued to be used to implement the underlying functionality of the game and are called using actions like pressing buttons in the GUI.

Inheritance played a large role in how we organised the structure of our game. Specifically, through the implementation of the different crops and animals. Using the Abstract classes Animal and Crop, we were able to sort most of the functionality of each crop and animal into two classes. We decided to do this due to that fact the all the animals and crop subclasses respectively had the same methods and variables but with different stats e.g. chicken have 50 health and 50 value whereas cows have 100 health and 100 value. If we had more time to develop the game, using this method meant it would be easier to give each crop and Animal special abilities by defining the abilities within each subclass. Another decision we made was whether to have an item interface. We decided against this as it would be simpler to have items stored as variables in the bag class and specific item functionality to be implemented in other classes methods. An example of this is when tending crops, the tend method will have different effects based on the item selected. The item is represented as a string and can only be selected if it is contained in the bag.

## Tests:

Our tests covered 86% of our non-GUI code. We believe this is the case because many of the getters and setters only accessed through the GUI. While we attempted to test the GUI, using the SetupScreenTest test for example, most of the functionally would have to be tested using Java Robots. This could potently cause problems due to screen sizes and window sizes, so we decided against this. Instead opting to use manual testing instead.

## Thoughts and feedback:

Sam:

I believe this project helped not only with my java coding experience but also helped with my understanding of object orientated coding practices. While I usually prefer to create practical applications, creating a game was probably a good idea especially as most of us are not used to programming in java before but have a basic understanding of how games work. I feel this may have been a key factor in choosing to produce a game over an application, but it may have been nice to have a bit more freedom in choosing what we wanted to create. Overall, I believe that this project has been a good experience and will help with my next endeavour into coding and project development.

Flynn:

I found the project to be quite enjoyable as it gave me an insight into how to effectively use java to code a small project in a group. However, I do wish there were a few more examples of how to do somethings in the project. Overall, I thought the project we quite fun and I enjoyed working on it and designing the program from scratch and I believe this experience will help me in the future.

## Retrospective:

Overall, we found that the project went relatively smoothly asides from a few things. One of the issues was to do with GitHub our IDE. Due to the implementation of our project we initially had both the GUI and command line games in the same project. We quickly realised that in order to have both games running from the same project we would have to copy many of the methods in order to preserve functionally from both. Later, in development we decided to separate the games into different project folders so that the changes in the methods when developing the GUI would not affect the command line game negatively.

In the end, we are quite happy with the finished game. One thing that we would do differently in future would be to not spend as much time on the command line game as once we had started the GUI version, we discovered that a fair bit of our work on the command line game was redundant. This would have given us more time to develop more features, however, we were still able to implement the core requirements in time.

## Effort:

Sam – 55 hours

Flynn – 50 hours

## Contribution:

Sam – 55%

Flynn – 45%