SENG201 (Software Engineering I) Project

Names: Yu Duan, Joyce Cheah
Student IDs: 45239028, 34012825

The application was structured with four packages, main, views, resources, and test. This was done to separate the game logic code, swing code, images, and unit tests, to manage code easier. The class ‘GameEnvironment’ is where all the objects of the other classes are created and initialized. This class also launches the application and is used to launch and close all the GUI windows.

Inheritance was used in classes ‘CrewMember’ and its subclasses, and ‘Item’ and its subclasses. The subclasses of ‘CrewMember’ are classes that extend class ‘CrewMember’. These classes are used to create the different types of crew members available for the players to chose. This was decided because all the types of crew members use all the methods in ‘CrewMember’, and also some of them change extends on the method in ‘CrewMember’. For example, the class ‘Engineer’, which extends ‘CrewMember’, has a method repair(). This method builds on the ‘CrewMember’ repair() method. The subclasses of ‘Item’ are classes that extend class ‘Item’, and they are ‘FoodItem’ and ‘MedicalItem’. Both subclasses use the methods in ‘Item’, but both ‘FoodItem’ and ‘MedicalItem’ require other attributes as shown in the UML class diagram.

A design choice that needed to be made was that whether or not create subclasses that extend ‘FoodItem’ or ‘MedicalItem’, as different types of food and medical items needed to be created. The decision was to create these items as instances of ‘FoodItem’ and ‘MedicalItem’, because they don’t need extra methods or attributes required thus making them subclasses is not optimal.

The package Collections was used to program the application. They are ArrayList, TreeSet and Hashmap. ArrayList was used the most, to contain the crew members, food items, medical items, outpost food items, outpost medical items, and others which can be seen in the UML class diagram. TreeSet and HashMap were used differently. They were used for temporary storage for data. They were used with Collections.frequency to manage the add to cart and remove from cart system in outpost.

The total unit test coverage is 20.6%, with a 0% test coverage on the views package, 75.5% on the main package, and 99.9% on the test package. The overall test coverage is low, due to the swing code in the views having a 0% test coverage. The swing code was tested using black box testing instead of unit tests, where the game

was played by different people. The main package containing all the code only has a 75.5% test coverage, because getters and setters were not tested, as the code for them are too simple, making it a waste of time to test. There were sections of code that could not be tested completed, such as methods with random effects. Other sections of code not tested using units test are methods that return a String.

Thoughts and Feedback

Yu Duan:

The project didn't go as smoothly as planned, mainly due to the lack of communication, lack of planning and time management. My partner fails to dedicate enough time to the project, which delayed the progress of the project. But the project is completed.

Improvements that can be made is to plan the project more thoroughly before hand, and communicate with others more often.

Joyce Cheah:

The project was more complicated than expected. I have been involved in more activities than I had time for, therefore failed to allocate enough time to contribute my fair share of this project. My partner has been very helpful and understanding. In the future, I will make sure that I have enough time and not commit myself to too many activities at any time.

Effort Spent:

Yu Duan: 100 hours.
Joyce Cheah: 70 hours.

Agreed % contribution from both partners:
Yu Duan: 60%
Joyce Cheah: 40%