Introduction

This report informs essential information about the SENG201 project and how the process is executed/implemented by Bach Vu and Linh Luu.

The report covers six parts. The first section indicates the structure of the application, any design insights/ideas we have made and collections we have used. In the second part, we give some explanations of unit test coverage and the pros and cons of its maintenance. Thirdly, there are some reflections about what went well and not well during the project and what improvements we could make for nexts. The fourth and fifth sections are about the declaration of partners’ contribution, included a brief describe of key involvement in project by individual and feedbacks on the assignment. Lastly, there is an attachment of the UML class diagram.

2. The structure of the application

**• The structure of your application and any design choices you had to make. We are particularly interested in communication between classes and how inheritance was used. You might want to reference your UML class diagram.**

* Our project is called “SpaceExplorer” which is a strategy and adventurous game. It is several-platform game which is supported by PC and various types of laptops.
* About its structure, we built multiple dependent and independent classes, a superclass and an interface to include abstract methods and constants.
* A superclass named Entity, we used to extend in Outpost, Spaceship, BlackHole and Planet class ...
* The independent Crew and Galaxy classes are not inherited by any others, however, they implement the entire status of crew members in Spaceship and the creatures of the galaxy. In details, every crew has a hull, hunger and morale rows and a supplements stored in Stock for use as well.
* The stock interface is covered classes Stock\_Medicine, Stock\_Food and Ship Module

**• Where collections have been used and in what form.**

* We mainly used ArrayList collection to store items in stock, entity, planets, the number of crews as well as their actions and ship modules.

**• An explanation of unit test coverage, and why you managed to get a high/low percentage coverage.**

* The unit test part covered:
* The valid statistics of the crew via hull, hunger and morale.
* The proper statistics of the spaceship with special chosen crews. For example, if the player chooses one Captain per four crews. Then automatically all figures should be increased by 25.
* Whether the use of stock is correct and how properly the figure changed. (cover the healing and supplement items)
* Test the movement of the ship.
* The correctness of the outpost trade section.

=> warrant/guarantee the balance of the game, keep it sustainable and detect unintended effect side.

**Include on the second page: • Your thoughts and feedback on the project.**

* We assume this project successfully not only improved our knowledge in JAVA, but also broaden our insight/mind about how a real game is made.
* Teamwork and communication skills are necessary for a successful project.
* If everyone is seriously aware of their duties and perform them passionately, and creatively with a harmoniously teamwork, the whole progress would high potentially smooth. Although facing difficulties is inevitable during the project, be patient and discriminately figure out the problems which might be the optimal solution. That was also what we suffered from/experienced in the project and how we went through those problems.
* In overall, we are both satisfied with this project. This is an ideal example/assignment for students to introduce JAVA and a strict chain of requirements that software engineers have to manage in teamwork projects.

**• A brief retrospective of what went well, what did not go well, and what improvements you could make for your next project.**

* At every beginning of the project, we got confused at each other ideas to construct and develop the GUI game because of different point of views.
* We skipped the sketching the outline part and went straight ahead to coding which brought us various struggles. We took first days to figure out, then decided to roll back to the first step and the progress finally started working very well on track.
* Got a little trouble with the length of classes. Especially, the GUI Game Environment contained a huge amount of details, therefore, it was hard to keep track what was going on and also the readability was downgraded significantly.
* In the following projects, we would consider more seriously from a better sight of how software engineer must have.
* Strictly comply with software requirements.
* During the project, there were no conflict happened.

**• The eﬀort spent (in hours) in the project per student. • A statement of agreed % contribution from both partners.**

* We have put so much effort on this assignment since very first weeks it released before the term break.
* After a short discussion, we agreed that Bach Viet contributed 60%, and Linh Khanh did 40% the project. In details,
* Bach:
* built the main structure
* fixed the tough bugs
* executes JAVA doc, blah blah.
* Linh worked with graphics, also had a hand in (involved in) coding and testing the game. (individual parts are alr shown in JAVA doc :) why we hatta write here again haizzz)