SENG201 Project Report

Authors: Lorenzo Fasano (Student ID:) and Jay Hamilton (Student ID: 84258116).

This report highlights the structure, testing, strengths and weaknesses of the application HeroesAndVillains.

The focus of this project has been to implement the basic design principles of object oriented programming using Java to familiarise ourselves with the Java API and to develop methods of collaborating on a small scale software project.

The project contains 81 Java classes which create flexible, modular and maintainable code. The main classes present in this project are Character.java, Collectable.java, Minigame.Java and Building.java. These four abstract classes form the backbone of the project structure and characterise the expected behaviour in the child classes. Most of the remaining classes such as Hero, Villain, Hospital, RockPaperScissors, HealingItem, and Armor extend one of the four initial classes. Please refer to the UML diagram for the complete list of these classes..

In the core structure of the project the use of enumerations was fundamental for storing and distinguishing between objects of the classes that implement Character.java, Building.java or Collectable.java. In order to implement the ability to store objects in the Shop inventory or in the HeroesSquad backPack , the class Inventory was implemented. The main property of Inventory is a HashMap<Collectable, Integer> that stores the type of Collectable item and its quantity . Supporting methods were then created to check if a collectable was in the inventory, to return a list of the inventory objects and quantities, and to add and delete elements from the inventory.

A similar concept was applied for the creation of the HealingWard class used inside the Hospital class to store the information of the Hero objects that are healed using HealingItem objects. In HealingWard, the main property is healingWard, a HashMap<Hero, Integer>, which stores the hero being healed and the time it will take until that hero is fully healed. The time updates in the HealingWard property are achieved by using a secondary thread and by constantly decreasing the Integer stored for each Hero object.

The communication between the HeroesSquad object and the Building objects was achieved by creating a series of methods that used the HeroesSquad object setters and getters to modify this object depending on events. At a higher level the existence of the Engine class and helpers allowed to keep track of the HeroesSquad object, the right current City and of the current Villain, this was done using GameWindowManager in the GUI application as explained in the next paragraph.

Once the command line version of the game was created the GUI version was started. The class GameWindowManager deals with opening and giving visibility to the windows in the right order and it is responsible of updating the HeroesSquad, Villains and list of City objects (world) while the game is running.

The GameWindowManager class also contains the main logic behind the implementation of the serialisation of the game current status and the scores saving.

Junit5 has been used for testing the application.A total of 41 JUnit testing classes were created to make sure we were writing reliable code. Junit extensions @BeforeEach, @AfterEach and @RepeatedTest(integer) where particularly useful for speeding up the tests creation and to make sure that tests were run independently from each other. Only after thoroughly testing the low level classes the higher level functionality, such as the implementation of City and HeroesSquad classes, was added and then tested. The method interact() present in objects that extend Buiding.java was useful for the creation of the command line version of the game and, once the game was finished it remained important to be able to test the functionality of Building, HealingWard, Inventory, HeroesSquad, City and Villains classes. The final test coverage of the project is around 52%.The modularity of the code helped having high testable code, however, no Swing window has been tested, in the next project more attention will be put in the discovery of new techniques to test the GUI components.

The overall result of this assignment has been rewarding as it was a significantly bigger project compared to what we have done in the past, the collaboration has been intense and a significant effort has been provided by both partners. Agreed deadlines were respected and the communication was consistent.The use of GitHub as a version control platform facilitated code updates and backups.

This project has been time consuming as both parties were new to software development using Java, scaling techniques and testing, this project helped both partners to learn how to use API documentation and how to seek help whenever the application did not work properly, this is probably the most valuable part of this experience.

Both partners agree that they contributed 50% each in the creation of this project, this includes testing, Javadoc, UML and general tasks such as the game design and the architecture implementation.

Lorenzo Fasano: My focus has been the creation of city, collectables and part of the engine packages and related Swing windows, testing and Javadoc. The most laborious processes were the creation of reliable low-level classes and the communication between the HeroesSquad object with each building.

Jay Hamilton: My focus has been the creation of characters, minigame and part of engine packages and related Swing windows, testing and Javadoc. The greatest difficulties of this part of the project consisted in creating the minigames functionality and integrate it in the game.

The project is present on GitHub as a private repository, anybody reading this document is kindly invited to request access to this repository to have proof ofwhat has been stated above and to see the effort put into this highly rewarding project.

A total of over 400 hours has been spent on the realisation of the project and both students were satisfied with the difficulty of the project and with the learning curve over this period.