# SENG201 Project Report

Authors: Lorenzo Fasano (Student ID: 34853558) and Jay Hamilton (Student ID: 84258116).

This report highlights the structure, testing, strengths and weaknesses of the application HeroesAndVillains created by Lorenzo Fasano and Jay Hamilton.

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 collaboration skills on a software development 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 Enum types 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 collectable objects in the Shop and HeroesSquad inventories, 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.

The implementation of a reliable system to organise Collectable items together with the implementation of abstract low-level classes was a key design feature that allowed a simple and modular design of the high-level structure.

A similar concept was applied in 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 implemented in the HealingWard 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.

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, it is also responsible of updating the HeroesSquad, Villains and World objects(An ArrayList of City Objects) while the game is running. This class also contains the main logic behind the serialisation of the games current status and the saving of the scores.

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(int) were particularly useful for speeding up the tests creation, to make sure that tests were run independently from each other and to test methods that use random events.

Only after thoroughly testing the low-level classes, was the higher-level functionality, such as the implementation of City and HeroesSquad classes, added and 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 was 55%. The modularity of the code helped ensure high test coverage, however, no Swing window was tested, in the next project more attention will be put in the discovery of new techniques to test the GUI components.

Overall 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 both partners made significant commitments to the project. Agreed deadlines were respected and the communication was consistent. The use of GitHub as a version control platform facilitated consistent 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 has been 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 game design and implementation of the architecture.

Lorenzo Fasano: My focus has been the creation of city, collectables and part of the engine packages and related Swing windows, testing and Javadoc. I have been implementing also the extra credit tasks involving classes serialisation. The most laborious process was 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. I was also in charge of making sure that the UI looked appealing and that the game logic worked as intended. The greatest difficulty of this part of the project consisted of creating the minigames and character abilities functionality and integrating 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 of what 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 it and with the learning curve over this period.