Andy Holden 97242455
Alex Liggett 69042786

SENG201 2018S1 Assignment

Heroes and Villains

Introduction

The specification for this assignment called for the creation of a 'Heroes and Villains' type game in Java with a Graphical User Interface, in accordance with a very detailed specification.

Game Structure

The game is based around a number of fundamental classes, namely the Hero, Villain, Team, PowerItems, Location and City classes. Some of these have a number of subclasses and/or are supplemented by enumerated types. As well as these there are some aggregating classes such as Villains and Cities used to group collections of subentities together. There are six classes for constructing GUI windows, some of which contain functional methods which were most neatly encapsulated within the window. Finally, there is a GameManager class which is used to hold the current state of the game, and from which GUI windows are launched.

Hero Class

The Hero class contains information about the individual heroes controlled by the user. An enumerated type HeroType is used to define specific kinds of heroes and their attributes, and an instance of this is set as an attribute in each instance of the Hero class upon construction.

Villain Class

Villains differ from heroes in that while different kinds of villains exist, there are no two the same in any game. This meant it was possible to entirely define the villains within an enumerated type, and an individual value from the type is instantiated as the villain in each City. The Villains class is used to aggregate the villains to ensure that the same villain is not allocated to two towns and that a villain is allocated to each town.

Team Class

The Team class is used to collect the Heroes and their inventory together. Heroes, PowerUps and HealingItems are held in ArrayLists, and a large number of supporting methods are available to interrogate and edit these.

PowerItems Class

PowerItems is a parent or super class containing generic attributes and methods related to such things as cost and amount carried of each specific type. From this are extended the PowerUps and HealingItems classes, for each of which in turn an enumerated type is used to set many of the (mostly immutable) values specific to each type of item.

Location Class

Location is a parent or super class containing the generic attributes of all locations on the map. An enumerated type is used to specify the individual types of locations available. The Location class is extended by 5 subclasses being the HomeBase, Hospital, Shop, PowerUpDen and VillainsLair classes. Each of these in turn hold functionality specific to location of that type.

City Class

City is a class used to set up and maintain the state of the map for each city. Locations on the map (N, E, S, W) are specified by way of an enumerated type and these are randomly mapped to locations from the Location

Andy Holden 97242455
Alex Liggett 69042786

class. An aggregating class Cities is used to contain the run loop for the main part of the game, and to move from one City to the next as the game progresses.

WelcomeScreen and TeamBuildScreen

As the names suggest these are used to set up and run the initial welcome screen, and the screen at which the user specifies which types of Hero they wish to have in their Team. The WelcomeScreen is launched from the GameManager, and a call to close this screen from within the GameManager launches the TeamBuildScreen. These two screens rely almost entirely on methods from the Hero and Team classes.

CityScreen

The CityScreen shows the map of the city and a panel representing the current location within the map. Panels are shown or hidden as required as the user moves around the map. CityScreen is initially launched from the GameManager, however it is closed when the player enters the VendorScreen to buy items or the BattleScreen to battle the villain. On exit of these screens a new CityScreen is called as required, elements of the game state being held in the GameManager or passed into the screen class as required.

VendorScreen and BattleScreen

These classes are used to set up and run the interface for purchasing items and battling the villain respectively. As well as GUI definition each of these contains a large amount of functionality previously held in the Shop and Battle (eliminated from final game) Classes, as the methods are closely tied to the functioning of the GUI. These screens are called from and return to the game manager.

FinalScreen

The FinalScreen is launched from the GameManager when the user either wins or loses the game, and also contains a small ‘Credits’ section. A new game can be launched from the FinalScreen.

GameManager

The GameManager class contains the main() method used to run the game, and is used to maintain the state of the game as the user transitions from one screen to another.

Unit Testing

Unit testing was undertaken on most of the fundamental classes of the game. In the command line version of the game it was found difficult to test a lot of the methods as they outputted directly to the console and often had random output. Many of these methods were found unsuitable in the GUI version of the game in any case, and were supplanted by new methods often within the GUI classes. This accounts for the low test coverage figure of 23.4%.

Highlights and Lowlights

The team working on this project worked successfully together and produced an application that meets the specification. In addition the team had a lot of fun devising the premise of the game and the various events that crop up along the way. Testing was an area where a lot of difficulty was experienced, and this would be an area from improvement in future projects. The project might also benefit from some additional refactoring to remove some methods from within the GUI classes, however this was not possible within the time allotted to complete the project.

Andy Holden 97242455
Alex Liggett 69042786

Individual Contributions

Because one team member is a part time student, face to face time during the project was of necessity minimal. The team first met on 3 April 2018 and after that held regular weekly meetings to discuss issues, resolve problems and allocate future work. In between times work progressed steadily, and immediate issues were resolved via email.

Andy Holden - 50%

Andy was responsible for the Villain, City, Location and Sounds classes and all related aggregating classes and subclasses. He also developed the main runtime loop for the game and the entirety of the CityScreen and VendorScreen. Andy was also the team's primary problem-solver.

Alex Liggett - 50%

Alex was responsible for UML modelling, the Team, Hero, PowerItem and all related subclasses. He developed the WelcomeScreen, TeamBuildScreen, BattleScreen and FinalScreen GUI classes and was responsible for final editing of the Javadoc and the bulk of the unit testing.

Final Thoughts

This assignment represented an enormous amount of work. Both team members have put a considerable amount of time into the project. While the team had a lot of fun working on it, a more modest specification might have made time to develop a more polished finished product (notwithstanding Parkinson's Law).