

SENG201-18S1 Assignment – Final Report

Jesse Sheehan 53366509
Manu Hamblyn 95140875

Application structure and design choices

Early planning was comprised of a review of all specified game components and their interactions, then identifying candidates for classes as well as commonalities between game components.

A significant early decision was that (almost) each component of the game should be in its own class, since an object can hold other objects. For example the object 'Inventory' can hold more than one 'Item'. The primary reason for this decision is to allow the core game logic to be decoupled from the GUI. Thus the actual game is not run by the GUI and there is no game logic in the GUI itself, rather the GUI presents an interface between the core game and the user and only calls/updates core functions as needed.

The design pattern used in the GUI is MVC. With the core game classes being the models and controllers, and the GUI being the view. Within the core game classes, both exceptions and returns were used to propagate information between the models and controller.

One example of commonality and therefore re-use between game components is that 'Hero', 'Villain' and 'InnKeeper' are all a kind of 'Character', so each of these is a class in its own right. The abstract class 'Character' is used to model things in common to all characters, and this is inherited by 'Hero', 'Villain' and 'InnKeeper'. A further example is that a 'Team' has an 'Inventory', as does a 'Shop'. Hence 'Inventory' can be used by both a 'Shop' and a 'Team'. Additionally, 'PowerUp', 'Map' and 'HealingItem' are all a kind of 'Item'. Thus we have an abstract class 'Item', which is inherited by 'PowerUpItem', 'HealingItem' and 'Map'. This is then combined with the 'Inventory', which holds an 'Item' (regardless of which kind of item). Inheritance can be identified on the included UML class diagram 'uml_core.jpg'. Please note that to keep the class diagrams easily viewable not all relationships are shown.

Internally (within classes) arrays (as opposed to collections) have been used to hold multiple objects when the number of objects in the array is not often changing.

Test Coverage

The unit test process achieved 75.4% coverage of the *core* game packages, with an overall test coverage of 18.6%. Only the core classes were covered with JUnit tests. Thorough testing has been implemented on the majority of classes within the core game packages. GUI classes have been omitted from JUnit testing due to their size, complexity and being unsure in how to proceed with testing them. Additionally the text based user interface includes a number of implemented tests.

The GUI classes (and therefore, the core classes) were tested extensively by the developers and also several non-developers. Hence, both verification and validation testing has been performed.

Assignment retrospective

What went well?

Team commitment. Team communication. Project organization (storing files on Git).

What did not go so well?

The free lunch did not eventuate. Time management could have been handled better.

Improvements for next project

Overall: Do more work earlier.

While it's clear that using a text interface provides a way to get something running that is at least useable it consumed time that could have been spent directly on the GUI. So a future approach might be to start with a GUI interface and only consider text as a fall-back. Unit testing assists in this process by helping to check that each class, or functional group of classes, is working.

Agreed contributions

Manu Hamblyn 36%

Constructed / wrote a number of GUI panels and dialogs in the GUI packages, plus classes in the core game packages, as well as continuously updated the "strings.json" (information manifest) file.

Jesse Sheehan 64%

Wrote many of the core classes. Set up the GitHub repository. Designed the overall architecture of the project.

Final thoughts and feedback on the assignment

The assignment has consumed a significant amount of time not only in writing code but also planning and documentation. The earliest possible release of the assignment details has been greatly appreciated.

It has been an enjoyable challenge despite being rather time consuming.