Coffee Maker Quest 412

Austin Choi

CS 1699 – FINAL DELIVERABLE

-Summary-

For the final deliverable, I decided to write a test-friendly version of Coffee Maker Quest. The new version is called Coffee Maker Quest 412. The user interface of the new program is identical to the original game, but the code has been designed to allow easy testing of major functionality.

The Player and Room classes only contain primitive values, so I was able to easily check them using a few unit tests. Some of the class methods were so simple that I did not feel the need to write tests for them. For example, the goNorth() method in the Player method simply incremented the player’s location variable by one.

Of course, with the auxiliary classes designed in such a simple manner, the bulk of the game logic ended up being contained in the main class. To check the main class, I ran property-based tests to make sure that all the original program’s functions and requirements were satisfied. Each test checked a property that satisfied one or more functionalities in the original requirements list.

Overall, the combination of unit and property-based tests allowed me to quickly find any major bugs in the program.

-Issues-

One of the issues I ran into was the testability of certain requirements. One functionality, FUN-HELP, required that the player be shown a list of possible commands after hitting a key. In order to fully test this function, I would have needed to analyze string output from the program. Since the string output for the command list could change frequently in a real-world development cycle, writing an automated test would have proven difficult and inefficient. I concluded that FUN-HELP and similar functionalities would be covered by manual testing instead.

Another issue arose from the simplicity of the auxiliary classes themselves. The Player class only contains primitive values and does not directly reference any room state. The player’s location is recorded as a single integer, which would be used as an index in an array of rooms. The goSouth() and goNorth() methods simply decremented and incremented this value. While this simplicity allowed for concise code and easy unit testing, it also made the class a little dangerous to work with; additional constraints needed to be written in the main program to make sure that the player could only go north or south in rooms with appropriate doors.

-Assessment of Quality-

Overall, Coffee Maker Quest 412 is a great improvement over its predecessor. The code base was written with testability in mind, so most of the program’s functionality is easily checked by automated tests. There is even an automated test to make sure that the player can actually win the game on a specific level (due to budget cuts though, we only have one written so far). This means that we would only need to manually test a few parts of the program to make sure that it is functioning normally before each release. Since automated testing is much faster and cheaper than manual testing, the new program is very easy to maintain and sweep for bugs. As of now, all of the automated tests have passed with flying colors, so I am confident that Coffee Maker Quest 412 is ready to ship to the masses.

-Location of Code-

https://github.com/M94/CS1632-D6