University of Pittsburgh

CS 1699 – DELIVERABLE 2: Unit Testing and Code Coverage

Test of Coffee Maker Quest

Rinat Gatyatullin, Frank Torchia

CS 1699

Bill Laboon

17 February 2015

Some of the difficulties we encountered when writing unit tests was relating to code coverage. We attempted to get as closely as we could to 100% code coverage but in the process we were left wondering whether or not we were stepping into the realm of exhaustive testing. It seems to us that there is a fine line between reaching sufficient code coverage and testing an application too exhaustively.

Another problem we encountered was with the ambiguity of return value for some of the functions we tested. For instance, the doSomething function returns 0 both when the user enters a successful command and an invalid command. This made writing unit tests for successful and wrong commands not as clear as we initially assumed. While this might have been an attempt to make us use mocks and stubs to be more familiar with them, we foresee encountering these problems in the future. We expect to encounter more ambiguity in return values and function actions when we are designing unit tests for various applications.

We had three tests fail: testDoSomethingn, testDoSomethingH, and testDoSomethingh. In testDoSomethingn and testDoSomethingh we were testing the function support for lower case input characters. The test for n failed because the equals statement used for the ‘N” command was simply equals and not equalsIgnoreCase, which was used for the other commands. Meanwhile, both of the other ‘H’ command tests failed because there was no code to support the help function. We were testing for a certain help output message, but because there was no code to do that it just printed out “What?”

Our code is located at: <https://github.com/rtg8/deliverable2_repo>





