**PROFESSOR WUMPUS**

CS 1632 – Deliverable 1:

Test Plan and Traceability Matrix

Group Members:

Anthony Poerio ([adp59@pitt.edu](mailto:adp59@pitt.edu))

Brandon Hedges ([bjh86@pitt.edu](mailto:bjh86@pitt.edu))

**INTRODUCTION**

Our chief concern while testing Profesor Wumpus was that we would *only* be able to do black-box testing, given that we did not have access to any of the source code, or knowledge of the program’s internals. Despite this concern, black-box testing a game like Professor Wumpus proved beneficial, because—as testers—we were able to experience the game exactly as end users would. And so, we were able to find defects naturally, in many of the same situations a real client would, rather than constructed cases based on our own knowledge of the code base.

Put another way: testing Professor Wumpus without any knowledge of the how it was made was a challenging, but useful, exercise because we were not incentivized to avoid—or attack—any particular areas of the requirements based on our knowledge of the system. The process was much more natural.

Having said that, testing Professor Wumpus was a mixed bag, with requirements that ranged from mildly to very difficult to test. Moreover, black-box testing was especially challenging for several of the game’s requirements that depended on either **a)** randomness, or **b)** knowledge of where game characters or items may be at any given time. Thus, while black-box testing Professor Wumpus, had necessarily relied only on what was printed to the console, which itself could be inaccurate (I.e. – itself subject to defects).

One of the most difficult requirements to test was that the print statement indicated that the TA was would be printed when the student was North, South, East, or West of the TA. This is difficult because the only indicator the user has of the location of the TA is the string that the TA is nearby. So in order to test whether the string printed, the movements of the TA with a specific input for the seed had to be mapped by intentionally running into the TA and losing the game many times. This had to be done before it could be reliably known that the TA was actually North, South, East or West of the student and the test case could be written.

Another challenging test was writing the execution steps for the testing the wall to wall boundaries. Part of writing these execution steps is preparing an exhaustive route that can test all the walls and avoid the TA in the process. Once again the TA’s position must be mapped before the test can begin and an appropriate seed must be found that allows the user to visit all the walls without interruption. This is a long and tedious process during black box testing.

Black box testing would be easier if there was some way to know where the Professor and the TA are, as a tester, otherwise it’s impossible to be 100% certain that the TA moves randomly, for example. Randomness needs to be statistically proven. Not definitively knowing all the TA’s moves makes it impossible to verify via black box testing. Assumptions can only be made based on what can be reasonably tested. White box testing is a start and has its place, but a full test plan should be a combination of white or grey box to be sure that testing is optimally accurate and the requirements are met.

Testing edge-cases for Professor Wumpus was comparatively simple, next to testing the elements that relied on some degree of randomness. For edge-cases, we focused on the requirements which specified user-input, and then tested values right at either end of the acceptable range. This proved fruitful, and we were able to find a defect using this particular method.

All in all, we are thankful for the opportunity to test a program like Professor Wumpus, under these circumstances. It has illustrated the positives and negatives of black-box testing, and provided a general understanding of the testing process, and why manual testing is not especially desirable—it’s tedious! We believe that with automated testing, all of the same actions could be done much more efficiently, and so that’s the chief area of improvement we would select as QA analysts for improvement to, and expansion of, this test suite.