Elias Rademacher
CS 362
6/9/15

My Software Testing Experience

Write a test report, in pdf format, describing your experience testing Dominion. Document in detail, including code coverage information, the status and your view of the reliability of the Dominion code of at least two of your classmates. This file is submitted as testreport.pdf in your dominion directory. Your test report should have a minimum of 1,000 words of text.

I have learned many things about software testing and about programming in general over the last ten weeks. Through all the tests I wrote for Dominion, I got much better at tracking down bugs in a program. I got three main takeaways from this: know the code, read the code, and don't trust your tests.

Huangma's implementation of dominion.c is the most reliable of the ones that I tested. The results of the tests run on Huangma's code can be seen in ./testResults/ huangma. There were two main tests that I ran on it: a suite of 8 unit tests, and testdominion.c which produces a log of the game being played that I used for differential analysis with my dominion.c. I used gcov to gather coverage information for the unit tests. The unit tests covered 8% of the lines in dominion.c and executed 11.3% of its branches. Based on coverage information that I got from gcov at a later time, testdominion.c probably covered about 65% of the lines in dominion.c.

Huangma's implementation of dominion.c had a bug such that it appeared that the players did not all start the game with 3 victory points. This was actually because of a mistake in the function that returns a player's score, scoreFor(), on line 444 of dominion.c. The loop that looks through a player's deck was iterating up to the size of the player's discard pile instead of the size of their deck. I found this bug by analyzing the output of testdominion.c. It is a bug that I had actually fixed earlier in my own implementation of dominion.c. This bug was easy to find and easy to fix. Other than this fault, huangma's dominion.c worked well. However, given the small size of my tests, it almost certainly contains many more bugs.

Perhaps the least reliable implementation of dominion.c that I encountered was that of luans. I ran the same tests on luans' code that I ran on huangma's code (described above). luans' dominion.c had the same bug as huangma's, except that luans' also caused a segmentation fault. I was not able to hunt down the cause of this seg fault. This seg fault is a very serious bug, because it renders the program completely inoperable. Also, the fact that it is relatively well hidden adds to its severity. Fortunately, I could reproduce it every time a ran the tests, so I could have tracked it down eventually. If it had only cropped up now and then, but still frequently enough to be a nuisance, then it would have been even worse.