Logan Wingard

11/16/17

CS_362_400_F2017

Assignment-5

Bug Reports

Through testing, I was able to pinpoint where the bugs were in smithy, Adventurer, and village. I found through my random tester that the smithy would never work as expected and looking into the code, found a 4 instead of a 3, for number of cards to draw in smithy, found something is causing the function playVillage to work as intended, but not discard it correctly, and playAdventurer adds an extra card.

I was able to pinpoint a bug in scoreFor using my unit tests resulting in player 0 not getting the points they should for victory cards.

I found in getWinners a bug causing player 0 to win even when player 1 has more vp.

Unit test 1 and 4 passed and found no bugs in getCost or gainCard

Test Report

Random Tests:

Randomtestcard2 tested the playVillage function. It covered 100% of the function.

Randomtestcard1 tested playSmithy and also covered 100% of the function.

Lastly, randomtestadventurer tested playAdventurer and covered 100% of the functions and all 12 branches.

It is safe to say these random tests covered enough of the code to thoroughly test the functions. My random tests also will check if the function worked as intended, which village "seemingly" did.

Unit Tests:

The unit tests ran until unittest2 caught a bug in scoreFor. It did not cover as much code as I would deem adequate, though it still managed to catch the only two bugs in these functions.

Debugging

The Random tests made the bugs in Adventurer and Smithy apparent right off the bat by showing me that none of the tests succeeded. Firsts thing I look for when a function doesn't give the

expected output is an off by one error. Luckily, this was the case for both. I was able to fix the for/while loops so they worked as intended. For finding the bugs in Village, I decided to use the GDB debugger. I was able to watch the program execute, however I was still unable to pinpoint where the bug was, or even that there was a bug. Due to the nature of the bug, it almost went undetectable until I looked at the function and noticed a 1 instead of a 0.