Logan Wingard
10/22/17
CS_362_400_F2017
Assignment-3

BUGS

Unittest1: Tested gainCard. No bugs

Unittest2: Tested scoreFor. Found a bug that caused the for loop that counts vp in a deck to iterate the number of times as cards in the discard pile instead of in the deck. (i.e. if there are 3 curses in the deck and no cards in the discard, the curses wouldn't get counted.

Unittest3: Tested getWinners. Found a bug that would return player 0 as the winner even if player 1 had more VP.

Unittest4: Tested getCost. No bugs.

Cardtest1: Tested smithy. Found a bug that causes the player to gain 4 cards instead of 3. (My bug added in an earlier assignment.)

Cardtest2: Tested Adventurer. Found a bug that causes a segmentation fault, caused by looking for a card in a slot in the hand greater than the number of cards in said hand.

Cardtest3: Tested Council Room. Found a bug that caused the player to draw every card that the other players should have drawn.

Cardtest4: Tested village. Found a bug causing the player to only gain 1 action instead of 2.

UNIT TESTING

According to unittestresult.out, 17.82% of all of dominion.c was executed throughout my tests. 17.82% of 578 lines total is 103 lines of code. Looking through unittestresult.out, I can see which lines have and have not been executed. Through this I can see gainCard had successfully executed all but 2 lines of code, which tells me I could have done a better job at branch coverage. I also see that due to the asserts in my tests, village(which aborts on the first test of the unittest) hardly got tested at all, though it successfully identified the bug.

UNIT TESTING EFFORTS

For unittests 1-4, I wanted to cover as much of the code as possible with my tests. For example, on gainCard, I had three for loops, 1 for each pile of cards (hand, deck, discard) however, because I did not test more than the number of cards in the supply count, it never executed the code in that if statement. So for my future tests, I wanted to make sure to cover every single line, like in getCost which tests the cost of every singly card implemented.