I have learned a lot things throughout testing this dominion code. I feel like have gotten much better at recognizing bugs in code along with my abilities to actually find a bug through writing tests. Coming in I had some experience with tests but had never learned how to make tests that did anything other than make sure nothing crashed. I had also never written tests in C so that was also a new experience.

Getting into writing the first test, I did not really know what I was doing. I got some basic knowledge on how assert works in C and then just tested some easy methods. I ended up testing initializeGame, shuffle, buyCard, and isGameOver. I was actually able to find an error in initializeGame although it was fairly minor. It was just an issue where every player was not getting a hand during the set up. I couldn't find errors in the other ones even though I am sure there were some.

The third assignment I thought I was being much more thorough. I wrote some random tests which generated a totally random hand/deck/discard for a totally random number of players and then had a random one of those players get the cardEffect of an adventurer. Of course, I didn't actually check for anything in this so all I could say was that adventurer probably would not crash the game. I was actually able to make it crash during the random occurrence that the player had less than 2 treasures in their discard/deck but I just added an extra component to the test to make sure that did not happen so that I could get the rest of the tests to run. I actually found an issue by just looking at the adventurer code after I had finished making the tests. I could have made my tests much better and more efficient if I had just made some simple checks like making sure the player had the same cards as before and making sure the player ended up with more money. The other card I tested was the council room. I wrote these tests almost identically to how I wrote the adventurer test. I ended up not being able to find any errors in the adventurer test.

Assignment 4 I did somewhat poorly. The game played through fine but I didn't actually realize that some of the cards being played were causing major problems with the game state. The end result of it was the player 1 ended up losing several cards for no reason. I was able to catch the bug later in the final project however. One of the big problems was that I was not outputting enough information. I don't know why the assignment review said I should print the player's treasure count since that is easy to find with the hand that is printed every turn but one thing I should have done was to print the deck and discard every turn. I actually added that extra stuff in later when working on the final project. I also should have printed out the player's hand/deck/discard every time they played a card also.

Lessons Learned:

Always plan out the tests before writing them. A lot of the test I wrote I just kind of winged it. I did not have a plan and it ended up really hurting both my work efficiency and the quality of my tests. If I had actually looked at what I was testing and thought about what was important to be checking when writing the tests I would have had much better tests. Another important thing to plan out is how you are going to actually set up a game state that is appropriate for actually testing that particular part of the code. I just ended up randomizing almost everything and probably wasted a lot of resources on random tests that were literally impossible to replicate in a real game.

Make sure your tests are actually done right. A lot of my tests seg faulted during the early stages and I spent a lot of time looking for errors in the dominion code to figure out why. In end however, it almost always ended up being a problem with my own test. I suppose it is to be expected though. It is incredibly

rare to write good code that is bug free on the first compile and there is no reason that tests should be any different. Tests are just as susceptible to bugs as other code.

Overall, I learned quite a bit while trying to figure out these assignment but I absolutely did not have a good time. Testing is not something that I find to very fun especially when testing code that should have just been scrapped to start with. Also, while the lectures provided some interesting information, I did not think any of topics were helpful when doing the testing assignments especially since most of the lessons were pretty much just basic common sense.

Reviewing other dominion code:

Cronisej:

Tests ran: unittest1.c unittest2.c unittest3.c unittest4.c cardtest1.c cardtest2.c cardtest3.c cardtest4.c randomtestadventurer.c randomtestcard.c testdominion.c

Bug: In the initialize game function where player 2 would not receive a starting hand.
Bug: village card is not providing the correct number of actions
Bug: great hall is failing and not providing the player with the correct number of cards
Bug: score is calculated incorrectly
Bug: sea hag trashes opponent's cards

Code coverage: 52.69%

With just over half the code covered, I quickly found 5 bugs in conisej's dominion code. I would say that at this point the code is very unreliable. Much more expensive testing would need to be done to determine how many more issues there are in the code.

Colliell:

Tests ran: unittest1.c unittest2.c unittest3.c unittest4.c cardtest1.c cardtest2.c cardtest3.c cardtest4.c randomtestadventurer.c randomtestcard.c testdominion.c

Bug: smithy does not provide the right number of cards
Bug: initialize game does not provide player 2 with a starting hand
Bug: adventurer sometimes causes the player to trash cards
Bug: sea hag is still trashing cards
Bug: score not being calculated right
Bug: lots of cards getting set to null values

Code coverage: 51.12%

Found even more issues with this code. One of most worrying things is the fact that tons of player's cards are being set to null throughout the game. This could potentially be very difficult to track down and fix. Again, I would of course say that the code is very unreliable.