Assignment 3

Bugs

By executing all the unit test and card test functions, I encountered a few bugs on the way. For unit test 1 and unit test 3, I introduced a minor bug that still allowed the unit test to pass. Unit test 1 is doing invalid and valid testing of the buyCard function and Unit test 3 is testing the shuffle function.

Unit test 1 function's bug:

```
//Bug is -2 for the supplyPos parameter
 a = buyCard(-2, &Game);
 assert(a == -1);
```

Unit test 3 function's bug:

```
 //Bug is -1000 for the array deckCount in struct Game
 Game.deckCount[player] = -1000;
 a = shuffle(player, &Game);
 assert(a == -1);
```

All the card test functions contained both tested the refactored functions, that contained bugs, and the cardEffect function which contained no bugs. The refactored functions of adventurerCard(...), council_roomCard(...), smithyCard(...), and villageCard(...) all contain bugs that personally introduced to make the card effects different than what they are supposed to be.

Unit Testing

The results of the code coverage for the unit test and card test were 100% line executed with all of them passing. Looking at the results of the entire dominion.c, I get:

File 'dominion.c'
Lines executed:20.21% of 564
Branches executed:22.78% of 417
Taken at least once:13.91% of 417
Calls executed:10.53% of 95
Creating 'dominion.c.gcov'

Analyzing the results more lets me see which functions were not executed, maybe due to bad code or bugs. For the unit tests, the buyCard function and the endTurn function are never called in the dominion function. The shuffle function is called 2 times and executed 94% of it's blocks executed. The isGameOver function, despite being called two times in dominion.c, is not called or executed in the gcov line examination. This could be due to the gcov not picking it up or a bug that is preventing it from being called. For the card tests, a similar problem was encountered as the refactor functions of adventurerCard(...), council_roomCard(...), and smithyCard(...) was said to have 0 calls and 0% blocks executed, despite the three card functions being called in the cardEffect function of dominion.c. This is most likely due to the bugs that were purposely introduced in those functions or an unknown bug that I'm unaware of. The refactor function villageCard(...) was called 1 and had 100% of it's blocks executed.

Unit Testing Efforts

All of the unit and card exectubiles test for crashes(before any change) and the state of the game(after changes). For unit test 1 and 3, which is the testing of the buyCard and shuffle functions, I executed two separate test, one that was designed to fail and one that was designed to pass. Designing a test for the buyCard function was simple, but it was more difficult to design one for the shuffle function. Unit test 1, 2, and 4 involved manipulating the members of the struct Game before testing the functions. For all the card test, I used the cardEffect function for the invaild test and the refactor functions for the valid test.