Assignment 5

James Palmer

Using teammate ogunsajb dominion file to test.

Refactoring My Tests:

In the initialize game function it did not draw 5 cards into the player's hand like the game should, so I manually fixed that mechanic. In the fullDeckCount function it was not counting the full deck, it was only counted if there was a certain card in the deck (i.e the card I was testing), so I fixed that by just having a count variable and adding deckCount, discardCount, and handCount to my adventurer file.

It was also brought to my attention from a teammate whom identified that my adventurer test card does not get added to the played card pile. I added an if statement that determines if the card in currentPlayer's hand at handPos is the adventurer card, and if it is it adds it to the played card pile. This was to account for instances where discarding the card might be handled by the function that calls adventurerCard(), or when the card is not actually in the players hand. This came up when testing smithy and Great Hall because both instances discarding the card and/or playing into the card pile were not functioning correctly, therefore, wasn't receiving proper testing output.

Final and major changes to my refactoring tests are as followed: calling functions of other files correctly. In short, function calls were different in ogunsajb than mine. This included main dominion.c file calls being incorrect calls to pull information. Another objected appointed to me was my use of incorrect implementation of resources. I used Visual Studios for my testing this go around because my knowledge of gcc is very limited and continuously ran into errors that seemed very time inefficient. In short, my files were not very maintainable and need a lot of work going forward in future classes and my career. My coding need a lot more work from the ground up to be at a professional level because it was too volatile in testing and many things had to be changed (such as variable names, basic coding concept).

Ogunsajb informed me that my testing parameters needed to be more focused (i.e. Great Hall draws a card, +1 action and discard) rather than testing every aspect and how each card combats each other. In short, I removed some fat from some of my unit testing to focus on only the specific card I am testing. Also, he taught me on how to look back at past repositories and see the changes he made from Assignment 2 to Assignment 4 and the bug differences there. In doing so, this allowed me to pinpoint the his bug a lot faster than manually looking at all 1400 lines of code in Dominion.c.

Bugs In TeamMates Code:

Teammate identified Bug smithy:

****

Error in playsmithy() where smithy call draws zero cards at a time. This is indicative of line 831 where there is only a playsmithy

call and not a forloop to start a drawing card process. This bug continues in case smithy because there isn't a discardCard call either.

I found this bug when attempting to insert a draw command directly into my testing files which showed that no cards have been discarded either.

****

```
** Card Test 2: Smithy **

Expected value: 5, Result: 0

Expected card count: 3, Result: 0

** End Card Test 2: Smithy **
```

Teammate identified Bug greathall:

****

Error in great_hall() where great hall card doesn't draw a card, doesn't add an action nor removes the card from hand. This is shown on line

859 that the case great_hall is blank, only a call to playGreat_hall. In short, like playsmithy() call, the lines were deleated rather than tampered with.

When calling the great_hall command, noticed that I wasn't adding a card to my hand. With forcing a card to my hand, I wasn't allowed to play another action.

This second finding forced the idea of my unittest to force both +1 card and +1 action lead to great_hall never being discarded from the hand. At somepoints this created

an infinate loop because the players hand could never reach 0 and "pass the turn" because great_hall was never discarded.

****

Bug Fixes:

I changed the trash flag that is passed to the discardCard function for the Great Hall card originally, so I changed it back. I made the for loop for the Smithy card (i=0; i<=3; i++) however it should have just said i<3, so reverted those changes. For the Village card I changed it from saying state->numActions + 2 to saying state->numActions = state->numActions + 2.

In the initialize game function it did not draw 5 cards into the players hand like the game should, so I manually fixed that mechanic.

In the fullDeckCount function it was not counting the full deck, it was only counted if there was a certain card in the deck (i.e the card I was testing), so I fixed that by just having a count variable and adding deckCount, discardCount, and handCount to my files.

Also, I ran into compiler issues with using "Remove secure warnings (_CRT_SECURE_NO_WARNINGS) from projects by default in Visual Studio." In short, I added #pragma warning(disable:4996) and _CRT_SECURE_NO_WARNINGS" into the Preprocessor Definitions white box to remove these errors. I believe this is a VS compiling error only for older versions.

Last fixes to my testing: I returned my trash flag in the discardCard() function to 0 to allow cards to be discarded again. This was changed in my testing to see if Great Hall call was directly effecting discardCard() function call or not.