Stephen Townsend
CS 362-400
10/22/17

Assignment 3

Bugs

When attempting to compie dominion with gcc -o unittest(#) dominion.c rngs.c unittest(#).c I am getting a message saying:

> /tmp/cckkT5IJ.o: In function `shuffle':
>
> dominion.c:(.text+0x678): undefined reference to `floor'
>
> collect2: error: ld returned 1 exit status

I am not sure what this bug is yet. As a result, I needed to remove the floor from the code to get it to compile and run.

Unittest1:

When running my unittest1 for buy cards, I am having a bug that when buying a card with the correct amount of coins I get a failed message. When I buy the card with more than enough coins it still fails. There must be something wrong in the dominion code or where the return value is.

Unittest3:

When testing the ability to update coins, my tests are failing. I have sent 10 coins into my hand for copper, silver, and gold. However, the total is not updating when I am checking if it equals 10, 20, or 30 respectively.

Cardtest2:

For the Adventurer, there is a bug that is causing a segment fault when it runs. This is most likely due to the fact that there were bugs that were introduced into the system. I attempted to correct this issue, but there seems to be something deeper that I cannot find. I removed code in my unit test till I only had two printf statements. Only the first statement would print out and then the test would seg fault. I am unsure why this is occurring.

Unit Testing

Unittest1:

> Lines Executed: 86.96% of 23

Unittest2:

> Lines Executed: 76.72% of 116

Unittest3:

> Lines Executed: 87.88% of 33

Unittest2:

> Lines Executed: 83.33% of 12

Stephen Townsend
CS 362-400
10/22/17

Cardtest1:

Lines Executed: 84.62% of 39

Cardtest2:

Lines Executed: 00.00% of

Cardtest3:

Lines Executed 84.00% of 25

Cardtest4:

Lines Executed: 84.00% of 25

It looks like the reason I have lower coverage in the 80% is because on my tests I have if else statements. They are for if true else false. As a result, only one of these will be used during running and thus the coverage does not touch on those sections.

Unit Testing Efforts

For creating unit testing I needed to look at what each function or card did. I decided to make tests that would check:

Card:

If a card was added into the players hand.

If the player played the specific card.

If the specific card was discarded.

If there were any extra actions added.

If there were any extra cards that were added to the players hand.

Made sure that the deck count was similar by subtracting a specific amount to the test structure.

Function:

I tested the functionality of each function. For example, update coins I put copper, silver, and gold in an array and sent it to the function. I then checked the result to see if function updated the coin count for each instance. For whose turn I sent in two different players to see what I would get back with the call. For buy cards, I used three different conditions. No having enough coins to buy a card, having enough coins, and having enough coins, but no potential to buy. Overall, I feel that I put the functions through the condition they were designed for.