Long Le
CS 362
Instructor: Jaki Sharief Shaik
Assignment 3 - Unit Testing

BUGS

Functional Unit Test Bugs:

No bugs were found the the 4 chosen functions: fullDeckCount, shuffle(), isGameOver(), and drawCard(). All of the functions ran as expected.

Card Unit Tests Bugs

Adventurer Card: The effects are to search the deck for 2 treasure cards add add it to the deck.

1. More than 2 cards are drawn when there are more than 2 treasure cards in the deck.
2. During a check to see if other cards are counted as treasure cards, the program segmentation faults.

Smithy Card: The effects are to draw 3 cards in the same turn.

1. The player does not draw 3 cards when smithy is played.
2. A different player draws 2 cards (instead of 3) when the user player plays smithy.

Treasure Map: The effects are that trashing 2 treasure map cards in hand will yield 4 gold to the player.

1. Even when only 1 Treasure Map card is in hand, the player increases the gold count into their deck.
2. The gold gained by playing Treasure Map, even after fulfilling the requirement of 2 Treasure Maps in hand, only gives the player 2 gold instead of 4.

Council Room: The effects are that the player should receive +4 cards and +1 buy.

1. Other players did not draw +1 card into their hand.

UNIT TESTING:

File 'unittest1.c'
Lines executed:60.32% of 63
Branches executed:91.67% of 24
Taken at least once:50.00% of 24
Calls executed:53.85% of 26
Creating 'unittest1.c.gcov'

File 'unittest2.c'
Lines executed:78.57% of 42
Branches executed:83.33% of 12
Taken at least once:58.33% of 12
Calls executed:69.23% of 13
Creating 'unittest2.c.gcov'

File 'unittest3.c'
Lines executed:75.00% of 40
Branches executed:100.00% of 12
Taken at least once:50.00% of 12
Calls executed:72.22% of 18
Creating 'unittest3.c.gcov'

File 'unittest4.c'
Lines executed:74.07% of 54
Branches executed:100.00% of 22
Taken at least once:63.64% of 22
Calls executed:68.18% of 22
Creating 'unittest4.c.gcov'

Unit testing for the 4 functions seem to have good coverage. Branch coverage for unittest3 and unittest4 is 100% . Unittest1 has 91.67% branch coverage and unittest2 has 83.33% coverage. Out of all of the unit tests, coverage for unittest2 can be slightly improved to increase confidence in the test suite.

Cardtest1 segmentation fault - no output can be shown.

File 'cardtest2.c'
Lines executed:91.55% of 71
Branches executed:100.00% of 30
Taken at least once:76.67% of 30
Calls executed:87.50% of 24
Creating 'cardtest2.c.gcov'

File 'cardtest3.c'
Lines executed:90.00% of 70
Branches executed:100.00% of 34
Taken at least once:76.47% of 34
Calls executed:83.33% of 18
Creating 'cardtest3.c.gcov'

File 'cardtest4.c'
Lines executed:89.29% of 56
Branches executed:100.00% of 24
Taken at least once:79.17% of 24
Calls executed:81.25% of 16
Creating 'cardtest4.c.gcov'

Coverage for the card tests are very good. With the exception of cardtest1, the other 3 card tests have 100% branch coverage. This means that the tests were written such that all of the branches were executed.

File 'dominion.c'
Lines executed:49.05% of 577
Branches executed:59.62% of 421
Taken at least once:41.33% of 421
Calls executed:31.82% of 110
Creating 'dominion.c.gcov'

Running the file reveals that the branch coverage is executed only 59.62% of the time. This means that at most, my test coverages are only covering 59.62% of the entire code. While more coverage does not necessarily define a good test suite, a good test suite should have good test coverage. Since the dominion.c code only covers roughly 60 percent of the code with every run, improving good test coverage could mean attempting to have as much coverage as possible for the individual functional tests.

UNIT TESTING EFFORTS:
Admittedly, I misunderstood the assignment the first time around and had to redo most of my work, last minute.

unittest1 (fullDeckCount): I tested the counting mechanism for this function. I only used 1 type of card and made sure the initial count is accurate to what was initialized. Next I drew 3 cards and made sure the function could accurate count the updated piles. Finally, I added 2 cards to

the discard pile and verified if this count function is accurate. Improvements on this test could be to expand the test to count the different types of cards, add in edge cases such as zero cards, 1 card, or many cards.

unittest2(shuffle): This was a simple test I made to see if the player's deck is shuffled. I tested with a deck of 4 cards, with different cards in each slot. The test was to check if any 1 card was out of the original order. If it was, the test is considered successful. There are many other improvements that could be made, such as repeating the same test many times, with different cards. I could also increase the size of the deck to make sure the function works with a higher range of cards. Improvements to edge cases could be made, such as checking whether there is only one type of card in the deck.

unittest3(isGameOver): This function verifies end game states. The first was to verify if the game has run out of its supply of province cards. Next was to verify if the game has run out of 3 supply piles. Finally, the function verifies that if the 2 end game conditions are not met, the game will continue. Improvements could be made by checking all of the different supply piles for emptiness.

unittest4(drawCard(): This function tests if the player's deck has decremented, the player's hand is incremented, and if the discard pile is shuffled into the deck if the player is out of cards to draw. Improvements could be made by also verifying the effects of other functions used within drawCard() is also activated. One example is to make sure the discard pile was actually shuffled before being drawn by the player.

cardtest1(adventurer): This tests how many treasure cards are drawn, up to 3 cards. It also tests whether any other cards are counted as treasure cards and drawn in place of treasure cards. An improvement to the test could be to check up to 10 treasure cards in the deck, and make sure only 2 treasure cards are drawn.

cardtest2(smithy): This function tests how many cards were drawn by each player, whether the cards were drawn from the user's deck, and whether or not the state of the victory and kingdom cards were changed.

cardtest3(treasure_map): This function tests the effects of 1 treasure map card in hand vs. more treasure cards in hand. It also checks how much gold is added to the player's deck.

cardtest4(council_room): Checks how many cards were drawn by each player and whether the card was drawn by the player. I forgot to check if the player had +1 buy, so this would be a vast improvement to the test.

unittestresults.out

```
Result for running dominion unit tests:
unittest1.c:
TESTING fullDeckCount():

Test 1 - check initial count.
Should be deck = 5, hand = 20, and discard = 0

Test 2 - draw 3 cards.
Should be deck = 2, hand = 23, and discard = 0

Test 3 - add 2 cards to discard pile.
Should be deck = 2, hand = 23, and discard = 2


ALL TESTS SUCCESSFULLY PASSED!
File 'unittest1.c'
Lines executed:60.32% of 63
Branches executed:91.67% of 24
Taken at least once:50.00% of 24
Calls executed:53.85% of 26
Creating 'unittest1.c.gcov'

unittest2.c:
TESTING shuffle():

Test 1 - verify that shuffle will work

Initial deck:4 6 9 0
Shuffled deck:0 4 6 9
ALL TESTS SUCCESSFULLY PASSED!
File 'unittest2.c'
Lines executed:78.57% of 42
Branches executed:83.33% of 12
Taken at least once:58.33% of 12
Calls executed:69.23% of 13
Creating 'unittest2.c.gcov'

unittest3.c:
```

TESTING isGameOver():

Test 1 - verify the game ends if there are no more province cards
TEST PASSED

Test 2 - verify the game ends if there are 3 empty supply piles
TEST PASSED

Test 3 - verify the game continues if there are province cards
and if there are less than 3 supply piles empty.
TEST PASSED

ALL TESTS SUCCESSFULLY PASSED!
File 'unittest3.c'
Lines executed:75.00% of 40
Branches executed:100.00% of 12
Taken at least once:50.00% of 12
Calls executed:72.22% of 18
Creating 'unittest3.c.gcov'

unittest4.c:
TESTING drawCard():

Test 1 - verify cards are drawn from the player's deck to the player's hand
Testing with 5 cards in the deck
TEST PASSED
TEST PASSED
TEST PASSED
TEST PASSED
TEST PASSED
TEST PASSED
TEST PASSED
TEST PASSED
TEST PASSED
TEST PASSED

Test 2 - verify cards once deck is empty, discarded cards are shuffled into deck
5 cards are added to the discard pile
TEST PASSED

TEST PASSED
TEST PASSED

ALL TESTS SUCCESSFULLY PASSED!
File 'unittest4.c'
Lines executed:74.07% of 54
Branches executed:100.00% of 22
Taken at least once:63.64% of 22
Calls executed:68.18% of 22
Creating 'unittest4.c.gcov'

File '<built-in>'
No executable lines
No branches
No calls
Removing '<built-in>.gcov'

cardtest1.c:
cardtest1 segmentation faults - will not be run
File 'cardtest1.c'
No executable lines
No branches
No calls
Removing 'cardtest1.c.gcov'

cardtest2.c:
TESTING smithy card effects:

Test 1 - verify players' drawn cards
Player 0 card on hand count is: 0
TEST FAILED!
ERROR: Player 0 should have drawn 3 cards!

Player 1 card on hand count is: 2
TEST FAILED!
ERROR: Player 1 should have drawn 0 cards!

Test 2 - verify if the user's deck was drawn from

TEST FAILED!
ERROR: Player 0's deck was not drawn from!

Test 3 - verify if the victory card or kingdom card piles were not changed

Testing count of victory cards.
TEST PASSED

Testing kingdom card #0
TEST PASSED

Testing kingdom card #1
TEST PASSED

Testing kingdom card #2
TEST PASSED

Testing kingdom card #3
TEST PASSED

Testing kingdom card #4
TEST PASSED

Testing kingdom card #5
TEST PASSED

Testing kingdom card #6
TEST PASSED

Testing kingdom card #7
TEST PASSED

Testing kingdom card #8
TEST PASSED

Testing kingdom card #9
TEST PASSED

TEST FAILURES DETECTED

File 'cardtest2.c'
Lines executed:91.55% of 71
Branches executed:100.00% of 30
Taken at least once:76.67% of 30
Calls executed:87.50% of 24
Creating 'cardtest2.c.gcov'

cardtest3.c:
TESTING Treasure Map card effects:

Test 1a - only 1 treasure map in player's hand

TEST FAILED!
ERROR: original gold count is 0 and current gold count is 2
ERROR: NO GOLD SHOULD BE ADDED TO DECK

Test 1b - 2 treasure map cards in player's hand
TEST FAILED!
ERROR: original gold count is 0 and current gold count is 2
ERROR: 4 gold cards should be received into deck

TEST FAILURES DETECTED
File 'cardtest3.c'
Lines executed:90.00% of 70
Branches executed:100.00% of 34
Taken at least once:76.47% of 34
Calls executed:83.33% of 18
Creating 'cardtest3.c.gcov'

cardtest4.c:
TESTING Council Room Card:

Test 1 - verify players' drawn cards
Player 0 card on hand count is: 4
TEST PASSED

Player 1 card on hand count is: 0
TEST FAILED!
ERROR: Player 1 should have drawn 1 card!

Test 2 - verify if the user's deck was drawn from
TEST PASSED

TEST FAILURES DETECTED
File 'cardtest4.c'
Lines executed:89.29% of 56
Branches executed:100.00% of 24
Taken at least once:79.17% of 24
Calls executed:81.25% of 16
Creating 'cardtest4.c.gcov'

File '<built-in>'
No executable lines
No branches
No calls
Removing '<built-in>.gcov'

File 'dominion.c'
Lines executed:31.72% of 577
Branches executed:33.97% of 421
Taken at least once:25.18% of 421
Calls executed:18.18% of 110
Creating 'dominion.c.gcov'