Aleksander Bruske
CS362
10/28/2018

Assignment 3: Results Report

Unit Testing

Unit Test 1: **isGameOver()**
Code Coverage:
Lines executed: **89.47% of 19**
All Tests Successful

Unit Test 2: **gainCard()**
Code Coverage:
Lines executed: **87.80% of 41**
All Tests Successful

Unit Test 3: **numHandCards()**
Code Coverage:
Lines executed: **100.00% of 22**
All Tests Successful

Unit Test 4: **supplyCount()**
Code Coverage:
Lines executed: **100.00% of 10**
All Tests Successful

Card Test 1: **Adventurer Card**
Code Coverage:
Lines executed: **81.82% of 33**
Test Failure

Card Test 2: **Steward Card**
Code Coverage:
Lines executed: **93.88% of 49**
1 Test Failure

Card Test 3: **CutPurse Card**
Code Coverage:
Lines executed: **97.01% of 67**
All Tests Failed

Card Test 4: **Smithy Card**
Code Coverage:
Lines executed: **96.00% of 25**
1 Test Failure

To summarize, the test coverage for all unit tests was quite satisfactory. As can be seen from the results above, the Adventurer Card Test had the lowest code coverage at 81.82% of 33 lines of code. Some unit tests had 100% code coverage such as Unit Test 3 and Unit Test 4. There can be a few reasons why not all unit tests ended with 100% coverage. One reason, for example, could be that a few if statements (or conditionals) never fired off. Moreover, another reason could be that not all of the kingdom cards (only 10 cards were used for each unit test) were used for testing and therefore some functions were also not executed. Therefore, one way to combat this issue is to create more tests incorporating additional cards and functions in the dominion code.

Unit Testing Efforts

In this section, I will discuss the unit testing efforts for the four cards: Adventurer, Smithy, Steward, and CutPurse.

The **Adventurer** card was tested in the following manner: the original hand was checked and then the hand was checked after the turn. Since the card reveals cards from the player's deck until two treasure cards are revealed, two cards would thus need to be checked to test whether the Adventurer card was working correctly.

The **Steward** card was tested in the following manner: Since the card's action and purpose is to give the player three options, which are to gain +2 cards, gain +2 coins, or trash two cards from the player's pile, I believed it was necessary to create unit tests for each of these three choices. Therefore, the original hand before the action was tested as well as the hand after the choice was executed in order to determine if the three choices for the card were working correctly.

The **CutPurse** card was tested in the following manner: Since the card's action is to either (1) force the player's opponents to discard at least one copper (if they contain copper) or (2) force the player's opponents to reveal their hand if they have no copper in hand, it was therefore important to test the card so that this action was implemented correctly. Therefore, the unit test for this card was checked with two to four player games. The first player was given CutPurse and was checked first, and therefore the first player should have 6 cards (the opponents should all have 5 cards). The first player purposely plays the CutPurse card in order to test the result that the first player will then have a total of five cards and the opponents will each have a total of four cards (since a card is discarded for each opponent player unless they have copper, which in this case they did not).

The **Smithy** card was tested in the following manner: Similarly to the Adventurer card, the hand counts of Smithy were tested before and after the card was played. Since the official function of the Smithy card is to gain +3 cards when played, it was thus natural to test the hand count after execution to see if the total had 3 more cards than the original hand count (before the card was played).

Like the card unit tests, the dominion function unit tests were tested in a similar fashion. That is: to test the function is a manner that makes sense to the official gameplay of Dominion.

Bugs

Based on the unit tests for the Dominion functions, there were no bugs found for each of the functions and all tests passed successfully. However, the same could not be said for the card unit tests. All the card unit tests failed.

The bug found in the Adventurer Card Effect function was that an important int variable *z* was found to be always initialized to 1. This variable determines whether the game state executes its discard function. And thus, since the variable was always initialized to 1, this, of course, caused the Adventurer card not to work properly.

The bug found in the Steward Card Effect function was that the drawCard() function was not being called twice for choice 1 (where the player can choose to draw +2 cards). This could easily be seen in the unit tests as the player's hand count did not increase by 2 after selecting that option from the card.

An unintended bug was found in the CutPurse Card Effect Function. It was spotted in the unit tests rather easily because the player playing the cutpurse card had a sum of four cards (like the opponents), instead of having a sum of five cards after executing the card.

The bug found in the Smithy Card Effect function was that the card was adding four cards to the player's hand instead of just three. This was also easily spotted in the unit test since the hand count of the player that just used the Smithy card contain +4 cards instead of just +3 cards.