Erin Alltop
1/20/18
CS362 – Winter 2018
Assignment 2

Refactor

For this assignment we were tasked with refactoring the dominion.c existing code to both implement five cards with their own functions as well as introducing bugs to four of these five cards. Both Smithy and Adventurer cards were required to have bugs introduced.

Creating functions for the individual cards was a fairly simple process. In an effort to keep the existing code intact as much as possible, I simply edited the switch statements to call a separate function for the respective cards instead of having the function in its entirety within the switch statement. For example:

```
switch(card) {
    case smithy:
      //smithy statements
      return 0;
}
```

BECOMES

```
switch(card){
    case smithy:
        smithyCard();
        break;
}
```

In this way, the switch statement could retain its integrity and the functions would be called in an appropriate manner. Note that I changed the function names to end with "-Card" to not confuse the function with other parts of the code.

Bugs

The four cards that I introduced bugs for are Smithy, Adventurer, Village, and Baron.

Smithy

For the Smithy card, I altered the discardCard function to discard the card from the current player's hand into the next player's hand. The player draws three cards as intended with no obvious issues. It seemed to me like this is a somewhat subtle bug and might not be caught by a typical test suite that might not test for the size of every player's cards and discard piles after every turn. It is subtle enough to go unnoticed.

Adventurer

In the Adventurer card, I changed the card that is drawn from the top of the player's deck to be the card at the bottom of the deck. This is the bug I found to most dramatically alter the outcome of the game, even plunging players into the negative score. However, it still seems to be a relatively subtle bug that could be easily made, easy to miss, and difficult to catch in a test suite as well.

Village

For the Village card, the big I inserted was a bit simpler. Instead of giving the player two actions, it gives the player only one action. I feel like this bug would potentially miss a basic test suite, though if a player is actually playing the game instead of a computer simulation it would be more easily spotted.

Baron

Within the Baron card, I altered the function subtly. If the player did not have an Estate card to discard for four coins and if there is at least one Estate card to be gained, then the player will gain the card as intended, but instead of decrementing the Estate card deck by one, the deck is incremented by one instead. This is a subtle bug because it would not easily be caught by a test suite and additionally only happens some of the time this card is played, making it even more difficult to detect without a careful code review.