Assignment 2

liam beckman

January 27, 2019

1 Refactor

Implemented five cards in seperate functions from switch statement. This was done for the following cards:

1. adventurer
2. smithy
3. council_room
4. feast
5. mine

The existing implementations within the switch statement were used to create a new function implementation. In the case of the adventurer card, a new function called `myAdventurer()` was made with the required variables included in both the funciton definiton as well as the functional call. The refactoring for the four remaining cards took on similar patterns.

1.1 Original Implementation in Switch Statement

```
int cardEffect(...)
{
    // ...

    //uses switch to select card and perform actions
    switch( card )
    {
        case adventurer:
            // implementation of adventurer card...

        // ...
    }

    // ...
}
```

1.2 Refactored Implementation in Seperate Function

```
int myAdventurer(...)
{
    // implementation of adventurer card...
}

int cardEffect(...)
{
    // ...

    //uses switch to select card and perform actions
    switch( card )
    {
        case adventurer:
            // call to myAdventurer function
            myAdventurer(...);

        // ...
    }

    // ...
}
```

2 Bugs

2.1 Bug Breakdown by Function

2.1.1 myAdventurer

Removed copper from `if` statement. So `drawntreasure++` will not execute if the player draws a copper, allowing them to gain more treasures than the vard should allow (i.e. 2 treasures) (1).

```
// BUG: copper is now counted toward drawntreasure
// ORIGINAL: if (cardDrawn == copper || cardDrawn == silver || cardDrawn == gold)
if (cardDrawn == silver || cardDrawn == gold)
    drawntreasure++;
```

2.1.2 myCouncil_room

The number of buys is decremented where it should be incremented. This gives the player an additional buy every time the Council Room card is played. The correct behavior would remove a buy every time this card is played (2).

```
// BUG: number of buys is decremented where it should be incremented.
// ORIGINAL: state->numBuys++;
```

```
state->numBuys--;
```

2.1.3 myFeast

The feast card is sent back to the player's deck instead of going to the discard pile (3).

```
// BUG: Add card to currentPlayer's deck instead of discard pile.
// ORIGINAL: gainCard(choice1, state, 0, currentPlayer);//Gain the card
gainCard(choice1, state, 1, currentPlayer);//Gain the card
```

2.1.4 mySmithy

The smithy card will be played by player, but will not be placed in the played pile (4).

```
// BUG: card will be played by player, but will not be placed in the played pile.
// ORIGNAL: discardCard(handPos, currentPlayer, state, 0);
discardCard(handPos, currentPlayer, state, 1);
```

2.2 Program Execution

At each step of the assignment, the program was compiled (`make all && ./playdom 2`) and run to confirm that successful 2-player execution took place, despite the function refactoring as well as the introduction of bugs. The condensed results of which follow.

2.2.1 Original File (No Alterations Made)

```
Starting game.
...
Finished game.
Player 0: 30
Player 1: 40
```

2.2.2 Refactored Function Implementations (No Bugs Introduced Yet)

```
Starting game.
...
Finished game.
Player 0: 51
Player 1: 4
```

2.2.3 Bug Introductions

```
Starting game.
```

```
...
Finished game.
Player 0: 88
Player 1: 4
```

3 Sources

- [1]: http://wiki.dominionstrategy.com/index.php/Adventurer
- [2]: http://wiki.dominionstrategy.com/index.php/Council_Room
- [3]: http://wiki.dominionstrategy.com/index.php/Feast
- [4]: http://wiki.dominionstrategy.com/index.php/Smithy