DREW WOLFE | ONID: WOLFEDR
CS 362-400 WINTER 2018
ASSIGNMENT: 5, BUG REPORT & DEBUGGING

Bug Reports

I foolishly chose my teammate Zheng's code, of which he ruthlessly rid of the bugs he manufactured by the time he submitted *Assignment 3*. I therefore, at his recommendation, used the *Assignment 2* `dominion.c` code where his manufactured bugs could be found.

The only changes I had to make regarding my tests against Zheng's code were aesthetic. While we used different function names as part of the card refactors in *Assignment 2*, neither of us altered the means by which the cards were called within the `cardEffect()` switch statement. I simply changed the output within a few `printf()` statements from, for example, "Random Test of simthy_card() Function" to "Random Test of play_smithy() Function". Alternatively, while Zheng and I both refactored *Adventurer*, *Smithy*, and *Village* cards as part of *Assignment 2*'s refactor, Zheng refactored the *Remodel* card while I refactored the *Council Room* card. Consequently, the test I developed in `randomtestcard1.c` simply tests the original code for the *Council Room* card effect as opposed to the refactored code Zheng developed.

Even with having a reasonably good idea of where the bugs should be found, while they were noticeable in gameplay, they required a very deep knowledge of game rules and expected outcomes in order to find them. Even with having worked with the `dominion` code for as long as we have, and being thoroughly familiar with the rules, I only discovered one of the bugs prior to "digging into the code" and running my tests. Both the first and second bugs became apparent in test outcomes. I used the bug report template from noverse.com and altered it to better reflect the nature of the discovered bugs, specifically tailoring it to reflect an in-terminal user-experience (as opposed to a web-based experience). The content of which is as follows:

DREW WOLFE | ONID: WOLFEDR
CS 362-400 WINTER 2018
ASSIGNMENT: 5, BUG REPORT & DEBUGGING

Title: Smithy Card Effect Error
Class: Serious Bug
Date: March 4, 2018
Reported By: Drew Wolfe (wolfedr)
Email: wolfedr@oregonstate.edu

File(s): dominion.c
Function(s): int play_smithy();
Line(s): 675 - 686

Is it reproducible: Yes

Description

In running a unit test (cardtest1.c) that validates the effect of the Smithy Card, three (3)of the tests failed (HAND COUNT, DECK COUNT, SCORE). Upon closer inspection, it was clear that the card mistakenly draws 4 cards rather than 3.

Steps to Produce/Reproduce

Compile the game code and play the game:
1. $ make all
2. $ ./playdom 956
This may need to be performed more than once in the event the Smithy Card is not played over the course of game play. In the event that it is, it should be clear that an extra card is drawn.

Expected Results

The Smithy Card should draw three (3), not four (4) cards during a single turn

Actual Results

The Smithy Card draws four (4) cards

Workarounds

I could not manage a workaround w/o manipulating the game code. Occasionally the Smithy Card is not drawn during gameplay, in which case the bug does not have an effect

Other Information

It's potentially worth reviewing the README file to aid in game compilation

DREW WOLFE | ONID: WOLFEDR
CS 362-400 WINTER 2018
ASSIGNMENT: 5, BUG REPORT & DEBUGGING

```
Title:       Village Card Effect Error
Class:       Serious Bug
Date:              March 4, 2018
Reported By:       Drew Wolfe (wolfedr)
Email:       wolfedr@oregonstate.edu

File(s):     dominion.c
Function(s): int play_village();
Line(s):     690 - 701

Is it reproducible: Yes

Description
------------------------------------------------------------------------------
In running a unit test (cardtest4.c) that validates the effect of the Village
Card, one (1) of the tests failed (ACTIONS). Upon closer inspection, it was clear
that the card mistakenly provides one less ACTION than it should.

Steps to Produce/Reproduce
------------------------------------------------------------------------------
Compile the game code and play the game:
1. $ make all
2. $ ./playdom 956
This may need to be performed more than once in the event the Village Card is not
played over the course of game play. In the event that it is, it should be clear
that it does not provide the correct number of ACTION outputs.

Expected Results
------------------------------------------------------------------------------
The Village Card should grant two (2) ACTION turns, not one (1) during a single
turn

Actual Results
------------------------------------------------------------------------------
The Village Card grants a single (1) ACTION turn during a single turn

Workarounds
------------------------------------------------------------------------------
I similarly could not manage a workaround w/o manipulating the game code.
Occasionally the Village Card is not drawn during gameplay, in which case the bug
does not have an effect

Other Information
------------------------------------------------------------------------------
It's potentially worth reviewing the README file to aid in game compilation

******************************************************************************

REFERENCE: Template courtesy of noverse.com
(http://noverse.com/blog/2012/06/how-to-write-a-good-bug-report/)
```

DREW WOLFE | ONID: WOLFEDR
CS 362-400 WINTER 2018
ASSIGNMENT: 5, BUG REPORT & DEBUGGING

Test Report

It was an exceptionally useful experience to be able to run my tests against someone else's code, even if, for the most part, the code was the same. The process definitely helped to clarify the necessity to follow convention for the sake of portability. It also helped to understand the utility of being able to reference a function within a function (essentially generate a pseudo-pointer) as seen in the `cardEffect()` function where references to cards within the switch statement can be used to execute in-line code or return the output of a function (in the case of our refactored card functions from *Assignment 2*). I updated the `MakeFile` as a means to tailor-make a couple of test suites above and beyond `unittestresults.out`. I created `completeTestSuite.out` and `completeExhaustiveTestSuite.out`. I added breaks designed to print in between individual tests with unique strings that could be used to search for the beginning of each output. This helped tremendously to parse the extensive output produced by `gcov`. Ultimately the test suite `completeTestSuite.out` ended up being the most useful as it highlighted the most salient findings while remaining comprehensive. It includes all unit tests as well as random tests along with `gcov` branch and function coverage testing.

It would be unfair to judge Zheng's *Assignment 2* code for "reliability" as part of the assignment's objective was to specifically manufacture bugs (that and the fact that Zheng is one of the most diligent coders I know… I mean he's got a bachelor's degree in mathematics! Of course he's going to be an amazing engineer!). Suffice it to say that if I were to run my test suite(s) against Zheng's final version of `dominion.c` (within which I've been assured that all manufactured bugs have been squashed), knowing that he's run his tests against it as well, I'd be more wary of the integrity of my tests if they came back with bugs than of the dependability of his fixed `dominion.c` code (which actually touches on a point I make in the *Debugging* section).

Overall, I believe I've learned how to develop useful, high-yielding and dependable tests for `dominion`, and software in general. The tests I've developed point me to the issues in the source code and, of course, the `gcov` output does an excellent job of highlighting where greater coverage could be developed. The test suite's most salient findings:

```
COMPLETE TEST RESULTS:

UNIT TEST 1: SHUFFLE() FUNCTION

Player's DECK COUNT = 0                Player's DECK COUNT = 10              Deck   shuffled   by   shuffle()
       PASSED                                 PASSED                       function
                                                                                  FAILED


SHUFFLE(): ONE OR MORE TESTS FAILED. SEE OUTPUT
FAILURES: 1

******************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ********************

UNIT TEST 2: GAINCARD() FUNCTION

GOLD = 0                                                                          PASSED
       PASSED                          +1 to DISCARD COUNT
                                              PASSED                      -1 GOLD supply & compare
+1 to DECK COUNT                                                                  PASSED
       PASSED                          +1 to HAND COUNT
```

DREW WOLFE | ONID: WOLFEDR

CS 362-400 WINTER 2018
ASSIGNMENT: 5, BUG REPORT & DEBUGGING

```
GAINCARD(): ALL TESTS COMPLETED SUCCESSFULLY!

****************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ******************

UNIT TEST 3: UPDATECOINS() FUNCTION


BONUS: 1                                COINT COUNT - GOLD                  COINT COUNT - SILVER
HAND COUNT: 0                           Expected output: 1                  Expected output: 13
                                        Actual output: 1                    Actual output: 3
COINT COUNT - COPPER                           PASSED                               FAILED
Expected output: 1
Actual output: 1                        BONUS: 3                            COINT COUNT - GOLD
       PASSED                           HAND COUNT: 5                       Expected output: 18
                                                                            Actual output: 3
COINT COUNT - SILVER                    COINT COUNT - COPPER                       FAILED
Expected output: 1                      Expected output: 8
Actual output: 1                        Actual output: 3                    UPDATECOINS(): ONE OR MORE TESTS
       PASSED                                   FAILED                      FAILED. SEE OUTPUT
                                                                            FAILURES: 3

****************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ******************

UNIT TEST 4: ISGAMEOVER() FUNCTION

PROVINCE count = 0                             PASSED                       Each of 3 piles in supplyCount =
       PASSED                                                               1
Each of 3 piles in supplyCount =        PROVINCE count NOT EMPTY                   PASSED
0

ISGAMEOVER(): ALL TESTS COMPLETED SUCCESSFULLY!

****************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ******************
CARD TEST 1: SMITHY
       FAIL: Incorrect HAND COUNT after playing SMITHY.
        Expected output 7, actual output 8
       FAIL: Incorrect DECK COUNT after playing SMITHY.
        Expected output 2, actual output 1
       FAIL: Incorrect SCORE after playing SMITHY.
        Expected output 1, actual output 2
       PASS: PLAYER-2's HAND COUNT correct after PLAYER-1 played SMITHY
       PASS: PLAYER-2's DECK COUNT correct after PLAYER-1 played SMITHY
       PASS: PLAYER-2's SCORE correct after PLAYER-1 played SMITHY
       PASS: Correct number of ESTATE cards after playing SMITHY
       PASS: Correct number of DUCHY cards after playing SMITHY
       PASS: Correct number of PROVINCE cards after playing SMITHY
       PASS: Initial card count #0 contains initial number of cards
       PASS: Initial card count #1 contains initial number of cards
       PASS: Initial card count #2 contains initial number of cards
       PASS: Initial card count #3 contains initial number of cards
       PASS: Initial card count #4 contains initial number of cards
       PASS: Initial card count #5 contains initial number of cards
       PASS: Initial card count #6 contains initial number of cards
       PASS: Initial card count #7 contains initial number of cards
       PASS: Initial card count #8 contains initial number of cards
       PASS: Initial card count #9 contains initial number of cards

SMITHY: ONE OR MORE TESTS FAILED. SEE OUTPUT

****************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ******************
```

DREW WOLFE | ONID: WOLFEDR
CS 362-400 WINTER 2018
ASSIGNMENT: 5, BUG REPORT & DEBUGGING

```
CARD TEST 2: ADVENTURER
        FAIL: Hand count after playing adventurer was not correct.
         Expected output 6, actual output 7
        PASS: TREASURE card was taken first
        PASS: TREASURE card was second draw
        PASS: Correct DECK COUNT for ADVENTURER (-2)
        PASS: Correct SCORE for ADVENTURER (no change)
        PASS: PLAYER-2's HAND COUNT correct after PLAYER-1 played ADVENTURER
        PASS: PLAYER-2's DECK COUNT correct after PLAYER-1 played ADVENTURER (no change)
        PASS: PLAYER-2's SCORE correct after PLAYER-1 played ADVENTURER (no change)
        PASS: Correct number of ESTATE cards after playing ADVENTURER (no change)
        PASS: Correct number of DUCHY cards after playing ADVENTURER (no change)
        PASS: Correct number of PROVINCE cards after playing ADVENTURER (no change)
        PASS: Initial card count #0 contains initial number of cards
        PASS: Initial card count #1 contains initial number of cards
        PASS: Initial card count #2 contains initial number of cards
        PASS: Initial card count #3 contains initial number of cards
        PASS: Initial card count #4 contains initial number of cards
        PASS: Initial card count #5 contains initial number of cards
        PASS: Initial card count #6 contains initial number of cards
        PASS: Initial card count #7 contains initial number of cards
        PASS: Initial card count #8 contains initial number of cards
        PASS: Initial card count #9 contains initial number of cards

ADVENTURER: ONE OR MORE TESTS FAILED. SEE OUTPUT

******************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ********************

CARD TEST 3: COUNCIL ROOM

PLAYER-1 gains 4 cards                                  Comparing gameState w/ testCpy supplyCount
        PASSED                                                  PASSED
                                                        Comparing gameState w/ testCpy supplyCount
PLAYER-1 gains 4 cards from his own pile                        PASSED
        PASSED                                          Comparing gameState w/ testCpy supplyCount
                                                                PASSED
PLAYER-1 number of buys increments                      Comparing gameState w/ testCpy supplyCount
        PASSED                                                  PASSED
                                                        Comparing gameState w/ testCpy supplyCount
Expecting no change to following VICTORY card piles             PASSED
PROVICE Pile                                            Comparing gameState w/ testCpy supplyCount
        PASSED                                                  PASSED
DUCHY Pile                                              Comparing gameState w/ testCpy supplyCount
        PASSED                                                  PASSED
ESTATE Pile                                             Comparing gameState w/ testCpy supplyCount
        PASSED                                                  PASSED

Expecting no change to testCards supplyCount            PLAYER-2 expected to gain 1 card
Comparing gameState w/ testCpy supplyCount                      PASSED
        PASSED
Comparing gameState w/ testCpy supplyCount              PLAYER-2 expected to gain 1 from PLAYER-2 own deck
        PASSED                                                  PASSED


COUNCIL ROOM: ALL TESTS COMPLETED SUCCESSFULLY!

******************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ********************

CARD TEST 4: VILLAGE
        PASS: Correct HAND COUNT for VILLAGE (no change)
        PASS: Correct DECK COUNT for VILLAGE (-1)
```

```
        PASS: Correct SCORE for VILLAGE (no change)
        FAIL: Incorrect ACTIONS count after playing VILLAGE.
         Expected output 3, actual output 2
        PASS: PLAYER-2's HAND COUNT correct after PLAYER-1 played VILLAGE
        PASS: PLAYER-2's DECK COUNT correct after PLAYER-1 played VILLAGE
        PASS: PLAYER-2's SCORE correct after PLAYER-1 played VILLAGE
        PASS: Correct number of ESTATE cards after playing VILLAGE
        PASS: Correct number of DUCHY cards after playing VILLAGE
        PASS: Correct number of PROVINCE cards after playing VILLAGE
        PASS: Initial card count #0 contains initial number of cards
        PASS: Initial card count #1 contains initial number of cards
        PASS: Initial card count #2 contains initial number of cards
        PASS: Initial card count #3 contains initial number of cards
        PASS: Initial card count #4 contains initial number of cards
        PASS: Initial card count #5 contains initial number of cards
        PASS: Initial card count #6 contains initial number of cards
        PASS: Initial card count #7 contains initial number of cards
        PASS: Initial card count #8 contains initial number of cards
        PASS: Initial card count #9 contains initial number of cards

VILLAGE: ONE OR MORE TESTS FAILED. SEE OUTPUT

********** SEARCHABLE STRING SEPARATOR: 8%^626+39 **********
Testing drawCard.
RANDOM TESTS.
ALL TESTS OK
********** SEARCHABLE STRING SEPARATOR: 8%^626+39 **********
Testing buyCard.
RANDOM TESTS.
ALL TESTS OK
********** SEARCHABLE STRING SEPARATOR: 8%^626+39 **********
********** RANDOM TEST **********

                Random Test of council_room cardEffect()

**************************************************************************
********** FINDINGS **********
TESTS PASSED: 25000
TESTS FAILED: 0

********** HOLY COW, RANDOM TESTS PASSED!!! **********

********** GCOV OUTPUT **********
********** SEARCHABLE STRING SEPARATOR: 8%^626+39 **********
********** RANDOM TEST **********

                    Random Test of play_smithy() Function

**************************************************************************

********** FINDINGS **********
TESTS PASSED: 2
TESTS FAILED: 24998


********** WOMP WOMP... RANDOM TEST FAILED **********
drawCard() ERROR: 0
discardCard() ERROR: 0
cardEffect() ERROR: 0
INVALID hand OR deck count: 24998

********** RUBBING IT IN... RANDOM TEST FAILED **********
```

DREW WOLFE | ONID: WOLFEDR

CS 362-400 WINTER 2018

ASSIGNMENT: 5, BUG REPORT & DEBUGGING

```
******************************** GCOV OUTPUT ********************************
******************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ********************
******************************** RANDOM TEST ********************************

                    Random Test of play_adventurer() Function

*****************************************************************************

********************************* FINDINGS **********************************
TESTS PASSED: 349
TESTS FAILED: 24651

*********************** WOMP WOMP... RANDOM TEST FAILED ***********************
drawCard() ERROR: 0
cardEffect() ERROR: 0
shuffle() ERROR: 0
INVALID Treasure Count: 0
INVALID hand OR deck count: 24651

********************** RUBBING IT IN... RANDOM TEST FAILED **********************

******************************** GCOV OUTPUT ********************************
******************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ********************
Function 'updateCoins'                  Calls executed:0.00% of 2               Function 'isGameOver'
Lines executed:0.00% of 11                                                      Lines executed:0.00% of 10
Branches executed:0.00% of 8            Function 'play_smithy'                  Branches executed:0.00% of 8
Taken at least once:0.00% of 8          Lines executed:0.00% of 5               Taken at least once:0.00% of 8
No calls                                Branches executed:0.00% of 2            No calls
                                        Taken at least once:0.00% of 2
Function 'gainCard'                     Calls executed:0.00% of 2               Function 'endTurn'
Lines executed:0.00% of 13                                                      Lines executed:0.00% of 20
Branches executed:0.00% of 6            Function 'play_adventurer'              Branches executed:0.00% of 6
Taken at least once:0.00% of 6          Lines executed:91.67% of 12             Taken at least once:0.00% of 6
Calls executed:0.00% of 1               Branches executed:100.00% of 10         Calls executed:0.00% of 3
                                        Taken at least once:90.00% of 10
Function 'discardCard'                  Calls executed:50.00% of 2              Function 'whoseTurn'
Lines executed:0.00% of 13                                                      Lines executed:100.00% of 2
Branches executed:0.00% of 6            Function 'getCost'                      No branches
Taken at least once:0.00% of 6          Lines executed:0.00% of 30              No calls
No calls                                Branches executed:0.00% of 28
                                        Taken at least once:0.00% of 28         Function 'fullDeckCount'
Function 'cardEffect'                   No calls                                Lines executed:0.00% of 9
Lines executed:4.88% of 205                                                     Branches executed:0.00% of 12
Branches executed:12.85% of 179         Function 'drawCard'                     Taken at least once:0.00% of 12
Taken at least once:1.68% of 179        Lines executed:36.36% of 22             No calls
Calls executed:3.57% of 56              Branches executed:33.33% of 6
                                        Taken at least once:16.67% of 6         Function 'supplyCount'
Function 'play_great_hall'              Calls executed:0.00% of 1               Lines executed:0.00% of 2
Lines executed:0.00% of 6                                                       No branches
No branches                             Function 'getWinners'                   No calls
Calls executed:0.00% of 2               Lines executed:0.00% of 24
                                        Branches executed:0.00% of 22           Function 'handCard'
Function 'play_remodel'                 Taken at least once:0.00% of 22         Lines executed:0.00% of 3
Lines executed:0.00% of 11              Calls executed:0.00% of 2               No branches
Branches executed:0.00% of 6                                                    Calls executed:0.00% of 1
Taken at least once:0.00% of 6          Function 'scoreFor'
Calls executed:0.00% of 5               Lines executed:0.00% of 24              Function 'numHandCards'
                                        Branches executed:0.00% of 42           Lines executed:0.00% of 2
Function 'play_village'                 Taken at least once:0.00% of 42         No branches
Lines executed:0.00% of 5               Calls executed:0.00% of 3               Calls executed:0.00% of 1
No branches
```


```
Function 'buyCard'                  Calls executed:0.00% of 2           No calls
Lines executed:0.00% of 13
Branches executed:0.00% of 6        Function 'initializeGame'           Function 'compare'
Taken at least once:0.00% of 6      Lines executed:0.00% of 62          Lines executed:0.00% of 6
Calls executed:0.00% of 4           Branches executed:0.00% of 46       Branches executed:0.00% of 4
                                    Taken at least once:0.00% of 46     Taken at least once:0.00% of 4
Function 'playCard'                 Calls executed:0.00% of 5           No calls
Lines executed:0.00% of 14
Branches executed:0.00% of 10       Function 'kingdomCards'             File 'dominion.c'
Taken at least once:0.00% of 10     Lines executed:0.00% of 13          Lines executed:5.58% of 556
Calls executed:0.00% of 3           No branches                         Branches executed:8.43% of 415
                                    No calls                            Taken at least once:3.13% of 415
Function 'shuffle'                                                      Calls executed:3.16% of 95
Lines executed:0.00% of 16          Function 'newGame'                  Creating 'dominion.c.gcov'
Branches executed:0.00% of 8        Lines executed:0.00% of 3
Taken at least once:0.00% of 8      No branches

******************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ********************
Function 'adventurerCrdTst'         Lines executed:90.48% of 42         Branches executed:89.29% of 56
Lines executed:87.50% of 48         Branches executed:100.00% of 10     Taken at least once:75.00% of 56
Branches executed:86.96% of 46      Taken at least once:80.00% of 10    Calls executed:83.33% of 30
Taken at least once:73.91% of 46    Calls executed:85.19% of 27         Creating
Calls executed:66.67% of 3                                              'randomtestadventurer.c.gcov'
                                    File 'randomtestadventurer.c'
Function 'main'                     Lines executed:88.89% of 90

******************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ********************
Function 'councilRoomCrdTst'        Lines executed:82.22% of 45         Branches executed:74.07% of 54
Lines executed:75.51% of 49         Branches executed:100.00% of 8      Taken at least once:50.00% of 54
Branches executed:69.57% of 46      Taken at least once:87.50% of 8     Calls executed:75.00% of 32
Taken at least once:43.48% of 46    Calls executed:68.00% of 25         Creating
Calls executed:100.00% of 7                                             'randomtestcard1.c.gcov'
                                    File 'randomtestcard1.c'
Function 'main'                     Lines executed:78.72% of 94

******************** SEARCHABLE STRING SEPARATOR: 8%^626+39 ********************
Function 'smithyCrdTst'             Lines executed:96.97% of 33         Branches executed:73.33% of 30
Lines executed:77.42% of 31         Branches executed:100.00% of 6      Taken at least once:50.00% of 30
Branches executed:66.67% of 24      Taken at least once:83.33% of 6     Calls executed:96.43% of 28
Taken at least once:41.67% of 24    Calls executed:95.65% of 23         Creating
Calls executed:100.00% of 5                                             'randomtestcard2.c.gcov'
                                    File 'randomtestcard2.c'
Function 'main'                     Lines executed:87.50% of 64
```

Debugging

I worked with the `aburasa/dominion` code and initially used *GDB* via the terminal, following the instruction from both our lectures as well as this Baylor computer science program document. I have to say, it generally felt like I was making shots-in-the-dark. I felt like the breakpoints I was defining were generally arbitrary and that I wasn't really finding issues within the code, given that the code compiles and runs. I moved to using *GDB* within an IDE and that made the process both faster and seemingly provided greater insight into the overarching process, but it still didn't expose "incorrect outcomes within functioning code". In general, it seems *GDB* would be incredibly useful to get code up and running (despite the fact that it may break external, non-dependent logical rules), but it's not especially good, or at least I wasn't able to flex or utilized its functionality in a way that exposed logical breaks in fully functioning code.

DREW WOLFE | ONID: WOLFEDR
CS 362-400 WINTER 2018
ASSIGNMENT: 5, BUG REPORT & DEBUGGING

Of course we have, as a class, been pouring over this code for weeks now, and between myself and a few of my now friends within the program, have recognized that there is an inherent bug in how the *Adventurer* card does not discard the played *Adventurer* card itself after it's been played. And while this became a more glaring and obvious issue overtime, it wasn't immediately apparent without an extensive understanding of the game's rules and it wasn't exposed by a debugging tool (though, again, that could be more a fault of my own ignorance rather than that of the tool). The code simply had to be updated to reflect, essentially as the last step, that the *Adventurer* card itself is discarded.

Additionally, while it should have been obvious against my own code, running my tests against Zheng's code seemed to highlight an issue in the test in `unittest3.c`, focused on the `updateCoins()` function. I had to re-check the math and update the impact in the difference between copper, silver, and gold coins.