Benjamin Tate
11/19/2017
CS 362 -- Section 400
Assignment 5

Bug Reports:

Bug #1:

- Title: Village card adds one action instead of two
- Bug ID: Village741
- Summary: This bug consistently causes the Village card to add only one action, when it should add two. The source of the bug seems to be line 741 of dominion.c.
- How to find it: It can be reliably found by checking the number of actions a given player has before and after any time they play the Village card.
- Code to reliably produce the bug: cardEffect(village, 0, 0, 0, &state, 0, 0)
- Software version: "Assignment 4 Finished" commit on 10/27/17.
- Estimated severity: Since this bug reliably occurs every time the Village card is played, thereby totally breaking that card, it should be considered very severe.
- Estimated priority: This bug won't interfere with testing of anything that doesn't involve the Village card, so needs only to be fixed before such testing occurs, or before the final product is shipped.

Bug #2:

- Title: Adventurer card adds wrong number of treasure
- Bug ID: Adventurer666
- Summary: This bug causes the Adventurer card to add the wrong number of treasure to the player's hand by failing to count how many have been drawn correctly. The source of the bug seems to be on line 666 of dominion.c.
- How to find it: The bug can be found fairly reliably by checking that the 2 treasure are added to the player's hand after playing the adventurer card. Although the bug occurred in every one of 200 random tests, its results are somewhat inconsistent, so it is possible that it could go unnoticed with the right deck configuration.
- Code to produce the bug: cardEffect(adventurer, 0, 0, 0, &state, 0, 0)
- Software version: "Assignment 4 Finished" commit on 10/27/17.
- Estimated severity: Since this bug occurs at least nearly every time the Adventurer card is played, it should be considered very severe.
- Estimated priority: This bug won't interfere with testing of anything that doesn't involve the Adventurer card, so needs only to be fixed before such testing occurs, or before the final product is shipped.

Bug #3:

- Title: Smithy card draws four cards instead of three
- Bug ID: Smithy695

- Summary: This bug consistently causes the Smithy card to draw the wrong number of cards (4 instead of 3). The source of the bug seems to be on line 695 of dominion.c.
- How to find it: Check that a player has 2 more cards in their hand after playing the Smithy card (+3, -1), and you will consistently find that they have an extra
- Code to reliably produce the bug: cardEffect(smithy, 0, 0, 0, &state, 0, 0)
- Software version: “Assignment 4 Finished” commit on 10/27/17.
- Estimated severity: Since this bug reliably occurs every time the Smithy card is played, thereby totally breaking that card, it should be considered very severe.
- Estimated priority: This bug won’t interfere with testing of anything that doesn’t involve the Smithy card, so needs only to be fixed before such testing occurs, or before the final product is shipped.

Test Report:

Testing Michael’s code was fairly similar to testing my own, except that I only knew of two cards that he had definitely put bugs in, Smithy and Adventurer. As it happened, my tests also covered the Village card, in which he had also included a bug. My tests’ coverage was of course just as good for his dominion.c as it was for mine, since the structure of the two files aren’t very different, and my tests did a good job of finding bugs. That said, I had to look a little harder at his code and/or add more useful print statements to my tests to figure out exactly *what* was going wrong rather than just that *something* was wrong. For example, in the Smithy card, I could tell that the wrong number of cards was being drawn, but I had to revise my error message in cardtest1.c to tell me how many were being drawn. That allowed me to see exactly what was causing the bug. Overall, I would say that Michael’s dominion code was fairly reliable, aside from a few easily fixable bugs.

Debugging:

Starting from running my Smithy card test (cardtest1), I knew that something was wrong with how many cards the Smithy card drew from the deck. I then ran the command ‘gdb playdom’ to start GDB, and ran ‘break smithyCase to set a breakpoint at the smithyCase() function where my refactored Smithy card code is. Next, I ran ‘run 10’ to run playdom for 10 games. A Smithy was played after several turns of the first game, triggering the breakpoint I set. I ran ‘bt’ to see what was going on with the program at that point. I then ran ‘watch state->handCount[currentPlayer] to keep track of how many cards were being drawn (Smithy should draw 3 cards). By typing ‘continue’ a few times, I could see the value of the player’s handCount increment twice just before the block was left. At this point, it was clear that the problem was that smithyCase() was set to draw 2 cards instead of 3, so I went into dominion.c, scrolled down to the function, and upon inspection, found that the for loop was set to go from i=0 to i=1, which of course would execute only twice. I changed the 2 to a 3, and ran my tests again, and the Smithy card passed them all.