

Assignment 5

Bug Reports

SMITHY - The first bug I found in this assignment was created by my teammate as part of HW3 in the Smithy function. He replaced the 0 for the trash flag with a 1 to have the player trash their card instead of place it in the discard pile. This bug is found on line 657 in dominion.c. I altered the code provided by my teammate to return a -1 if the players hand reaches zero. Once that happens, the code continues to run but will return -1 with each loop of the Smithy card test.

ADVENTURER – The bug that I found in Adventurer only stops drawing treasure when it is equal to two. Meaning that it draws treasure for 0, 1, and 2. This gives the player three Treasure cards instead of the two that it should get with the Adventurer card.

SCOREFOR – The scoreFor() works as intended other than the bug that I found in which the score for the discard pile is counted twice instead of the players's score for their hand being counted. This bug is on line 443 of dominion.c. Because of this bug, the score for the player is not returned correctly and my test function returns a failure.

The above bug in Smithy was also exposed in my Council Room and Outpost card test functions. The tests on the discard pile after those test functions are run returns "FAILED".

- Debugging

SMITHY - It took me almost 2 full days of work to realize what was happening in the Smithy function of my teammates code. Because the bug that he introduced has the player trash their cards, after 10 loops of my smithy test function, the code gets a segmentation fault. I finally realized that the code was trashing(NOT discarding) the players 10 cards in their hand and deck. After that the player did not have a card to trash and the function was trying to access memory locations that did not exist. In order to find this bug, I changed how many loops were in my test loop and narrowed the "max-number-of-loops-till-seg-fault" down to 10. I inserted return statements in all of the associated functions so that if any of the parameters were out of range or the function tried to access invalid memory the function would return -1. In order to do this, I changed the Smithy function to return an integer instead of being void. Finally, I commented out sections of the code to narrow down where the bug was located. After finding that the bug was in discardCard(), I backtracked and found that the trash flag

was set to 1 instead of 0. Upon further investigation I found that this was a bug introduced by my teammate.

ADVENTURER – My teammate created a bug in adventurer that caused the function to draw three cards instead of two. This did not have an effect on my function as my function merely tested if my funcAdventure returned 0 or -1. The bug that I created in Adventurer created an infinite loop. I realize now that I could have had my test function check the number of cards drawn or in their hand before and after the function. It would have been a much more thorough test if I had that in place.

OUTPOST – My outpost test function sent 20 random numbers from 0 to 100 as hand position 1-5. The Outpost card allows a player to discard any 1 card from their hand. My function recorded which card was discarded and that Outpost had been played. There were no bugs found in this function.

INITIALIZEGAME – my test function for the initialization of the game verified that players were given the correct number of cards. My test function passed on my team members code. No bugs were found.

SCOREFOR - My unit test four scoreFor() involved incrementally changing the cards to cards of different values and then checking the value returned by scorefor(). Dominion failed this test because there is a bug within scorefor(). The function does not measure the score in the players hand. It measures the score in the discard pile twice. This bug is found on line 443 of dominion.c.

SHUFFLE – The Shuffle function works as intended without bugs. It passed my test in which I shuffled the deck multiple times and checked that no more than 20% of the array spots were the same card after the shuffle.

NUMHANDCARDS – This function returns the number of cards in each players hand. It returned 0 errors as I switched between each player, manually changed the number of cards in each players hand, and then checked, through numHandCards(), the number of cards in each players hand.

SMITHY UNIT TEST - This version of Smithy also fails because of the bug introduced by my teammate that trashes the cards instead of discarding them.

ADVENTURER – I only had to make a few changes to this code to make it work. This code could definitely be more thorough in checking all of the different changes that playing an Adventurer card causes to happen. Due to the bug that I have in my Adventurer function I was unable to check those things. The Catch-22 is that this test was the only thing I could write for my adventure function. Without that bug almost any adventurer function would pass my test.

COUNCIL_ROOM – this test actually surprised me and that it revealed the bug that my teammate introduced into the discard function. I test the number of cards in the discard pile after playing Council_Room. The return is that there are zero cards in the pile when there should be one. One card is trashed because of the trash card flag being switched in my teammates code.

OUTPOST – The outpost card allows the player to take another turn but at the end of the turn the player will discard the Outpost card. This function detects the same bug in discardCard() in which the card is trashed instead of placed on the discard pile.

NOTE:The combined results of these tests could theoretically be used to locate the bug within the discard function.

- Test Report.

Noverse Bug Reporting Template

=====

Title: Seg. Fault when playing the Smithy card.

Class: "Serious Bug"

Date: 11.26.18

Reported By: Mark Moore

Email: moorem9@oregonstate.edu

Product: dominion.c(and associated files)

Platform: Written in C

Is it reproducible: Yes – every time Smithy is played.

Description

In my bug test, when I send Smithy directly into the cardEffect() function, the code crashes without returning any text from that whole test session.

Steps to Produce/Reproduce

When this function call is sent into dominion.c...

cardEffect(smithy,0,0,0,&G,0,&coin_bonus);

the code immediately crashes. Smithy allows the player to draw 3 cards. Then, the player will discard the Smithy card and continue with their hand.

Expected Results

In my test the expectation is that the player will draw 3 cards. Each time `cardEffect(Smithy)` is called, this will happen. Occasionally, reshuffling of the players deck is expected because the player is repeatedly drawing 3 cards from their deck.

Actual Results

When the above function is sent to `dominion.c`, the code will crash and does not print anything to screen even when there are print commands within the code.

Workarounds

I have, thus far, been unable to find a workaround to this bug.

Attachments

NONE

Other Information

NONE