

In my opinion, the test of dominion is very interesting but it is very hard. If we don't understand the game rules, we might face many trouble in writing test. In the dominion game, each card has its own effect. For example, the card council_room will add four cards and one buy, the smithy card will add three cards and the village card will add one card and two actions. As a result, understand the effect of each card will help us write the test more easily. Besides, it is also very important to get familiar with the variable in dominion.c. For instance, if we need to check the turn of the player then we could printf the value of game->whoseTurn. This will also make writing test more easily.

During the test, the first assignment is requiring to choose five cards and refactor the code. The five cards i pick are council_room,smithy,village,great_hall and cutpurse. I refactor these card in function call and make two bugs in these card. The first bug is in the council_room card, i change the value of "i < 4" to be "i < 3" which means the drawcard function will be called only 3 times. Another bug is in the smithy card, i change the value of "i < 3" to be "i < 2" which means the drawcard function will be called only 2 times. The first assignment is not hard but it is necessary to understand how this game works. Therefore, it is better to play the game of dominion online to get familiar with the game rules.

The second assignment is requiring to write eight unit tests. Four unit test for the function except(card implementation) and Four unit test for the cards. Then, we need to get the coverage of the test by using gcov command in makefile. The four functions i choose are numHandcards,kingdonCards,whoseTurn and shuffle function. The four cards i choose are smithy, village,great_hall and council_room. After running the test, i don't find any bugs on the unit test for the function. However, i find two bugs in the unit test for the card which is the bug i change in the first assignment. The unittestresult.out file shows the coverage of the test: Lines executed:21.51% of 558. The coverage is not large because i only pick several functions and cards to test.The second is also not hard but it is very useful to understand how unit test works.

The third assignment is requiring to write random test for three cards. The first one is adventurer and random pick the second one and the third one then print out the result with coverage and compare with the unit test. The card i choose to write random test are adventurer, smithy and village. For each cards i check how the card works in dominion.c. For instance, the smithy will add three cards which means the number of the card on hand(before use the card) will be more than the number of the card on hand(after use the card). Similarly, the village will add one card and two action.In the test, i check some function and if there is a error, it will get the numbers of the error and print out to the output file For instance, set r equal to the initialize function and check the value of r. If r is not correct, then add 1 to the num(which use to get the total error numbers) similar for the effect function.Besides, i also check the number of the card on hand(before use the card) and the number of the card on hand(after use the card). Furthermore, village will add two action so i check the number of action (before use the

card) and the number of action (after use the card). Based on the test result, randomtestadventurer.out shows:

Lines executed:24.01% of 558.

the randomtestcard1.out shows:

Function 'function_renamed_smithy'

Lines executed:100.00% of 5

Branches executed:100.00% of 2

Taken at least once:100.00% of 2

Calls executed:100.00% of 2.

randomtestcard2.out shows:

Lines executed:20.25% of 558.

i feel the random test will have higher coverage than the unit test because it almost cover check all conditions(by using random variable) for the card that we want to test.however, the unit test will not use random variable and it might check some specific case inside of random case. The third assignment is becoming hard and it is very useful to understand how random test works in a specific game test.

The last assignment is the hardest assignment. It requires to write random test for the whole game with player 2-4 and test seed will be 42. And we also required to write the diffdominion which used to compare the testdominion result with our classmates. In this assignment, it is very important to understand how this whole game works. So play the game again is very useful. The Agans 'principles Understand the system also share the same idea. After the test, the result shows the coverage of my testdominion is almost 60% and the coverage of my classmate is almost 63%. This assignment is very hard because i get familiar with some card such as smithy, council_room, village but not for the all cards. Besides, i feel that i am not familiar with several functions. Furthermore , i also face some challenge in debugging.

During the comparing with the code of my classmates, i feel it is very hard to decide whose code is more reliable. For example, if there has some bugs in my code, then the code will not be reliable. However, the coverage is not 100% so there must be some case our test does not cover. I think the code of my classmates are reliable. I test two of my classmates and the first result show the coverage is almost 63% and the second is 67.86%.

In conclusion, this game test is very interesting but it is hard if the tester have no test experience before. Beside, debugging will be a necessary part for tester. The coverage usually shows the test cover for the whole game which is useful and necessary. I have learn a lot about this testing experience. However, with the knowledge of testing increasing, i feel the test is actually very hard.

