

REFACTOR DOCUMENTATION

I refactored the following 6 cards from outside the cardEffect function in the dominion.c file: (1) baron, (2) minion, (3) ambassador, (4) great_hall, (5) tribute, and (6) mine. The refactored functions are named baron_ref, minion_ref, ambassador_ref, great_hall_ref, and mine_ref in order to avoid confusion with the card names in the switch statement. The functions are above the cardEffect function and are executed when called in the cardEffect(int card...) place.

The refactored functions rely on the parameters being passed in through the cardEffect switch statement. Taking the cards outside of the cardEffect would mean that the parameters being passed in through the cardEffect could not be called to outside the cardEffect function (i.e. int card, int choice1, int choice2, int choice3, struct gameState *state, int handPos, int *bonus). Therefore, the refactored functions take the necessary parameters needed for that specific function to execute. It was not necessary to add all the same parameters for each function because some parameters, such as int choice 3 and int *bonus, are not called/used in any of the refactored functions.

To ensure each change to the functions were working, I compiled the code every time I created a new refactored function. I decided to refactor the great_hall function as well because it appears that the card great_hall is called outside of the cardEffect function quite often. However, there were errors in compiling in the player.c, rt.c, and playdom.c files, and therefore I was unable to test complete functionality of the functions without fixing the rest of the code-base.

BUGS

For each function, I introduced two bugs. For the baron_ref function, on line 698, I changed the while loop from while(card_not_discarded) to while(!card_not_discarded) – this would cause the function of the while-loop to not execute unless a card is discarded, opposite of the intention of the function. I also changed line 711, and removed the assignment of card_not_discarded.

For the minion_ref function, on line 779, I changed the for loop by removing the initial “i = 0” – this would cause the variable “i” to remain at 4 from the previous loop. On line 797, I added a return statement to exit the function completely, but only when the handCount is greater than 4 – this would prematurely end the function without loop through each if statement appropriately.

For the ambassador_ref function, on line 811, I changed the “or” statement to an “and” statement; from “if (choice2 > 2 || choice2 < 0)” to “if (choice2 > 2 && choice2 < 0)”. This would cause the if-statement not to run at all because choice2 can only be one number, not two. On line 826, I changed the variable from “j++” to “i++,” which would cause the for loop to end earlier than expected.

For the tribute_ref function, on line 891, I changed the if-statement to not execute anything, completely removing the return -1; this will cause the rest of the function to continue. On line 915, I switched the two lines of break and discardCard, calling break first before the discardCard function, which will never execute if a trashed card is found.

For the mine_ref function, on line 942, I changed the if then else statement to just an if statement, this will cause the if statement to execute both if the next player's discard and deckCount are equal to 0 and less than or equal to 1. On line 967, I changed the for-loop from "i = 0" to "i = 2"; this will cause the for loop to execute only once instead of looping through each card.