

## Refactor changes:

The refactoring changes were essentially taking the original code and "copy-pasting" into a separate function. The functions were called card\_\_\_\_\_ in which the blank is filled with the name of the card. The code of the card function was then read through to understand what variables were needed for the function to run properly. After, the code was copied over, the original code was replaced with the call for the function. Functions that required a return number were changed into int variables to be returned. The header file was then edited with the new functions.

Below for header edits added:

```
int cardMine(struct gameState *state, int currentPlayer, int choice1, int choice2, int handPos);
void cardTribute(struct gameState *state, int nextPlayer, int tributeRevealedCards[], int currentPlayer);
int cardAmbassador(struct gameState *state, int handPos, int currentPlayer, int choice1, int choice2);
void cardMinion(struct gameState *state, int handPos, int currentPlayer, int choice1, int choice2);
void cardBaron(struct gameState *state, int choice1, int currentPlayer);
```

Example of function made based on original code:

```
int cardAmbassador(struct gameState *state, int handPos, int currentPlayer, int choice1, int choice2){
    int i;
    int j;
    //changed 0 to 1
    j = 1; //used to check if player has enough cards to discard

    if (choice2 > 2 || choice2 < 0)
    {
        return -1;
    }

    if (choice1 == handPos)
    {
        return -1;
    }

    for (i = 0; i < state->handCount[currentPlayer]; i++)
    {
        if (i != handPos && i == state->hand[currentPlayer][choice1] && i != choice1)
        {
            j++;
        }
    }
    if (j < choice2)
    {
        return -1;
    }

    /*if (DEBUG)
    {
        printf("Player %d reveals card number: %d\n", currentPlayer, state->hand[currentPlayer][choice1]);
    }*/

    //increase supply count for choosen card by amount being discarded
    state->supplyCount[state->hand[currentPlayer][choice1]] += choice2;

    //each other player gains a copy of revealed card
    for (i = 0; i < state->numPlayers; i++)
    {
        if (i != currentPlayer)
        {
            gainCard(state->hand[currentPlayer][choice1], state, 0, i);
        }
    }

    //discard played card from hand
    discardCard(handPos, currentPlayer, state, 0);

    //trash copies of cards returned to supply
    for (j = 0; j < choice2; j++)
```

```

case ambassador:
    j = 0;          //used to check if player has enough cards to discard

    if (choice2 > 2 || choice2 < 0)
    {
        return -1;
    }

    if (choice1 == handPos)
    {
        return -1;
    }

    for (i = 0; i < state->handCount[currentPlayer]; i++)
    {
        if (i != handPos && i == state->hand[currentPlayer][choice1] && i != choice1)
        {
            j++;
        }
    }
    if (j < choice2)
    {
        return -1;
    }

    if (DEBUG)
        printf("Player %d reveals card number: %d\n", currentPlayer, state->hand[currentPlayer][choice1]);

    //increase supply count for choosen card by amount being discarded
    state->supplyCount[state->hand[currentPlayer][choice1]] += choice2;

```

### Bugs:

In the baron card function, the number in comparison for choice1 was changed from 0 to 1. This means that the Boolean instead of discarding when false, is now discarding when true. The second but introduced was erasing the “else” part of the function, therefore not increasing to the next card iterator.

In the minion card function, the number of actions was not increased therefore no action is seen as taken. The second bug introduced is the discard hand if the number of cards in hand is greater than 0. This means that the card will be discarded rather than kept.

In the ambassador function, the check for seeing if the player has enough cards to discard is 1 rather than 0. This will be compared to choice2 and therefore the comparison will be one card off. The second bug is the deletion of the DEBUG function.

In the tribute function, the comparison for discard count + deck count was changed from 1 to 0. The else for increasing actions was completely deleted so no action card has been noted to be found.

In the mine function, the return for one of the if statements was changed from -1 to 0. The discardCard was also changed from 0 to 1.