Emily Sorg  Assignment 2 10/13

**Refactor:**

For the refactor, I kept it fairly simple:

- Moved the code under the switch statements to their own functionscalled $(cardname)Card
- Passed in necessary parameters from cardEffect and added  currentPlayer when necessary
- Renamed function choice parameter names to make more sense on a card-basis
- Added in declarations for i/j
- Moved tributeRevealedCards declaration out of the cardEffect function

**Bugs:**

**Baron**

1. I elected to just not return the result of baronCard and just return 0 no matter what.  I chose this because I actually committed like this and it's a genuine bug that I myself created.

```
baronCard(choice1, state, currentPlayer);

return 0;
```

2. Commented out the part where hand is decremented on discarding an estate.  This might come up if another card is used to draw cards, and not enough new cards are added.  Chosen this since it seems like something I'd forget to do.

```
//  state->hand[currentPlayer][state->handCount[currentPlayer]] = -1;
```

**Minion**

1. Commented out the part that checks if opposing players have at least 5 cards.  This would be uncovered if they had been forced to discard down cards and would then have the benefit of getting to redraw. Chosen since this would be less likely to come up from random or manual testing.

```
//other players discard hand and redraw if hand size > 4
        for (i = 0; i < state->numPlayers; i++)
        {
            if (i != currentPlayer)
            {
//                if ( state->handCount[i] > 4 )
//                {
```

2. Made option one give 3 coins instead of 2.  Chosen to cover gold mechanics.

```
if (choiceGold) {
    state->coins = state->coins + 3
```

**Ambassador**

1. Removed the test to see if the card being looked at is equal to the position of the card to be discarded, such that it will usually end up being one higher.  The card will now not return -1 if

the player asks to discard one more card than he has, so the supply of a card can now go up beyond its maximum rate. Chosen to test extreme edgecase. (There is another bug here that would have made this not work but supposing you fix that and make it this):

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

2. Changed trash flag to 0 on the discardCard call, so they never end up trashing the card. Will be noticed by players if they trashed something unsavory and expected to never see it again. Chosen to cover discard testing.

```
if (state->hand[currentPlayer][i] == state->hand[currentPlayer][toDiscard]) {
  discardCard(i, currentPlayer, state, 0);
  break;
```

**Tribute**

1. Removed one drawCard reward. Would be noticed by disappointed players. Chosen to cover drawing testing.

```
else if (tributeRevealedCards[i] == estate ...)
 { //Victory Card Found
    drawCard(currentPlayer, state);
```

2. Removed a deck count de-increment in the scenario the next player has no cards in their deck. Will likely cause issues if they shuffle and get to the end of their new deck as deck counts get out of sync. Chosen to cover deck counts.

```
     shuffle(nextPlayer,state);//Shuffle the deck
    }
    tributeRevealedCards[0] = state->deck[nextPlayer][state->deckCount[nextPlayer]-1];
    state->deck[nextPlayer][state->deckCount[nextPlayer]--] = -1;
    //state->deckCount[nextPlayer]--;
```

**Mine**:

1. Changed the upper bound of the enum of which card to gain to estate, meaning the player will be unable to gain an estate if they wish. This is likely to only ever come up in natural play if the player uses this card try exhaust the supply of estates pile to foce endgame, or in random testing. Chosen to cover game function.

```
    if (choiceCardToGain > treasure_map || choiceCardToGain < estate)
    {
        return -1;
```

2. I thought it would be fun for the last one to just not fix the bug in mine where it's not actually setting the trash flag for the trashed treasure. Would always come up when played properly.

```
  discardCard(i, currentPlayer, state, 0);
```