

# **PROJECT 1**

## **[Card Game: WAR]**

Andrew Adame

CSC-17A

10/25/2020

## ● Introduction

### ○ TITLE: WAR

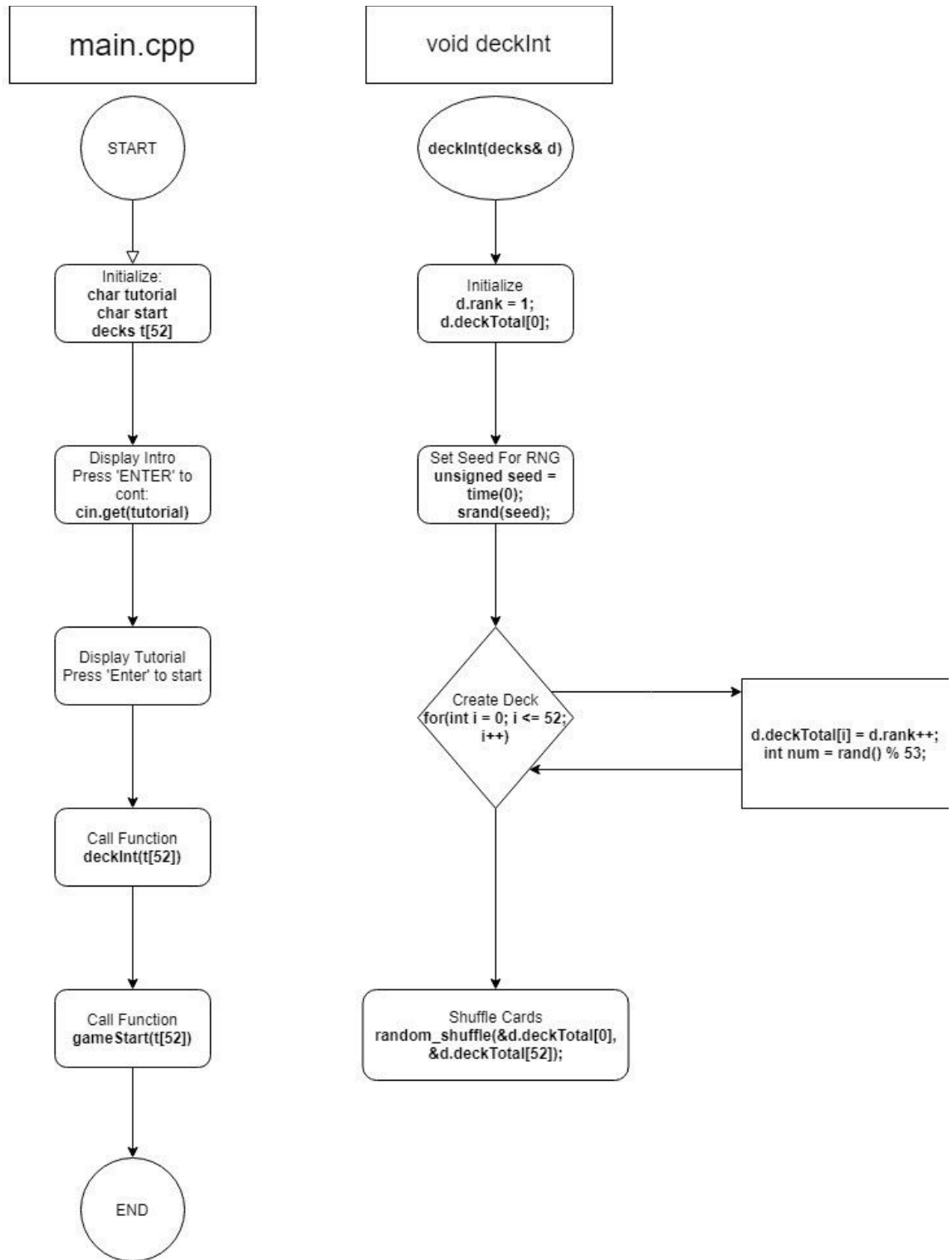
- This is an incomplete program meant to simulate one of my favorite childhood card game War. It is a basic luck-based card game where two players split a deck equally amongst each other. They each draw a card from their decks and place them face up . Whoever's card has the highest rank (from Ace to King) wins the round, and loses their card. Meanwhile, the loser takes back their card as well as their opponent's and places them at the bottom of their deck. The objective is to be the first player to get rid of their deck. If both players draw cards with the same rank, they enter a state of "War" and must draw 3 cards, two face down and the last face up. The loser has to take all 8 cards. If the players tie again, the state of War repeats until there is a loser.

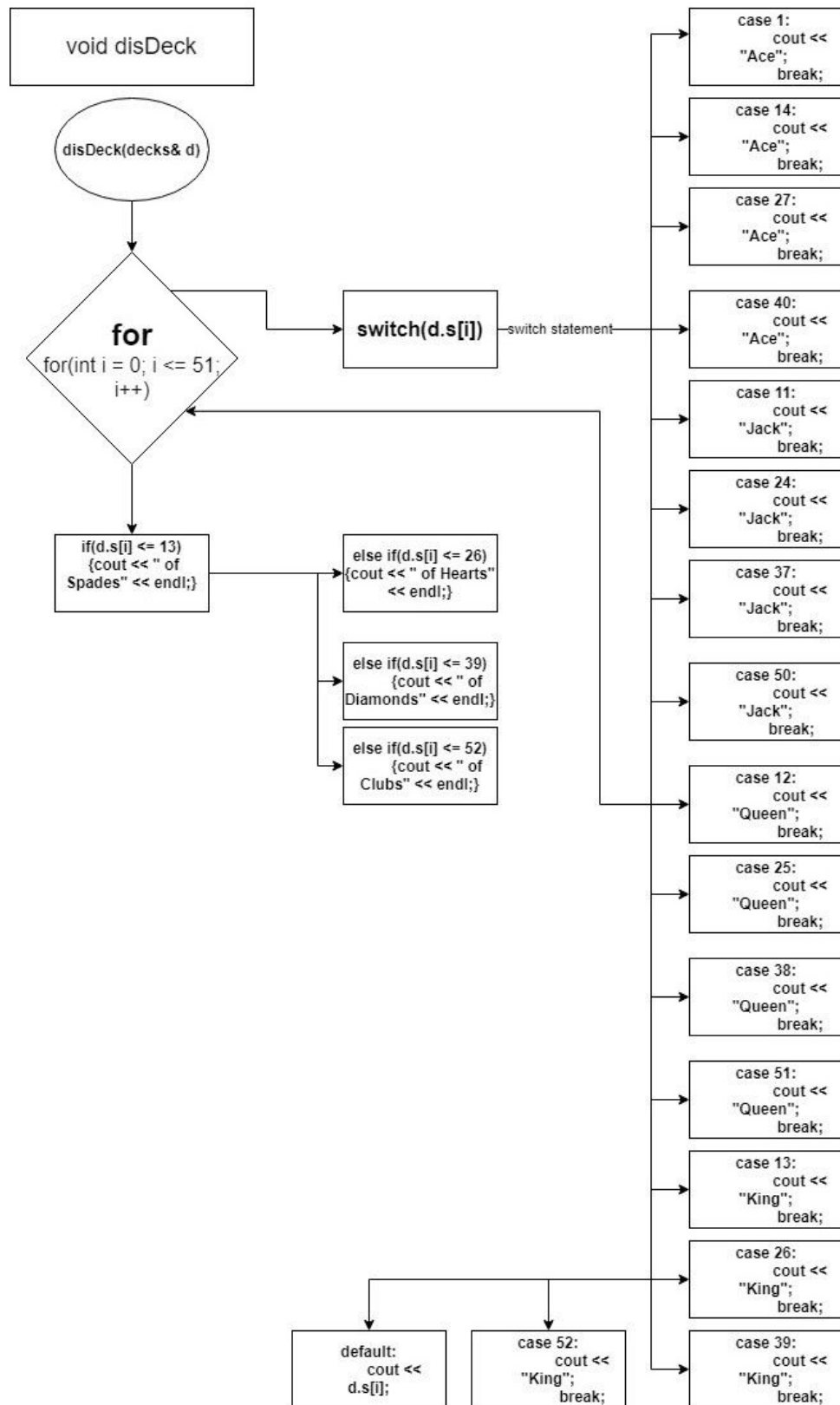
### ○ SUMMARY

- **Project Size:** 182 Lines
- **Number of Variables:**
- This project contains concepts learned from the chapters of the book, including structures and allocation of arrays. I also used the `random_shuffle` function to randomly rearrange the arrays. Unfortunately I did not complete the project due to my poor time management, even when given a time extension. This was, however, definitely an informative learning experience. I had two ideas as to how the program would've worked; one was that there would be three arrays; a total deck, player

deck, and opponent deck. The idea was that the deck would be initialized, shuffled, then divided evenly into the two arrays. If the player wins, they lose one element and the opponent gains one. The other idea was based on the deck you made during classes, using the character arrays to represent the face values and ranks, with actual ranks representing the cards is an array of 52 elements.. The idea I went with was very similar, but instead of character arrays, I was using a series of case statements that go into use once it is time to display the cards to the player. The one constant throughout these ideas was that the game ends either after a limit (which would be a 1000 plays being measured by an increment variable in a while loop, because War can sometimes last a very long time), if the player or the opponent has no more elements in their arrays, or if the player decides to quit, which then chooses the winner based on who has the least amount of elements left in their arrays. These unfinished ideas are located in the ProjectVersion folder. What the program is in its current state is a basic deck of cards that shuffle and display all 52 cards in a random order. I have a more completed version in the Project Versions folder (mainv4), which was an attempt draw card for both the player and opponent, but decided to turn in a simpler code as the other version was not working.

○ **FLOWCHART**





○ **PROGRAM**

/\*

\* File: main.cpp

\* Author: Andrew Adame

\* Purpose: The purpose of this program is to simulate

\* the card game "War" using a variety of things I have learned

\* in CSC-17A so far such as structures and dynamic arrays.

\*

\* Created on October 13, 2020, 9:17 AM

\*/

#include <cstdlib>

#include <iostream>

#include <iomanip>

#include <ctime> //For seed

#include <algorithm> //For shuffle

using namespace std;

struct decks

{

int rank; //Rank of Card

```

int deckTotal [52]; //Total Deck Array

int s[13];          //Suit Array

int deckPlyr [26]; //Player Deck

int deckOpp [26];  //Opponent Deck

};


void deckInt(decks&);

void disDeck(decks&);

int main(int argc, char** argv)
{
    decks t[52];

    char tutorial;

    char start;

    cout << "Welcome to Andrew's Game of War!" << endl;

    cout << "_____ " << endl;

    cout << "Press 'ENTER' to read the tutorial!";

    cin.get(tutorial);


    cout << "Here are the rules:" << endl;

    cout <<
    "
    _____
    _____ " << endl;

    cout << "You will be facing off against the computer." << endl;

```

```
cout << "Each of you will split the deck, and receive ";

cout << "26 cards, none of which you or your opponent are allowed to see." << endl;

cout << "When the game starts, you and your opponent will draw a card." << endl;

cout << "The program will directly inform you the card's rank and suit." << endl;

cout << "(Although in this game, only your card's rank matters)" << endl;

cout << "When you and your opponent's cards are displayed, whoever has the highest value
card" << endl;

cout << "will win the round and lose their card, while the loser gains both cards in their deck."
<< endl;

cout << "(Order of value goes like so; 2,3,4,5,6,7,8,9,10, Jack, Queen, King, Ace)" << endl;

cout << "The objective of the game is to run out of cards in your deck." << endl;

cout << "If both you and your opponent draws the same rank card, the two " << endl;

cout << "of you will enter a state of WAR!" << endl;

cout << "The program will draw three cards from each of your decks, and only display the
face/value of the third card." << endl;

cout << "Whoever has the highest rank card will lose the three cards, and the loser will gain all
six." << endl;

cout << "If you and your opponent tie again, this process will repeat until one of you
eventually loses." << endl;

cout <<
"
_____
_____ " << endl;

cout << "Press 'ENTER' to start!" << endl;
```



```
cin.get(start);

deckInt(t[52]);

disDeck(t[52]);

return 0;
}

//INITIALIZING DECK

void deckInt(decks& d)
{
    //INITIALIZE ARRAY

    d.rank = 1;

    d.s[0];

    //SET SEED

    unsigned seed = time(0);

    srand(seed);

    //CREATE DECK

    for(int i = 0; i <= 52; i++)
    {
        d.s[i] = d.rank++;
    }
}
```

```

        int num = (rand() % 100) + 1;
    }

//SHUFFLE CARDS

    random_shuffle(&d.s[0], &d.s[52]);

}

void disDeck(decks& d)
{
    for(int i = 0; i <= 51; i++)
    {
        //Labeling Face Cards

        switch(d.s[i])
        {
            //ACES/////////////////

            case 1:

                cout << "Ace";

                break;

            case 14:

                cout << "Ace";

                break;

            case 27:

```

```
        cout << "Ace";

        break;
case 40:

        cout << "Ace";

        break;

////////////////////////////////

//JACKS////////////////////////////////

case 11:

        cout << "Jack";

        break;
case 24:

        cout << "Jack";

        break;
case 37:

        cout << "Jack";

        break;
case 50:

        cout << "Jack";

        break;

////////////////////////////////

//QUEENS////////////////////////////////
```

case 12:

cout << "Queen";

break;

case 25:

cout << "Queen";

break;

case 38:

cout << "Queen";

break;

case 51:

cout << "Queen";

break;

//////////

//KINGS//////////

case 13:

cout << "King";

break;

case 26:

cout << "King";

break;

case 39:

cout << "King";

```
        break;

    case 52:

        cout << "King";

        break;

    //////////////////////////////////

    //Numbered Cards

    default:

        cout << d.s[i];

    }


    if(d.s[i] <= 13)

    {

        cout << " of Spades" << endl;

    }

    else if(d.s[i] <= 26)

    {

        cout << " of Hearts" << endl;

    }

    else if(d.s[i] <= 39)

    {

        cout << " of Diamonds" << endl;
```

```
}  
else if(d.s[i] <= 52)  
{  
    cout << " of Clubs" << endl;  
}  
cout << endl;  
}  
}
```