Project 1

<Quadruple War>

CIS-5

Luke Gleason

02/03/22

Introduction

Each player turns up a card at the same time and the player with the higher card takes both cards and puts them, face down, on the bottom of his stack.

If the cards are the same rank, it is War. Each player turns up one card face down and one card face up. The player with the higher cards takes both piles (six cards). If the turned-up cards are again the same rank, each player places another card face down and turns another card face up. The player with the higher card takes all 10 cards, and so on.

The winner of the game is decided when one player has all of the other players cards.

Summary

Project Size: ~250 lines

When I think of card games, I think back to middle school when I'd play my favorite card game — Quadruple War. Although the game currently isn't as fun through a program, I have successfully completed a no-frills version of the game.

There are some key features missing from the game currently due to my limited knowledge. Right now, the program doesn't actually exchange the card between the players. During project 2, I hope to solve this using arrays. By doing so, this will allow the probability to become more akin to the "real-life" version of the game.

Although, the base game is there. The program allows you to choose a player, then play the card game. If the two players card are equal, then they enter war. I hope you enjoy!

Example Output

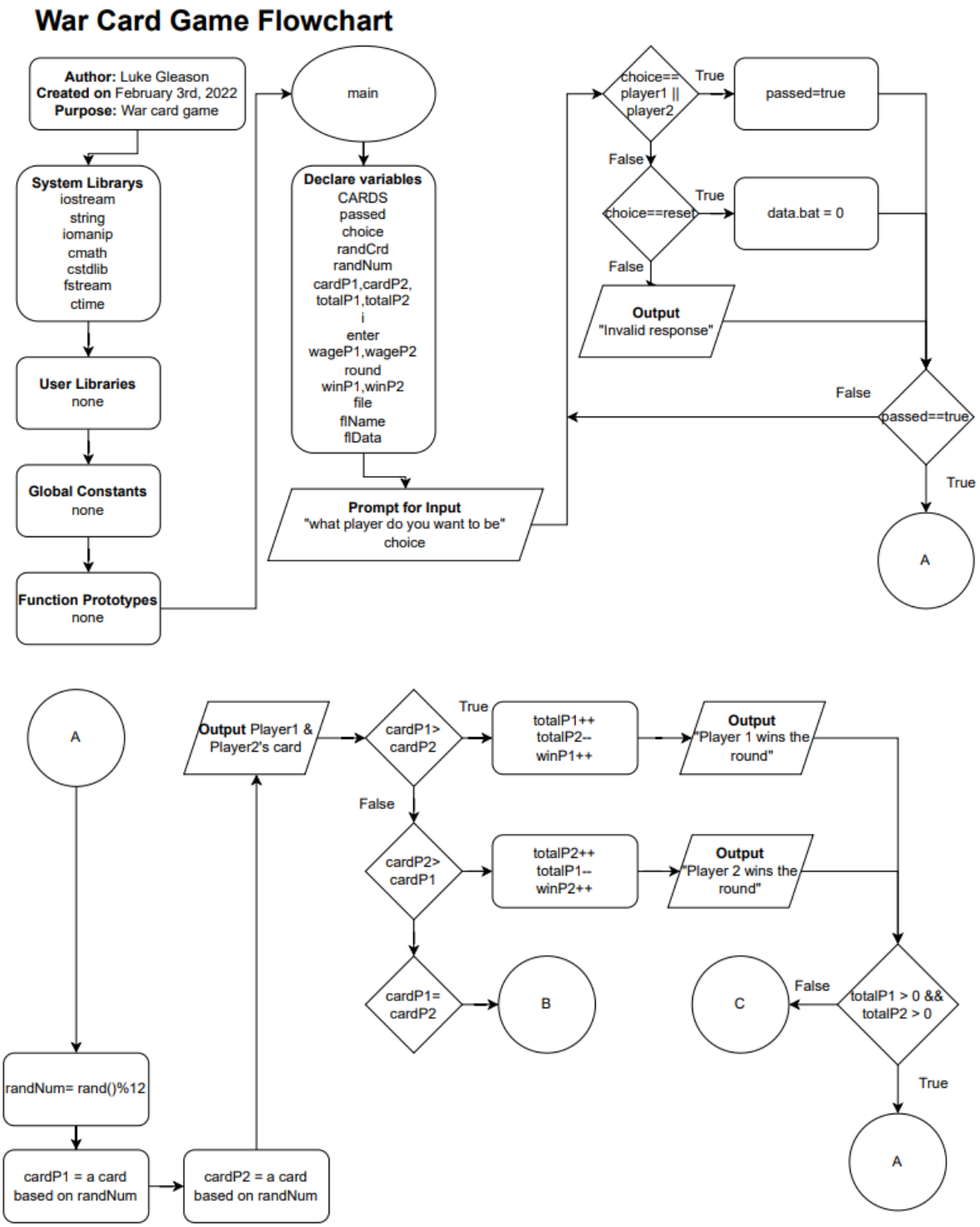
```
Welcome to the game of Quadruple War!
Would you like to play as Player 1 or Player 2?
Type 'player1' or 'player2'
Type 'reset' to reset your game counter
incorrect
Invalid input. Please enter 'player1' OR 'player2'
pLayeR2




Player 1:     Player 2:      Round:1
*********     *********
*       *     *       *
*   J   *     *   6   *
*       *     *       *
*********     *********
Player 1 wins the round!
Player 1 total cards:   9
Player 2 total cards:   7
Round Complete. Type anything & press enter to continue.
```

```
Player 1:     Player 2:      Round:4
*********     *********
*       *     *       *
*   J   *     *   J   *
*       *     *       *
*********     *********
Both players have the same card! Time for WAR.
Player 1 will wager the full 4 cards
Player 2 will wager the full 4 cards
Type anything & press any key to continue.
f
Player 1:     Player 2:      Round:4
*********     *********
*       *     *       *
*   5   *     *   7   *
*       *     *       *
*********     *********
Player 2 wins the war!
Player 1 total cards:   5
Player 2 total cards:   11
Round Complete. Type anything & press enter to continue.
```

Flowchart


War Card Game Flowchart
Author: Luke Gleason
Created on February 3rd, 2022
Purpose: War card game
main
System Librarys
iostream
string
iomanip
cmath
cstdlib
fstream
ctime
User Libraries
none
Global Constants
none
Function Prototypes
none
Declare variables
CARDS
passed
choice
randCrd
randNum
cardP1,cardP2,
totalP1,totalP2
i
enter
wageP1,wageP2
round
winP1,winP2
file
flName
flData
Prompt for Input
"what player do you want to be"
choice
choice== player1 || player2
True
passed=true
False
choice==reset
True
data.bat = 0
False
Output
"Invalid response"
False
passed==true
True
A
A
randNum= rand()%12
cardP1 = a card based on randNum
cardP2 = a card based on randNum
Output Player1 & Player2's card
cardP1> cardP2
True
totalP1++
totalP2--
winP1++
Output
"Player 1 wins the round"
False
cardP2> cardP1
totalP2++
totalP1--
winP2++
Output
"Player 2 wins the round"
cardP1= cardP2
B
C
False
totalP1 > 0 && totalP2 > 0
True
A


B
totalP1>4
True
totalP1-=4
wageP1=4
totalP2>4
totalP2-=4
wageP2=4
randNum= rand()%12
False
wageP1=totalP1
totalP1=0
wageP2=totalP2
totalP1=0
cardP1 = a card based on randNum
cardP2 = a card based on randNum
cardP1> cardP2
True
totalP1 = totalP1 + wageP1 + wageP2
winP1++
Output "Player 1 wins the war"
totalP1 > 0 && totalP2 > 0
A
False
cardP2> cardP1
True
totalP2 = totalP2 + wageP1 + wageP2
winP2++
Output "Player 2 wins the war"
C
False
cardP1= cardP2
True
totalP1+=wageP1
totalP2+=wageP2
Output "No one wins"
B
C
totalP1>0
True
Output "Player 1 won!"
read data.bat
store line as flData
write data.bat
False
Output "Player 2 won!"
return 0
exit from main
Close data.bat
write flData+1

Cross Reference for Project 1

Chapter	Section	Topic	Where Line #"s	Pts	Notes
2	2	cout	53-56, 73		
	3	libraries	10 through 16	8	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals	35-49		No variables in global area, failed project!
	5	Identifiers	35, 79		
	6	Integers	42, 49	3	
	7	Characters	40	3	
	8	Strings	37, 38, 43	3	
	9	Floats No Doubles	45, 46	3	Using doubles will fail the project, floats OK!
	10	Bools	36	4	
	11	Sizeof *****	n/a		
	12	Variables 7 characters or less	35-49		All variables <= 7 characters
	13	Scope ***** No Global Variables	n/a		
	14	Arithmetic operators	125, 126, 132, 133		
	15	Comments 20%+	66, 227	5	Model as pseudo code
	16	Named Constants	35		All Local, only Conversions/Physics/Math in Global area
	17	Programming Style ***** Emulate	n/a		Emulate style in book/in class repositiory
3	1	cin	59, 163		
	2	Math Expression	230, 240		
	3	Mixing data types ****	n/a		
	4	Overflow/Underflow ****	n/a		
	5	Type Casting	186,190	4	
	6	Multiple assignment *****	n/a		

	7	Formatting output	233-241	4	
	8	Strings	37, 38, 43	3	
	9	Math Library	86	4	All libraries included have to be used
	10	Hand tracing ******	n/a		
4	1	Relational Operators	61, 67, 212		
	2	if	201, 207	4	Independent if
	4	If-else	152-160	4	
	5	Nesting	58-79	4	
	6	If-else-if	67-73	4	
	7	Flags *****	n/a		
	8	Logical operators	67	4	
	11	Validating user input	58-79	4	
	13	Conditional Operator	228	4	
	14	Switch	87-99	4	
5	1	Increment/Decrement	82, 209	4	
	2	While	81	4	
	5	Do-while	58-79	4	
	6	For loop	61-64	4	
	11	Files input/output both	233-239	8	
	12	No breaks in loops ******	n/a		Failed Project if included
****** Not required to show			Total	100	

Program

```
* File:   main.cpp
 * Author: Luke Gleason
 * Created on February 2, 2022, 4:06PM
 * Purpose:  War Project 1 Version 2
 *           Adding endgame data
 */

//System Level Libraries
#include <iostream>  //Input-Output Library
#include <string>    //String Library
#include <iomanip>   //Formatting Library
#include <cmath>     //Math Library
#include <cstdlib>   //General Purpose Library
#include <fstream>   //File Read/Write Library
#include <ctime>     //Time library

using namespace std;

//User Defined Libraries

//Global Constants, not Global Variables
//These are recognized constants from the sciences
//Physics/Chemistry/Engineering and Conversions between
//systems of units!

//Function Prototypes
```

```
//Execution begins here!
int main(int argc, char** argv) {
    //Initialize Random Seed once here!
    srand(static_cast<unsigned>(time(0)));

    //Declare + Initialize Variables
    const short CARDS = 8;
    bool passed = false;              //INTRO -- Used to validate user input.
    string choice;                   //INTRO -- Stores user's choice to be player one OR player two
    string randCrd;                   //GAME -- Stores random card
    int randNum;                      //Game -- Random number to choose card
    char cardP1=0,cardP2=0;           //GAME -- Stores each player's card
    short totalP1=CARDS,totalP2=CARDS;  //GAME -- Keeps track of number of cards.
    int i;                           //GAME -- Loop increment
    string enter;                    //GAME -- Used to add a break between sequences
    short wageP1,wageP2;              //GAME -- Wager for WAR
    float round=0;                    //GAME -- used to determine the round
    float winP1=0, winP2=0;           //GAME -- tallies each player's wins for calculation at end.
    fstream file;                    //FILE -- fstream
    string flName = "data.dat";       //FILE -- stores file name
    int flData;                      //FILE -- stores file data


    //INTRO -- Prompt the user to choosing a player
       cout<<"Welcome to the game of Quadruple War!"<<endl;
       cout<<"Would you like to play as Player 1 or Player 2?"<<endl;
       cout<<"Type 'player1' or 'player2'"<<endl;
```

```
    cout<<"Type 'reset' to reset your game counter"<<endl;

do { //Do-while loop to repeat until user provides proper input
    cin>>choice;
    //Input validation -- convert input to all uppercase
    for(int i=0;choice[i]!='\0';i++) {
      if(choice[i]<=122 && choice[i]>=97) //'122' = 'z' & '97' == 'a'
        choice[i]-=32;                  // Subtracting 32 converts to upper
    }

    //Input validation -- check if user inputted supported string
    if (choice == "PLAYER1" || choice == "PLAYER2"){
        passed = true;
    } else if (choice == "RESET") {
        file.open(flName, ios::out); //Opening file to write game number
        file<<0;
        file.close();
        cout<<"Successfully reset game counter. Please enter 'player1' OR 'player2'"<<endl;
    } else {
        //If user input invalid response, then send them a message
        //to inform them on the correct response.
        cout<<"Invalid input. Please enter 'player1' OR 'player2'"<<endl;
    }
} while (passed == false);

while (totalP1 > 0 && totalP2 > 0){
    round++; //round incrementor
    //Card generation
```

```
i=0;
while (i<=2) {
    randNum = rand()%12;
    switch (randNum){
        case 1: randCrd = "1";break;
        case 2: randCrd = "2";break;
        case 3: randCrd = "3";break;
        case 4: randCrd = "4";break;
        case 5: randCrd = "5";break;
        case 6: randCrd = "6";break;
        case 7: randCrd = "7";break;
        case 8: randCrd = "8";break;
        case 9: randCrd = "9";break;
        case 10: randCrd = "J";break;
        case 11: randCrd = "K";break;
        case 12: randCrd = "Q";break;

    }
    // Give random card to either player 1 or player 2
    if (i == 0){ //give card to player 1
        cardP1 = static_cast<char>(randCrd[0]);

    }
    if (i == 1){ //give card to player 2
        cardP2 = static_cast<char>(randCrd[0]);
    }
    i++;
}
```

```
//Display each player's cards.
cout<<endl<<endl<<endl<<endl;
cout<<"Player 1:     Player 2:      "<<"Round:"<<round<<endl;
cout<<"*********     *********"<<endl;
cout<<"*       *     *       *"<<endl;
cout<<"*   "<<cardP1<<"   *     *   "<<cardP2<<"   *"<<endl;
cout<<"*       *     *       *"<<endl;
cout<<"*********     *********"<<endl;

//Compare the cards using their ASCII value
//We can do this because cardP1&cardP2 are chars
if (cardP1 > cardP2){ //If Player1's card is greater
   totalP1++; //add 1 to player one total
   totalP2--; //subtract 1 from player one total
   winP1++;
   cout<<"Player 1 wins the round!"<<endl;
}

if (cardP1 < cardP2){ //If Player 2's card is greater
   totalP1--; //add 1 to player one total
   totalP2++; //subtract 1 from player one total
   winP2++;
   cout<<"Player 2 wins the round!"<<endl;
}

wageP1=0; //reset the wage for the beginning of war
wageP2=0;
```

```
while (cardP1 == cardP2){ //If Player 1 & 2's card are equal
   cout<<"Both players have the same card! Time for WAR."<<endl;
   if(totalP1<4){ //Set P1 to 1 if then don't have enough cards for full war
      wageP1=totalP1; //sets current total to player wage
      totalP1=0;      //since wage = current total, set total to 0
      cout<<"Player 1 doesn't have enough cards! They will wager "<<wageP1<<endl;
   } else {
      totalP1-=4; //wage 4 cards.
      wageP1=4;
      cout<<"Player 1 will wager the full 4 cards"<<endl;
   }

   if(totalP2<4){ //Set P2 to 1 if then don't have enough cards for full war
      wageP2=totalP2;   //sets current total to player wage
      totalP2=0;        //since wage = current total, set total to 0
      cout<<"Player 2 doesn't have enough cards! They will wager "<<wageP2<<endl;
   } else {
      totalP2-=4;
      wageP2=4;
      cout<<"Player 2 will wager the full 4 cards"<<endl;
   }

   cout<<"Type anything & press any key to continue."<<endl;
   cin>>enter;

   //Random card generation for war
   //Next update will put this into a function.
   i=0;
```

```
while (i<=2) {
   randNum = rand()%12;
   switch (randNum){
      case 1: randCrd = "1";break;
      case 2: randCrd = "2";break;
      case 3: randCrd = "3";break;
      case 4: randCrd = "4";break;
      case 5: randCrd = "5";break;
      case 6: randCrd = "6";break;
      case 7: randCrd = "7";break;
      case 8: randCrd = "8";break;
      case 9: randCrd = "9";break;
      case 10: randCrd = "J";break;
      case 11: randCrd = "K";break;
      case 12: randCrd = "Q";break;
   }
// Give random card to either player 1 or player 2
if (i == 0){ //gives card to player 1 first
   cardP1 = static_cast<char>(randCrd[0]);

}
if (i == 1){ //gives card to player 2
   cardP2 = static_cast<char>(randCrd[0]);
}
i++;
}
cout<<"Player 1:     Player 2:      "<<"Round:"<<round<<endl;
cout<<"*********     *********"<<endl;
```

```
        cout<<"*       *     *       *"<<endl;
        cout<<"*   "<<cardP1<<"   *     *   "<<cardP2<<"   *"<<endl;
        cout<<"*       *     *       *"<<endl;
        cout<<"*********     *********"<<endl;

        if (cardP1 > cardP2){ //If Player1's card is greater
            totalP1 = totalP1 + wageP1 + wageP2;
            winP1++;
            cout<<"Player 1 wins the war!"<<endl;
        }

        if (cardP1 < cardP2){ //If Player 2's card is greater
            totalP2 = totalP2 + wageP1 + wageP2;
            winP2++;
            cout<<"Player 2 wins the war!"<<endl;
        }
        if (cardP1 == cardP2) { //If player 1 = player 2.
            totalP1+=wageP1;    //Give player1 back their wage
            totalP2+=wageP2;    //Give player2 back their wage
            cout<<"Wow! Both players have the same card. We enter another round of
war!"<<endl;
            cout<<"Type anything & press any key to continue."<<endl;
            cin>>enter;
        }
    }

    cout<<"Player 1 total cards:\t"<<totalP1<<endl;
    cout<<"Player 2 total cards:\t"<<totalP2<<endl;
    cout<<"Round Complete. Type anything & press enter to continue."<<endl;
```

```
        cin>>enter;


    }
        //If the while loop is over, that means one player has zero points.
        totalP1>0 ? cout<<endl<<"Congrats! Player 1 won!"<<endl : cout<<endl<<"Congrats!
Player 2 won!"<<endl;
        cout<<showpoint<<setprecision(2);
        cout<<"Player 1 Rounds Won: "<<winP1<<" ("<<winP1/round<<"%) "<<endl;
        cout<<"Player 2 Rounds Won: "<<winP2<<" ("<<winP2/round<<"%) "<<endl;


        file.open(flName, ios::in); //Opening file to access old game number
        file >> flData;
        cout<<fixed<<setprecision(0);
        cout<<"Game Number: "<<flData + 1<<endl;
        file.close();


        file.open(flName, ios::out); //Opening file to write game number
        file<<flData + 1;
        file.close();


    //Exit the program
    return 0;
}
```