

War/I Declare War Card Game

Laurie Guimont

CSC 5

Summer 2016

45276

TABLE OF CONTENTS

<u>INTRODUCTION</u>	<u>3</u>
<u>HOW THE CARD GAME WORKS</u>	<u>3</u>
OBJECT OF THE GAME	3
RULES OF THE GAME	3
HOW TO WAGE WAR	3
<u>MY APPROACH TO THE GAME</u>	<u>4</u>
TRANSLATING GAME PLAY RULES TO PROGRAMMING LANGUAGE	4
SIMILARITIES TO THE CARD GAME	4
DIFFERENCES FROM THE CARD GAME	5
<u>THE LOGIC OF IT ALL</u>	<u>5</u>
FLOWCHART	5
<u>CONSTRUCTS & CONCEPTS UTILIZED</u>	<u>9</u>
IOSTREAM LIBRARY	9
CSTDLIB LIBRARY	10
CTIME LIBRARY	10
IOMANIP LIBRARY	10
STRING LIBRARY	10
CMATH LIBRARY	10
FSTREAM LIBRARY	10
DATA TYPES	11
CONDITIONAL STATEMENTS	11
LOOPS	11
<u>PROOF OF A WORKING PRODUCT</u>	<u>11</u>
<u>REFERENCES</u>	<u>15</u>
<u>PROGRAM</u>	<u>15</u>

Introduction

*“WAR! huh! Yeah,
What is it good for? Absolutely...”
–Edwin Starr*

...any good pass time with friends and family (the card game, that is).

War, also referred to by some as I Declare War, is a popular 2-Player standard card game. The game is played by people of all ages, and can serve as a useful way of helping young children learn how to count. All the players are required to do is display the cards they are dealt and compare them. Although the game does not require much logic and may be considered simplistic to some, the game is still extremely competitive if the right cards are dealt in the right order, or in other words, in an order that helps players beat their opponent. This dynamic of having cards randomly arranged in a way that benefits either player appealed to me and inspired me to write a program around the game.

How the Card Game Works

Object of the Game

To accumulate all 52 cards.

Rules of the Game

War is typically a two person game. The game is very simple:

1. Shuffle and deal the cards evenly between the two players. Therefore, each player should have 26 cards. Jokers are not used in this game.
2. Players should then turn over the top cards in their pile at the same time. Whoever has the higher value card wins both cards. The ranks of cards are as follows:
 - All number cards are valued according to their number.
 - Of the face cards the Ace is the highest overall card, followed by the King, then Queen, and the Jack is the lowest ranked face card. Face cards beat number cards.
3. Keep playing until one of the players has collected all of the cards in the deck.

How to Wage War

If the players turn over cards that have the same card value, war is waged! At this point, both players must place 2 to 4 cards faced down, then turn over the proceeding card. Whichever

player has the higher war card gets all the cards put down, including the cards faced down and the cards that initiated the war.

Note: The number of cards placed faced down before overturning one is based upon player preference. I have seen games played where only 2 cards were placed faced down, but have also seen games where 4 cards were placed down. The latter is interesting because while placing your cards down, the players count and say aloud, "1, 2, 3, 4" then proceed with "I declare war" while overturning the fifth card at the same time the word "war" is said.

Note: More than one war can be declared in a round. If players throw the same war card down, another round of "faced down" cards must be placed along with another war card. This process should be repeated until one of the players has a higher war card.

My Approach to the Game

Translating Game Play Rules to Programming Language

While thinking about how I was going to program this game, a couple questions arose:

- "Since the card game has four suites, meaning four of each card, how do I tell the computer that I want to limit the number of times a random number is chosen?"
- "Should I have the computer 'deal' 26 cards to the user/player, and then have the player chose from their 'hand'?"
- "How will a player win or lose the game?"

After a couple of hours of planning my program and toiling with the above questions, I did some research and found that most (if not all) of the questions I was asking myself had a common answer: arrays. Well, I hadn't learned arrays at that point. Therefore, I had to come up with a way to cope with these problems using the constructs and concepts that I already knew.

Similarities to the Card Game

My War program follows the same rules of play as the card game:

- The user and the computer "throw down" a card, then it is determined who has the higher card
- If the same card is thrown down, both user and computer place cards faced down, then reveal the war card.
 - This is repeated if the war cards are the same.

Differences from the Card Game

The main difference from the card game is the score. I decided that in order to determine the winner of each round in the game, I would numerically value each card and add up points.

- If you win, you gain the value of the card you put down as well as the value of the card the computer put down.
- If you lose, you lose the value of the card you put down. The same goes for a computer loss.

Since the face cards don't have a numeric value in the regular card game, I assigned a value to each according to their rank in the game. Therefore, the Jack is valued at 11 points and the Ace is valued at 14 points, with the Queen at 12 points, and the King at 13.

In response to my concern about limiting the number of times a card is chosen, I decided to not worry about that and treat the game as if two player were not being dealt cards, but instead were drawing them and putting them back in the pile, all while keeping score. It's like "War, with Replacements."

Finally, a player wins or loses based on their score at the time they decide to finish the game. If the player is tired of playing after a while, they can exit the game and the computer will tell them their final score. If the score is higher than the computer's, they have one. If not, they have lost.

The Logic of it All

Flowchart

Since my flowchart is extremely long, I will break it up into smaller pieces and accompany it with pseudocode here. To view my complete flowchart, please visit:

<http://www.gliffy.com/go/publish/10902783>

Put in opening comments

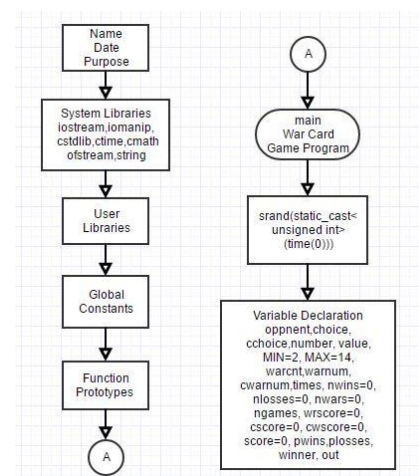
Bring in system libraries

Enter main, then immediately

set random number seed

Declare all variables, initiate some

now and some later.



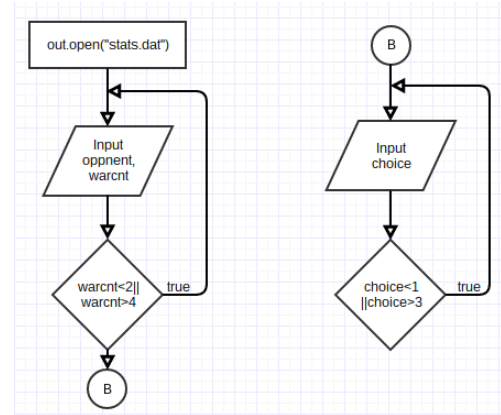
Open file that we will write data to

Input opponent name and number of faced down cards

Loop this step until valid data is inputted.

Input valid menu choice.

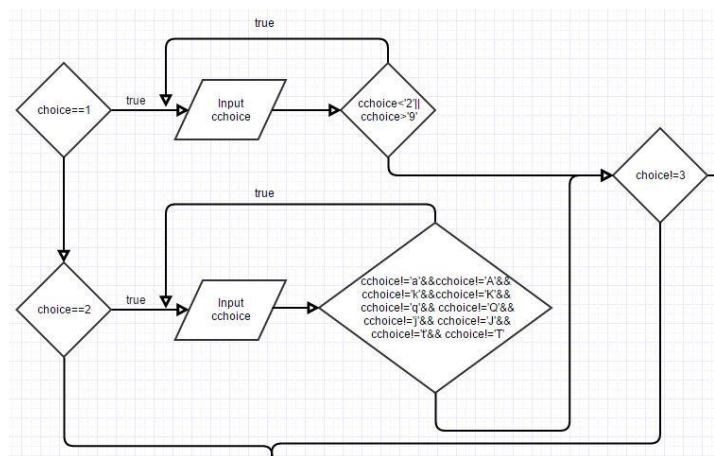
Verify valid data with while loop again



If “number card” or “face card”,

input card choice

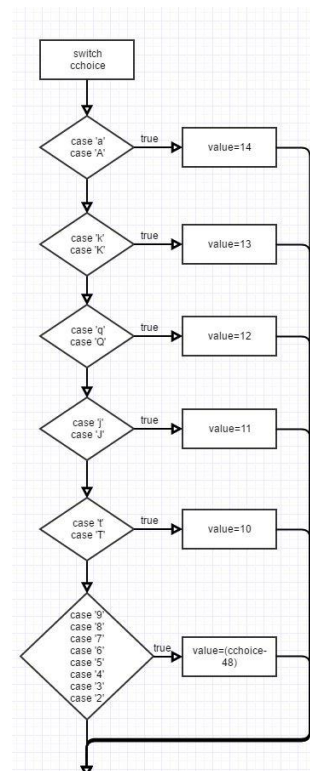
Verify that all data is valid.



As long as “End Menu” is not chosen,

computer processes input by applying

assigned value in switch statement



Computer chooses a random number and
compares number to input value

If value is bigger than random number

User wins round and

Score is calculated

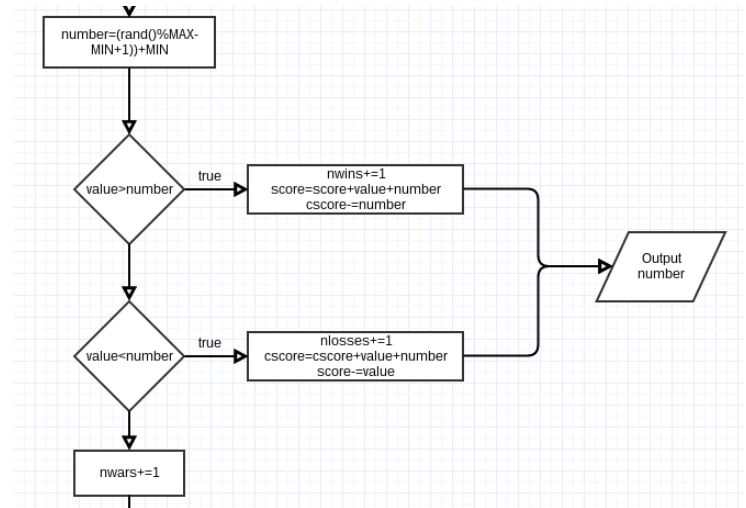
If value is smaller, computer wins.

Computer wins round

Score is calculated

If both numbers are equal

User has entered war



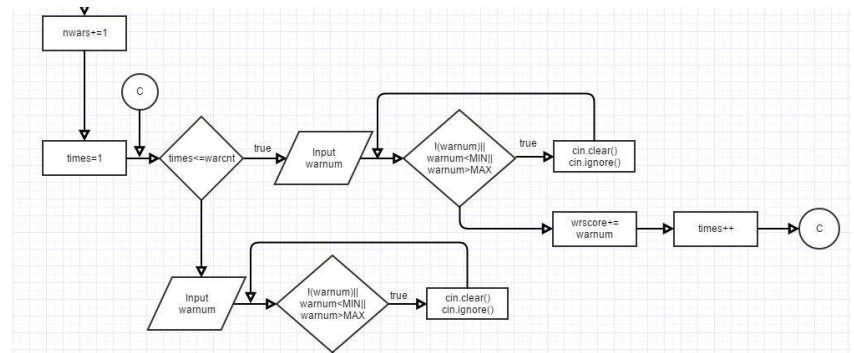
Prompt user to enter face down cards

According to initial input at start
of the game

Validate

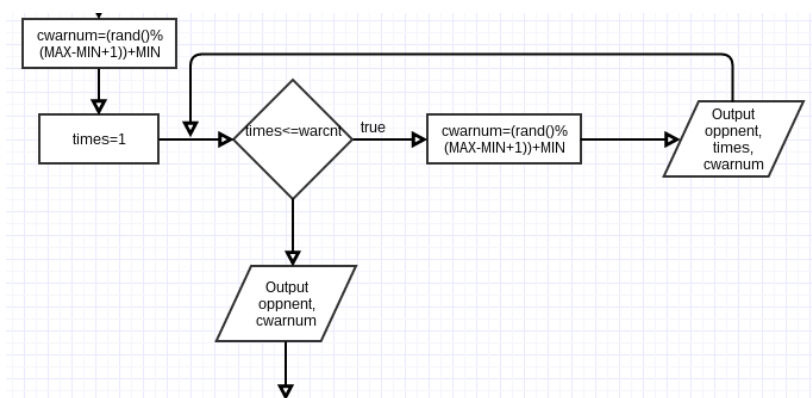
Prompt user to enter war card

Validate

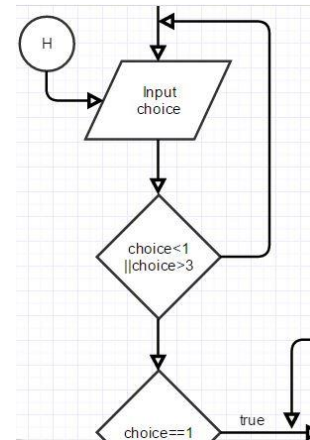


Computer now chooses faced down cards

Computer chooses war card



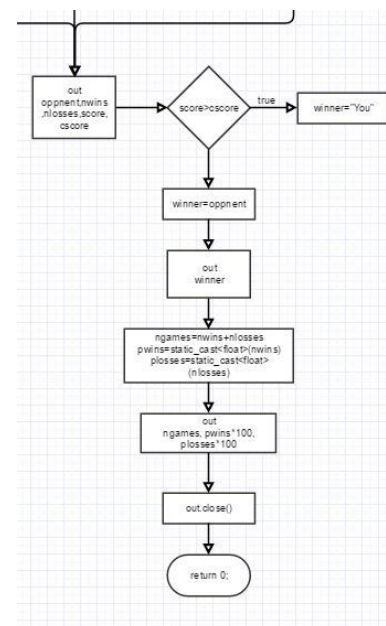
*Return to initial choice and repeat entire process
as long as “End Program”(3) is not selected.*



*If 3 is selected, thank user for playing and write
finishing stats to the output file.*

Calculate some statistics then close the file

Return 0; the program is complete



Constructs & Concepts Utilized

iostream Library

Name	Frequency	Description	Location
static_cast	3	Statically cast as different variable	Line 27,308,309
cout	58	Output Data	Throughout
cin	25	Input Data	Throughout
getline()	1	Reads string data	Line 50
cin.ignore()	4	Prevented input problems	Line 156,172,214,230
cin.clear()	4	Stopped infinite loop	Line 155,171,213,229

cstdlib Library

Name	Frequency	Description	Location
srand()	1	Random # seed	Line 27
rand()	5	Generates rand #	Line 125,179,181,237,239

ctime Library

Name	Frequency	Description	Location
time	1	Set current time	Line 27

io manip Library

Name	Frequency	Description	Location
fixed	1	Format final game stats	Line 311
setprecision()	1	Format final game stats	Line 311
showpoint	1	Format final game stats	Line 311
setw()	10	Format final game stats	Line 268-270,278-280,290,291,293,294

string Library

Name	Frequency	Description	Location
string	2	Declare var.	Line 29,42
getline()	already mentioned	already mentioned	already mentioned

cmath Library

Name	Frequency	Description	Location
abs()	2	Neg. Score Alert Point Difference	Line 270,274

fstream Library

Name	Frequency	Description	Location
out.open()	1	Open file	Line 46
out.close()	1	Close file	Line 316

out	12	Write to file	Line 288- 294,301,305,311- 313
ofstream	1	Declare var.	Line 43

Data Types:

Data Types	Frequency	Location
int	11	Line 32,33,36,37,40
unsigned int	7	Line 27,30,38,39
char	1	Line 31
string	2	Line 29,42
float	2	Line 41
ofstream	1	Line 43

Conditional Statements:

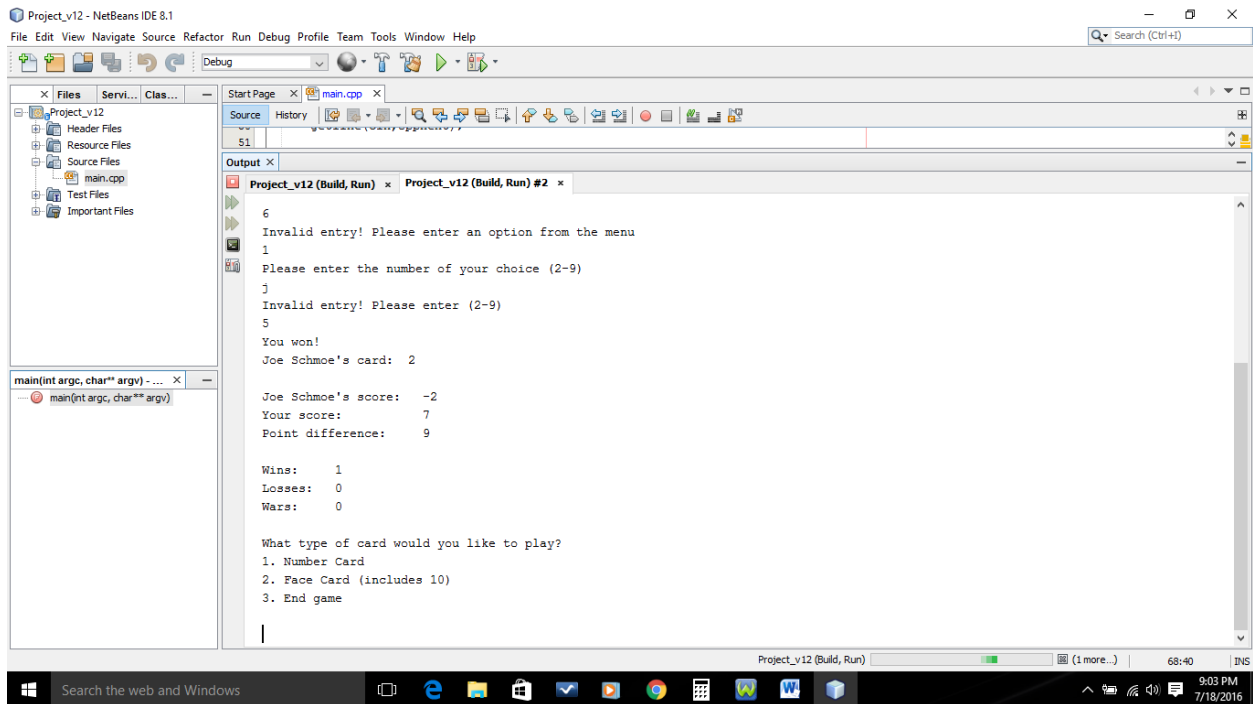
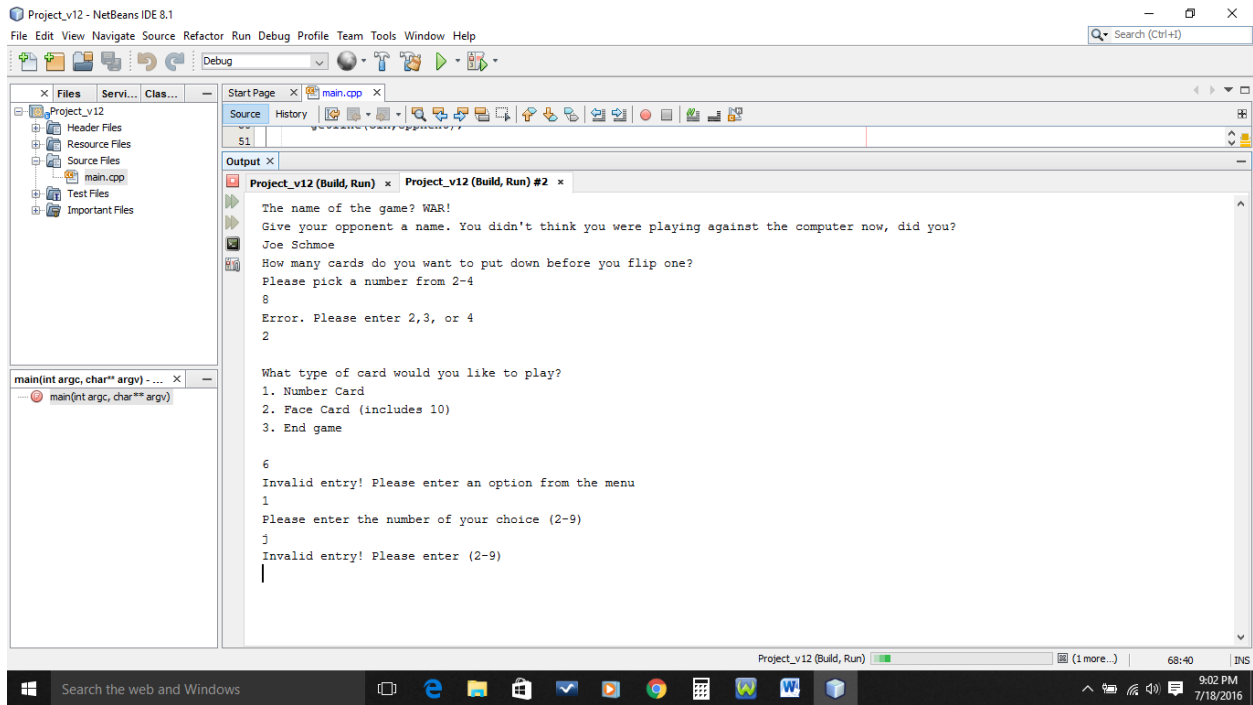
Conditional Statement	Frequency	Starting Location
if	2	Line 102,272
if/else	1	Line 297
if/else if	4	Line 77,126,189,248
switch	1	Line 103

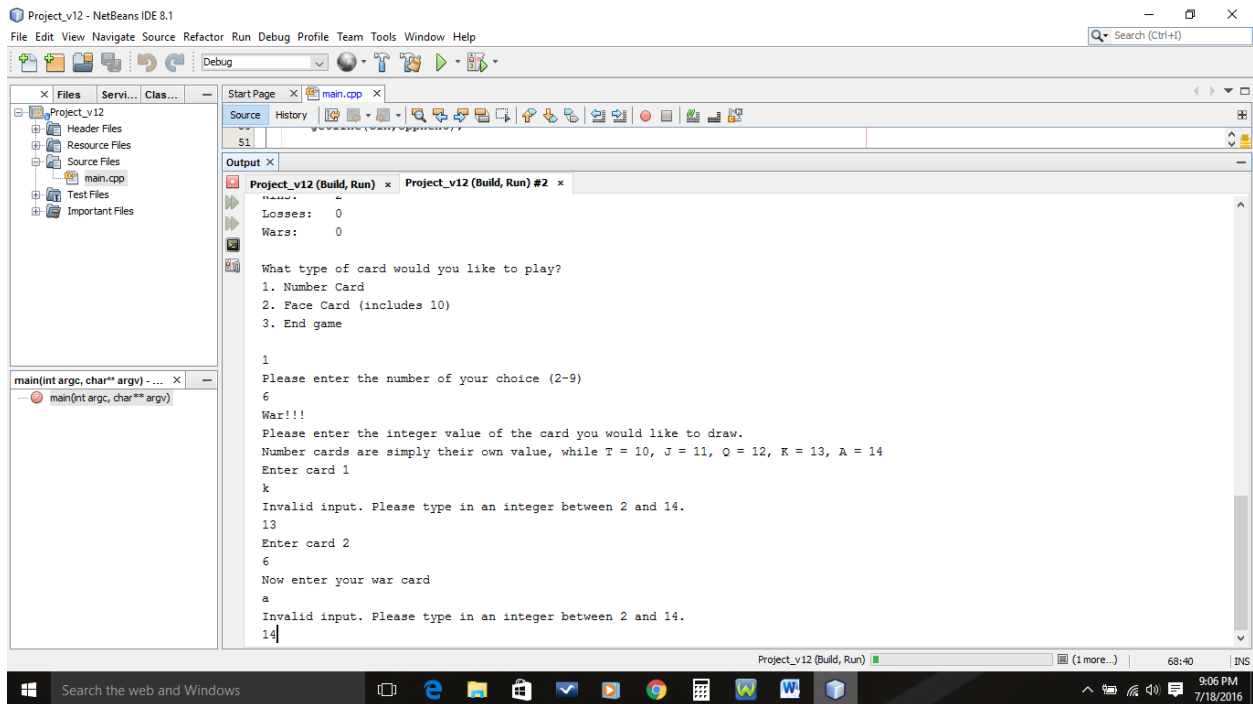
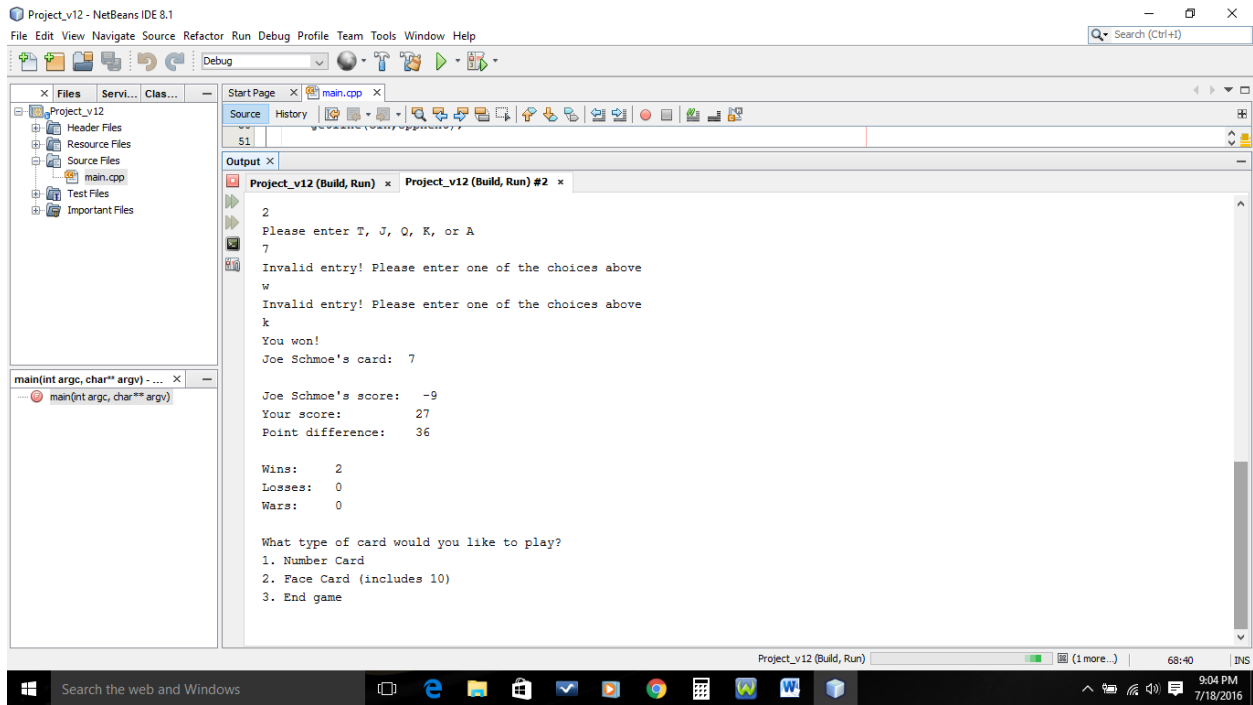
Loops:

Loops	Frequency	Starting Location
for	4	Line 149,180,207,238
while	9	Line 57,72,82,92,154,170, 202,212,228
do-while	1	Line 63

Proof of a Working Product

In the event, that my program does not work once it reaches Dr. Lehr, I have provided some screenshots that prove that the program did work at one time on the next few pages.





Project_v12 - NetBeans IDE 8.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Debug

Project_v12

- Header Files
- Resource Files
- Source Files
- main.cpp
- Test Files
- Important Files

main(int argc, char** argv) - ...

main(int argc, char** argv)

Source History

Output

Project_v12 (Build, Run) x Project_v12 (Build, Run) #2 x

```
14
Joe Schmoe's Card 1: 7
Joe Schmoe's Card 2: 9
Joe Schmoe's war card: 9
You won the battle!

Joe Schmoe's score: -40
Your score: 97
Point difference: 137

Wins: 3
Losses: 0
Wars: 1

What type of card would you like to play?
1. Number Card
2. Face Card (includes 10)
3. End game

3
Thank you for playing!

RUN SUCCESSFUL (total time: 7m 5s)
```

Project_v12 (Build, Run) 68:40 DNS

Search the web and Windows

9:08 PM 7/18/2016

csc5_2016/stats.dat at master · L-Elyse/csc5_2016 · GitHub

GitHub, Inc. [US] https://github.com/L-Elyse/csc5_2016/blob/master/Project/Project_v12/stats.dat

Apps HP Links Links for United States Microsoft Websites MSN Websites Windows Live UCR Wireless Network Downloads Textbook Selling!! Warranty HP Other bookmarks

L-Elyse / csc5_2016

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

Branch: master csc5_2016 / Project / Project_v12 / stats.dat Find file Copy path

L-Elyse Signed-off-by: L-Elyse <lguimont@student.rcc.edu> #2e3593 a minute ago

1 contributor

13 lines (10 sloc) | 251 Bytes Raw Blame History

```
1 Here are your Final Game Stats:
2
3 Opponent: Joe Schmoe
4 Wins: 3
5 Losses: 0
6
7 Your final score: 97
8 Joe Schmoe's final score: -40
9 Winner: You!
10 Total games played: 3
11 You won 100.0% of the hands you played
12 You lost 0.0% of the hands you played
```

© 2016 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub API Training Shop Blog About

Search the web and Windows

9:10 PM 7/18/2016

References

1. Dr. Lehr's Lectures & Lab
2. "Starting Out with C++: From Control Structures through Objects" Gaddis, Tony. 8th Edition. (Textbook)
3. www.cplusplus.com (only for the use of cin.clear();)

Program

```
/*
 * File:   main.cpp
 * Author: Laurie Guimont
 * Created on July 18, 2016, 1:30 AM
 * Purpose: War Card Game
 */

//System Libraries
#include <iostream> //Input/Output Stream Library
#include <iomanip>   //Formatting Library
#include <ctime>    //Unique Seed Value Library
#include <cstdlib>  //Random Value Library
#include <string>   //String Library
#include <fstream>  //File I/O
#include <cmath>    //Math Library
using namespace std;

//User Libraries

//Global Constants

//Function Prototypes

//Execution Begins Here!
int main(int argc, char** argv) {
    //Set the Random Number Seed
    srand(static_cast<unsigned int>(time(0)));
    //Declare variables, no doubles
    string opponent; //Who you will be playing
    unsigned int choice; //User menu option
    char cchoice; //User input representing card they want to play
    int number; //Random number chosen set to present time
    int value; //Value of each card
    const int MIN=2; //Minimum value to choose from
    const int MAX=14; //Maximum value to choose from
    int warcnt; //Number of faced down cards before flipping in war
    int warnum,cwarnum; //Card choice during war
    unsigned int times; //for Loop counter
    unsigned int nwins=0,nlosses=0,nwars=0,ngames;
    int score=0,wrscore=0,cscore=0,cwscore=0;
    float pwins,plosses;
    string winner;
    ofstream out;

    //Open File & Enter Primary Input Data
    out.open("stats.dat");
```

```

cout<<"The name of the game? WAR!"<<endl;
cout<<"Give your opponent a name. You didn't think you were ";
cout<<"playing against the computer now, did you?"<<endl;
getline(cin,oppnent);

cout<<"How many cards do you want to put down before you flip one?"<<endl;
cout<<"Please pick a number from 2-4"<<endl;
cin>>warcnt;

//Input Validation
while(warcnt<2||warcnt>4){
    cout<<"Error. Please enter 2,3, or 4"<<endl;
    cin>>warcnt;
}

//Process and Output the Data in the Loop
do{
    cout<<endl;
    cout<<"What type of card would you like to play?"<<endl;
    cout<<"1. Number Card"<<endl;
    cout<<"2. Face Card (includes 10)"<<endl;
    cout<<"3. End game"<<endl<<endl;
    cin>>choice;

    //Input Validation
    while(choice<1||choice>3){
        cout<<"Invalid entry! Please enter an option from the menu"<<endl;
        cin>>choice;
    }

    if(choice==1){
        cout<<"Please enter the number of your choice (2-9)"<<endl;
        cin>>cchoice;

        //Input Validation
        while(cchoice<'2'||cchoice>'9'){
            cout<<"Invalid entry! Please enter (2-9)"<<endl;
            cin>>cchoice;
        }
    }
    else if(choice==2){
        cout<<"Please enter T, J, Q, K, or A"<<endl;
        cin>>cchoice;

        //Input Validation
        while(cchoice!='a'&&cchoice!='A'&&cchoice!='k'&&cchoice!='K'&&
            cchoice!='q'&&cchoice!='Q'&&cchoice!='j'&&cchoice!='J'&&
            cchoice!='t'&&cchoice!='T'){
            cout<<"Invalid entry! Please enter one of the choices "
                "above"<<endl;
            cin>>cchoice;
        }
    }

    //Process the card choice
    if(choice!=3){
        switch(cchoice){
            case 'a':
            case 'A':value=14;break;
            case 'k':
            case 'K':value=13;break;
            case 'q':
            case 'Q':value=12;break;

```



```

        case 'j':
        case 'J':value=11;break;
        case 't':
        case 'T':value=10;break;
        case '9':
        case '8':
        case '7':
        case '6':
        case '5':
        case '4':
        case '3':
        case '2':value=(cchoice-48);break;
    }

    //Determine win, loss, or war
    number = (rand() % (MAX - MIN + 1)) + MIN;
    if(value>number){
        nwins+=1;
        score=score+value+number;
        cscore-=number;
        cout<<"You won!"<<endl;
        cout<<oppnent<<"'s card:  "<<number<<endl;
    }
    else if(value<number){
        nlosses+=1;
        score-=value;
        cscore=cscore+value+number;
        cout<<"Sorry. You lost."<<endl;
        cout<<oppnent<<"'s card:  "<<number<<endl;
    }
    else{
        nwars+=1;
        cout<<"War!!!"<<endl;
        cout<<"Please enter the integer value of the card you ";
        cout<<"would like to draw."<<endl;
        cout<<"Number cards are simply their own value, while T = 10,"
            " J = 11, Q = 12, K = 13, A = 14"<<endl;

        //Player's "Faced Down" Cards
        for(times=1;times<=warcnt;times++){
            cout<<"Enter card "<<times<<endl;
            cin>>warnum;

            //Input Validation
            while(!(warnum)||warnum<MIN||warnum>MAX){
                cin.clear();
                cin.ignore();
                cout<<"Invalid input. Please type in an integer";
                cout<<" between 2 and 14."<<endl;
                cin>>warnum;
            }

            wrscore+=warnum;
        }

        //Player's War Card
        cout<<"Now enter your war card"<<endl;
        cin>>warnum;

        //Input Validation
        while(!(warnum)||warnum<MIN||warnum>MAX){
            cin.clear();
            cin.ignore();

```

```

        cout<<"Invalid input. Please type in an integer";
        cout<<" between 2 and 14."<<endl;
        cin>>warnum;
    }

    //Opponent's "Faced Down" Cards
    cwarnum = (rand() % (MAX - MIN + 1)) + MIN;
    for(times=1;times<=warcnt;times++){
        cwarnum = (rand() % (MAX - MIN + 1)) + MIN;
        cwscore+=cwarnum;
        cout<<oppnent<<"'s Card "<<times<<": "<<cwarnum<<endl;
    }

    //Opponent's War Card
    cout<<oppnent<<"'s war card: "<<cwarnum<<endl;

    if(warnum>cwarnum){
        nwins+=1;
        score=score+value+number+wrscor+cwscore+warnum+cwarnum;
        cscore=cscore-number-cwscore-cwarnum;
        cout<<"You won the battle!"<<endl;
    }
    else if (warnum<cwarnum){
        nlosses+=1;
        score=score-value-warnum-wrscor;
        cscore=cscore+value+number+cwscore+wrscor+cwarnum+warnum;
        cout<<"You lost this battle."<<endl;
    }
    else{
        while(warnum==cwarnum){ //Must War Again!
            nwars+=1;
            cout<<"War!!!"<<endl;

            //Player's "Faced Down" Cards
            for(times=1;times<=warcnt;times++){
                cout<<"Enter card "<<times<<endl;
                cin>>warnum;

                //Input Validation
                while(!(warnum)||warnum<MIN||warnum>MAX){
                    cin.clear();
                    cin.ignore();
                    cout<<"Invalid input. Please type in an";
                    cout<<" integer between 2 and 14."<<endl;
                    cin>>warnum;
                }

                wrscor+=warnum;
            }

            //Player's War Card
            cout<<"Now enter your war card"<<endl;
            cin>>warnum;

            //Input Validation
            while(!(warnum)||warnum<MIN||warnum>MAX){
                cin.clear(); //Used to stop infinite loop
                cin.ignore();
                cout<<"Invalid input. Please type in an integer";
                cout<<" between 2 and 14."<<endl;
                cin>>warnum;
            }
        }
    }
}

```

```

//Opponent's "Faced Down" Cards
cwarnum = (rand() % (MAX - MIN + 1)) + MIN;
for(times=1;times<=warcnt;times++){
    cwarnum = (rand() % (MAX - MIN + 1)) + MIN;
    cwscore+=cwarnum;
    cout<<oppnent<<"'s Card "<<times<<": ";
    cout<<cwarnum<<endl;
}

//Computer's War Card
cout<<oppnent<<"'s war card: "<<cwarnum<<endl;

if(warnum>cwarnum){
    nwins+=1;
    score=score+value+number+wrscscore+cwscore+warnum+
        cwarnum;
    cscore=cscore-number-cwscore-cwarnum;
    cout<<"You won the battle!"<<endl;
}
else if (warnum<cwarnum){
    nlosses+=1;
    score=score-value-warnum-wrscscore;
    cscore=cscore+value+number+cwscore+wrscscore+
        cwarnum+warnum;
    cout<<"You lost this battle."<<endl;
}
}
}

//Game Stats
cout<<endl;
cout<<oppnent<<"'s score: "<<setw(4)<<cscore<<endl;
cout<<"Your score:      "<<setw(4)<<score<<endl;
cout<<"Point difference: "<<setw(4)<<abs(score-cscore)<<endl;

if(score<0){
    cout<<"Oh no! You're in the negative!"<<endl;
    cout<<"You need to score "<<abs(score)<<" points to get out ";
    cout<<"the red zone"<<endl;
}
cout<<endl;
cout<<"Wins:      "<<setw(3)<<nwins<<endl;
cout<<"Losses:   "<<setw(3)<<nlosses<<endl;
cout<<"Wars:     "<<setw(3)<<nwars<<endl;
}
}
while(choice!=3);

cout<<"Thank you for playing!"<<endl;

//Finishing Stats - Output to a File
out<<"Here are your Final Game Stats:"<<endl<<endl;
out<<"Opponent: "<<oppnent<<endl;
out<<"Wins:      "<<setw(4)<<nwins<<endl;
out<<"Losses:   "<<setw(4)<<nlosses<<endl;
out<<endl;
out<<"Your final score: "<<setw(4)<<score<<endl;
out<<oppnent<<"'s final score: "<<setw(4)<<cscore<<endl;

//Determine Winner of Game
if(score>cscore)
    winner="You!";

```

```

else
    winner=oppnent;
out<<"Winner:  "<<winner<<endl;

//Calculate Number of Games
ngames=nwins+nlosses;
out<<"Total games played: "<<ngames<<endl;

//Calculate Percentage of Wins and Losses
pwins=static_cast<float>(nwins)/ngames;
plosses=static_cast<float>(nlosses)/ngames;

out<<fixed<<setprecision(1)<<showpoint;
out<<"You won  "<<pwins*100<<"% of the hands you played"<<endl;
out<<"You lost "<<plosses*100<<"% of the hands you played"<<endl;

//Close File & Exit Stage Right!
out.close();
return 0;
}

```