PROJECT 1V2
Abdul Hakim Sheikh
CSC-5 (43592)
BlackJack Game V2

Introduction

This is the initialization for the previous blackjack project. There are four players in this game. One is the dealer, there are two AI players, and one is the user. For the sake of brevity, I will refer to the user as the player. The player will be presented with two randomly drawn cards from a fully shuffled deck. The player will have the opportunity to see one of two cards that the dealer has. The player must sum the values of the cards in his hand and decide whether or not to draw another card. There is a betting system wherein the initial amount of money each player has is presented along with the amount of money betted each game. This information is read from a file. If the players card sum is less than 21 but greater than that of the dealer, the player wins the bet. All losing players will automatically pay the other players their dues. If the converse holds true, then the dealer wins. If both parties have an equal sum, it is a tie. It is in the players hand whether to bet or not. If the player stops betting, the game will be over and the player will be asked if they want to play again.

The player must guess whether or not the sum of his/her hand is greater than that off the dealer. There are a few rules to the game however. If the dealers hand equals less than 17, the dealer must draw another card by default. If the player (or dealer) has one or more aces in hand, than the value of the ace must decrement to a value of one, as opposed to 11. All face cards are valued at 10, and all number cards hold the same value as their number.

There are several functions drawn out in the program that incorporate all the rules into the game.

Version 2 additions:

1. Two AI players have been added (ai1, ai2);
2. Several new variables have been added in order to make this program function (bool ai1Turn,ai2Turn, ai1Hand, ai2Hand, cai1Hnd, cai2Hnd) along with a betting system which incorporates 2 new variables (bank and bet). In order to read from the settings file, the stringstream user library has been added in order to read each line from the file as a string and convert them into integer values to be used in any function within main.

Summary

468 Lines total
20 Lines of white space
Time taken: about 2 weeks
Number of major variables: 20
Number of Functions used: 18

Pseudocode

Initialize

Create a card deck
Shuffle the deck
Ensure drawn cards remain drawn
Draw a card
Assign values to cards
Incorporate rules of blackjack (i.e. sum the hands, bust if over 21, dealer automatically hits if hand sum is less than 17 and decrement values of aces if sum is over 21)
Read initial money and bet amount from file
Format the output
Set parameters to begin game
Display the user interface

Main function

Give player choice to hit or stay
Sum the hand of both the dealer and the player
If the players score is less than 21, but greater than the dealer, player wins.
If the players score is less than 21 but less than the dealer's score, dealer wins.
If the player and dealer's scores are even, declare a tie.

If the players score exceeds 21, player automatically loses.
If players score exceeds 21 but an ace is in the hand, then value of ace will decrement to 1 and give the player a chance to win
After player's turn, AI's 1 and 2 will follow their algorithm and play their turns.
Repeat process until player either busts, wins or ties
Display updated UI
Output results to both file and console window
Ask player if the want to play again

End

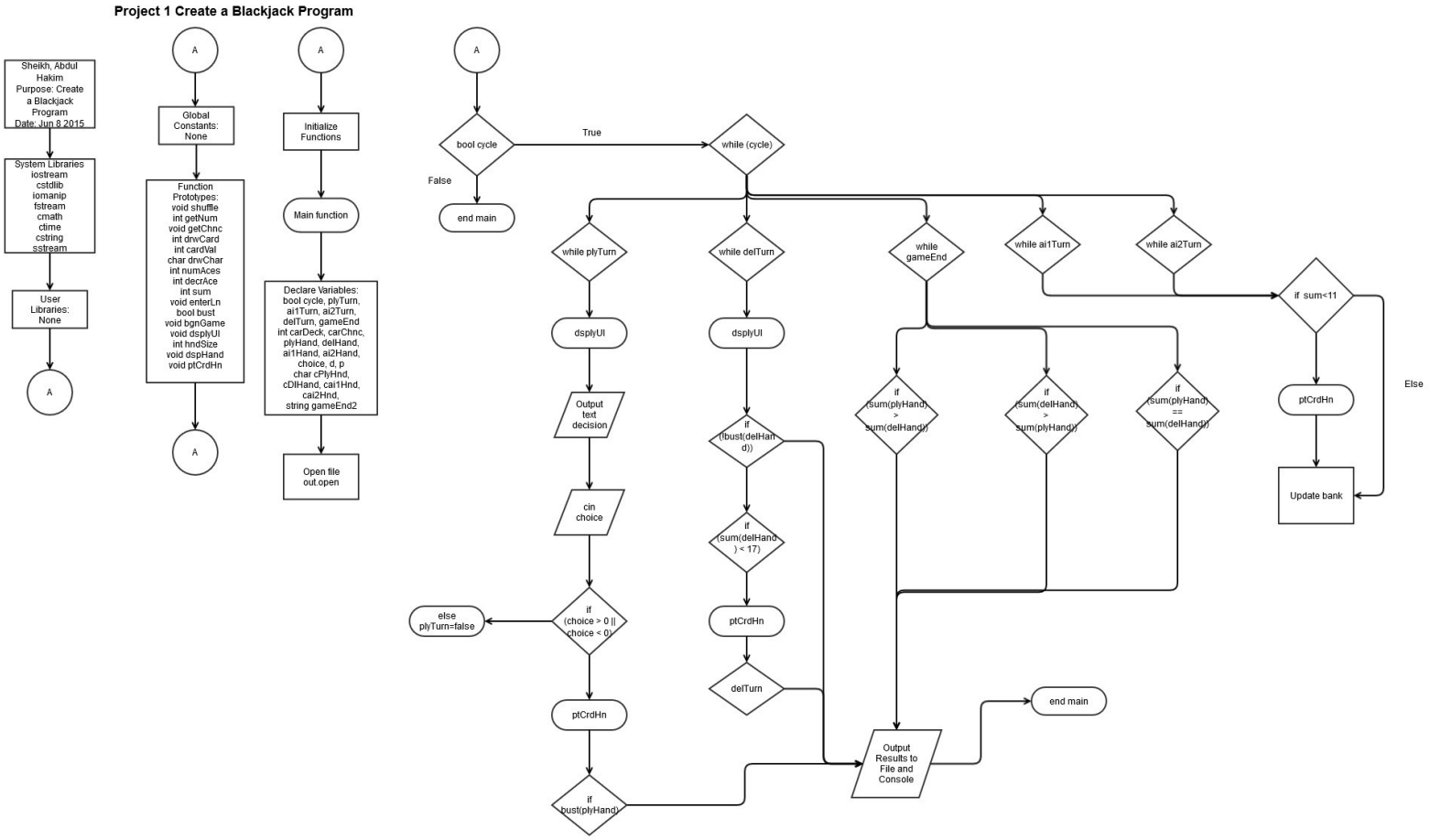
Major Variables

Type	Name	Description	Location
Bool	cycle	Cycles the game	Main Function
	plyTurn	Player's Turn	Main Function
	delTurn	Dealer's Turn	Main Function
	endGame	End Game	Main Function
	ai1Turn	AI1's Turn	Main Function
	ai2Turn	AI2's Turn	Main Function
Int	carDeck	Card Deck	Main Function
	carChnc	Card Chance	Main Function
	plyHand	Player's Hand Value	Main Function
	delHand	Dealers Hand Value	Main Function
	choice	Hit or stay choice	Main Function
	d and p	Sum of dealers and players hand respectively	Main Function
	ai1Hand	AI1's Hand Value	Main Function
	ai2Hand	AI2's Hand Value	Main Function
Char	cPlyHnd	Face value of Players Hand	Main Function
	cDlHand	Face Value of Dealers Hand	Main Function
	cai1Hnd [][]	Face value of AI1's hand	Main Function
	cai2Hnd [][]	Face value of AI2's Hand	Main Function
String	endGame	Text output for endgame	Main Function

C++ Constructs

1. Functions and Function Prototypes (Void, Ints and Chars)
2. If-Else Statements
3. For-Loops
4. While Loops
5. Ternary Operators
6. File Input and Output
7. Arrays (One and Two Dimensional) ([][])
8. Modulus Function and Random Number Seeding (srand() and %)
9. Boolean Expressions
10. Switch Statements
11. Arithmetic Operators (+,-,*,/)
12. Assignment and Increments Operators (+= and ++)

Flow Chart


Project 1 Create a Blackjack Program
Sheikh, Abdul Hakim
Purpose: Create a Blackjack Program
Date: Jun 8 2015
System Libraries
iostream
cstdlib
iomanip
fstream
cmath
ctime
cstring
sstream
User Libraries: None
A
Global Constants: None
Function Prototypes:
void shuffle
int getNum
void getChnc
int drwCard
int cardVal
char drwChar
int numAces
int decrAce
int sum
void enterLn
bool bust
void bgnGame
void dsplyUI
int hndSize
void dspHand
void ptCrdHn
Initialize Functions
Main function
Declare Variables:
bool cycle, plyTurn, ai1Turn, ai2Turn, delTurn, gameEnd
int carDeck, carChnc, plyHand, delHand, ai1Hand, ai2Hand, choice, d, p
char cPlyHnd, cDlHand, cai1Hnd, cai2Hnd,
string gameEnd2
Open file out.open
bool cycle
True
False
end main
while (cycle)
while plyTurn
while delTurn
while gameEnd
while ai1Turn
while ai2Turn
if sum<11
Else
dsplyUI
Output text decision
cin choice
if (choice > 0 || choice < 0)
else plyTurn=false
ptCrdHn
if bust(plyHand)
if (!bust(delHand))
if (sum(delHand) < 17)
delTurn
if (sum(plyHand) > sum(delHand))
if (sum(delHand) > sum(plyHand))
if (sum(plyHand) == sum(delHand))
Output Results to File and Console
Update bank

Program

```
/*
 * File:   main.cpp
 * Author: The Sheikh
 * Purpose: Make a Black Jack Program, Initalize it for Project 2
 * Created on May 6, 2015, 7:35 PM
 */

//System Libraries
#include <iostream>
#include <iomanip>
#include <cstdlib>
#include <fstream>
#include <cmath>
#include <ctime>
#include <string>
#include <sstream>

using namespace std;

//User Libraries

//Global Constants

//Function Prototypes

void shuffle ( int[] )                                                  ;//Function for Shuffling
deck of cards.
int  getNum  ( int[] )                                                  ;//Function for getting
number of cards in deck
void getChnc ( int[] , int [][2])                                       ;//Function for getting the
chance for drawing specific cards (Ace,2,3,4... --->King)
int  drwCard ( int[] , int [][2])                                       ;//Function to draw a card
and return card value
int  cardVal ( int )                                                    ;//Function enumerating the
numeric values of each card
char drwChar ( int )                                                    ;//Function to show
character value of each card, rather ineptly named but it works
int  numAces ( int [] )                                                 ;//
void decrAce ( int [] )                                                 ;//Function to decrement the
value of the aces should the sum of the hand equal more than 21
int  sum     ( int [] )                                                 ;//Function for adding the
value of hand
void enterLn ( int )                                                    ;//Function to add white
space
bool bust    ( int [] )                                                 ;//Function to determine
whether player has busted or not
void bgnGame ( int [], int [][2], int [], char [], int [], char [], int [], char [], int [], char [])       ;//Function
```

```
to begin game
void dsplyUI ( char [], char [], char [], char [], int [], int [], int [], int [],int, int, int, int, bool ) ;//Function
to display user interface
int  hndSize ( char [])                                                    ;//Function to display hand
size
void dispHnd ( bool , char [])                                             ;//Function to display the
hand
void ptCrdHn ( int [], int [][2], char [], int [])                         ;//Function to put
card in hand
void clrScrn ( )                                                           ;//Function to clear screen in
console window
void enterLn ( )                                                           ;//Function to dictate how
many lines to clear screen



//Declare Functions

//The shuffle function
void shuffle (int carDeck[]){
   for (int i=1; i<=13; i++){
      carDeck[i]=4;
   }
}

int getNum (int carDeck[]){
   int num = 0;
   for (int i=1; i<=13; i++){
      num += carDeck[i];
   }
   return num;
}
//This function determines the probability of drawing a specific card type again
void getChnc (int carDeck[], int carChnc[][2]){

   for (int i=0; i<getNum(carDeck);){          //This for-loop cycles through the total number of cards in
the deck
      for (int j=1; j<=13; j++){             //This for-loop cycles through the specific values associated
with playing card decks to create chances
         int x=carDeck[j];                  //This variable x is the number of that type [j] of card
         if ( x == 0 ){                    //If the number of the card [j] is 0, then
            carChnc[j][0]=0;               //Set chance of drawing the card as 0
            carChnc[j][1]=0;               //Set Chance of drawing the card as 0
         }
         else {                           //As long as x is not zero
            carChnc[j][0]= i+1;            //Beginning position for card type [j]
            carChnc[j][1]= i+x;            //Ending position for card type [j]
            i+=x;                         //i is incremented by the amount of card type [j]
         }
```

```
        }
    }
}
//Purpose of this function is to draw a card. This function will call the carDeck and carChnc arrays to draw a card
//In order to randomize the drawing of the card, I have seeded a random number with respect to time
int drwCard (int carDeck[], int carChnc[][2]){
    getChnc (carDeck, carChnc);                 //Calling chance function
    srand (time(0));                            //Seed random number based on time
    int x= getNum(carDeck);                     //Number of cards in deck
    int y= 1 + (rand() % x);                    //Random number modulus number of cards in deck +1 to yield number within range of cards
    for (int i=1; i<=13; i++){
        if (y>=carChnc[i][0] && y<=carChnc[i][1]){
            carDeck[i]--;
            return i;
        }
    }
}
//Purpose of this function is to assign an integer value to each card, so that they can be summed later. Case 1 represents an Ace,
//And the value assigned to it is 11 in accordance with Blackjack rules and all face cards J,Q,K are represented by 11,12,13 respectively, yield a value of 10
//The cards will be given character values for the purpose of output in the next function
int cardVal(int i){
    switch(i){
        case 1:
            return 11;
        case 2:
            return 2;
        case 3:
            return 3;
        case 4:
            return 4;
        case 5:
            return 5;
        case 6:
            return 6;
        case 7:
            return 7;
        case 8:
            return 8;
        case 9:
            return 9;
        case 10:
            return 10;
        case 11:
            return 10;
        case 12:
```

```
            return 10;
        case 13:
            return 10;
    }

}

//Enumerating the various cards in the deck
char drwChar(int i){
    switch(i){
        case 1:
            return 'A';
        case 2:
            return '2';
        case 3:
            return '3';
        case 4:
            return '4';
        case 5:
            return '5';
        case 6:
            return '6';
        case 7:
            return '7';
        case 8:
            return '8';
        case 9:
            return '9';
        case 10:
            return 'T';
        case 11:
            return 'J';
        case 12:
            return 'Q';
        case 13:
            return 'K';
    }
}
//This function
int numAces(int handVal[]){
    int numAces = 0;
    for(int i = 0; handVal[i] > 0; i++){
        if (handVal[i] == 11){
            numAces++;
        }
    }
    return numAces;
}
//This function, as per the rules of blackjack, decrements the value of the ace from 11 to 1 if the sum
```

```
exceeds 21 otherwise.
void decrAce(int handVal[]){
    for(int i= 0;i < 52;i++){
        if (handVal[i] == 11){
            handVal[i] = 1;
            break;
        }
    }
}
//This function sums the hand value of either the dealer or the player
int sum (int handVal[]){
    int sum = 0;
    for (int i=0; handVal[i]>0; i++){
        sum += handVal[i];
    }
    return sum;
}
//This function computes whether players hand results in a bust or not
bool bust (int handVal []){
    if (sum (handVal) > 21){
        if (numAces(handVal) > 0){
            decrAce(handVal);
            return false;
        }else{
            return true;
        }

    }
    else {
        return false;
    }
}
//This function is to begin the game. It calls in all the variables that need to be computed to play the
game
//The deck, the probability or chance function, the player hand, the dealer hand, and the character
values thereof
void bgnGame (int carDeck [], int carChnc[][2], int plyHand[], char cPlyHnd[], int delHand[], char
cDlHand[], int ai1Hand[], char cai1Hnd[], int ai2Hand[], char cai2Hnd[]){
        shuffle(carDeck);
        int x = drwCard (carDeck, carChnc);
        plyHand[0] =  cardVal(x);
        cPlyHnd[0] = drwChar(x);
        x = drwCard(carDeck, carChnc);
        plyHand[1] = cardVal(x);
        cPlyHnd[1] = drwChar(x);
        int y = drwCard (carDeck, carChnc);
        delHand [0] = cardVal(y);
        cDlHand [0] = drwChar (y);
        int z = drwCard (carDeck, carChnc);
```

```
        delHand[1] = cardVal(z);
        cDlHand[1] = drwChar(z);
        int a = drwCard (carDeck, carChnc);
        ai1Hand[0] = cardVal(a);
        cai1Hnd[0] = drwChar(a);
        int b = drwCard (carDeck, carChnc);
        ai1Hand[1] = cardVal(b);
        cai1Hnd[1] = drwChar(b);
        int c = drwCard (carDeck, carChnc);
        ai2Hand[0] = cardVal(c);
        cai2Hnd[0] = drwChar(c);
        int d = drwCard (carDeck, carChnc);
        ai2Hand[1] = cardVal(d);
        cai2Hnd[1] = drwChar(d);
}

void clrScrn(){
 enterLn(100);
}

void enterLn(int i){
 for(; i > 0; i--){
 cout << endl;
 }
}

void dsplyUI(char cPlyHnd[], char cDlHand[], char cai1Hnd[], char cai2Hnd[], int ai1Hand[], int
ai2Hand[], int plyHand[], int delHand[],int plyCash, int delCash, int ai1Cash, int ai2Cash, bool
endGame){
 clrScrn();
 cout << "($" << delCash << ")   Dealer Hand: ";
 dispHnd(!endGame, cDlHand);
 cout << endl << "($" << ai1Cash << ") Ai1 Hand:    ";
 dispHnd(!endGame, cai1Hnd);
 cout << endl << "($" << ai2Cash << ") Ai2 Hand:    ";
 dispHnd(!endGame, cai2Hnd);
 cout << endl << "($" << plyCash << ") Player Hand: ";
 dispHnd(false, cPlyHnd);
 cout << "  Hand Value: " << sum(plyHand) << endl;
}

int hndSize(char hand[]){
 int i = 0;
 while(hand[i] > 0){
 i++;
 }
 return i;
}
```

```
void dispHnd(bool plyTurn, char hand[]){

 int size = hndSize(hand);
 for (int i=0; i < size; i++){
 if (i == 0){
 if (plyTurn){
 cout << "?,";
 }else{
 cout << hand[i] << ",";
 }
 }else{
 if (i < (size - 1)){
 cout << hand[i] << ",";
 }else{
 cout << hand[i];
 }

 }
   }
}

void ptCrdHn(int carDeck[], int carChnc[][2], char cHand[], int hand[]){
 int x = drwCard(carDeck, carChnc);
 int y = hndSize(cHand);
 hand[y] = cardVal(x);
 cHand[y] = drwChar(x);
}

int main(int argc, char** argv) {
//Declare Variables
    ifstream in ("settings.txt");
    string line;
    ofstream out;

    if(!in){
        cout << "Could not find settings.txt" << endl;
        return -1;
    }
    (getline(in, line));
    int bank;
    stringstream(line) >> bank;
    int bet;
    (getline(in, line));
    stringstream(line) >> bet;
    in.close();
    cout<< bet << endl;
    cout<< bank <<endl;

    int plyCash = bank;
```

```
    int delCash = 0;
    int ai1Cash = bank;
    int ai2Cash = bank;


 bool cycle = true;
 while(cycle){
 //13 Cards in deck starting from Ace = 1 to King = 13
    int carDeck  [14];
    //
    int carChnc  [14][2];
    //Player Hand Array, keeps track of number of all the cards drawn for the player
    int  plyHand  [15] = {};                    //Initialize  values of array as '0'
    char cPlyHnd  [15] = {};                     //Character assignment for each spot in array (i.e. 11=Jack,
12=Queen, 13=King etc.)

    int  delHand  [15] = {};                    //Initialize  values of array as '0'
    char cDlHand  [15] = {};                     //Character assignment for each spot in array (i.e. 11=Jack,
12=Queen, 13=King etc.)

    int  ai1Hand  [15] = {};
    char cai1Hnd  [15] = {};

    int  ai2Hand  [15] = {};
    char cai2Hnd  [15] = {};

    int choice;                                 //The choice input variable for whether or not player wants to hit
or stay
    int d = sum (delHand);                      //Sum of dealers hand
    int p = sum (plyHand);                      //Sum of players hand
    //Output file
    out.open("blackjack.txt");
    cout<<"Writing to file"<<endl;
    bgnGame (carDeck, carChnc, plyHand, cPlyHnd, delHand, cDlHand, ai1Hand, cai1Hnd, ai2Hand,
cai2Hnd);

 bool ai1Turn = true;
 bool ai2Turn = true;
 bool plyTurn = true;
 bool delTurn = true;
 bool gameEnd = false;
  string endGame = "";
 while(plyTurn){
 dsplyUI(cPlyHnd, cDlHand, cai1Hnd, cai2Hnd, ai1Hand, ai2Hand, plyHand, delHand, plyCash,
delCash, ai1Cash, ai2Cash, false);
 cout<<"Would you like to hit or stay (1 for hit, 0 for stay) "<<endl;
 cin>>choice;
```

```
   if (choice > 0 || choice < 0){
      ptCrdHn(carDeck, carChnc, cPlyHnd, plyHand);
      if (bust(plyHand)){
            endGame = "You busted.";
            plyTurn = false;
      }else{
      plyTurn = true;
      }
   }else{
      plyTurn = false;
   }
}
 while(ai1Turn){
    if (sum(ai1Hand) <= 11){
       //ai1 hits
       ptCrdHn(carDeck, carChnc, cai1Hnd, ai1Hand);
    }else{
       ai1Turn = false;
    }
 }
 while(ai2Turn){
    if (sum(ai2Hand) <= 11){
       //ai2 hits
       ptCrdHn(carDeck, carChnc, cai2Hnd, ai2Hand);
    }else{
       ai2Turn = false;
    }
 }
 while(delTurn){
   dsplyUI(cPlyHnd, cDlHand, cai1Hnd, cai2Hnd, ai1Hand, ai2Hand, plyHand, delHand, plyCash,
delCash, ai1Cash, ai2Cash, false);
   if (!bust(delHand)){
      if (sum(delHand) < 17){
         ptCrdHn(carDeck, carChnc, cDlHand, delHand);
         delTurn = true;
      }else{
         delTurn = false;
      }
   }else{
      dsplyUI(cPlyHnd, cDlHand, cai1Hnd, cai2Hnd, ai1Hand, ai2Hand, plyHand, delHand, plyCash,
delCash, ai1Cash, ai2Cash, false);
      endGame = endGame + " Dealer busted.";
      delTurn = false;
   }
 }

 while(!gameEnd){
    string gameEnd2 = "";
```

```
 if (sum(plyHand) > sum(delHand) && !bust(plyHand) || (bust(delHand) && !bust(plyHand))){
    gameEnd2 = "You won!";
    out << "You won!";
 plyCash += bet;
 delCash -= bet;
 }
 if ((sum(delHand) > sum(plyHand) && !bust(delHand)) || bust(plyHand)){
 gameEnd2 =  "Dealer beat you!";
 out  << "Dealer beat you!" << endl;
 plyCash -= bet;
 delCash += bet;
 }
 if (sum(delHand) == sum(plyHand) && !bust(plyHand)){
 gameEnd2 =  "It's a tie!";
 out  << "It's a tie!" << endl;
 }

 //Ai1
 if (sum(ai1Hand) > sum(delHand) && !bust(ai1Hand) || (bust(delHand) && !bust(ai1Hand))){

 ai1Cash += bet;
 delCash -= bet;
 }
 if ((sum(delHand) > sum(ai1Hand) && !bust(delHand)) || bust(ai1Hand)){
    ai1Cash -= bet;
    delCash += bet;
 }
 if (sum(delHand) == sum(plyHand) && !bust(ai1Hand)){
    //AI1 tied with dealer.
 }

 //Ai2
 if (sum(ai2Hand) > sum(delHand) && !bust(ai2Hand) || (bust(delHand) && !bust(ai2Hand))){

 ai2Cash += bet;
 delCash -= bet;
 }
 if ((sum(delHand) > sum(ai2Hand) && !bust(delHand)) || bust(ai2Hand)){
    ai2Cash -= bet;
    delCash += bet;
 }
 if (sum(delHand) == sum(plyHand) && !bust(ai2Hand)){
    //AI1 tied with dealer.
 }


    dsplyUI(cPlyHnd, cDlHand, cai1Hnd, cai2Hnd, ai1Hand, ai2Hand, plyHand, delHand, plyCash,
delCash, ai1Cash, ai2Cash, true);
    cout << endGame << endl;
```

```
    out << endGame << endl;
    cout << gameEnd2 << endl;
 gameEnd = true;
 }
 cout << "Do you want to play again? Y/N ";
 char input;
 cin >> input;
 if (input == 'Y' || input == 'y'){
 cycle = true;
 }else{
 cycle = false;
 }
 };
 //Close File
       out.close();
   return 0;
}
```