Project 1

CIS-5

Blake Gilbert

INSTRUCTIONS

The game is Blackjack. The rules are simple. You draw cards to get as close to 21 as you can without going over. If you go over, the dealer automatically wins. If you are worried that drawing another card will cause you to go over 21, you can “stay”. You will be prompted to draw another card. Press ‘y’ to draw; press ‘n’ to stay.

After your turn is over, it is now the dealer’s turn. The dealer will draw cards until they get 17 or higher. Between each draw, the player must tell the computer they are ready to continue by pressing any key, then pressing enter. If the dealer goes over 21, the player automatically wins. If the dealer gets 17 or higher, the dealer must stay. At this point, the dealer compares their hand to the player’s hand. Whoever has the highest hand wins.

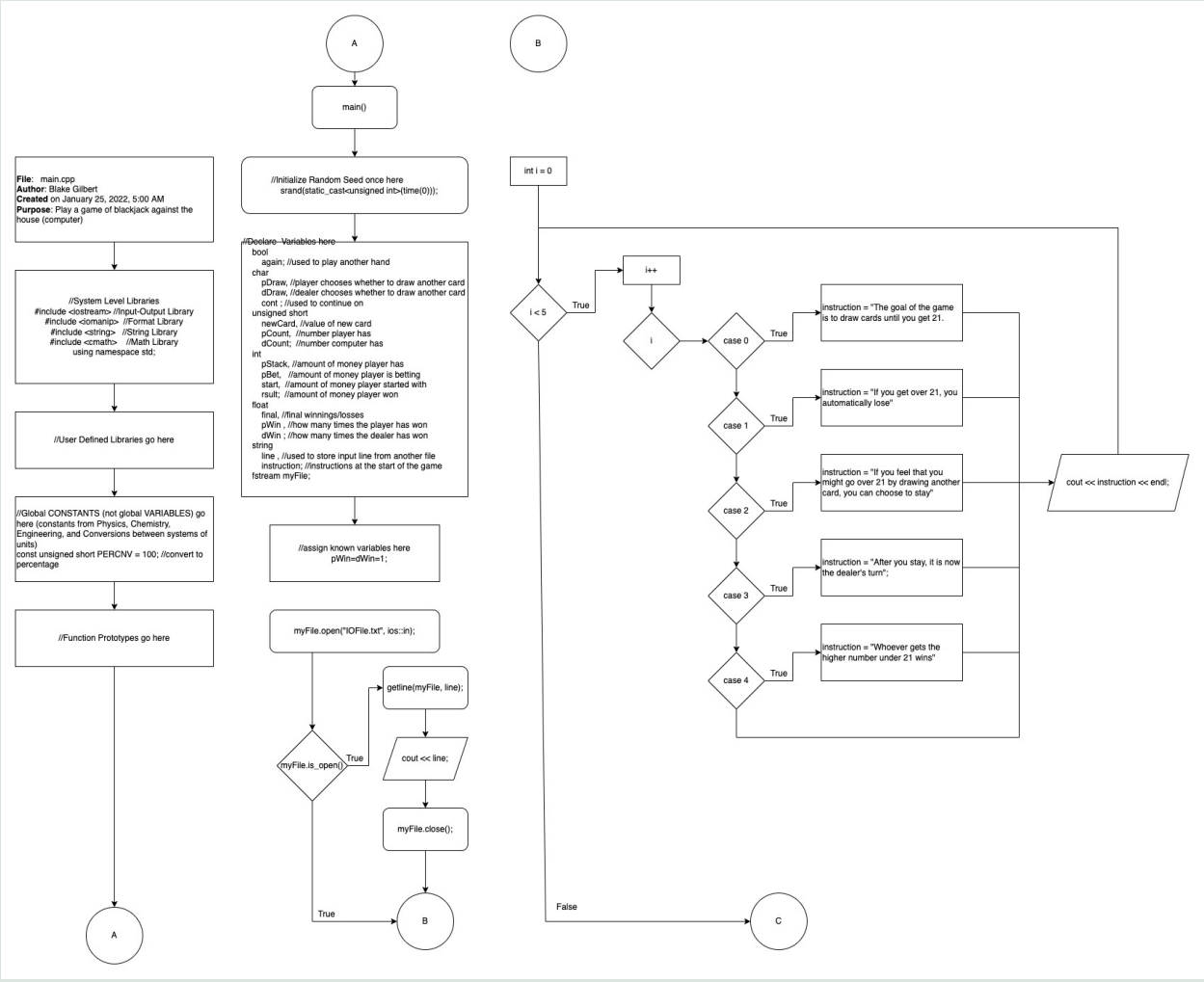
Before any hands are played, players enter how much money they are starting out with. At the start of each turn, the player enters how much they would like to bet. The money is deducted from the player’s stack as soon as they bet. It is only returned if the player wins the hand. The player will also receive the matching amount from the dealer.

You may play as many hands as you like, even if you end up running out of money. You will just owe money to the casino, which is not a good place to be in.

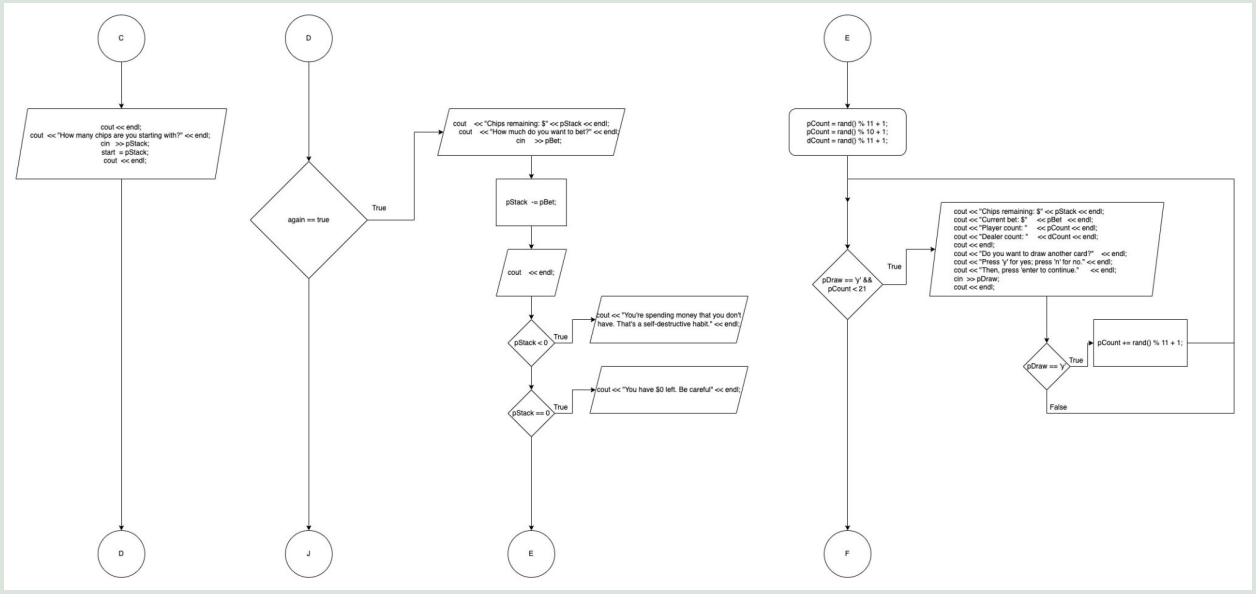
GAMEPLAY

```
The name of the game is Blackjack!
The goal of the game is to draw cards until you get 21.
If you get over 21, you automatically lose
If you feel that you might go over 21 by drawing another card, you can choose to stay
After you stay, it is now the dealer's turn
Whoever gets the higher number under 21 wins

How many chips are you starting with?
200

Chips remaining: $200
How much do you want to bet?
50

Chips remaining: $150
Current bet: $50
Player count: 4
Dealer count: 7

Do you want to draw another card?
Press 'y' for yes; press 'n' for no.
Then, press 'enter to continue.
n

Dealer flips other card
Dealer now has 17

Dealer has 17 and must stay.

Dealer wins
Player loses $50
Chips remaining: $150

Play another hand?
Your chance of winning is 50%
Press 'y' for yes; press 'n' for no.
Then, press 'enter to continue.
```

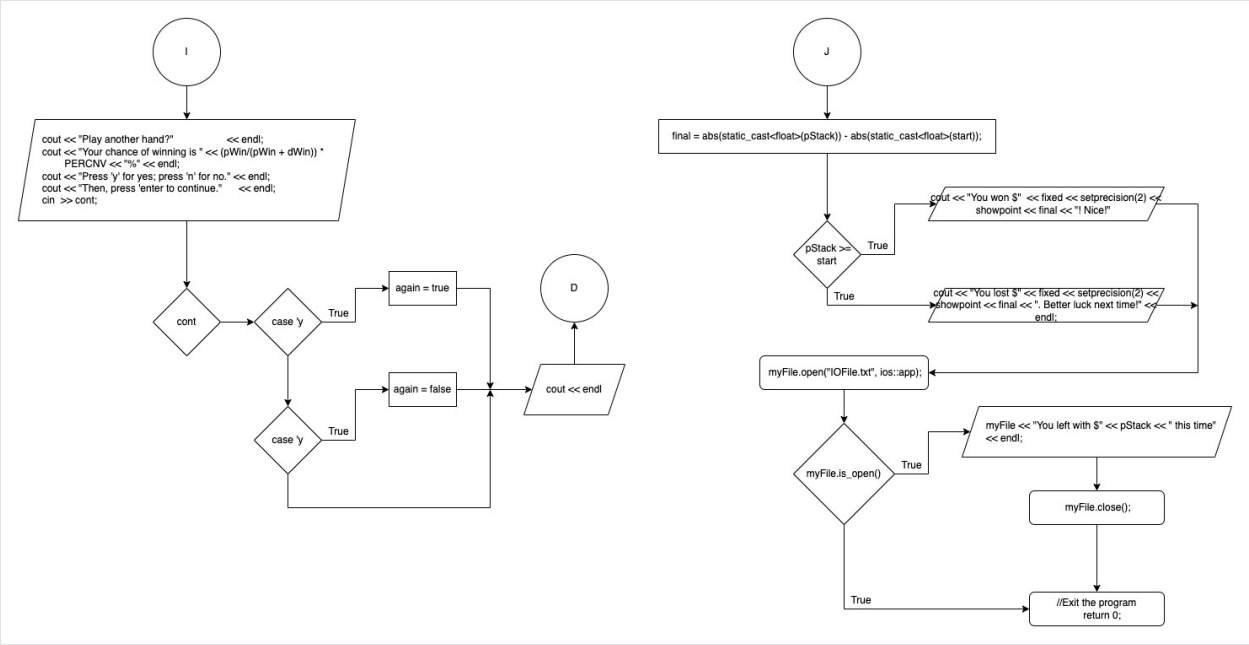
FLOWCHARTS


File: main.cpp
Author: Blake Gilbert
Created on January 25, 2022, 5:00 AM
Purpose: Play a game of blackjack against the house (computer)
//System Level Libraries
#include <iostream> //Input-Output Library
#include <iomanip> //Format Library
#include <string> //String Library
#include <cmath> //Math Library
using namespace std;
//User Defined Libraries go here
//Global CONSTANTS (not global VARIABLES) go here (constants from Physics, Chemistry, Engineering, and Conversions between systems of units)
const unsigned short PERCNV = 100; //convert to percentage
//Function Prototypes go here
A
A
main()
//Initialize Random Seed once here
srand(static_cast<unsigned int>(time(0)));
//Declare Variables here
bool
again; //used to play another hand
char
pDraw, //player chooses whether to draw another card
dDraw, //dealer chooses whether to draw another card
cont ; //used to continue on
unsigned short
newCard, //value of new card
pCount, //number player has
dCount; //number computer has
int
pStack, //amount of money player has
pBet, //amount of money player is betting
start, //amount of money player started with
rsult; //amount of money player won
float
final, //final winnings/losses
pWin , //how many times the player has won
dWin ; //how many times the dealer has won
string
line , //used to store input line from another file
instruction; //instructions at the start of the game
fstream myFile;
//assign known variables here
pWin=dWin=1;
myFile.open("IOFile.txt", ios::in);
myFile.is_open()
True
getline(myFile, line);
cout << line;
myFile.close();
True
B
B
int i = 0
i < 5
True
i++
i
case 0
True
instruction = "The goal of the game is to draw cards until you get 21.
case 1
True
instruction = "If you get over 21, you automatically lose"
case 2
True
instruction = "If you feel that you might go over 21 by drawing another card, you can choose to stay"
case 3
True
instruction = "After you stay, it is now the dealer's turn";
case 4
True
instruction = "Whoever gets the higher number under 21 wins"
cout << instruction << endl;
False
C

FLOWCHARTS (cont.)



FLOWCHARTS (cont.)



PSEUDOCODE

INCLUDE System Level Libraries
- Input-Output Library
- Format Library
- String Library
- Math Library
- File IO Library

USING Namespace
- standard namespace

DEFINE global constants
- percentage conversion const

FUNCTION main
- INITIALIZE Random Seed once

- DECLARE Variables
  - variable that is used to play another hand
  - variable that is player choosing whether to draw another card
  - variable that is dealer choosing whether to draw another card
  - variable that is used to continue on
  - variable that is value of new card
  - variable that is number player has
  - variable that is number computer has
  - variable that is amount of money player has
  - variable that is amount of money player is betting
  - variable that is amount of money player started with
  - variable that is amount of money player won
  - variable that is final winnings/losses
  - variable that is how many times the player has won
  - variable that is how many times the dealer has won
  - variable that is used to store input line from another file
  - variable that is instructions at the start of the game
  - variable that is this file

- ASSIGN player and dealer wins (must start at one for percentage calculation)

```
    OPEN file to be inputted from
    IF file is open
        GET first line
        OUTPUT first line
        CLOSE FILE
    END IF

    LOOP until the loop has ran 5 times
        REASSIGN a new string to instruction each iteration
        OUTPUT the instuction variable each iteration
    END LOOP
    OUTPUT line break to create an empty line

    OUTPUT a prompt asking the player how many chips they are starting with
    INPUT a number for starting chips
    ASSIGN the value of player's starting chips to a different variable so program knows
how many player started with, even after losing chips
    OUTPUT line break to create empty line

    LOOP until player wants to stop playing
        OUTPUT how many chups are remaining
        OUTPUT a prompt that asks players how much they want to bet
        INPUT a player bet
        DECREMENT player's bet from chip stack
        OUTPUT line break to create empty line
        IF player has a negative balance, OUTPUT warning
        IF player has no money left, OUTPUT warning

        INCREMENT player card count by a random card number between 1-11
        INCREMENT player card count by a random card number between 1-11
        INCREMENT dealer card count by a random card number between 1-11

        LOOP until player wants to stay
            OUTPUT chips remaining
            OUTPUT current bet
            OUTPUT player's card count
            OUTPUT dealer's card coutn
            OUTPUT line break to create empty line
            OUTPUT prompt asking if player wants to draw another card
            OUTPUT instructions to draw/stay (part1)
```

```
        OUTPUT instructions to draw/stay (part2)
        INPUT  player decision to draw or stay
        OUTPUT line break tp create an empty line
        IF player wants to draw another card, INCREMENT player card count by
random number between 1-11
    END LOOP

    IF player's card count is over 21
        OUTPUT message saying that player busted
    ELSE
        OUTPUT message saying that dealer flipped their other card
        INCREMENT dealer card count by random number between 1-11
        OUTPUT dealers current card count
        OUTPUT line break to create empty line

        IF dealers count is less than 17
            LOOP until count is greater than 17
                OUTPUT message saying that dealer drew another card
                OUTPUT line break to create an empty line
                INCREMENT dealer count by a random number between 1-11
                OUTPUT dealer card count
                OUTPUT line break to create an empty line
                OUTPUT message prompting player to continue
                OUTPUT message prompting player to continue
                INPUT if player is ready to continue
                OUTPUT line break to create an empty line
            END LOOP
            IF dealer count is greater than 21
                OUTPUT message saying that dealer busted
                OUTPUT line break to create empty line
            END IF
        ELSE
            OUTPUT that dealer must stay due to their current card count
        END ELSE
    END ELSE
```

```
        IF player count is greater than dealer count or dealer has more than 21
            OUTPUT message saying that player wins
            OUTPUT message saying how much player gained
            INCREMENT player stack by player bet and dealer's matching bet
            OUTPUT chips remaining
            OUTPUT line break to create an empty line
          ELSE IF dealer has a higher count than player and less than 21 or player has
more than 21
            OUTPUT message saying that dealer wins
            OUTPUT how much player lost
            OUTPUT chips remaining
            OUTPUT line break to create an empty line
          ELSE
            OUTPUT that the hand tied
            OUTPUT how much player lost
            OUTPUT chips remaining
            OUTPUT line break to create an empty line
        END ELSE

        OUTPUT message asking if player wants to play another hand
        OUTPUT message containing MATH EXPRESSION that calculates win
percentage
        OUTPUT instructions to play another hand
        OUTPUT instructions to play another hand
        INPUT  decision to play another hand

        SWITCH
          CASE set variable to continue if player says yes. BREAK
          CASE set variable to STOP if player says no. BREAK
        END SWITCH

        OUTPUT line break to create empty line
    END LOOP

    ASSIGN absolute difference between starting chips and ending chips
    IF player leaves with more money than they started
        OUTPUT win message with total winnings
    ELSE
        OUTPUT lose message with total losings
    END ELSE
```

```
    OPEN file to be appended
    IF file is open
        APPEND message with final result
        CLOSE FILE
    END IF

    EXIT program
END FUNCTION
```

CODE

```
/*
* File:   main.cpp
* Author: Blake Gilbert
* Created on January 25, 2022, 5:00 AM
* Purpose: Play a game of blackjack against the house (computer)
*
*/

//System Level Libraries
#include <iostream> //Input-Output Library
#include <iomanip>  //Format Library
#include <string>   //String Library
#include <cmath>    //Math Library
#include <fstream>  //File IO Library
using namespace std;

//User Defined Libraries go here

//Global CONSTANTS (not global VARIABLES) go here (constants from Physics, Chemistry,
Engineering, and Conversions between systems of units)
const unsigned short PERCNV = 100; //convert to percentage

//Function Prototypes go here

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

   //Declare  Variables here
   bool
       again; //used to play another hand
   char
       pDraw, //player chooses whether to draw another card
       dDraw, //dealer chooses whether to draw another card
       cont ; //used to continue on
   unsigned short
       newCard, //value of new card
```

```
        pCount,  //number player has
        dCount;  //number computer has
    int
        pStack, //amount of money player has
        pBet,   //amount of money player is betting
        start,  //amount of money player started with
        rsult;  //amount of money player won
    float
        final, //final winnings/losses
        pWin , //how many times the player has won
        dWin ; //how many times the dealer has won
    string
        line       , //used to store input line from another file
        instruction; //instructions at the start of the game
    fstream myFile;

    //assign known variables here
    pWin=dWin=1;
    myFile.open("IOFile.txt", ios::in);
    if (myFile.is_open()) {
        getline(myFile, line);
        cout << line;
        myFile.close();
    }

    //instructions
    for (int i = 0; i < 5; i++) {
        switch(i){
            case 0: instruction = "The goal of the game is to draw cards until you get
21."; break;
            case 1: instruction = "If you get over 21, you automatically lose"; break;
            case 2: instruction = "If you feel that you might go over 21 by drawing
another card, you can choose to stay"; break;
            case 3: instruction = "After you stay, it is now the dealer's turn"; break;
            case 4: instruction = "Whoever gets the higher number under 21 wins";
break;
        }
        cout << instruction << endl;
    }
    cout << endl;

    //initial money
```

```
   cout  << "How many chips are you starting with?" << endl; //prompt the player to
input starting money
   cin   >> pStack;
   start  = pStack;
   cout  << endl;

   //actual game
   do {
       //bet
       cout    << "Chips remaining: $" << pStack << endl;
       cout    << "How much do you want to bet?" << endl;
       cin     >> pBet;
       pStack  -= pBet;
       cout    << endl;
       if (pStack < 0)  cout << "You're spending money that you don't have. That's a
self-destructive habit." << endl;
       if (pStack == 0) cout << "You have $0 left. Be careful" << endl;

       //initial card values
       pCount = rand() % 11 + 1; //add random card to players hand
       pCount = rand() % 10 + 1; //add random card to players hand
       dCount = rand() % 11 + 1; //other card gets revealed later

       //player's turn
       do {
           //prompt player to draw new card
           cout << "Chips remaining: $" << pStack << endl;
           cout << "Current bet: $"     << pBet   << endl;
           cout << "Player count: "     << pCount << endl;
           cout << "Dealer count: "     << dCount << endl;
           cout << endl;
           cout << "Do you want to draw another card?"    << endl;
           cout << "Press 'y' for yes; press 'n' for no." << endl;
           cout << "Then, press 'enter to continue."      << endl;
           cin  >> pDraw;
           cout << endl;
           //add new card if player chooses yes
           if  (pDraw == 'y')   pCount += rand() % 11 + 1;
       } while (pDraw == 'y' && pCount < 21);

       //computer's turn
       if (pCount > 21) {
```

```
            cout << "Player busted." << endl; //game is over if player goes over 21
            cout << endl;
        }
        else {
            //dealer flips other card
            cout   << "Dealer flips other card"   << endl;
            dCount += rand() % 11 + 1; //reveal dealer's other card
            cout   << "Dealer now has " << dCount << endl;
            cout   << endl;
            //dealer continues to draw if they have less than 17; dealer must stay if
they have 17 or higher

            if (dCount < 17) {
                while (dCount < 17) {
                    cout    << "Dealer draws another card." << endl;
                    cout    << endl;
                    dCount  += rand() % 11 + 1; //dealer draws another card
                    cout    << "Dealer now has " << dCount  << endl;
                    cout    << endl;
                    cout    << "To continue, press any key" << endl;
                    cout    << "Then, press 'enter'"        << endl;
                    cin     >> cont;
                    cout    << endl;
                }
                if (dCount > 21) {
                    cout   << "Player wins. Dealer busted." << endl;
                    cout   << endl;
                }
            }
            else cout << "Dealer has " << dCount << " and must stay." << endl << endl;

            //compare player cards to dealer cards to see who wins
            if (pCount > dCount || dCount > 21) {
                cout   << "Player wins."   << endl;
                cout   << "Player gains $"    << pBet       << endl;
                pStack += (pBet * 2);
                cout   << "Chips remaining: $" << pStack    << endl;
                cout   << endl;
            }
            else if (dCount > pCount && dCount <= 21 || pCount > 21) {
                cout   << "Dealer wins"   << endl;
                cout   << "Player loses $"     << pBet      << endl;
```

```
                cout   << "Chips remaining: $" << pStack    << endl;
                cout   << endl;
            }
            else {
                cout    << "It's a tie hand."    << endl;
                cout    << "Player loses $" << 0 << endl;
                pStack += pBet;
                cout    << "Chips remaining: $" << pStack << endl;
                cout    << endl;
            }
        }

        //ask if player wants to continue
        cout << "Play another hand?"                    << endl;
        cout << "Your chance of winning is " << (pWin/(pWin + dWin)) * PERCNV << "%" <<
endl;
        cout << "Press 'y' for yes; press 'n' for no." << endl;
        cout << "Then, press 'enter to continue."      << endl;
        cin  >> cont;
        switch (cont) {
            case 'y': again = true ; break;
            case 'n': again = false; break;
            default : again = false; break;
        }
        cout << endl;
    } while (again);

    //display final winnings/losses
    final = abs(static_cast<float>(pStack)) - abs(static_cast<float>(start));
    pStack >= start
            ? cout << "You won $"  << fixed << setprecision(2) << showpoint << final <<
"! Nice!"                  << endl
            : cout << "You lost $" << fixed << setprecision(2) << showpoint << final <<
". Better luck next time!" << endl;

    //keep track of money that player left with each visit
    myFile.open("IOFile.txt", ios::app);
    if (myFile.is_open()) {
        myFile << "You left with $" << pStack << " this time" << endl;
        myFile.close();
    }
    //Exit the program
```

```
    return 0;
}
```