PROJECT 1

BLACK JACK

Marcus Burton

CSC/CIS-5

47471

11/5/17

Introduction

Title: BlackJack

This is your standard blackjack casino game

The player starts with laying down a bet. Then the player gets two randomly generated cards between 1 and 10.

Once the player is happy with their cards, they double down their bet. If the player busts(go over 21), then they are disqualified and the dealer wins by default, having to never even reveal their cards.

Assuming neither the player nor dealer busts, only three possibilities can occur, and only one of them result in player victory:

1. The player has a higher card number than the dealer = VICTORY
2. The dealer has a higher card number than the player = LOSS
3. The player and dealer have an identical number = DRAW

This is a game of odds, and is very exciting and fun.

Summary

Project size: 213 lines

Number of variables: 16

I primarily used do while loops and if statements to complete the project. Since every decision is based on asking the player for a hit, for loops were not necessary. They perhaps could have been used to set a variable for 52 individual cards, with only 4 of the same number being allowed. This project would be good for the second project for that very purpose of making an actual 52 card deck before the game even begins. I would also like to shuffle said deck in the next project as well.

Many of the ideas we learned in class were implemented, such as using a random number generator, Boolean data type, adding to existing values, taking in user inputs, comparing them and so on.

This project took me about two days to complete overall.

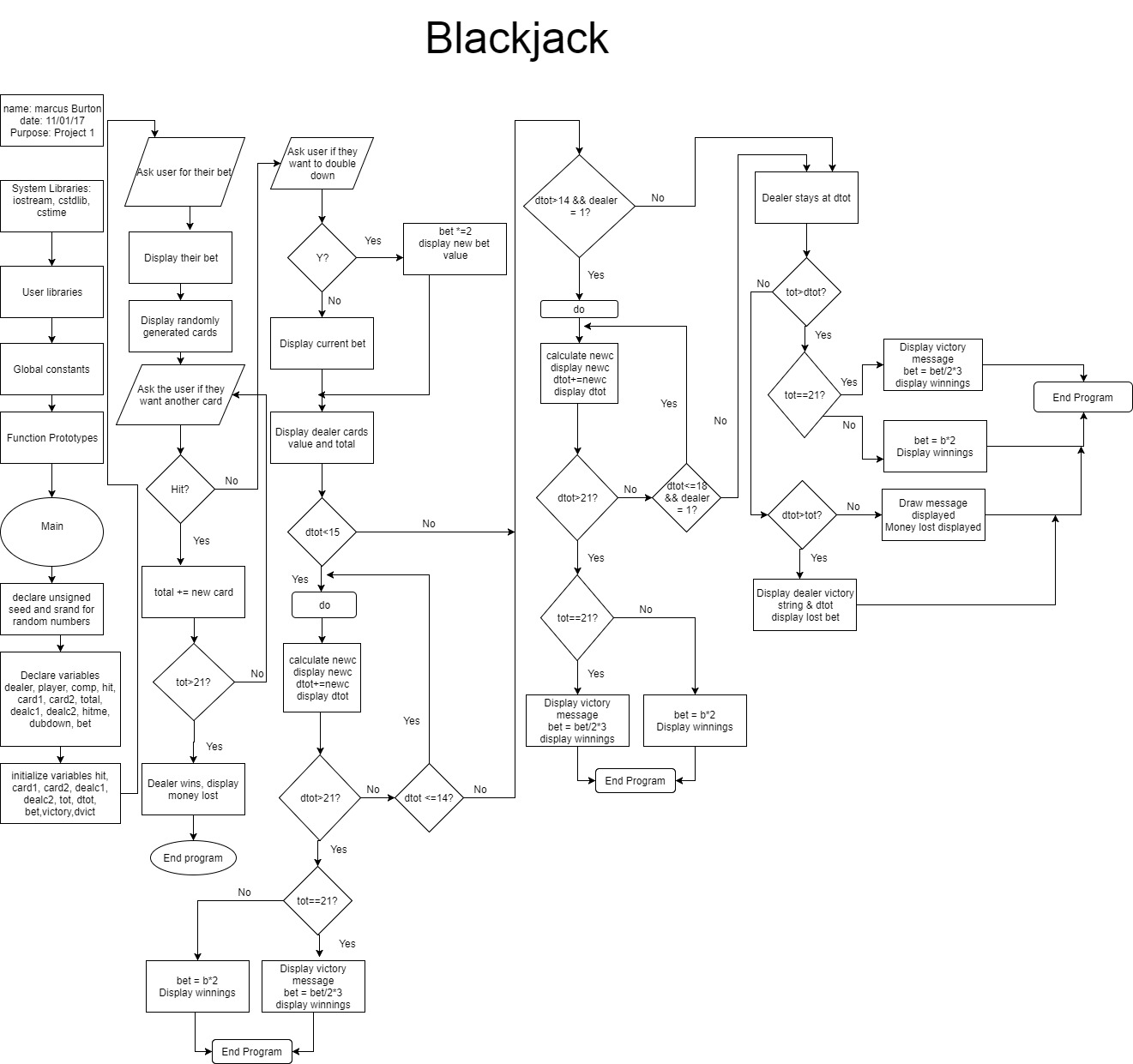
It was not a difficult problem to program, but I rewrote it several times as I found better ways to do it, and I believe there’s better ways still if I used different methods to calculate the cards and bets. I also believe I could add in a component that brings several other CPU players to truly emulate a blackjack game.

I also feel it’s worth noting that I used Boolean to emulate a 50/50 chance for the computer to hit on a number >=15 and <=18. I chose 18 as the high point because it’s the general rule to not hit higher than that. If Boolean=true than the PC has a chance to hit on a 14. It works well enough to emulate a random choice, though it’s only a 50/50 choice

Description:

The primary purpose of this game was to create a game versus a very primitive AI.

Flow Chart



Pseudo Code

*Initialize*

*Input player bet*

*Display two cards*

*Ask user for a hit*

*While the user hits*

*{*

*Add new card to previous value*

*Display value*

*If total >21*

*{*

*Dealer wins by default*

*}*

*Ask for another card*

*Loop up back to while statement*

*}*

*Ask user to double down*

*If user says yes*

*{*

*Double the bet and display*

*}*

*If user says no, display bet*

*Display dealers cards and total*

*If dealer has <15*

*{*

*Do the following*

*{*

*Add new card value(hit) to previous value*

*Display*

*If dealer is above 21*

*{*

*Victory for the player message displayed*

*If player is at 21*

*{*

*Display 21 victory message*

*Calculate 2:3 odds for player bet*

*Display the winnings*

*}*

*Else the player isn’t at 21*

*{*

*Display normal winnings*

*}*

*End program*

*}*

*}*

*Continue previous do loop as long as dealer has less than 15*

*}*

*If the dealer has more than 15, and Boolean is true*

*{*

*Do the following*

*{*

*Add new hit to previous card value*

*Display the total*

*If dealer is above 21*

*Display victory message for player*

*If player is at 21*

*{*

*Display 2:3 odds winnings*

*}*

*If player is not at 21*

*{*

*Display normal double winnings*

*}*

*End program*

*}*

*}*

*Do the above loop again so long as dealer has less than 18 and Boolean =true*

*If the player has more than the dealer*

*{*

*Player wins*

*If player is at 21*

*{*

*2:3 wininnings printed*

*}*

*Else*

*{*

*Bet is merely doubled*

*}*

*}*

*Else if dealer > player*

*{*

*Dealer wins*

*Player loses bet*

*}*

*Else*

*It’s a draw*

*No one wins*

*END PROGRAM*

Cross-List

**Cross Reference for Project 1**

**Where in Code**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter** | **Section** | **Topic** | **Line number** |
| 2 | 2 | cout | 44,46-48,50,51,58,61,66,67,75,82,88,92,97,98,105,107,  144,116,121-137,139,146,148,153,164,  171,173,178,186,179 |
|  | 3 | libraries | iostream, iomanip, cmath, cstdlib, fstream, string, ctime |
|  | 4 | variables/literals | 22,25-30,34-38,57,59,87,104,106,115,120,136,138,  147,152,172,177 |
|  | 5 | Identifiers | 22,25-30,39,40,57,104,136 |
|  | 6 | Integers | 21,26-28,30,39,40,57,104,136 |
|  | 7 | Characters | 29 |
|  | 8 | Strings | 41-42 |
|  | 9 | Floats No Doubles | n/a |
|  | 10 | Bools | 25,132,161 |
|  | 11 | Sizeof \*\*\*\*\* | n/a |
|  | 12 | Variables 7 characters or less | 22,25-30,39,40,57,104,136 |
|  | 13 | Scope \*\*\*\*\* No Global Variables | n/a |
|  | 14 | Arithmetic operators | 34-38,87,115,120,147,152,172,177 |
|  | 15 | Comments 20%+ | 22,25-32,39,40,42,4  4,48,50,51,53,44,  56,60,63,68,73,81,87,91,96,100,102  ,138,140,160,161,165,169 |
|  | 16 | Named Constants | 41,42 |
|  | 17 | Programming Style \*\*\*\*\* Emulate | n/a |
|  |  |  |  |
| 3 | 1 | cin | 45,52,76,83, |
|  | 2 | Math Expression | n/a |
|  | 3 | Mixing data types \*\*\*\* |  |
|  | 4 | Overflow/Underflow \*\*\*\* |  |
|  | 5 | Type Casting | n/a |
|  | 6 | Multiple assignment \*\*\*\*\* |  |
|  | 7 | Formatting output |  |
|  | 8 | Strings | 41,42 |
|  | 9 | Math Library | n/a |
|  | 10 | Hand tracing \*\*\*\*\*\* | n/a |
|  |  |  |  |
| 4 | 1 | Relational Operators | 54,63,85,100,108,112,129,132,140,144,161,166,169,  183, |
|  | 2 | if | 63,100,166 |
|  | 4 | If-else | 85 & 91, 112 & 119, 144 & 151, 169 & 176 |
|  | 5 | Nesting | 54-80, 86-90, 101-130, 133-162, 167-182, 184-187 |
|  | 6 | If-else-if | 167-183 |
|  | 7 | Flags \*\*\*\*\* | n/a |
|  | 8 | Logical operators | 132,161 |
|  | 11 | Validating user input | 54,58,129,161 |
|  | 13 | Conditional Operator | 54,63,85,100,108,112,129,132,140,144,161,169,183 |
|  | 14 | Switch | n/a |
|  |  |  |  |
| 5 | 1 | Increment/Decrement | n/a |
|  | 2 | While | 54 |
|  | 5 | Do-while | 102-129,134-161 |
|  | 6 | For loop | n/a |
|  | 11 | Files input/output both | 205,209,212 |
|  | 12 | No breaks in loops \*\*\*\*\*\* |  |
|  |  |  |  |
|  |  |  |  |
| \*\*\*\*\*\* Not | required to | show |  |

Page 1

void result(int tot, int dtot);

//Execution Begins Here!

int main(int argc, char\*\* argv) {

unsigned seed = time(0); // for the random number

srand(seed);

bool dealer = rand() & 1; // generate random choice for dealer to hit or not past 14

int player; // variable for the player

int comp; // variable for the dealer

int hit, card1, card2,total,dealc1,dealc2; //variables for cards nad hits and total

char hitme,dubdown; // variables to get a hit and to double down

int bet; // initial bet

//Initialize Variables

player=0; // start the values with zeroes

comp=0;

hit=(rand() % (11 - 1)) + 1; // generates a number between 1 and 10

card1=(rand() % (11 - 1)) + 1;

card2=(rand() % (11 - 1)) + 1;

dealc1=(rand() % (11 - 1)) + 1;

dealc2=(rand() % (11 - 1)) + 1;

int tot=card1+card2; // inital cards for player

int dtot=dealc1+dealc2; // initial card for dealer

const string victory = "Player wins! at ";

const string dvict = "Dealer wins! at"; // create strings for victory or defeat messages to make things easier

cout<<"How much would you like to bet?"<<endl; // ask user for bet

cin>>bet;

cout<<"You have bet a total of $"<<bet<<endl;

cout<<"Your cards are such"<<endl;

cout<<card1<<" and "<<card2<<", totaling "<<tot<<endl;// hand out the cards to the player and add them up

cout<<"Would you like another card?"<<endl; // ask player for another card

cout<<"Type Y for a hit, and N for no hit"<<endl; // any character but N would work

cin>>hitme;

// loop playing every time the player asks for a new card

while(hitme=='Y') //only give a hit when the player hits Y

{

// adding new hit

int newc=(rand()%(11-1))+1;

cout<<"You get a "<<newc<<endl;

tot+=newc;

// showing total

cout<<"total is "<<tot<<endl;

if(tot>21) // if over 21, player automatically loses

{

cout<<"Dealer wins by default!!"<<endl;

cout<<"I'm afraid you lose your clever bets of $"<<bet<<endl;

result(tot,dtot);

return 0; // end program

}

// add another hit question

cout<<"Another card?"<<endl;

cin>>hitme;

}

// offering the player a double down

cout<<"Would you like to double down? You are at "<<tot<<endl;

cin>>dubdown;

cout<<endl;

if(dubdown=='Y')

{

bet\*=2; // if yes, updating their bet amount

cout<<"Your bet is now "<<bet<<endl;

}

else // if not, bet stays the same

cout<<"Your bet remains at "<<endl;

cout<<endl;

// dealer begins to show cards

cout<<"Dealer gets a "<<dealc1<<" and a "<<dealc2<<endl;

cout<<"Dealer is at "<<dtot<<endl;

cout<<endl;

if(dtot<15) // dealer ALWAYS hits at less than 15(general casino rules)

{

do // perform the following only if the dealer is less than 15

{

int newc=(rand()%(11-1))+1; // new card variable initialized

cout<<"Dealer gets a "<<newc<<endl;

dtot+=newc; //adding new value

cout<<"Dealer's total is "<<dtot<<endl;

if(dtot>21) // dealer automatically loses at bust

{

cout<<victory<<tot<<endl; // standard victory message along with the total

if(tot==21) // only display if the player is at 21 for 3:2 odds

{

cout<<"Your intelligent betting today has won you a handsome prize"<<endl;

bet = bet/2\*3; // 21 wins a 2:3 ratio

cout<<"You have earned $"<<bet<<" at a 3:2 ratio due to getting 21!"<<

"congratulations!"<<endl;

result(tot,dtot);

}

else{

bet=bet\*2;

cout<<"Your intelligent betting has won you $"<<bet<<endl;

result(tot,dtot);

}

return 0; // end program as dealer would have lost

}

}

while(dtot<=14); // only continue do loop if dtot<=14

}

if(dtot>14 && dealer) // dealer may or may not hit above 15, leave it up to random true or false

{

do // perform the following only if the dealer is less than 15

{

int newc=(rand()%(11-1))+1;

cout<<"Dealer gets a "<<newc<<endl;

dtot+=newc; //adding new value

cout<<"Dealer's total is "<<dtot<<endl;

if(dtot>21) // dealer automatically loses at bust

{

cout<<victory<<tot<<endl;

if(tot==21)

{

cout<<"Your intelligent betting today has won you a handsome prize"<<endl;

bet = bet/2\*3;

cout<<"You have earned $"<<bet<<" at a 3:2 ratio due to getting 21!"<<

"congratulations!"<<endl;

result(tot,dtot);

}

else{

bet=bet\*2;

cout<<"Your intelligent betting has won you $"<<bet<<endl;

result(tot,dtot);

}

return 0;

}

}

//50/50 odds for dealer to hig

while(dtot<=18 && dealer); // don't want the dealer randomly hitting on a high value so I close 18 for the high point. boolean decides randomly if dealer will take the odds

}

cout<<"Dealer stays at "<<dtot<<endl;

// final score, player only wins if tot>dtot

if(tot>dtot)

{

cout<<victory<<tot<<endl;

if(tot==21) // only run following if the player is at

{

cout<<"Your intelligent betting today has won you a handsome prize"<<endl;

bet = bet/2\*3;

cout<<"You have earned $"<<bet<<" at a 3:2 ratio due to getting 21!"<<

"congratulations!"<<endl;

result(tot,dtot);

}else{

bet=bet\*2;

cout<<"Your intelligent betting has won you $"<<bet<<endl;

}

}

else if(dtot>tot)

{

cout<<dvict<<dtot<<endl;

cout<<"I'm afraid you lost your clever bets of $"<<bet<<endl;

result(tot,dtot);

}

else

cout<<"It's a draw! Dealer wins by default"<<

"\nAll bets are returned to the player, totaling $"<<bet<<endl;

result(tot,dtot);

return 0;

}

void result(int tot, int dtot)

{

ofstream out;

out.open("BlackjackWinOrLost.dat");

out<<"The player has a grand total of! "<<tot<<endl;

out<<"The dealer has a grand total of! "<<dtot<<endl;

if(tot>dtot)

{

out<<"The player is victorious! Huzzah!"<<endl;

}

else if(dtot>tot)

{

out<<"The dealer is victorious! Huzzah!"<<endl;

}

else

out<<"It's a tie! ouch!"<<endl;

}