Project 1

<Blackjack>

CSC-5 47982

Name: Stephen Anderson

Date: 10-25-2013

**Introduction**

Title: Blackjack

This is the classic card game, Blackjack. The game starts by a dealer dealing each player (including himself) 2 cards, once face up and the other face down. The player then adds up the total of his cards (eg. 5 + 7 = 12, 8 + facecard = 18) and decides to either “Hit” to get another card from the dealer and add it to his total, “Stay” to not receive another card and accept his current total, or “Double down” by doubling his bet, but having to hit only once more. Alternatively, if the player is given 2 of the same card to start, they may choose to “Split” in which each card becomes its own hand, and the player plays each hand individually against the dealer. The objective is to get the cards total value as close to 21 as possible without going over 21. When the player finally decides he feels close enough to 21, he chooses to stay and the dealer then reveals his face down card. The dealer must then hit until his total is greater than or equal to 17, and if he goes over 21, the player automatically wins. If the player got closer to 21 than the dealer, the player wins an amount of money equal to his initial bet (unless doubled down), but if the dealer got closer to 21, the player loses his bet.   
The game may be replayed as much as desired until the player runs out of money or decides to stop playing. When the player decides to stop playing, they can choose to save their score (the amount of money they had at the time of quitting) into a file for later viewing. When the program is re-launched, they can choose to view the scores they have saved.

**Summary**

Project size: 540 line  
Comments: 43 (not including heading), or 1 comment per 13 lines of code  
Number of variables: around 20, including variables declared in functions

I feel my program is vastly improved from its first version (see: project 1). I managed to make the code less than half the length while adding functionality, readability and ease of access for other programmers. Use of arrays to store the count of each card in the deck and the player and dealers hands managed to save a significant amount of space. Better datatype selection  
could result in less flawed inputs from users (making bet not an int, not using cin for char). This version is still lacking insurance and other players, but has the feature of allowing the player to choose how many decks to play with.   
  
It took about half a week to finish the program.  
The program wasn’t terribly difficult to code as I already had a simple version of the program and simply needed to update it with newer concepts.  
  
I used the internet to look up rules of Blackjack and how to perform various functions in C++ that we had not learned about or that I had forgotten.

**References**

Textbook

Pseudo-Code

Int main

Decide to view saved scores or not

Enter bet

Draw cards

Total cards

Dealer draws cards

Choose to hit, stay or double down

Continue to choose to either hit or stay until stay or bust

Total cards

Reveal dealer cards

Total dealer cards

If closer to 21 than dealer

Win

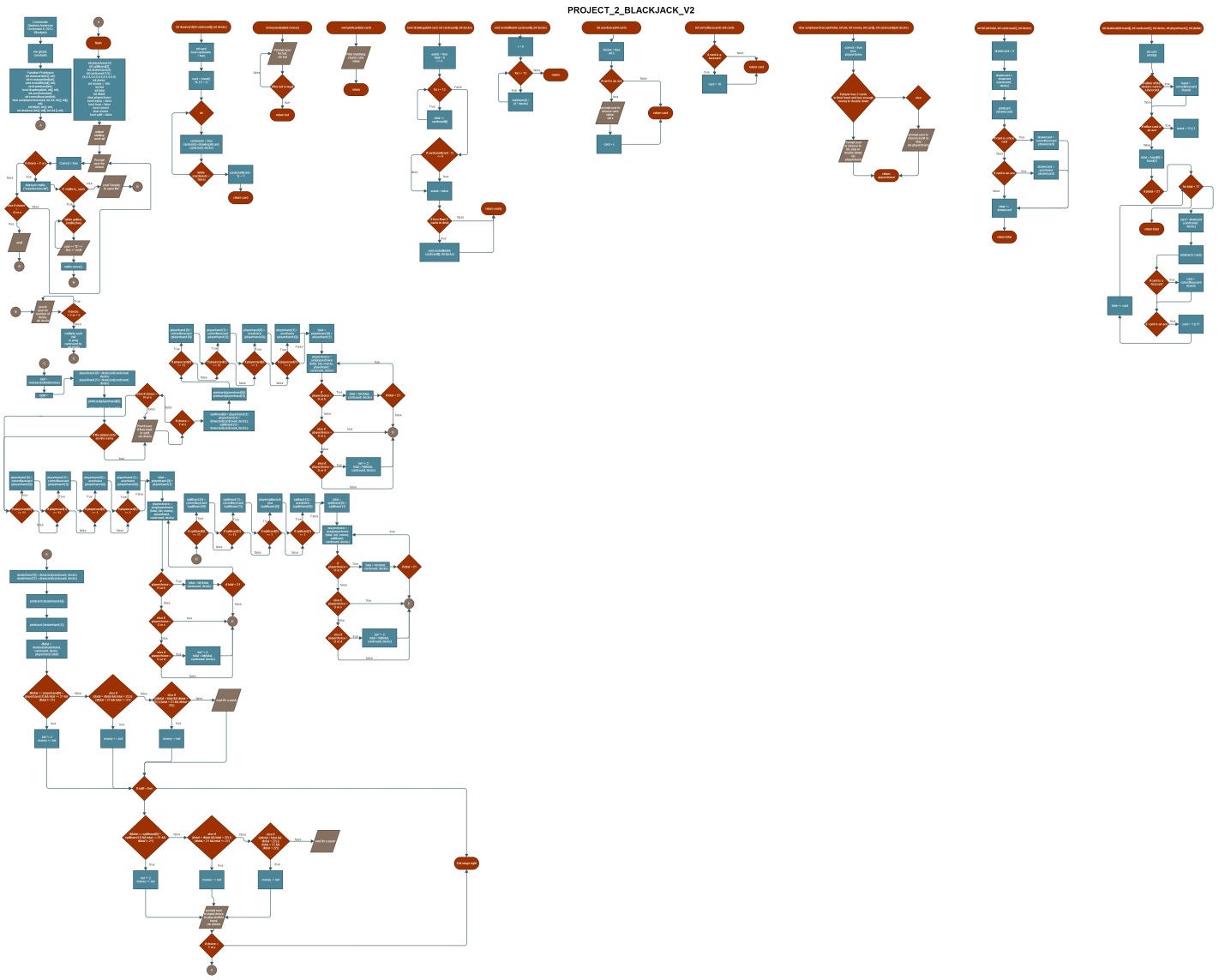
If farther away or bust

Lose

Choose whether to play another hand

If not, choose whether to save score into file

Walk away with tons of money

Flowchart (this was as big as I could get it…)

For more detail, please view project\_2\_blackjack\_v2.jpg

Code for int main

//Begin execution

int main(){

//Begin Random Number Generator

srand(time(0));

//Declare Variables

int playerhand [2]; //Array that holds the cards in the players starting hand

int splithand[2]; //Array that holds a players hand in the event of a split

int dealerhand [2]; //Array that holds the cards in the dealers starting hand

int cardcount [13] = {4,4,4,4,4,4,4,4,4,4,4,4,4}; //Array that holds all the cards in the deck and manages how many cards exist

int decks; //Controls how many decks are being played

int money = 100; //How much money the player has, starts at 100 and changes in run-time

int bet; //The bet the player makes through the moneyandbet function

int total; //The total of the player's hand

int dtotal; //The total of the dealer's hand

char playerchoice; //The players choice to hit, stay or double down as returned from the askplayerchoice function

bool replay = false; //Used to rerun program from beginning if player wants to play another hand

bool loop = false; //Used to determine if hitting can be done more than once after the askplayerchoice function

bool correct; //Used to make sure bool choice is a valid option and loops if it isn't

char choice; //Used by the user to answer yes or no

bool split = false; //Used to determine if a split is possible or not

string line; //For reading from saved scores

//Begin game, ask for bet

cout << " \_ \_ \_\_\_\_\_\_\_ \_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_ \_\_ \_\_\_\_\_\_\_ " << endl

<< " | | \_ | || || | | || || |\_| || |" << endl

<< " | || || || \_\_\_|| | | || \_ || || \_\_\_| " << endl

<< " | || |\_\_\_ | | | || | | || || |" << endl

<< " | || \_\_\_|| |\_\_\_ | \_|| |\_| || || \_\_\_| " << endl

<< " | \_ || |\_\_\_ | || |\_ | || ||\_|| || | " << endl

<< " |\_\_| |\_\_||\_\_\_\_\_\_\_||\_\_\_\_\_\_\_||\_\_\_\_\_\_||\_\_\_\_\_\_\_||\_| |\_||\_\_\_\_\_\_\_| " << endl << endl

<< " \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ " << endl

<< " | || | " << endl

<< " |\_ \_|| \_ | " << endl

<< " | | | | | | " << endl

<< " | | | |\_| | " << endl

<< " | | | | " << endl

<< " |\_\_\_| |\_\_\_\_\_\_\_| " << endl << endl

<< " \_\_\_\_\_\_\_ \_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_ \_ \_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_ \_ "

<< "| \_ || | | \_ || || | | | | || \_ || || | | | "

<< "| |\_| || | | |\_| || || |\_| | | || |\_| || || |\_| | "

<< "| || | | || || \_| | || || || \_| "

<< "| \_ | | |\_\_\_ | || \_|| |\_ \_\_\_| || || \_|| |\_ "

<< "| |\_| || || \_ || |\_ | \_ || || \_ || |\_ | \_ | "

<< "|\_\_\_\_\_\_\_||\_\_\_\_\_\_\_||\_\_| |\_\_||\_\_\_\_\_\_||\_\_\_| |\_||\_\_\_\_\_\_\_||\_\_| |\_\_||\_\_\_\_\_\_||\_\_\_| |\_| " << endl;

cout << endl << "Would you like to view your saved scores? (Y/N)" << endl;

do{

cin >> choice;

correct = true;

if(choice == 'Y' || choice == 'y'){

ifstream myfile ("savedscores.txt");

if (myfile.is\_open()){

while (getline (myfile,line)){

cout << "$" << line << endl;

}

cout << endl;

myfile.close();

}

else{

cout << "Unable to open file" << endl << endl;

}

}

else if(choice == 'N' || choice == 'n'){

cout << endl;

}

else{

cout << endl << "Please enter either Y or N to make a choice." << endl;

correct = false;

}

}while (correct == false);

cout << "We will start by giving you some money to gamble with!" << endl;

cout << endl << "How many decks would you like to play with? (1-5)" << endl;

do{

correct = true;

cin >> decks;

if(decks <1 || decks > 5){

cout << endl << "Please enter a number between 1 and 5." << endl;

correct = false;

}

else{

for(int i = 0; i < 13; ++i)cardcount[i] = 4 \* decks;

}

}while (correct == false);

do{

bet = moneyandbet(money);

split = false;

playerhand [0] = drawcard(cardcount, decks);

playerhand [1] = drawcard(cardcount, decks);

cout << endl << "You drew";

printcard(playerhand [0]);

cout << "and";

printcard(playerhand [1]);

cout << endl;

if(playerhand[0] == playerhand[1] && bet \* 2 <= money){

cout << endl << "Would you like to split? (Y/N)" << endl;

do{

correct = true;

split = false;

cin >> choice;

if(choice == 'Y' || choice == 'y') split = true;

else if(choice == 'N' || choice == 'n') split = false;

else{

cout << endl << "Please enter either Y or N to make a choice." << endl;

correct = false;

}

}while (correct == false);

if(split == true){

splithand[0] = playerhand [1];

playerhand [1] = drawcard(cardcount, decks);

splithand [1] = drawcard(cardcount, decks);

cout << endl << "Your first hand is";

printcard(playerhand[0]);

cout << "and";

printcard(playerhand[1]);

if(playerhand [0] >= 11) playerhand [0] = correctfacecard(playerhand [0]);

if(playerhand [1] >= 11) playerhand [1] = correctfacecard(playerhand [1]);

if(playerhand [0] == 1) playerhand [0] = acechoice(playerhand [0]);

if(playerhand [1] == 1) playerhand [1] = acechoice(playerhand [1]);

total = playerhand [0] + playerhand [1];

cout << endl << "Your total is " << total << "." << endl;

cout << endl << "We'll play your first hand first." << endl;

//Asks player to decide what to do from appropriate options and loops when needed.

if(total < 21){

do{

loop = false;

playerchoice = askplayerchoice(total, bet, money, playerhand, cardcount, decks);

if (playerchoice == 'H' || playerchoice == 'h'){

total = hit(total, cardcount, decks);

if(total < 21)loop = true;

}

else if(playerchoice == 'S' || playerchoice == 's'){

loop = false;

}

else if(playerchoice == 'D' || playerchoice == 'd'){

bet \*= 2;

total = hit(total, cardcount, decks);

loop = false;

}

if(total > 21)cout << endl << "You bust!" << endl;

}while (loop == true);

}

cout << endl << endl << "Your second hand is";

printcard(splithand[0]);

cout << "and";

printcard(splithand[1]);

if(splithand [0] >= 11) splithand [0] = correctfacecard(splithand [0]);

if(splithand [1] >= 11) splithand [1] = correctfacecard(splithand [1]);

if(splithand [0] == 1) splithand [0] = acechoice(splithand [0]);

if(splithand [1] == 1) splithand [1] = acechoice(splithand [1]);

total = splithand [0] + splithand [1];

cout << endl << "Your total is " << total << "." << endl;

cout << "Now we will play your second hand." << endl;

//Asks player to decide what to do from appropriate options and loops when needed.

if(total < 21){

do{

loop = false;

playerchoice = askplayerchoice(total, bet, money, splithand, cardcount, decks);

if (playerchoice == 'H' || playerchoice == 'h'){

total = hit(total, cardcount, decks);

if(total < 21)loop = true;

}

else if(playerchoice == 'S' || playerchoice == 's'){

loop = false;

}

else if(playerchoice == 'D' || playerchoice == 'd'){

bet \*= 2;

total = hit(total, cardcount, decks);

loop = false;

}

if(total > 21)cout << endl << "You bust!" << endl;

}while (loop == true);

}

}

}

if(split == false){

//Begin ace choice and face card fixing

if(playerhand [0] >= 11) playerhand [0] = correctfacecard(playerhand [0]);

if(playerhand [1] >= 11) playerhand [1] = correctfacecard(playerhand [1]);

if(playerhand [0] == 1) playerhand [0] = acechoice(playerhand [0]);

if(playerhand [1] == 1) playerhand [1] = acechoice(playerhand [1]);

total = playerhand [0] + playerhand [1];

cout << "Your total is " << total << "." << endl;

//Dealer starts here

dealerhand [0] = drawcard(cardcount, decks);

dealerhand [1] = drawcard(cardcount, decks);

cout << endl << "The dealer drew";

printcard(dealerhand [0]);

cout << "and another card." << endl;

//Asks player to decide what to do from appropriate options and loops when needed.

if(total < 21){

do{

loop = false;

playerchoice = askplayerchoice(total, bet, money, playerhand, cardcount, decks);

if (playerchoice == 'H' || playerchoice == 'h'){

total = hit(total, cardcount, decks);

if(total < 21)loop = true;

}

else if(playerchoice == 'S' || playerchoice == 's'){

loop = false;

}

else if(playerchoice == 'D' || playerchoice == 'd'){

bet \*= 2;

total = hit(total, cardcount, decks);

loop = false;

}

if(total > 21)cout << endl << "You bust!" << endl;

}while (loop == true);

}

}

//Dealer continues with AI

cout << endl << "The dealer reveals that he has";

printcard(dealerhand [0]);

cout << "and";

printcard(dealerhand [1]);

dtotal = dealerai(dealerhand, cardcount, decks, playerhand, total);

if(total == playerhand[0] + playerhand [1] && total == 21 && dtotal != 21){

bet \*= 2;

cout << endl << endl << "Congratulations! You got a Blackjack! You won twice your bet, so you recieve $" << bet << " for winning!" << endl;

money += bet;

}

else if((total > dtotal && total < 22) || (dtotal > 21 && total <= 21)){

if(dtotal > 21)cout << "The dealer busts!" << endl;

cout << endl << endl << "Congratulations! You won! You recieve $" << bet << "." << endl;

money += bet;

}

else if((dtotal > total && dtotal < 22) || (total > 21 && dtotal < 22)){

cout << endl << endl << "Sorry, you lost. You lose $" << bet << "." << endl;

money -= bet;

}

else if (total == dtotal){

cout << endl << endl << "It's a push. You are returned your bet of $" << bet << "." << endl;

bet = 0;

}

if(split == true){

if(total == splithand[0] + splithand [1] && total == 21 && dtotal != 21){

bet \*= 2;

cout << endl << endl << "Congratulations! You got a Blackjack! You won twice your bet, so you recieve $" << bet << " for winning!" << endl;

money += bet;

}

else if((total > dtotal && total < 22) || (dtotal > 21 && total <= 21)){

if(dtotal > 21)cout << "The dealer busts!" << endl;

cout << endl << endl << "Congratulations! You won! You recieve $" << bet << "." << endl;

money += bet;

}

else if((dtotal > total && dtotal < 22) || (total > 21 && dtotal < 22)){

cout << endl << endl << "Sorry, you lost. You lose $" << bet << "." << endl;

money -= bet;

}

else if (total == dtotal){

cout << endl << endl << "It's a push. You are returned your bet of $" << bet << "." << endl;

bet = 0;

}

}

if(money > 0){

cout << endl << endl << "Would you like to play another hand? (Y/N)" << endl;

do{

cin >> choice;

correct = true;

if(choice == 'Y' || choice == 'y') replay = true;

else if(choice == 'N' || choice == 'n'){

replay = false;

cout << endl << endl << "Would you like to save your score to a file? (Y/N)" << endl;

do{

cin >> choice;

correct = true;

if(choice == 'Y' || choice == 'y'){

ofstream myfile ("savedscores.txt");

if (myfile.is\_open()){

myfile << money << endl;

myfile.close();

}

else{

cout << "Unable to open file";

}

}

else if(choice == 'N' || choice == 'n'){

cout << endl << "Have a nice day!";

}

else{

cout << endl << "Please enter either Y or N to make a choice." << endl;

correct = false;

}

}while (correct == false);

}

else{

cout << endl << "Please enter either Y or N to make a choice." << endl;

correct = false;

}

}while (correct == false);

}

else{

cout << endl << endl << "Uh oh! You're all out of money!" << endl << "If you'd like more, simply make payments of $100 to Stephen Anderson." << endl;

replay = false;

}

}while (replay == true);

system("PAUSE");

return 0;

}