**Project 1**

**Blackjacks**

**Louidor, Richemond**

**Course: CSC 5**

**SEC: 46091**

**DATE: 08/1/15**

**Table of Contents**

**Topics:**

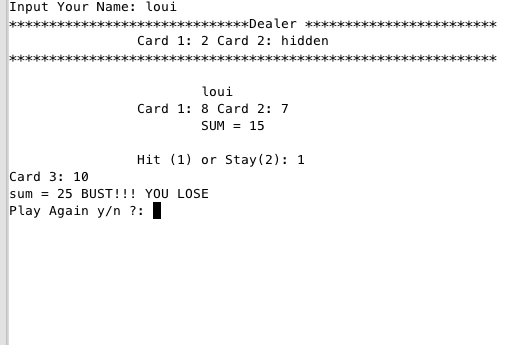
1. **Introduction**
2. **Game Play and Rules**
3. **Development and Logic**
4. **Reference**
5. **Summary**

**INTRODUCTION:**

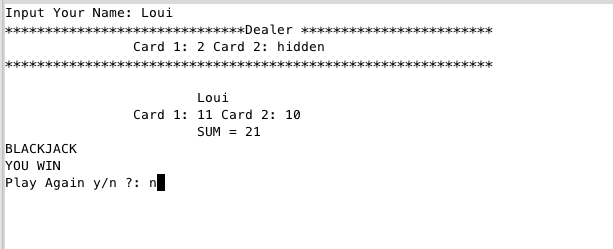
Blackjack, also known as twenty-one it is thought that it origins started in France back in the 1700s. Blackjack is widely known and popular game in the United States, mostly common in Casinos. Nowadays, blackjack can be played right from home or on the go either on a smartphone or on a personal computer. Finally I get my chance of creating my own version of blackjack.

**Game Rule:**

1. Winning
   1. Player hit Blackjack (21) on first deal.
   2. Player sum is greater than dealer sum after additional cards are dealt.
   3. Player hit Blackjack (21) after additional cards are dealt.
   4. Dealer bust (dealer sum is >21)
2. Losing
   1. Dealer hit Blackjack.
   2. Player Bust during additional card dealing
   3. Player sum less than computer card sums.
   4. Computer cheats.
3. Push
   1. Player sum equal dealer sum (pSum == dSum)

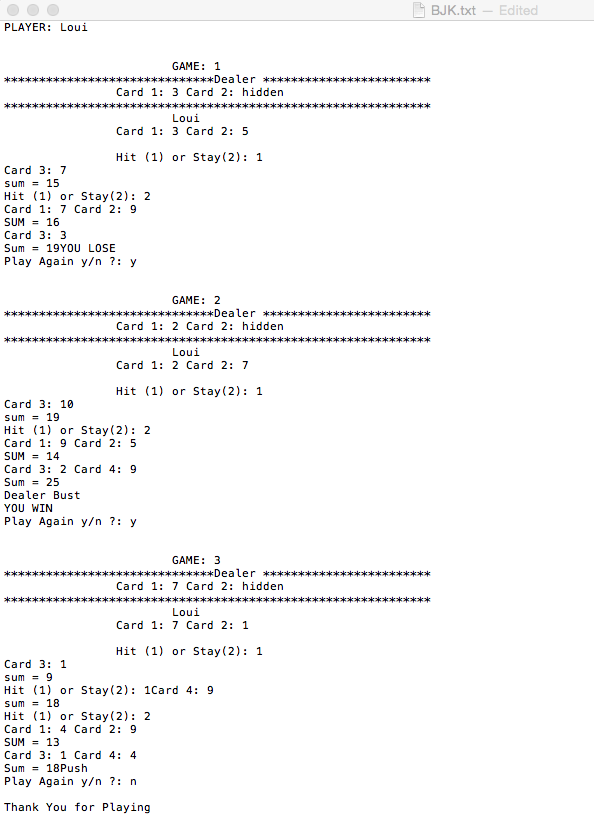


**Game Play:**

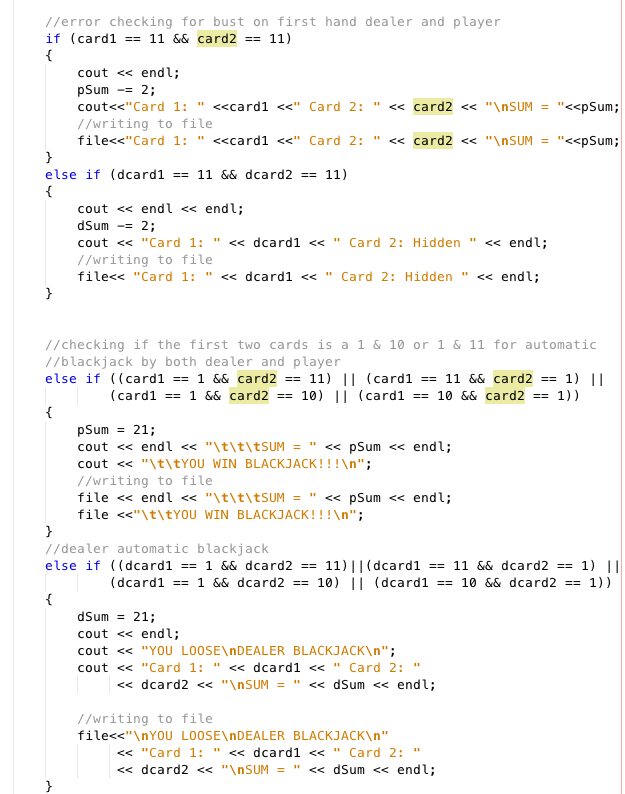
1. Greet and Ask for player name.
   1. Player input name
2. Dealer First two cards are dealt.
   1. Dealer card 1 is reveal
   2. Dealer card 2 is hidden
3. Player First two cards are dealt.
   1. Player cards are added up making sure player do not bust on first dealt or if player hit blackjack for an automatic win.
   2. If player sum is less than 21 player decides to hit or stay.
   3. Hit ->to deal a 3rd card and beyond or stay with current cards.
4. Hit
   1. Player sum is greater than 21, player loses.

b. Player sum is equal to 21, player wins .

1. Stay -> stay with current cards.
2. Dealer:
   1. Checking dealer sum to make sure dealer doesn’t bust on first dealt.
   2. Dealer sum is equal to 21 player loses
   3. Dealer hit if sum of first two cards are less than 17.
   4. Dealer stay when sum is equal to or greater than 17.
   5. Player win if dealer sum is greater than 21 during dealer hit.
3. Dealer sum and Player sum are being compare the greater value wins.
4. Game prompt player to play again y/n.
5. Yes: game starts over again, No: game ends, thank you for playing message displays.
6. Game record is written to “BJK.txt” file.

****

**DEVELOPMENT AND LOGIC**

1. System Libraries used:
   1. Iostream -> C++ library deals with namespace std
   2. cstdlib -> C++ standard library
   3. fstream -> file i/o, write to file
   4. ctime -> ctime library helps generate the random values.
2. Global Variable:
   1. ofstream file -> purpose to write results to file also implemented inside functions.
3. Function Prototypes:
   1. userIn -> user decision to play game again
   2. hitStay -> Player chooses to hit or stay
   3. dDeal -> Dealer cards are reveal and third card is dealt
   4. win-> Player wins
   5. lose -> Player loses
   6. bJack -> Player or dealer hit blackjacks
   7. pBust -> Player bust sum is over 21
   8. dBust -> Dealer bust sum is over 21
   9. push -> Game push dealer sum equals player sum
4. Variables Declared:
   1. card1 -> Player card1.
   2. card2 ->Player card 2
   3. dcard1 -> Dealer card 1
   4. dcard2 -> Dealer card 2
   5. dSum -> Dealer sum
   6. hit -> Player hit
   7. pSum ->Player sum
   8. count -> Game counter
   9. name -> Player name
5. Greeting and get user name.
6. Write info to file.
7. do –while uses char variable “userIn” to prompt player to end game or to continue.
8. Random numbers are generated for both dealer and player first two cards in a for loop, which loops about 52 times. The number 52 was chosen because it resembles a 52 deck of cards. Also sum is calculated right after first two cards are dealt.
9. Basic error checking, if not check both dealer and player have the possibility of having a sum greater than 21. Also, checks for blackjack from both the player and the dealer cards and subtract 2 from sum.
10. After Error checking, program proceeds to letting player choose either to hit or stay..
11. If player choose to hit, hitStay function is enabled and third card is dealt to player, sum is recalculated Player.
12. Player Sum go through Error checking
    1. greater than 21 player bust, pBust function is being enabled.
    2. Equal to 21 player hit blackjack, player win, bJack function is being enabled
    3. Player stay than player sum is pass back to main.
13. dDeal function becomes enabled, dealer cards are revealed. Dealer card 1 &2, and sum get pass by value to function.
    1. Dealer sum less than 17 dealer deals third card
    2. If dealer sum is greater or equal to seventeen and less than 21 then dealer sum get pass back to main.
14. Dealer sum and player sum are being compared either win, lose or push function becomes enabled. To determined which value is greater.
15. Player is prompt to play again or to quit.
    1. Play again process starts all over.
    2. Quit thank message is displayed
    3. Game ends
    4. Game play is written to file.

**Summary**

Blackjack seems so simple at first but for me it was quite a challenge. Especially having to pay attention to every little detail. I implemented a lot of if or else if statements either nested inside a loop or loops are nested inside of them. Blackjack deals mostly with numbers either the statement is true or false and if statements are best option for me at my current C++ level. One of the troubles I ran into for a couple of days, is trying to figured out how to stop my do-while loop inside the hitStay function as soonest the player hits blackjacks or the player sum goes over 21. Also, it become quite tedious trying to get the if else statements in the right place. Most of time were mostly spent experimenting with the code how the if statements are effected by a little changed or recoding some block of codes all over, or look at the code through a different perspectives. Though my code provide some of the basis of playing blackjack, during Project two I hope to get my blackjack running a little bit closer to how blackjack is being played at a casino, and a smarter A/I.

**Reference**

1. <http://www.wopc.co.uk/history/blackjack/blackjack>
2. <http://www.cplusplus.com/reference/cstdlib/rand/>
3. Textbook -> Tony Gadis 7th Ed.
4. Youtube.com C++ tutorial videos