

Project 1

<Blackjack (21)>



May 11, 2025

Name: Belarmino, J miguel

CIS-5 42565

**Introduction**

Title: Blackjack(21)

This game is a simplified version of the classic Blackjack card game played in casinos. The game consists of one player competes against a computer programmed dealer. At the start of the game the user inputs their name and then the player/user is given a random value between 2-21 which represent 2 cards but for now I only have it as numbers. At the same time the dealer/computer is given a random value between 2-11 which represent as one card. The player then chooses to hit/draw more cards or stand and depending on what value the user has it will also depend on what range of value the user can draw. After the player hit/draw cards without busting or going over the max value(21) then the computer will have a chance to draw more cards and same thing with the user, depending on what value the dealer has it will also depend on what range of value the dealer can draw, but there’s a twist the dealer must follows a simple rule: they keep drawing cards until their total reaches at least 17, then they stand. After both turns are done, the final totals are compared whoever highest value within set condition or reach 21 or closest to 21 will win the round. If both have the same total, the rounds can also end up with a draw.

**Summary**

Project size: about 157 lines

The number of variables: about 18

The number of Library: 6

This project includes many concepts that we learned from chapter 1-5. It also has many potentials to evolve for the next project. For example, the game could improve by organizing the code into functions, using arrays to represent actual playing cards with names and values and many more.

I picked this card game because it’s one of my favorite games to play. It made the project more engaging to me.

This project took me about 35 hours and still counting, from brainstorming, researching and sketching ideas on paper to writing and putting all the codes together. Even though there were many struggles along the way, solving each problem felt really rewarding especially when I got it to work on how I want it to work. Throughout the process, I kept thinking of new ideas and features to add, which leave me unsatisfied for some parts but overall, this was a great learning experience. I was able to reflect on what I’ve learned so far and even learned some more along the way. This game still has more potential and I look forward to make some improvements on it.

**Pseudo Code**

*Initialize*

*Prompt user to enter their name*

*Open the file*

*If open*

*While read file*

*If open*

*Read data of wins and rounds*

*Close file*

*Set found to true*

*If found*

*Then “Resume round”*

*Else*

*“New Game”*

*Set play to true*

*Do loop*

*Set player total to 0*

*Set computer total to 0*

*Player cards generate random draw between the range of 2-21*

*Add to player’s total*

*Computer cards generate random draw between the range of 2-11*

*Add to computer’s total*

*Show the player’s drawn card and total*

*Show the computer’s first drawn card and total*

*Do loop for player’s decision*

*Prompt the player to hit or stand*

*Input player’s move*

*If player chooses to hit*

*If player’s total is 10 or below*

*Player cards generate random draw between the range of 2-11*

*Else*

*Player cards generate random draw between the range of 1-10*

*Add to player’s total*

*Show the player’s drawn card and total*

*If player’s total exceed over 21*

*Show player busted*

*Exit loop*

*If player’s total equal to 21*

*Show player Blackjack winner*

*Increment player’s win count*

*Exit loop*

*While player choses hit*

*If player did not bust or go over*

*Prompt dealer’s turn*

*While dealer’s total is less than 17*

*If dealer’s total is 10 or below*

*Delaer cards generate random draw between the range of 2-11*

*Else*

*Dealer cards generate random draw between the range of 1-10*

*Add to player’s total*

*Show the player’s drawn card and total*

*If dealer’s total exceeds over 21*

*Show dealer busted*

*Show player win*

*Increment player’s win count*

*Else If dealer’s total greater than player’s total*

*Show dealers wins*

*Else if dealer’s total equal to player’s total*

*Show draw*

*Else*

*Show player win*

*Increment player’s win count*

*Increment round’s count*

*Calculate of win rate*

*Show win rate*

*Ask player if they want to play again*

*Input player’s decision*

*Set play true if input ‘y’ or ‘Y’ other wise false*

*While play is true*

*Open file*

*Write wins space rounds*

*Close file*

*Exit Program*

**Screenshots of input/output down below**

**Screenshots of input/output**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Major Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Variable Names | Description | Location |
| Constant Integers | MINCARD | Minimum card value | Global constant |
|  | MAXCARD | Maximum card value | Global constant |
|  | MAXVAL | Highest value allowed in Blackjack | Global constant |
| Integers | pCards | Player's card value | Inside main |
|  | cCards | Computer's card value | Inside main |
|  | pTotal | Player's total | Inside main |
|  | cTotal | Computer's total | Inside main |
|  | wins | Track total wins | Inside main |
|  | rounds | Track total rounds | Inside main |
| Float | winRate | Calculate win rate | Inside main |
| Character | move | Input For hit or stand | Inside main |
|  | again | Input to to play again | Inside do while loop |
| String | name | Player's name | Inside main |
|  | rWord | temporary file word | Inside input file loop |
| Fstream | in | Input file stream | Used for loading file |
|  | out | Output file stream | Used for saving file |
| Bool | play | Controls game loop | Game loop condition |
|  | found | Checks if player data was found | After loading file |