

Project 1

Title

Go Fish

course

CIS-17A

Author

Paola Alcala

Game introduction

52 cards

4 suits: Hearts, Spades, Diamonds, Clubs.

13 ranks: Ace, One, Two, Three, Four, Five, Six, Seven, Eight, Nine, Ten, Jack, Queen, King.

The cards are divided in suits and ranks. The goal is to collect as many complete sets of 2 suit matching cards. Collect cards needed from players or go fishing from the remaining cards.

How to play Go Fish

1. Deal 7 cards to each player
2. Player 1 chooses a player and asks for a card.
3. Either one of each players gives up the card or says "Go Fish".
4. If player is told to "go fish" they grab a card from the center pile and add it to their hand.
5. If player gets a set of 2 matching cards remove from hand and add to matching cards pile.
6. The player's turn will end when they went "fishing" for a card.
7. If player secures a card from another player the get another turn.
8. Continue until one player runs out of cards.
9. The game is won by the player who has collected the most matching set of 4 cards.

Development Summary:

Lines of code: 402

Comment Lines: 63

Total Lines of code: 465

Version 1:

Initial Deck Setup and Player Structure

I created a text file to hold the deck of cards, formatted with each suit on a separate line and cards separated by commas.

Used ifstream to read the file and print the card data to the compiler.

Split each line by commas and store each card in a 2D array.

Set up a Player structure to represent the players in the game.

Version 2:

Deck Shuffling and Save Functionality

Implement shuffling for the deck using the cstdlib and ctime libraries.

Saved the shuffled deck to a new file, overwriting it each time the deck is shuffled

Version 3:

Splitting the Deck and Creating Center Pile

I Create a center pile data structure that initially holds all shuffled cards and serves as the source for dealing. I Implement a function to deal cards to each player, removing them from the center pile as they're assigned to each player's hand.

Version 4:

Finding Matches in Players' Hands

Develop a function to search for matches in each player's hand by comparing the ranks of cards (e.g., "Ace of Spades" and "Ace of Diamonds").

Use a variable to track matched cards for each player and maintain a score for each match.

Versions 5:

Player Turn Logic for Requesting Cards.

I Implement a feature where players can ask each other for cards based on rank. I created a statement within a do-while loop to alternate between players as long as there are cards in the center pile. I created a function for players to request cards from each other.

Version 6:

Gameplay Enhancements

I remove matched cards from both players' hands after a successful request. I decided to display updated hands for both players.

I Prompted players to "Go Fish" if no match is found.

I updated my program so that when a player draws from the center pile, it updates and display their hand with the new card.

I included an announcement whether each player has the requested card or not. I updated my case switch so that the game ends when one of the player runs out of cards. The two scores are compared and a winner is announced.

Version 7:

Finally I cleaned up my code and removed unnecessary outputs that I implemented to test my code and make sure my data was being manipulated correctly.

Example Inputs and Outputs

```
Output
Project_1_V_6 (Build, Run) × Project_1_V_7 (B
Ready to play Go Fish?
You will need 2 players to play.
Enter the number of players: █
```

example output:
"Enter the number of player:"

example input: 2

example output:
"Enter the name for Player 1:"

example input: Paola

example output:
"Player: 1 Paola"

"Enter the name for Player 2:"

example input: Sunny

```
Enter name for Player 1: Paola
Player: 1 Paola
```

```
Enter name for Player 2: █
```

```
Output
Project_1_V_6 (Build, Run) × Project_1_V_7 (Build, Run) × Project_1_V_7 (Run) ×
*****
Welcome Players!
Player 1: Paola
Player 2: Sunny
*****

Players' Hands After Dealing:
Paola's hand: eight of Diamonds, nine of Diamonds, three of Diamonds, queen of Spades, eight of Spades, six of Clubs, nine of Spades
Sunny's hand: six of Spades, five of Spades, ace of Diamonds, queen of Clubs, two of Hearts, four of Hearts, ace of Spades

Match found: eight of Diamonds and eight of Spades
Match found: nine of Diamonds and nine of Spades

Paola's Matches: eight of Diamonds, eight of Spades, nine of Diamonds, nine of Spades
Paola's Score: 2

Paola's hand after finding initial matches: Paola's hand: three of Diamonds, queen of Spades, six of Clubs
Paola's Score: 2

Match found: ace of Diamonds and ace of Spades

Sunny's Matches: ace of Diamonds, ace of Spades
Sunny's Score: 1

Sunny's hand after finding initial matches: Sunny's hand: six of Spades, five of Spades, queen of Clubs, two of Hearts, four of Hearts
Sunny's Score: 1
```

example output:

Welcomes players. Shows you each hand. Searches for matches in initial deck. Takes a score of initial matches

Paola's turn.

Current hand: Paola's hand: three of Diamonds, queen of Spades, six of Clubs
Enter a rank to ask for (e.g., Ace, Two, Three): █

example output:

Shows your updated deck.
"Enter a rank to ask for:"

example input: three
(keep in lower case)

Sunny does not have any threes. Paola goes fishing!
Paola drew a two of Clubs from the center pile.

Paola's hand after asking: Paola's hand: three of Diamonds, queen of Spades, six of Clubs, two of Clubs
Paola's updated score: Paola's Score: 2

Sunny's hand after being asked: Sunny's hand: six of Spades, five of Spades, queen of Clubs, two of Hearts, four of Hearts

Sunny's turn.
Current hand: Sunny's hand: six of Spades, five of Spades, queen of Clubs, two of Hearts, four of Hearts
Enter a rank to ask for (e.g., Ace, Two, Three): █

example output:

Example of when a player does not have the card asked for.

example input:

"Sunny does not have any threes. Paola goes fishing!"

Paola has a queen of Spades!

New match found in Sunny's hand: queen of Clubs and queen of Spades

Sunny's hand after asking: Sunny's hand: six of Spades, five of Spades, two of Hearts, four of Hearts
Sunny's updated score: Sunny's Score: 2

Paola's hand after being asked: Paola's hand: three of Diamonds, six of Clubs, two of Clubs

Sunny's turn.
Current hand: Sunny's hand: six of Spades, five of Spades, two of Hearts, four of Hearts
Enter a rank to ask for (e.g., Ace, Two, Three): █

example output:

Example of when a player does have the card asked for.

example input:

"Paola has the card!"

```
-----  
Paola's hand after being asked: Paola's hand: four of Diamonds  
-----
```

```
Paola's turn.
```

```
Current hand: Paola's hand: four of Diamonds
```

```
Enter a rank to ask for (e.g., Ace, Two, Three): four
```

```
Paola asks Sunny for cards of rank four.
```

```
*****
```

```
Sunny does not have any fours. Paola goes fishing!
```

```
Paola drew a four of Clubs from the center pile.
```

```
New match found in Paola's hand: four of Diamonds and four of Clubs  
-----
```

```
Paola's hand after asking: Paola's hand:
```

```
Paola's updated score: Paola's Score: 8  
-----
```

```
Sunny's hand after being asked: Sunny's hand: queen of Hearts, jack of Spades, seven of Clubs
```

```
Game over! Final scores:
```

```
Paola's Score: 8
```

```
Sunny's Score: 15
```

```
The winner is Sunny with 15 matches!
```

```
RUN FINISHED; exit value 0; real time: 21m 59s; user: 0ms; system: 0ms
```

```
█
```

example output:

Example of when a player has an empty hand and the game ends.

The program compares scores and announces a winner.