Project 1

Poker

Course

CIS/CSC-5

SECTION

40404/0405/40414/40415

Due Date

February 02, 2024

Author

Mariam Mikhail

Table of contents

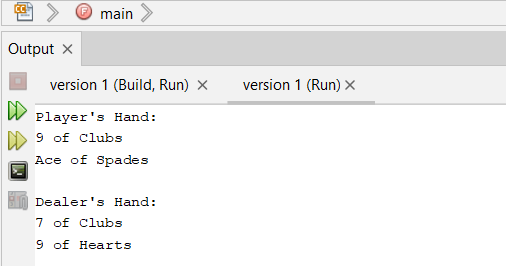
| **INTRODUCTION**………………………………………………………………  **Gameplay and rules**……………………………………………………………… | 3  3 |
| --- | --- |
| Version 1………………………………………………………………………….  Version 2………………………………………………………………………….  Version 3………………………………………………………………………….  Version 4…………………………………………………………………………. | 4  5  6  7 |
| Card game on C++………………………………………………………………  Flaw Chart………………………………………………………………………... | 8  9 |

Introduction:

The Poker project aims to implement a text-based version of the classic card game Poker. The project utilizes fundamental programming concepts such as primitive data types, conditional statements, looping, user input, output formatting, file I/O, and error handling.

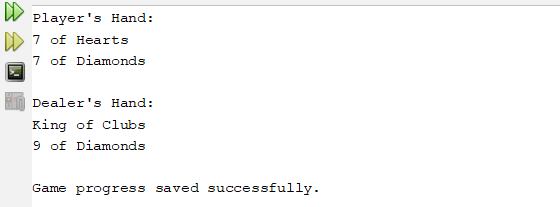
Gameplay and rules:

Poker is a classic card game where players compete to form the best hand possible using a standard deck of 52 cards. Players are pitted against a computer dealer in this text-based version of Poker. Each player starts with two cards; the goal is to achieve a hand value as close to 21 as possible without exceeding it. The value of each card is determined by its rank, with numbered cards having their face value, face cards (Jack, Queen, King) worth 10 points each, and Aces initially valued at 11 points. Players can 'hit' to receive additional cards or 'stand' to keep their current hand. After the player's turn, the dealer reveals their cards and draws additional cards until their hand value reaches at least 17. The winner is determined by comparing the total values of the players' and dealers' hands. The game combines strategy, probability, and risk assessment, making it both challenging and entertaining. Enjoy the thrill of Poker and aim for that winning hand.

Version 1:  
  
  
We have implemented basic game logic for Poker.

* Used primitive data types for cards, player hands, and scores.
* Incorporated conditional statements for game decisions.
* Utilized looping for repetitive actions like dealing cards and determining winners.
* Accepted user input for actions like betting and discarding cards.
* Displayed output with basic formatting for game updates.
* File I/O or error handling has yet to be implemented.

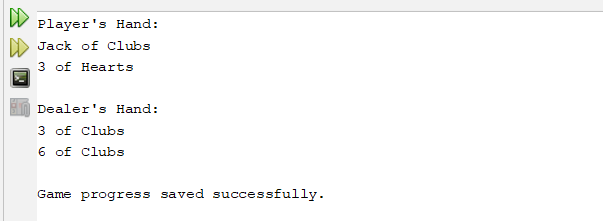
Version 2:



Enhanced User Interface

In Version 2, the user interface is enhanced for better usability. More explicit prompts and messages are provided to guide the player. Error handling is added to validate user inputs and provide informative messages for invalid actions. Basic file I/O functionality is integrated to allow players to save and load game progress.

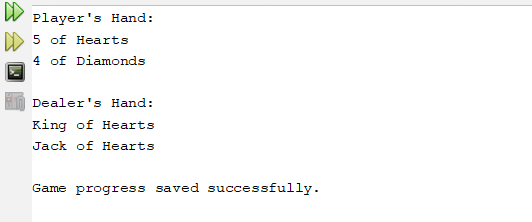
Version 3:



Refinement and Bug Fixes

Version 3 focuses on refining the codebase and addressing any bugs or issues identified in previous versions. The code is refactored for improved organization and readability. Bugs related to game logic and user input validation are fixed. More robust error handling is implemented to handle edge cases and unexpected inputs.

Version 4:



Final Polishing

The final version, Version 4, involves polishing the project for submission. Detailed comments are added to the codebase to enhance documentation and readability. Thorough testing is conducted to ensure the stability and correctness of the game. Additionally, code optimizations are made where possible to improve efficiency.

Card Game: Poker

Here is a picture of the program running the game. In this picture, you will see the player’s hand and the value of the cards in his/her hand. Furthermore, the program will give you options to choose from according to the hand you have. When you pick the program, it will give you the shape, number of new cards, and total value. Ultimately, it will tell you which player won, you or the dealer, in this case, the program.



Flow chart:

