BBM203

Programming Assignment 1

Prepared by:
Özge Kökyay

Problem Definition

I was expected implement an application that is a game named Klondike is the classical version of Solitaire played via file input and output. I was also expected to only use arrays as data structure. The application should be able to deal cards to the table after reading deck file and then print the result of every moves after executing commands and should be able to follow the rules of Klondike and handle the restrictions about how the game should operate.

Solution Approach

I started to project from reading input files and I implemented several arrays to keep data of the cards. I made an array named **deck** that keeps all cards for initial dealing into piles. I used **upper triangular matrix** while initializing tableau area. Another array named **stock** keeps the data of cards in stock.
While using arrays as a data structure, the hardest part was the boundaries of the arrays because of that arrays has no bound check and it needs to pay too much attention to prevent segmentation fault.

Thanks god there are data structures other than arrays.

Solution Approach and Class Diagram

I developed the assignment in only one file **Main.cpp.** I did not need more files because I simply implement it like this.
In **Main.cpp** I build two classes that named **Solitaire** and **Main**.
Main class simply reads arguments from command shell and send them into **Solitaire class**.
Solitaire class has two main function named **fileReader** and **printTable**.

- *fileReader* initializes the arrays that used for the game and reads command and execute them.
- *printTable* prints table after every command into the output file.

Explanation of Arrays

These arrays are holding cards in the foundation area.

- string heart[15];
- string diamond[15];
- string spades[15];
- string clubs[15];

This array is holding all cards with the same order in the file

- string deck[DECK_SIZE];

This array is holding cards that will deal into the tableau.

- string rem_deck[REM_DECK_SIZE];

This array is holding cards that will keep in the stock.

- string stock[STOCK_SIZE];

This is a multidimensional array holding 7 piles and maximum 20 cards for each pile.

- string table[ROW][COLUMN];

This array is holding cards that can be seem to the player.

- string visible[50];

This array is holding cards in the waste area.

- string stockArr[3]={"0","0","0"};

This array is holding cards remaining from stock.

- string stockRem[24];