62410 | Machine oriented programming

Projekt 2 - Deadline 10/05 - 2022

Gruppe 16

Salim Belal Omar

Brugernavn	Email	Studienummer	Uddannelse
s193472	s193472@student.dtu.dk	s193472	diploming. Softwaretek.

Omar Abdulrahim Ebrahim

Brugernavn	Email	Studienummer	Uddannelse
s163576	s163576@student.dtu.dk	s163576	diploming. Softwaretek.

Thomas Arildtoft

Brugernavn	Email	Studienummer	Uddannelse
s193564	s193564@student.dtu.dk	s193564	diploming. Softwaretek.

Youtube link to the video

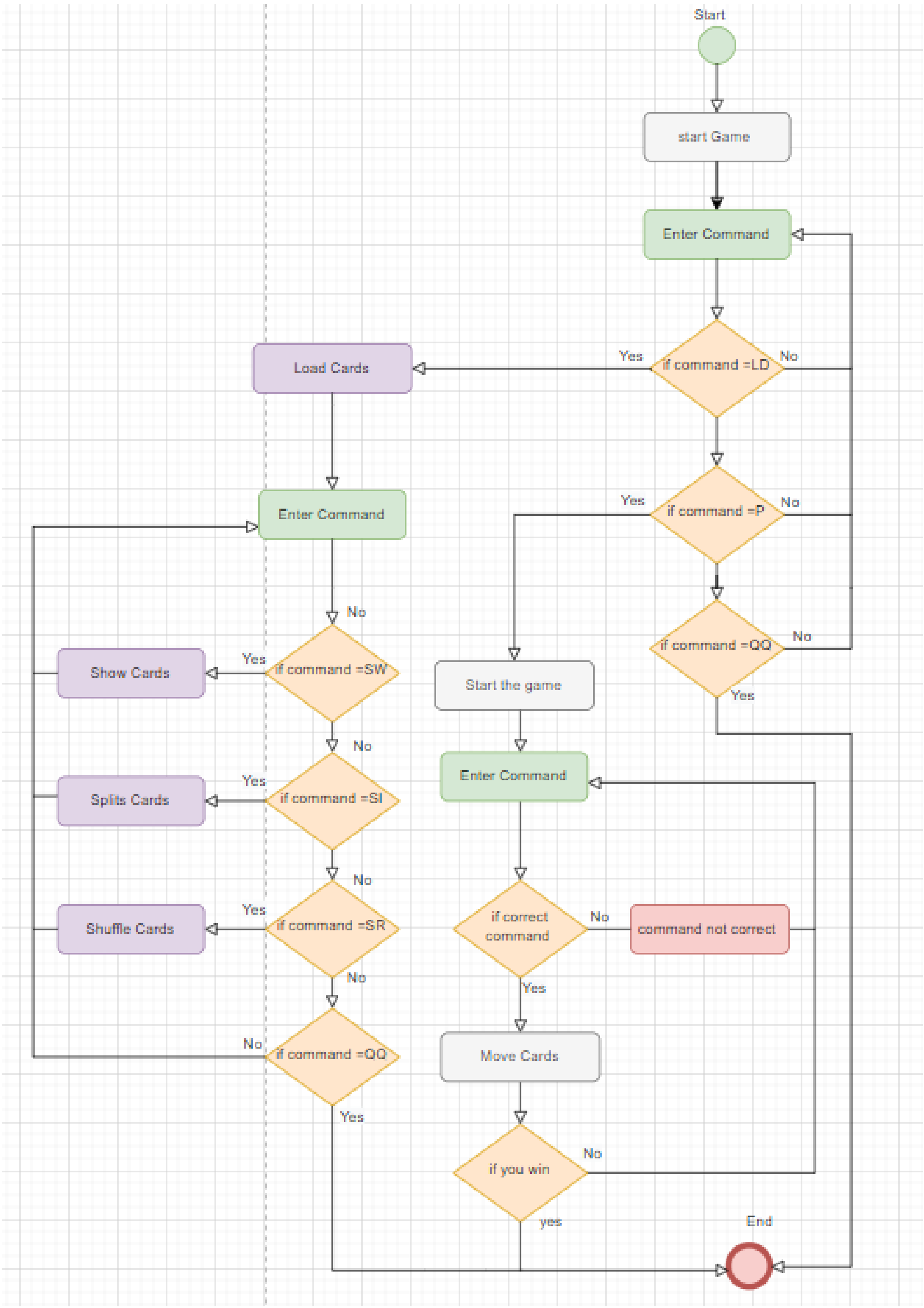
Indholdsfortegnelse

Requirements

- The game must be implemented in C.
- Cards from the deck and columns need to be stored in a linked list and not in an Array.
- The game has 52 cards and 4 suits
- First character must represent the value of the card (A = Ace, cards 2 - 9 as numbers, T = 10, J = Jack, Q = Queen and K = king)
- The second character represents the suit of the card (C = Clubs, D = Diamonds, H = hearts and S = spades)
- Each card should be represented by 2 Characters meaning 4C = 4 of Clubs.
- If the card is not visible, then we just use []
- Before the cards is displayed, the raw board must be setup containing the 7 rows C1 - C7 and the 4 foundation piles F1 - F4
- At the bottom, 3 lines named
  - **Last Command** > Shows last msg that was inserted into **Input**
  - **Message** > Either an error message or an OK if the last command was executed or not
  - **INPUT** > Input new commands.
- The program, if completed, should be supporting 9 commands.
  1. LD <filename>
  2. SW
  3. SI <split>
  4. SR
  5. SD <filename>
  6. QQ
  7. P
  8. Q
  9. <Game Moves>

Design

Flow Chart diagram



Analysis

Terminal Window	This is the window from where we can see the game overview. I should show the 7 Columns, the foundation and the 3 lines. - Last Command - Message - Input
Deck of cards	A deck of cards that consist of 52 divided into 4 suits and from Ace to King. The deck is divided into 7 columns where 31 cards are visible and the remaining is invisible.
Suits	The 4 suits we know from a normal deck of cards Diamonds = D, Clubs = C, Hearts = H and Spades = S.
Foundations	The four foundations are labeled F1 - F4, the cards from the columns must be moved in order of suit and from Ace to King.
Columns	The 7 Columns C1 - C7 where the cards are dealt into, the cards need to be dealt in the order left to right 1-6-7-8-9-10-11 cards.
Message	The message shows if the last command had an error or not, if it displays " OK " everything is as the word says ok, if it says error, there is something wrong.
Input	Where we input the commands for the game.
Last command	Visible in the terminal window, this line shows the last name that was inserted in the input prompt.

LD < filename >	<filename> represents the name of the file,

	that contains the deck of cards we wish to load, if this is your first time playing just enter “ LD “
SW	Shows all the cards the card deck contains, in the order they have in the deck.
SI < split >	Command to split the deck, first it splits it into two piles of cards.
SR	Command to shuffle the cards randomly.
SD	Saves the cards from the card deck into a file that is specified by < filename >
QQ	Command to quit the game.
P	Command to start a game with the current card deck and put the game into the play phase.
Q	Command that quits the game and then goes back to the start phase.
<Game Moves>	Command to move card.

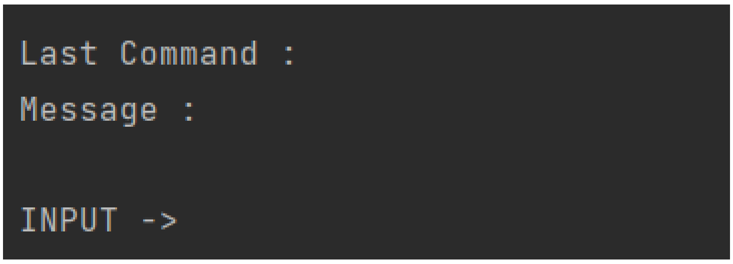
Implementation

Git hub: https://github.com/OmarEbrahim199/Gr_16_Project_2

Notices.
If the game not working and you can not load the cards **please change the working directory** in **CLion** by go to **Run/Debug Configurations** and then **Working directory** and select for example >>
C:\Users\YourUserName\CLionProjects\Gr_16_Project_2

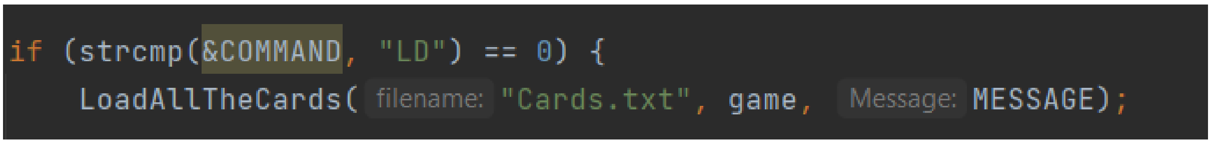
We have tried to make the code more simple and not that sophisticated by implementing it.
There are several classes in the project like header files and main files.
We have faced a lot of difficulties during code writing, but in the end it works.

when the game starts the terminal going to shows these messages
LAST Command and Message and INPUT > as it shows down.



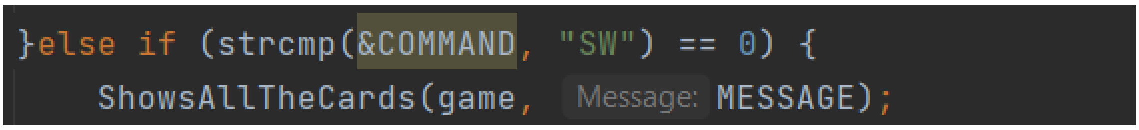
```
Last Command :
Message :

INPUT ->
```

if the command equal to <LD> load the card the program will execute that mathed



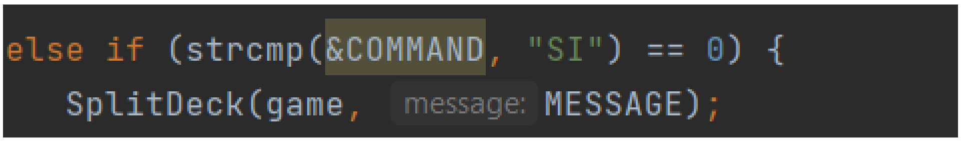
```
if (strcmp(&COMMAND, "LD") == 0) {
    LoadAllTheCards( filename: "Cards.txt", game,  Message: MESSAGE);
```

Else if the command equal to <SW> reveal the card the program will execute that mathed



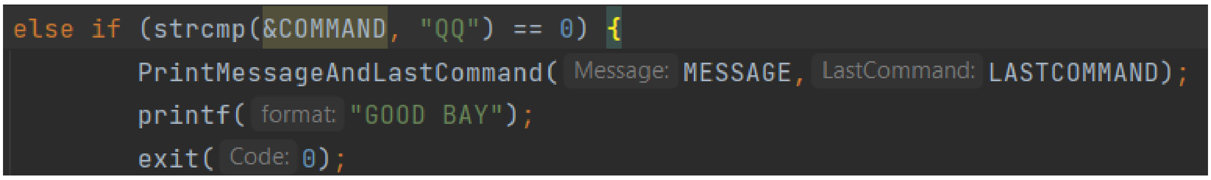
```
}else if (strcmp(&COMMAND, "SW") == 0) {
    ShowsAllTheCards(game,  Message: MESSAGE);
```

Else if the command equal to <SI> and split deck the card the program will execute that mathed



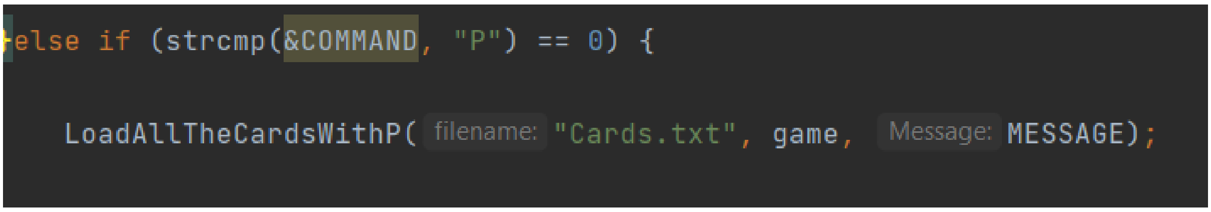
```
else if (strcmp(&COMMAND, "SI") == 0) {
   SplitDeck(game, message: MESSAGE);
```

Else if the command equal to <QQ> ends the program.



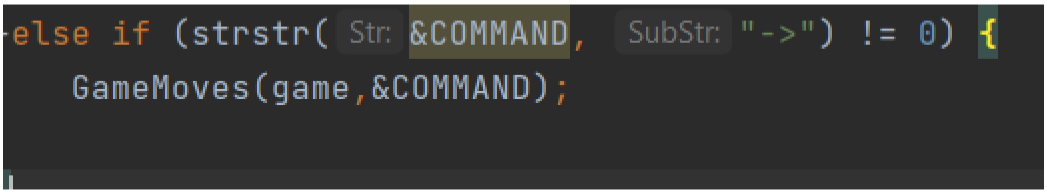
```
else if (strcmp(&COMMAND, "QQ") == 0) {
        PrintMessageAndLastCommand( Message: MESSAGE, LastCommand: LASTCOMMAND);
        printf( format: "GOOD BAY");
        exit( Code: 0);
```

Else if the command equal to <P> move to play mode



```
else if (strcmp(&COMMAND, "P") == 0) {

   LoadAllTheCardsWithP( filename: "Cards.txt", game, Message: MESSAGE);
```

Else if the command contains < -> > move that card from/ to the selected piles.



```
else if (strstr( Str: &COMMAND, SubStr: "->") != 0) {
   GameMoves(game,&COMMAND);
```

Conclusion

We have put our main focus on the functionality of the game, meaning all the functions that are required works, but we haven't had enough time to create and test our game. So if we had more time, that is where our focus would lie.

Team member contribution

Task	Omar	Salim	Thomas
Project structure	5 timer	6 timer	7 timer
Analysis	4 timer	5 timer	5 timer
Design	7 timer	9 timer	10 timer
Game Logic	4 timer	4 timer	4 timer
Implementation	59 timer	54 timer	52 timer
Video	1 timer	1 timer	1 timer

Appendix

https://www.tutorialspoint.com/cprogramming/c_pointers.htm
https://www.tutorialspoint.com/cprogramming/c_header_files.htm
https://www.tutorialspoint.com/cprogramming/c_file_io.htm

https://www.geeksforgeeks.org/linked-list-set-1-introduction/

https://jborza.com/games/2020/07/12/solitaire-cli.html

Books

Introduction to computing systems, from bits and gates to C/C++ and beyond, 3rd Edition by Yale Patt and Sanjay Patel:

Chapter 14 Functions

Chapter 15 testing and debugging

Chapter 16 Pointer and array Chapter 18 I/O