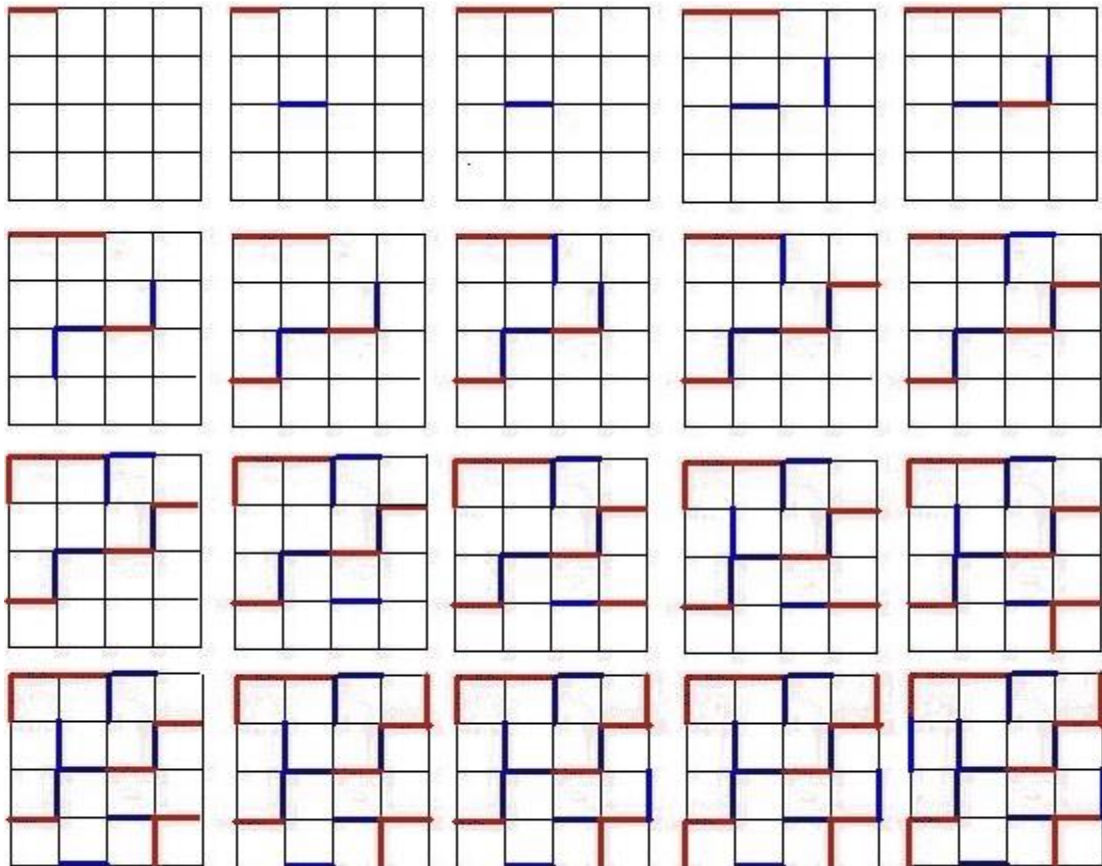


# Dots and boxes Report

## Description:

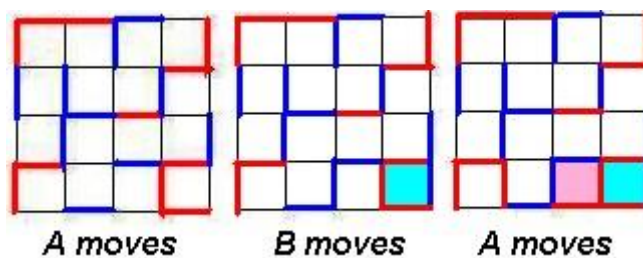
Dots and boxes is a two-person game based on a rectangular lattice of points. Each player, in turn, draws a horizontal or vertical line connecting two adjacent points. Whenever placement of a line complete a single  $1 \times 1$  square, the square is colored in, the player scores one point, and the player having completed the square receives an additional move.



*Red: Player A*

*Blue: Player B*

In the first part of the game, the players will avoid adding the third side of a square.

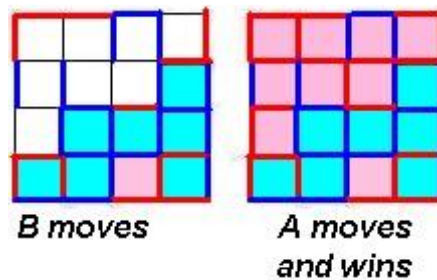


*A moves*

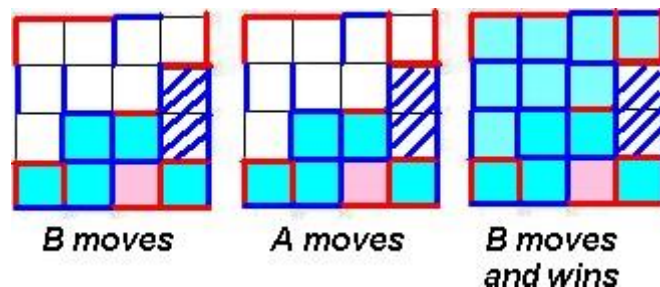
*B moves*

*A moves*

As soon as this is no longer possible, the player (A) who makes the next move will enable his opponent B to form a sequence of squares, called a chain. Player (B) will end his move by adding a (A) segment which does not close any square and possibly opens a new chain to .



The best move is not necessarily the one closing as many squares as possible and opening the shortest chain to the opponent.



Sometimes it is more convenient to force the other player to open a chain that it is longer than the one currently available. This can be achieved, for example, by completing all but two squares of the available chain and forming a rectangle (a so-called [domino](#)) out of the last two.

The winning strategies of this game are very complex and have not yet been completely classified. In 2001, David Wilson showed the  $3 \times 5$  game is a win for the second player.

- **Features :**

1. Different board sizes to choose from {2 – 7 }
2. Game saves so you can come back anytime and saves any number of games
3. You can load from a list of saved games to play (unlimited number of games)
4. Easy to use interface and graphic effect
5. Unlimited number of Undo and Redo
6. Showing top-10 games of the top-score users
7. Time-passed up dating each game

## Design Overview:

Creating the main 2-D array:

When the user enter the number of points the program creates an array with size  $(2*n)-1$

When the game started , the program ask the user if he want to play human vs human ask for their name OR human vs computer ask for player 1 name

then start a loop player1 turn , player2 turn ... and so on . but if any player form at least one box , he will take another turn

\*top 10

After the round has ended the players' names and their scores are saved and sorted in file "player's rank" to be display in top 10 section

\*Undo/Redo:

For making unlimited number of undo and redo we save the 2-D array of the grid in 3-D array so we can save each turn in an index and call it back when the user undo and in redo we call the next index if the redo .

The user cannot redo without making undo before that.

\*Save/Load:

When the user press "save", the program creates a new file with the name of the two players and saves the game to it. and if the players still playing and they made change in the game then press save the file will be updated.

Therefore, in this way we can save multiple number of games.

When we load a game, the program shows a list of the saved games and ask the user to choose a game to load.

## Assumptions:

Maximum number of point in the grid is 7 points.

## Data Structure:

3-D arrays / 2-D arrays / 1-D array

Pointers - Strings

- Description of the important functions:

1.Main menu: Display 4 choses to the user [start game – load game – top 10 - exit]

\*And from "start game" user should select human vs human OR human vs computer

And from "load game" user should choose

2.grid: Display Whose player turn is, Number of moves for each player, Scores of each player, Number of remaining lines, Time passed in minutes and seconds.

3.Rank: when the game ended, program append the score of each player to the file of ranking then sort scores from higher to lower

\*So when user want to show the top 10, the program will open this file and display the first 10 scores.

4.formbox: to check if this player's turn formed any boxes and if this turn formed at least one box give him another turn

5.Input: take the input form user and make sure this input is valid input

6.Undo/Redo: For making unlimited number of undo and redo we save the 2-D array of the grid in 3-D array so we can save each turn in an index and call it back when the user undo and in redo we call the next index if the redo.

\*The user cannot redo without making undo before that.

7.Save/Load: When the user press “save”, the program creates a new file with the name of the two players and saves the game to it. and if the players still playing and they made change in the game then press save the file will be updated.

\*Therefore, in this way we can save multiple number of games.

\*When we load a game, the program shows a list of the saved games and ask the user to choose a game to load.

Flow chart and pseudo code:

User Manual:

Sample runs:

\*Inter face(welcome screen):



\*Main menu :

```
=====Dots and Boxs=====
1-START PLAY
2-LOAD GAME
3-Top 10
4-Exit

What would you like to do :
```

\*Select human vs human

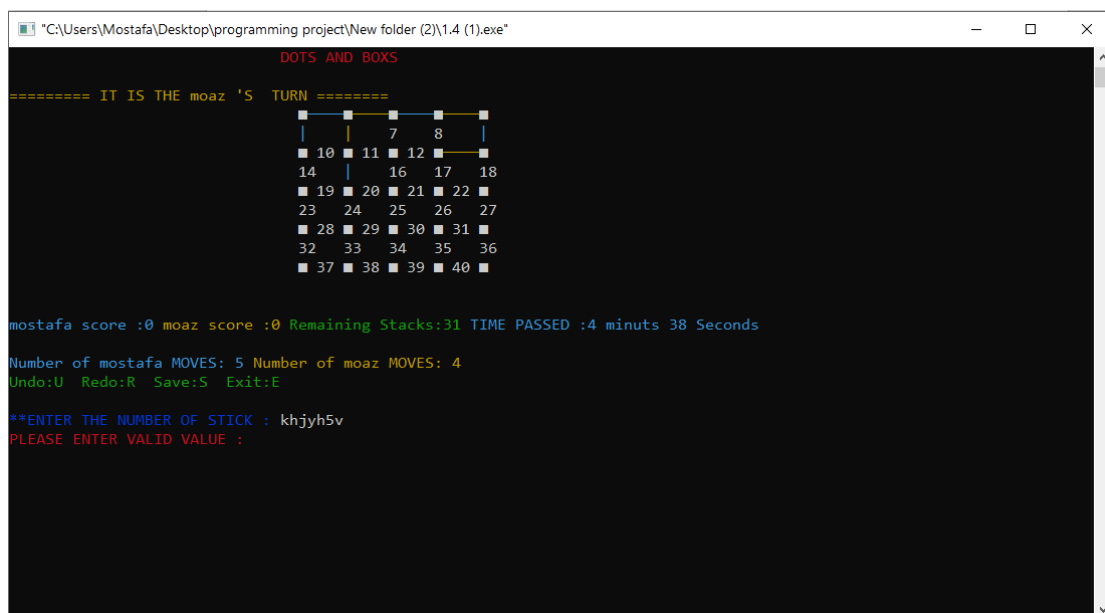
(OR) HUMAN VS COMPUTER :

```
1-PLAYER 1 vs PLAYER 2
2-PLAYER 1 vs COMPUTER
(1 or 2) :
```

\*ENTER name of player 1 and 2 then number of points:

```
PLEASE ENTER PLAYER 1 NAME : mostafa
PLEASE ENTER PLAYER 2 NAME : moaz
ENTER NUM OF POINTS : 5
```

When enter input invalid, program will ask user to input valid number :



The screenshot shows a Windows application window titled "C:\Users\Mostafa\Desktop\programming project\New folder (2)\1.4 (1).exe". The application displays a 4x4 grid of numbers from 1 to 40, arranged in 4 rows and 10 columns. The grid is as follows:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

Below the grid, the text "mostafa score :0 moaz score :0 Remaining Stacks:31 TIME PASSED :4 minuts 38 Seconds" is displayed. A menu is shown with the following options: "Number of mostafa MOVES: 5", "Number of moaz MOVES: 4", "Undo:U", "Redo:R", "Save:S", and "Exit:E". The prompt "\*\*ENTER THE NUMBER OF STICK : khjyh5v" is shown, followed by the message "PLEASE ENTER VALID VALUE :".

Player 2 won the round :

```
"C:\Users\Mostafa\Desktop\programming project\New folder (2)\1.4 (1).exe"
DOTS AND BOXS

| B | B | B | B |
| B | A | B | A |
| B | A | B | A |
| B | B | B | B |

mostafa score :4 moaz score :12 Remaining Stacks:0 TIME PASSED :7 minuts 11 Seconds
Number of mostafa MOVES: 17 Number of moaz MOVES: 23
Undo:U Redo:R Save:S Exit:E

congratulations ,moaz YOU WON THIS ROUND :)

EXIT:E MAIN MENU : M
(E or M) :
```

Game tied:

```
"C:\Users\Mostafa\Desktop\programming project\New folder (2)\1.4 (1).exe"
DOTS AND BOXS

| B | B |
| A | A |

mostafa score :2 akram score :2 Remaining Stacks:0 TIME PASSED :0 minuts 19 Seconds
Number of mostafa MOVES: 6 Number of akram MOVES: 6
Undo:U Redo:R Save:S Exit:E

THIS GAME IS TIED

EXIT:E MAIN MENU : M
(E or M) :
```



## References:

<https://www.geeksforgeeks.org/>  
<https://en.cppreference.com/w/c>  
<https://stackoverflow.com/>  
<https://www.quora.com/>