DOTS\_and\_BOXES

Game Description:

Dots and Boxes is a game for two players : two players or man vs computer . Starting with an empty grid of dots, two players take turns adding a single horizontal or vertical line between two unjoined adjacent dots. The player who completes the fourth side of a 1x1 box (or groups of one or more adjacent boxes) earns one point (s) and takes another turn the player with higher score is the winner .

Features:

our game has important features :

1. The game's colors are consistent
2. Time from starting game
3. Computer has needed intelligence to close boxes
4. Undo and Redo
5. Save and load the game
6. Ranking top 10
7. The game is made on git hub
8. We use header files
9. There is some intelligence in computer turn
10. There is important feature added to the game it is three levels not only two

We can say that these features make the game efficient for playing and enjoying.

Overview:

the program is divided into header files that have functions used in the main file to create this program we have some sub problems to build it

1)we want to display the interface main menu that tells the user to choose one of the operations to start new game or to load saved game or to see the top ten that have high score

we have conditions

2)if the user choose to see top ten it will appear the a new window with top ten and their scores

3)if the user choose to load a saved game then there will be a window ask him to choose the required file if he choose the file true the saved game will appear

4)if the user want to start new game a new window will appear ask him to choose the level he want to play easy or medium or hard after that he chose new window will appear asking him to choose the mode of the game one player against the computer or two players

after that the game will appear and the information of every player the players names , number of moves of each player and the score of each player and we must create general information like the number of lines remaining for the end of the game then you can choose the row and column to put the line after any change occur there will be we need to change color by changing the player that have the turn to play

5)after the game ended the window must have these messages the name of the winner with congratulations and asking the user if he want to quit the game or return to main menu

So the solution of these problems will be explained in the algorithm and it will be exchanged to code that create that wonderful game

Assumption:

There is no assumption our program is ready to any test cases

Data structures:

We used 3 arrays

1. compgame
2. gamearr
3. movesplayed

we used 2 structures

1. player
2. users

Description of the important functions and Header files :

1. In header file “color.h” function SetColor is used for changing colors in the game depending on the color you pass
2. In header file “scannum.h” function readint that is so important to get numbers as inputs from the user and preventing program from crashing
3. In header file “printgame.h” function printgrid is used for printing the game and informations of players
4. In heaeder file “check\_boxes.h” function checkboxes is used to check if the box is closed to shad it
5. In header file “computer.h” function compturn is used to make computer play with some intelligence
6. In header file “undo.h” function undo is used to retrun to the previous move
7. In header file “playerinfo.h” initializing of structures player anad users
8. In header file “gamefunction.h” function game function is used to make the game loop it is the mother function of the game as all functions included into it

algorithm:

check box:

to close shade boxes we will make function called check box that checks the if the box is closed by this move so the function is

we check if the line he chose is horizontal so we will check the box above and down of it how ? we see if the horizontal line above it is existing or not and we check if the vertical line above the line to right and to left if they all are found then the box will be shade by the letter A if the first player was putter to the last line in the box and the letter will be B if the second player who closed the box then check the box under the horizontal line firstly we check if the horizontal line under the entered line then check if the vertical line under the entered line from right and left is existed or not if the first player then put A otherwise put B

we do the same logic with vertical lines to check the against line then the horizontal lines closing the box then if all these lines were existed with the symbol specified for the player

undo:

we creat a 2d array that carry the information of the game if the game is played it will carry the at the first column the number of the row of the line put the second column the col of the line entered the third col the number of player that should play the fourth column is the number of row of the symbol that close the box at right or down the fifth column is for the column o f the col of the symbol closing the box the sixth column is for the number of row of the symbol closing the box at left or up the seventh column is for the number of column of the symbol closing the box at left or up and the rows of the array refer to the number of the move -1 as an example the first move you can find at the row 0 and so on if we want to undo we remove the last move and put the previous move

redo:

before any redo there must be undo has done so we call the next move from the array that carry the all moves of the game

t1= calculate time at the first of the game

noundo =0

for (j=0 to j<size )

{compgame[0][j] = j

compgame[j][0] =j

if (j%2!=0)

{for ( I =1 to i<size )

{compgame [i][j] = dot

Compgame [i][j] =v

I=i+2}

}

Else if (j%2 ==0 )

{

For ( I =1 to i<size)

{ compgame [i][j] =dot

Compgame [i+1][j] =v

}

}

gameon =1 and check =0

call printgrade function

while (gameon){

diplay (enter number of row and column by hexadecimal digits

enter 0, 0 for undo1, 1 for redo 2, 2 for save 3, 3 for main menu )

specify the color of displaying according to the player

if(gamer==1)

SetColor(Blue)

else

SetColor(Red)

Display (player 's turngamer)

if(gamer==2 and mode == 1) //computer turn function

{

Call computerturn function

}

else

{

if(gamer==1)

{

SetColor(Blue)

}

else

{

SetColor(Red)

}

Display(ENTER ROW)

row = call readint() function

display(ENTER COL)

col =call readint() function

}

SetColor(White)

system(clear)

ai =0

if(row==0 and col==0) //function of undo

{

Call undo function

}

else if(row == 2 and col == 2) //save game

{

s\_n=0

while(1)

{ s\_n=readint() function

if (s\_n==1or s\_n==2 or s\_n==3)

{

dispplay(chose file you want to save (1 or 2 or 3 )

saving file with needed informations

} else

{display(invalid file)}

else if (row == 3 and col == 3) //if user want to return to main menu

{

gamer =0

}

else if( row > 0 and row < size and col > 0 and col < size )

{

if(compgame[row][col] == h or compgame[row][col] == v or (row == 1 and col == 1))

{

if (row == 1 and col == 1) //for redo

{

if (noundo>0)

{

row =movesplayed[nofmoves][0]

col=movesplayed[nofmoves][1]

gamer=movesplayed[nofmoves][2]

noundo-=-noundo-1

}

else

{

display(there is no moves to redo )

call printgrid function

}

}

else if ( gamer==1)

{

noundo=0

}

//check if we played this move before

for(i=0 to i < nofmoves)

{

if(row == movesplayed[i][0] and col == movesplayed[i][1])

{

availablemove = 0

printf(not available move)

break;

i++

}

}

if(availablemove)

{

gamearr[row][col]=compgame[row][col]

movesplayed[nofmoves][0]=row

movesplayed[nofmoves][1]=col

movesplayed[nofmoves][2]=gamer

if(gamer == 1)

{ moves1++}

else

{moves2++}

nofmoves

call check box function

if(check)

{

if(gamer == 1)

\*score1 += check

else

\*score2 += check

}

else

{

if(gamer==1)

gamer=2

else

gamer=1

}

}

else

availablemove = 1

}

else

diplay(not available move)

}

else

display(not available move)

call print grid function

if(nofmoves == totallines) //end the game

{

gameon = 0

diplay(END GAME)

if(\*score1 > \*score2)

{

SetColor(Blue)

display(congratulation nameof player you are the winner)

SetColor(Green);

gamer = 1

}

else if(\*score2 > \*score1)

{

SetColor(Red);

display(congratulation name of player you are the winner)

SetColor(Green)

gamer = 2

}

else

{

Display (Tie Game)

gamer = 0

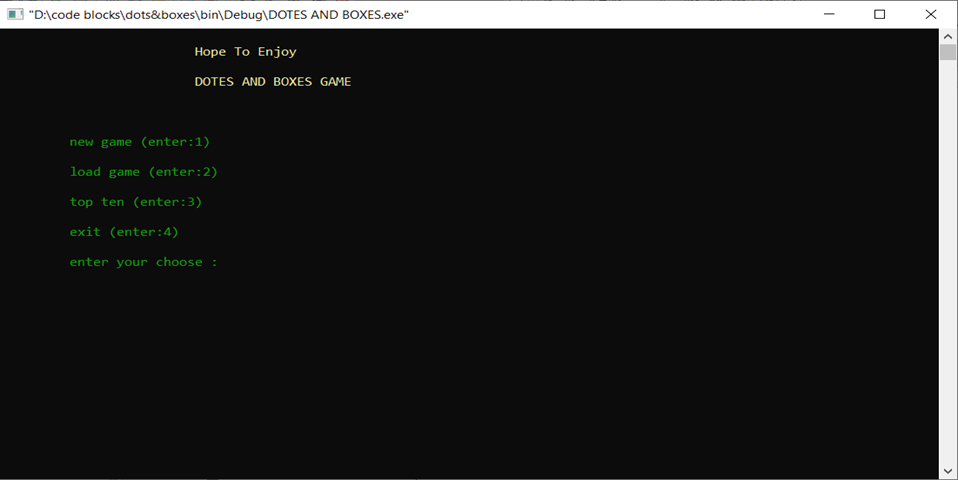
}

}

User Manual:

When you open game :

You will see this window

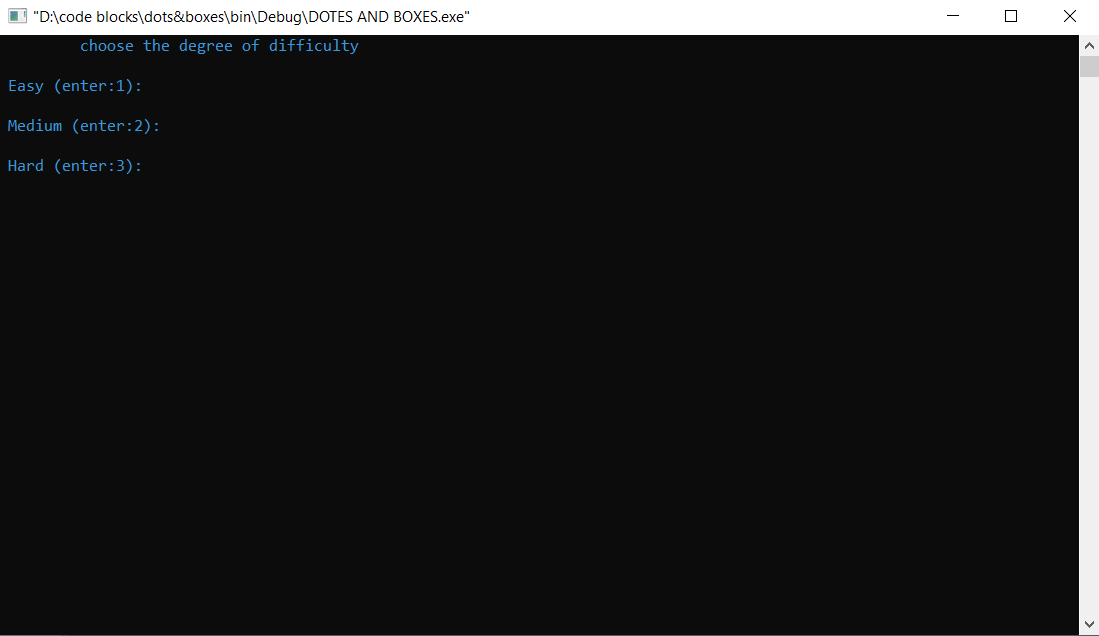


To open new game write 1 then click enter

To load saved game write 2 then click enter

To see top ten played the game write 3 then click enter

To exit the game write 4 then click enter

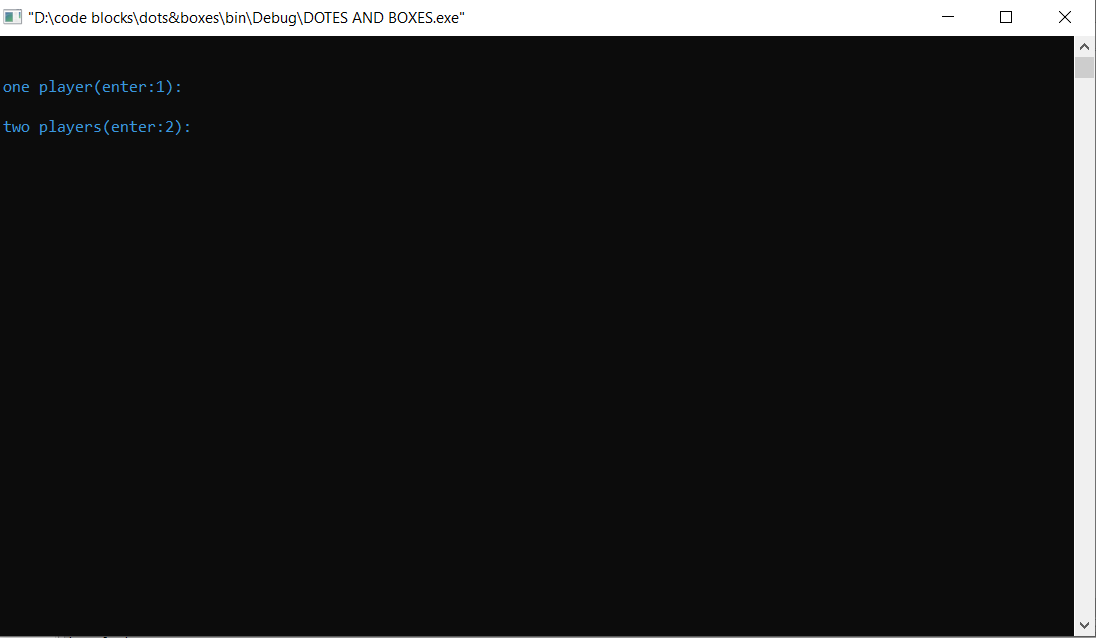
When you choose to open new game:

To play easy game Click 1 then enter

To play medium game Click 2 then enter

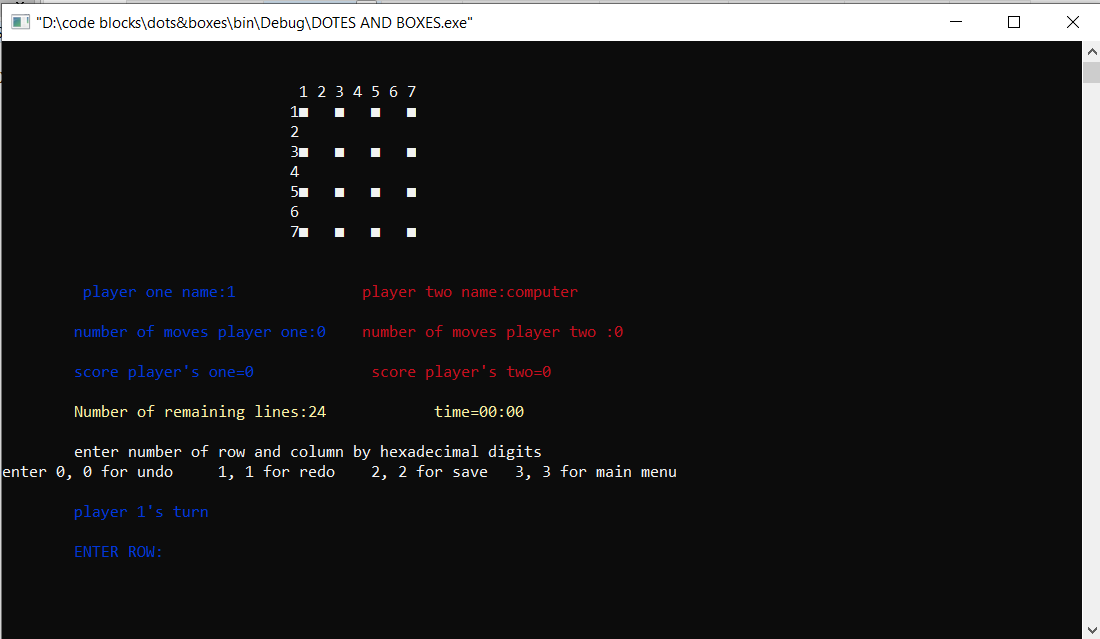
To play hard game Click 3 then enter

When you choose the level you will find:



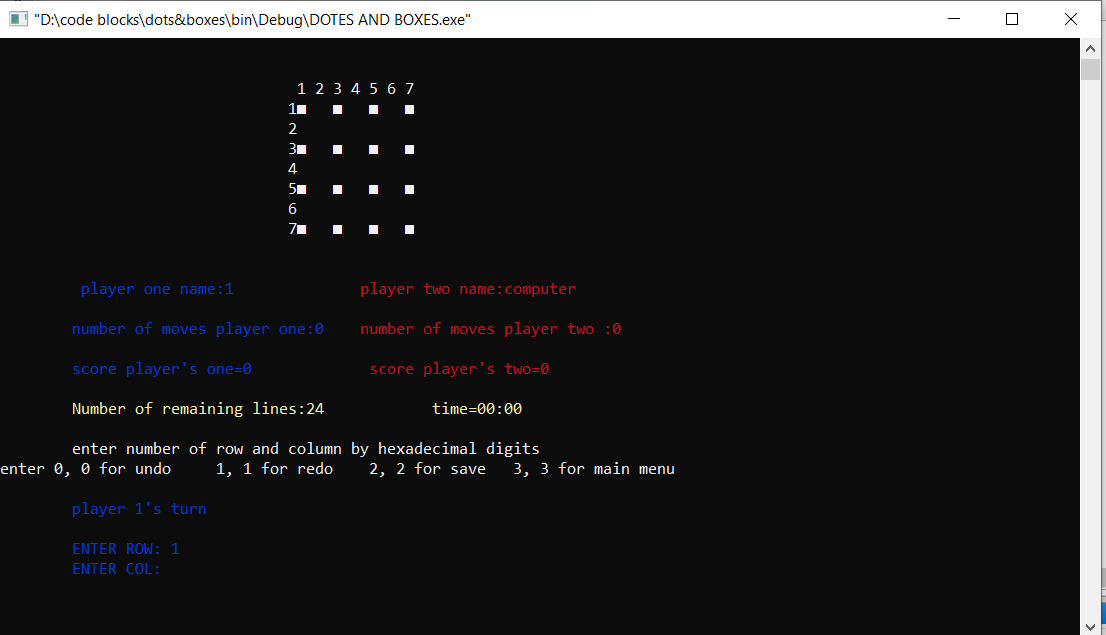
To play with computer choose 1 to play with a friend choose 2

The game will appear how to play it

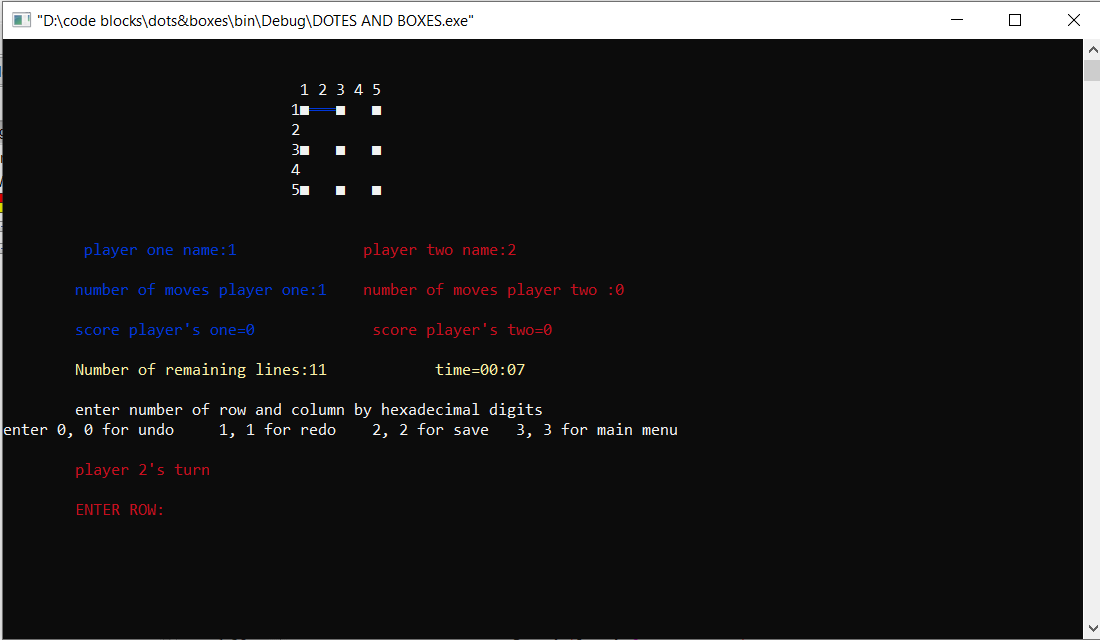


Now you want to specify the place of the line you want to draw

The game will ask you the number of row you will write then click enter

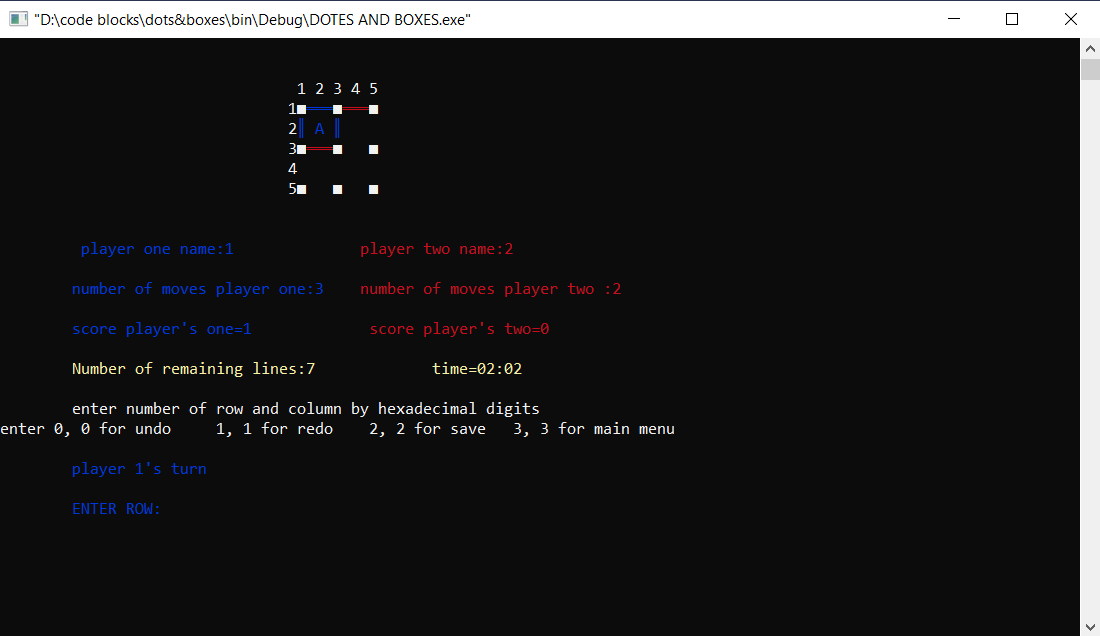


Then the game will ask you the number of col write it then click enter



The line is drawn then the second player will take turn and color of the player will change

Try to close box by putting the last line to close to git points



The player with higer score will win

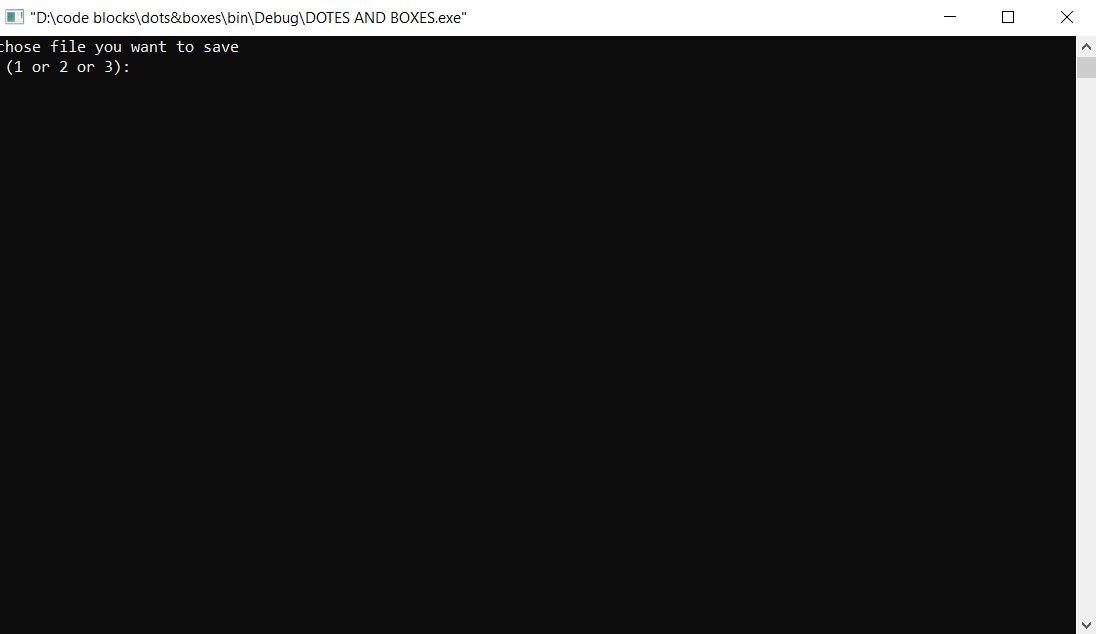


If you want to undo move while playing choose for rows 0 and for col 0

If you want to redo move while playing choose for row 1 and for col 1

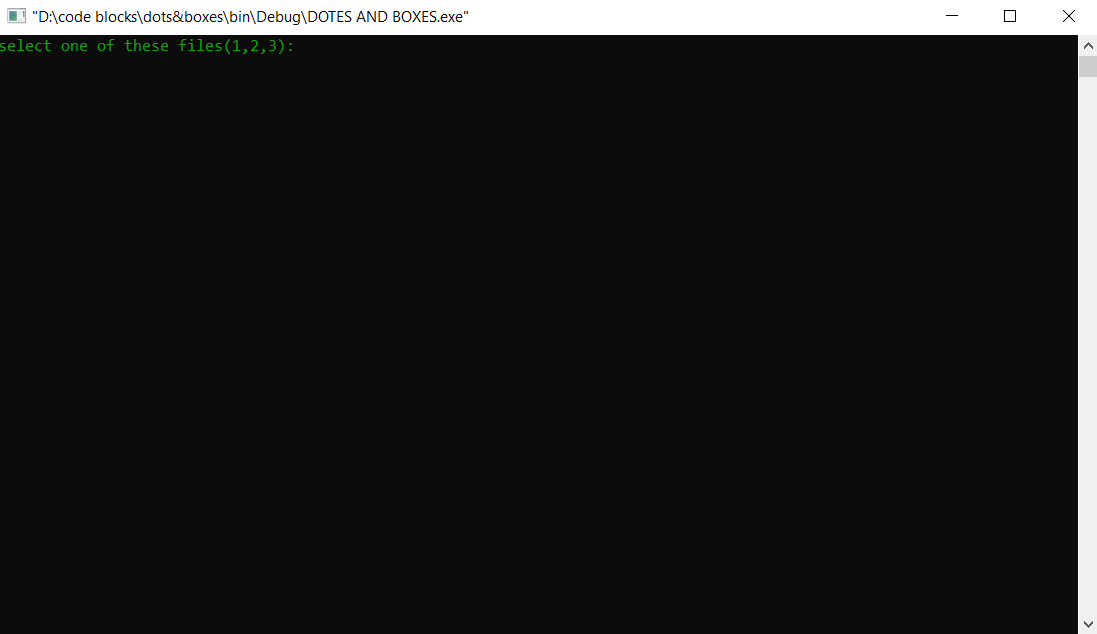
If you want to save the game while playing choose for row 2 and for col 2

If you want to return to main menu choose for row 3 and for col 3

If you want to save this window will appear

Choose the file you to save in (1 or 2 or 3)

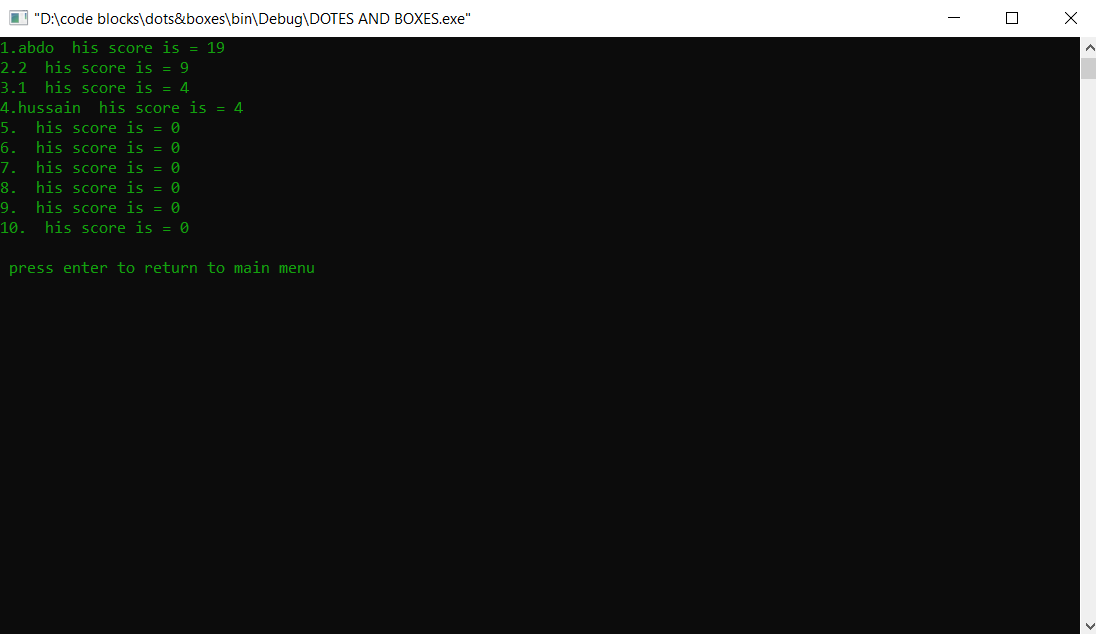
If you want to load the saved game



Choose (1 or 2 or 3 )

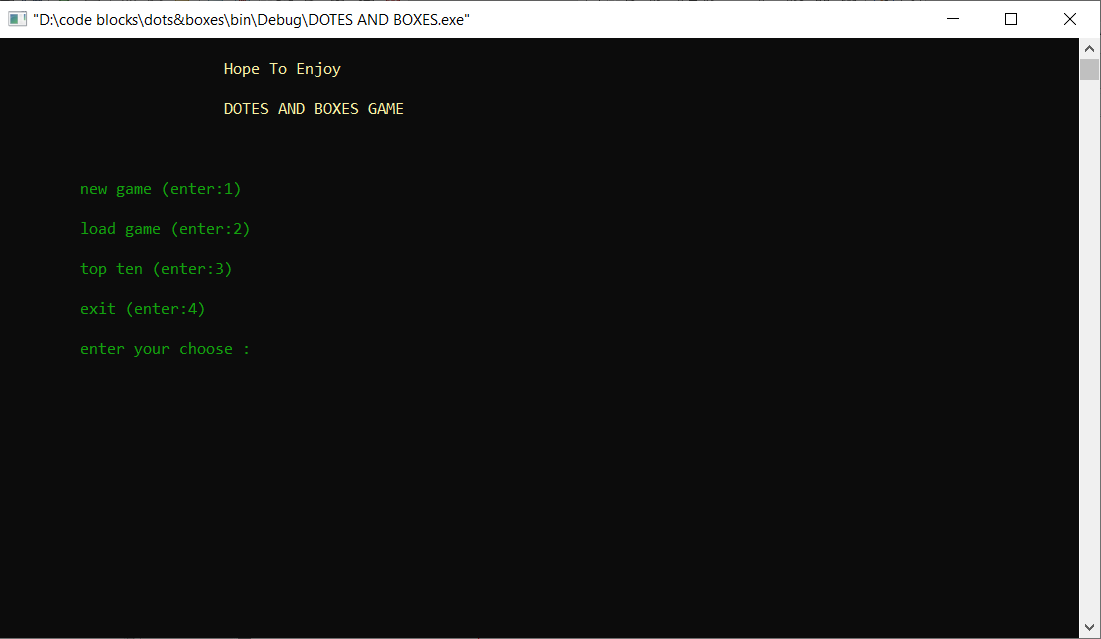
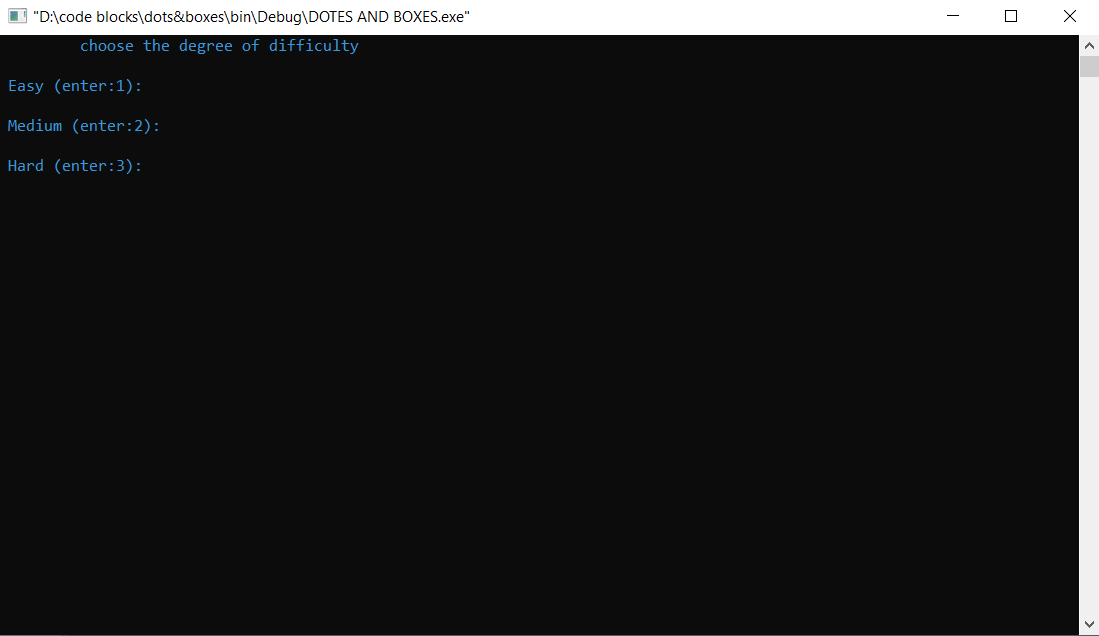
To get the saved file

If you want to see the top ten in the game in the main menu enter 3

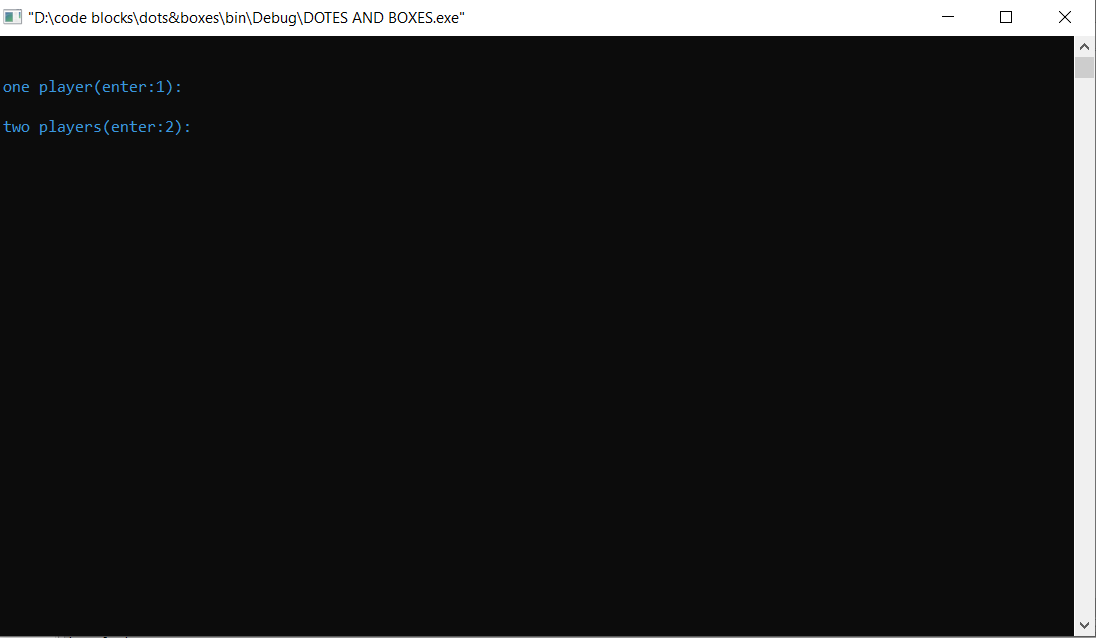


If you want to exit from the game enter 4 in the main menu or after finishing the playing either with computer or your friend

Sample runs:

 The first interface for the game   
game level

Mode of game



Playing with computer



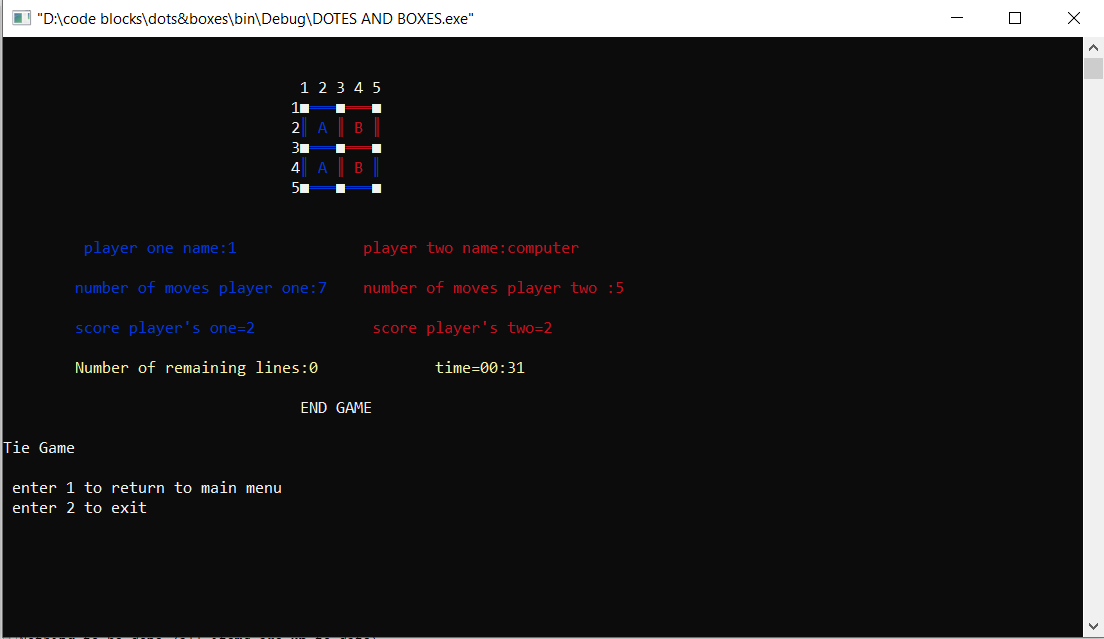
Winnig against computer



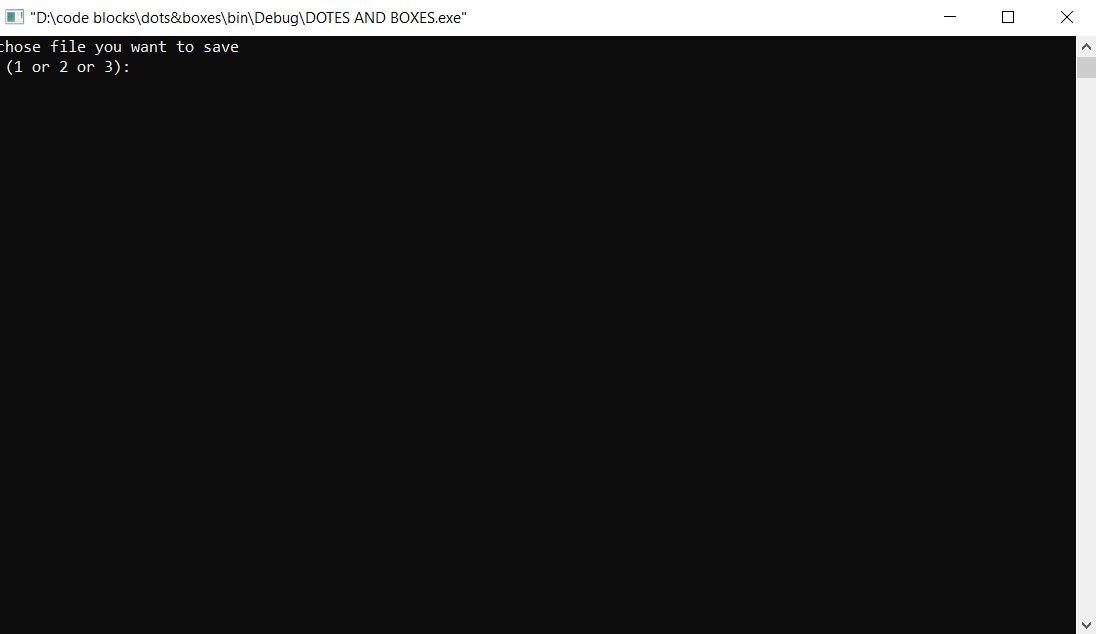
Loosing from computer

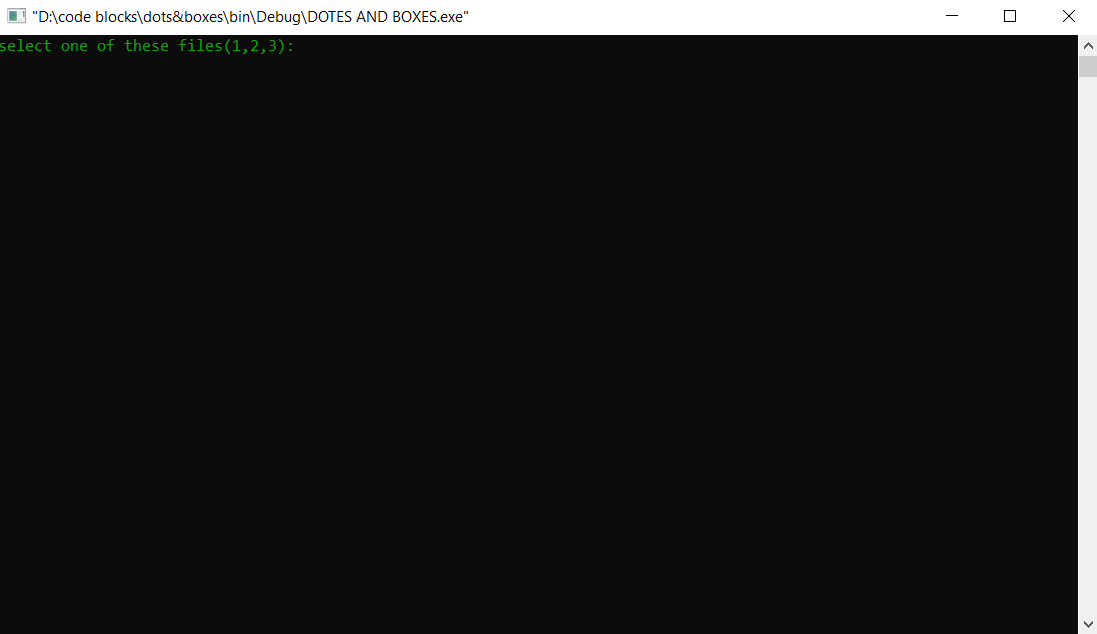


Tie with computer

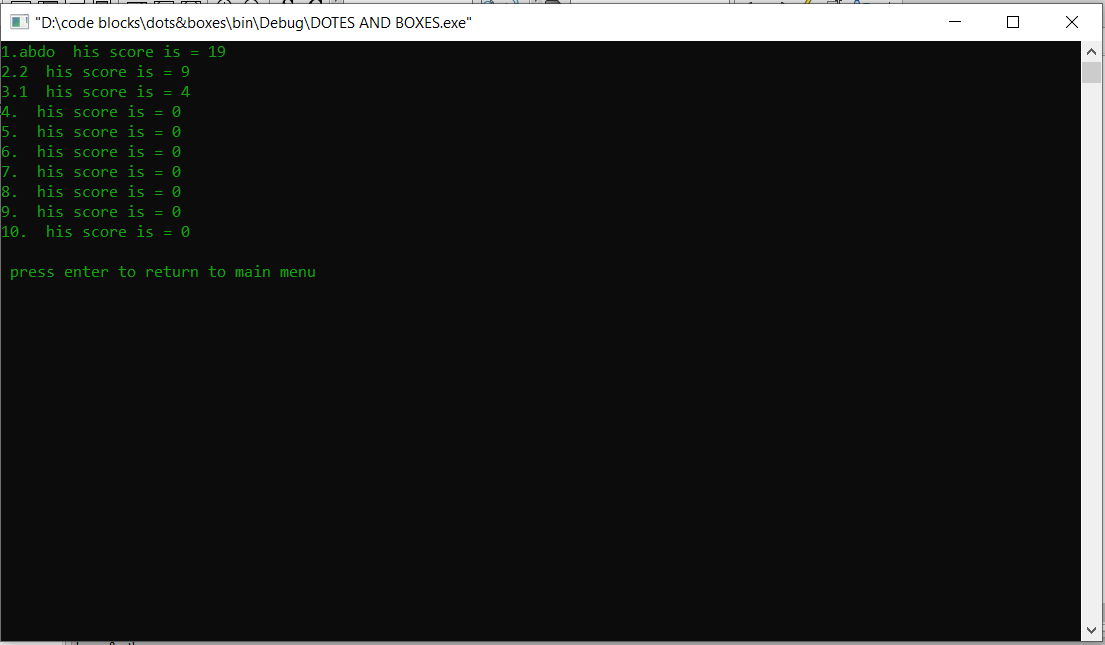


Saving



loading

top ten



References:

1)Stack over flow

2) from youtube <https://www.youtube.com/watch?v=vroe36JNhiU&list=PLkIliLHi5M4Lt2k8jILysU873YU6p_dui&index=17>

3) <https://www.educative.io/edpresso/how-to-use-the-sprintf-method-in-c>

Made by :

Hussain Mohamed and Abd Alrhman Magdy

we use git hub :

https://github.com/hussainmansour/DOTS\_and\_BOXES