

# BOX STARs



9/1/2022  
PROGRAMMING PROJECT

ABDULRAHMAN ALAA'ELDIN  
ABDULLAH TAMAN

## Dots and Boxes Reports contents

### Introduction

Description.....	p4
Assumptions.....	p4
References.....	p4

### User Manual

#### A. Menus

Main Menu.....	p4
Play Ten Menu.....	p5
Top Ten Menu.....	p5
New Game Menu.....	p5
1. Difficulty Menu.....	p6

#### B. Inside the game

1. Playing Moves.....	p7
Winning a Box.....	p7
Winning a Chain.....	p8
Winning a Game.....	p8
2. Significant Options in our Application.....	p9

### Features, Game Design & Architecture

#### A. Game design and Architecture

1. A quick glance on the building of the gamed.....	p10
---	-----

#### B. Features

1. Header files.....	p11
2. 3 Grids.....	p12

### Header files, Functions & Data Structures

#### A. Header Files and Included Functions

Table 3.1.....	p13
----------------	-----

#### B. Data Structure

Table 3.2.....	p15
----------------	-----

**Flow charts**

A. Game loop.....	p17
B. Main.....	p19
C. Play.....	p20
D. Start New Game.....	p20

**Debugging & Conclusive Description**

A. Debugging.....	p21
B. Description.....	p22

## Description

C program made by Abdul-Rahman Alaa and Abdullah Taman. The program is about the famous game “Dots & Boxes”. We have used the Code::Blocks IDE. The application consists of modules, each module has a specific function. We hope you could gently review the previous pages to have a deep understand of the project.

Hope you enjoy using our version of this game.

---

## Assumptions

We assumed that the time required is the time since the game started. So the counter of time doesn't stop until the game ends.

## References

The input check method was copied from a Youtube channel, However, this is the only part we copied a code, the code was written by us. You can see how organized it is.

## User Manual

The game “Dots & Box” is a game that can be played solo or with two competing player. It’s mainly about connecting two dots to form a line. Each line is counted as move. If a player manages to connect some dots to form a square, he earn one point, or a box. In addition to that, he gets an extra move.

The player with the biggest number of boxes is the winner.  
Let’s get into the details of our application.

### Main Menu

The place where the tale begins, the tale of honor and excite. A player has three options to select. The first is play menu, second is top 10 menu and the third for quitting the game, which is not a good choice, because this is a game that can make you happy.

If you choose to open the top ten list, you’ll be able to meet the most skillful players in the history of our applications. Seeing their names of each one of them and his score.

```
                Welcome to our project
                Dots & Boxes game

er : 1)
ist(enter : 2)
er : 3)

e number of your choice : 1
```

## Play Menu

It's the real start of the game. You can see that there are four options. First one takes you into a new game, second one loads an old game and third one takes you back to Main Menu.

```
New Game(enter : 1)
Load Game(enter : 2)
Back to main menu(enter : 3)
Enter the number of your choice : _
```

Choosing the load menu takes you directly to the play screen and we're showing in seconds.

## Top Ten Menu

```
Top 10 list
=====
1- Name : qq Score : 25
2- Name : abdo Score : 4
3- Name : rr Score : 4
4- Name : ee Score : 3
5- Name : Score : 0
6- Name : Score : 0
7- Name : Score : 0
8- Name : Score : 0
9- Name : Score : 0
10- Name : Score : 0

Press enter to back _
```

## New Game Menu

Here you can choose the mode you're willing to play in. Whether solo or vs a friend. Of course, you can return to the Main Menu. Choosing one player and two players both take you the Enter Name Menu. Of course, user enters one name if he's playing against computer. Then You'll be able to choose the size of grid you're wishing to play in, in the Game Difficulty Menu.

```
Choose mode : _
one player mode(enter : 1)
two players mode(enter : 2)

back(enter : 3)
back to main menu(enter : 4)
Enter the number of your choice : _
```

## Difficulty Menu

Here you can choose on choice from three choices, back to the New Game Menu and back to Main Menu.

```

Choose difficulty :-
Beginner mode(enter : 1)
Amateur mode(enter : 2)
Advanced mode(enter : 3)

Back(enter : 4)
Back to main menu(enter : 5)
Enter the number of your choice :

```

It allows you to choose one of the three grids prepared. The 6\*6 grid, 12\*12 grid and 18\*18 grid. First one is linked with **Beginner** choice, second is linked with **Amateur** choice and the third with **Advanced** choice, each represented with the input 1,2 or 3 respectively.

The shapes of these grids is in the **Features section**.

### Playing Moves

By inserting two numbers, one is odd the other is even, a move is played.

If user inserts an odd number then an even number, a horizontal line is drawn in the place you choose on the grid. If user inserts even then odd, a vertical line is drawn in the place

```

      1 2 3 4 5
1 ■   ■   ■
2
3 ■   ■   ■
4
5 ■   ■   ■

Player one name : Abdo           Player two name : Taman
Player one score : 0             Player two score : 0
Player one moves : 0             Player two moves : 0

No. of remaining lines : 12      Time 0 : 0

Save Game(enter : 1, 1)
Undo(enter : 2, 2)
Redo(enter : 3, 3)
Back to main menu(enter : 4, 4)

Player 1 turn

Enter row number : 

```

### Winning a Box

By completing a square, the player closes a box and earns a point.

```

      1 2 3 4 5
1  ■  ■  ■
2
3  ■  ■  ■
4
5  ■  ■  ■

Player one name : Abdo
Player one score : 0
Player one moves : 0

Player two name : Taman
Player two score : 0
Player two moves : 0

No. of ramaining lines : 12
Time 0 : 0

Save Game(enter : 1, 1)
Undo(enter : 2, 2)
Redo(enter : 3, 3)
Back to main menu(enter : 4, 4)

Player 1 turn

Enter row number : 

```

## Winning a Chain

When a player wins closes more than a box in row, he is said to win a chain.

```

      1 2 3 4 5
1  ■  ■  ■  ■
2  || B ||
3  ■  ■  ■  ■
4  || A || A ||
5  ■  ■  ■  ■

Player one name : Abdo
Player one score : 2
Player one moves : 5

Player two name : Taman
Player two score : 1
Player two moves : 5

No. of ramaining lines : 2
Time 2 : 30

Save Game(enter : 1, 1)
Undo(enter : 2, 2)
Redo(enter : 3, 3)
Back to main menu(enter : 4, 4)

Player 1 turn

Enter row number : 

```

## Winning a Game

By closing the highest number of boxes, a player is said to win the game.

In case of scoring the same points, there is no winner and the game ends, as follows.



```

      1 2 3 4 5
1  ■ — ■ — ■
2  || B || B ||
3  ■ — ■ — ■
4  || A || A ||
5  ■ — ■ — ■

Player one name : Abdo
Player one score : 2
Player one moves : 6

Player two name : Taman
Player two score : 2
Player two moves : 6

No. of ramaining lines : 0
Time 3 : 22

Save Game(enter : 1, 1)
Undo(enter : 2, 2)
Redo(enter : 3, 3)
Back to main menu(enter : 4, 4)

Game End

TIE (no winner)

Play again(enter : 1)

Back to main menu(enter : 2)

Exit(enter : 3)

```

### Significant Options in our Application

Our application undos and redos easily, whether you're playing against a computer or a friend, you can undo and redo as many times as you want and the program doesn't crash not glitches.

You can also as any screen back to main menu.

You can exit the whole game and end this tale.

This is everything. Hope you enjoy the game.

## Game Design and Architecture

Thanks to God, we made a very strong design, a design that helped us to write the undo and redo codes simply. My colleague designed it very well.

The game was mainly a sequence of actions that helps the user to open the game and play.

The main menu was easy to design, and we designed the program so that it called in any other function as we put return to main menu usually available.

The leaderboard was recorded in an ASCII file, and we read this file when the user opens it.

The "load game" option was depending also on a text file also.

By calling The "start a new game" a function we change the screen and open a screen to choose the mode, then insert the names and finally for choosing the grid, so it was like a backbone in our program and takes us directly to the game loop.

The game loop was a while loop in the "GameLoop" function which we put in a certain source file just like the previous important functions. This function receives all the information given by the user to display and analyze. For printing the screen with every move we wrote a function that prints the game, and after printing it returns again to the game loop function. Then the user plays and everything is fine. At the end of the loop, there is "BoxCheck" function that checks if a player closed a box in the previous move, if he did this function gives an indicator to the game loop to let this player who closed a box to play an extra move. It loops.

After all moves are done, the game ends and we take the user to the place he wants. Whether to play a new game or return to main menu or whatever he wants.

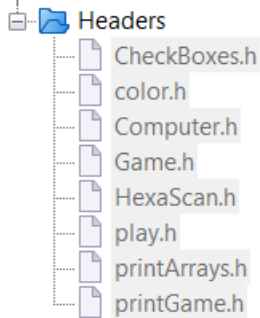
# Features

- Splitting our project on different files and use header files

We have split our project into Header files. The header files are:

CheckBoxes.h, color.h, Game.h, HexaScan.h, play.h, printArrays.h and printGame.jh.

Usage of each function was mentioned before.



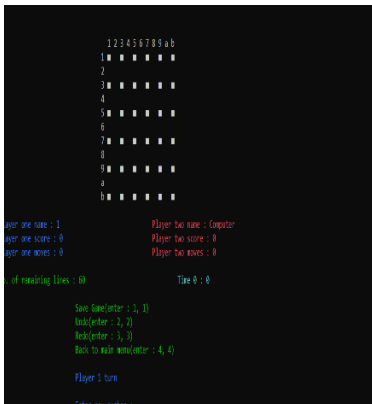
- Giving the user the option to user three Grids not just 2

We have given the user the option to choose between three grids. These grids are Beginner, Amateur and Advanced.

## BEGINNER



## AMATUER



## ADVANCED

```

      1 2 3 4 5 6 7 8 9 a b c d e f g h
1 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
2 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
3 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
4 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
5 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
6 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
7 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
8 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
9 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
a ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
b ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
c ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
d ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
e ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
f ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
g ■ ■ ■ ■ ■ ■ ■ ■ ■ ■
h ■ ■ ■ ■ ■ ■ ■ ■ ■ ■

Player one name : 3                      Player two name : 2
Player one score : 0                    Player two score : 0
Player one moves : 0                    Player two moves : 0

No. of remaining lines : 144             Time 0 : 0

Save Game(enter : 1, 1)
Undo(enter : 2, 2)
Redo(enter : 3, 3)
Back to main menu(enter : 4, 4)

Player 1 turn

Enter row number : ■

```

This last one was tricky, but we managed to handle it. It's known that the hexadecimal system has only 16 digit. In the grid there were eighteen option. We reached a formula that could solve this in Hexa scan function. Here it is.. With the help of this algorithm we managed to transform the scanned numbers to readable values for the array, like 1,2,3,4...18.

```

int HexaScan()
{
    char scan[100];
    int decimal;
    fflush(stdin);
    gets(scan);
    if(scan[0] > 48 && scan[0] < 58)
    {
        decimal = scan[0]-48;
    }else if(scan[0] > 96 && scan[0] < 105){
        decimal = scan[0]-87;
    }else{
        decimal = -1;
    }
    return decimal;
}

```

## Header Files and Functions in them

Header file	Included functions	Importance of functions
CheckBoxes.h	checkbox	It checks whether the player has won w abos and gives him an extra move.
Color.h	Red, blue, reset, black, purple, green, cyan, yellow	Each function give the screen a specific color.
Computer.h	Computer "void"	Plays for the computer during solo mode
Game.h	GameLoop "void"	The most important function in the game, prints the grid and modify it easily.
HexaScan.h	Hexascan "int"	Returns the numeric value of the inputs of the user. If he inserts a wrong input, it returns -1 so that he inserts right input
Play.h	play	<p><b>-Gives the game loop the all details of the grid and the game.</b></p> <p><b>-Receive the name of the user, opens loads a game.</b></p>

`printGame.h`      `Gameprint`      prints the grid with all of it details

---

## Data Structure

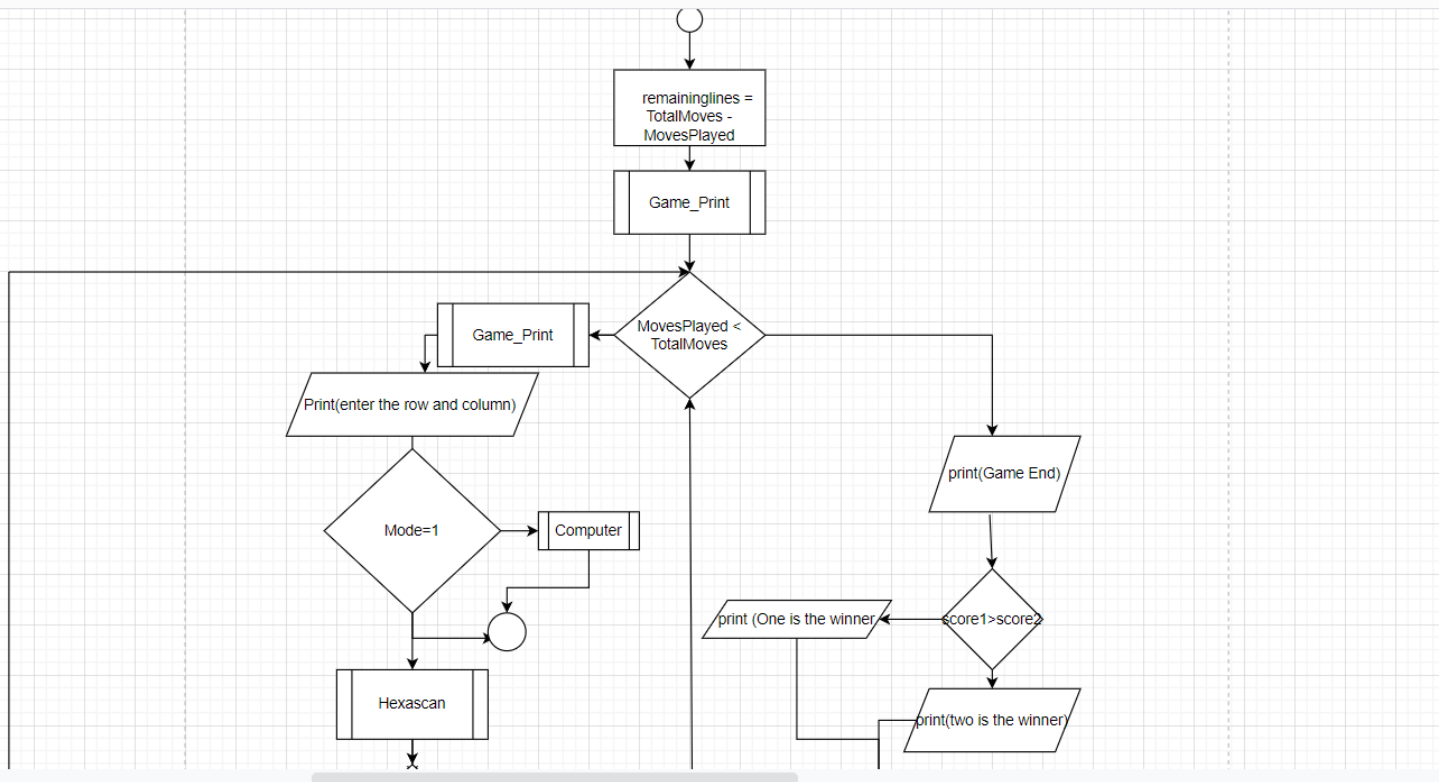
Name	Type	Elements	Usage
name1	Character Array	Name of first user	Helps us to save and reload the first player's name
name2	Character Array	Name of first user	Helps us to save and reload the second player's name
Grid	Character Array	Characters that form and shapes the grid with all it's elements, it changes during the game	The grid itself, the most important array in the game
GameDetails	Integer Array	For each play: the row,column,player number,box horizontal coordinate, box vertical coordinate, second box horizontal coordinate "if the player closed two bixes", its vertical coordinate.	Significant in the undo and redo.
scan	CharacterArray	User inputs in the game before transforming them to numbers.	Significant for recievg the inputs of the user and transforming them into numbers, so that the could be printed.
Player	Structure of integers and characters.	Player one name and score and total moves.	Save these information, then recalling them when we need to save these information in another variable that would be printed int the game.
Top	Structure of integers and characters.	Player one name and score and total moves	Save these information, then recalling them when we need to save these information in another variable that would be printed int the game.

top	Array of structures	Structures of type Top	Used for saving and reloading the top ten players
-----	---------------------	------------------------	---

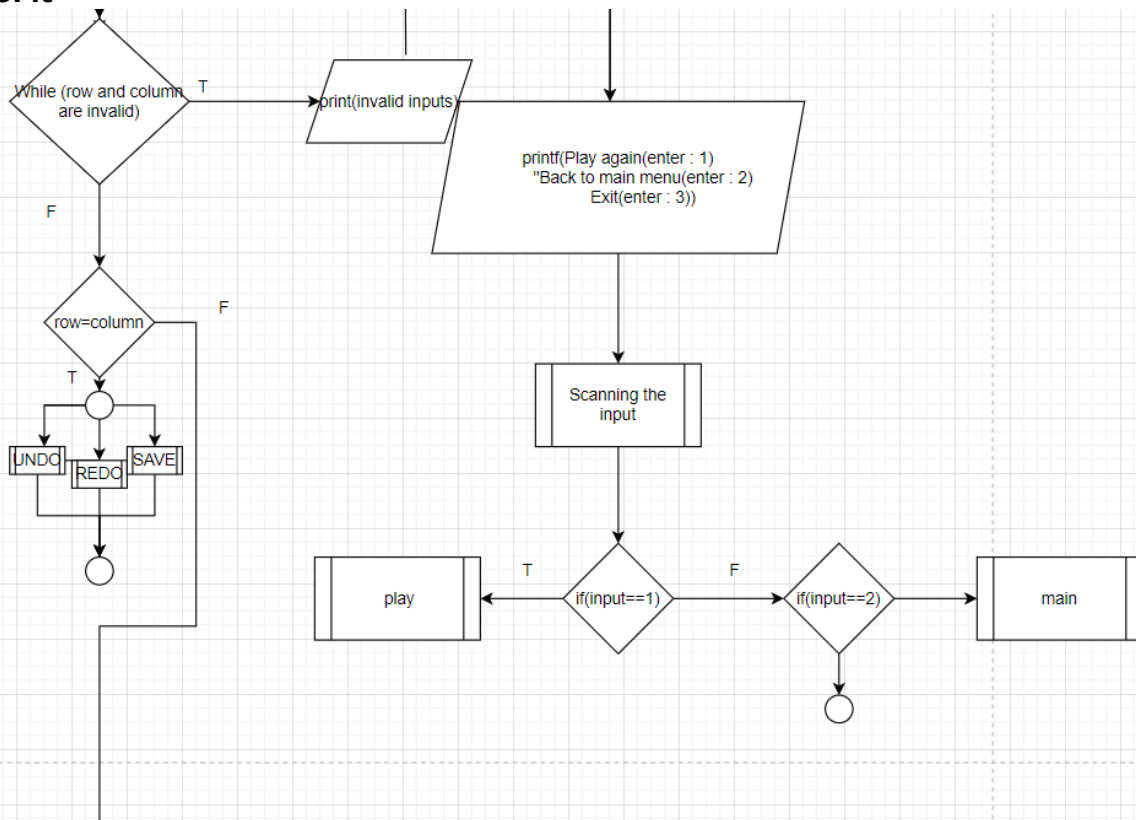


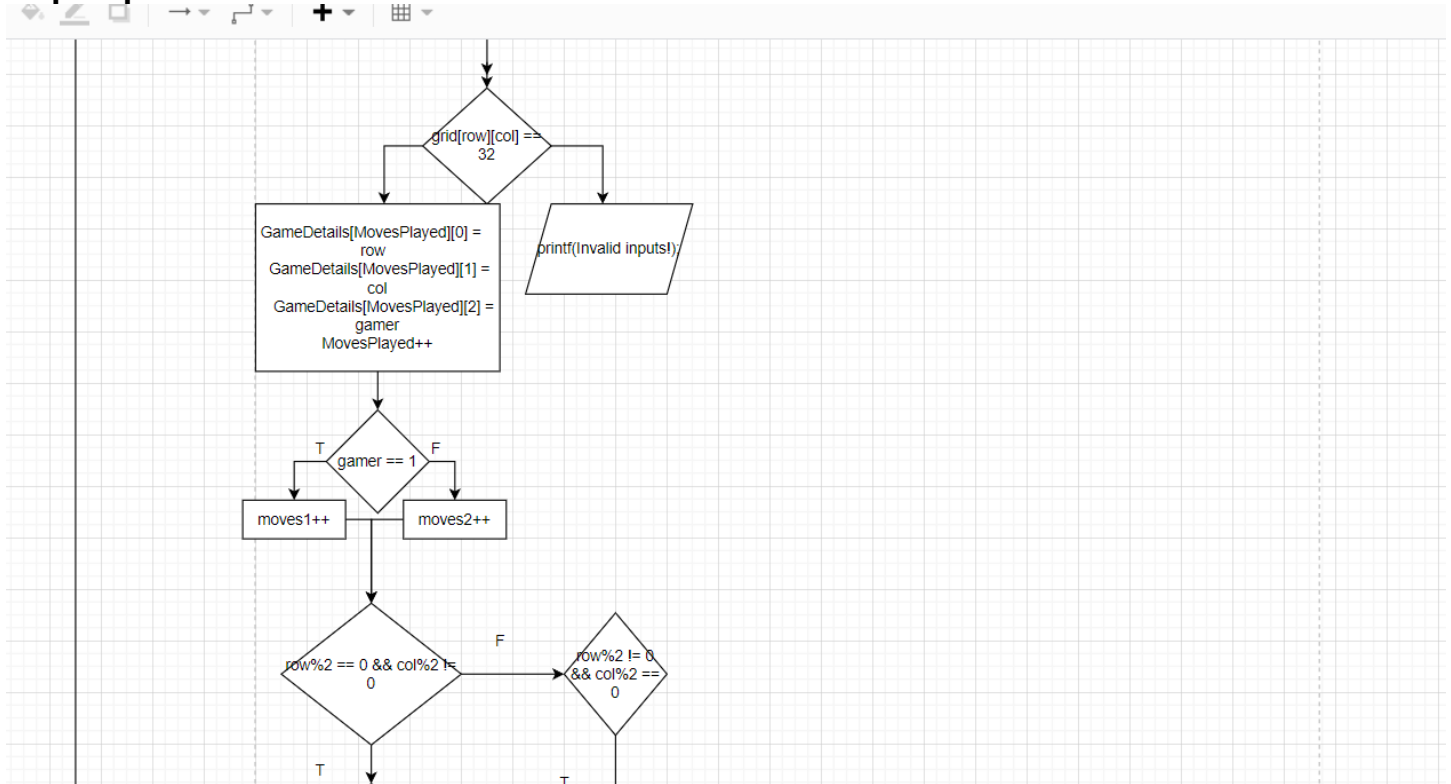
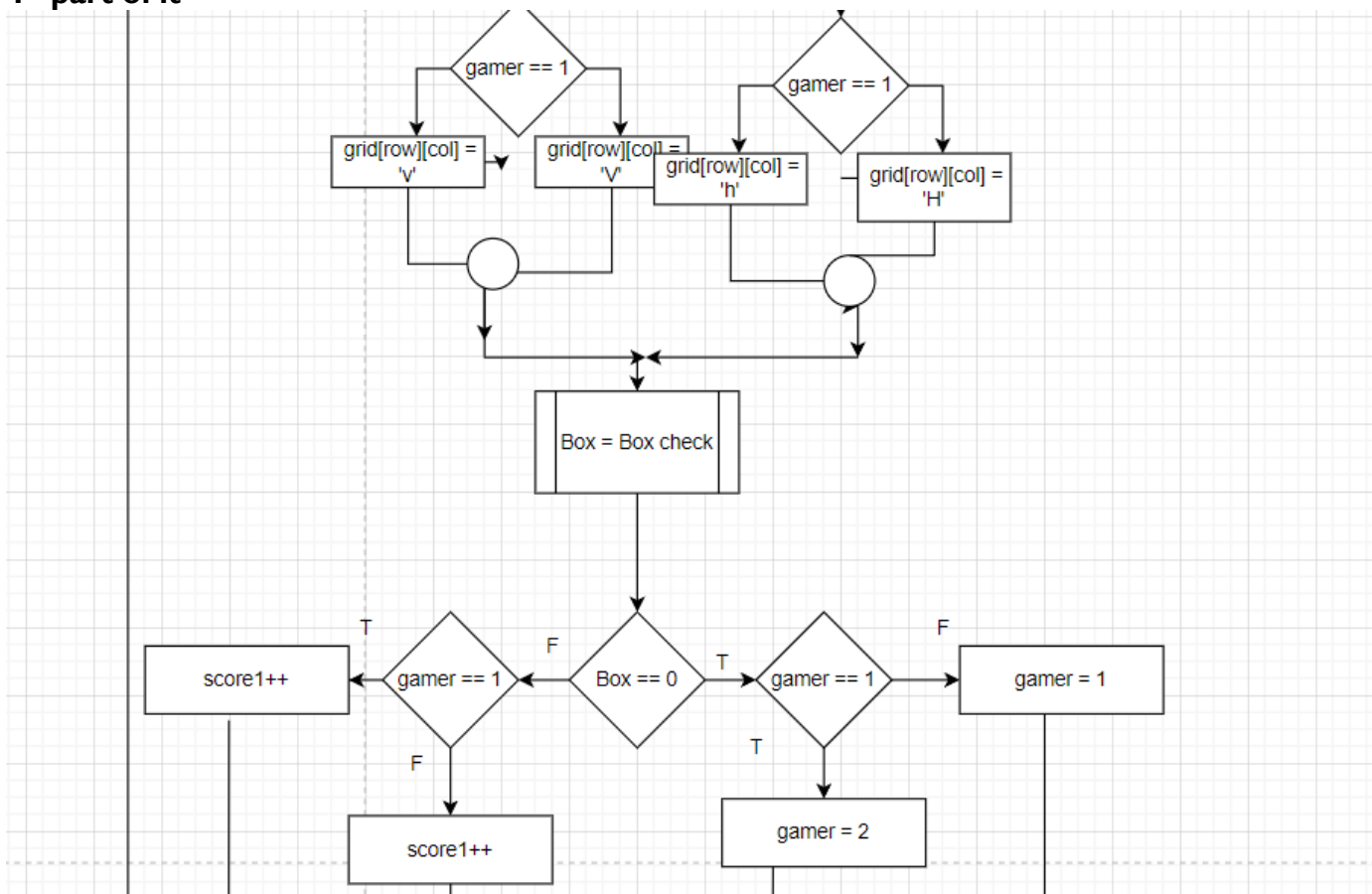
# Flow Charts

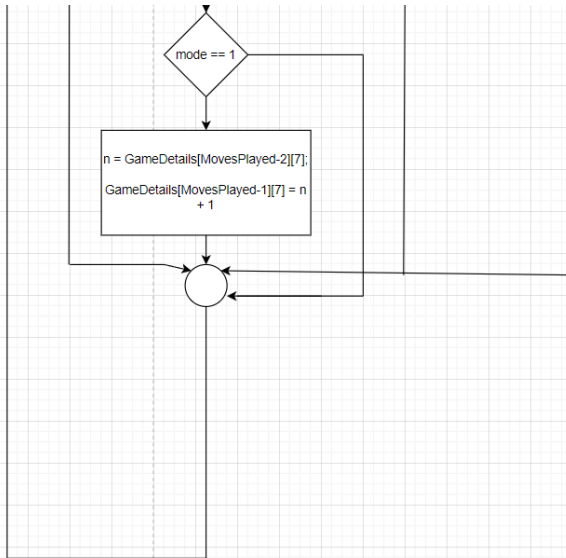
## I-GAME LOOP



## 2nd part of it

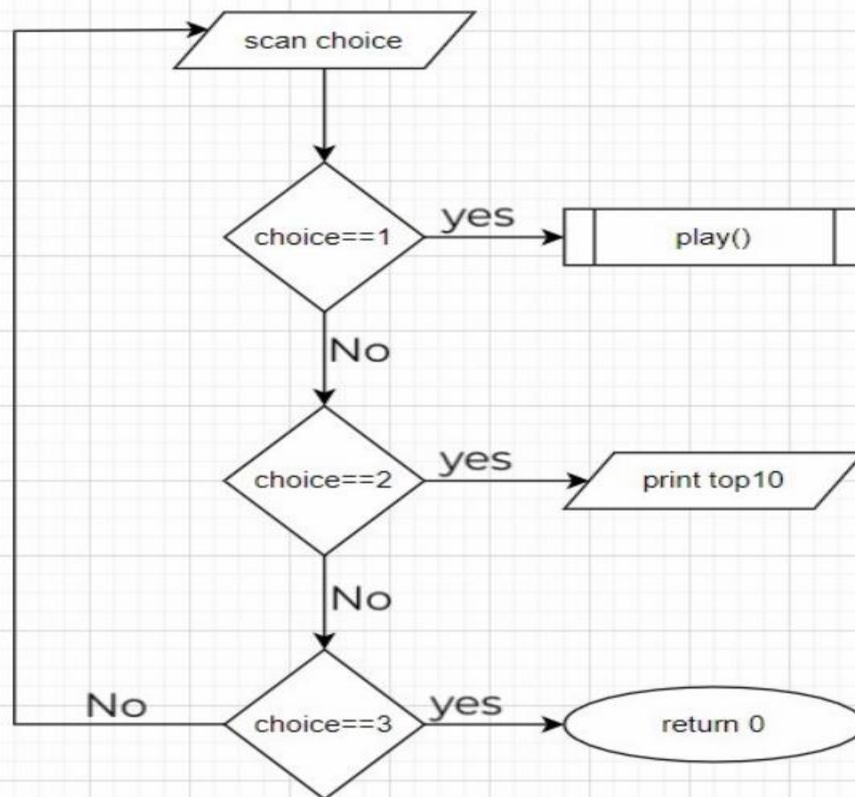


3<sup>rd</sup> part pf it4<sup>th</sup> part of it

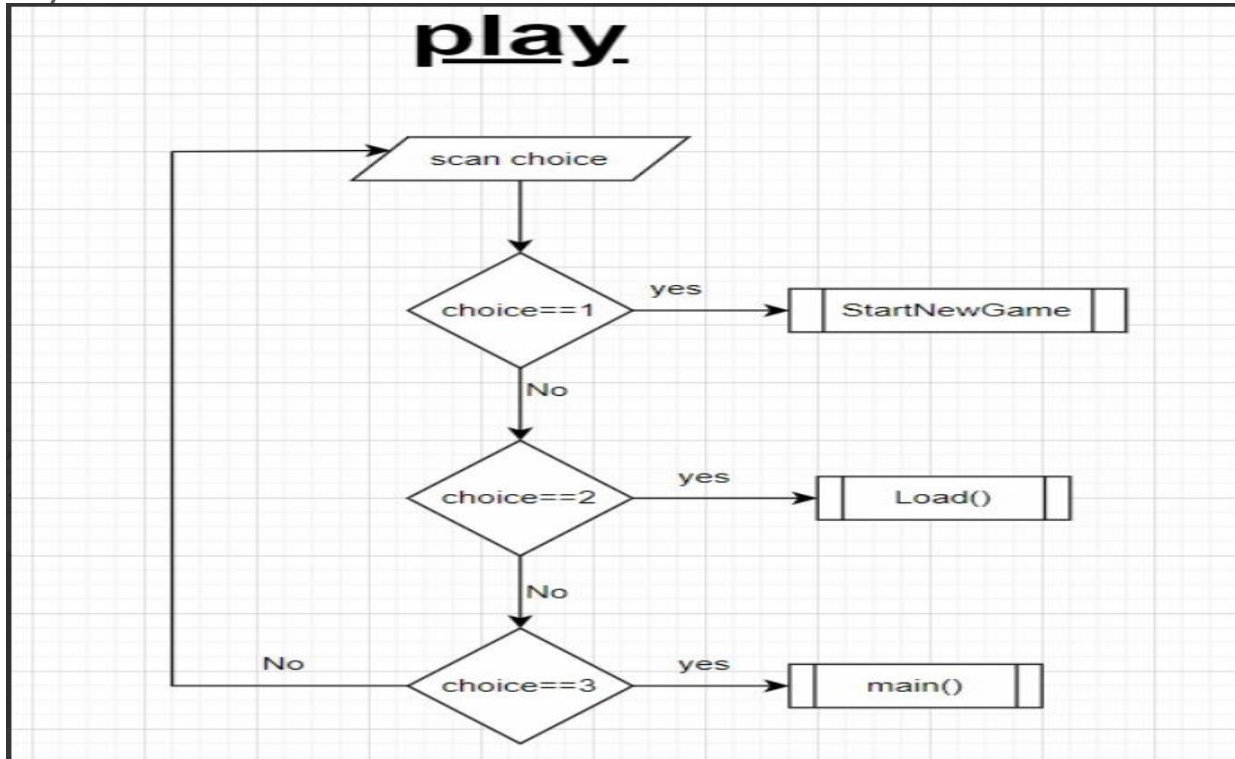
5<sup>th</sup> part of it

Main

## Mian

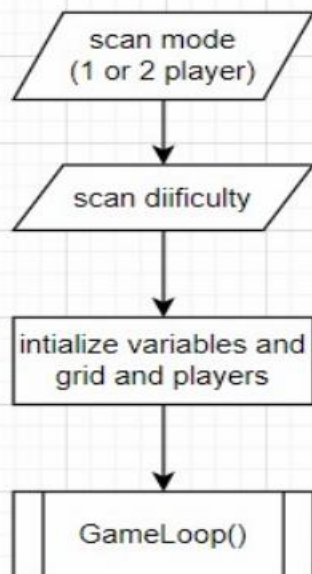


Play



Start new game

## StartNewGame



## Debugging

We used game details array for this, it was helpful. We printed the values of the move we wanted sometimes to make sure of our inputs.

```
//=====
//for debugging we print GameDetails after every move
/*printf("\n\n");
for(int i = 0; i < TotalMoves; i++)
{
    for(int j = 0; j < 8; j++)
    {
        printf("%d ", GameDetails[i][j]);
    }
    

print, prints, print f

printf("\n");
}*/
//=====
```

## Description

C program made by Abdul-Rahman Alaa and Abdullah Taman. The program is about the famous game “Dots & Boxes”. We have used the Code::Blocks IDE. The application consists of modules, each module has a specific function. We hope you could gently review the previous pages to have a deep understand of the project.

Hope you enjoy using our version of this game.