BACHELOR OF TECHNOLOGY

COLLEGE OF ENGINEERING PUNE

2 OCTOBER 2022

SUBMITTED BY : Pavankumar Mhaske

NUMBER SHIFTING GAME (COLSOLE APPLICATION)

USING C LANGUAGE

Under the Guidance of

Mr . Saurabh Shukla Sir (MySirG)

MySirG.com

iNeuron

INTRODUCTION

One of the Most Important Factor in the human beings is the entertainment . Games are the one of the major section of the entertainment specially children's have extreem craze towords the computer games . So it is good thing to play games which makes entertainment as well as develops some logical thinking .

This program is written in C language and hence it's very fast to execute . This program can run on any device which have windows platform . The project design and ensuring the program is modular in nature made the project more easy to modify and achieved all the benefits of the procedure oriented programming . There are very limited inbuilt functionalities provided by the C language but this program uses it's own advance functionalities internally by making the use of the basic functionalities of the C .

This project report includes full user manual as well as the whole Driver code that was written .

PROBLEM DEFINITION

Now a days childerens and students are spending too much of the time in playing computer games . Currently available games are capable of making full of the entertainment but it will becomes the cherry on the top if they get additional benefits of logical thinking developments in parallel to the entertainment .

OUTLINE OF THE SOLUTION

The purpose of the program is entertainment with logical thinking skills development . hence it s acceptable and fruitful to spent time littel bit of extra time in playing games . Solving the puzzle sometime becomes boarding task , but in this program we club the puzzel solving techniques with the games and due to computer as an intermediate device the combination of the three the logical thinking , entertainment and computer device made the program much more valuable

DETAILED DESIGN

1) Program Structure

The program is devided into 14 files .
The main.c file contains the driver code inside the main() function . Other functions are created in separate files to extend the functionality and keep the main() function clean (to achieve the molularization)
It is good practice to include this files in order they going to used so that if function in one included files need functions in another included file it will be taken care by maintaing the order of includation .

The order of the files includation is :

```
#include "Fflush.c"
#include "Welcome.c"
#include "Rules.c"
#include "Name.c"
#include "Challenge.c"
#include "Mode.c"
#include "TopBoard.c"
#include "Matrix.c"
#include "Invalid_Move.c"
#include "Escape.c"
#include "Result.c"
#include "MenuCard.c"
#include "Shifting.c"
```

The program uses some advanced functionalities which are not avialable in the c language . This advance functionalities are created by making use of the c language internally . By considering the user experience and user interface as primary objects extra lines of the code get added and this code is going on repeating again and again so to overcome this issue program is divided into many files and each file contains bunch of relevent functions .

2) Functions and their Explanation

1 . Fflush.c file contains the Fflush( ) function

prototype : char Fflush(char)

During the program development it is observed that accepting the arrow keys as an input is far away from the normal character input so each time it is creating an problamatic situation for the accepting the next character input . We have tried to solve this problam by using the C inbuilt functionality "fflush(stdin)" but this that functionality is aslo not able to clean the input buffer properly .

Here to solve this problem I Created the new functionality to clear the the buffer completly immediately just after the user input . The input can be of any type no matter the input entered by user is normal character (including all inputs of length one) or arrow keys input .

So I strictly followed one technique (Rule) in the entire program

" Insted of using getch() for the character input , Use Fflush(getch()) "

How Fflush(getch()) works ?

There are two scenarios :

1) User input is normal character

Fflush( ) functions identifies this is normal character input and just returns the same input

2) User input is arrow key

On the input of arrow key input buffer receives two things

i) First is the a character whose ASCII code on console window is -32 (But you may wonder this same character shows it's ASCII code as 0 on the VS Code terminal window)

ii) Second is actual responce whose ASCII codes are

72 = UP
80 DOWN
75 LEFT
77 RIGHT

Fflush( ) functions identifies an arrow key input and ignore the first scroll input character then returns second character which is actual responce

2 . Welcome.c file contains the welcome( ) function

prototype : void welcome(void)

The welcome() function cantains the too much of the lines of the code hence consumes so many line in the program so it can make our main() function bulky that's why it is good practice to make a saparate functions for welcome screen interface .

The number of lines of the code in the welcome() functions are incresed by the resone of the provoiding the good user experice and colourful user interface .

3 . Rules.c file contains the displayRules( ) function

prototype : void displayRules(void)

The displayRules() function is the second window screen . When the displayRules() function first clear the window and then accept input to to contiue the game . After the input it display the rules the game . Again accept input to continue game (this input is confirmation that user has read all the Rules of the game) .

The number of lines of the code in the welcome() functions are incresed by the resone of the provoiding the good user experice and colourful user interface .

4 . Name.c file contains the getName( ) function

prototype : char* getName(void)

First getName() function first clear the window . The getName() accepts the character array input of maximum length 16 bytes (16 characters) and returns the the address of the memory locations . Here getName() uses the concept of the dynamic memory allocation(DMA) using calloc( ) function .

5 . Challenge.c file contains the challenge( ) function

prototype : int challenge(char*)

First challenge() function first clear the window . The getName() accepts argument as the base address of the character array created in the main() function and returns the integer value . The returned interger value is the number of the challenges user got in the game .

User can select any one one challenge from the given two challenges .

1) Custume challenge (user can change the maximum number of moves)

2) Auto challenge (user will get the predecided number of move)

6 . Mode.c file contains the mode( ) function

prototype : int** challenge(char*)

First made() function first clear the window . The challenge() accepts argument as the base address of the character array created in the main() function and returns the Base address of the 2D integer Array .

Here mode() uses the concept of the dynamic memory allocation(DMA) using calloc( ) function .

User can select any one one challenge from the given two challenges .

1) Randome Mode (For Each Level : Random Orientation of elements)

2) Default Mode (For Each Level : Fixed Orientation of elements)

7 . TopBoard.c file contains the displayNameChance( ) function

prototype : void displayNameChance(char name[] , int chance)

First displayNameChance() function first clear the window . The displayNameChance() accepts two arguments . First is the base address of the character array created in the main() function , second is the int value as an input and returns the Base address of the 2D integer Array

.

Here mode() uses the concept of the dynamic memory allocation(DMA) using calloc( ) function .
User can select any one one challenge from the given two challenges .
1) Randome Mode (For Each Level : Random Orientation of elements)
2) Default Mode (For Each Level : Fixed Orientation of elements)

8 . Matrix.c file contains the matrix( ) function

prototype : void matrix(int **)

The matrix() accepts 2D Array as an argument and then print the matrix .

9 . Invalid_Move.c file contains the invalid( ) function

prototype : void invalid(void)

The invalid() just prints the invalid move interface .

10 . Escape.c file contains the escape( ) function

prototype : char escape(void);

The escape() just prints the interface of escape and return the character .

11 . Result.c file contains the win( ) and lose() functions

prototypes : int win(int **A,char* , int);
void lose(void);

The win() accepts three arguments
i) Base address of the 2D Array ,
ii) Base address of the
iii) integer value
and returns the integer value which creates the condition to terminate the current level of game .

The lose() just print the lose interface .

12 . MenuCard.c file contains the menuCard( ) function

prototype : char* menuCard(void)

The menuCard() dispalys the menu to exit or continue the game . It accepts the character input which is deciding the game to be continued or exit and returns the string value (Characher array)

13 . Shifting.c file contains the shiftUp( ) , shiftDown( ) , shiftRight( ) , shiftLeft( ) functions

prototypes :
```
void shiftUp(int **int *row,int *col)
void shiftDown(int **B,int *row,int *col)
void shiftRight(int **B,int *row,int *col)
void shiftLeft(int **B,int *row,int *col);
```

All the four functions accepts three arguments
i) Base address of the 2D Array
ii) Address of the int row variable
iii) Address of the int col variable

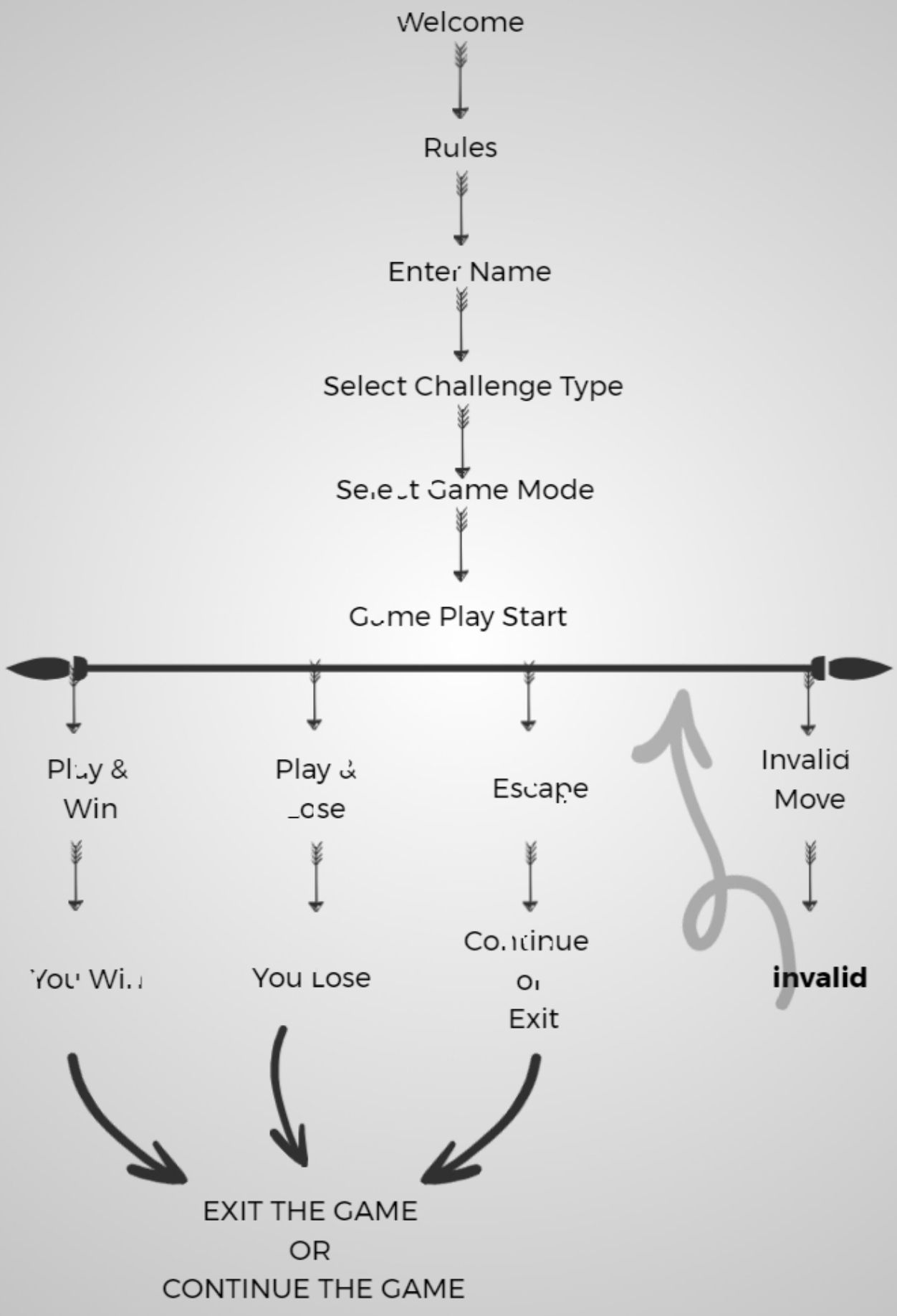
i) shiftUp() = Move element Up which is at the down of the empty position and hence empty block to the Bottom . If empty position is it'self at the Bottom then the element which is at the top in the same col will be shifted to the Bottom and hance empty position takes it's position .

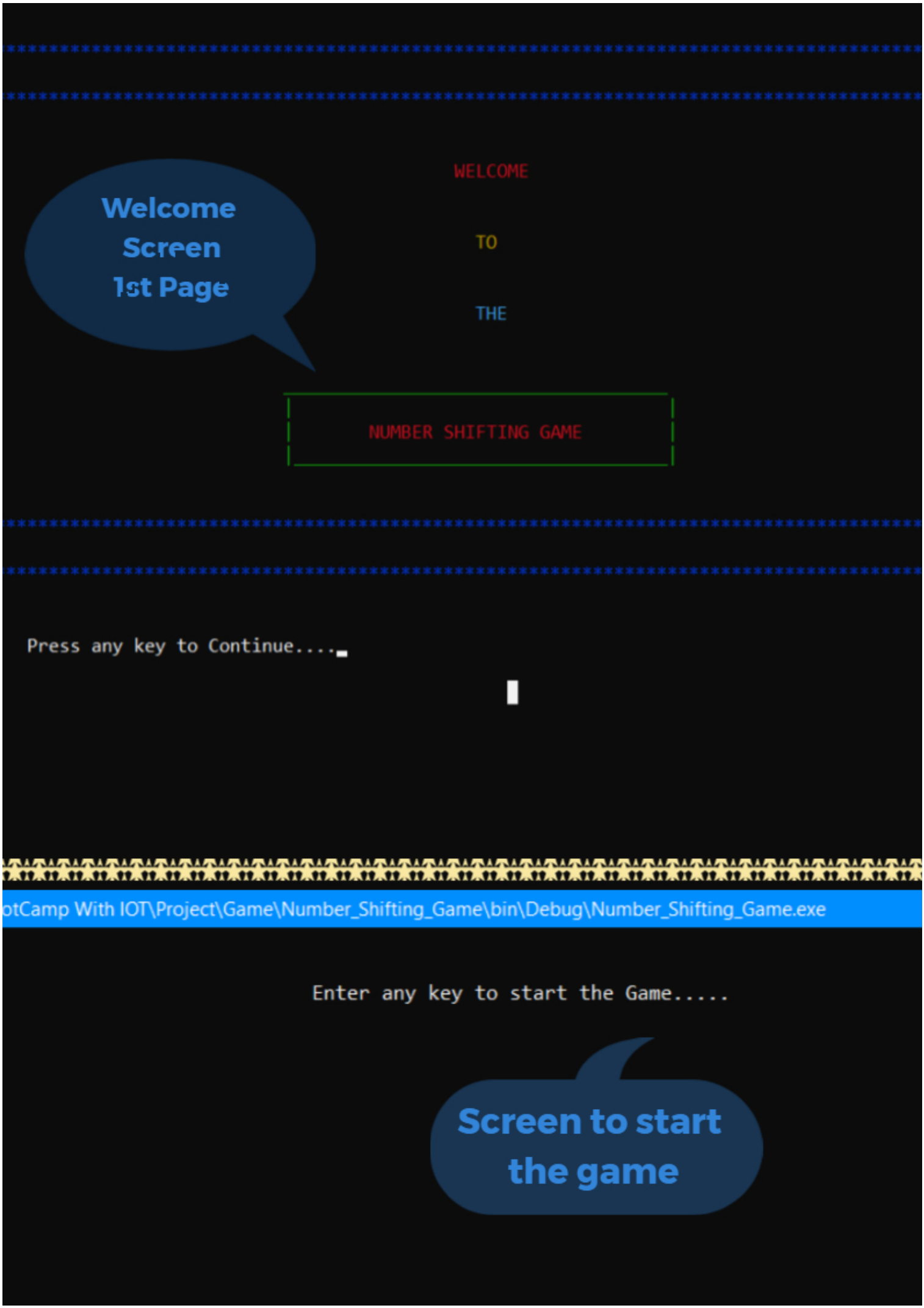
ii) shiftDown() = Move element Down which is on the head of the empty position and hence empty block to the Up . If empty position is it'self at the Top then the element which is at the bottom in the same col will be shifted to the Top and hance empty position takes it's position .

iii) shiftLeft() = Move element Left which is at the right of the empty position and hence empty block to the Right . If empty position is it'self at the RightMost in the row then the element which is at the LeftMost in the same row will be shifted to the RightMost position and hance empty position takes it's position .

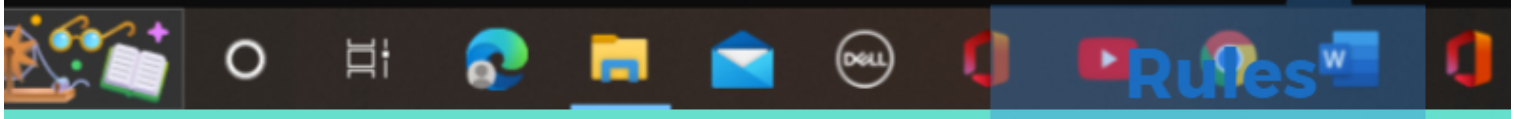
iv) shiftRight() = Move element Right which is at the left of the empty position and hence empty block to the Left . If empty position is it'self at the LeftMost in the row then the element which is at the RightMost in the same row will be shifted to the LeftMost position and hance empty position takes it's position .

TESTING , VALIDATION & PROGRAM'S FLOW

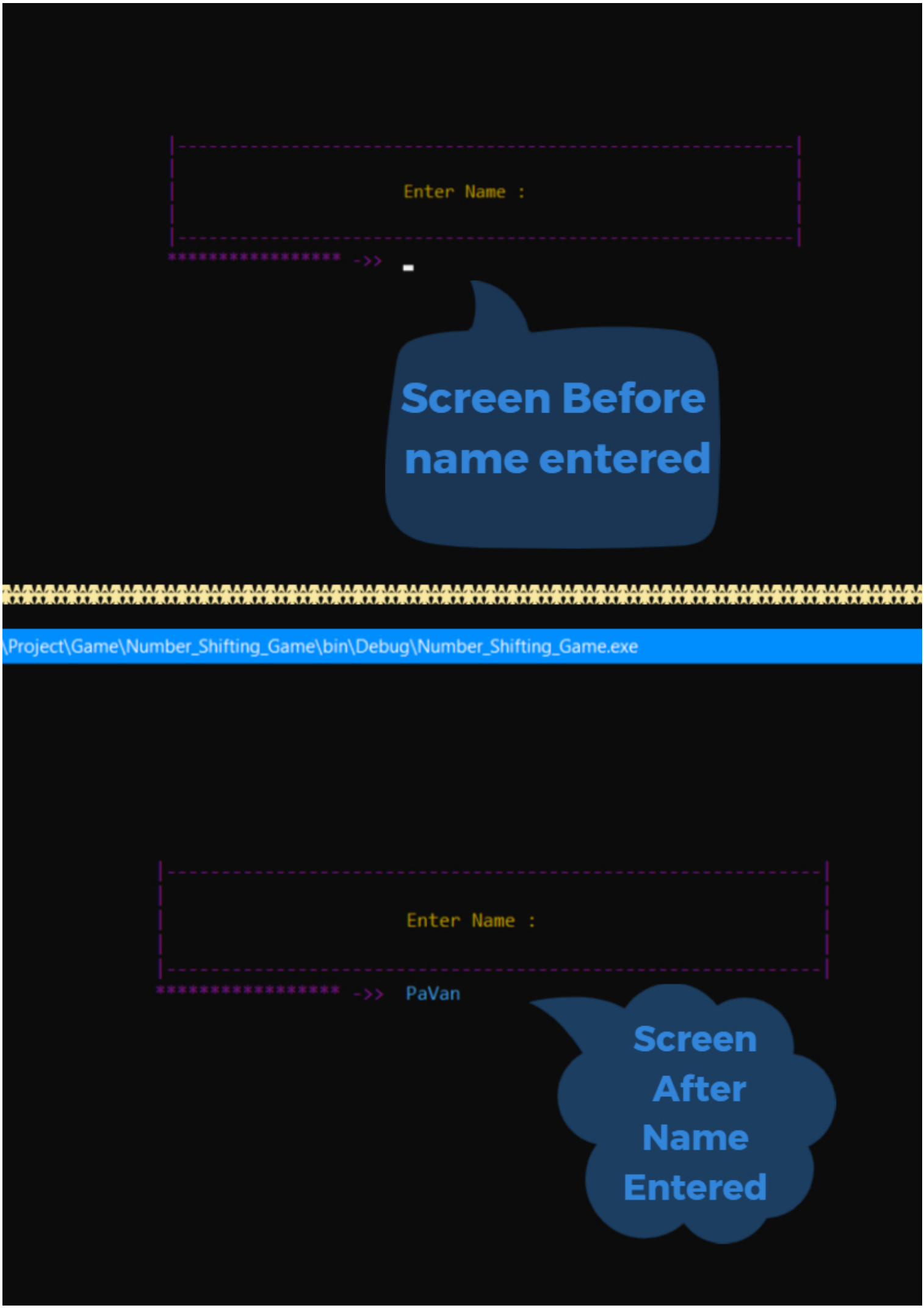


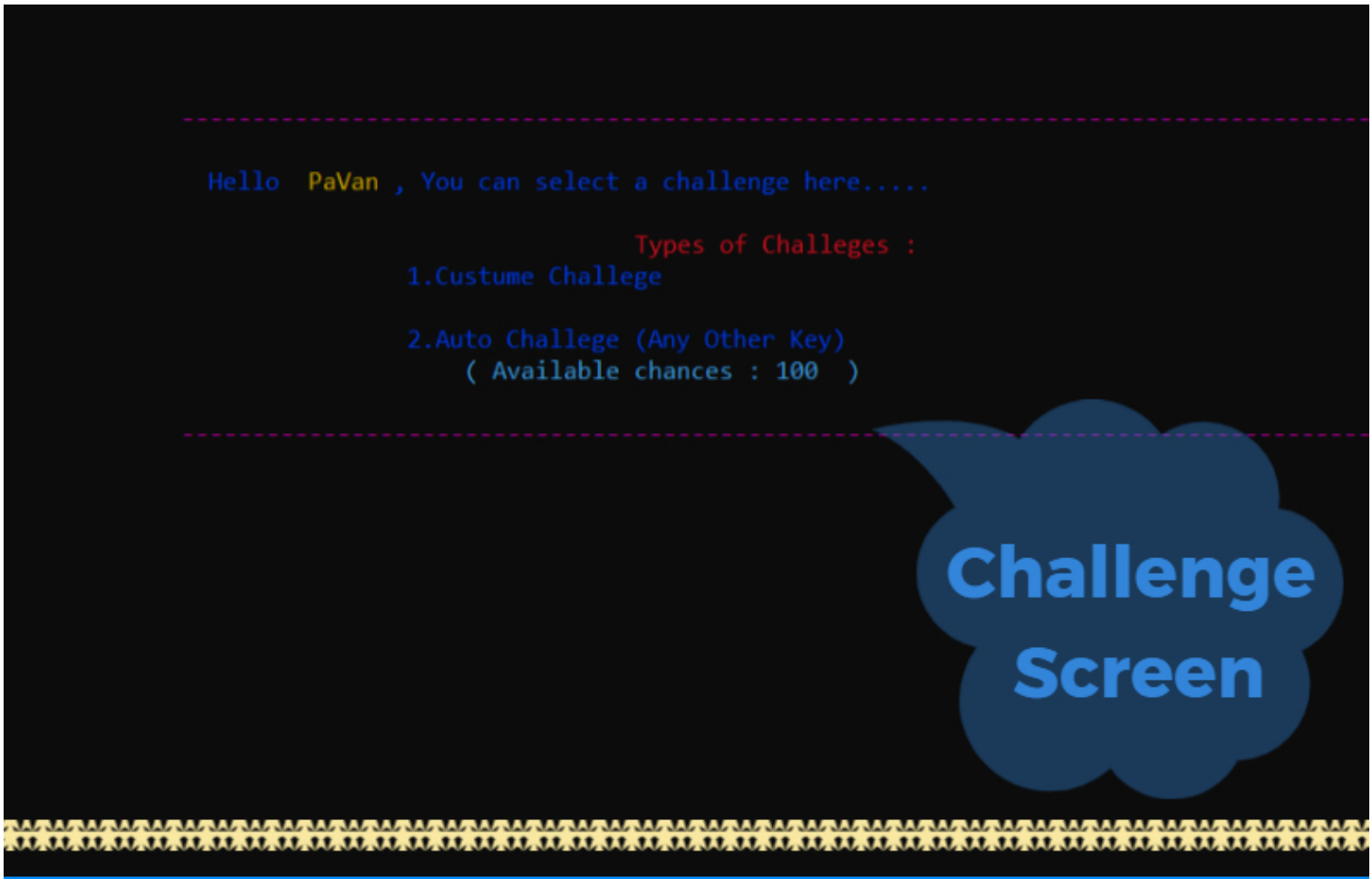

Welcome
Screen
1st Page
WELCOME
TO
THE
NUMBER SHIFTING GAME
Press any key to Continue....
otCamp With IOT\Project\Game\Number_Shifting_Game\bin\Debug\Number_Shifting_Game.exe
Enter any key to start the Game.....
Screen to start
the game

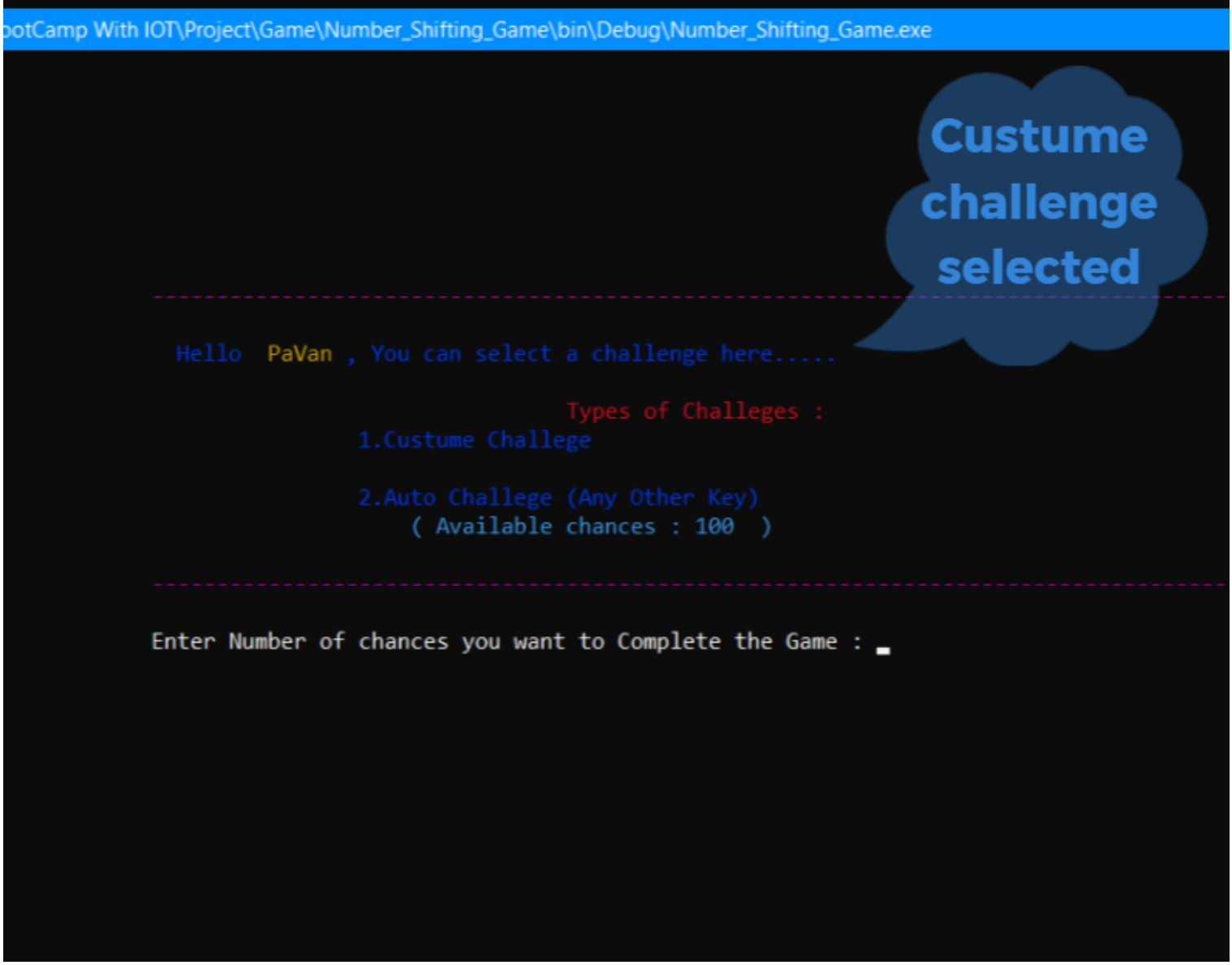
```
                    NUMBER SHIFTING GAME

********************************************************************************
********************************************************************************

                          RULES OF THIS GAME

  1.You can move only 1 step at a time by arrow key
                  Move UP     : by UP Arrow Key
                  Move DOWN   : by DOWN Arrow Key
                  Move LEFT   : by LEFT Arrow Key
                  Move RIGHT  : by RIGHT Arrow Key

  2.You can move number at empty position only

  3.For each valid move : Your total number of move will decreased by 1

  4.Wining situations : Number in a 4*4 matrix should be in order from 1 to 15
                                Winning Situation :
                  --------------------------
                  |  1  |  2  |  3  |  4  |
                  |  5  |  6  |  7  |  8  |
                  |  9  |  10 |  11 |  12 |
                  |  13 |  14 |  15 |     |
                  --------------------------

  5.You can Exit the game at any time by pressing 'E' or 'e' key

    So try to win in minimum no of move

                  Happy Gaming , Good Luck !

                  Enter any key to start.....
```



Rules Screen


Enter Name :
********** ->>
Screen Before name entered
\Project\Game\Number_Shifting_Game\bin\Debug\Number_Shifting_Game.exe
Enter Name :
********** ->> PaVan
Screen After Name Entered


Hello PaVan , You can select a challenge here.....
Types of Challeges :
1.Custume Challege
2.Auto Challege (Any Other Key)
(Available chances : 100)
Challenge Screen


botCamp With IOT\Project\Game\Number_Shifting_Game\bin\Debug\Number_Shifting_Game.exe
Custume challenge selected
Hello PaVan , You can select a challenge here.....
Types of Challeges :
1.Custume Challege
2.Auto Challege (Any Other Key)
(Available chances : 100)
Enter Number of chances you want to Complete the Game :

```
----------------------------------------------------------------------------

  Hello  PaVan , You can select a challenge here.....

                                Types of Challeges :
                    1.Custume Challege

                    2.Auto Challege (Any Other Key)
                        ( Available chances : 100  )

----------------------------------------------------------------------------

Enter Number of chances you want to Complete the Game : 10



**************************************************

                You have selected : CUSTUME CHALLENGE
                You will get : 10   Chances

**************************************************


          Press any key to Continue......
```

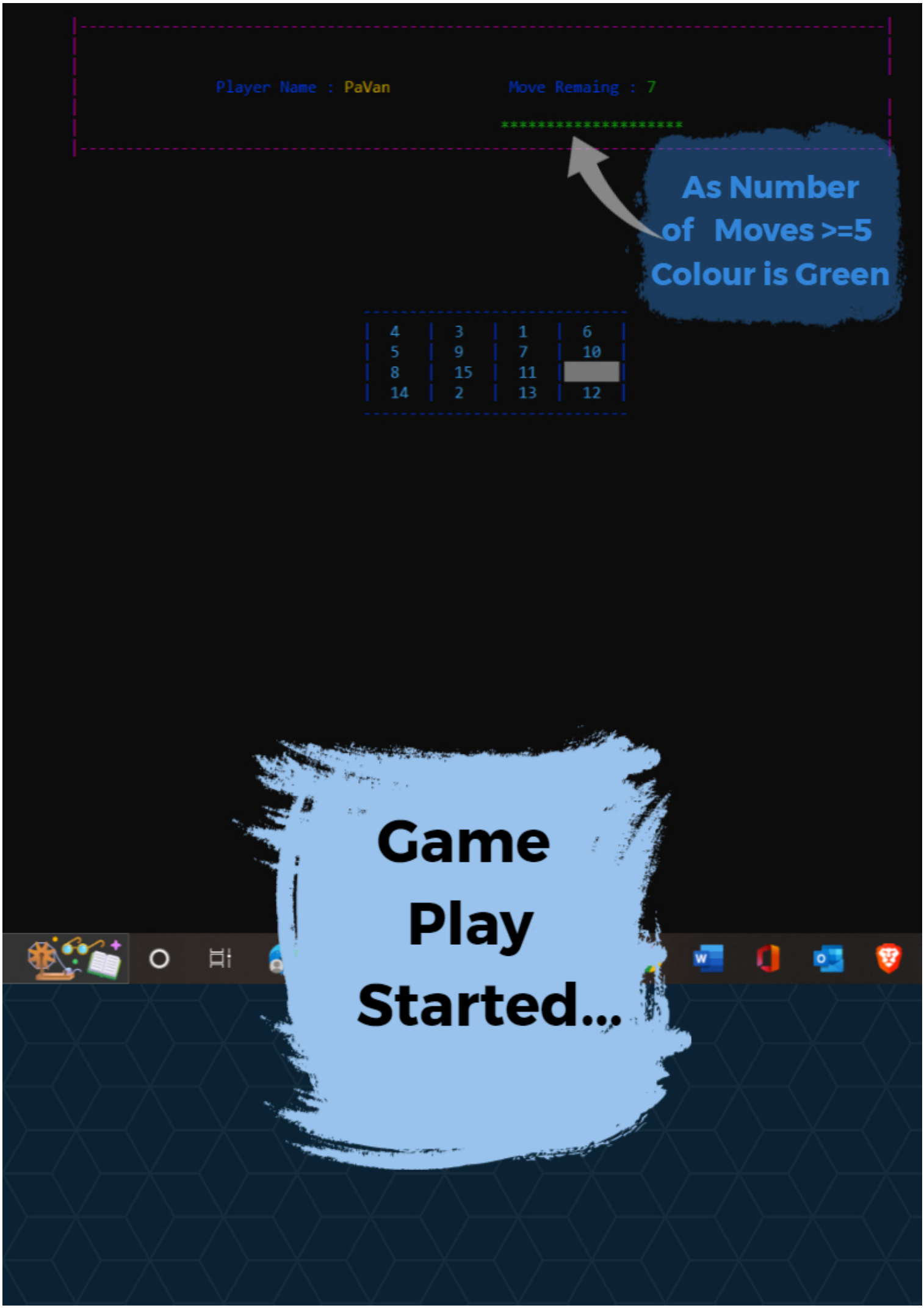
Number of Moves Entered

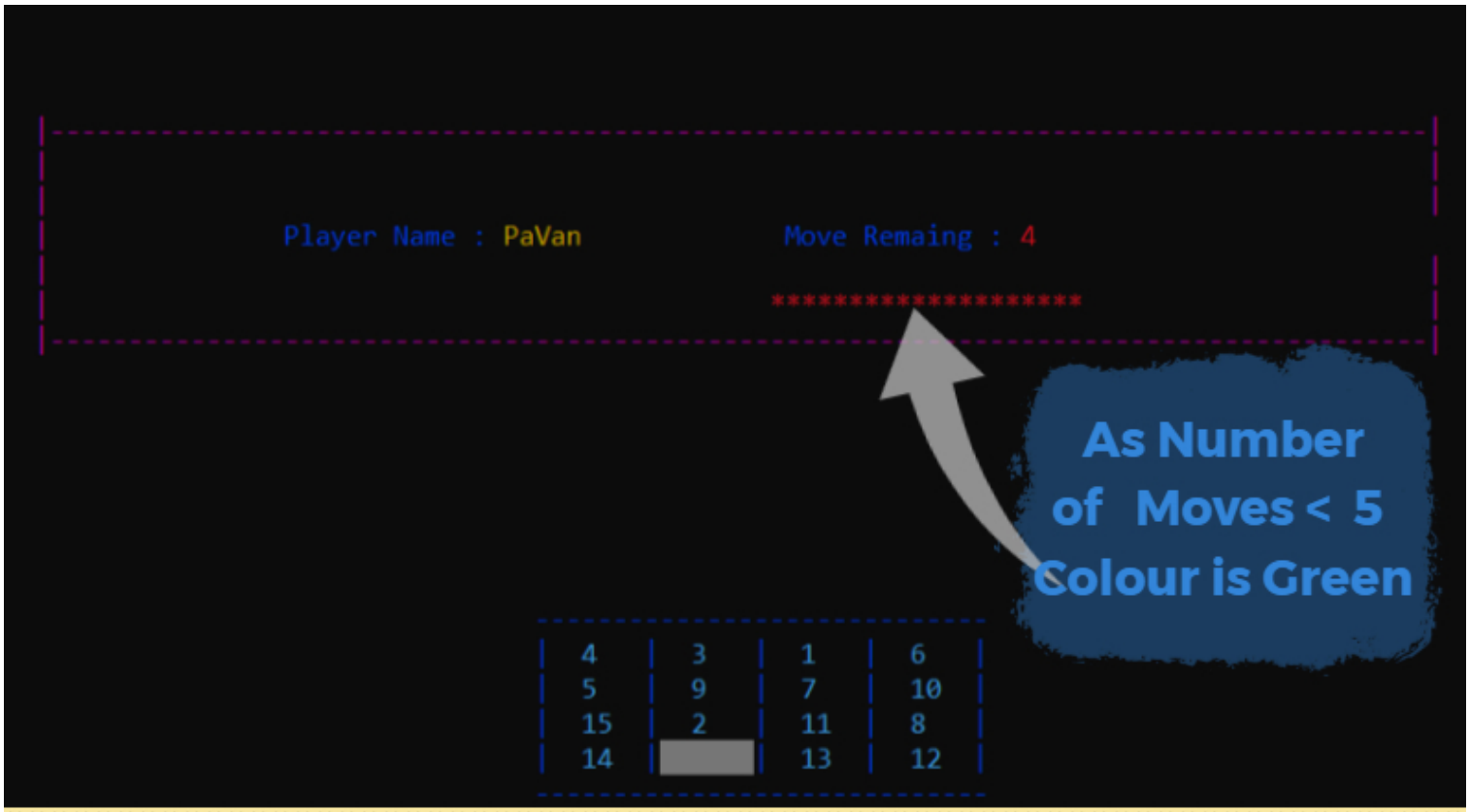
```
--------------------------------------------------------------------------------

  Hello  PaVan , You can select Mode of the Numbers-Shifting-Game  here.....

                              Types of Modes :
                  1.Random Mode
                      ( For Each Level : Random Orientation of elements  )

                  2.Default Mode (Any Other Key)
                      ( For Each Level  : Fixed Orientation of elements  )
--------------------------------------------------------------------------------


For
        1. Random Mode --> Press 1

        2. Default Mode --> Press any Other key
                                          -
```

Mode Selection Screen

```
--------------------------------------------------------------------------------

  Hello  PaVan , You can select Mode of the Numbers-Shifting-Game  here.....

                              Types of Modes :
                  1.Random Mode
                      ( For Each Level : Random Orientation of elements  )

                  2.Default Mode (Any Other Key)
                      ( For Each Level  : Fixed Orientation of elements  )
--------------------------------------------------------------------------------


For
        1. Random Mode --> Press 1

        2. Default Mode --> Press any Other key


**********************************************************
        You have selected : Random Mode
**********************************************************

                  Press any key to Continue...
```

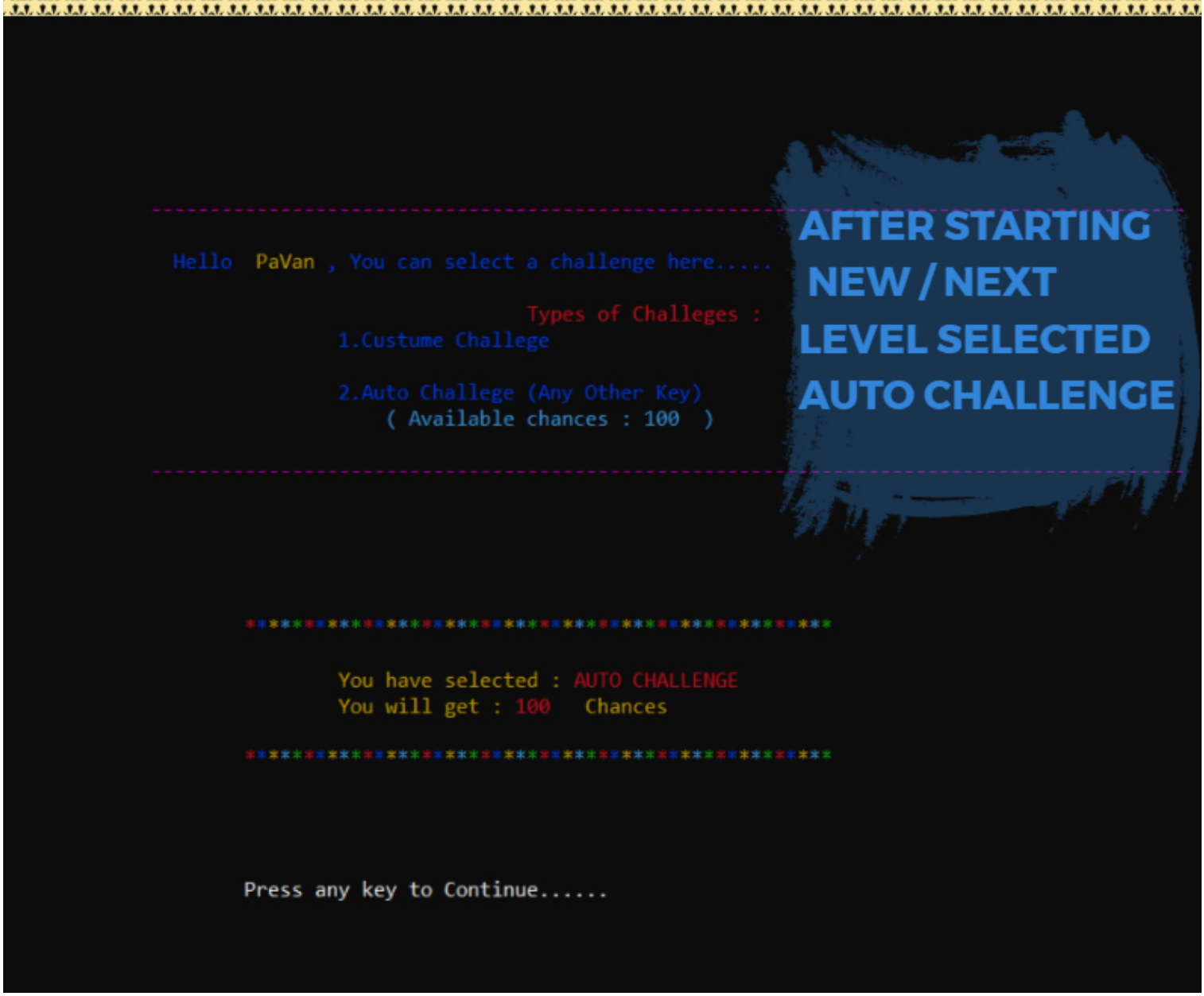
Mode selected

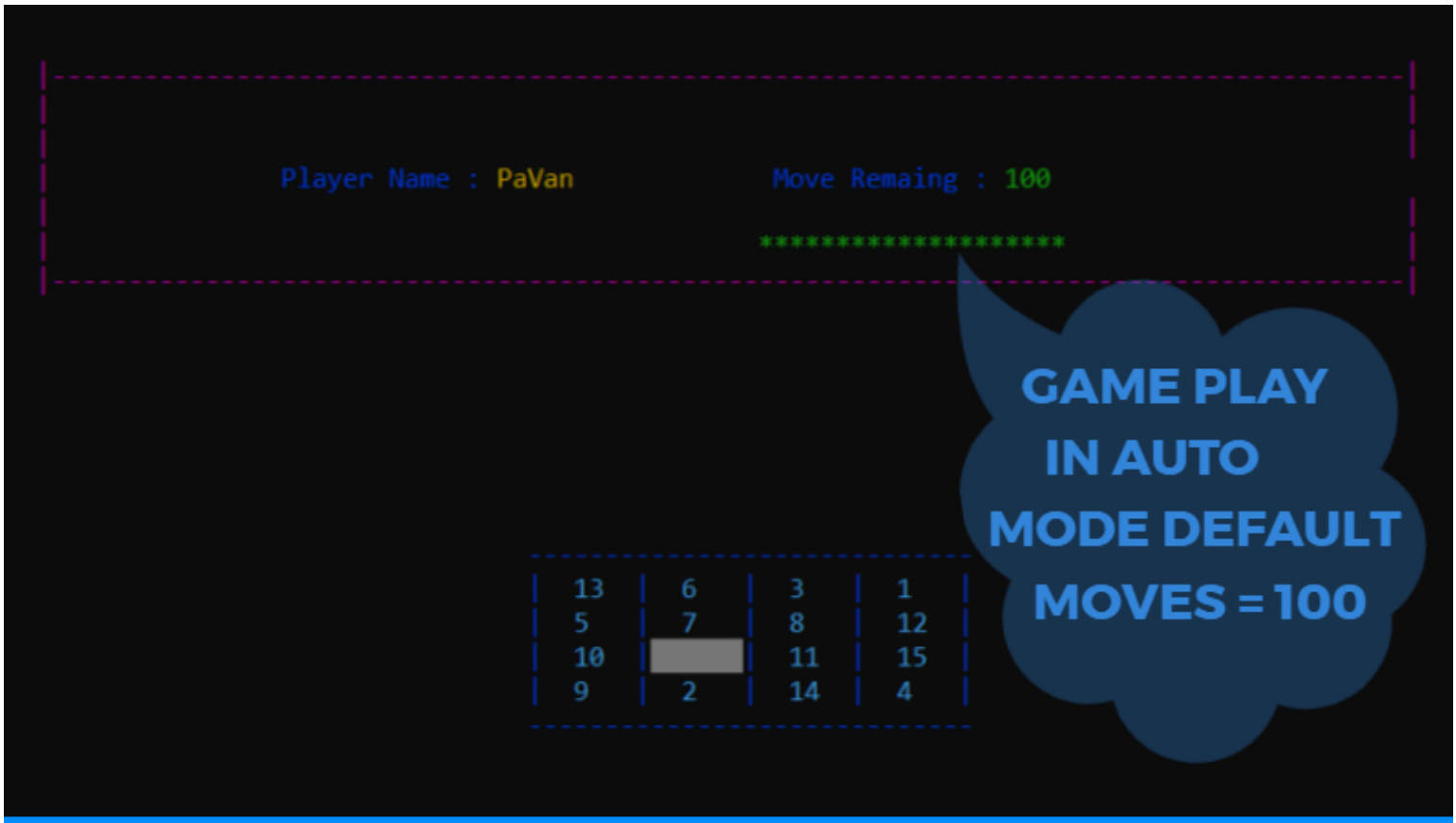

Player Name : PaVan
Move Remaing : 7
****
As Number of Moves >=5 Colour is Green
4 | 3 | 1 | 6
5 | 9 | 7 | 10
8 | 15 | 11 |
14 | 2 | 13 | 12
Game Play Started...



With IOT\Project\Game\Number_Shifting_Game\bin\Debug\Number_Shifting_Game.exe

BootCamp With IOT\Project\Game\Number_Shifting_Game\bin\Debug\Number_Shifting_Game.exe





IOT\Project\Game\Number_Shifting_Game\bin\Debug\Number_Shifting_Game.exe

Player Name : PaVan Move Remaing : 89

****

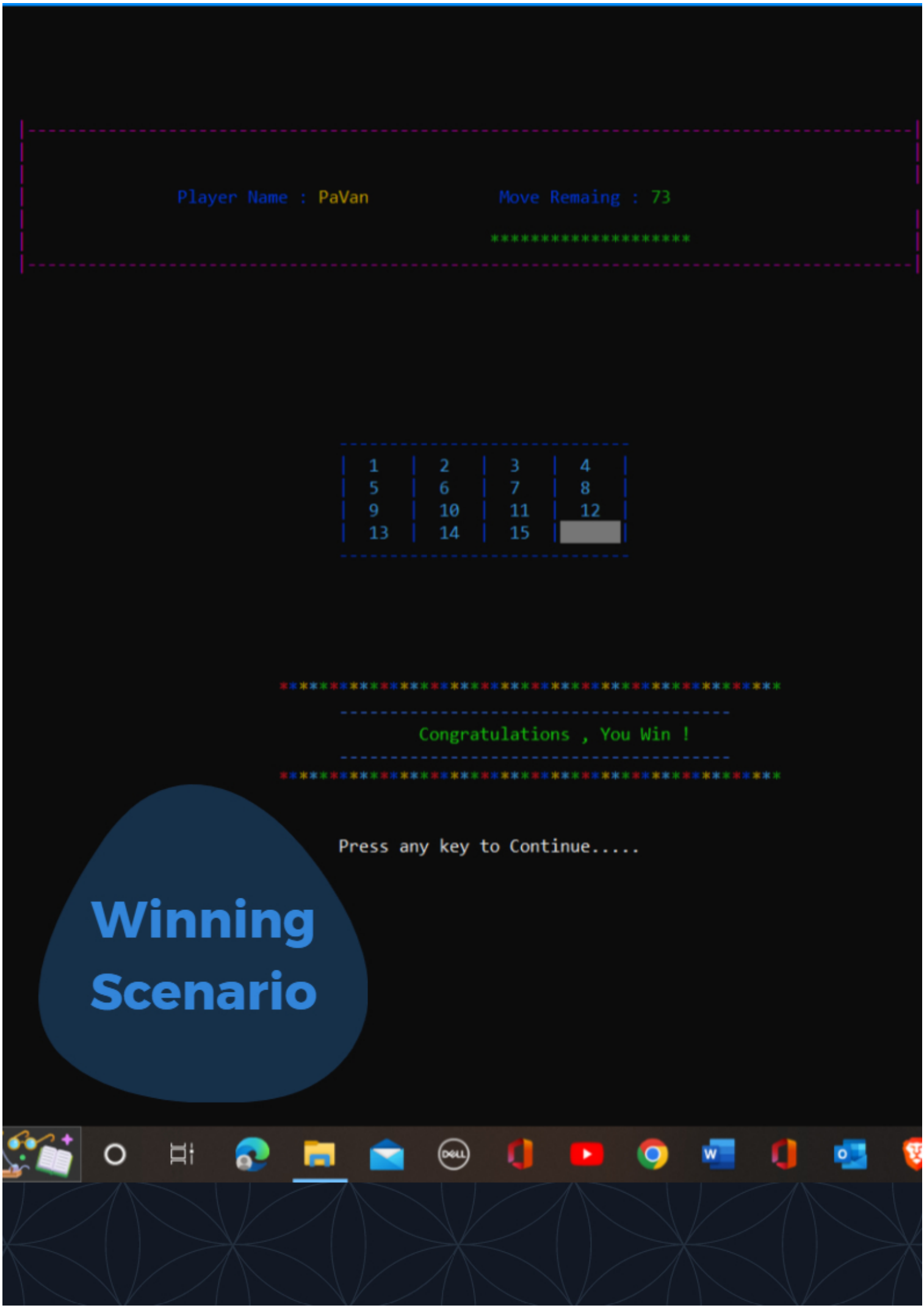
We can Escape anytime in-between the Game

1	6	3	4
5	7	8	12
9	10	11	15
13	2	14	

You have chosen the Exit Option Are you sure, you want to exit ?

Press-> Yes : 'y' , No : 'Press any other key'

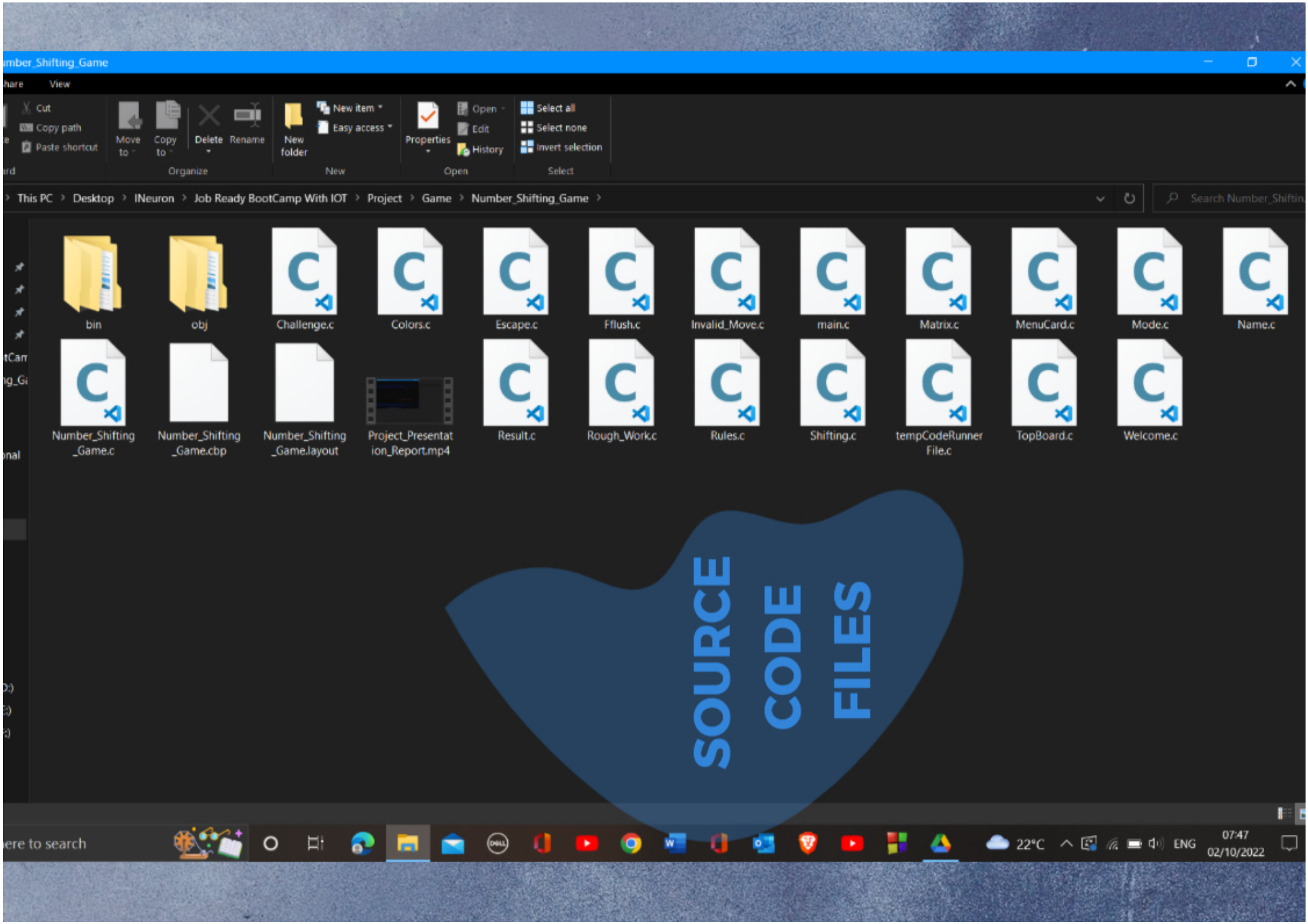
ere to search


Player Name : PaVan
Move Remaing : 73
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15
Congratulations , You Win !
Press any key to Continue.....
Winning Scenario

```
**********************************************************************************************
                                    Number Shifting Game
**********************************************************************************************



                                        Well Played


----------------------------------------
        Press
        1.Exit
        2.Continue Game(Press any other key)
----------------------------------------
```

tCamp With IOT\Project\Game\Number_Shifting_Game\bin\Debug\Number_Shifting_Game.exe

```
    ------------------------------------------------------------------------------------------

     Hello  PaVan , You can select Mode of the Numbers-Shifting-Game  here.....

                                    Types of Modes :
                        1.Random Mode
                            ( For Each Level : Random Orientation of elements  )

                        2.Default Mode (Any Other Key)
                            ( For Each Level  : Fixed Orientation of elements  )
    ------------------------------------------------------------------------------------------


    For
            1. Random Mode --> Press 1

            2. Default Mode --> Press any Other key

                Default Mode selected

            **************************************************
                    You have selected : Default Mode
            **************************************************

                        Press any key to Continue...
```


mber_Shifting_Game
hare
View
Cut
Copy path
Paste shortcut
Move to
Copy to
Delete
Rename
Organize
New item
Easy access
New folder
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
This PC
Desktop
INeuron
Job Ready BootCamp With IOT
Project
Game
Number_Shifting_Game
Search Number_Shiftin
bin
obj
Challenge.c
Colors.c
Escape.c
Fflush.c
Invalid_Move.c
main.c
Matrix.c
MenuCard.c
Mode.c
Name.c
Number_Shifting_Game.c
Number_Shifting_Game.cbp
Number_Shifting_Game.layout
Project_Presentation_Report.mp4
Result.c
Rough_Work.c
Rules.c
Shifting.c
tempCodeRunnerFile.c
TopBoard.c
Welcome.c
SOURCE CODE FILES
ere to search
22°C
ENG
07:47
02/10/2022

main.c

C: > Users > DELL > Desktop > INeuron > Job Ready BootCamp With IOT > Project > Game > Number

```c
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<time.h>
#include<string.h>


#include "Fflush.c"
#include "Welcome.c"
#include "Rules.c"
#include "Name.c"
#include "Challenge.c"
#include "Mode.c"
#include "TopBoard.c"
#include "Matrix.c"
#include "Invalid_Move.c"
#include "Escape.c"
#include "Result.c"
#include "MenuCard.c"
#include "Shifting.c"

void main(void)
{
    welcome();
    displayRules();

    char *game ;
    game = "on" ;
    int level = 0 ;

    while (game=="on")
    {

        char name[30] ;
        strcpy(name ,  getName());

        int stop = 1 ;  // Game exit variable
        int  row , col; // Position of the Empty Block
```

including all files

&

Driver Code Starts

⚠ 0

Type here to search

File Edit Selection View Go Run Terminal Help

main.c

C: > Users > DELL > Desktop > INeuron > Job Ready BootCamp With IOT > Project > Game > Nu

```
#include "shifting.c"

void main(void)
{
    welcome();
    displayRules();

    char *game ;
    game = "on" ;
    int level = 0 ;

    while (game=="on")
    {

        char name[30] ;
        strcpy(name ,  getName());

        int stop = 1 ;  // Game exit variable
        int  row , col; // Position of the Empty Block
        int chance = challenge(name) ; // Number of chances Player

        int **A = mode(name);
        int loopVar = 1 ;
        for(int i=0 ;  i<4 && loopVar  ; i++)
        {
            for(int j=0;j<4;j++)
            {
                if(A[i][j]==0)
                {
                    row = i ;
                    col = j ;
                    loopVar = 0;
                    break ;
                }
            }
        }
        /* code */
        char move ;
```

0 ⚠ 0

Type here to search

```c
/* code */
char move ;
while( chance>0 && stop)
{
    // printf("Player Name : %s\t\tMove Remaining : %d\n\n\n",name,chance);

    displayNameChance(name,chance);
    matrix(A);

    // Move Validation

    while (1)
    {
        fflush(stdin);
        move = getch();

        if( move==-32)
        {
            move = getch();
            break;
        }
        else if(move=='e' || move=='E' || move==72 || move==75 || move==77 || move==80
        {
            break;
        }
        else
        {
            system("cls");
            // printf("Player Name : %s\t\tMove Remaining : %d\n\n\n",name,chance);
            displayNameChance(name,chance);
            matrix(A);
            invalid();
        }

    }

    // Shifting Element According to the move
```

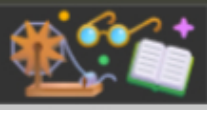
```
        // Shifting Element According to the move
        if( move==72 || move==75 || move==77 || move==80 )

        {
            switch(move)
            {
            case 72:
                // shiftUp(A,&row,&col);
                shiftDown(A,&row,&col);
                break;
            case 80:
                shiftUp(A,&row,&col);
                // shiftDown(A,&row,&col);
                break;
            case 75:
                shiftRight(A,&row,&col);
                // shiftLeft(A,&row,&col);
                break;
            case 77:
                shiftLeft(A,&row,&col);
                // shiftRight(A,&row,&col);
                break;
            }

            if(win(A,name,chance))
            {
                // displayNameChance(name, chance);
                chance  = -1 ;
            }
            else
            {
                // Reducing Moves
                chance-- ;
            }

        }
        // Choice is Exit
        else
```

main.c

> Users > DELL > Desktop > INeuron > Job Ready BootCamp With IOT > Project > Game > Number_Sh

```
13                      break;
14                  }
15
16                  if(win(A,name,chance))
17                  {
18                      // displayNameChance(name, chance);
19                      chance  = -1 ;
20                  }
21                  else
22                  {
23                      // Reducing Moves
24                      chance-- ;
25                  }
26
27              }
28              // Choice is Exit
29              else
30              {
31                  char exitChar=escape();
32                  if(exitChar=='y' || exitChar=='Y')
33                      stop = 0;
34              }
35          }
36
37          if(chance==0)
38          {
39              // printf("Player Name : %s\t\tMove Remaining : %d\n\n\n",nar
40              displayNameChance(name , chance);
41              matrix(A);
42              lose();
43          }
44
45          game = menuCard();
46
47      }
48
49  }
50
```

Driver Code End



Type here to search

In Project Report we have added the Driver code only .
The actual program is made of the 14 source code files which are shown in project report and around 1200 lines of code .

Entertainment

+

Logical ability

Number Shifting Game

Thank You Very Much

– *Pavankumar Mhaske*