Alexandria University
Faculty Of Engineering
CSED
Level 1 – Term 1

Programming 1

Prof. Nagia M. Ghanem

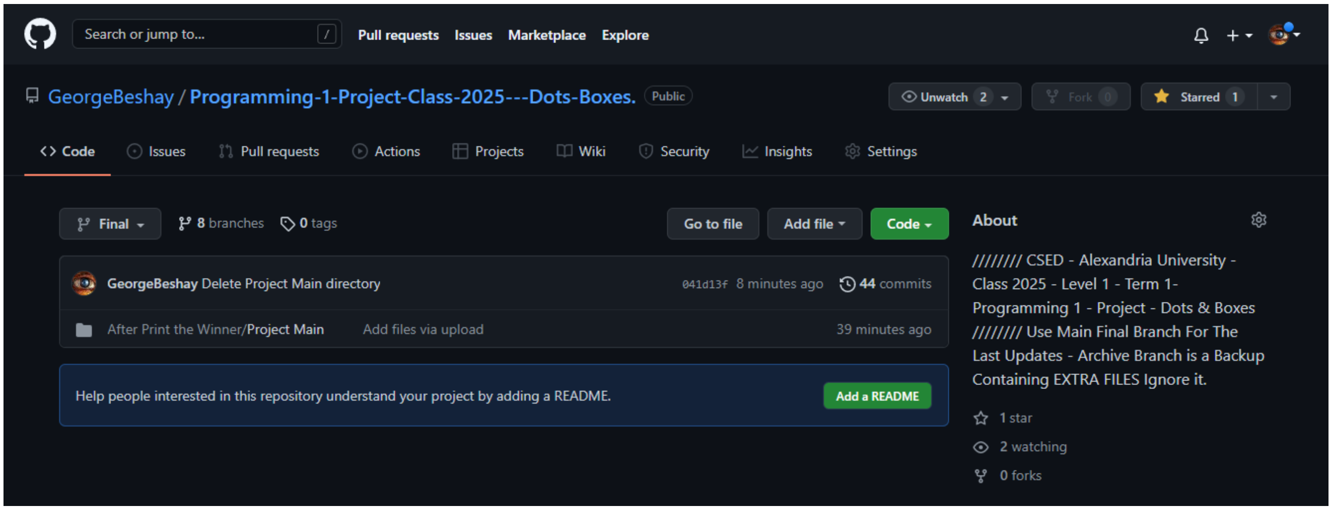
Dots & Boxes Game – Project

By:
1st Student Name: Mariam Aziz Gerges Zaki
1st Student ID: 20011889
2nd Student Name: George Samy Wahba Beshay
2nd Student ID: 20010435

1) Description

- *We implemented **"Dots & Boxes"** Game (PC Version) Using the **C** as a programming language, that is working efficiently without crashing under the assumptions that will be mentioned later in the report.*
- *Our Game folder is consisting from 34 files **(.h , .c , .txt , .exe)** that's because we have tried to divide the game to modules **(Header files)** so it can be easily applied and debugged.*
- *Our **game.c** File (main.c) is consisting from nearly 2200 lines.*
- *The Game divided can be divided to 5 Main Blocks*
  - PvP(Player Vs Player) Beginner Mode
  - PvP Advanced Mode
  - PvC Beginner Mode
  - PvC Advanced Mode
  - Top 10 List For Both PvC & PvP Modes.

- *As for the workspace and IDE of the team members, we have been using more than 1 application, and they are listed Below:*
  - Visual Studio Code #Coding
  - GitHub #Workspace For team members to cooperate
  GeorgeBeshay/Programming-1-Project-Class-2025---Dots-Boxes.: //////// CSED - Alexandria University - Class 2025 - Level 1 - Term 1- Programming 1 - Project - Dots & Boxes //////// Use Main Final Branch For The Last Updates - Archive Branch is a Backup Containing EXTRA FILES Ignore it. (github.com)
  PS: The link gives no access to edit/modify in the repository



GeorgeBeshay / Programming-1-Project-Class-2025---Dots-Boxes. Public

GeorgeBeshay Delete Project Main directory   041d13f 8 minutes ago   44 commits

After Print the Winner/Project Main   Add files via upload   39 minutes ago

Help people interested in this repository understand your project by adding a README.

About

//////// CSED - Alexandria University - Class 2025 - Level 1 - Term 1- Programming 1 - Project - Dots & Boxes //////// Use Main Final Branch For The Last Updates - Archive Branch is a Backup Containing EXTRA FILES Ignore it.

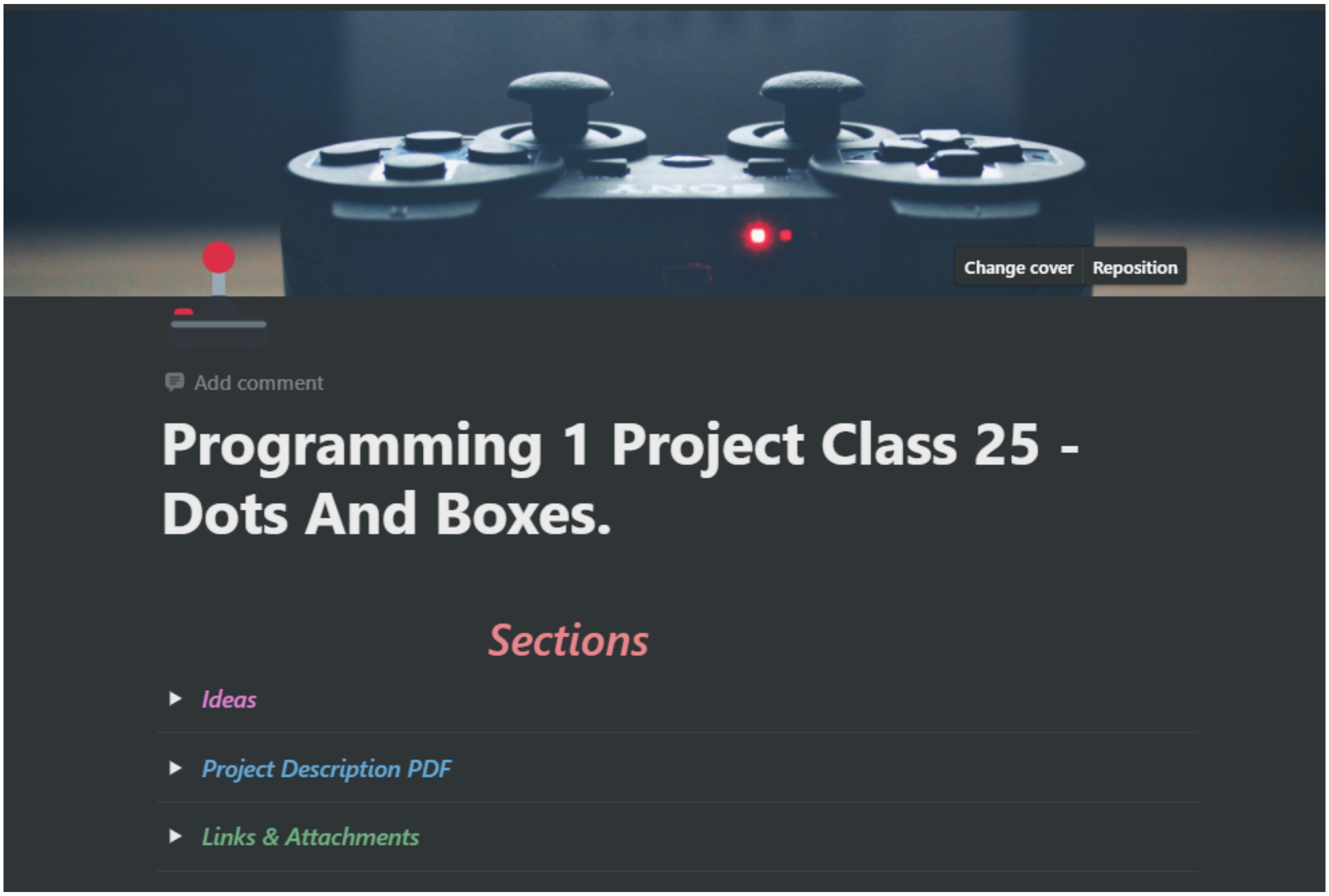
1 star

2 watching

0 forks

- Notion #Workspace For team members to cooperate
https://elderly-coelurus-bf6.notion.site/Programming-1-Project-Class-25-Dots-And-Boxes-99a4bdf305164d16aa1d749fcbbb16e1

PS: The link gives no access to edit/modify in the workspace



Programming 1 Project Class 25 - Dots And Boxes.

Sections

- *Ideas*
- *Project Description PDF*
- *Links & Attachments*

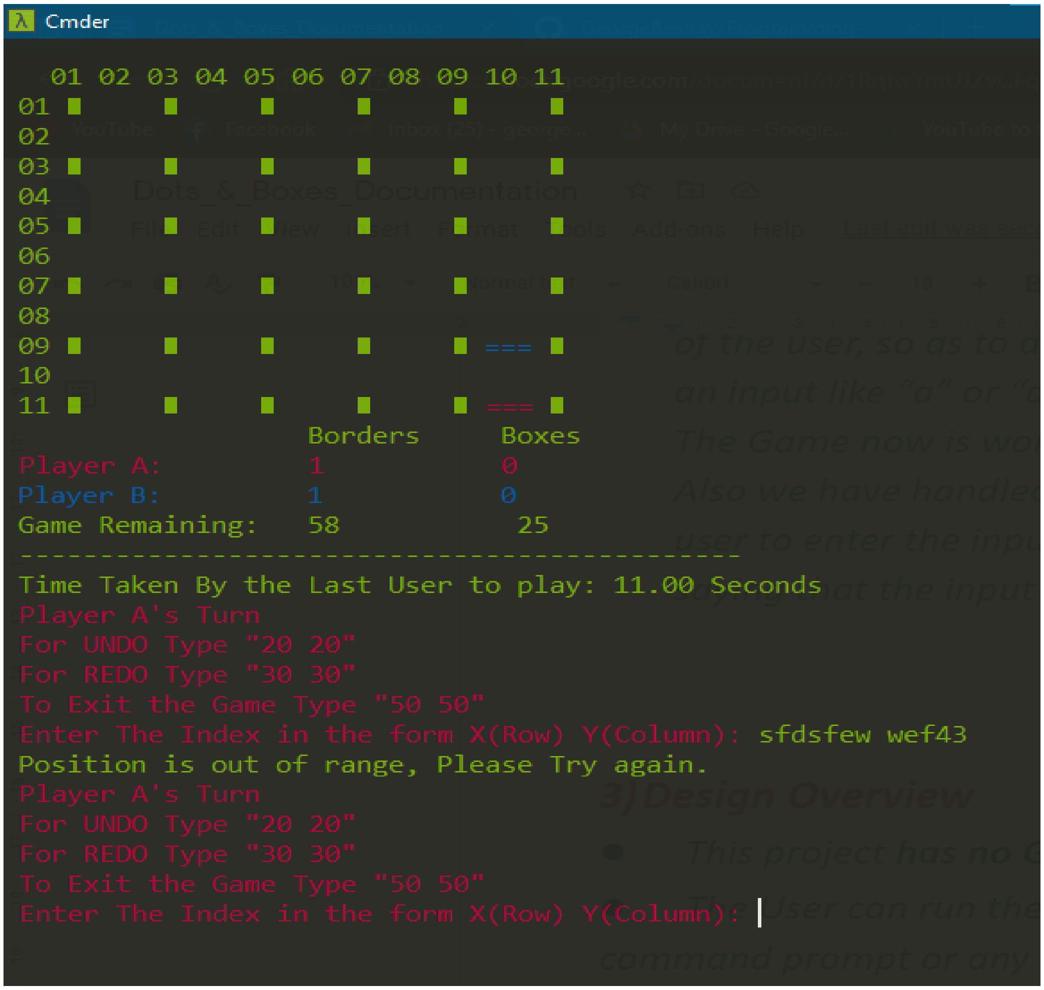
- *The Game Logic has been Developed By the students after full realization of the game concepts and rules, from the provided Wikipedia link in the project description PDF.*
- *The Game Path is defined as:*
  ** Main-Menu > Difficulty Level > New or Existing Game > Game Start Process > Game End.*

- *Easily Playable Game, The User pass the inputs through the keyboard, by entering the choice number wanted.*

2) Features

- ***4 Different Modes** Listed below:*
  - PvP Beginner Mode
  - PvP Advanced Mode
  - PvC Beginner Mode
  - PvC Advanced Mode
- ***Save and Loading** Games in **all of the 4 Modes mentioned before.***
- ***Undo & Redo** Option in **all of the 4 Modes mentioned before.***
- *Displaying the **Current Game_Info in details** after each move like:*
  - number of borders remaining
  - number of boxes remaining
  - Each player current score
- *Displaying **time taken by each user** to play his move.*
- ***Automatic Score Updating System***
  After each game end, The 2 players' scores will be automatically compared to the current Top 10 list in the specific mode that the users are playing, and if any of the scores is high enough, it will be shown in the top 10 list. if the user is already present in the list, his score will be updated.

- *AI Module, Providing the PvC Mode with computer logic to play against, Based on Checking each box and its surrounding borders, then deciding which box is better to place a border at.*
- *We have been working on creating a gate / filter for the input of the user, so as to avoid crashing when a character is given as an input like “a” or “abc”* ***IN ALL THE GAME MODES***
  The Game now is working efficiently with no cases crashing at, Also we have handled the characters input, by Re-directing the user to enter the input again, after printing a warning message saying that the input given is not valid or out of range.



Cmder

```
  01 02 03 04 05 06 07 08 09 10 11
01 █     █     █     █     █     █
02
03 █     █     █     █     █     █
04
05 █     █     █     █     █     █
06
07 █     █     █     █     █     █
08
09 █     █     █     █     █ === █
10
11 █     █     █     █     █ === █
                  Borders     Boxes
Player A:         1           0
Player B:         1           0
Game Remaining:   58           25
---------------------------------------------
Time Taken By the Last User to play: 11.00 Seconds
Player A's Turn
For UNDO Type "20 20"
For REDO Type "30 30"
To Exit the Game Type "50 50"
Enter The Index in the form X(Row) Y(Column): sfdsfew wef43
Position is out of range, Please Try again.
Player A's Turn
For UNDO Type "20 20"
For REDO Type "30 30"
To Exit the Game Type "50 50"
Enter The Index in the form X(Row) Y(Column):
```

3) Design Overview

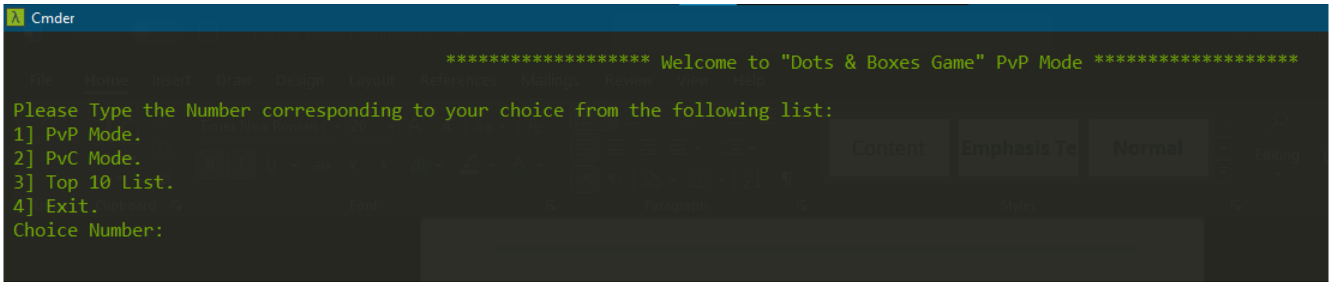
- *This project **has no GUI***
- *To avoid user distraction by the different sentences that would be printed, we have used a specific colour for each user listed below:*

- Player A >> Red

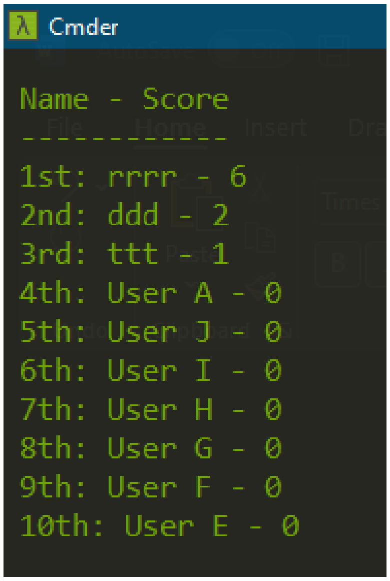
- Player B >> Blue

- Game Adminstration >> Green

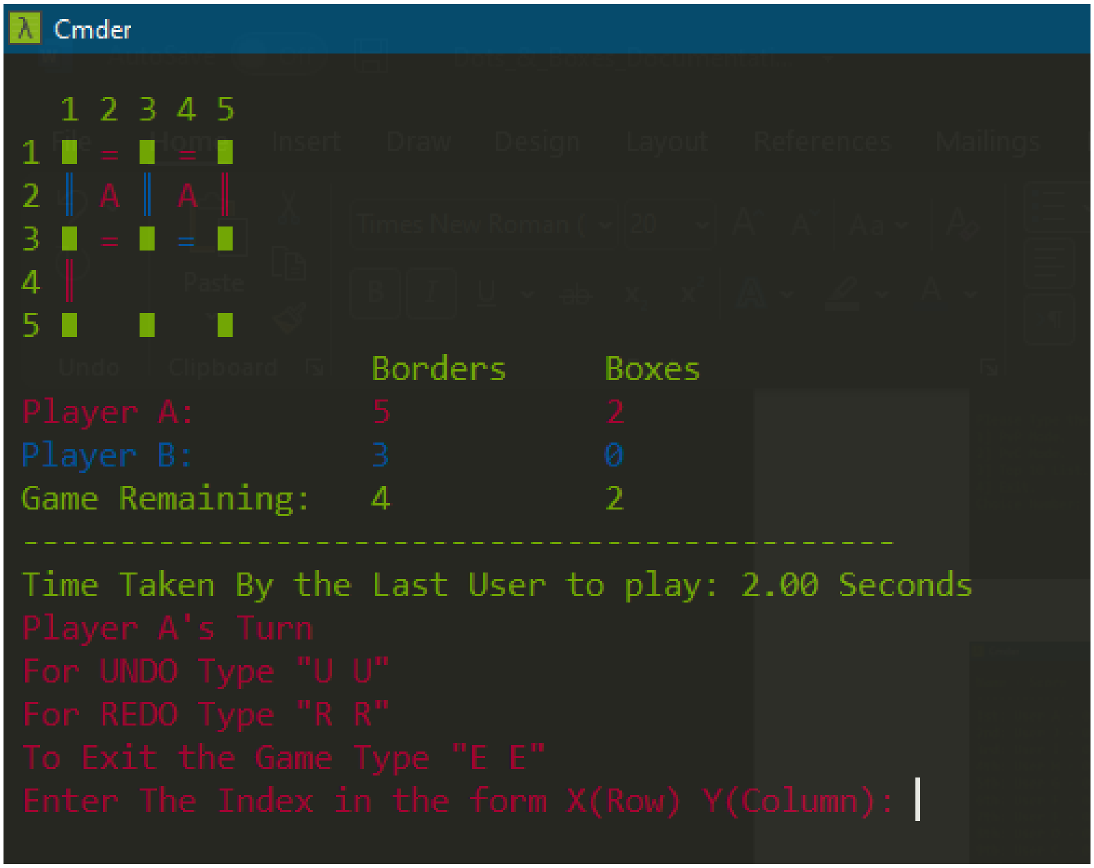
- *The User can run the Game.exe through the windows command prompt or any external terminal like "**Cmder**" etc …*
- *The Below Screenshots are Random Screenshots taken During the Game.exe Running on **"Cmder"***



```
                                             ******************* Welcome to "Dots & Boxes Game" PvP Mode *******************

Please Type the Number corresponding to your choice from the following list:
1] PvP Mode.
2] PvC Mode.
3] Top 10 List.
4] Exit.
Choice Number:
```

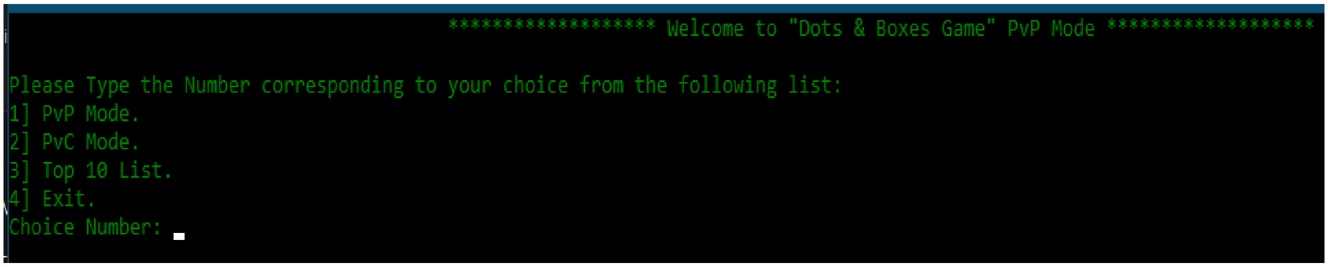


```
Name - Score
------------
1st: rrrr - 6
2nd: ddd - 2
3rd: ttt - 1
4th: User A - 0
5th: User J - 0
6th: User I - 0
7th: User H - 0
8th: User G - 0
9th: User F - 0
10th: User E - 0
```

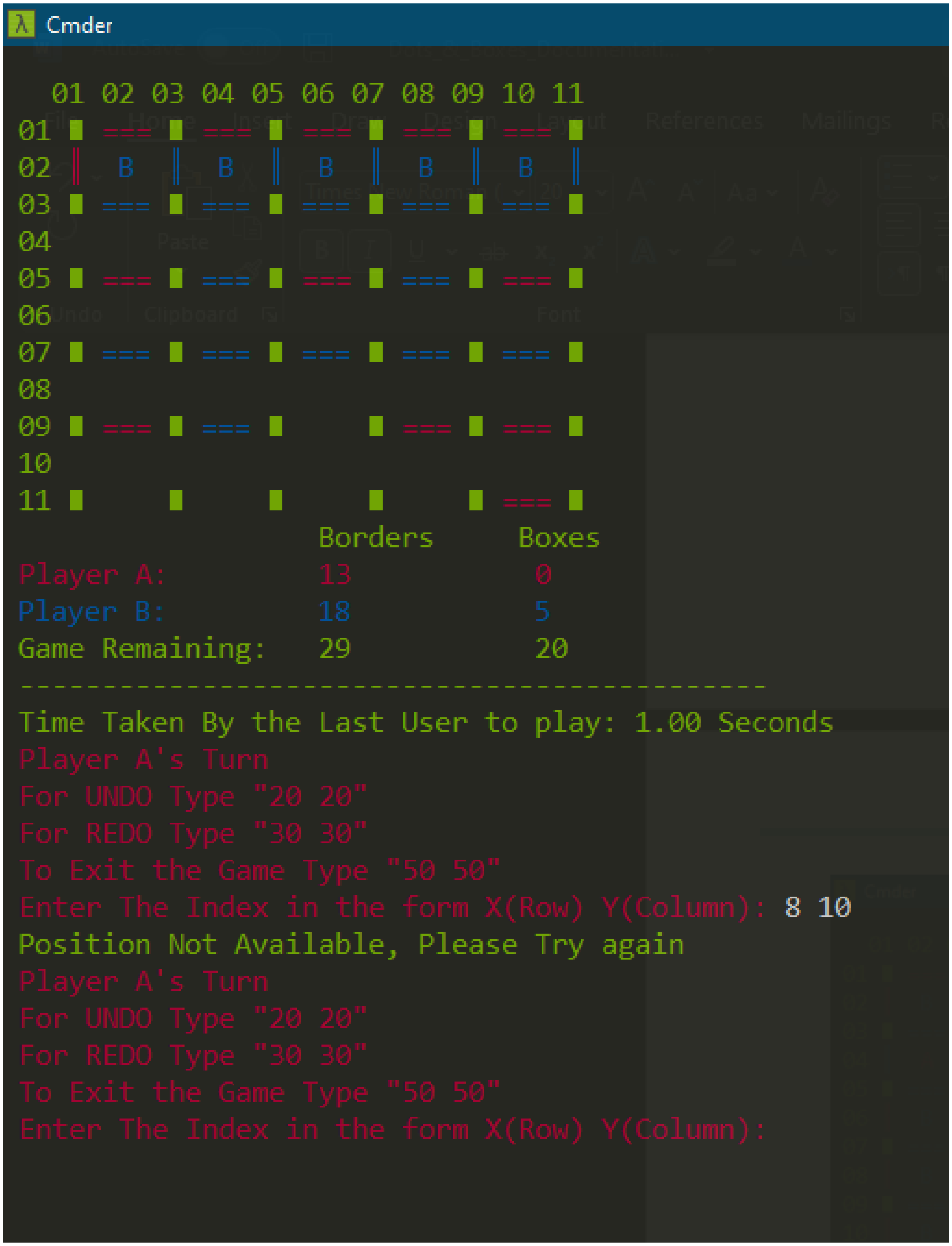


```
  1 2 3 4 5
1 █ = █ = █
2 ║ A ║ A ║
3 █ = █ = █
4 ║
5 █   █   █
                    Borders     Boxes
Player A:           5           2
Player B:           3           0
Game Remaining:     4           2
---------------------------------------------
Time Taken By the Last User to play: 2.00 Seconds
Player A's Turn
For UNDO Type "U U"
For REDO Type "R R"
To Exit the Game Type "E E"
Enter The Index in the form X(Row) Y(Column):
```

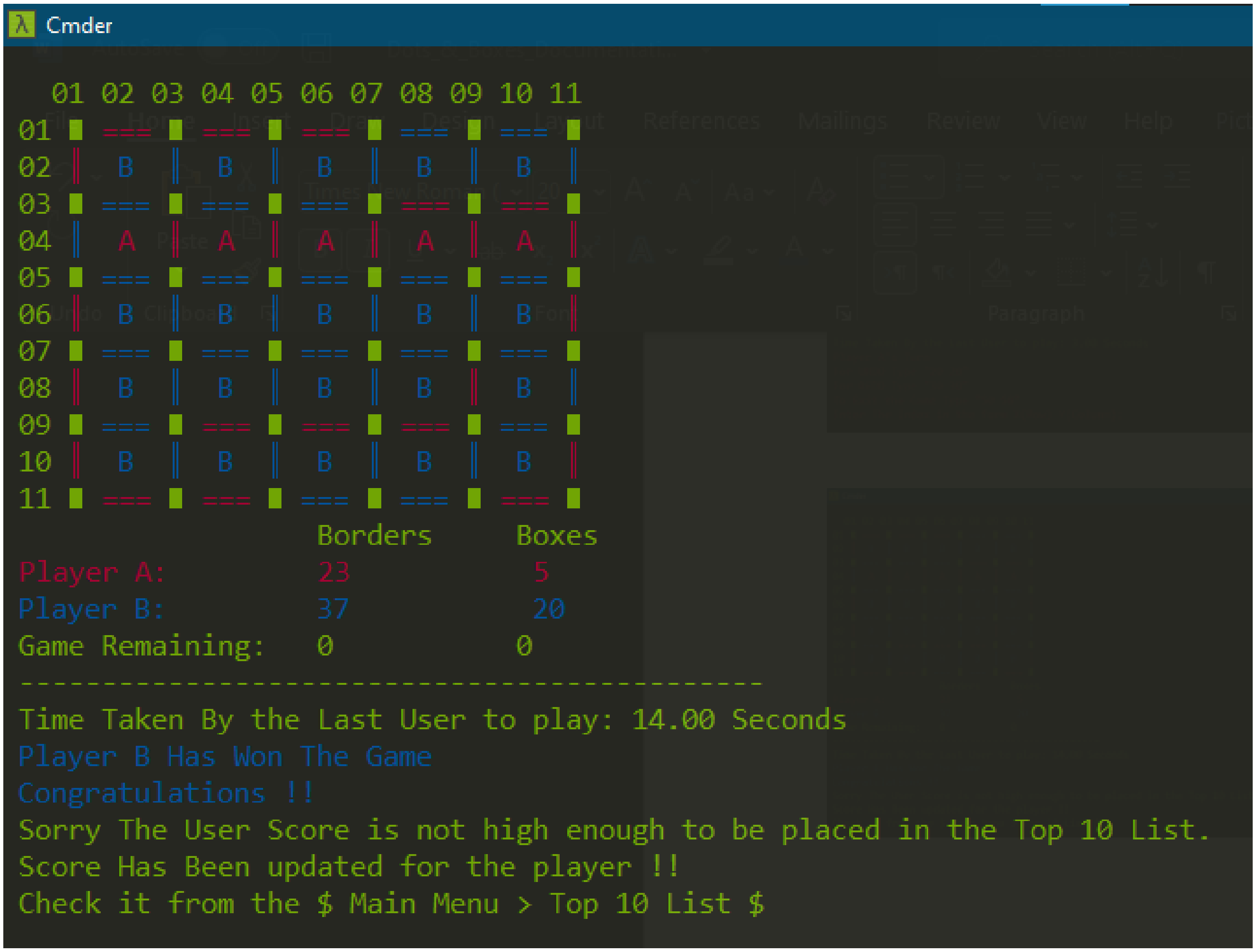
As Mentioned before, we can run the "Game.exe" by the normal windows command prompt:



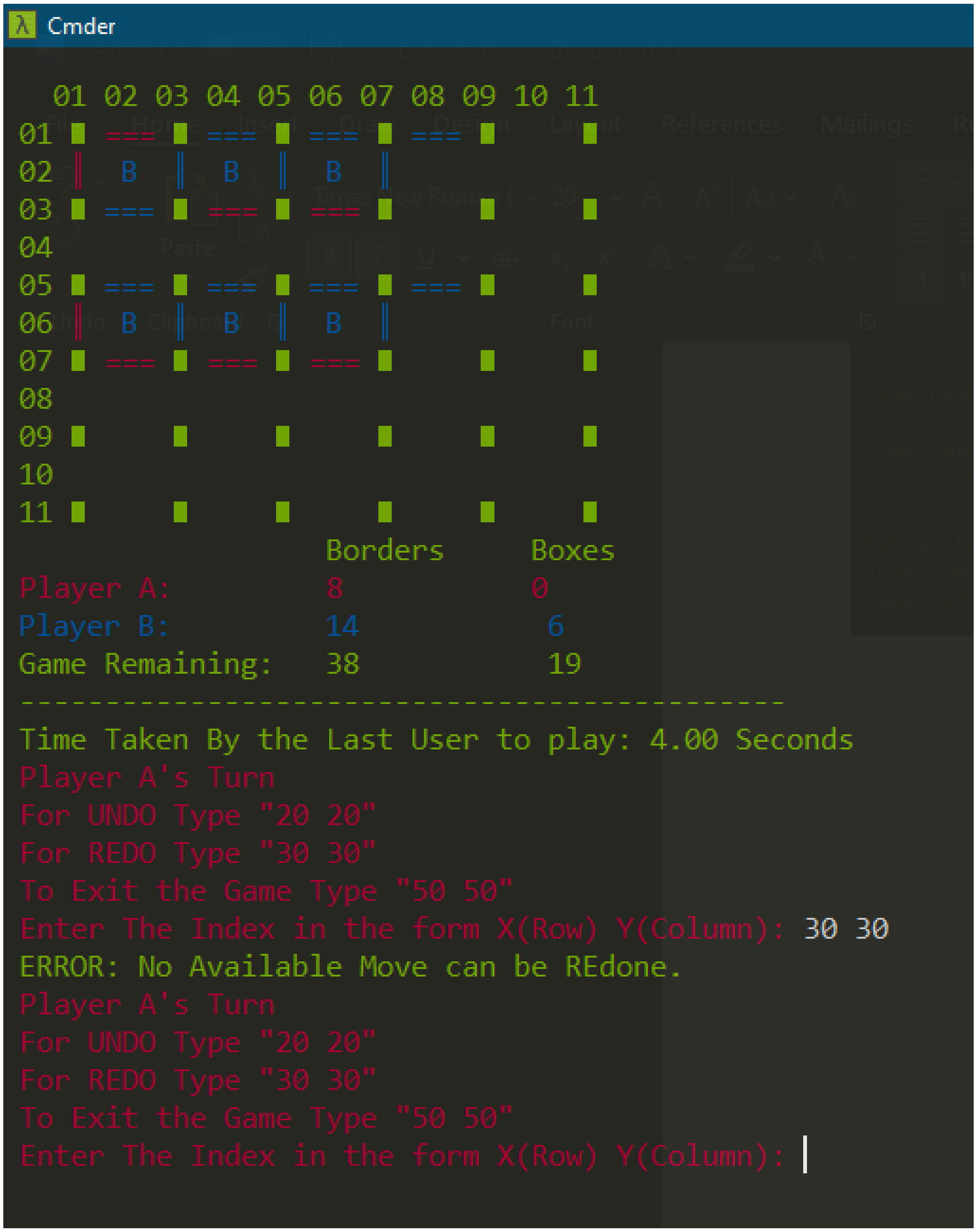
```
                                        ******************* Welcome to "Dots & Boxes Game" PvP Mode *******************

Please Type the Number corresponding to your choice from the following list:
1] PvP Mode.
2] PvC Mode.
3] Top 10 List.
4] Exit.
Choice Number: _
```



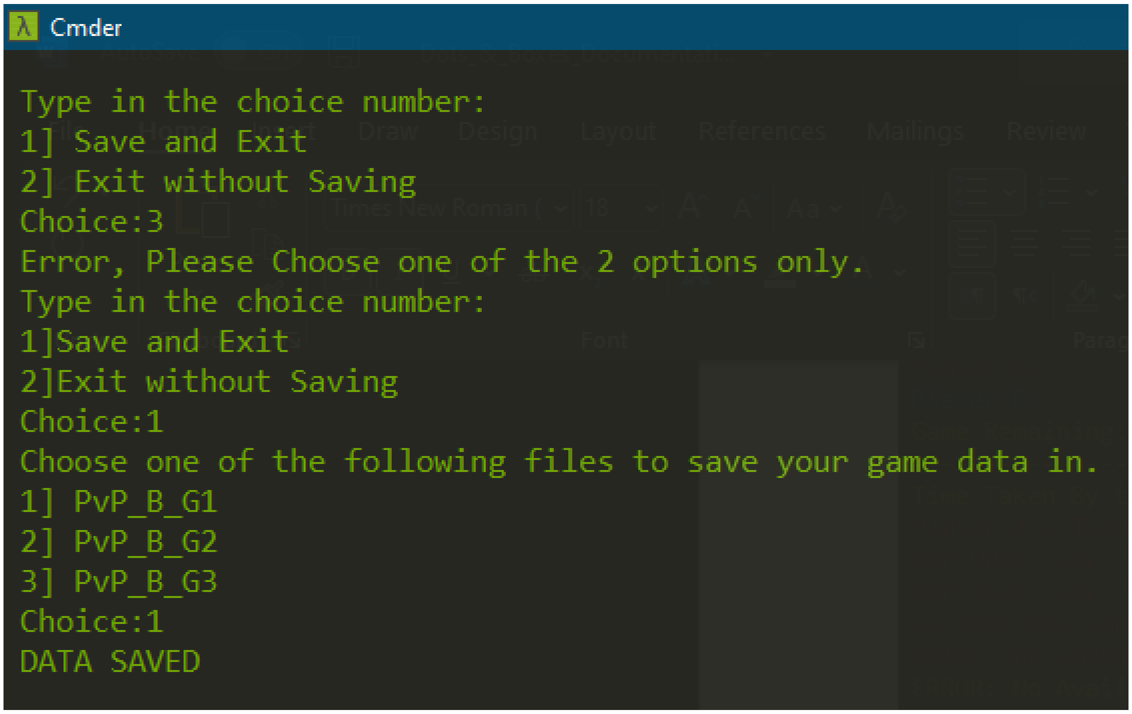
```
   01 02 03 04 05 06 07 08 09 10 11
01 ▮ === ▮ === ▮ === ▮ === ▮ === ▮
02 ║  B  ║  B  ║  B  ║  B  ║  B  ║
03 ▮ === ▮ === ▮ === ▮ === ▮ === ▮
04
05 ▮ === ▮ === ▮ === ▮ === ▮ === ▮
06
07 ▮ === ▮ === ▮ === ▮ === ▮ === ▮
08
09 ▮ === ▮ === ▮     ▮ === ▮ === ▮
10
11 ▮     ▮     ▮     ▮     ▮ === ▮
                  Borders     Boxes
Player A:         13          0
Player B:         18          5
Game Remaining:   29          20
---------------------------------------------
Time Taken By the Last User to play: 1.00 Seconds
Player A's Turn
For UNDO Type "20 20"
For REDO Type "30 30"
To Exit the Game Type "50 50"
Enter The Index in the form X(Row) Y(Column): 8 10
Position Not Available, Please Try again
Player A's Turn
For UNDO Type "20 20"
For REDO Type "30 30"
To Exit the Game Type "50 50"
Enter The Index in the form X(Row) Y(Column):
```



```
  01 02 03 04 05 06 07 08 09 10 11
01 █ === █ === █ === █ === █ === █
02 ║  B  ║  B  ║  B  ║  B  ║  B  ║
03 █ === █ === █ === █ === █ === █
04 ║  A  ║  A  ║  A  ║  A  ║  A  ║
05 █ === █ === █ === █ === █ === █
06 ║  B  ║  B  ║  B  ║  B  ║  B  ║
07 █ === █ === █ === █ === █ === █
08 ║  B  ║  B  ║  B  ║  B  ║  B  ║
09 █ === █ === █ === █ === █ === █
10 ║  B  ║  B  ║  B  ║  B  ║  B  ║
11 █ === █ === █ === █ === █ === █
                  Borders     Boxes
Player A:         23           5
Player B:         37           20
Game Remaining:   0           0
---------------------------------------------
Time Taken By the Last User to play: 14.00 Seconds
Player B Has Won The Game
Congratulations !!
Sorry The User Score is not high enough to be placed in the Top 10 List.
Score Has Been updated for the player !!
Check it from the $ Main Menu > Top 10 List $
```



```
  01 02 03 04 05 06 07 08 09 10 11
01 █ === █ === █ === █ === █     █
02 ║  B  ║  B  ║  B  ║
03 █ === █ === █ === █     █     █
04
05 █ === █ === █ === █ === █     █
06 ║  B  ║  B  ║  B  ║
07 █ === █ === █ === █     █     █
08
09 █     █     █     █     █     █
10
11 █     █     █     █     █     █
                  Borders     Boxes
Player A:         8           0
Player B:         14           6
Game Remaining:   38           19
---------------------------------------------
Time Taken By the Last User to play: 4.00 Seconds
Player A's Turn
For UNDO Type "20 20"
For REDO Type "30 30"
To Exit the Game Type "50 50"
Enter The Index in the form X(Row) Y(Column): 30 30
ERROR: No Available Move can be REdone.
Player A's Turn
For UNDO Type "20 20"
For REDO Type "30 30"
To Exit the Game Type "50 50"
Enter The Index in the form X(Row) Y(Column): 
```



```
Type in the choice number:
1] Save and Exit
2] Exit without Saving
Choice:3
Error, Please Choose one of the 2 options only.
Type in the choice number:
1]Save and Exit
2]Exit without Saving
Choice:1
Choose one of the following files to save your game data in.
1] PvP_B_G1
2] PvP_B_G2
3] PvP_B_G3
Choice:1
DATA SAVED
```

4) Assumptions

- *The Following assumptions **SHOULD BE TAKEN IN CONSIDERATION***

  ---------------------- VERY IMPORTANT ----------------------

  ** On Loading a pre-existing game, The user can not undo the last move made before saving the game.*

  ***PS:** Even after loading the game, on trying to undo / redo, system will print that there is no move can be undone and **WILL NOT CRASH** however, The User STILL **CAN UNDO/REDO** the moves made after loading the game.*

** About the Undo/Redo Concept*
in PvP: The Undo option is given to the user so as to undo the last player move.
so when player B choose to undo, player A Move will be undone.
when player A choose to undo, player B move will be undone.
when player B choose to redo, player A move that was undone, will be redone.
*notice that the redo option happens only when a move is undone, **ONLY AT THIS INSTANT,** but if a move was undone, then the player made a new move, now there is no move can be redone, and this is reasonable.*
*in PvC: The Undo option -which is ofcourse given only to the Human player- will undo **ALL** the computer moves, including the boxes created in a row consecutive.*

** As mentioned before, The code has been built to handle any kind of inputs (characters,numbers) without any program crashing, displaying an error message to the user, informing him that the given input was not valid (incorrect) or out of range, or even that the place chosen is already taken and not available.*

5) Data Structures

- *We have created multiple structures, that are needed during the game operations in the 4 different modes.*

PS: All of the Data Structures defined for the Beginner Mode are *also defined for the Advanced Mode but we have modified the name of the structure by adding “A_” before the structure name, also we have modified the used sizes of boards so as to follow up the Advanced_Board Size.*

1] Player_Ingame_Info Structure

Members:

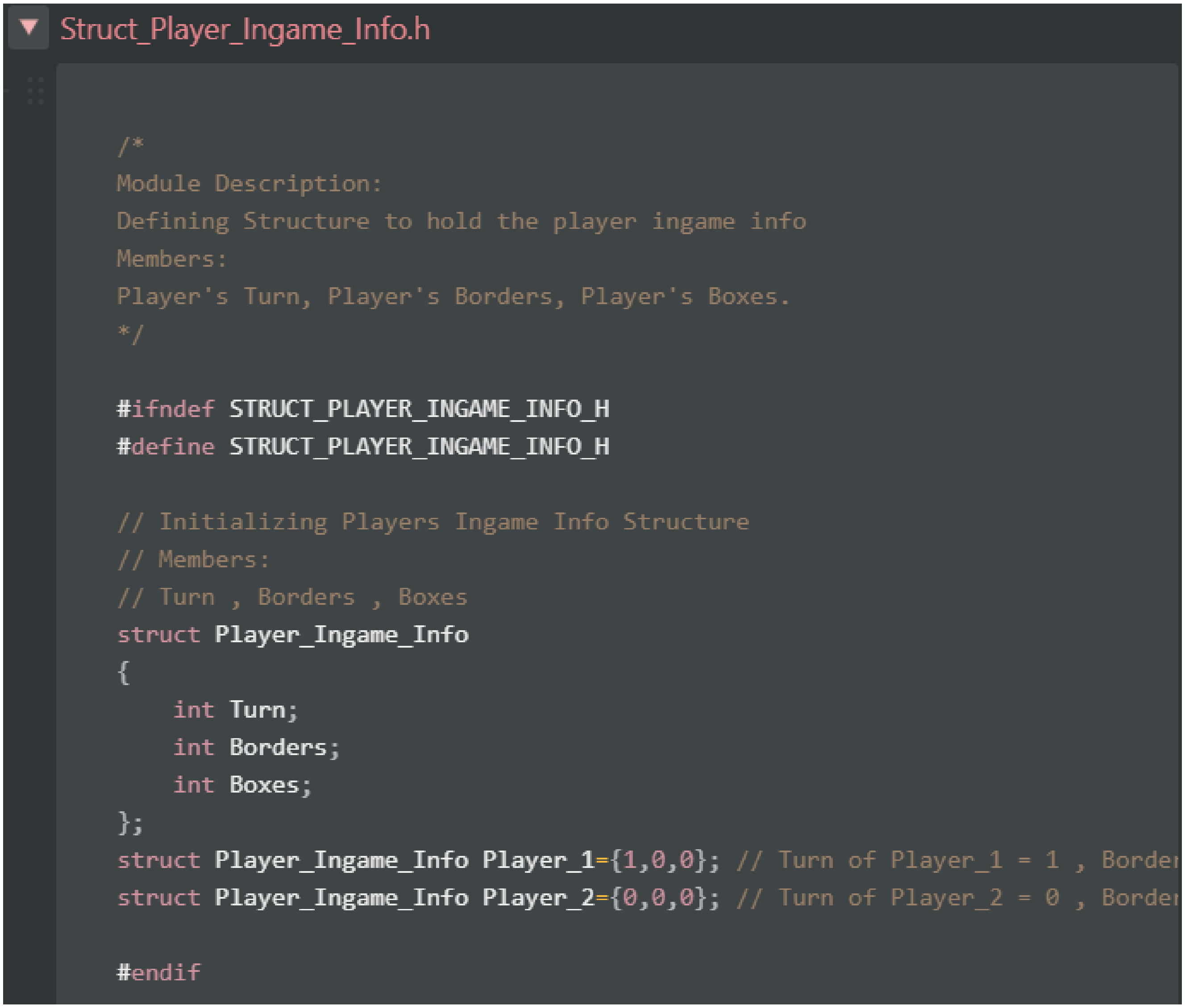
** Turn*

** Borders*

** Boxes*

It is clear that this structure is assigned for each player (A and B), identifying the turn and borders and boxes of each of them.

The below SS is from Notion.



Struct_Player_Ingame_Info.h

```c
/*
Module Description:
Defining Structure to hold the player ingame info
Members:
Player's Turn, Player's Borders, Player's Boxes.
*/

#ifndef STRUCT_PLAYER_INGAME_INFO_H
#define STRUCT_PLAYER_INGAME_INFO_H

// Initializing Players Ingame Info Structure
// Members:
// Turn , Borders , Boxes
struct Player_Ingame_Info
{
    int Turn;
    int Borders;
    int Boxes;
};
struct Player_Ingame_Info Player_1={1,0,0}; // Turn of Player_1 = 1 , Borde
struct Player_Ingame_Info Player_2={0,0,0}; // Turn of Player_2 = 0 , Borde

#endif
```

2] Game_Info_Structure

Members:

** Remaining_Borders*

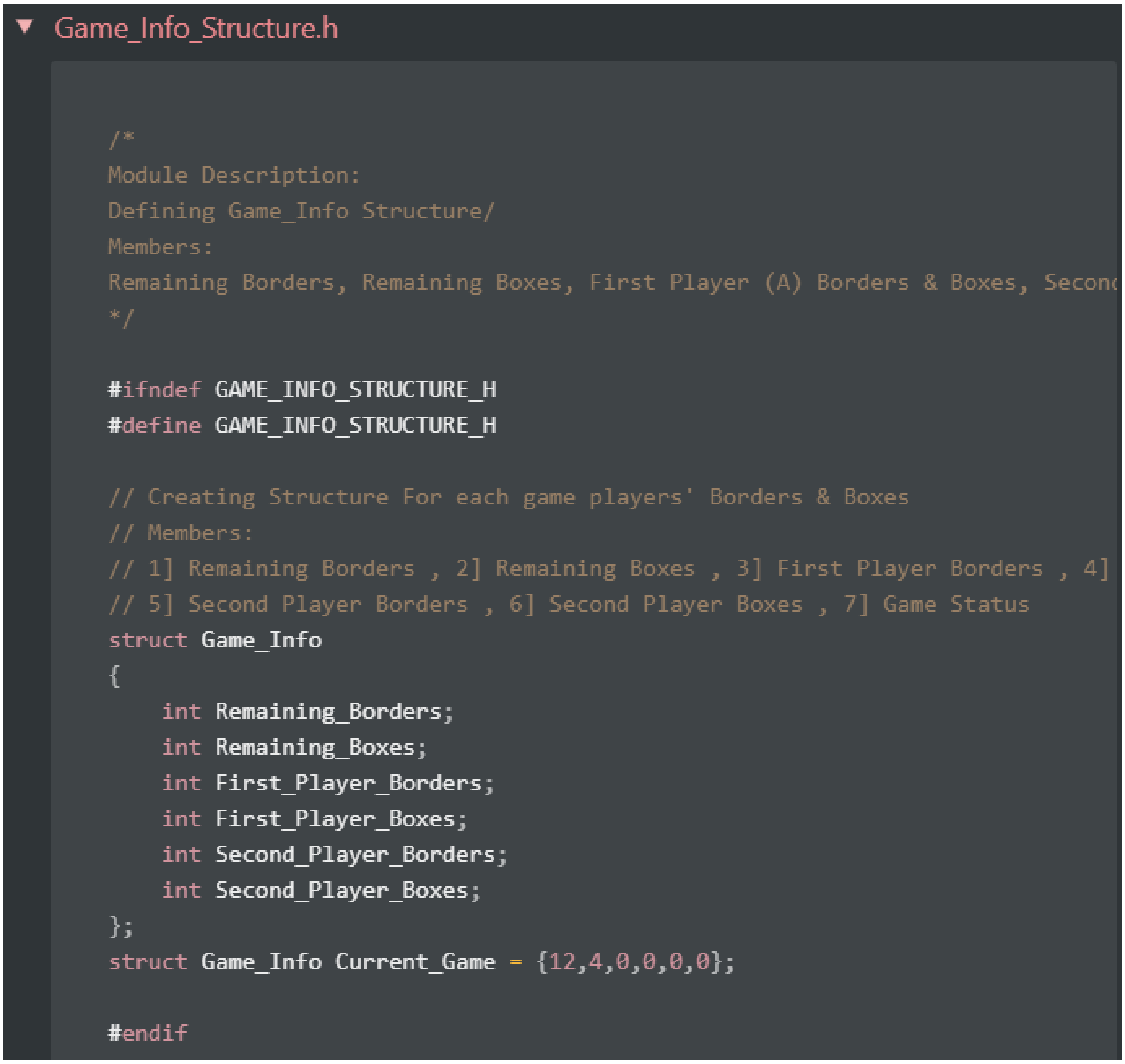
** Remaining_Boxes*

** First_Player_Borders*

** First_Player_Boxes*

** Second_Player_Borders*

** Second_Player_Boxes*



Game_Info_Structure.h

```
/*
Module Description:
Defining Game_Info Structure/
Members:
Remaining Borders, Remaining Boxes, First Player (A) Borders & Boxes, Secon
*/

#ifndef GAME_INFO_STRUCTURE_H
#define GAME_INFO_STRUCTURE_H

// Creating Structure For each game players' Borders & Boxes
// Members:
// 1] Remaining Borders , 2] Remaining Boxes , 3] First Player Borders , 4]
// 5] Second Player Borders , 6] Second Player Boxes , 7] Game Status
struct Game_Info
{
    int Remaining_Borders;
    int Remaining_Boxes;
    int First_Player_Borders;
    int First_Player_Boxes;
    int Second_Player_Borders;
    int Second_Player_Boxes;
};
struct Game_Info Current_Game = {12,4,0,0,0,0};

#endif
```

3] U_R_B_Game_Status_Player_Info Structure

U_R_B Stands for: Undo_Redo_Beginner

Members:

** Remaining_Borders*

** Remaining_Boxes*

** FP_Borders (First Player)*

** FP_Boxes*

** SP_Borders (Second Player)*

** SP_Boxes*

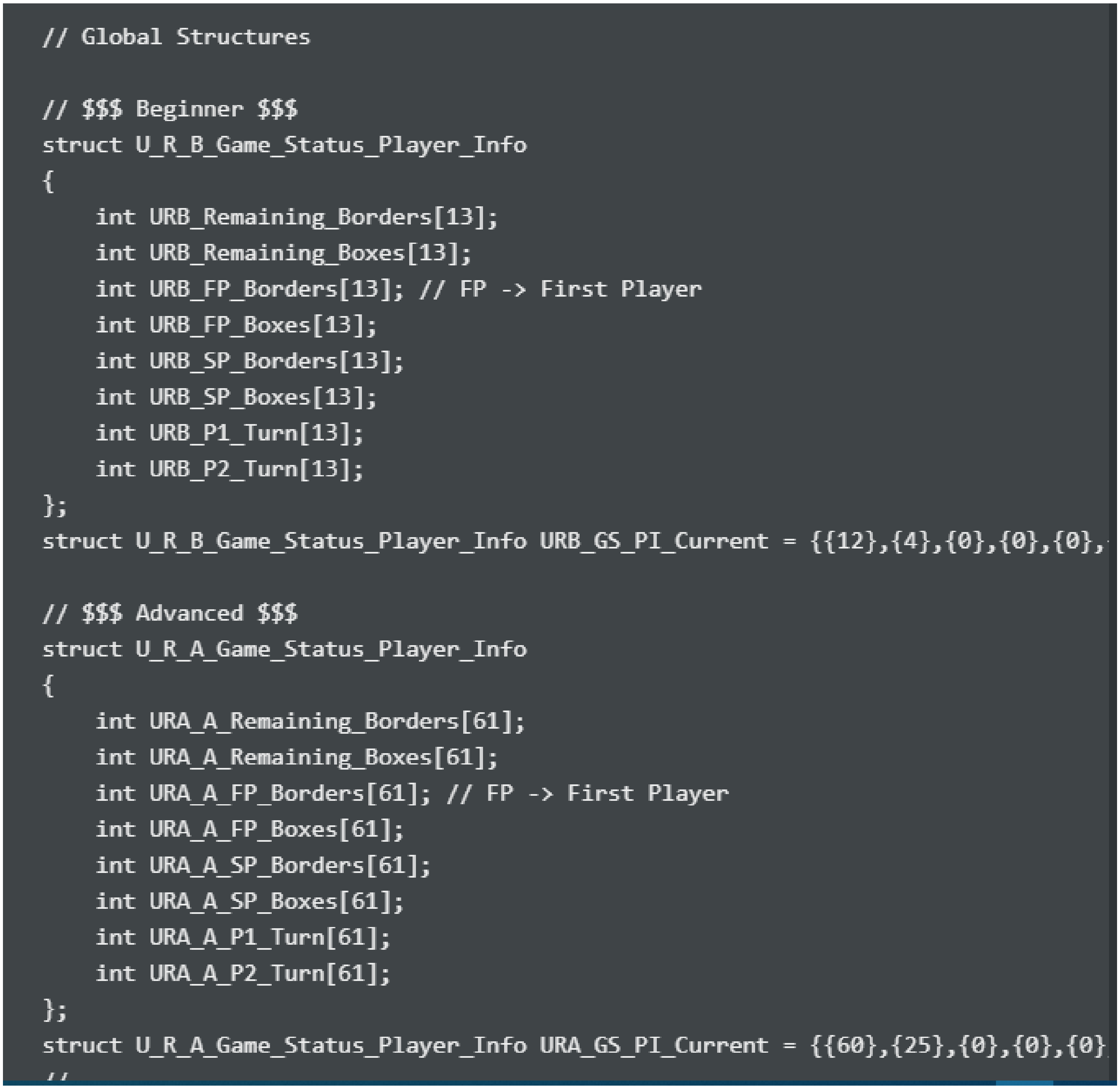
** P1_Turn (Player 1)*

** P2_Turn*

Notice that, For the Undo/Redo Module, our idea for it was to initialize a 3D Array that we can imagine as all of the game boards successive and parallel to each other, the 3rd dimension is the number of borders, so we can visualize it by saying that we are taking a screenshot of the game after each border placed, and when we request undo,redo we copy the game_data from screenshot corresponding to the game layer before(undo) / After(redo).

Note that same structure has been defined for the Advanced mode, with name ***"U_R_A_Game_Status_Player_Info"***

U_R_A -> Undo_Redo_Advanced.

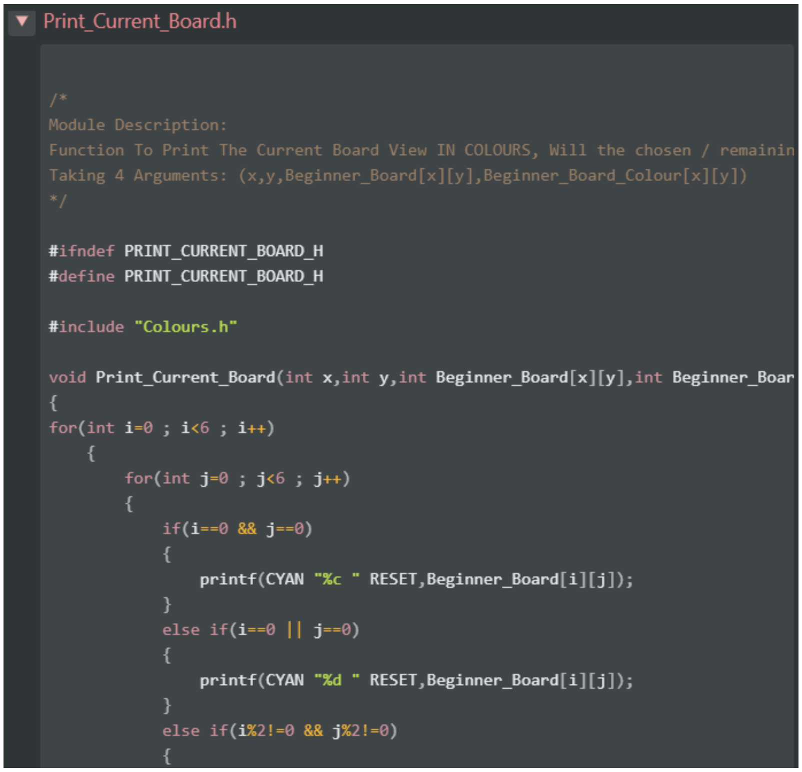


```
// Global Structures

// $$$ Beginner $$$
struct U_R_B_Game_Status_Player_Info
{
    int URB_Remaining_Borders[13];
    int URB_Remaining_Boxes[13];
    int URB_FP_Borders[13]; // FP -> First Player
    int URB_FP_Boxes[13];
    int URB_SP_Borders[13];
    int URB_SP_Boxes[13];
    int URB_P1_Turn[13];
    int URB_P2_Turn[13];
};
struct U_R_B_Game_Status_Player_Info URB_GS_PI_Current = {{12},{4},{0},{0},{0},

// $$$ Advanced $$$
struct U_R_A_Game_Status_Player_Info
{
    int URA_A_Remaining_Borders[61];
    int URA_A_Remaining_Boxes[61];
    int URA_A_FP_Borders[61]; // FP -> First Player
    int URA_A_FP_Boxes[61];
    int URA_A_SP_Borders[61];
    int URA_A_SP_Boxes[61];
    int URA_A_P1_Turn[61];
    int URA_A_P2_Turn[61];
};
struct U_R_A_Game_Status_Player_Info URA_GS_PI_Current = {{60},{25},{0},{0},{0}
//
```

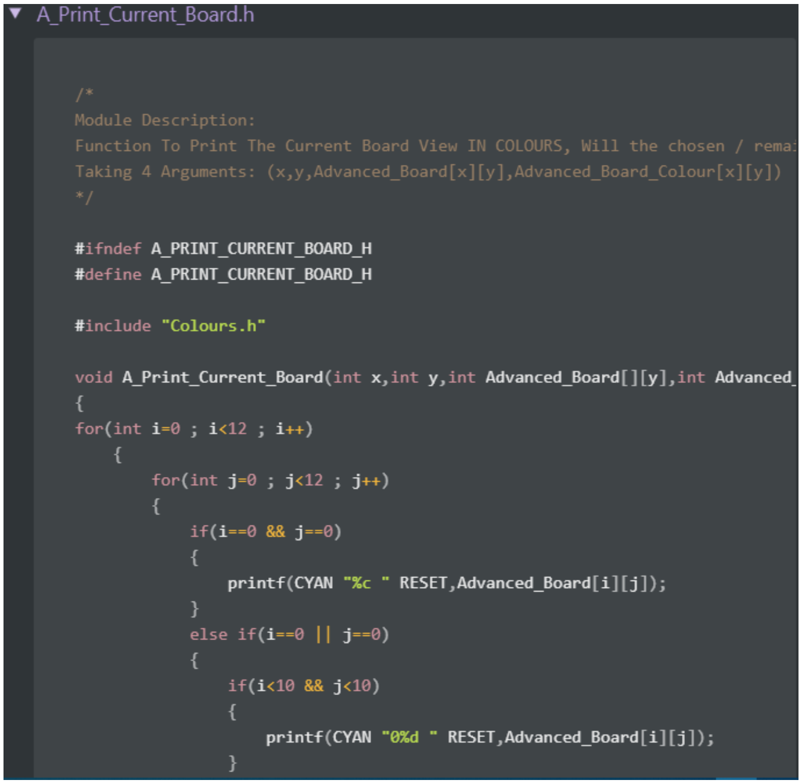
6) Description of the important functions/modules:

- *RECALL: ALL OF THE BELOW FUNCTIONS ARE DEFINED FOR THE BEGINNER MODE, AND OTHER FUNCTIONS WITH THE SAME LOGIC HAVE BEEN BUILT FOR THE ADVANCED MODE Also Note that: All the modules are available on Notion and github in the game folder.*
  About the important Functions/Modules that we have used, here are some of the important used functions/modules

1] Print_Current_Board.h
This Module defines a function having the same name, used to print the current board in the beginner mode after applying the last played move and closing any new boxes (if created).



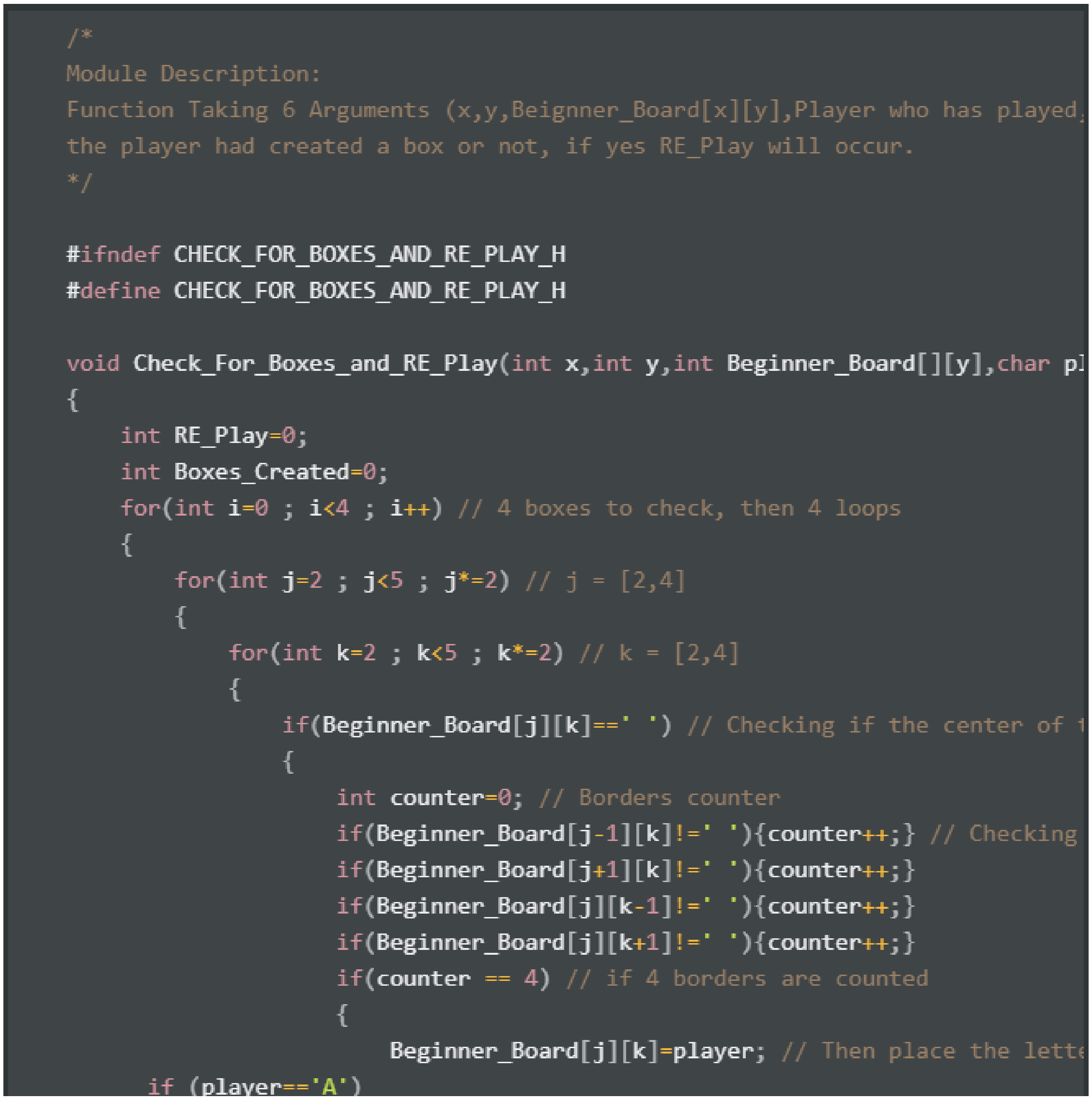
```
/*
Module Description:
Function To Print The Current Board View IN COLOURS, Will the chosen / remainin
Taking 4 Arguments: (x,y,Beginner_Board[x][y],Beginner_Board_Colour[x][y])
*/

#ifndef PRINT_CURRENT_BOARD_H
#define PRINT_CURRENT_BOARD_H

#include "Colours.h"

void Print_Current_Board(int x,int y,int Beginner_Board[x][y],int Beginner_Boar
{
for(int i=0 ; i<6 ; i++)
    {
        for(int j=0 ; j<6 ; j++)
        {
            if(i==0 && j==0)
            {
                printf(CYAN "%c " RESET,Beginner_Board[i][j]);
            }
            else if(i==0 || j==0)
            {
                printf(CYAN "%d " RESET,Beginner_Board[i][j]);
            }
            else if(i%2!=0 && j%2!=0)
            {
```



```
/*
Module Description:
Function To Print The Current Board View IN COLOURS, Will the chosen / rema
Taking 4 Arguments: (x,y,Advanced_Board[x][y],Advanced_Board_Colour[x][y])
*/

#ifndef A_PRINT_CURRENT_BOARD_H
#define A_PRINT_CURRENT_BOARD_H

#include "Colours.h"

void A_Print_Current_Board(int x,int y,int Advanced_Board[][y],int Advanced_
{
for(int i=0 ; i<12 ; i++)
    {
        for(int j=0 ; j<12 ; j++)
        {
            if(i==0 && j==0)
            {
                printf(CYAN "%c " RESET,Advanced_Board[i][j]);
            }
            else if(i==0 || j==0)
            {
                if(i<10 && j<10)
                {
                    printf(CYAN "0%d " RESET,Advanced_Board[i][j]);
                }
```

2] Check_For_Boxes_and_RE_Play.h

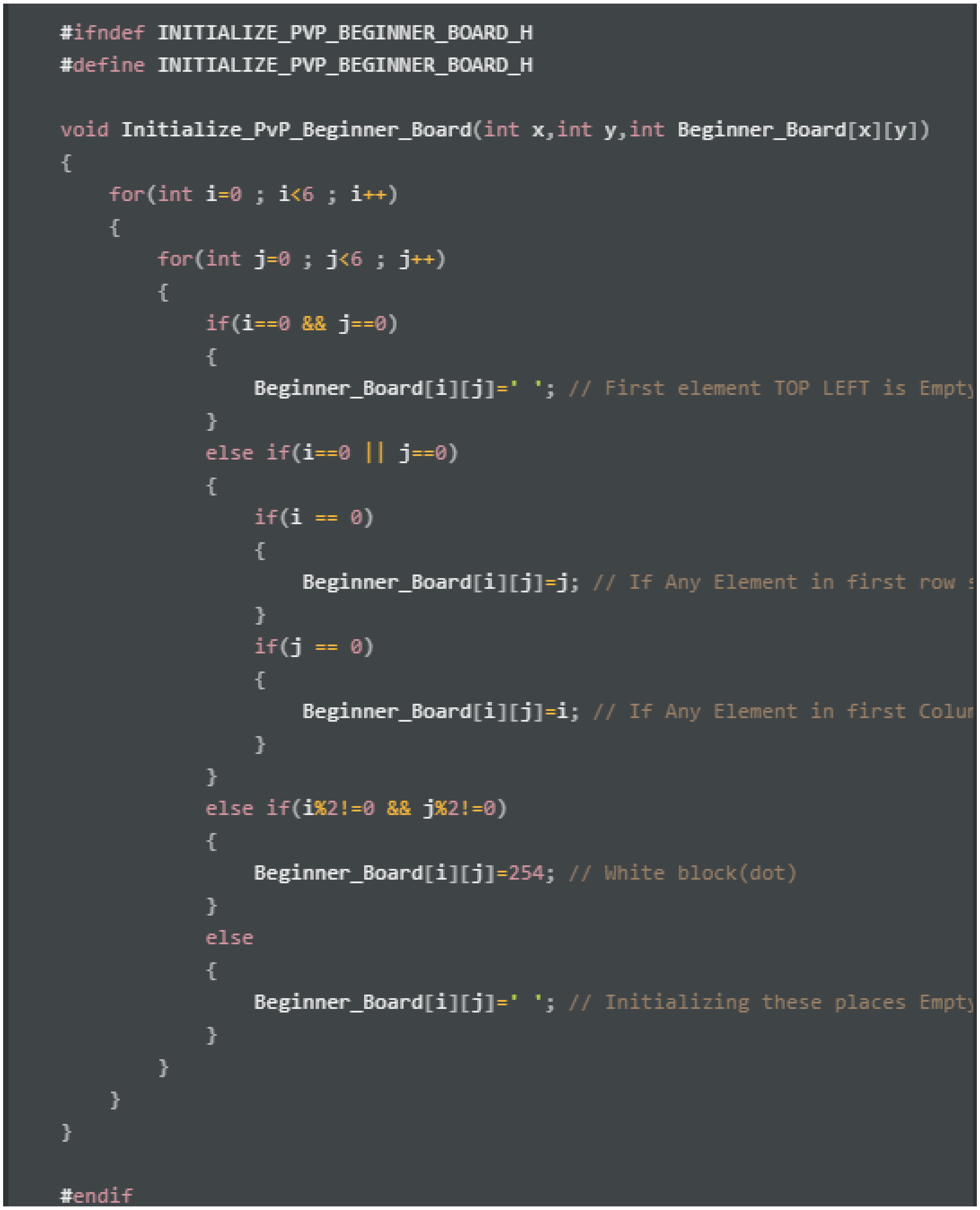
This module is created to define a function that checks all he board boxes and decide whether a new box has been closed or not, so as if a player has closed a box, it returns a specific value indicating the number of boxes created by this move, and the re-directing the game-play turn to the same player that closed the box.



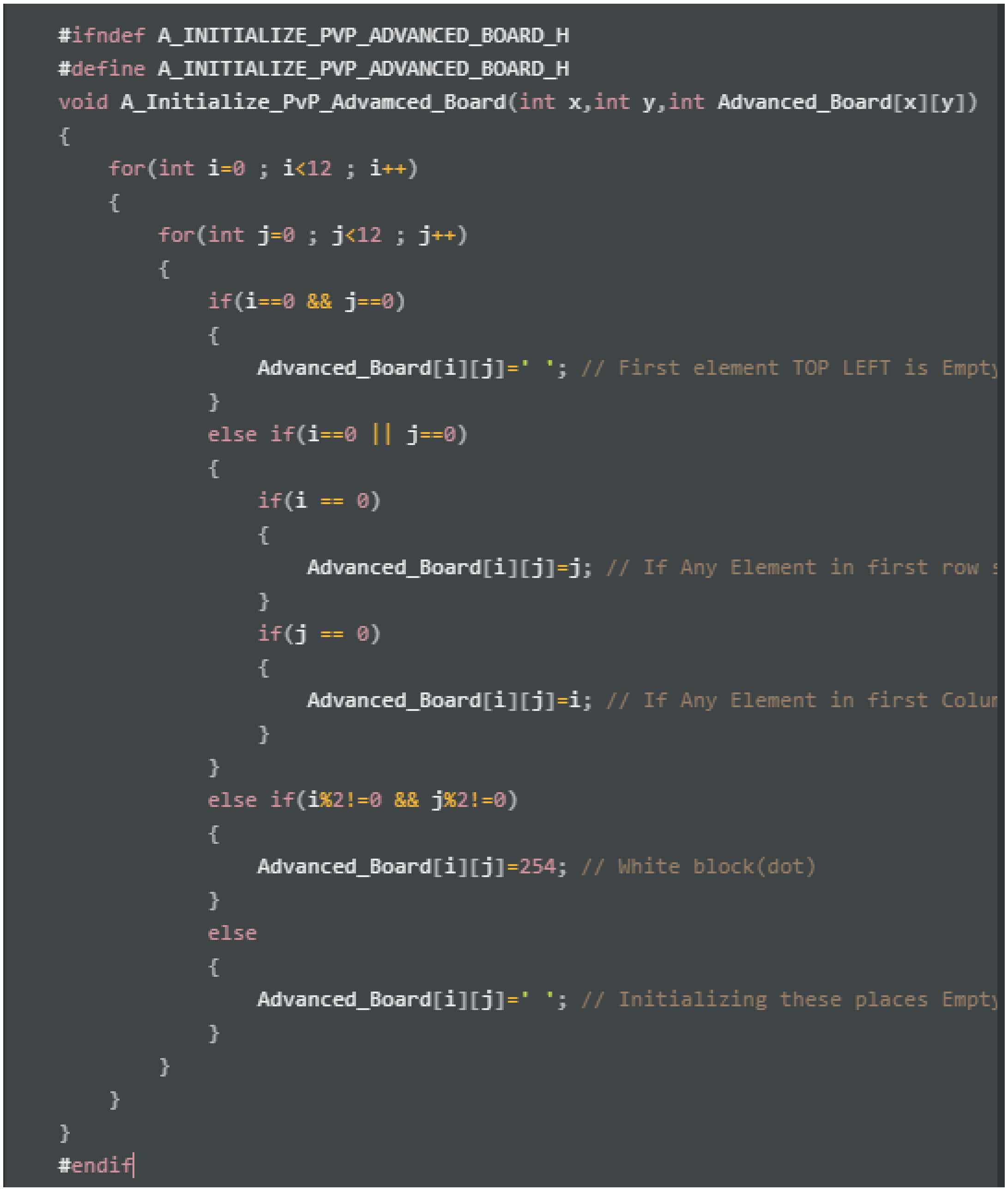
```
/*
Module Description:
Function Taking 6 Arguments (x,y,Beignner_Board[x][y],Player who has played,
the player had created a box or not, if yes RE_Play will occur.
*/

#ifndef CHECK_FOR_BOXES_AND_RE_PLAY_H
#define CHECK_FOR_BOXES_AND_RE_PLAY_H

void Check_For_Boxes_and_RE_Play(int x,int y,int Beginner_Board[][y],char pl
{
    int RE_Play=0;
    int Boxes_Created=0;
    for(int i=0 ; i<4 ; i++) // 4 boxes to check, then 4 loops
    {
        for(int j=2 ; j<5 ; j*=2) // j = [2,4]
        {
            for(int k=2 ; k<5 ; k*=2) // k = [2,4]
            {
                if(Beginner_Board[j][k]==' ') // Checking if the center of t
                {
                    int counter=0; // Borders counter
                    if(Beginner_Board[j-1][k]!=' '){counter++;} // Checking
                    if(Beginner_Board[j+1][k]!=' '){counter++;}
                    if(Beginner_Board[j][k-1]!=' '){counter++;}
                    if(Beginner_Board[j][k+1]!=' '){counter++;}
                    if(counter == 4) // if 4 borders are counted
                    {
                        Beginner_Board[j][k]=player; // Then place the lette
      if (player=='A')
```

3] Initialize_PvP_Beginner_Board.h

This Module defines a function with the same name, The functions initialize each element in the array to its specific value, whether it is an empty space or a DOT or a numbering column/row.



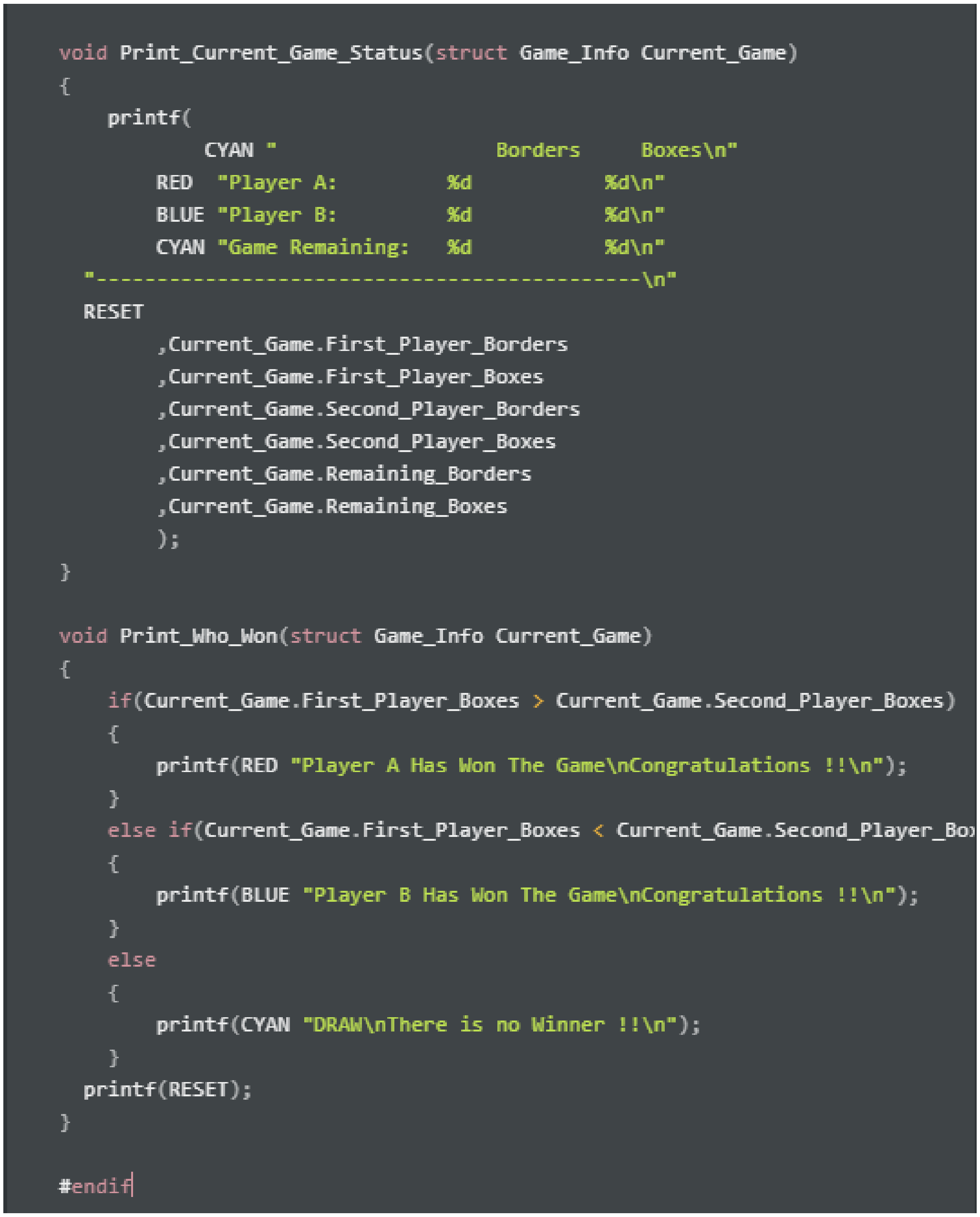
```
#ifndef INITIALIZE_PVP_BEGINNER_BOARD_H
#define INITIALIZE_PVP_BEGINNER_BOARD_H

void Initialize_PvP_Beginner_Board(int x,int y,int Beginner_Board[x][y])
{
    for(int i=0 ; i<6 ; i++)
    {
        for(int j=0 ; j<6 ; j++)
        {
            if(i==0 && j==0)
            {
                Beginner_Board[i][j]=' '; // First element TOP LEFT is Empty
            }
            else if(i==0 || j==0)
            {
                if(i == 0)
                {
                    Beginner_Board[i][j]=j; // If Any Element in first row s
                }
                if(j == 0)
                {
                    Beginner_Board[i][j]=i; // If Any Element in first Colu
                }
            }
            else if(i%2!=0 && j%2!=0)
            {
                Beginner_Board[i][j]=254; // White block(dot)
            }
            else
            {
                Beginner_Board[i][j]=' '; // Initializing these places Empty
            }
        }
    }
}

#endif
```



```
#ifndef A_INITIALIZE_PVP_ADVANCED_BOARD_H
#define A_INITIALIZE_PVP_ADVANCED_BOARD_H
void A_Initialize_PvP_Advamced_Board(int x,int y,int Advanced_Board[x][y])
{
    for(int i=0 ; i<12 ; i++)
    {
        for(int j=0 ; j<12 ; j++)
        {
            if(i==0 && j==0)
            {
                Advanced_Board[i][j]=' '; // First element TOP LEFT is Empty
            }
            else if(i==0 || j==0)
            {
                if(i == 0)
                {
                    Advanced_Board[i][j]=j; // If Any Element in first row
                }
                if(j == 0)
                {
                    Advanced_Board[i][j]=i; // If Any Element in first Colu
                }
            }
            else if(i%2!=0 && j%2!=0)
            {
                Advanced_Board[i][j]=254; // White block(dot)
            }
            else
            {
                Advanced_Board[i][j]=' '; // Initializing these places Empty
            }
        }
    }
}
#endif
```

4] Print_Current_Game_Status.h

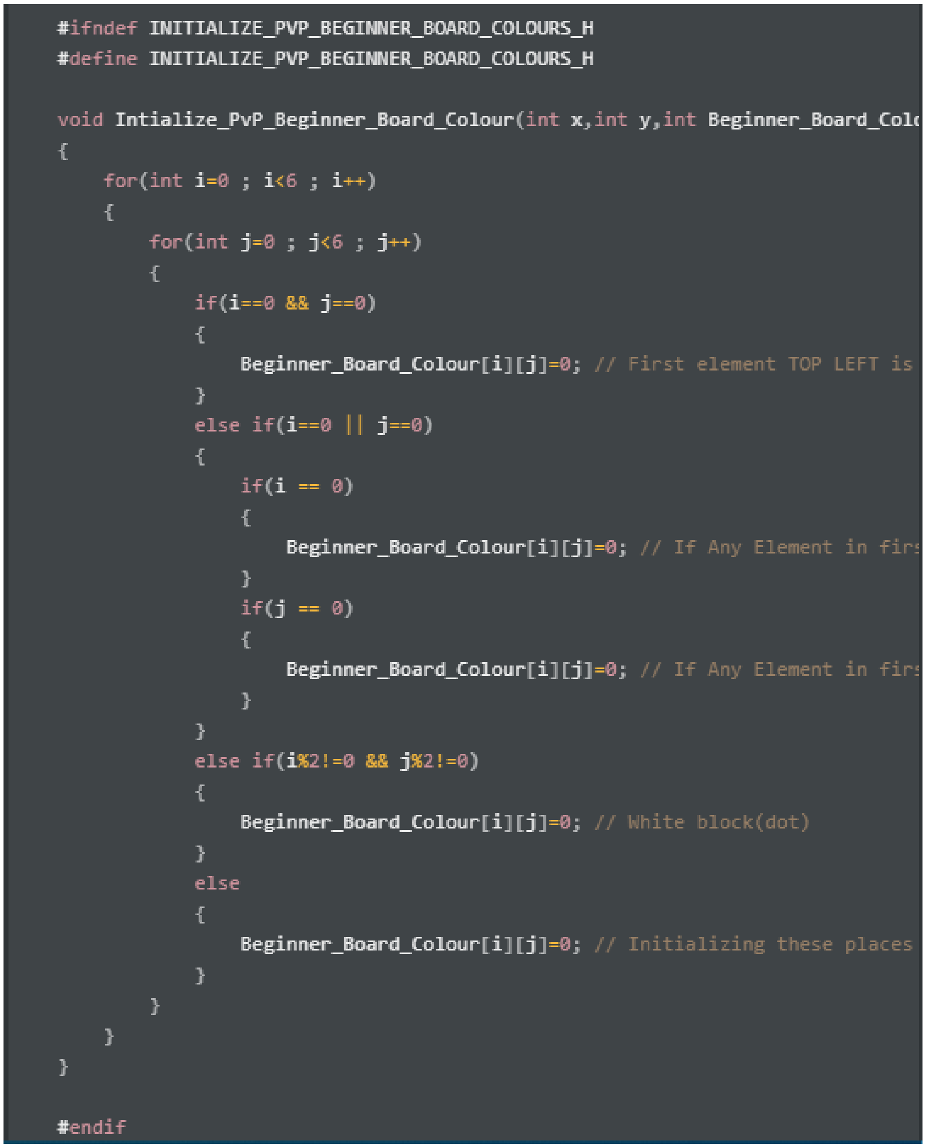
This Module defines a function with the same name, that takes the Game_info Structure as a parameter and displays the game status in a decent and arranged way. also defines a function that prints the game winner, the game info structure as a parameter.



```
void Print_Current_Game_Status(struct Game_Info Current_Game)
{
    printf(
            CYAN "                   Borders     Boxes\n"
        RED  "Player A:         %d           %d\n"
        BLUE "Player B:         %d           %d\n"
        CYAN "Game Remaining:   %d           %d\n"
  "---------------------------------------------\n"
  RESET
        ,Current_Game.First_Player_Borders
        ,Current_Game.First_Player_Boxes
        ,Current_Game.Second_Player_Borders
        ,Current_Game.Second_Player_Boxes
        ,Current_Game.Remaining_Borders
        ,Current_Game.Remaining_Boxes
        );
}

void Print_Who_Won(struct Game_Info Current_Game)
{
    if(Current_Game.First_Player_Boxes > Current_Game.Second_Player_Boxes)
    {
        printf(RED "Player A Has Won The Game\nCongratulations !!\n");
    }
    else if(Current_Game.First_Player_Boxes < Current_Game.Second_Player_Box
    {
        printf(BLUE "Player B Has Won The Game\nCongratulations !!\n");
    }
    else
    {
        printf(CYAN "DRAW\nThere is no Winner !!\n");
    }
  printf(RESET);
}

#endif
```

5] Initialize_PvP_Beginner_Board_Colours.h

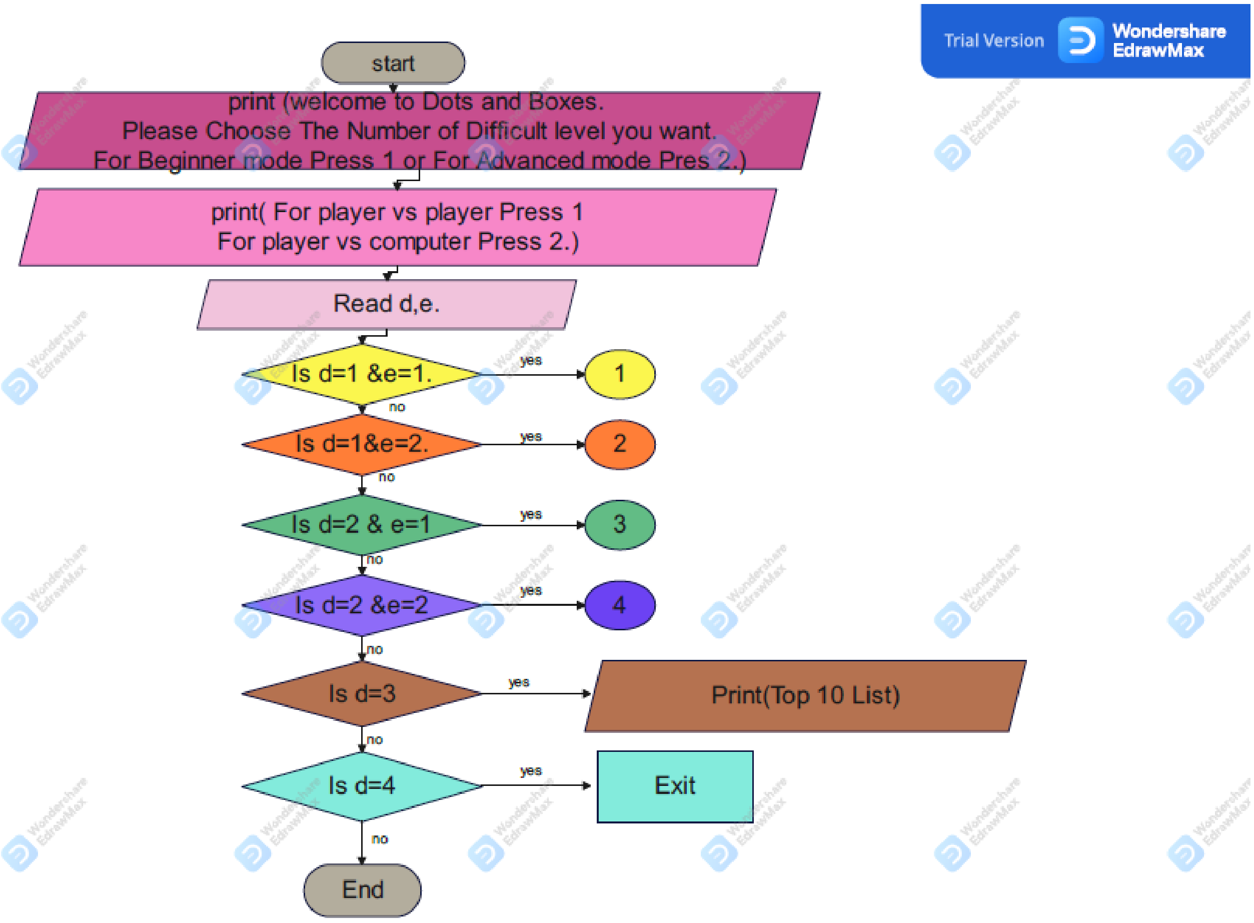
Similar to number 3, This function initializes the colour board, in which player A is given a certain colour (RED) and player B is given a certain colour (BLUE) and the game elements that are already present like the dots and numbering rows and columns are given (GREEN) as mentioned before.



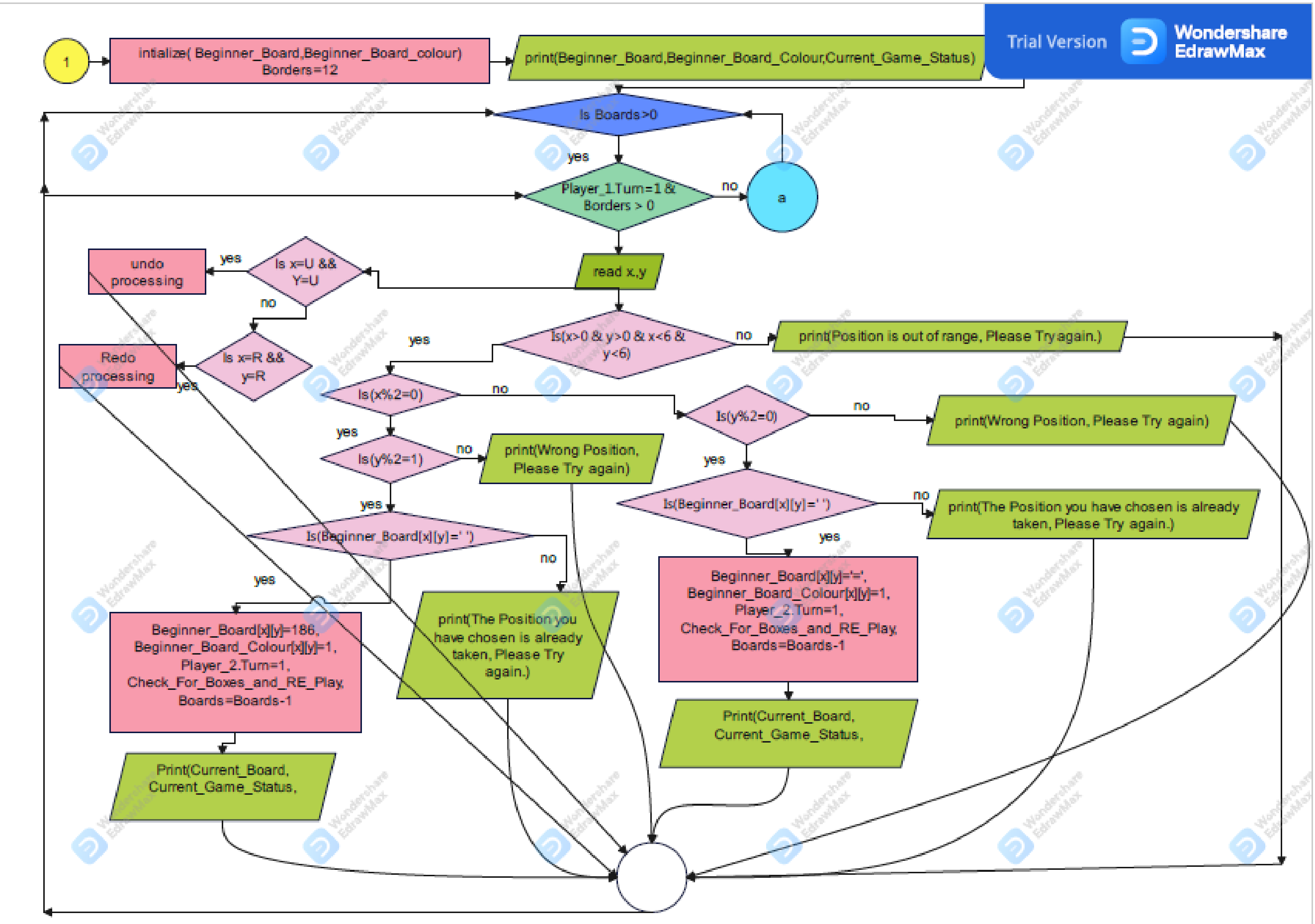
```
#ifndef INITIALIZE_PVP_BEGINNER_BOARD_COLOURS_H
#define INITIALIZE_PVP_BEGINNER_BOARD_COLOURS_H

void Intialize_PvP_Beginner_Board_Colour(int x,int y,int Beginner_Board_Colo
{
    for(int i=0 ; i<6 ; i++)
    {
        for(int j=0 ; j<6 ; j++)
        {
            if(i==0 && j==0)
            {
                Beginner_Board_Colour[i][j]=0; // First element TOP LEFT is
            }
            else if(i==0 || j==0)
            {
                if(i == 0)
                {
                    Beginner_Board_Colour[i][j]=0; // If Any Element in firs
                }
                if(j == 0)
                {
                    Beginner_Board_Colour[i][j]=0; // If Any Element in firs
                }
            }
            else if(i%2!=0 && j%2!=0)
            {
                Beginner_Board_Colour[i][j]=0; // White block(dot)
            }
            else
            {
                Beginner_Board_Colour[i][j]=0; // Initializing these places
            }
        }
    }
}

#endif
```

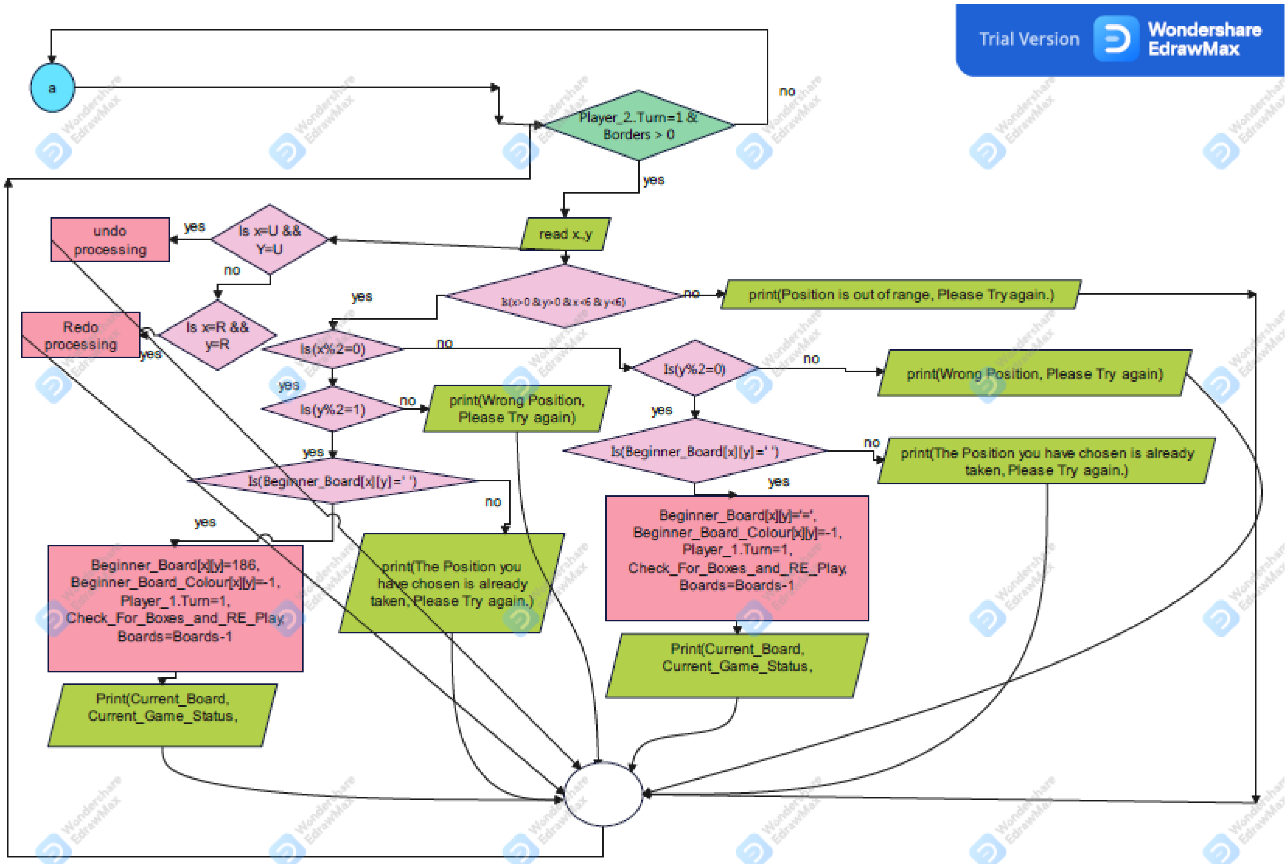
7) Flow chart and pseudo code

- ***We make Flowchart For Main (Game) by using Wondershare EdrawMax Program.***
- ***For a better Quality and high resolution, The Flowchart File will be included as pdf in the Documentation Folder.***

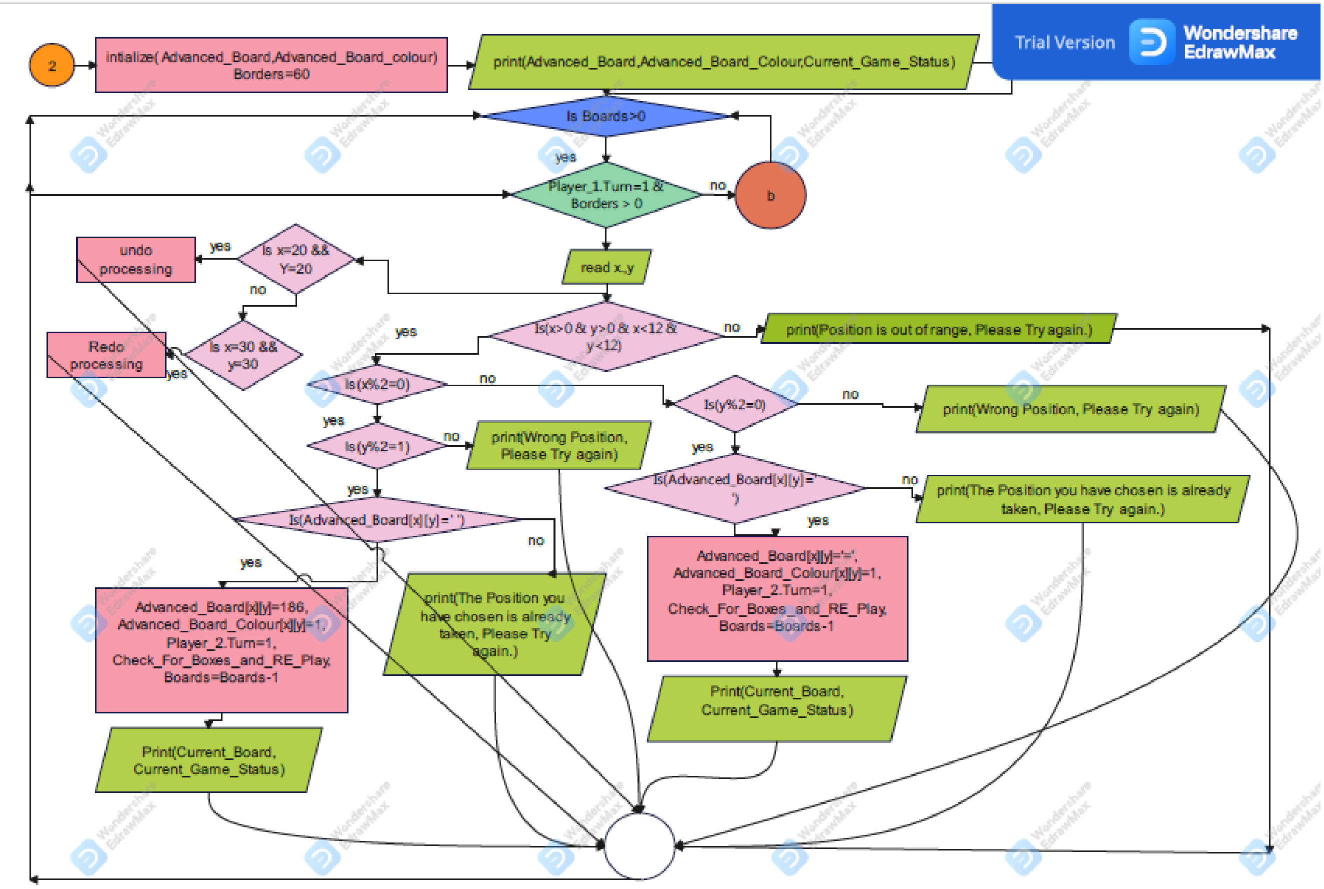

Trial Version
Wondershare EdrawMax
start
print (welcome to Dots and Boxes.
Please Choose The Number of Difficult level you want.
For Beginner mode Press 1 or For Advanced mode Pres 2.)
print(For player vs player Press 1
For player vs computer Press 2.)
Read d,e.
Is d=1 &e=1.
yes
1
no
Is d=1&e=2.
yes
2
no
Is d=2 & e=1
yes
3
no
Is d=2 &e=2
yes
4
no
Is d=3
yes
Print(Top 10 List)
no
Is d=4
yes
Exit
no
End

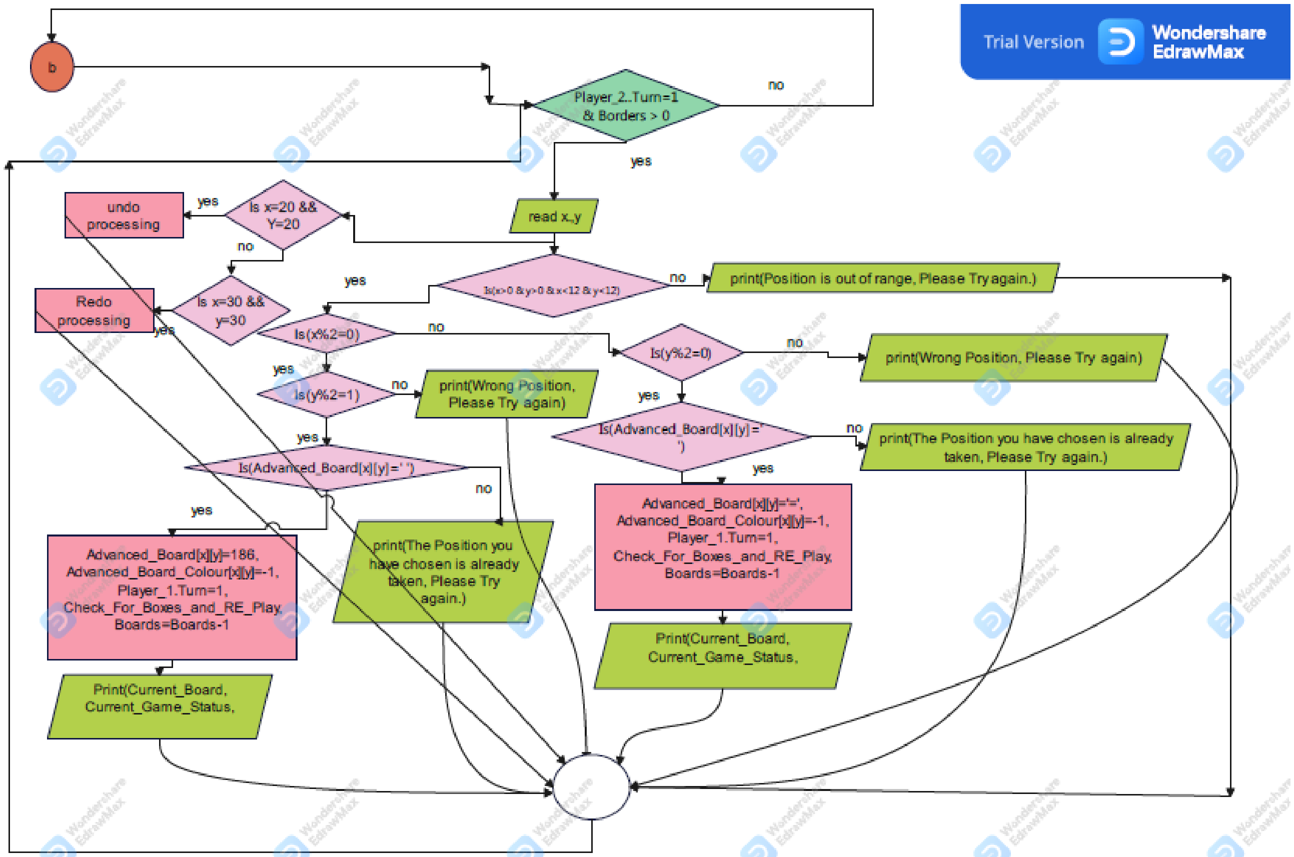
●


1
intialize(Beginner_Board,Beginner_Board_colour)
Borders=12
print(Beginner_Board,Beginner_Board_Colour,Current_Game_Status)
Is Boards>0
yes
Player_1.Turn=1 &
Borders > 0
no
a
read x,y
undo
processing
yes
Is x=U &&
Y=U
no
Redo
processing
yes
Is x=R &&
y=R
Is(x>0 & y>0 & x<6 &
y<6)
no
print(Position is out of range, Please Try again.)
yes
Is(x%2=0)
no
Is(y%2=0)
no
print(Wrong Position, Please Try again)
yes
Is(y%2=1)
no
print(Wrong Position,
Please Try again)
yes
Is(Beginner_Board[x][y]=' ')
no
print(The Position you have chosen is already
taken, Please Try again.)
yes
Beginner_Board[x][y]='=',
Beginner_Board_Colour[x][y]=1,
Player_2.Turn=1,
Check_For_Boxes_and_RE_Play,
Boards=Boards-1
Print(Current_Board,
Current_Game_Status,
yes
Is(Beginner_Board[x][y]=' ')
no
yes
Beginner_Board[x][y]=186,
Beginner_Board_Colour[x][y]=1,
Player_2.Turn=1,
Check_For_Boxes_and_RE_Play,
Boards=Boards-1
print(The Position you
have chosen is already
taken, Please Try
again.)
Print(Current_Board,
Current_Game_Status,

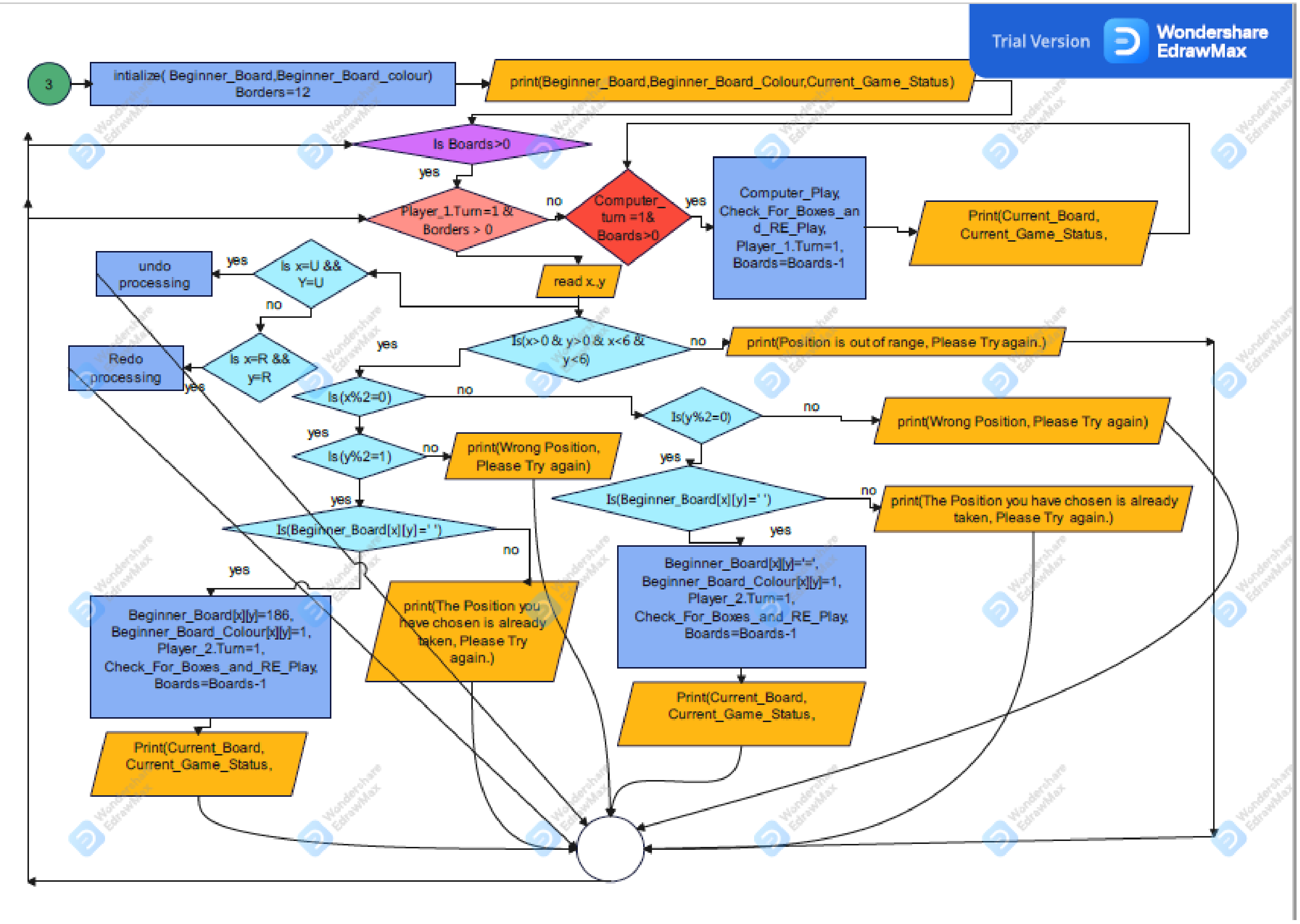

a
Player_2_Turn=1 &
Borders > 0
no
yes
read x,y
undo
processing
yes
Is x=U &&
Y=U
no
Is(x>0 & y>0 & x<6 & y<6)
no
print(Position is out of range, Please Try again.)
yes
Redo
processing
yes
Is x=R &&
y=R
Is(x%2=0)
no
Is(y%2=0)
no
print(Wrong Position, Please Try again)
yes
Is(y%2=1)
no
print(Wrong Position,
Please Try again)
yes
Is(Beginner_Board[x][y]=' ')
no
print(The Position you have chosen is already
taken, Please Try again.)
yes
yes
Is(Beginner_Board[x][y]=' ')
no
Beginner_Board[x][y]='=',
Beginner_Board_Colour[x][y]=-1,
Player_1_Turn=1,
Check_For_Boxes_and_RE_Play,
Boards=Boards-1
yes
Beginner_Board[x][y]=186,
Beginner_Board_Colour[x][y]=-1,
Player_1_Turn=1,
Check_For_Boxes_and_RE_Play,
Boards=Boards-1
print(The Position you
have chosen is already
taken, Please Try again.)
Print(Current_Board,
Current_Game_Status,
Print(Current_Board,
Current_Game_Status,

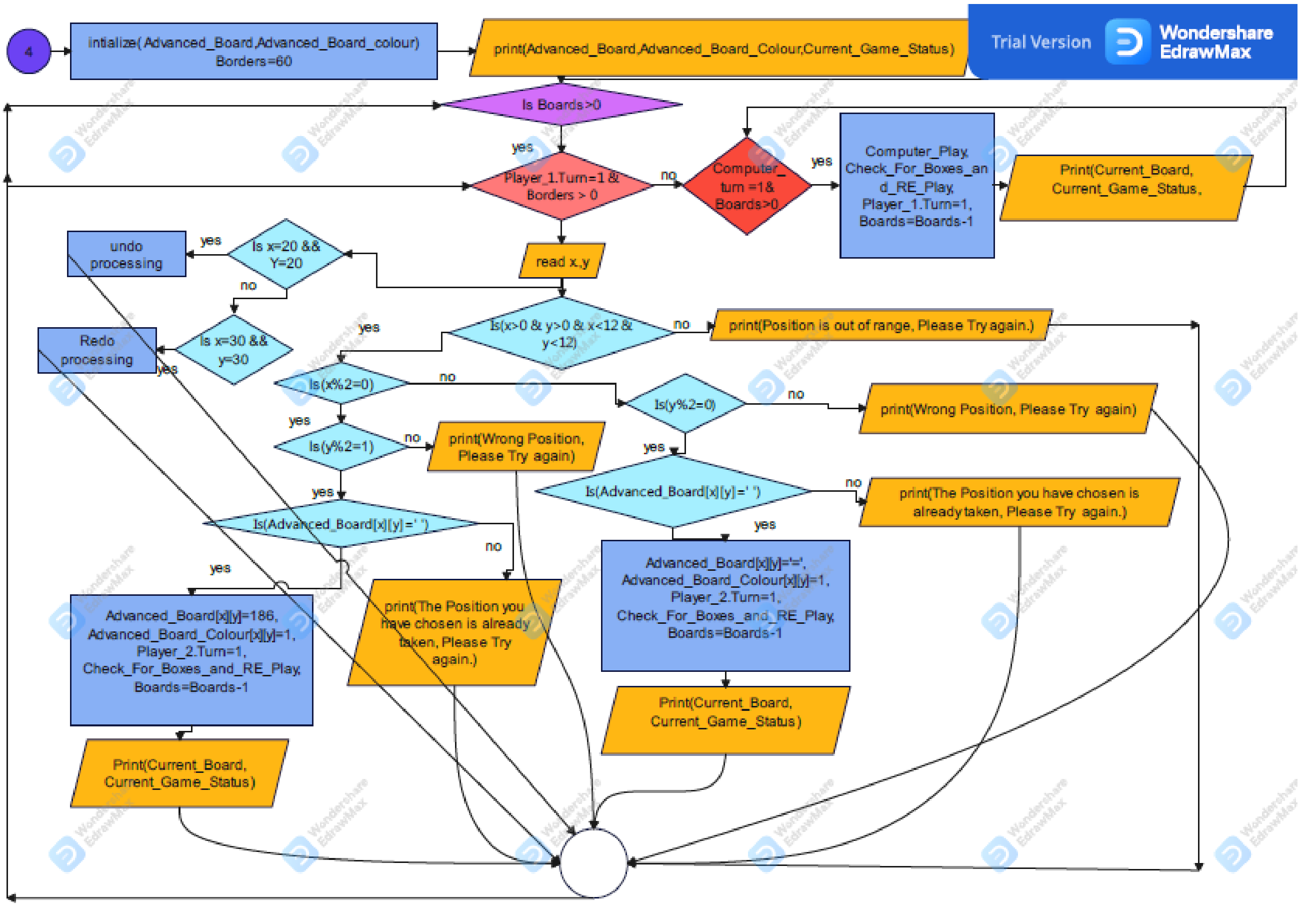
-


2
intialize(Advanced_Board,Advanced_Board_colour)
Borders=60
print(Advanced_Board,Advanced_Board_Colour,Current_Game_Status)
Is Boards>0
yes
Player_1.Turn=1 &
Borders > 0
no
b
read x,y
undo
processing
yes
Is x=20 &&
Y=20
no
Redo
processing
yes
Is x=30 &&
y=30
yes
Is(x>0 & y>0 & x<12 &
y<12)
no
print(Position is out of range, Please Try again.)
Is(x%2=0)
no
yes
Is(y%2=1)
no
print(Wrong Position,
Please Try again)
yes
Is(Advanced_Board[x][y]=' ')
no
yes
Advanced_Board[x][y]=186,
Advanced_Board_Colour[x][y]=1,
Player_2.Turn=1,
Check_For_Boxes_and_RE_Play,
Boards=Boards-1
Print(Current_Board,
Current_Game_Status)
print(The Position you
have chosen is already
taken, Please Try
again.)
Is(y%2=0)
no
print(Wrong Position, Please Try again)
yes
Is(Advanced_Board[x][y]='
')
no
print(The Position you have chosen is already
taken, Please Try again.)
yes
Advanced_Board[x][y]='=',
Advanced_Board_Colour[x][y]=1,
Player_2.Turn=1,
Check_For_Boxes_and_RE_Play,
Boards=Boards-1
Print(Current_Board,
Current_Game_Status)


b
Player_2..Turn=1
& Borders > 0
no
yes
read x,y
undo
processing
yes
Is x=20 &&
Y=20
no
Redo
processing
yes
Is x=30 &&
y=30
yes
Is(x>0 & y>0 & x<12 & y<12)
no
print(Position is out of range, Please Try again.)
Is(x%2=0)
no
yes
Is(y%2=1)
no
print(Wrong Position,
Please Try again)
yes
Is(Advanced_Board[x][y]=' ')
no
yes
Advanced_Board[x][y]=186,
Advanced_Board_Colour[x][y]=-1,
Player_1.Turn=1,
Check_For_Boxes_and_RE_Play,
Boards=Boards-1
print(The Position you
have chosen is already
taken, Please Try
again.)
Print(Current_Board,
Current_Game_Status,
Is(y%2=0)
no
print(Wrong Position, Please Try again)
yes
Is(Advanced_Board[x][y]='
')
no
print(The Position you have chosen is already
taken, Please Try again.)
yes
Advanced_Board[x][y]='=',
Advanced_Board_Colour[x][y]=-1,
Player_1.Turn=1,
Check_For_Boxes_and_RE_Play,
Boards=Boards-1
Print(Current_Board,
Current_Game_Status,

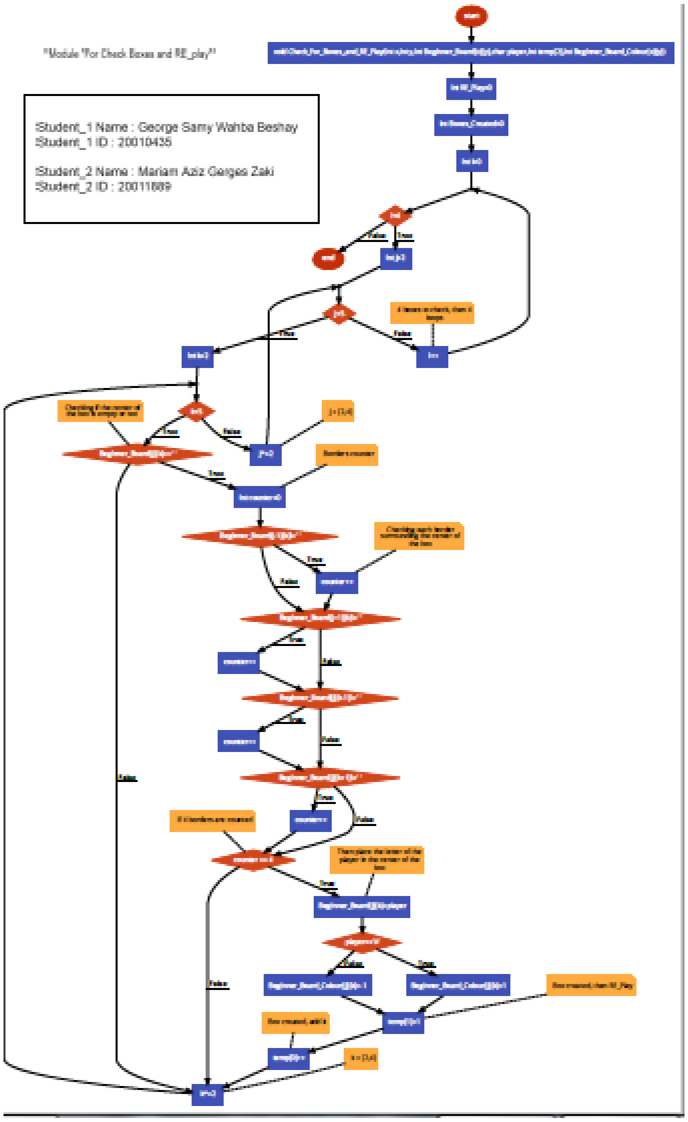
-


3
intialize(Beginner_Board,Beginner_Board_colour)
Borders=12
print(Beginner_Board,Beginner_Board_Colour,Current_Game_Status)
Is Boards>0
yes
Player_1.Turn=1 &
Borders > 0
no
Computer_
turn =1&
Boards>0
yes
Computer_Play,
Check_For_Boxes_an
d_RE_Play,
Player_1.Turn=1,
Boards=Boards-1
Print(Current_Board,
Current_Game_Status,
undo
processing
yes
Is x=U &&
Y=U
no
read x,y
Redo
processing
yes
Is x=R &&
y=R
yes
Is(x>0 & y>0 & x<6 &
y<6)
no
print(Position is out of range, Please Try again.)
Is (x%2=0)
no
Is(y%2=0)
no
print(Wrong Position, Please Try again)
yes
Is (y%2=1)
no
print(Wrong Position,
Please Try again)
yes
Is(Beginner_Board[x][y]=' ')
no
print(The Position you have chosen is already
taken, Please Try again.)
yes
yes
Is(Beginner_Board[x][y]=' ')
no
yes
Beginner_Board[x][y]='=',
Beginner_Board_Colour[x][y]=1,
Player_2.Turn=1,
Check_For_Boxes_and_RE_Play,
Boards=Boards-1
Beginner_Board[x][y]=186,
Beginner_Board_Colour[x][y]=1,
Player_2.Turn=1,
Check_For_Boxes_and_RE_Play,
Boards=Boards-1
print(The Position you
have chosen is already
taken, Please Try
again.)
Print(Current_Board,
Current_Game_Status,
Print(Current_Board,
Current_Game_Status,

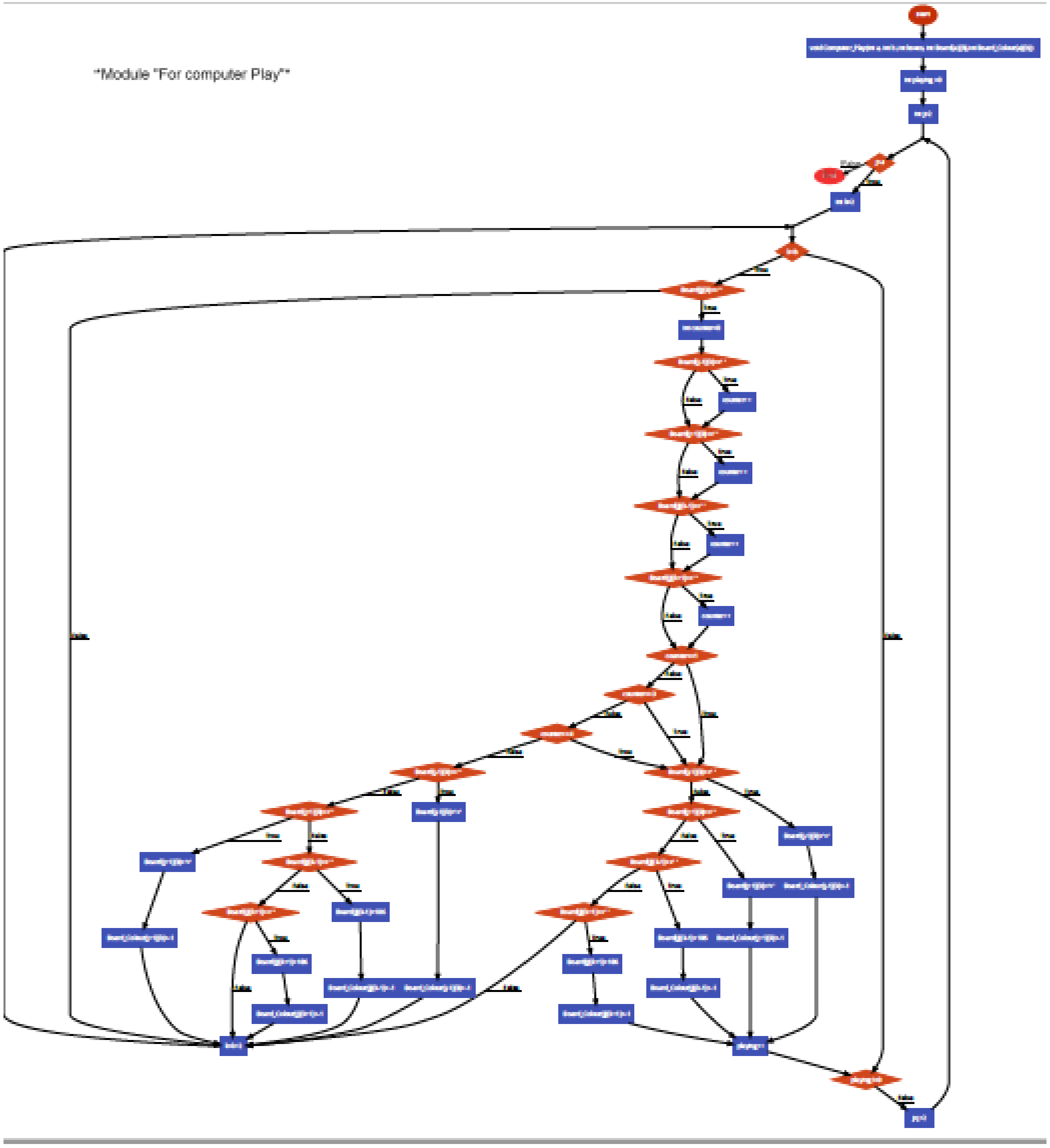


- ***We also make Flowcharts For Modules Files.***
- ***For a better Quality and high resolution,These***

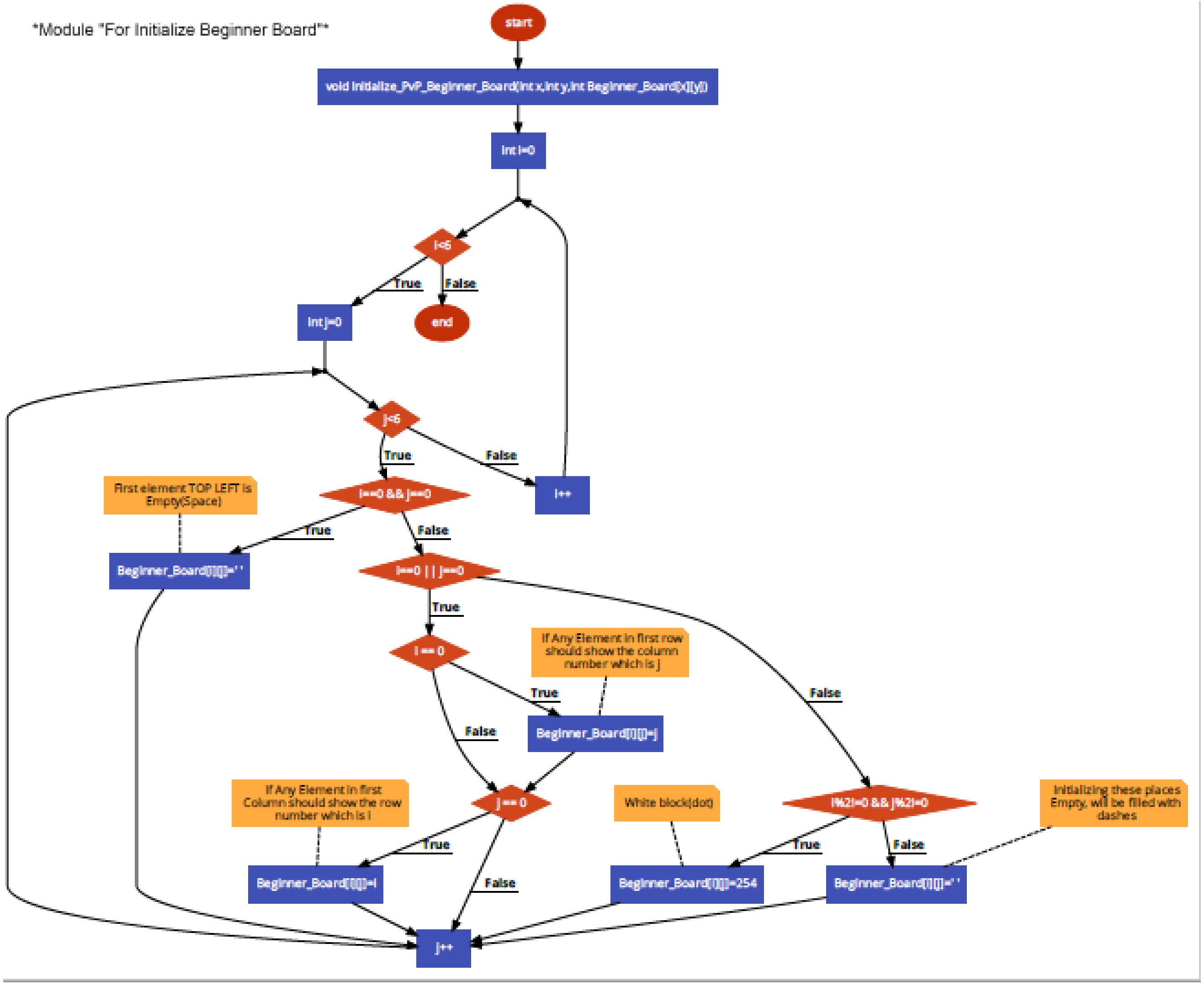
Flowcharts will be included as pdf file in the Documentation Folder.


"Module "For Check Boxes and RE_play""
Student_1 Name : George Samy Wahba Beshay
Student_1 ID : 20010435
Student_2 Name : Mariam Aziz Gerges Zaki
Student_2 ID : 20011689
start
end
True
False
Borders counter

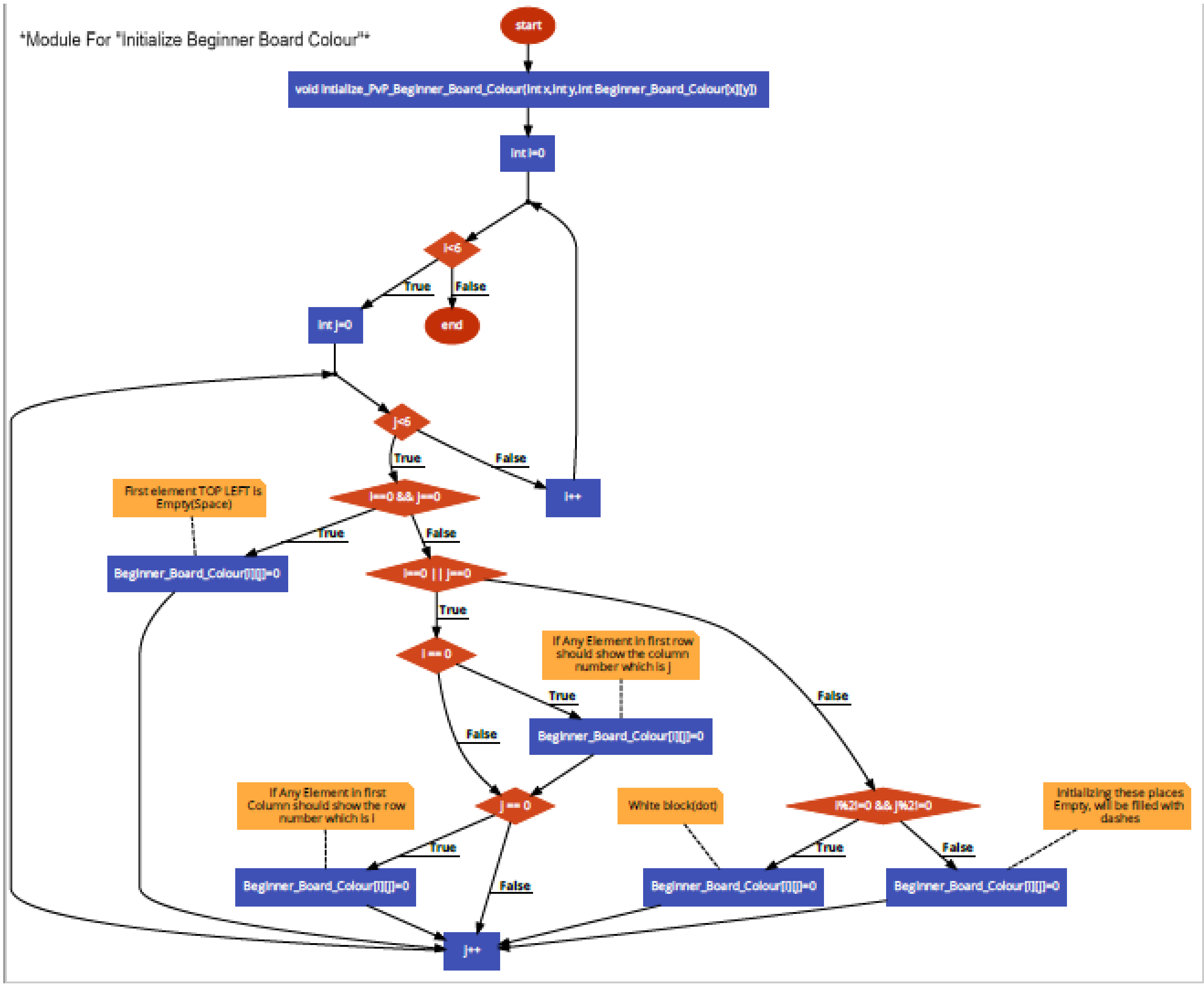
-
- *- This Flowchart for Module Check Boxes and RE_Play.*



- ***-This Flowchart for Module Computer Play.***

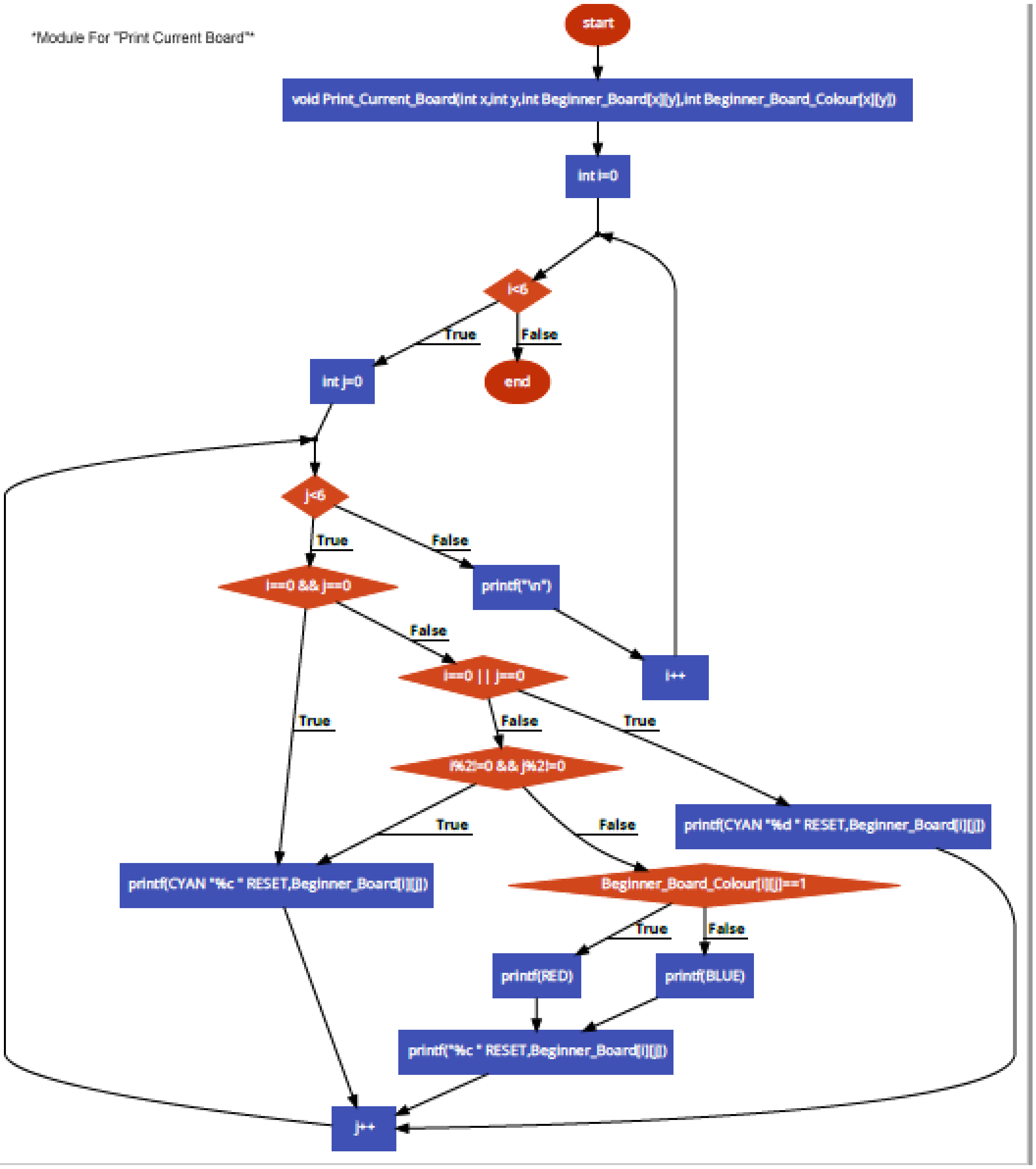

Module "For Initialize Beginner Board"
start
void Initialize_PvP_Beginner_Board(int x,int y,int Beginner_Board[x][y])
int i=0
i<6
True
False
end
int j=0
j<6
True
False
i++
i==0 && j==0
True
False
First element TOP LEFT is Empty(Space)
Beginner_Board[i][j]=' '
i==0 || j==0
True
False
i == 0
If Any Element in first row should show the column number which is j
True
False
Beginner_Board[i][j]=j
j == 0
If Any Element in first Column should show the row number which is i
True
False
Beginner_Board[i][j]=i
White block(dot)
i%2!=0 && j%2!=0
True
False
Initializing these places Empty, will be filled with dashes
Beginner_Board[i][j]=254
Beginner_Board[i][j]=' '
j++

-
- ***-This Flowchart for Module Initialize Beginner Board.***

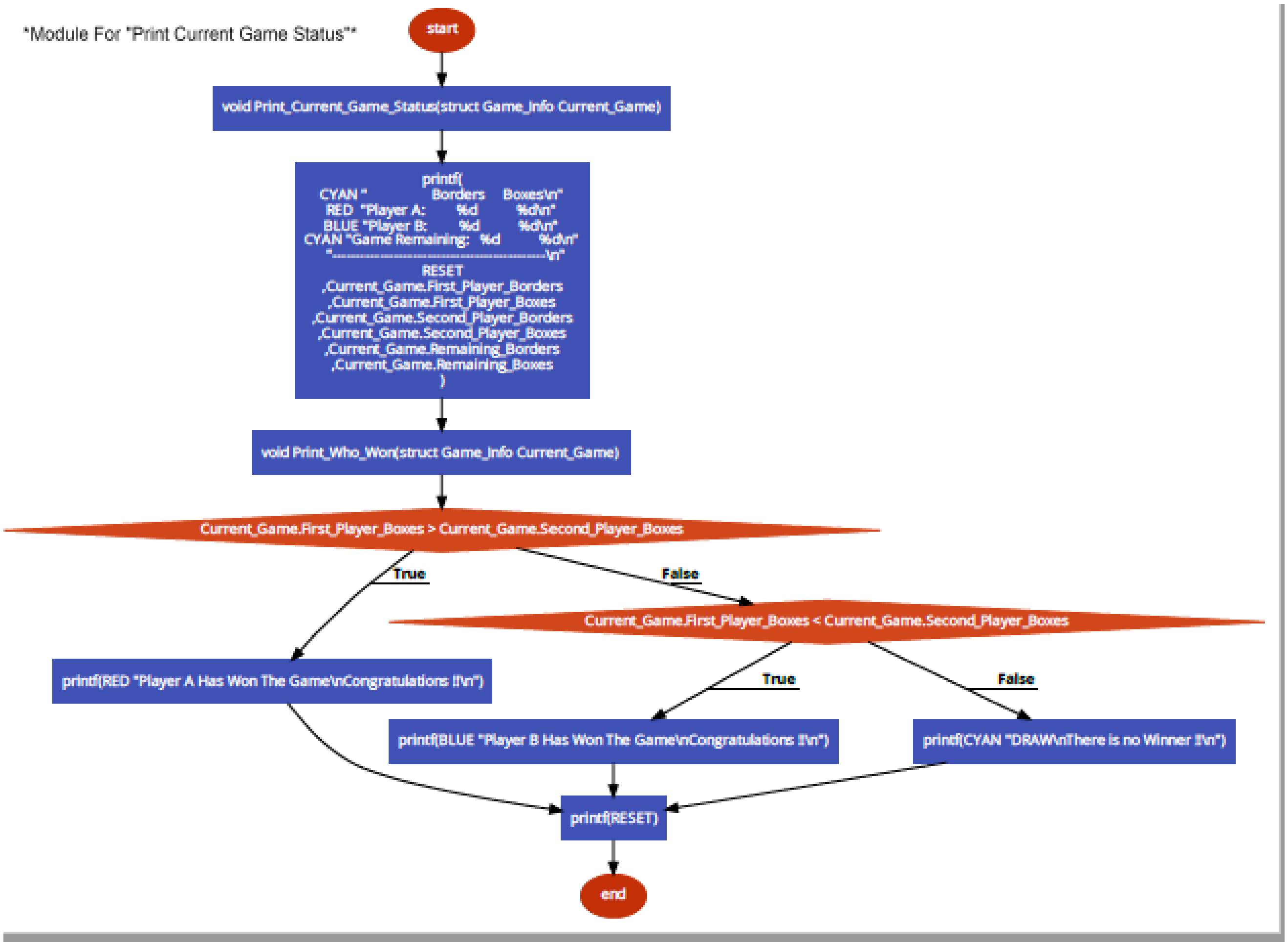
- 

Module For "Initialize Beginner Board Colour"
start
void Intialize_PvP_Beginner_Board_Colour(int x,int y,int Beginner_Board_Colour[x][y])
int i=0
i<6
True
False
int j=0
end
j<6
True
False
i++
i==0 && j==0
First element TOP LEFT is Empty(Space)
True
False
Beginner_Board_Colour[i][j]=0
i==0 || j==0
True
i == 0
If Any Element in first row should show the column number which is j
True
False
Beginner_Board_Colour[i][j]=0
False
j == 0
If Any Element in first Column should show the row number which is i
True
False
Beginner_Board_Colour[i][j]=0
White block(dot)
i%2!=0 && j%2!=0
True
False
Initializing these places Empty, will be filled with dashes
Beginner_Board_Colour[i][j]=0
Beginner_Board_Colour[i][j]=0
j++

- ***This Flowchart for Module Initialize Beginner Board Colour.***

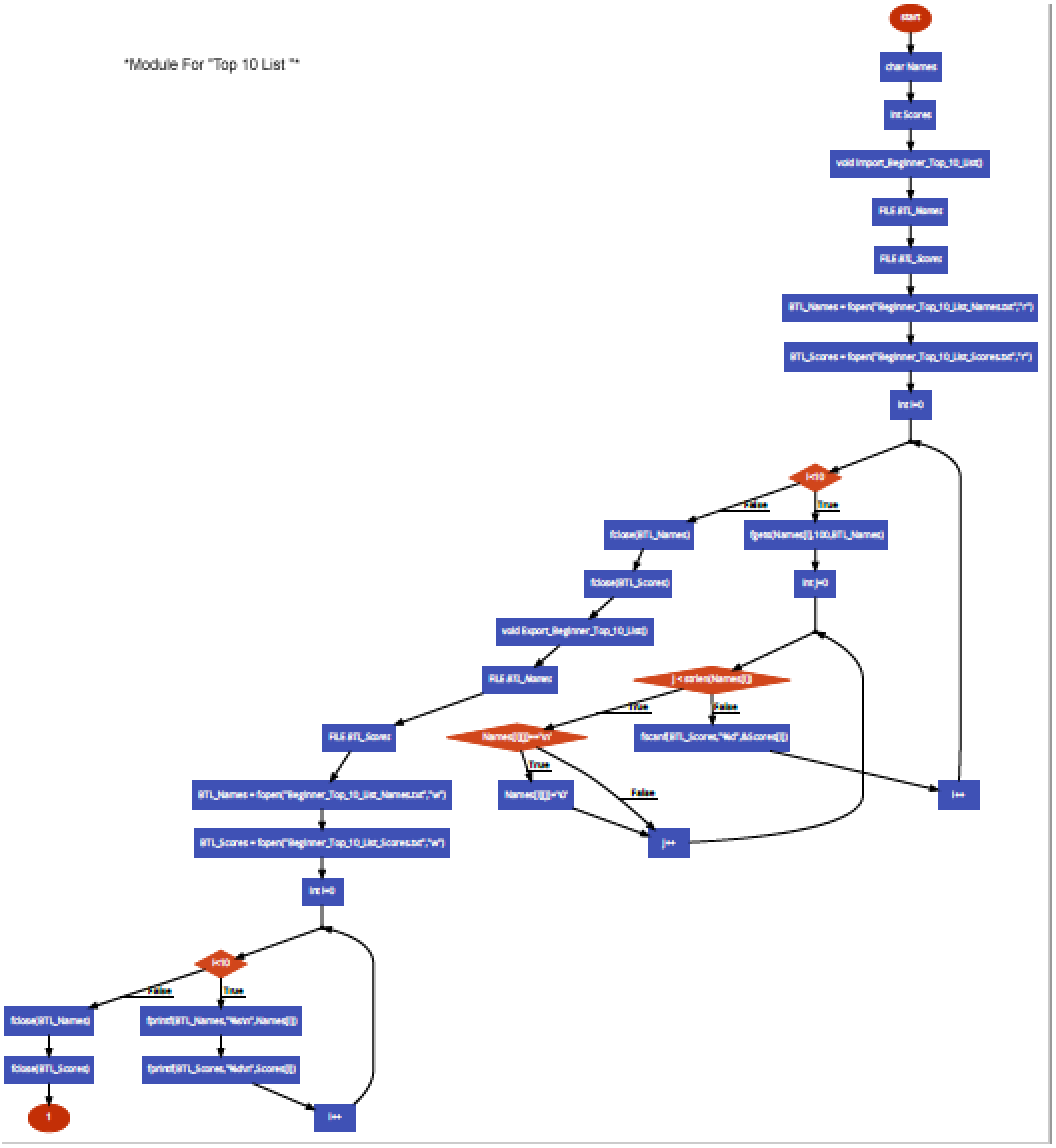

Module For "Print Current Board"
start
void Print_Current_Board(int x,int y,int Beginner_Board[x][y],int Beginner_Board_Colour[x][y])
int i=0
i<6
True
False
int j=0
end
j<6
True
False
i==0 && j==0
printf("\n")
False
i==0 || j==0
i++
True
False
True
i%2!=0 && j%2!=0
True
False
printf(CYAN "%d " RESET,Beginner_Board[i][j])
printf(CYAN "%c " RESET,Beginner_Board[i][j])
Beginner_Board_Colour[i][j]==1
True
False
printf(RED)
printf(BLUE)
printf("%c " RESET,Beginner_Board[i][j])
j++

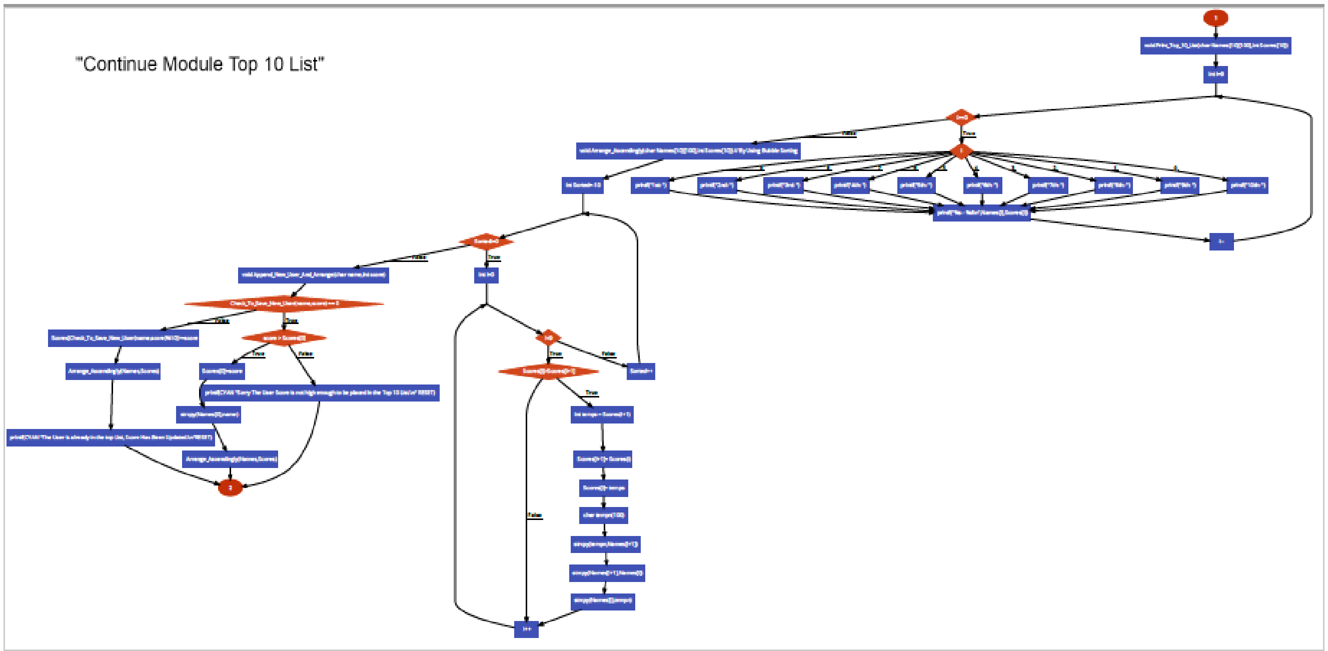
- ***-This Flowchart for Print Current Board.***

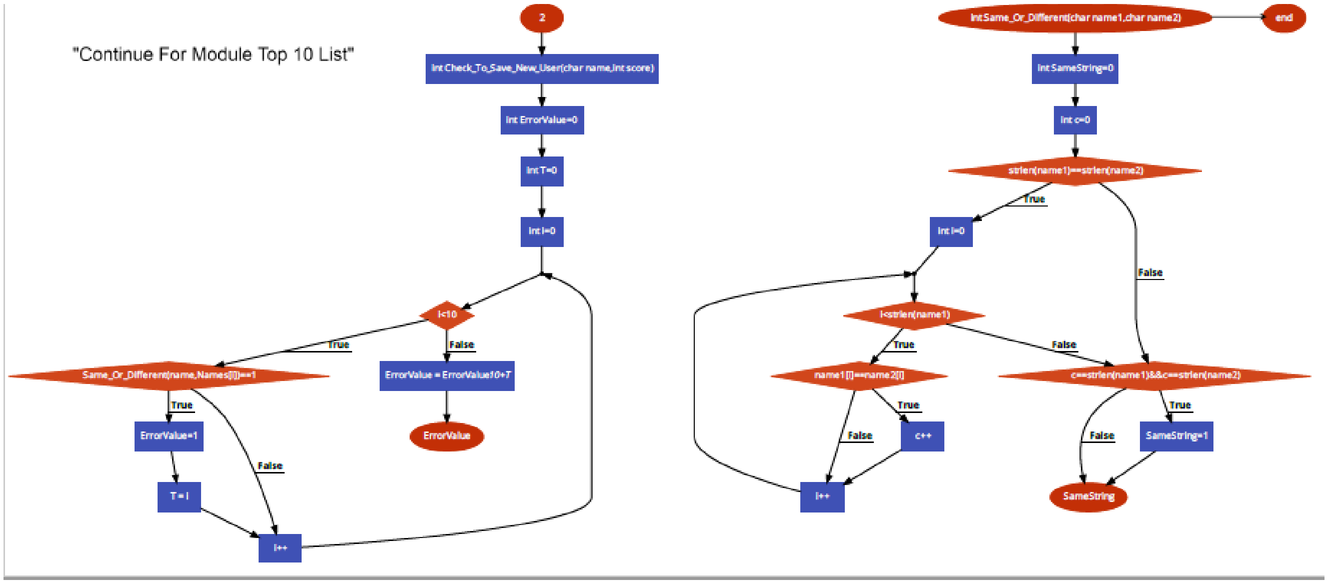

Module For "Print Current Game Status"
start
void Print_Current_Game_Status(struct Game_Info Current_Game)
printf(
CYAN "   Borders   Boxes\n"
RED "Player A: %d   %d\n"
BLUE "Player B: %d   %d\n"
CYAN "Game Remaining: %d   %d\n"
"--\n"
RESET
,Current_Game.First_Player_Borders
,Current_Game.First_Player_Boxes
,Current_Game.Second_Player_Borders
,Current_Game.Second_Player_Boxes
,Current_Game.Remaining_Borders
,Current_Game.Remaining_Boxes
)
void Print_Who_Won(struct Game_Info Current_Game)
Current_Game.First_Player_Boxes > Current_Game.Second_Player_Boxes
True
False
Current_Game.First_Player_Boxes < Current_Game.Second_Player_Boxes
printf(RED "Player A Has Won The Game\nCongratulations !!\n")
True
False
printf(BLUE "Player B Has Won The Game\nCongratulations !!\n")
printf(CYAN "DRAW\nThere is no Winner !!\n")
printf(RESET)
end

- ***-This Flowchart for Module Print Current Game Status.***

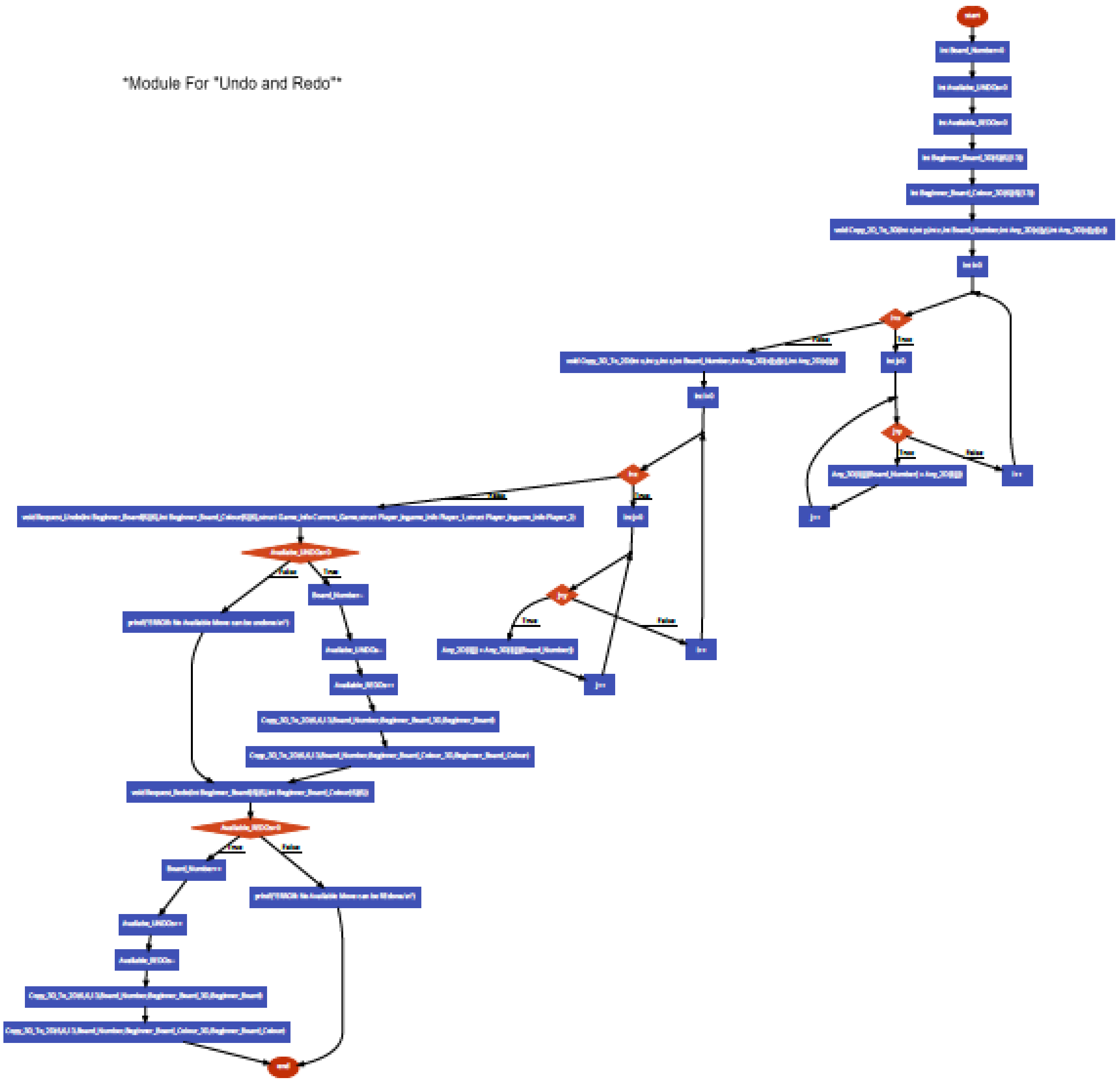
Module For "Top 10 List "


start
char Names
int Scores
void Import_Beginner_Top_10_List()
FILE BTL_Names
FILE BTL_Scores
BTL_Names = fopen("Beginner_Top_10_List_Names.txt","r")
BTL_Scores = fopen("Beginner_Top_10_List_Scores.txt","r")
int i=0
i<10
False
True
fclose(BTL_Names)
fgets(Names[i],100,BTL_Names)
fclose(BTL_Scores)
int j=0
void Export_Beginner_Top_10_List()
FILE BTL_Names
j < strlen(Names[i])
True
False
FILE BTL_Scores
Names[i][j]=='\n'
fscanf(BTL_Scores,"%d",&Scores[i])
True
False
BTL_Names = fopen("Beginner_Top_10_List_Names.txt","w")
Names[i][j]='\0'
i++
BTL_Scores = fopen("Beginner_Top_10_List_Scores.txt","w")
j++
int i=0
i<10
False
True
fclose(BTL_Names)
fprintf(BTL_Names,"%s\n",Names[i])
fclose(BTL_Scores)
fprintf(BTL_Scores,"%d\n",Scores[i])
1
i++


"Continue Module Top 10 List"
1
void Print_Top_10_List(char Names[10][100],int Scores[10])
int i=0
False
True
void Arrange_Ascending(char Names[10][100],int Scores[10]) // By Using Bubble Sorting
printf("%s - %d\n",Names[i],Scores[i])
i++
void Append_New_User_And_Arrange(char name,int score)
Check_To_Save_New_User(name,score) == 0
Scores[Check_To_Save_New_User(name,score)%10]=score
score > Scores[0]
Arrange_Ascending(Names,Scores)
Scores[0]=score
printf(CYAN "Sorry The User Score is not high enough to be placed in the Top 10 List\n" RESET)
strcpy(Names[0],name)
printf(CYAN "The User is already in the top List, Score Has Been Updated\n" RESET)
Arrange_Ascending(Names,Scores)
2
int temps = Scores[i+1]
Scores[i+1]= Scores[i]
Scores[i]= temps
char tempn[100]
strcpy(tempn,Names[i+1])
strcpy(Names[i+1],Names[i])
strcpy(Names[i],tempn)
i++

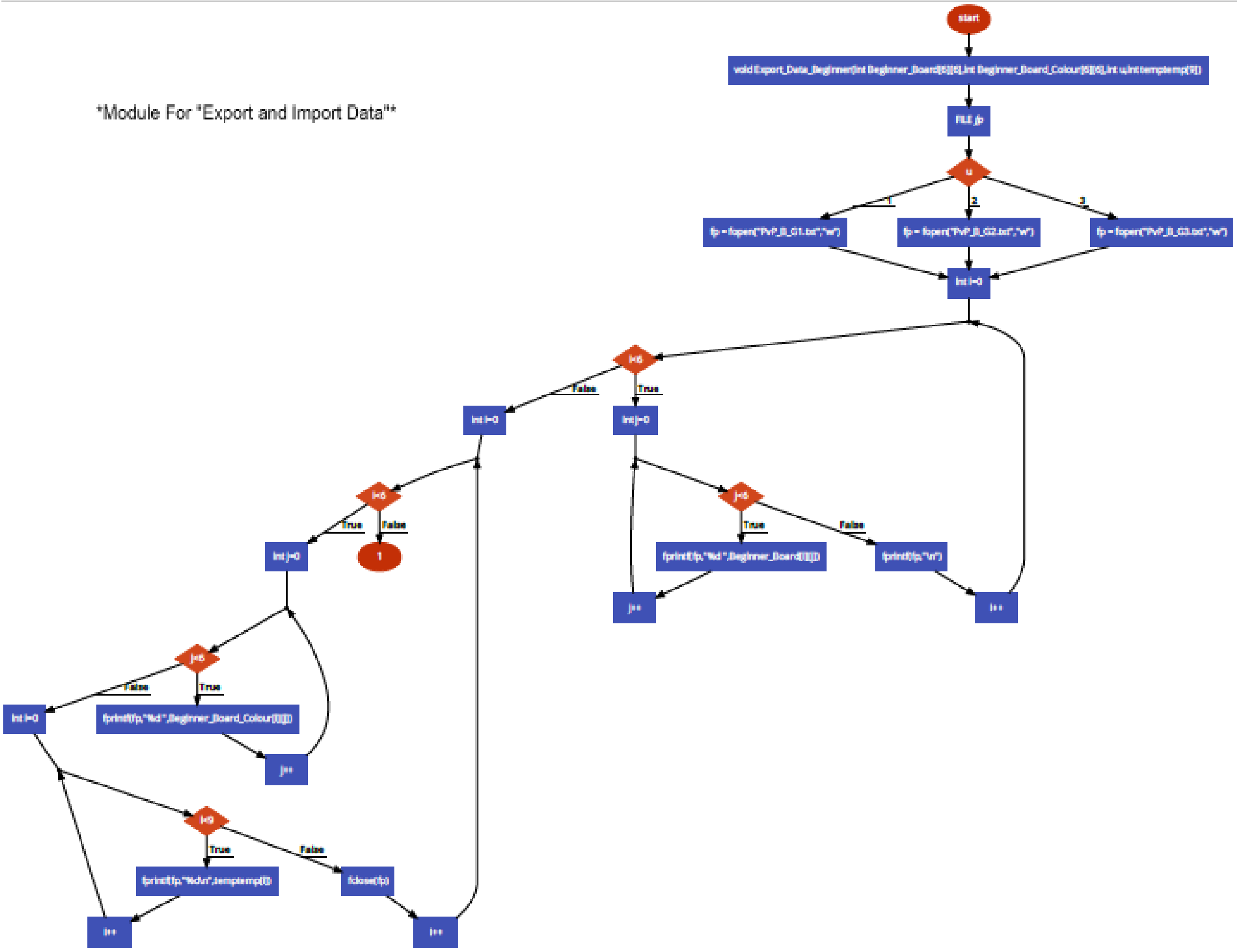

"Continue For Module Top 10 List"
2
Int Check_To_Save_New_User(char name,Int score)
Int ErrorValue=0
Int T=0
Int I=0
I<10
True
False
Same_Or_Different(name,Names[I])==1
ErrorValue = ErrorValue10+T
True
ErrorValue=1
False
ErrorValue
T = I
I++
IntSame_Or_Different(char name1,char name2)
end
Int SameString=0
Int c=0
strlen(name1)==strlen(name2)
True
Int I=0
False
I<strlen(name1)
True
False
name1[I]==name2[I]
c==strlen(name1)&&c==strlen(name2)
True
False
c++
False
True
SameString=1
I++
SameString

- ***-This Flowchart for Module Top 10 List.***

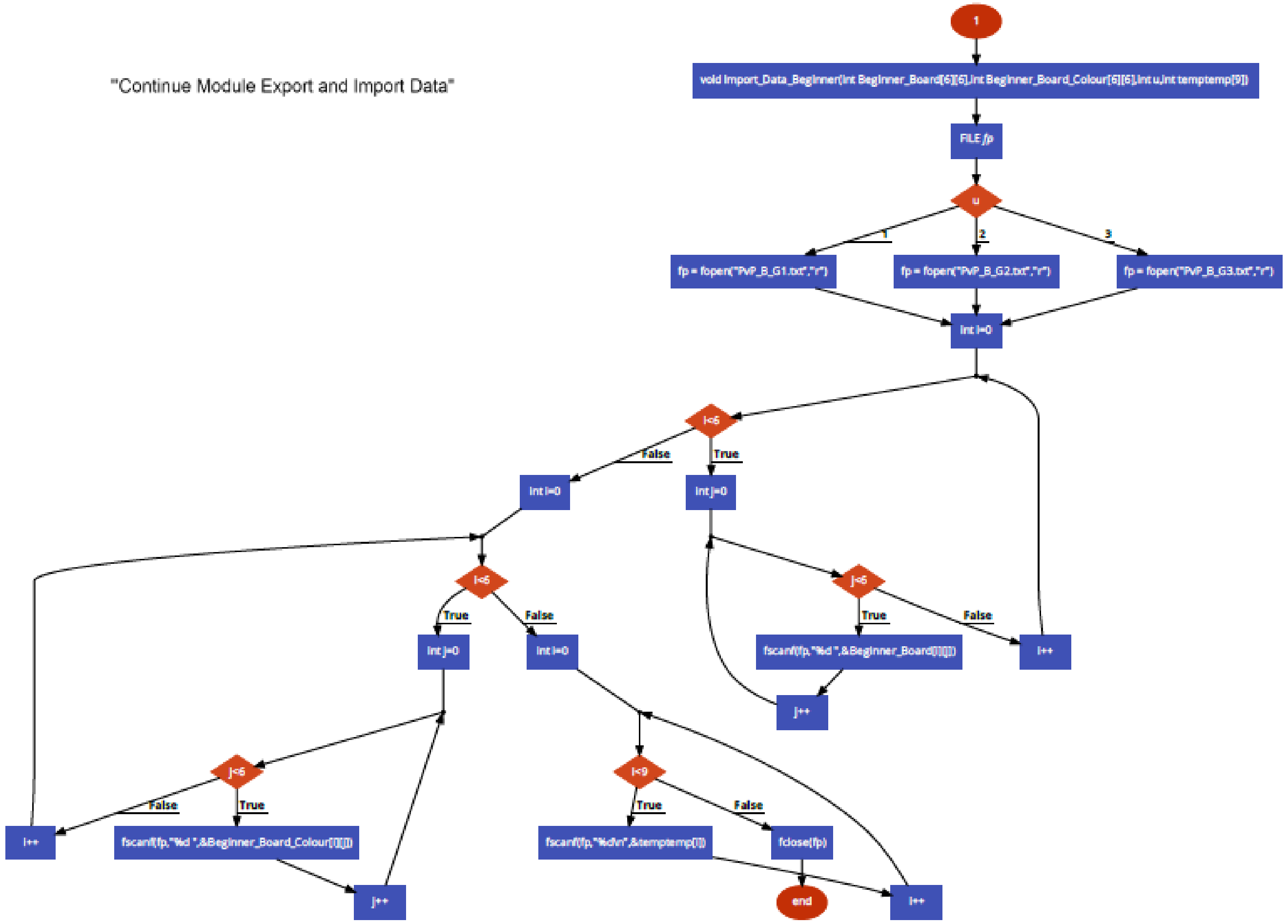


- ***This Flowchart for Module Undo and Redo.***

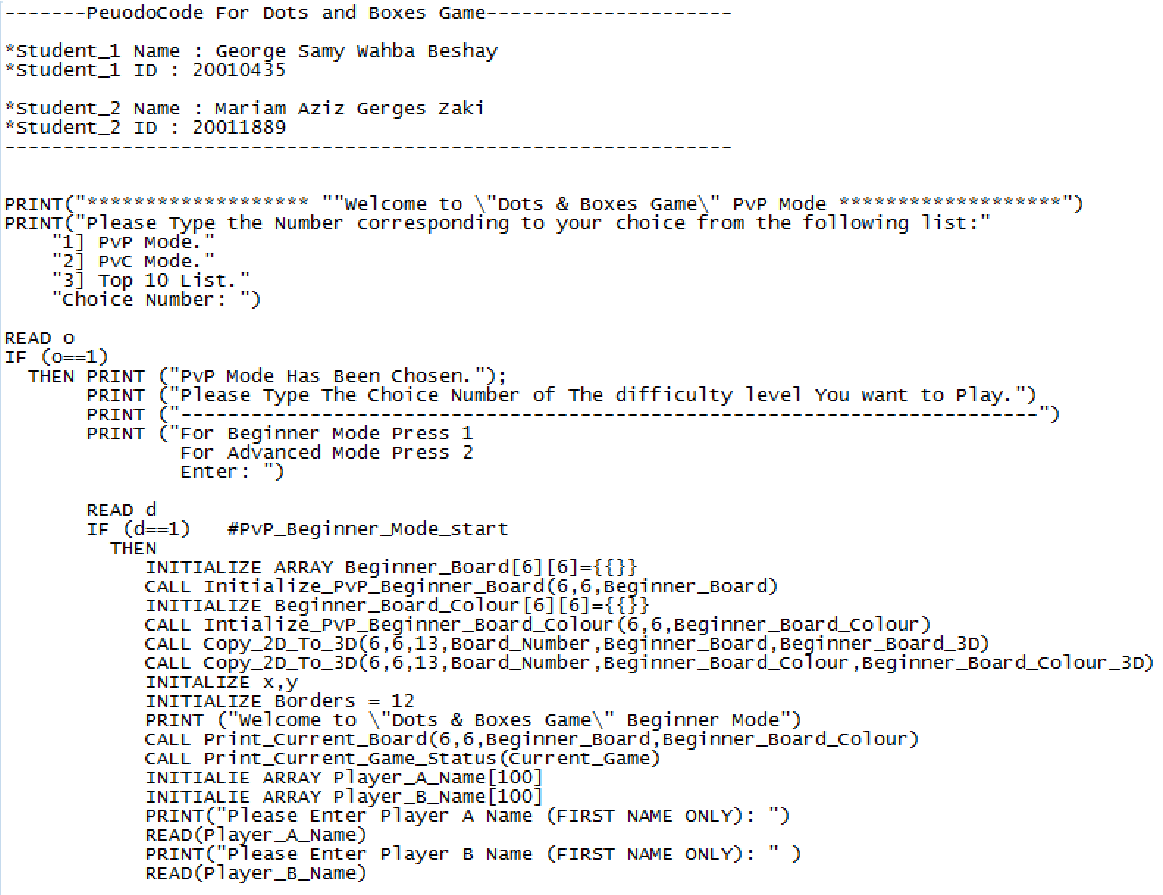
Module For "Export and Import Data"


start
void Export_Data_Beginner(int Beginner_Board[6][6],int Beginner_Board_Colour[6][6],int u,int temptemp[9])
FILE fp
u
1
2
3
fp = fopen("PvP_B_G1.txt","w")
fp = fopen("PvP_B_G2.txt","w")
fp = fopen("PvP_B_G3.txt","w")
int i=0
i<6
False
True
int i=0
int j=0
j<6
True
False
fprintf(fp,"%d ",Beginner_Board[i][j])
fprintf(fp,"\n")
j++
i++
i<6
True
False
int j=0
1
j<6
False
True
int i=0
fprintf(fp,"%d ",Beginner_Board_Colour[i][j])
j++
i<9
True
False
fprintf(fp,"%d\n",temptemp[i])
fclose(fp)
i++
i++

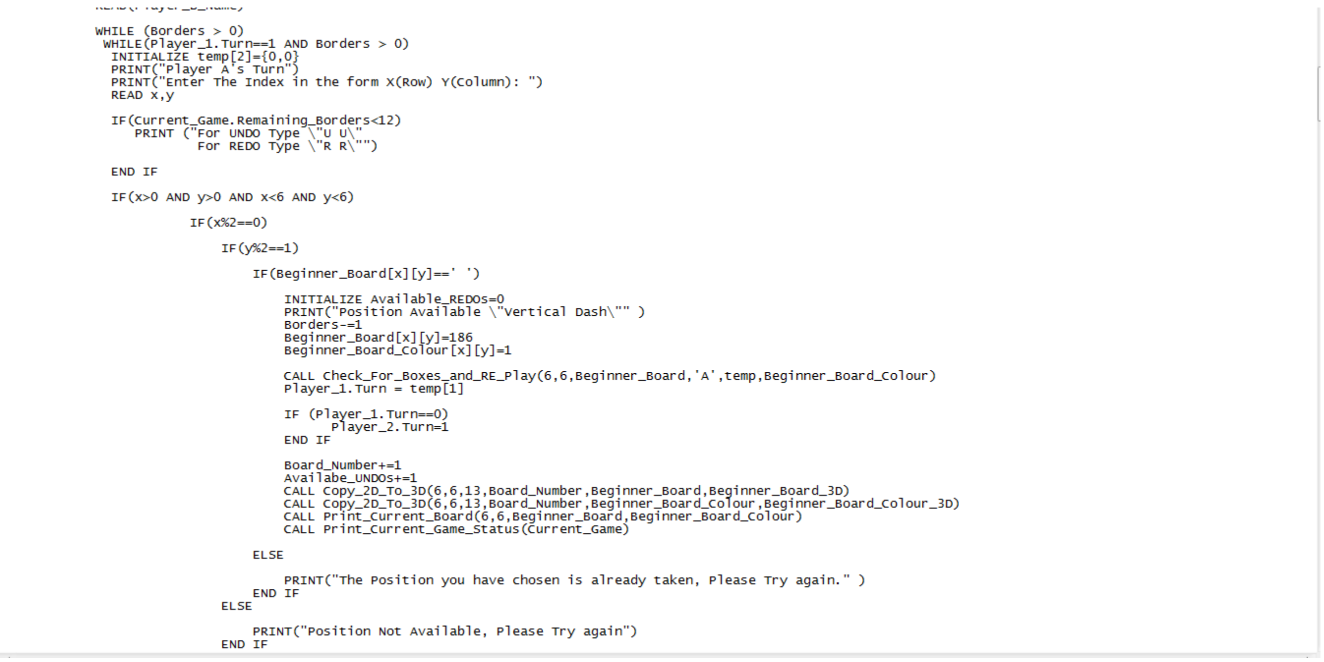
"Continue Module Export and Import Data"


1
void Import_Data_Beginner(int Beginner_Board[6][6],int Beginner_Board_Colour[6][6],int u,int temptemp[9])
FILE fp
u
1
2
3
fp = fopen("PvP_B_G1.txt","r")
fp = fopen("PvP_B_G2.txt","r")
fp = fopen("PvP_B_G3.txt","r")
int i=0
i<6
False
True
int i=0
int j=0
i<6
True
False
int j=0
int i=0
j<6
True
False
fscanf(fp,"%d ",&Beginner_Board[i][j])
i++
j++
j<6
False
True
i++
fscanf(fp,"%d ",&Beginner_Board_Colour[i][j])
j++
i<9
True
False
fscanf(fp,"%d\n",&temptemp[i])
fclose(fp)
end
i++

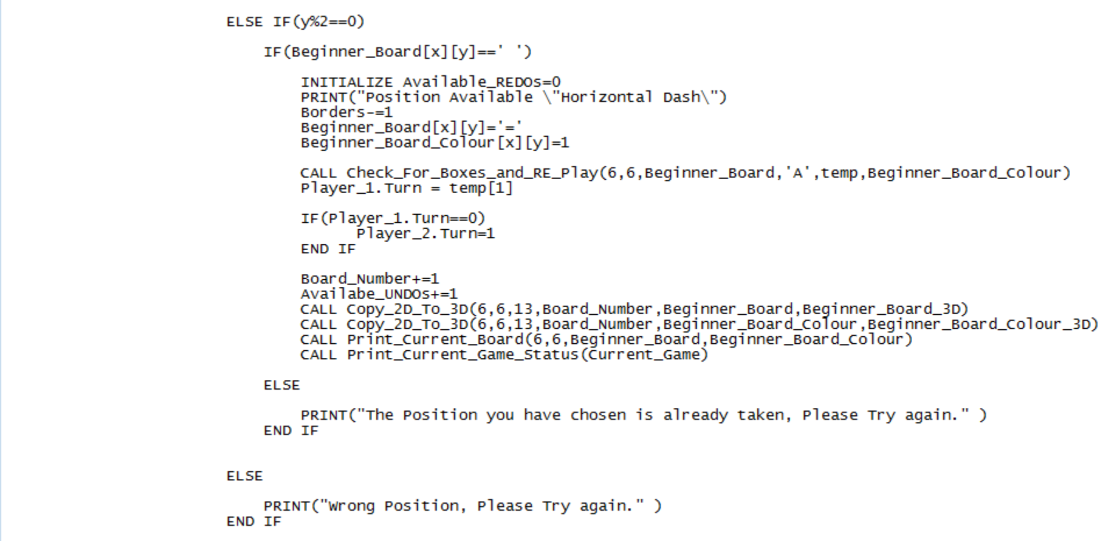
- *-This Flowchart For Export and Import Data.*
- ***We also make pseudocode for the Game.***
- ***For a better Quality and high resolution, The pseudocode will be included as text file in the Documentation Folder.***



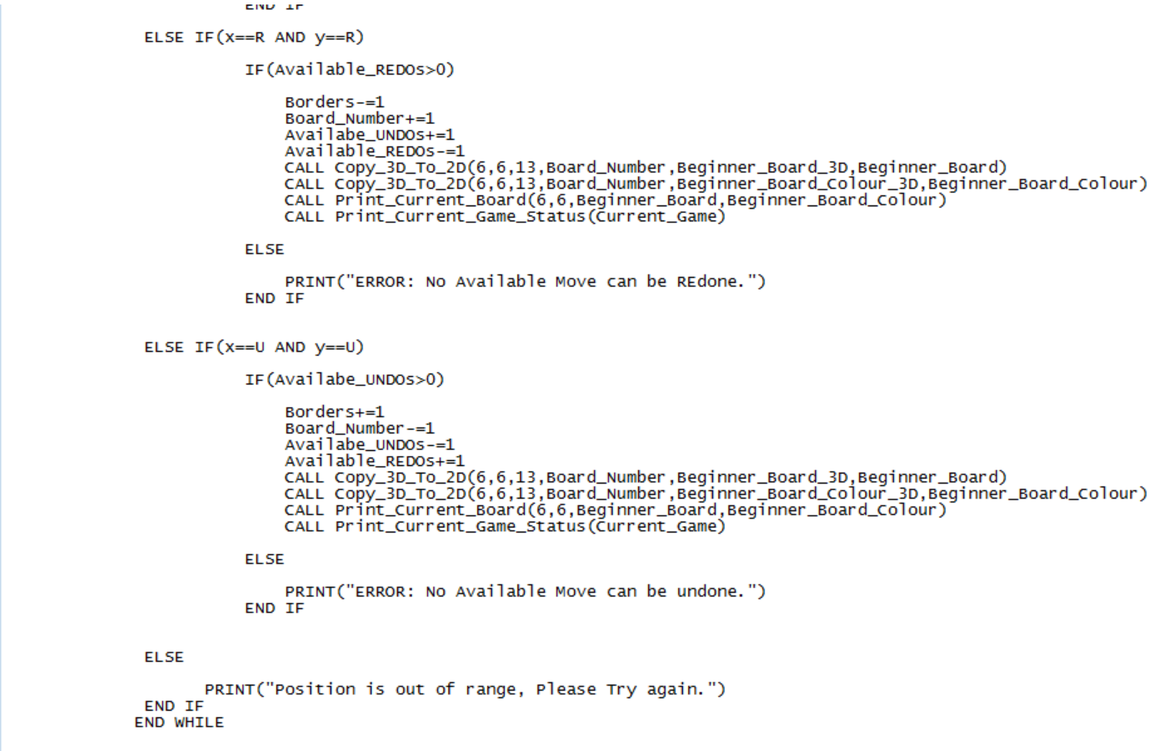
```
-------PeuodoCode For Dots and Boxes Game---------------------

*Student_1 Name : George Samy Wahba Beshay
*Student_1 ID : 20010435

*Student_2 Name : Mariam Aziz Gerges Zaki
*Student_2 ID : 20011889
--------------------------------------------------------------


PRINT("******************* ""Welcome to \"Dots & Boxes Game\" PvP Mode *******************")
PRINT("Please Type the Number corresponding to your choice from the following list:"
    "1] PvP Mode."
    "2] PvC Mode."
    "3] Top 10 List."
    "Choice Number: ")

READ o
IF (o==1)
  THEN PRINT ("PvP Mode Has Been Chosen.");
       PRINT ("Please Type The Choice Number of The difficulty level You want to Play.")
       PRINT ("----------------------------------------------------------------------")
       PRINT ("For Beginner Mode Press 1
               For Advanced Mode Press 2
               Enter: ")

       READ d
       IF (d==1)   #PvP_Beginner_Mode_start
         THEN
            INITIALIZE ARRAY Beginner_Board[6][6]={{}}
            CALL Initialize_PvP_Beginner_Board(6,6,Beginner_Board)
            INITIALIZE Beginner_Board_Colour[6][6]={{}}
            CALL Intialize_PvP_Beginner_Board_Colour(6,6,Beginner_Board_Colour)
            CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board,Beginner_Board_3D)
            CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board_Colour,Beginner_Board_Colour_3D)
            INITALIZE x,y
            INITIALIZE Borders = 12
            PRINT ("Welcome to \"Dots & Boxes Game\" Beginner Mode")
            CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
            CALL Print_Current_Game_Status(Current_Game)
            INITIALIE ARRAY Player_A_Name[100]
            INITIALIE ARRAY Player_B_Name[100]
            PRINT("Please Enter Player A Name (FIRST NAME ONLY): ")
            READ(Player_A_Name)
            PRINT("Please Enter Player B Name (FIRST NAME ONLY): " )
            READ(Player_B_Name)
```



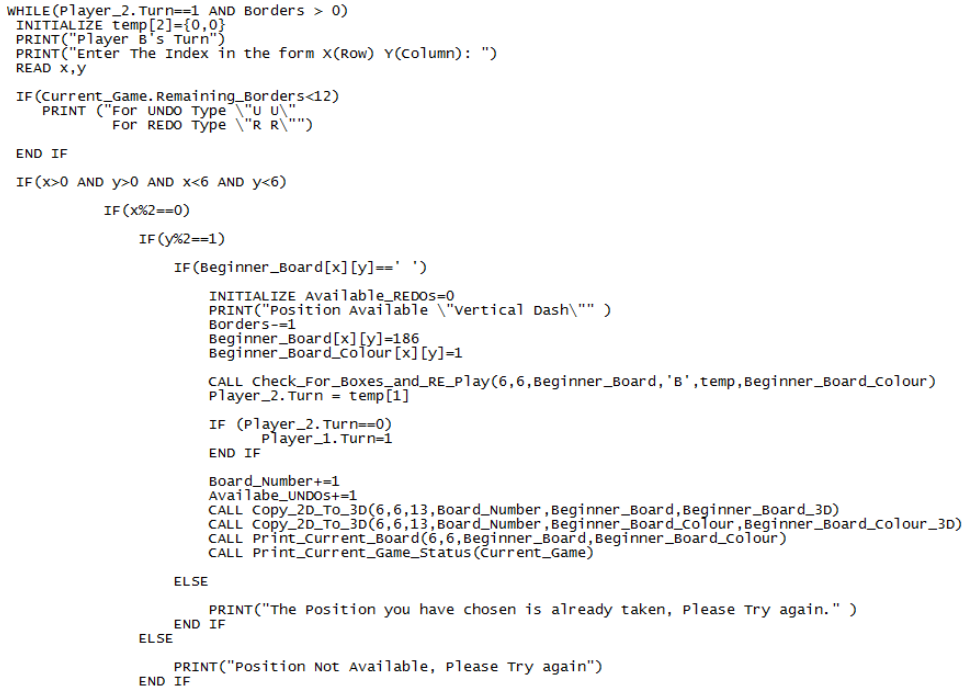
```
WHILE (Borders > 0)
 WHILE(Player_1.Turn==1 AND Borders > 0)
  INITIALIZE temp[2]={0,0}
  PRINT("Player A's Turn")
  PRINT("Enter The Index in the form X(Row) Y(Column): ")
  READ x,y

  IF(Current_Game.Remaining_Borders<12)
     PRINT ("For UNDO Type \"U U\"
             For REDO Type \"R R\"")

  END IF

  IF(x>0 AND y>0 AND x<6 AND y<6)

            IF(x%2==0)

                IF(y%2==1)

                    IF(Beginner_Board[x][y]==' ')

                        INITIALIZE Available_REDOs=0
                        PRINT("Position Available \"Vertical Dash\"" )
                        Borders-=1
                        Beginner_Board[x][y]=186
                        Beginner_Board_Colour[x][y]=1

                        CALL Check_For_Boxes_and_RE_Play(6,6,Beginner_Board,'A',temp,Beginner_Board_Colour)
                        Player_1.Turn = temp[1]

                        IF (Player_1.Turn==0)
                              Player_2.Turn=1
                        END IF

                        Board_Number+=1
                        Availabe_UNDOs+=1
                        CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board,Beginner_Board_3D)
                        CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board_Colour,Beginner_Board_Colour_3D)
                        CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
                        CALL Print_Current_Game_Status(Current_Game)

                    ELSE

                        PRINT("The Position you have chosen is already taken, Please Try again." )
                    END IF
                ELSE

                    PRINT("Position Not Available, Please Try again")
                END IF
```



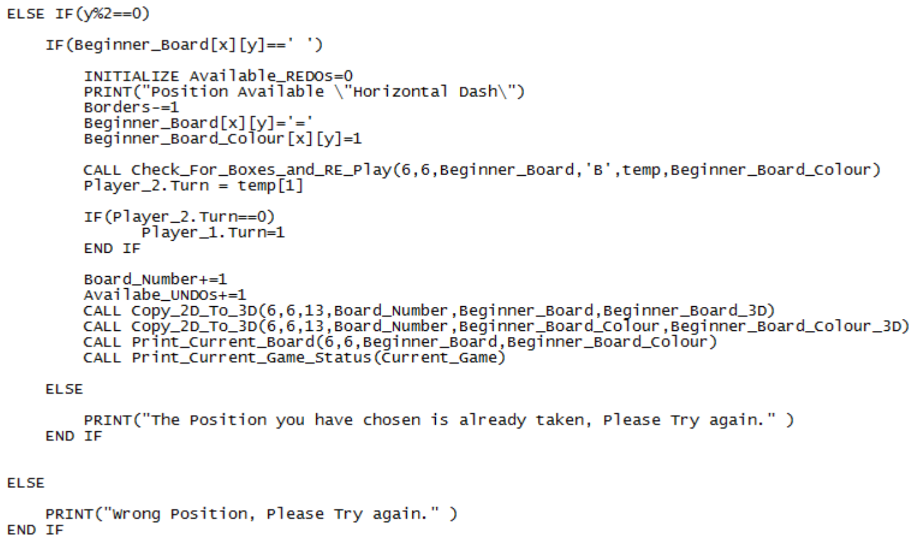
```
ELSE IF(y%2==0)

    IF(Beginner_Board[x][y]==' ')

        INITIALIZE Available_REDOs=0
        PRINT("Position Available \"Horizontal Dash\")
        Borders-=1
        Beginner_Board[x][y]='='
        Beginner_Board_Colour[x][y]=1

        CALL Check_For_Boxes_and_RE_Play(6,6,Beginner_Board,'A',temp,Beginner_Board_Colour)
        Player_1.Turn = temp[1]

        IF(Player_1.Turn==0)
              Player_2.Turn=1
        END IF

        Board_Number+=1
        Availabe_UNDOs+=1
        CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board,Beginner_Board_3D)
        CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board_Colour,Beginner_Board_Colour_3D)
        CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
        CALL Print_Current_Game_Status(Current_Game)

    ELSE

        PRINT("The Position you have chosen is already taken, Please Try again." )
    END IF


ELSE

    PRINT("Wrong Position, Please Try again." )
END IF
```



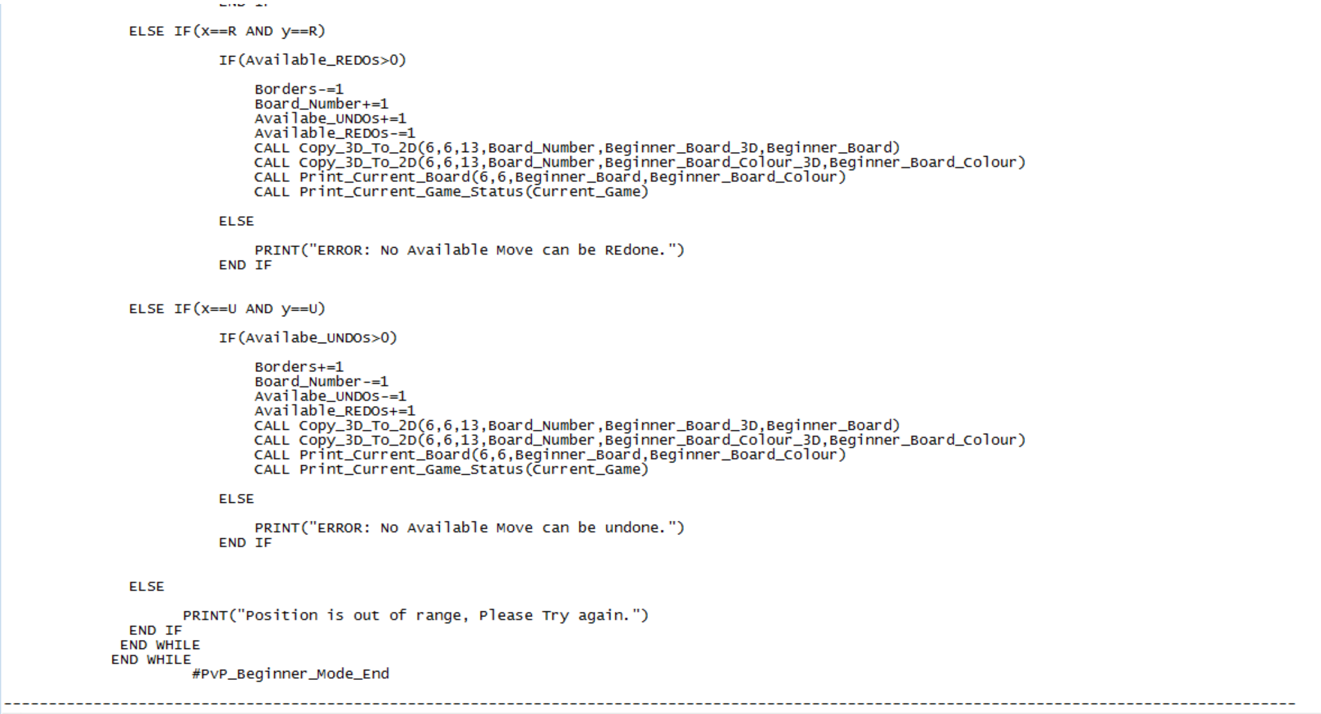
```
           END IF
 ELSE IF(x==R AND y==R)

           IF(Available_REDOs>0)

               Borders-=1
               Board_Number+=1
               Availabe_UNDOs+=1
               Available_REDOs-=1
               CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_3D,Beginner_Board)
               CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_Colour_3D,Beginner_Board_Colour)
               CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
               CALL Print_Current_Game_Status(Current_Game)

           ELSE

               PRINT("ERROR: No Available Move can be REdone.")
           END IF


 ELSE IF(x==U AND y==U)

           IF(Availabe_UNDOs>0)

               Borders+=1
               Board_Number-=1
               Availabe_UNDOs-=1
               Available_REDOs+=1
               CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_3D,Beginner_Board)
               CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_Colour_3D,Beginner_Board_Colour)
               CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
               CALL Print_Current_Game_Status(Current_Game)

           ELSE

               PRINT("ERROR: No Available Move can be undone.")
           END IF


 ELSE

       PRINT("Position is out of range, Please Try again.")
 END IF
END WHILE
```



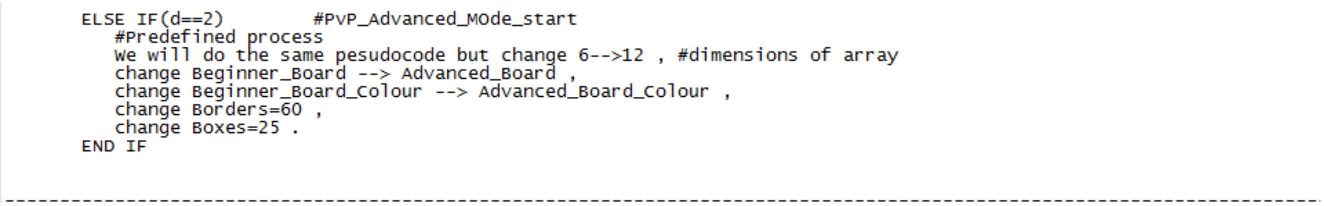
```
WHILE(Player_2.Turn==1 AND Borders > 0)
 INITIALIZE temp[2]={0,0}
 PRINT("Player B's Turn")
 PRINT("Enter The Index in the form X(Row) Y(Column): ")
 READ x,y

 IF(Current_Game.Remaining_Borders<12)
    PRINT ("For UNDO Type \"U U\"
            For REDO Type \"R R\"")

 END IF

 IF(x>0 AND y>0 AND x<6 AND y<6)

           IF(x%2==0)

               IF(y%2==1)

                   IF(Beginner_Board[x][y]==' ')

                       INITIALIZE Available_REDOs=0
                       PRINT("Position Available \"Vertical Dash\"" )
                       Borders-=1
                       Beginner_Board[x][y]=186
                       Beginner_Board_Colour[x][y]=1

                       CALL Check_For_Boxes_and_RE_Play(6,6,Beginner_Board,'B',temp,Beginner_Board_Colour)
                       Player_2.Turn = temp[1]

                       IF (Player_2.Turn==0)
                             Player_1.Turn=1
                       END IF

                       Board_Number+=1
                       Availabe_UNDOs+=1
                       CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board,Beginner_Board_3D)
                       CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board_Colour,Beginner_Board_Colour_3D)
                       CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
                       CALL Print_Current_Game_Status(Current_Game)

                   ELSE

                       PRINT("The Position you have chosen is already taken, Please Try again." )
                   END IF
               ELSE

                   PRINT("Position Not Available, Please Try again")
               END IF
```



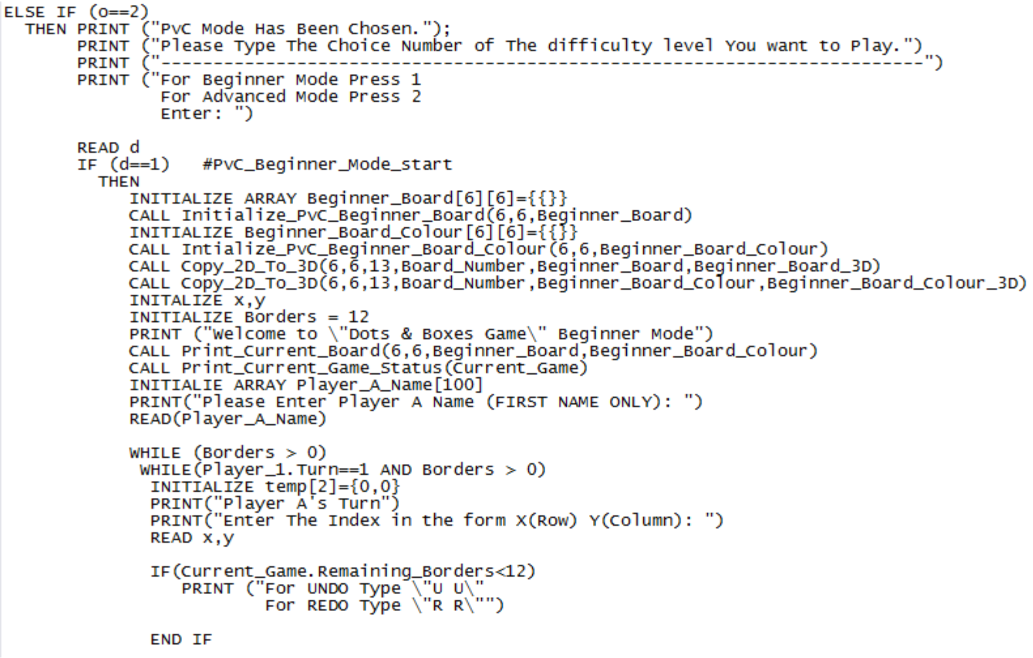
```
ELSE IF(y%2==0)

    IF(Beginner_Board[x][y]==' ')

        INITIALIZE Available_REDOs=0
        PRINT("Position Available \"Horizontal Dash\")
        Borders-=1
        Beginner_Board[x][y]='='
        Beginner_Board_Colour[x][y]=1

        CALL Check_For_Boxes_and_RE_Play(6,6,Beginner_Board,'B',temp,Beginner_Board_Colour)
        Player_2.Turn = temp[1]

        IF(Player_2.Turn==0)
              Player_1.Turn=1
        END IF

        Board_Number+=1
        Availabe_UNDOs+=1
        CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board,Beginner_Board_3D)
        CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board_Colour,Beginner_Board_Colour_3D)
        CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
        CALL Print_Current_Game_Status(Current_Game)

    ELSE

        PRINT("The Position you have chosen is already taken, Please Try again." )
    END IF


ELSE

    PRINT("Wrong Position, Please Try again." )
END IF
```



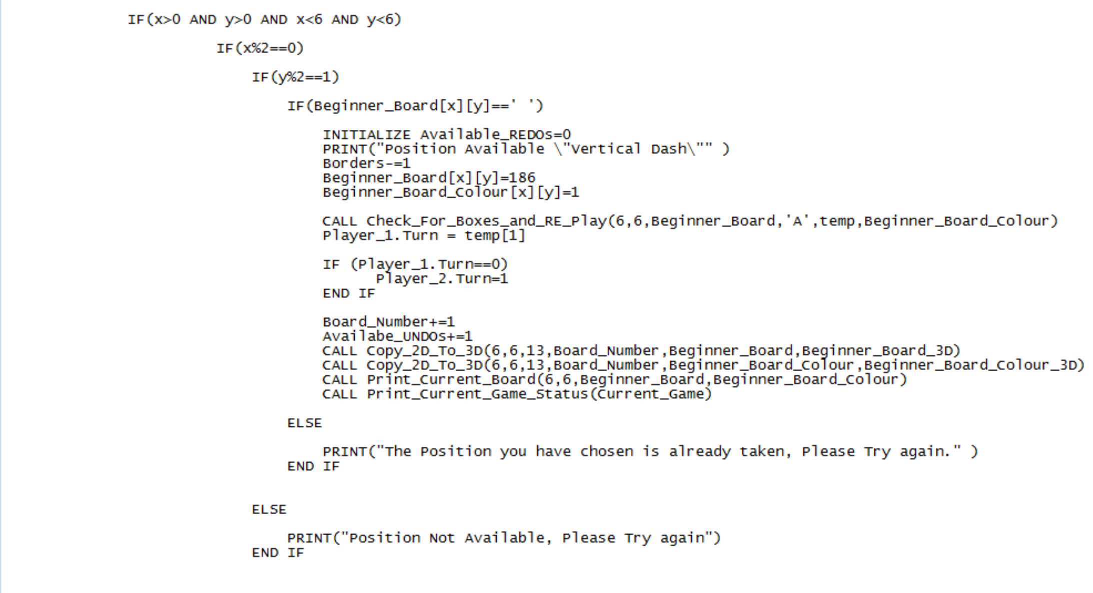
```
                END IF
      ELSE IF(x==R AND y==R)

                IF(Available_REDOs>0)

                    Borders-=1
                    Board_Number+=1
                    Availabe_UNDOs+=1
                    Available_REDOs-=1
                    CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_3D,Beginner_Board)
                    CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_Colour_3D,Beginner_Board_Colour)
                    CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
                    CALL Print_Current_Game_Status(Current_Game)

                ELSE

                    PRINT("ERROR: No Available Move can be REdone.")
                END IF


      ELSE IF(x==U AND y==U)

                IF(Availabe_UNDOs>0)

                    Borders+=1
                    Board_Number-=1
                    Availabe_UNDOs-=1
                    Available_REDOs+=1
                    CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_3D,Beginner_Board)
                    CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_Colour_3D,Beginner_Board_Colour)
                    CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
                    CALL Print_Current_Game_Status(Current_Game)

                ELSE

                    PRINT("ERROR: No Available Move can be undone.")
                END IF


      ELSE

            PRINT("Position is out of range, Please Try again.")
      END IF
     END WHILE
    END WHILE
             #PvP_Beginner_Mode_End
--------------------------------------------------------------------------------------------------------------------------------------------
```



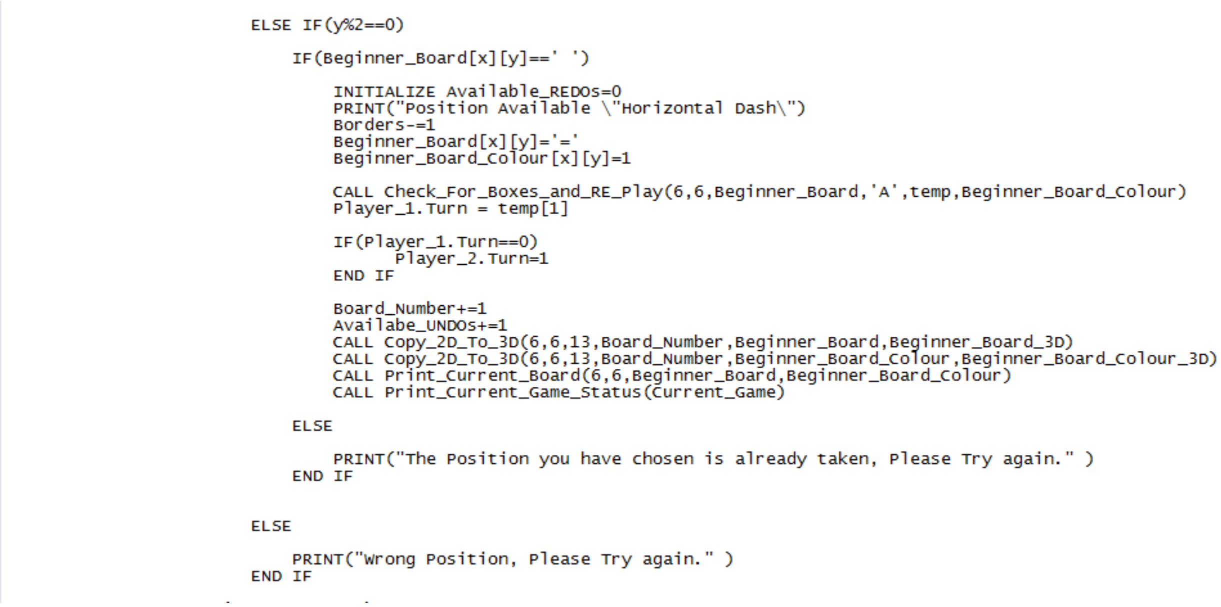
```
ELSE IF(d==2)        #PvP_Advanced_MOde_start
   #Predefined process
   We will do the same pesudocode but change 6-->12 , #dimensions of array
   change Beginner_Board --> Advanced_Board ,
   change Beginner_Board_Colour --> Advanced_Board_Colour ,
   change Borders=60 ,
   change Boxes=25 .
END IF

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



```
ELSE IF (o==2)
  THEN PRINT ("PvC Mode Has Been Chosen.");
       PRINT ("Please Type The Choice Number of The difficulty level You want to Play.")
       PRINT ("---------------------------------------------------------------------------")
       PRINT ("For Beginner Mode Press 1
               For Advanced Mode Press 2
               Enter: ")

       READ d
       IF (d==1)   #PvC_Beginner_Mode_start
         THEN
            INITIALIZE ARRAY Beginner_Board[6][6]={{}}
            CALL Initialize_PvC_Beginner_Board(6,6,Beginner_Board)
            INITIALIZE Beginner_Board_Colour[6][6]={{}}
            CALL Intialize_PvC_Beginner_Board_Colour(6,6,Beginner_Board_Colour)
            CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board,Beginner_Board_3D)
            CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board_Colour,Beginner_Board_Colour_3D)
            INITALIZE x,y
            INITIALIZE Borders = 12
            PRINT ("Welcome to \"Dots & Boxes Game\" Beginner Mode")
            CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
            CALL Print_Current_Game_Status(Current_Game)
            INITIALIE ARRAY Player_A_Name[100]
            PRINT("Please Enter Player A Name (FIRST NAME ONLY): ")
            READ(Player_A_Name)

            WHILE (Borders > 0)
             WHILE(Player_1.Turn==1 AND Borders > 0)
              INITIALIZE temp[2]={0,0}
              PRINT("Player A's Turn")
              PRINT("Enter The Index in the form X(Row) Y(Column): ")
              READ x,y

              IF(Current_Game.Remaining_Borders<12)
                 PRINT ("For UNDO Type \"U U\"
                         For REDO Type \"R R\"")

              END IF
```



```
IF(x>0 AND y>0 AND x<6 AND y<6)

          IF(x%2==0)

              IF(y%2==1)

                  IF(Beginner_Board[x][y]==' ')

                      INITIALIZE Available_REDOs=0
                      PRINT("Position Available \"Vertical Dash\"" )
                      Borders-=1
                      Beginner_Board[x][y]=186
                      Beginner_Board_Colour[x][y]=1

                      CALL Check_For_Boxes_and_RE_Play(6,6,Beginner_Board,'A',temp,Beginner_Board_Colour)
                      Player_1.Turn = temp[1]

                      IF (Player_1.Turn==0)
                            Player_2.Turn=1
                      END IF

                      Board_Number+=1
                      Availabe_UNDOs+=1
                      CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board,Beginner_Board_3D)
                      CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board_Colour,Beginner_Board_Colour_3D)
                      CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
                      CALL Print_Current_Game_Status(Current_Game)

                  ELSE

                      PRINT("The Position you have chosen is already taken, Please Try again." )
                  END IF


              ELSE

                  PRINT("Position Not Available, Please Try again")
              END IF
```



```
              ELSE IF(y%2==0)

                  IF(Beginner_Board[x][y]==' ')

                      INITIALIZE Available_REDOs=0
                      PRINT("Position Available \"Horizontal Dash\")
                      Borders-=1
                      Beginner_Board[x][y]='='
                      Beginner_Board_Colour[x][y]=1

                      CALL Check_For_Boxes_and_RE_Play(6,6,Beginner_Board,'A',temp,Beginner_Board_Colour)
                      Player_1.Turn = temp[1]

                      IF(Player_1.Turn==0)
                            Player_2.Turn=1
                      END IF

                      Board_Number+=1
                      Availabe_UNDOs+=1
                      CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board,Beginner_Board_3D)
                      CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board_Colour,Beginner_Board_Colour_3D)
                      CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
                      CALL Print_Current_Game_Status(Current_Game)

                  ELSE

                      PRINT("The Position you have chosen is already taken, Please Try again." )
                  END IF


              ELSE

                  PRINT("Wrong Position, Please Try again." )
              END IF
              ELSE IF(x==R AND y==R)

                        IF(Available_REDOs>0)

                            Borders-=1
                            Board_Number+=1
                            Availabe_UNDOs+=1
                            Available_REDOs-=1
                            CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_3D,Beginner_Board)
                            CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_Colour_3D,Beginner_Board_Colour)
                            CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
                            CALL Print_Current_Game_Status(Current_Game)

                        ELSE

                            PRINT("ERROR: No Available Move can be REdone.")
                        END IF


              ELSE IF(x==U AND y==U)

                        IF(Availabe_UNDOs>0)

                            Borders+=1
                            Board_Number-=1
                            Availabe_UNDOs-=1
                            Available_REDOs+=1
                            CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_3D,Beginner_Board)
                            CALL Copy_3D_To_2D(6,6,13,Board_Number,Beginner_Board_Colour_3D,Beginner_Board_Colour)
                            CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
                            CALL Print_Current_Game_Status(Current_Game)

                        ELSE

                            PRINT("ERROR: No Available Move can be undone.")
                        END IF


              ELSE

                    PRINT("Position is out of range, Please Try again.")
              END IF

             END WHILE


            WHILE(Player_2.Turn==1 AND Borders > 0)
             INITIALIZE temp[2]={0,0}
             CALL Computer_Play(6,6,4,Beginner_Board,Beginner_Board_Colour)
             Borders-=1
             CALL Check_For_Boxes_and_RE_Play(6,6,Beginner_Board,'B',temp,Beginner_Board_Colour)
             Player_2.Turn = temp[1]

             IF (Player_2.Turn==0)
               Player_1.Turn=1
             END IF

             CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board,Beginner_Board_3D)
             CALL Copy_2D_To_3D(6,6,13,Board_Number,Beginner_Board_Colour,Beginner_Board_Colour_3D)
             CALL Print_Current_Board(6,6,Beginner_Board,Beginner_Board_Colour)
             CALL Print_Current_Game_Status(Current_Game)

            END WHILE
           END WHILE

                          #PvC_Beginner_Mode_End
---------------------------------------------------------------------------------------------------------------------------
      ELSE IF(d==2)       #PvC_Advanced_Mode_start
         #Predefined process
         We will do the same pervious pesudocode but change 6-->12 , #dimensions of array
         change Beginner_Board --> Advanced_Board ,
         change Beginner_Board_Colour --> Advanced_Board_Colour ,
         change Borders=60 ,
         change Boxes=25 .
      END IF
                  #PvC_Advanced_Mode_End

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

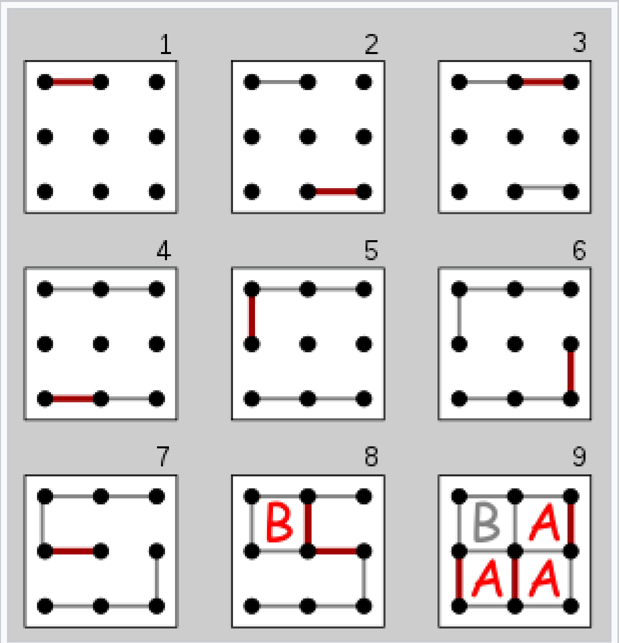
8) User Manual:

Game Description:

Dots and Boxes *is a pencil-and-paper game for two players (sometimes more). It was first published in the 19th century by French mathematician Édouard Lucas, who called it* ***la pipopipette****.[1] It has gone by many other names,[2] including the* ***dots and dashes****,* ***game of dots****,[3]* ***dot to dot grid****,[4]* ***boxes****,[5] and* ***pigs in a pen****.[6]*

- *The game starts with an empty grid of dots. Usually two players take turns adding a single horizontal or vertical line between two unjoined adjacent dots. A player who completes the fourth side of a 1×1 box earns one point and takes another turn. A point is typically recorded by placing a mark that identifies the player in the box, such as an initial. The game ends when no more lines can be placed. The winner is the player with the most points.[2][7] The board may be of any size grid. When short on time, or to learn the game, a 2×2 board (3×3 dots) is suitable.[8] A 5×5 board, on the other hand, is good for experts.[9]*

- *The diagram on the right shows a game being played on a 2×2 board (3×3 dots). The second player ("B") plays a rotated mirror image of the first player's moves, hoping to divide the board into two pieces and tie the game. But the first player ("A") makes a sacrifice at move 7 and B accepts the sacrifice, getting one box. However, B must now add another line, and so B connects the center dot to the center-right dot, causing the remaining unscored boxes to be joined together in a chain (shown at the end of move 8). With A's next move, A gets all three of them and ends the game, winning 3–1*

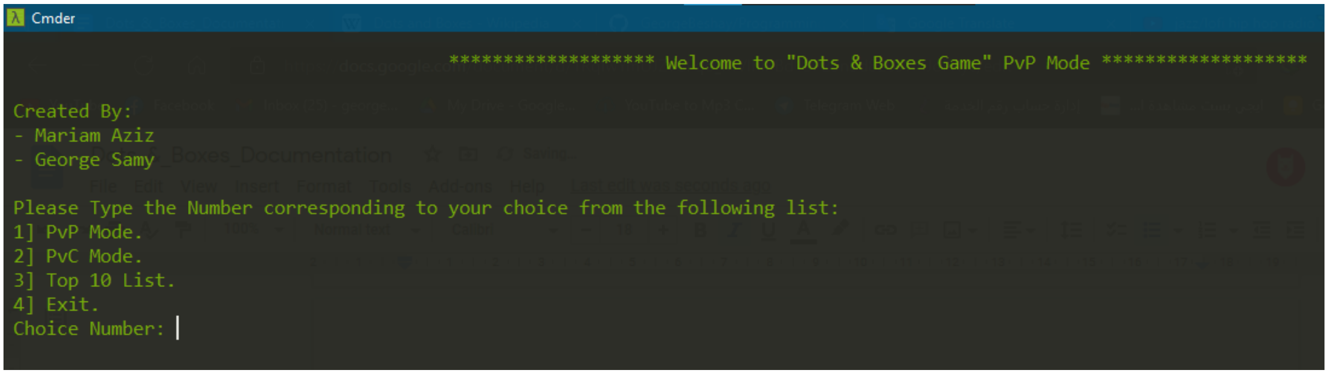


Example game of Dots and Boxes on a 2 × 2 square board

For our Project, Here is the Player Manual:

*1st. Run the **"Game.exe"***

2nd. Choose one of the 4 options displayed infront of you by entering the choice number.



```
                    ******************* Welcome to "Dots & Boxes Game" PvP Mode *******************

Created By:
- Mariam Aziz
- George Samy

Please Type the Number corresponding to your choice from the following list:
1] PvP Mode.
2] PvC Mode.
3] Top 10 List.
4] Exit.
Choice Number: |
```

For Player Vs Player Mode Enter "1"

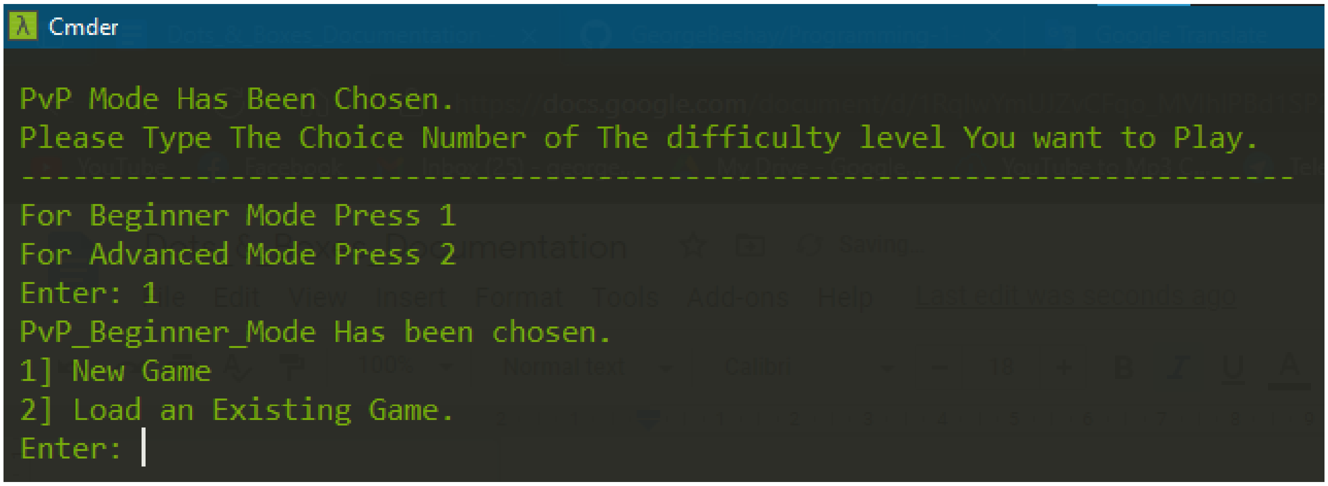
For Player Vs Computer Mode Enter "2"

To Check the Top 10 List Enter “3”

To Exit the Game Enter “4”

3rd. After choosing the PvP Mode or PvC Mode, Choose the difficulty level you want whether ***Beginner or Advanced***

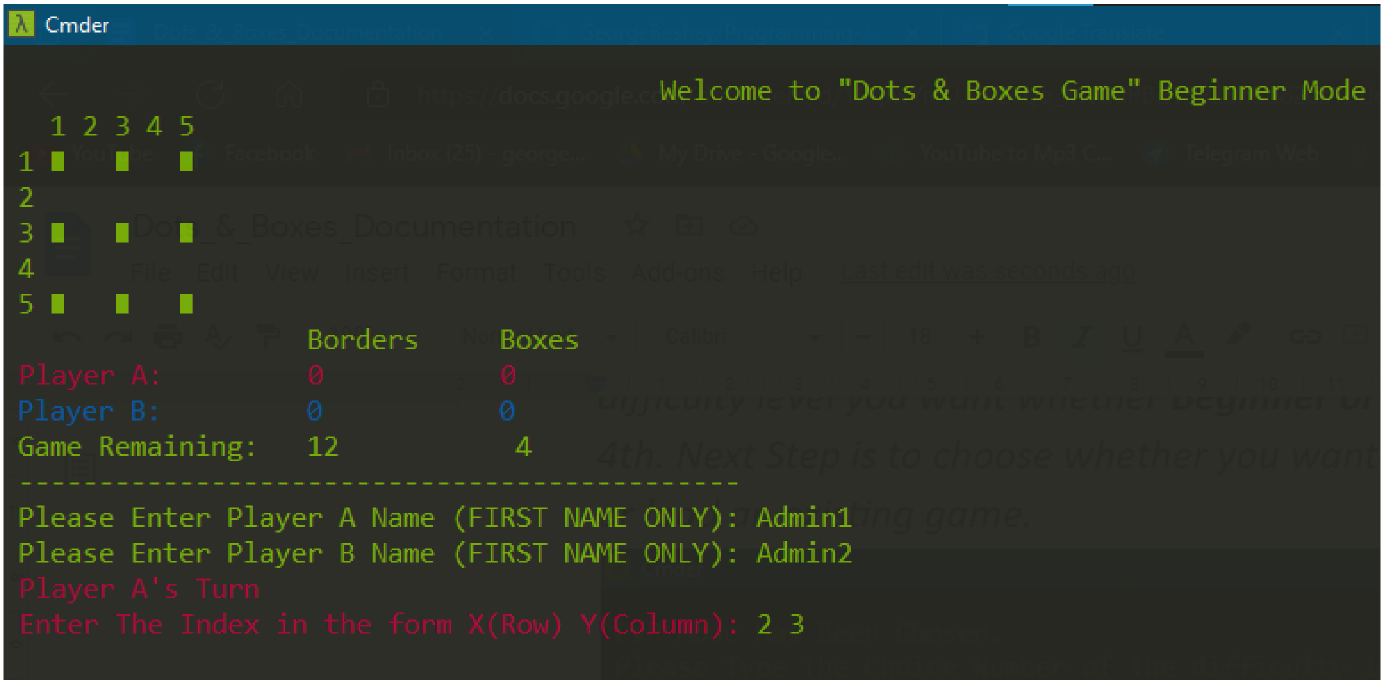
4th. Next Step is to choose whether you want to create a new game or load an existing game.



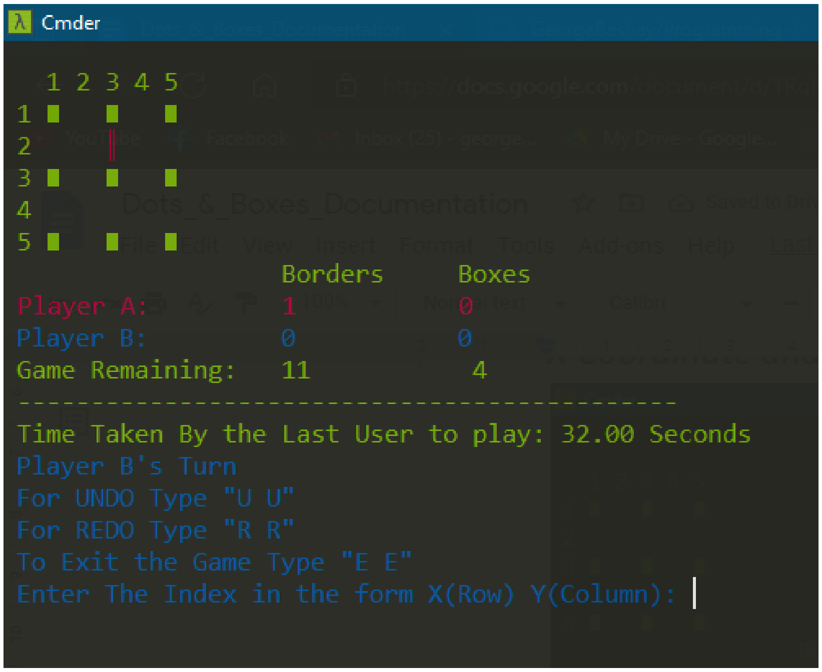
```
PvP Mode Has Been Chosen.
Please Type The Choice Number of The difficulty level You want to Play.
-------------------------------------------------------------------------
For Beginner Mode Press 1
For Advanced Mode Press 2
Enter: 1
PvP_Beginner_Mode Has been chosen.
1] New Game
2] Load an Existing Game.
Enter:
```

5th. How to Play & Win ..??

By entering the vertical dash or the horizontal dash co-ordinates from the displayed grid infront of you leaving a space between the X-Coordinate and the Y-Coordinate

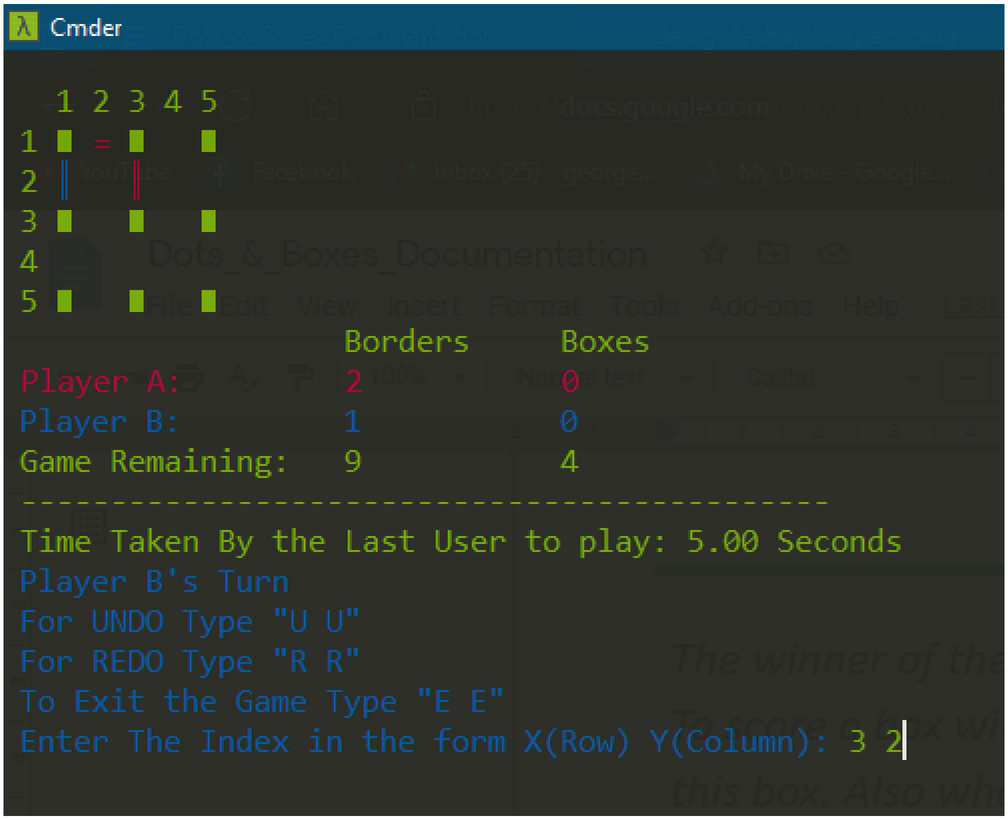


```
                                        Welcome to "Dots & Boxes Game" Beginner Mode
  1 2 3 4 5
1 ▮   ▮   ▮
2
3 ▮   ▮   ▮
4
5 ▮   ▮   ▮

                    Borders      Boxes
Player A:           0            0
Player B:           0            0
Game Remaining:     12            4
---------------------------------------------
Please Enter Player A Name (FIRST NAME ONLY): Admin1
Please Enter Player B Name (FIRST NAME ONLY): Admin2
Player A's Turn
Enter The Index in the form X(Row) Y(Column): 2 3
```

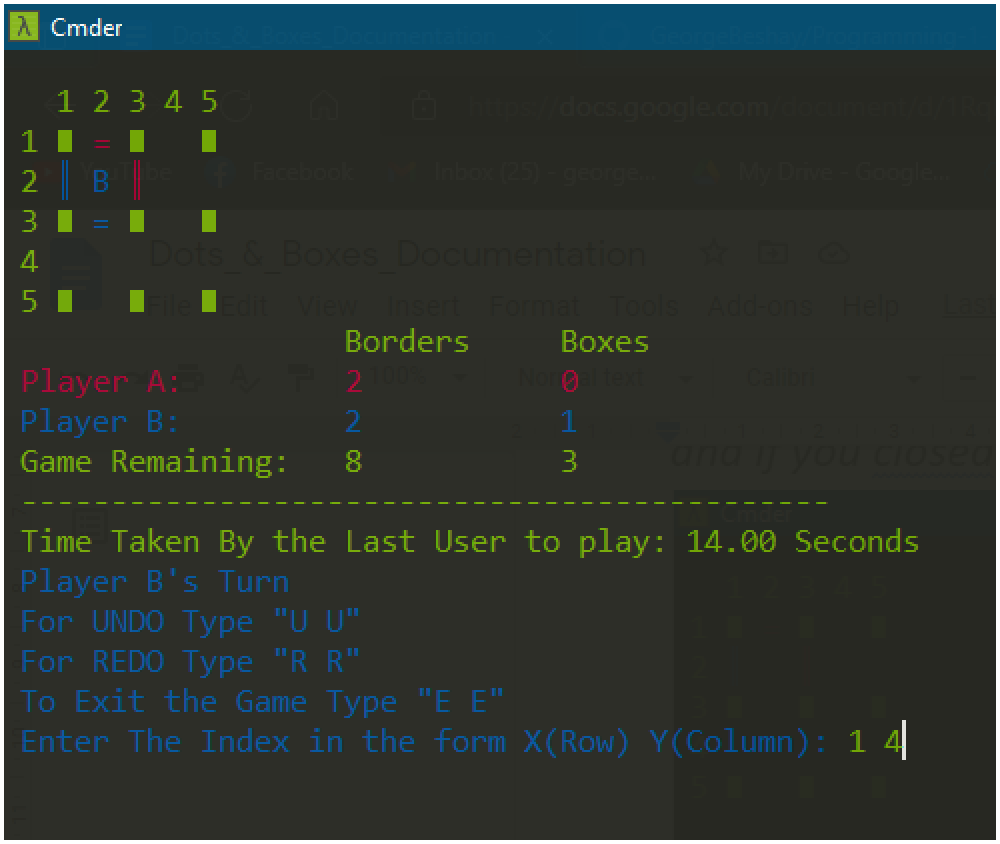


```
  1 2 3 4 5
1 ▮   ▮   ▮
2     ‖
3 ▮   ▮   ▮
4
5 ▮   ▮   ▮
                  Borders     Boxes
Player A:         1           0
Player B:         0           0
Game Remaining:   11           4
---------------------------------------------
Time Taken By the Last User to play: 32.00 Seconds
Player B's Turn
For UNDO Type "U U"
For REDO Type "R R"
To Exit the Game Type "E E"
Enter The Index in the form X(Row) Y(Column):
```

The winner of the game is identified by who is having more boxes. To score a box with your letter, you have to be the last player closing this box. Also when you close a box the gama turn will still be with you, so notice that after closing a box, you will still play once more, and if you closed one more box you will play again and again ….

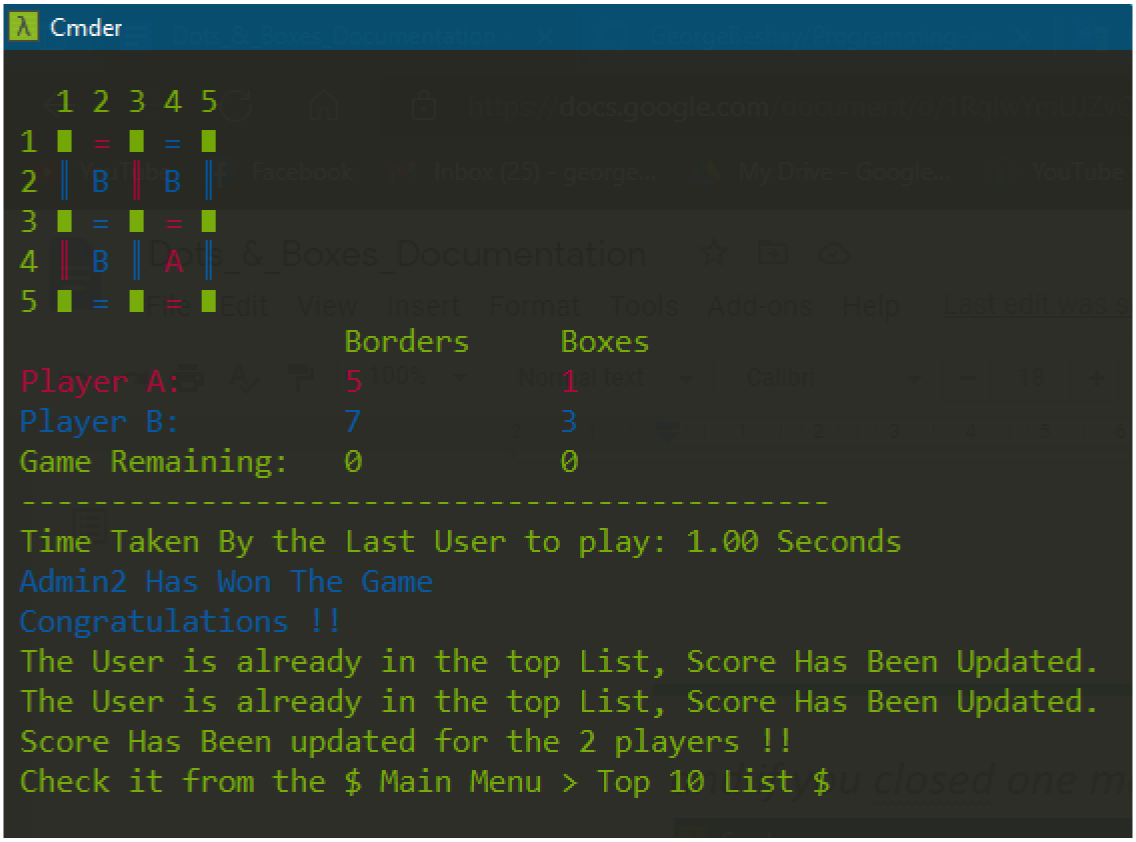


```
  1 2 3 4 5
1 ▮ = ▮   ▮
2 ║   ║
3 ▮   ▮   ▮
4
5 ▮   ▮   ▮
                  Borders     Boxes
Player A:         2           0
Player B:         1           0
Game Remaining:   9           4
---------------------------------------------
Time Taken By the Last User to play: 5.00 Seconds
Player B's Turn
For UNDO Type "U U"
For REDO Type "R R"
To Exit the Game Type "E E"
Enter The Index in the form X(Row) Y(Column): 3 2
```



```
  1 2 3 4 5
1 ▮ = ▮   ▮
2 ║ B ║
3 ▮ = ▮   ▮
4
5 ▮   ▮   ▮
                   Borders     Boxes
Player A:          2           0
Player B:          2           1
Game Remaining:    8           3
---------------------------------------------
Time Taken By the Last User to play: 14.00 Seconds
Player B's Turn
For UNDO Type "U U"
For REDO Type "R R"
To Exit the Game Type "E E"
Enter The Index in the form X(Row) Y(Column): 1 4
```

5th. To Undo or Redo, Enter the displayed message between the “” mark, for example, in the Beginner mode you should type “U U” to undo and “R R” To redo.
if there is no available move that can be undone/redone, dont worry! a message will be displayed saying that there is no move can be undone/redone.

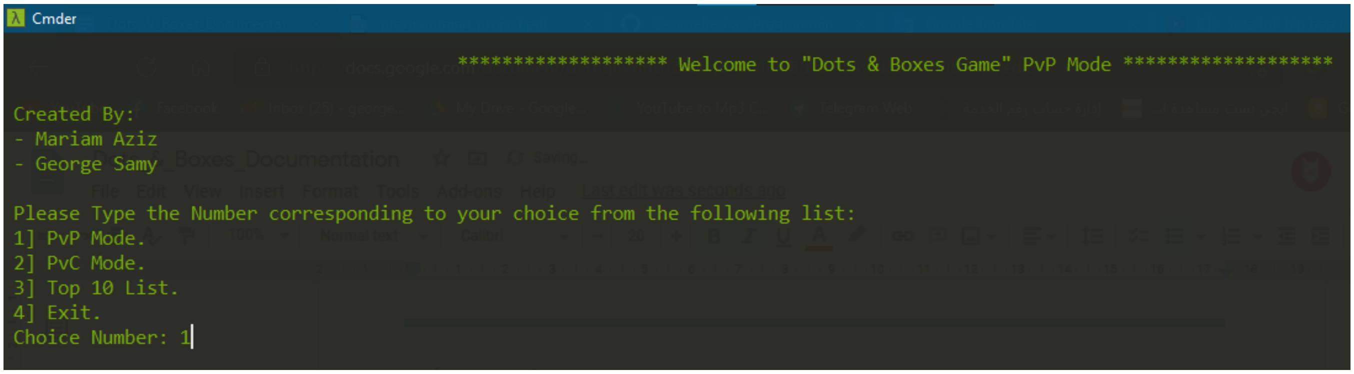


```
  1 2 3 4 5
1 █ ═ █ ═ █
2 ║ B ║ B ║
3 █ ═ █ ═ █
4 ║ B ║ A ║
5 █ ═ █ ═ █
                  Borders     Boxes
Player A:         5           1
Player B:         7           3
Game Remaining:   0           0
---------------------------------------------
Time Taken By the Last User to play: 1.00 Seconds
Admin2 Has Won The Game
Congratulations !!
The User is already in the top List, Score Has Been Updated.
The User is already in the top List, Score Has Been Updated.
Score Has Been updated for the 2 players !!
Check it from the $ Main Menu > Top 10 List $
```

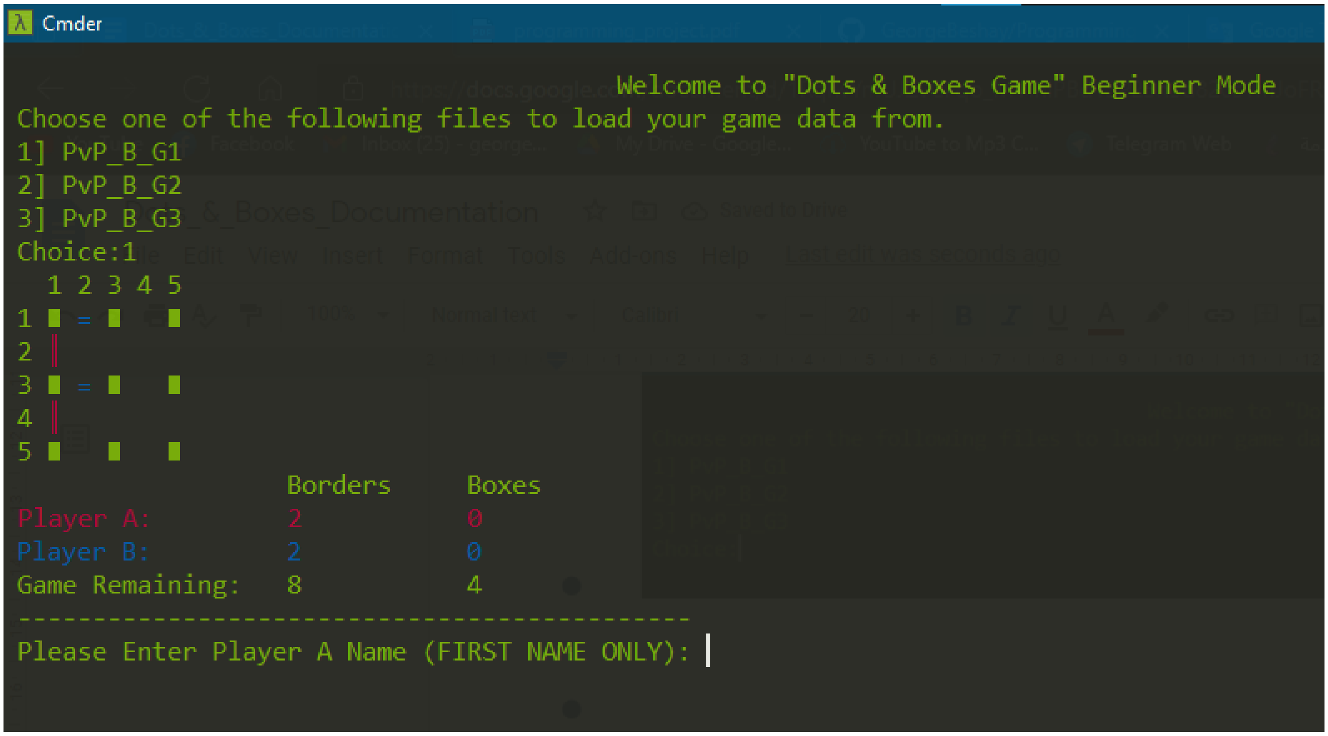
*6th. After the Game ends, **Automatically** both players' scores will be compared to the current top 10 list, if any of them has a score greater than the last person in the list, his name will be passed to the top 10 list and he can check it out from the main menu >> top 10 list >> mode.*

*7th. To exit the game while playing, enter **"E E"** >> choose whether to save the game or not.*

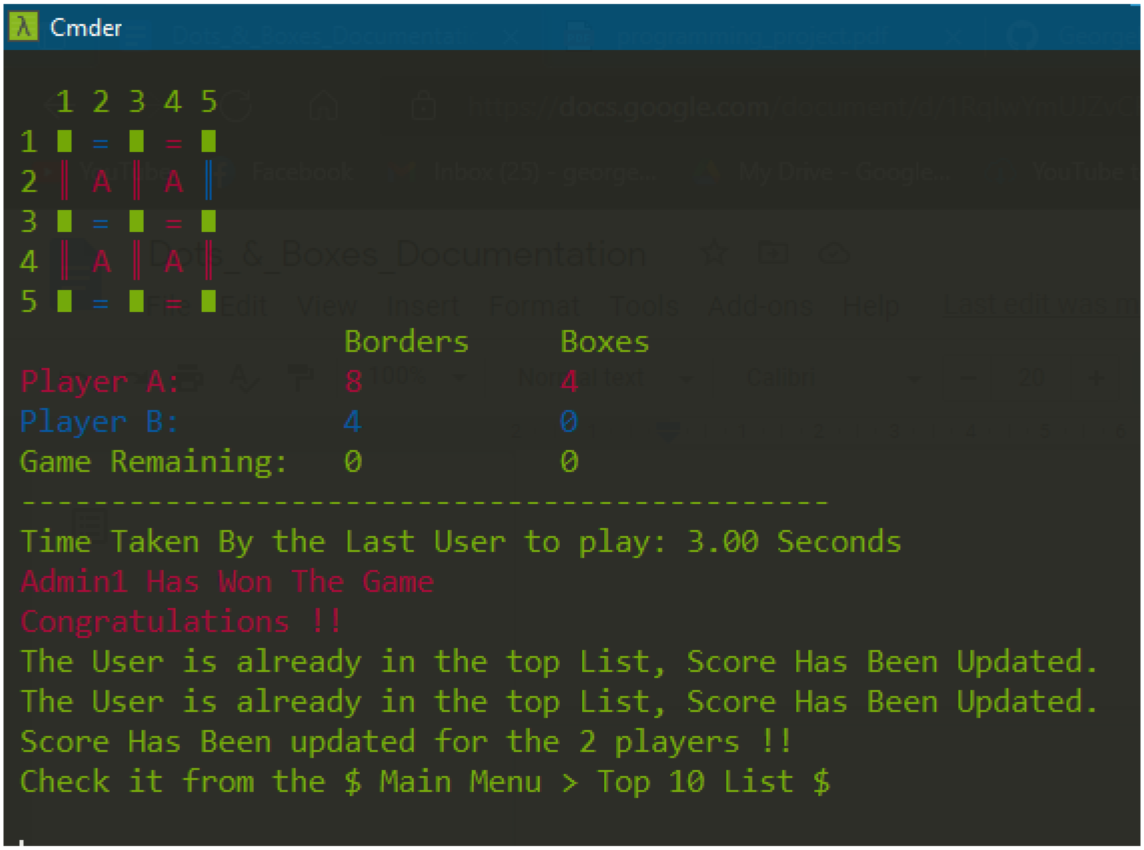
9) Sample runs:



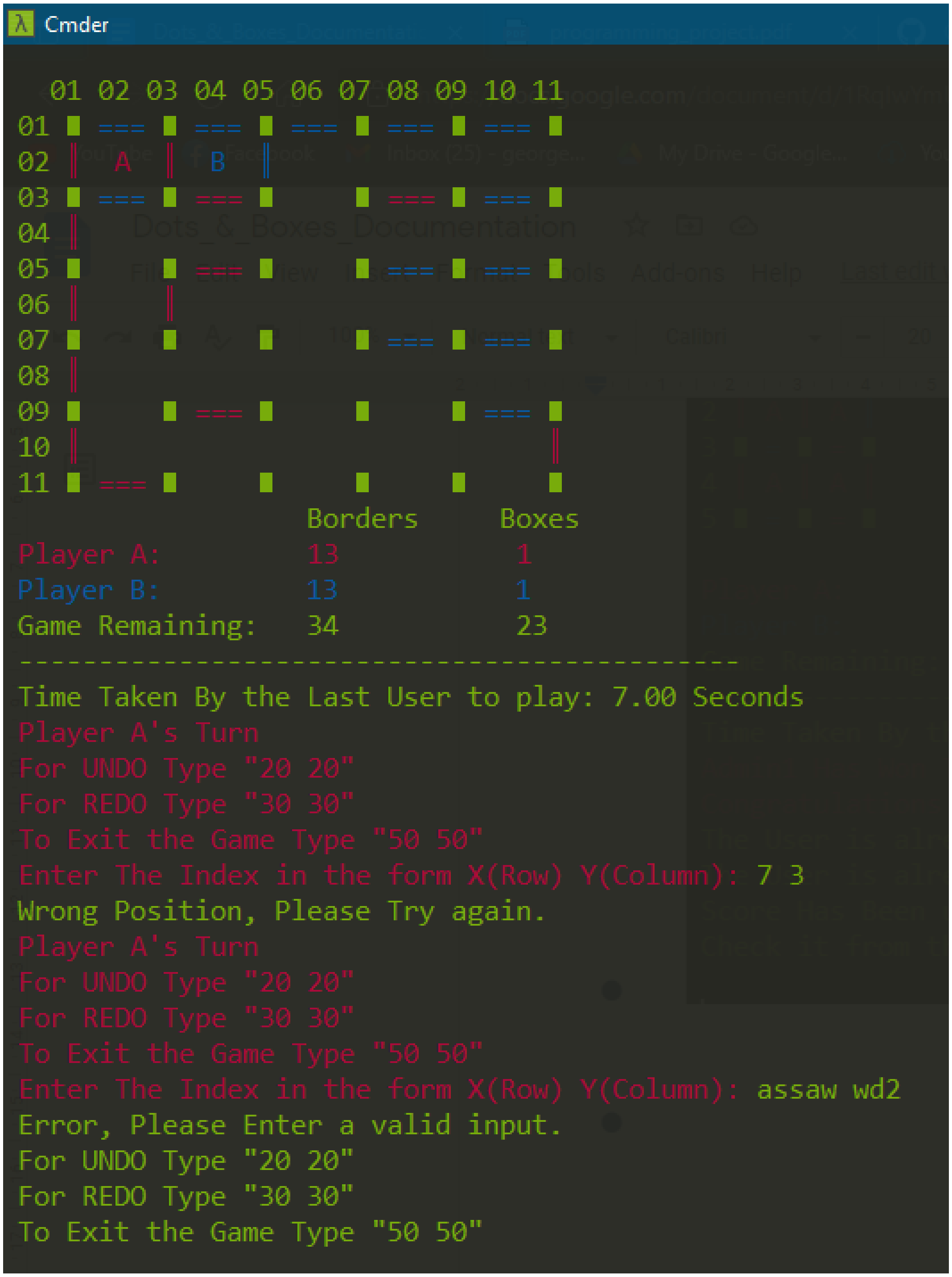
```
                    ****************** Welcome to "Dots & Boxes Game" PvP Mode ******************

Created By:
- Mariam Aziz
- George Samy

Please Type the Number corresponding to your choice from the following list:
1] PvP Mode.
2] PvC Mode.
3] Top 10 List.
4] Exit.
Choice Number: 1
```



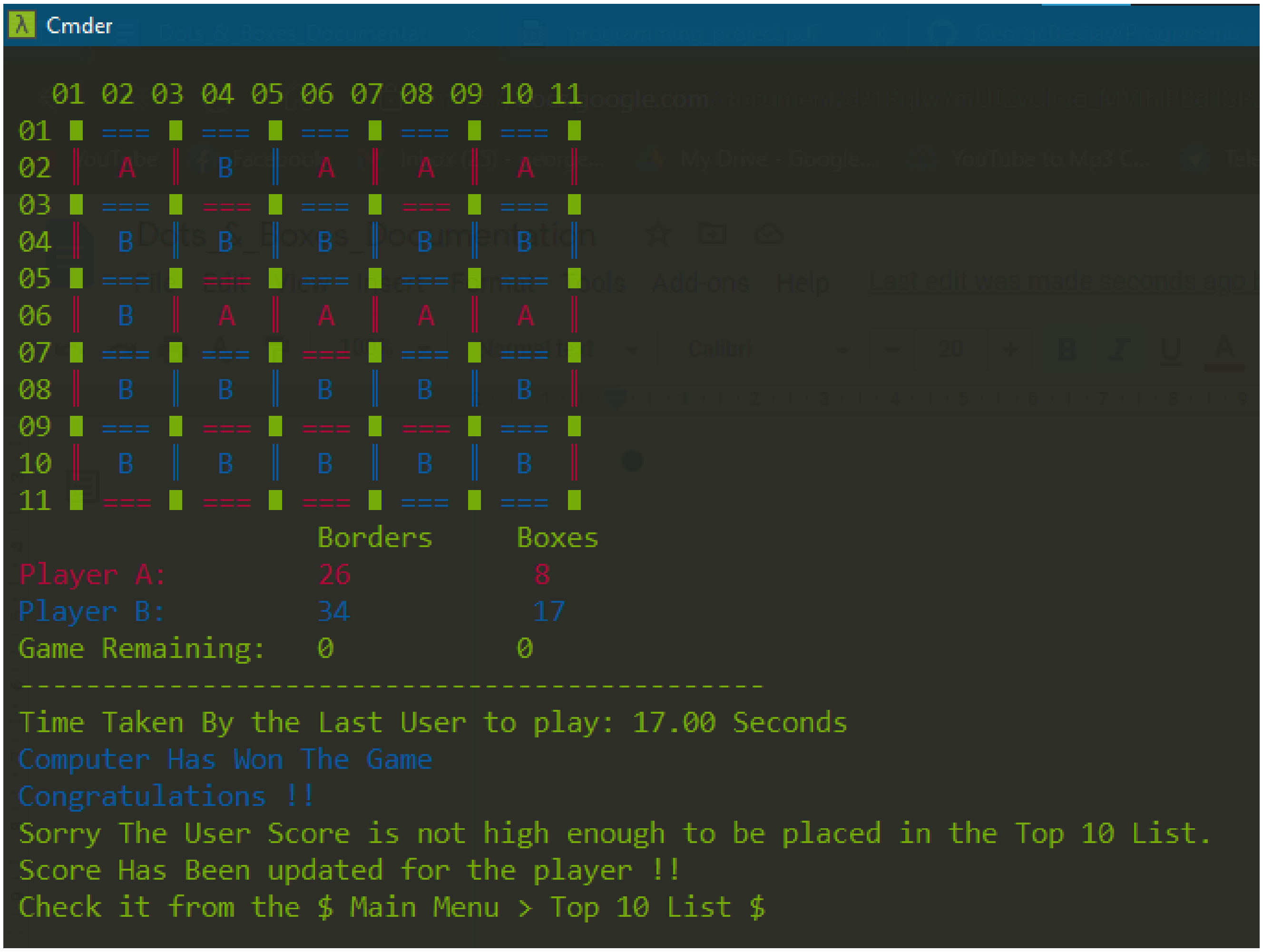
```
                                   Welcome to "Dots & Boxes Game" Beginner Mode
Choose one of the following files to load your game data from.
1] PvP_B_G1
2] PvP_B_G2
3] PvP_B_G3
Choice:1
  1 2 3 4 5
1 ▮ = ▮   ▮
2 ║
3 ▮ = ▮   ▮
4 ║
5 ▮   ▮   ▮
                  Borders     Boxes
Player A:         2           0
Player B:         2           0
Game Remaining:   8           4
---------------------------------------------
Please Enter Player A Name (FIRST NAME ONLY):
```



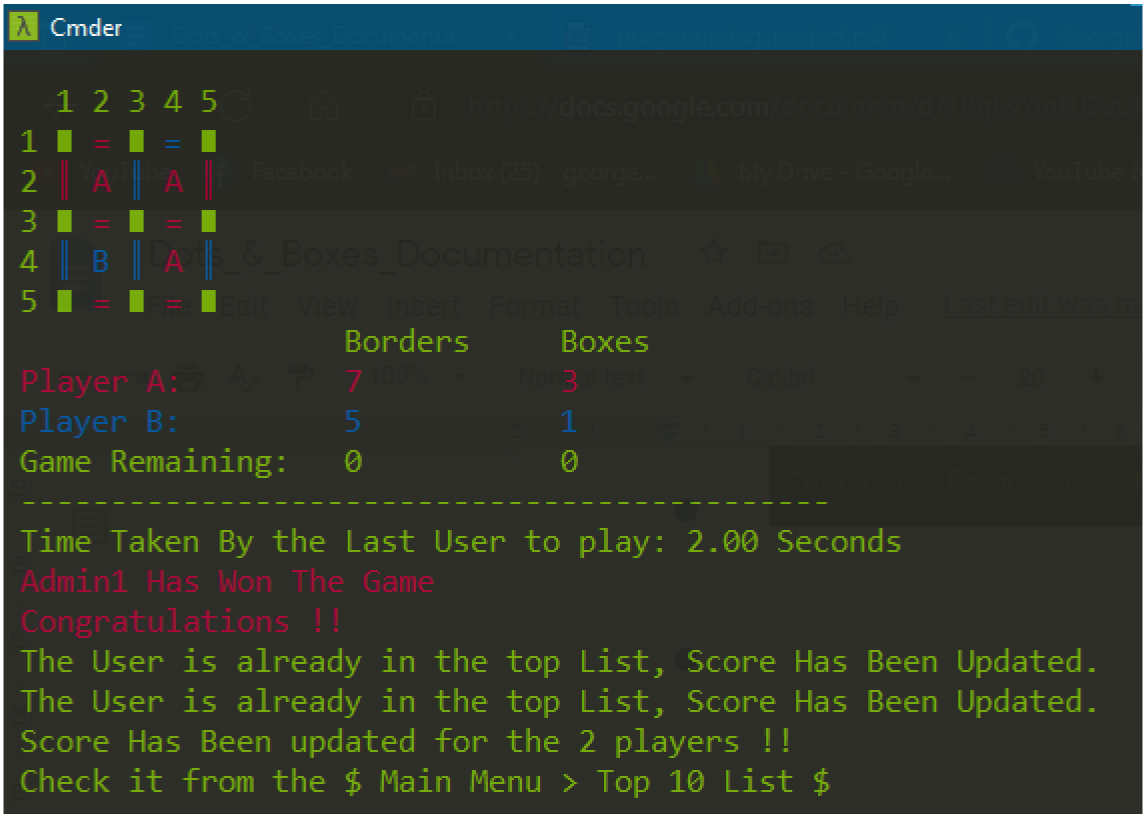
```
  1 2 3 4 5
1 ▮ = ▮ = ▮
2 ║ A ║ A ║
3 ▮ = ▮ = ▮
4 ║ A ║ A ║
5 ▮ = ▮ = ▮
                  Borders     Boxes
Player A:         8           4
Player B:         4           0
Game Remaining:   0           0
---------------------------------------------
Time Taken By the Last User to play: 3.00 Seconds
Admin1 Has Won The Game
Congratulations !!
The User is already in the top List, Score Has Been Updated.
The User is already in the top List, Score Has Been Updated.
Score Has Been updated for the 2 players !!
Check it from the $ Main Menu > Top 10 List $
```



```
  01 02 03 04 05 06 07 08 09 10 11
01 ▮ === ▮ === ▮ === ▮ === ▮ === ▮
02 ‖  A  ‖  B  ‖
03 ▮ === ▮ === ▮     ▮ === ▮ === ▮
04 ‖
05 ▮     ▮ === ▮     ▮ === ▮ === ▮
06 ‖     ‖
07 ▮     ▮     ▮     ▮ === ▮ === ▮
08 ‖
09 ▮     ▮ === ▮     ▮     ▮ === ▮
10 ‖                             ‖
11 ▮ === ▮     ▮     ▮     ▮     ▮
                  Borders     Boxes
Player A:         13           1
Player B:         13           1
Game Remaining:   34           23
---------------------------------------------
Time Taken By the Last User to play: 7.00 Seconds
Player A's Turn
For UNDO Type "20 20"
For REDO Type "30 30"
To Exit the Game Type "50 50"
Enter The Index in the form X(Row) Y(Column): 7 3
Wrong Position, Please Try again.
Player A's Turn
For UNDO Type "20 20"
For REDO Type "30 30"
To Exit the Game Type "50 50"
Enter The Index in the form X(Row) Y(Column): assaw wd2
Error, Please Enter a valid input.
For UNDO Type "20 20"
For REDO Type "30 30"
To Exit the Game Type "50 50"
```

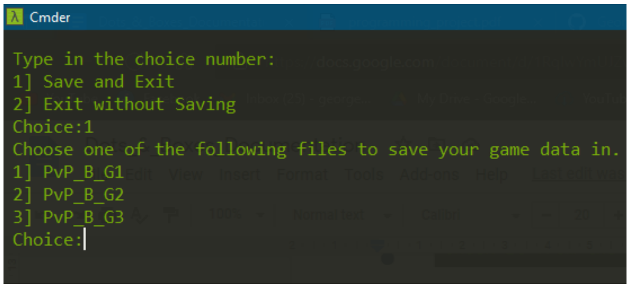


```
   01 02 03 04 05 06 07 08 09 10 11
01 █ === █ === █ === █ === █ === █
02 ║  A  ║  B  ║  A  ║  A  ║  A  ║
03 █ === █ === █ === █ === █ === █
04 ║  B  ║  B  ║  B  ║  B  ║  B  ║
05 █ === █ === █ === █ === █ === █
06 ║  B  ║  A  ║  A  ║  A  ║  A  ║
07 █ === █ === █ === █ === █ === █
08 ║  B  ║  B  ║  B  ║  B  ║  B  ║
09 █ === █ === █ === █ === █ === █
10 ║  B  ║  B  ║  B  ║  B  ║  B  ║
11 █ === █ === █ === █ === █ === █
                  Borders     Boxes
Player A:         26           8
Player B:         34           17
Game Remaining:   0           0
---------------------------------------------
Time Taken By the Last User to play: 17.00 Seconds
Computer Has Won The Game
Congratulations !!
Sorry The User Score is not high enough to be placed in the Top 10 List.
Score Has Been updated for the player !!
Check it from the $ Main Menu > Top 10 List $
```



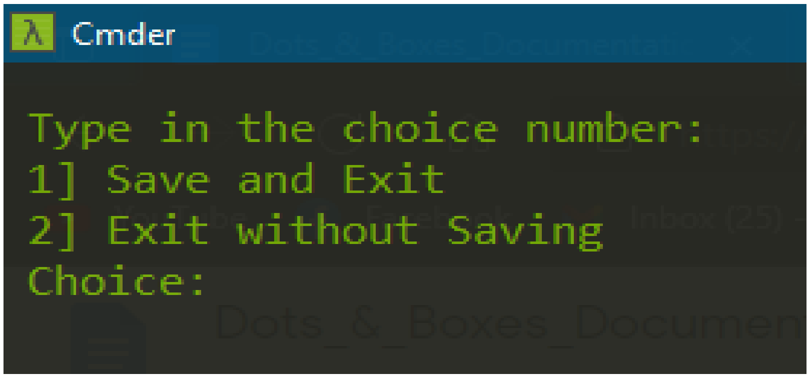
Cmder

```
  1 2 3 4 5
1 ▮ = ▮ = ▮
2 ║ A ║ A ║
3 ▮ = ▮ = ▮
4 ║ B ║ A ║
5 ▮ = ▮ = ▮
                  Borders      Boxes
Player A:         7            3
Player B:         5            1
Game Remaining:   0            0
---------------------------------------------
Time Taken By the Last User to play: 2.00 Seconds
Admin1 Has Won The Game
Congratulations !!
The User is already in the top List, Score Has Been Updated.
The User is already in the top List, Score Has Been Updated.
Score Has Been updated for the 2 players !!
Check it from the $ Main Menu > Top 10 List $
```



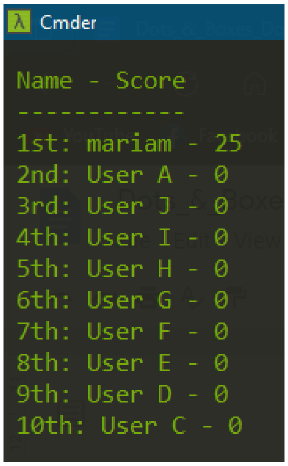
Cmder

```
Type in the choice number:
1] Save and Exit
2] Exit without Saving
Choice:1
Choose one of the following files to save your game data in.
1] PvP_B_G1
2] PvP_B_G2
3] PvP_B_G3
Choice:
```



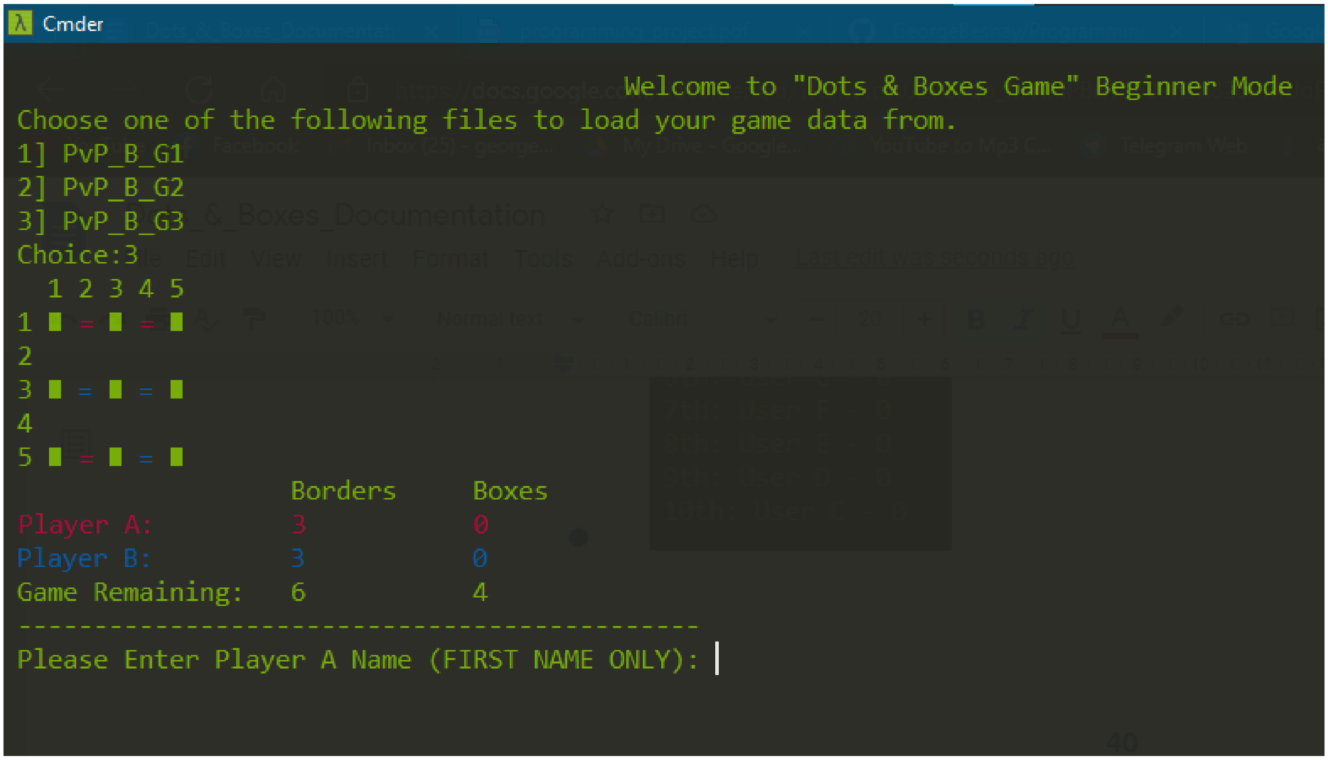
Cmder

```
Type in the choice number:
1] Save and Exit
2] Exit without Saving
Choice:
```

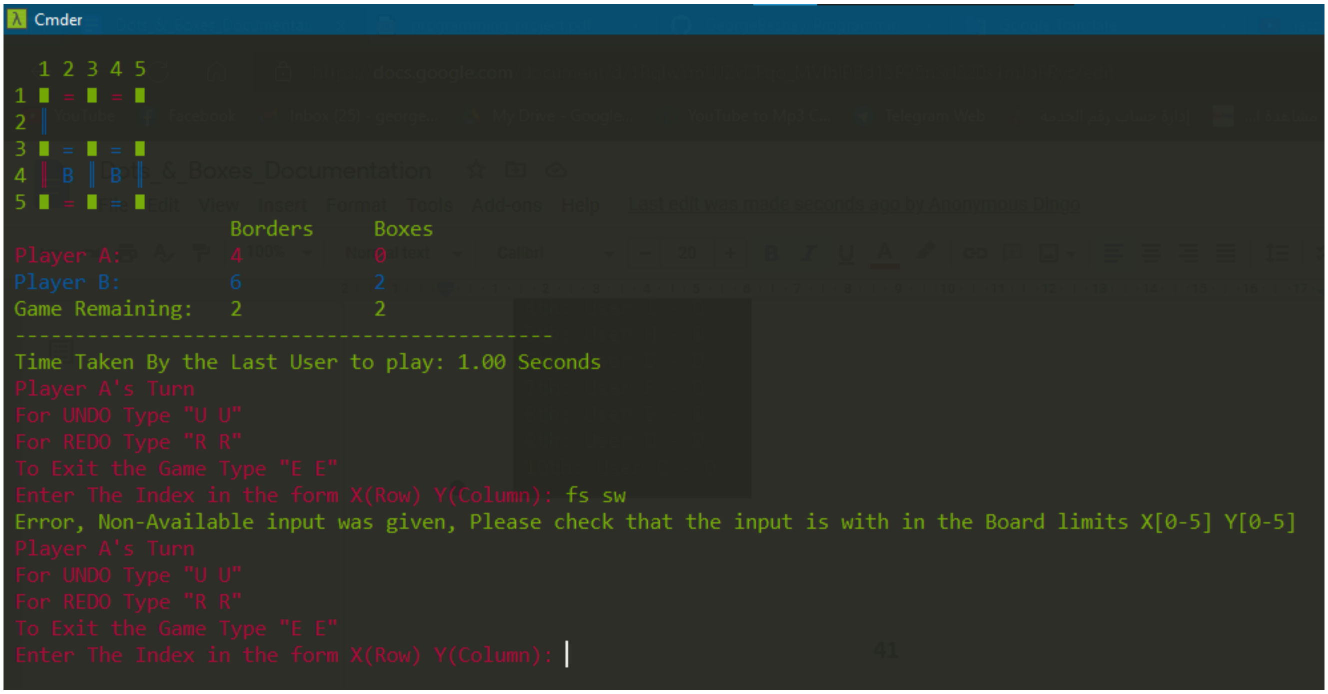


Cmder

```
Name - Score
------------
1st: mariam - 25
2nd: User A - 0
3rd: User J - 0
4th: User I - 0
5th: User H - 0
6th: User G - 0
7th: User F - 0
8th: User E - 0
9th: User D - 0
10th: User C - 0
```

- 

```
                                          Welcome to "Dots & Boxes Game" Beginner Mode
Choose one of the following files to load your game data from.
1] PvP_B_G1
2] PvP_B_G2
3] PvP_B_G3
Choice:3
  1 2 3 4 5
1 ▮ = ▮ = ▮
2
3 ▮ = ▮ = ▮
4
5 ▮ = ▮ = ▮
                  Borders     Boxes
Player A:         3           0
Player B:         3           0
Game Remaining:   6           4
---------------------------------------------
Please Enter Player A Name (FIRST NAME ONLY): 
```

- 

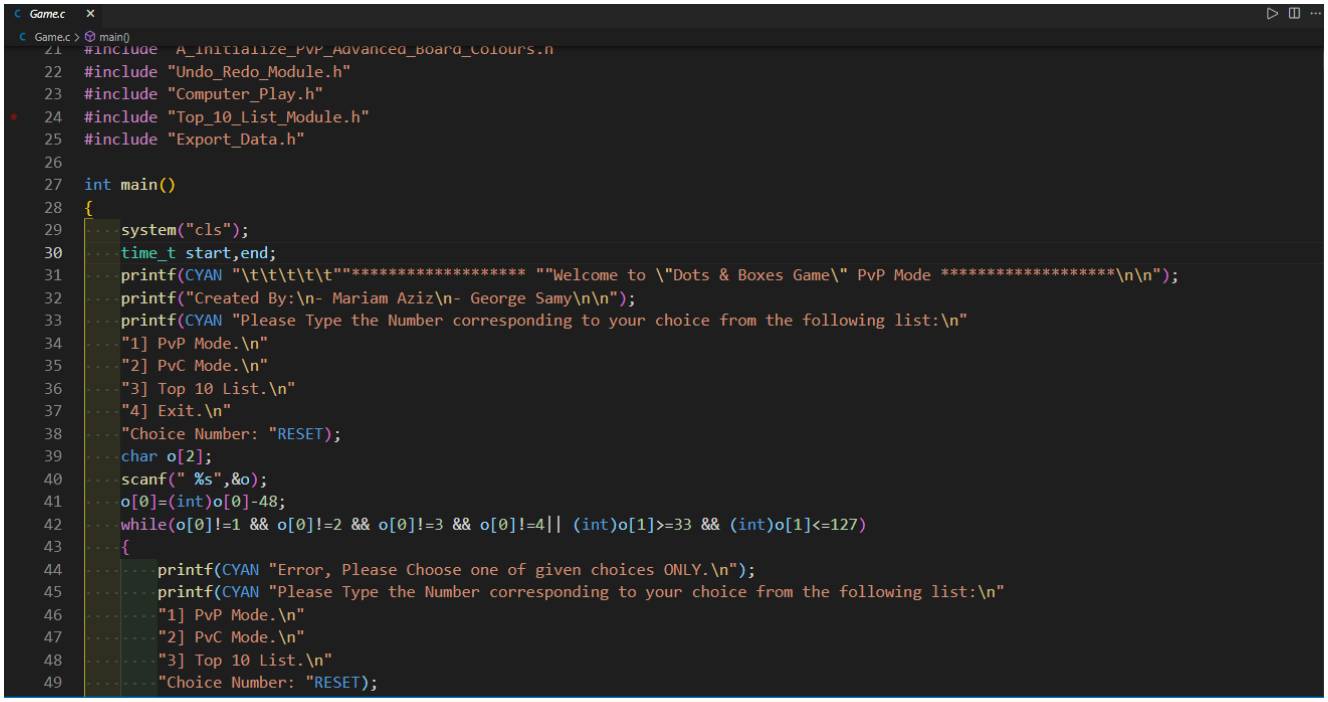
```
  1 2 3 4 5
1 ▮ = ▮ = ▮
2 ║
3 ▮ = ▮ = ▮
4 ║ B ║ B ║
5 ▮ = ▮ = ▮
                  Borders     Boxes
Player A:         4           0
Player B:         6           2
Game Remaining:   2           2
---------------------------------------------
Time Taken By the Last User to play: 1.00 Seconds
Player A's Turn
For UNDO Type "U U"
For REDO Type "R R"
To Exit the Game Type "E E"
Enter The Index in the form X(Row) Y(Column): fs sw
Error, Non-Available input was given, Please check that the input is with in the Board limits X[0-5] Y[0-5]
Player A's Turn
For UNDO Type "U U"
For REDO Type "R R"
To Exit the Game Type "E E"
Enter The Index in the form X(Row) Y(Column): 
```

10) References:

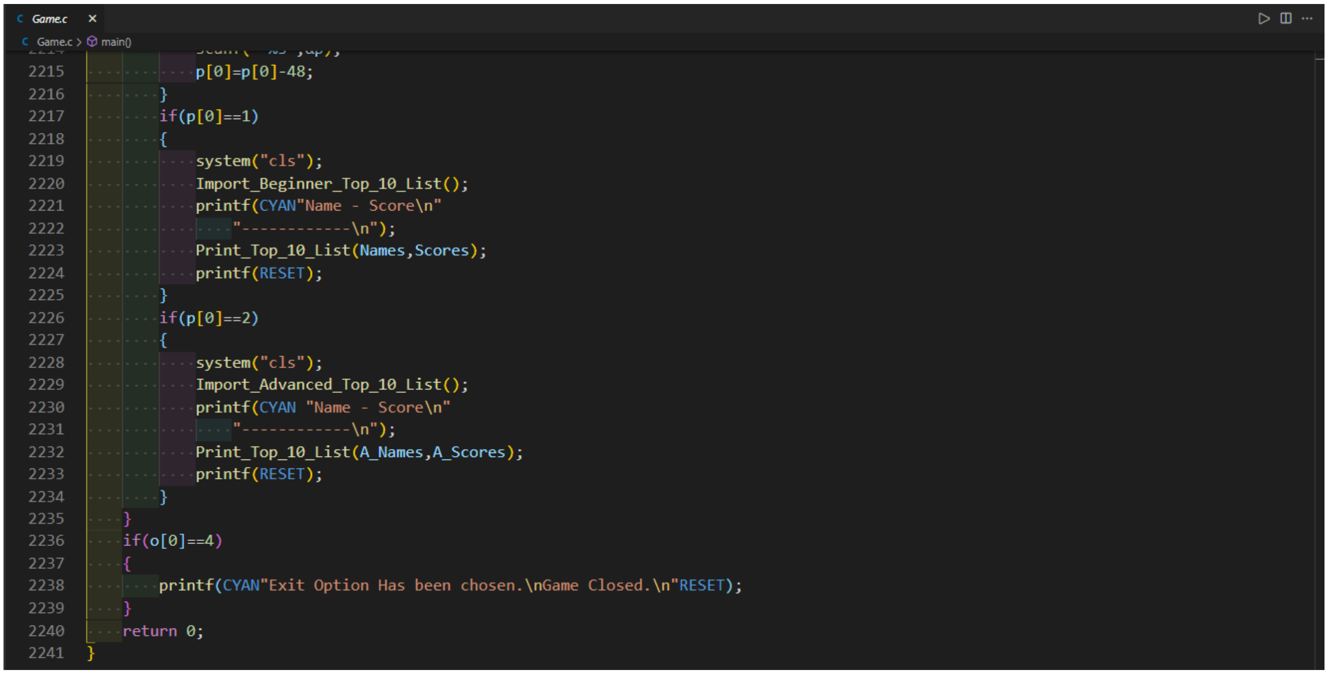
- *C Programming A Modern Approach, 2nd Edition by K. N. King*
- *Lectures Slides*
- *https://stackoverflow.com*
- *https://www.geeksforgeeks.org*
- *Dots and Boxes - Wikipedia*

11) Extra:

*Here are some random ScreenShots from the **“game.c”** file*



```c
#include "A_Initialize_PvP_Advanced_Board_Colours.h"
#include "Undo_Redo_Module.h"
#include "Computer_Play.h"
#include "Top_10_List_Module.h"
#include "Export_Data.h"

int main()
{
    system("cls");
    time_t start,end;
    printf(CYAN "\t\t\t\t\t""****************** ""Welcome to \"Dots & Boxes Game\" PvP Mode ******************\n\n");
    printf("Created By:\n- Mariam Aziz\n- George Samy\n\n");
    printf(CYAN "Please Type the Number corresponding to your choice from the following list:\n"
    "1] PvP Mode.\n"
    "2] PvC Mode.\n"
    "3] Top 10 List.\n"
    "4] Exit.\n"
    "Choice Number: "RESET);
    char o[2];
    scanf(" %s",&o);
    o[0]=(int)o[0]-48;
    while(o[0]!=1 && o[0]!=2 && o[0]!=3 && o[0]!=4|| (int)o[1]>=33 && (int)o[1]<=127)
    {
        printf(CYAN "Error, Please Choose one of given choices ONLY.\n");
        printf(CYAN "Please Type the Number corresponding to your choice from the following list:\n"
        "1] PvP Mode.\n"
        "2] PvC Mode.\n"
        "3] Top 10 List.\n"
        "Choice Number: "RESET);
```



```c
            scanf(" %s",&p);
            p[0]=p[0]-48;
        }
        if(p[0]==1)
        {
            system("cls");
            Import_Beginner_Top_10_List();
            printf(CYAN"Name - Score\n"
                "-----------\n");
            Print_Top_10_List(Names,Scores);
            printf(RESET);
        }
        if(p[0]==2)
        {
            system("cls");
            Import_Advanced_Top_10_List();
            printf(CYAN "Name - Score\n"
                "-----------\n");
            Print_Top_10_List(A_Names,A_Scores);
            printf(RESET);
        }
    }
    if(o[0]==4)
    {
        printf(CYAN"Exit Option Has been chosen.\nGame Closed.\n"RESET);
    }
    return 0;
}
```

END OF REPORT