**School of Engineering**

**(FAR WESTERN UNIVERSITY)**

**Mahendranagar, Kanchanpur**

****

**[ Subject Code: CT 125]**

A MINOR PROJECT REPORT ON

**“Block-Games”**

|  |  |
| --- | --- |
| **Submitted By:**  **Anuradha Bhatt 3**  **Dipa Joshi 12**  **Dipak Raj Giri 13**  **Menuka Paneru 25** | **Submitted to:**  **Er. Birendra Singh Dhami**  **from**  **Department of Engineering, FWU** |

**04/2079**

# ABSTRACT

Block-Games is a computer action game, whose goal is to provide an intelligent board to response to the actions by user. In this we develop a board based on side selection functions considering smoothness, different color for different objects, and representative symbol. Scores given by the functions are aggregated by sum of total blocks, and the player who makes the greater number of blocks will be the winner. To perform these tasks, we examine several algorithm variants of different crossover and environmental selection operators. And if needed overloaded operators too.

# ACKNOWLEDGEMENT

The success of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of our second semester project work whatever we have done is only due to t0 such guidance and we would not forgot to thank them. Firstly, we would like to thank institute of engineering for including the project work as the part of curriculum which helps to explore our knowledge practically. Then special thanks to our programming-teacher Er. Birendra Singh Dhami whose valuable guidance has been the one that helped us to patch this project and make it easier. His suggestion and instruction have served as the major contributor toward the completion of execution module of this project. Last but not the least we would like to thank to the department of computer engineering and HOD Er. Toran Prashad Bhatt for providing us with all faculty that was required.

Table of Contents

[ABSTRACT i](#_Toc108821384)

[ACKNOWLEDGEMENT ii](#_Toc108821385)

[LIST OF FIGURES v](#_Toc108821386)

[INTRDOUCTION 1](#_Toc108821387)

[1.1 Background 1](#_Toc108821388)

[1.2 Problem statement 2](#_Toc108821389)

[1.3 Objectives 3](#_Toc108821390)

[1.4 Application 3](#_Toc108821391)

[1.5 Game Features 3](#_Toc108821392)

[1.6 Feasibility Analysis 4](#_Toc108821393)

[1.6.1 Economic Feasibility 4](#_Toc108821394)

[1.6.2 Technical Feasibility 4](#_Toc108821395)

[1.6.3 Operational Feasibility 4](#_Toc108821396)

[1.7 System Requirements 5](#_Toc108821397)

[1.7.1 Hardware Requirements 5](#_Toc108821398)

[1.7.2 Software Requirements 5](#_Toc108821399)

[LITERATURE REVIEW 6](#_Toc108821400)

[2.1 Paper Pencil Game 6](#_Toc108821401)

[2.2 Game Development 6](#_Toc108821402)

[METHODOLOGY 8](#_Toc108821403)

[3.1 Analysis 8](#_Toc108821404)

[3.2 Process 9](#_Toc108821405)

[3.2.1 Graphics Library Functions 9](#_Toc108821406)

[3.2.2 Click Action 10](#_Toc108821407)

[3.2.3 Advanced Game 11](#_Toc108821408)

[3.2.4 Help 12](#_Toc108821409)

[3.2.5 About 13](#_Toc108821410)

[3.2.6 Classic 14](#_Toc108821411)

[RESULT CONCLUSION AND RECOMMENDATION 15](#_Toc108821412)

[4.1 Results 15](#_Toc108821413)

[4.2 Limitations 19](#_Toc108821414)

[4.3 Problem Faced 19](#_Toc108821415)

[4.4 Conclusion 19](#_Toc108821416)

[4.5 Future Scope and Recommendation 19](#_Toc108821417)

[References 20](#_Toc108821418)

# LIST OF FIGURES

[**Figure 1 (2,2) Dots and Boxes game** 2](#_Toc108821308)

[**Figure 2 Tic Tac Toe** 7](#_Toc108821309)

[**Figure 3 Advanced Board** 11](#_Toc108821310)

[**Figure 4 Help Section** 12](#_Toc108821311)

[**Figure 5 About Section** 13](#_Toc108821312)

[**Figure 6 Classic Board** 14](#_Toc108821313)

[**Figure 7 User Interface** 15](#_Toc108821314)

[**Figure 8 Advance Game** 15](#_Toc108821315)

[**Figure 9 Intermediate Game** 16](#_Toc108821316)

[**Figure 10 Game Ending** 17](#_Toc108821317)

[**Figure 11 Classic Game** 18](#_Toc108821318)

# 

**CHAPTER 1**

# INTRDOUCTION

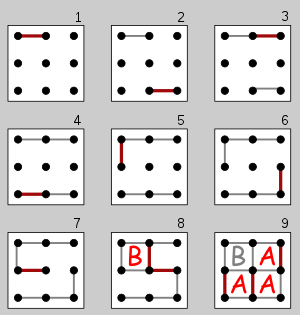
## 1.1 Background

**BG** stands for Block-Games. This game is modified version of **Dots and Boxes** game. This is two player paper pencil game with two different representative symbols as for an example let us take knought and cross. The player who succeeds in making a greater number of blocks in the given play board wins the game. It is an example of zero-sum game i.e.; one person’s gain is equivalent to another loss.

Let’s us first know about Dots and Boxes game. This game was invented by French Mathematician Edward Lucas in 1891, who called it ***pipopipette***. **Dots and Boxes** is a two player (sometimes more) game that using paper and pencil. At earlier time, this game is very popular in European, being taken as the project of international computer Olympic tournament game. It has gone by many other names, including the **dots and dashes**, **game of dots**, **dot to dot grid**, **boxes**,[]](https://en.wikipedia.org/wiki/Dots_and_Boxes#cite_note-5) and **pigs in a pen**.

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. 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. A 5×5 board, on the other hand, is good for experts.

The diagram below 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.



**Figure 1 (2,2) Dots and Boxes game**

So now we have taken some changes to represent it in graphical board and make it easier to play and look attractive. We have named Block-Games to the to this project. The changes will be discussed in methodology section.

## 1.2 Problem statement

* The problem or what we have to do is to design the Block-Games which provides following functionalities,
* Both the players can choose any line (except the line already chosen) on their respective turn.
* A box should be completed when the its four sides are selected.
* The point should give to the player who makes the complete box and when any player makes a box, he will get a bonus chance too.
* The respective symbol should be printed when a player completes a box.

## 1.3 Objectives

* To make this game playable in PC/Laptop/tablet.
* To make the game easy to play.
* To make the game look attractive.
* To make this game with some operations automated.
* To show the concepts of OOP, C++ and how it is used in real world problems.

## 1.4 Application

* For entertainment by kids sometimes youths too.
* To study concepts of OOP and C++.
* To create games similar to it having some additional features**.**

## 1.5 Game Features

* This game will have welcome animation.
* This game allows players to enter their name
* This game will provide toss mechanism for the player for the first turn
* This game will itself count the points of each player and declare winner itself.

## 1.6 Feasibility Analysis

### 1.6.1 Economic Feasibility

* To make this game we have used an PC with normal requirement so we don’t need any funds for the hardware.
* To make this game we have used the software which are free.
* For the research purpose all the data and information are available in google or internet so no investment to hire a researcher.
* This game is available for everyone for free on the GitHub with source code for study purpose.

### 1.6.2 Technical Feasibility

* Since this game is made with C++, it is platform independent.
* The executable file of this game is made for windows OS so this executable is dependent on Windows OS.
* This game can be played on PC/Laptop/Tablet with windows OS, for other OS we have to use that source code to build the respective executable for that OS.

### 1.6.3 Operational Feasibility

* The user interface of the game is very simple and easy to understand.
* There is help section for user which will guide the newbie.
* The game is tested very well and the bugs are fixed.
* The requirements to run the game are normal so it a run smoothly.

## 1.7 System Requirements

### 1.7.1 Hardware Requirements

* PC/Laptop/Tablet
* Processor: 1.4 Ghz,32 bit or more
* Ram: 512MB or more
* HDD or SSD: compatible with processor
* Display: (700 x 750) Capable video adapter and monitor
* Mouse or Touch screen

### 1.7.2 Software Requirements

* + Operating system: windows 7, windows 10, windows 8, windows xp.
  + Compiler tdm\_gcc 32 bit (included graphics .h)
  + IDE: Codeblocks, Vscode, dev c++, turbo C++ etc.

**CHAPTER 2**

# LITERATURE REVIEW

## 2.1 Paper Pencil Game

Paper-and-pencil games or paper-and-pen games (or some variation on those terms) are [games](https://en.wikipedia.org/wiki/Game) that can be played solely with [paper](https://en.wikipedia.org/wiki/Paper) and [pencils](https://en.wikipedia.org/wiki/Pencil) (or other [writing implements](https://en.wikipedia.org/wiki/Writing_implement)), usually without erasing. Some popular examples of pencil-and-paper games include [Tic-tac-toe](https://en.wikipedia.org/wiki/Tic-tac-toe), [Sprouts](https://en.wikipedia.org/wiki/Sprouts_(game)), [Dots and Boxes](https://en.wikipedia.org/wiki/Dots_and_Boxes), [Hangman](https://en.wikipedia.org/wiki/Hangman_(game)), [M.A.S.H.](https://en.wikipedia.org/wiki/MASH_(game)), [Paper soccer](https://en.wikipedia.org/wiki/Paper_soccer), and [Spellbinder](https://en.wikipedia.org/wiki/Spellbinder_(paper-and-pencil_game)). The term is unrelated to the use in [role-playing games](https://en.wikipedia.org/wiki/Role-playing_game) to differentiate [tabletop games](https://en.wikipedia.org/wiki/Tabletop_role-playing_game) from [role-playing video games](https://en.wikipedia.org/wiki/Role-playing_video_game).

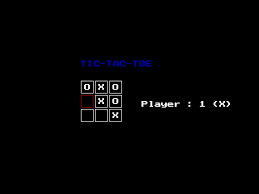
[Board games](https://en.wikipedia.org/wiki/Board_game) where pieces are never moved or removed from the board once being played, particularly [abstract strategy games](https://en.wikipedia.org/wiki/Abstract_strategy_game) like [Gomoku](https://en.wikipedia.org/wiki/Gomoku) and [Connect Four](https://en.wikipedia.org/wiki/Connect_Four), can also be played as pencil-and-paper games.

# 2.2 Game Development

Game Development is **the art of creating games and describes the design, development and release of a game**. It may involve concept generation, design, build, test and release. Generally, now days games are built in game development softwares like **Unity, Godot, GameMaker, Amazon Lumberyard** etc. There are now days all the games are developing for computers i.e., playable on computer devices. Tic tac toe is the game mostly developed by beginners. Tic tac toe is (3,3) board game which is generally build with the concept of array or string or linked structure (using self-referential class).

The game which we are going to build is not so much familiar for beginner developers though there are some developers who had developed this game using javascript, python. Since we are in coding and we don’t have understanding of these programming languages, we don’t have any reference for coding. But we have found form other game projects that for graphics we can use graphics.h library, SDL, OpenGl, Allegro, Cinder, SFML etc. Out of these graphics.h is easy to use for beginners and it helps beginners to have good understanding of Pixels, Graphics. Graphics.h library is not available in the modern compilers but we can add it and easily use it whereas other libraries provide more features but they are not beginner friendly.

Tic Tac Toe game Using grapics.h library:



**Figure 2 Tic Tac Toe**

Mainly we are inspired form our childhood days when we used to play this game with friends.

**CHAPTER 3**

# METHODOLOGY

## 3.1 Analysis

Following are the important points after analysis:

* To build this game we have used C++ with graphics.h library.
* To make board we used shape making classes for creating rectangle, circle, square etc. with required color.
* To store different entities, we used objects/object array/object pointers of required class.
* We have created methods to create logic the logic to perform the actions.
* It uses concept of classes and objects so it divides the game into small pieces using classes and functions.
* To represent the sides, we make the objects of classes and give them some functions to operate on this.
* File handling is used in about and the help section.
* Graphics is used for showing the output which is a predefined library.

## 3.2 Process

The game is divided into four sections as, Advanced, help, about and classic.

Main

Help

Advanced

Classic

About

### 3.2.1 Graphics Library Functions

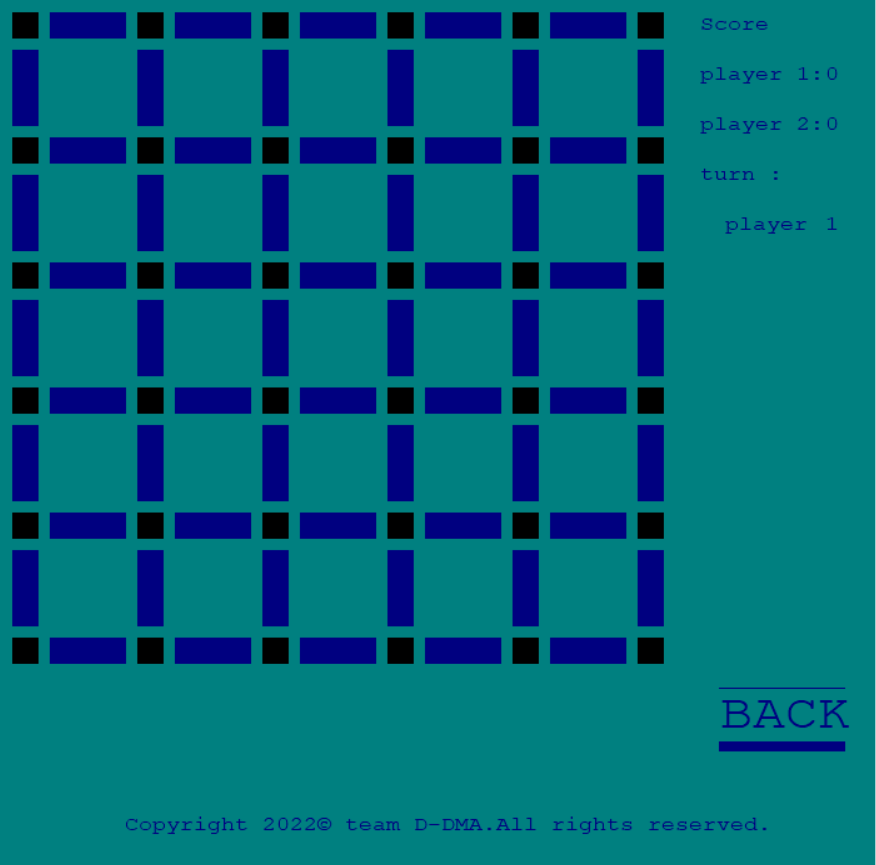
* initwindow (), setbkcolor ()
* rectangle(args), circle(args)
* setcolor (), setfillstyle *()*, *floodfill* ()
* outtextxy (), settextcolor (), readimagefile ()

### 3.2.2 Click Action

* POINT cursorpos;
* GetCursorPos(&cursorpos);
* x = cursorpos.x;
* y = cursorpos. y;
* GetAsyncKeyState (VK\_LBUTTON)
* The cursorpos is used to store the position of the cursor.
* The GetCursorPos () is used to get the position of cursor.
* The GetAsyncKeyState () is used to get mouse click.
* After getting the click with internal game logic the game responds.

### 3.2.3 Advanced Game

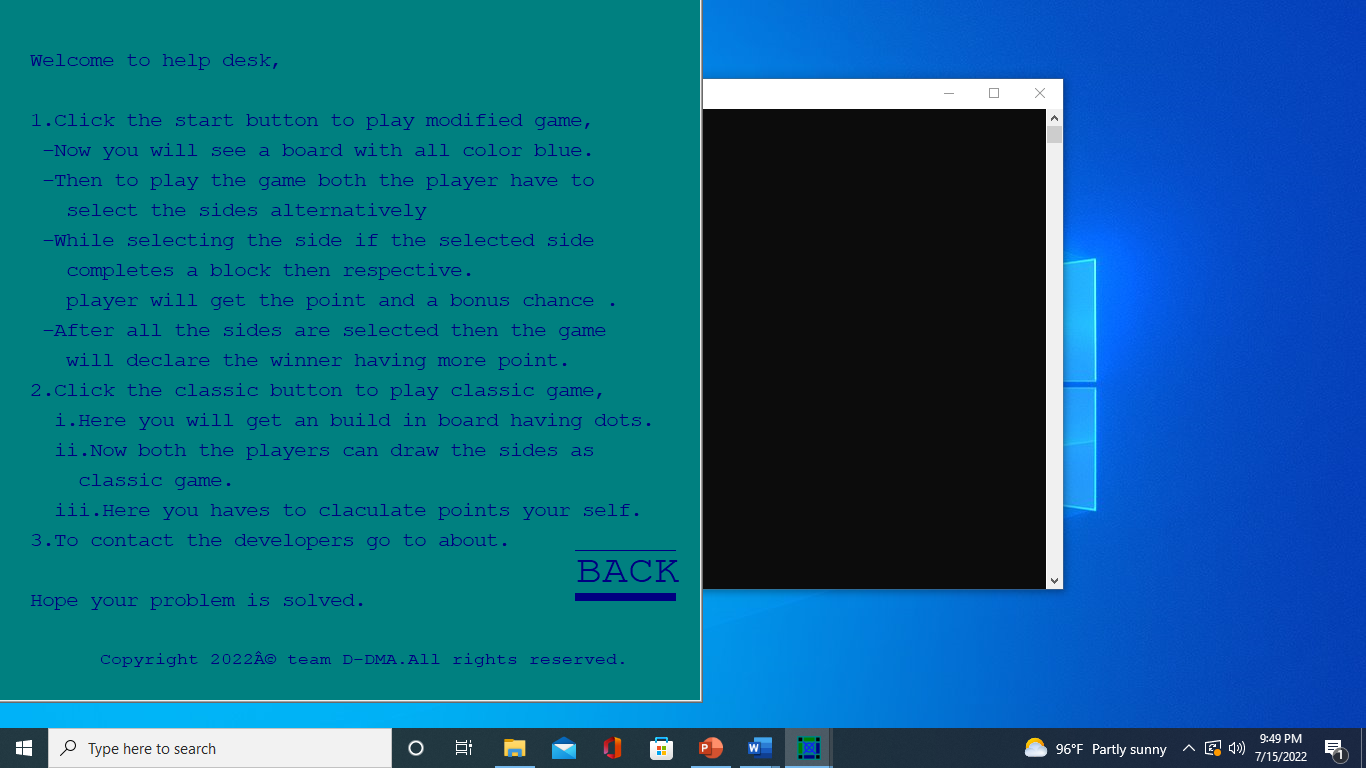
The advance board contains horizontal and vertical rectangles which are the objects having attributes color, coordinate and number. At the right corner there is score shown and the players turn. At the bottom there is a back button. This can be done with the classes, objects(array), static data member, control statements, loops, functions, inheritance pointers(this), graphics, library functions etc.



**Figure 3 Advanced Board**

### 3.2.4 Help

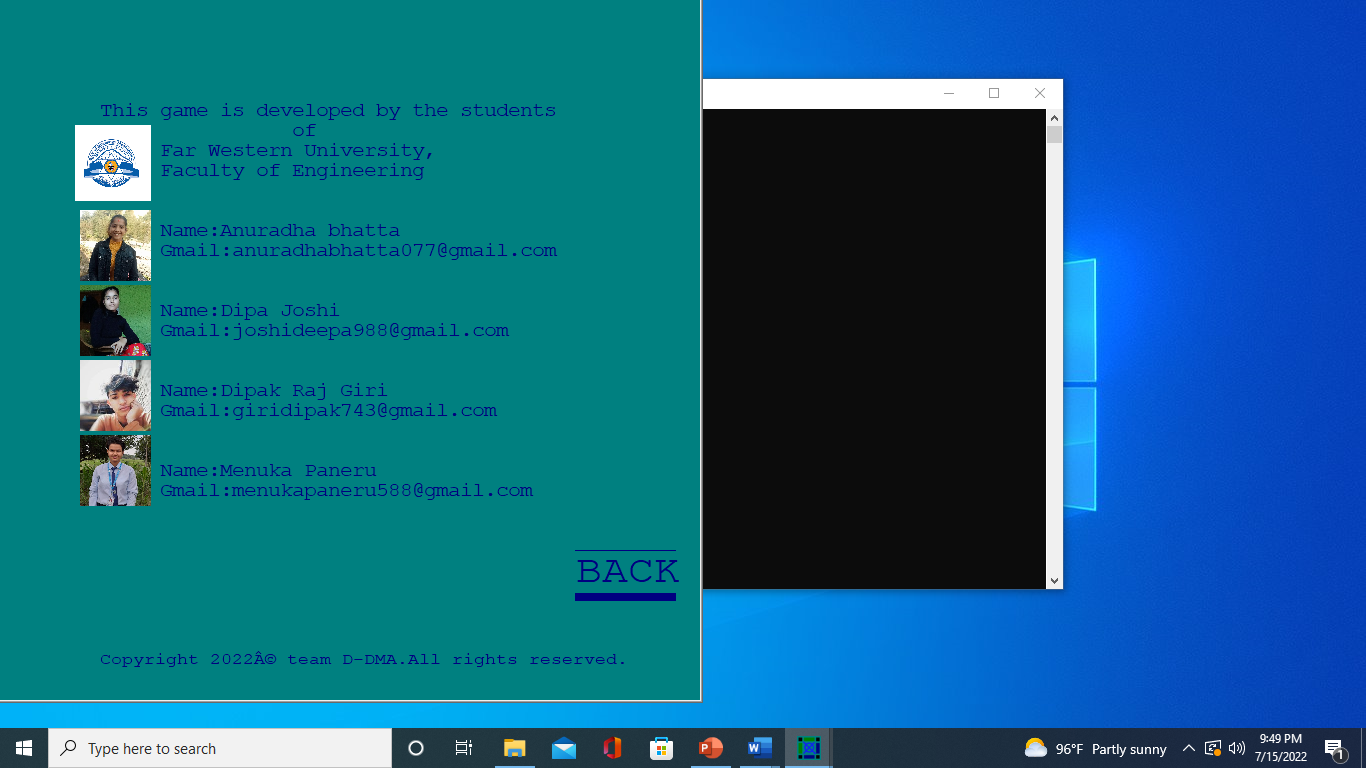
This section contains the expected information to assist the player.To create this section we have used C++ basic concepts, file handling and graphics.



**Figure 4 Help Section**

### 3.2.5 About

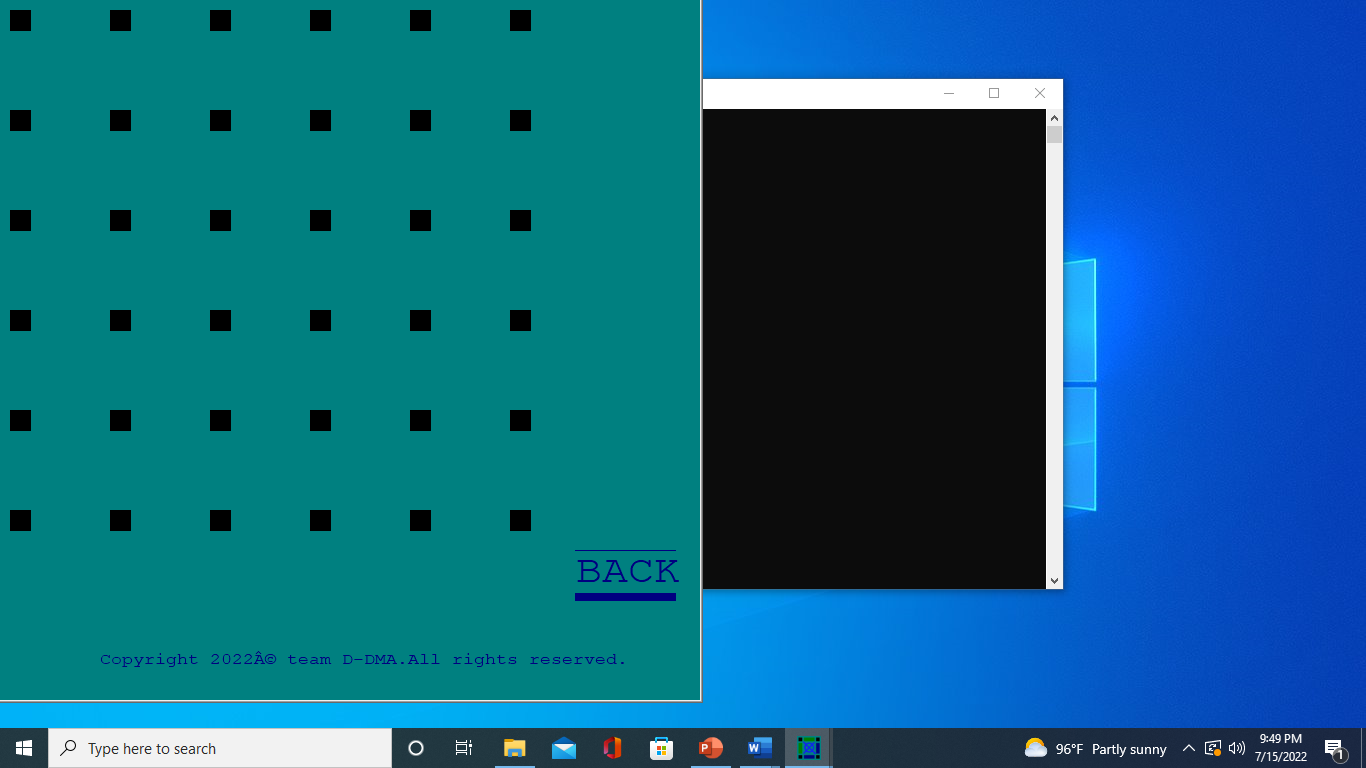
This section contains the expected information about developers .To create this section we have used C++ basic concepts, file handling and graphics.



**Figure 5 About Section**

### 3.2.6 Classic

This board allow the player to play the game as classic dots-and-boxes game.To create this we have used the C++ basic concepts, inheritance(hierarchial) and graphics.

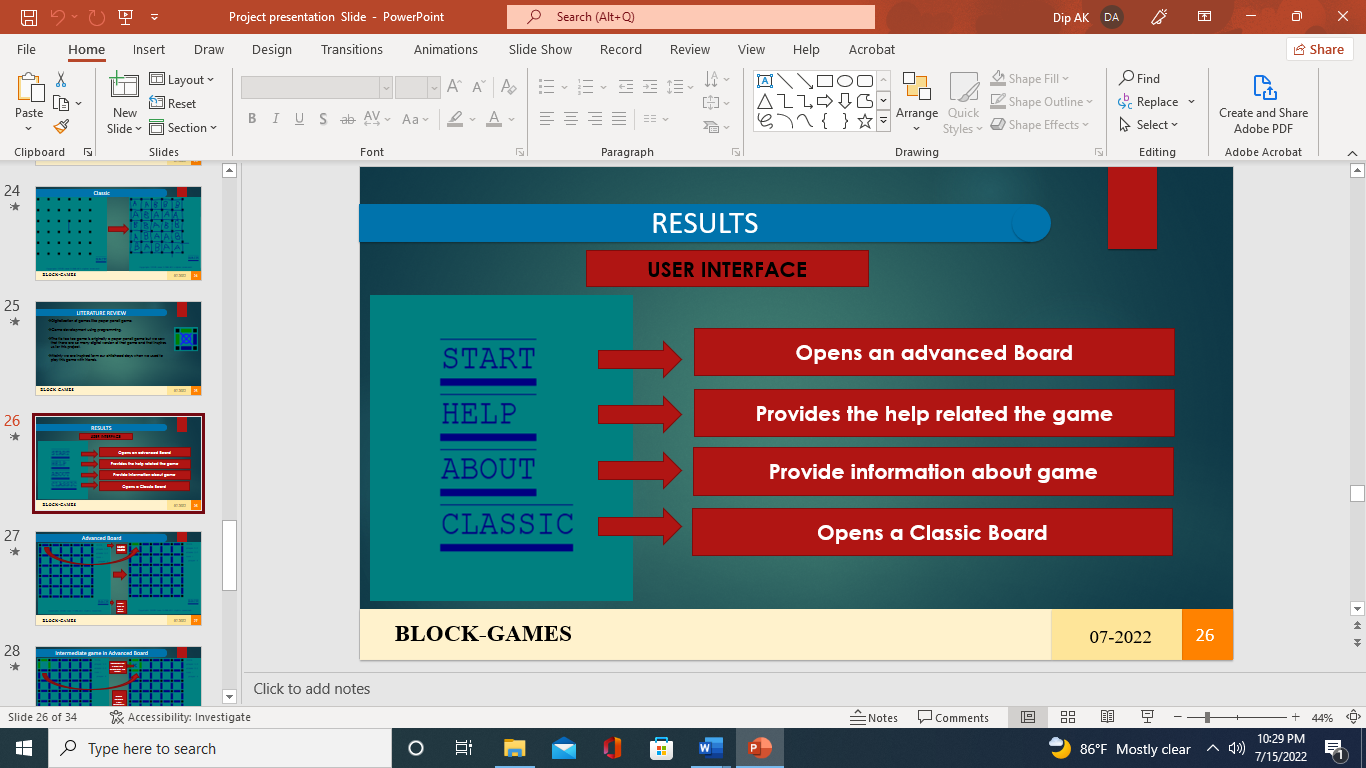


**Figure 6 Classic Board**

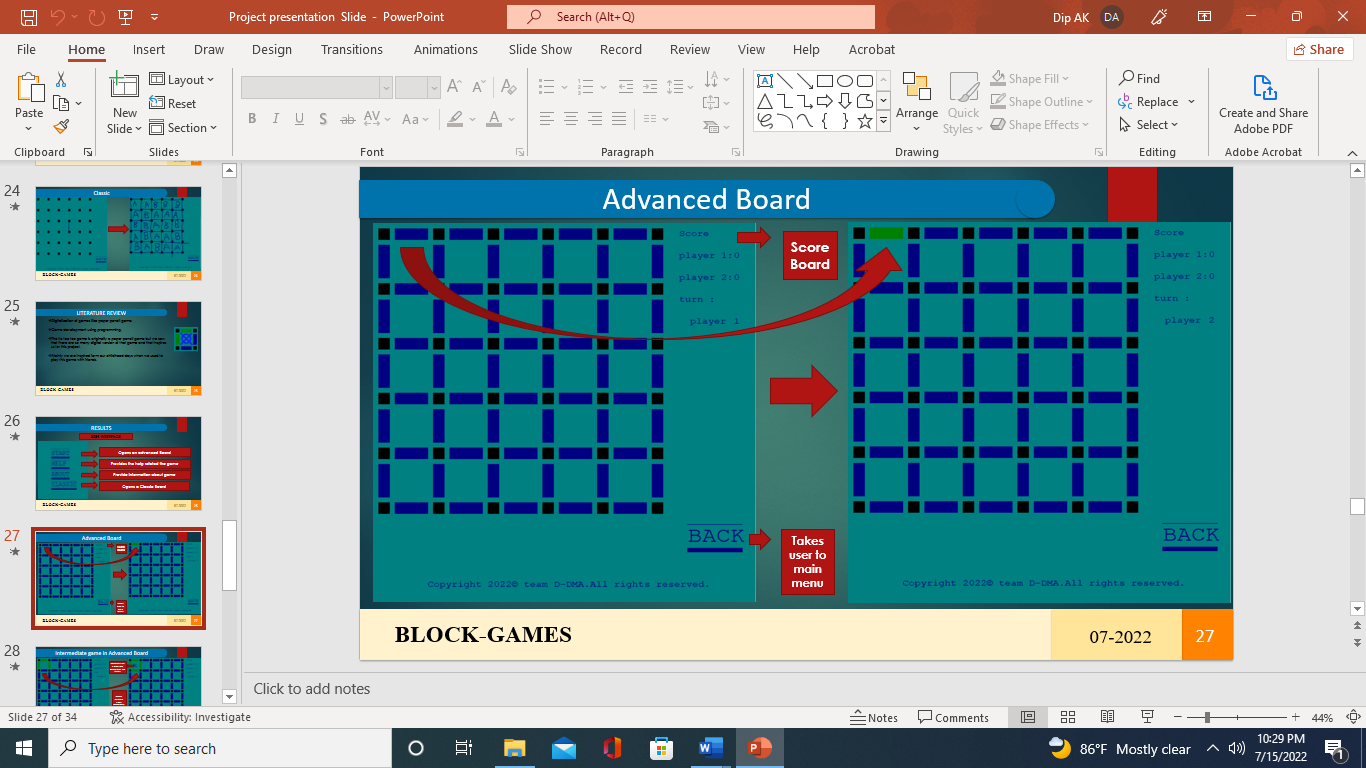
**CHAPTER 4**

# RESULT CONCLUSION AND RECOMMENDATION

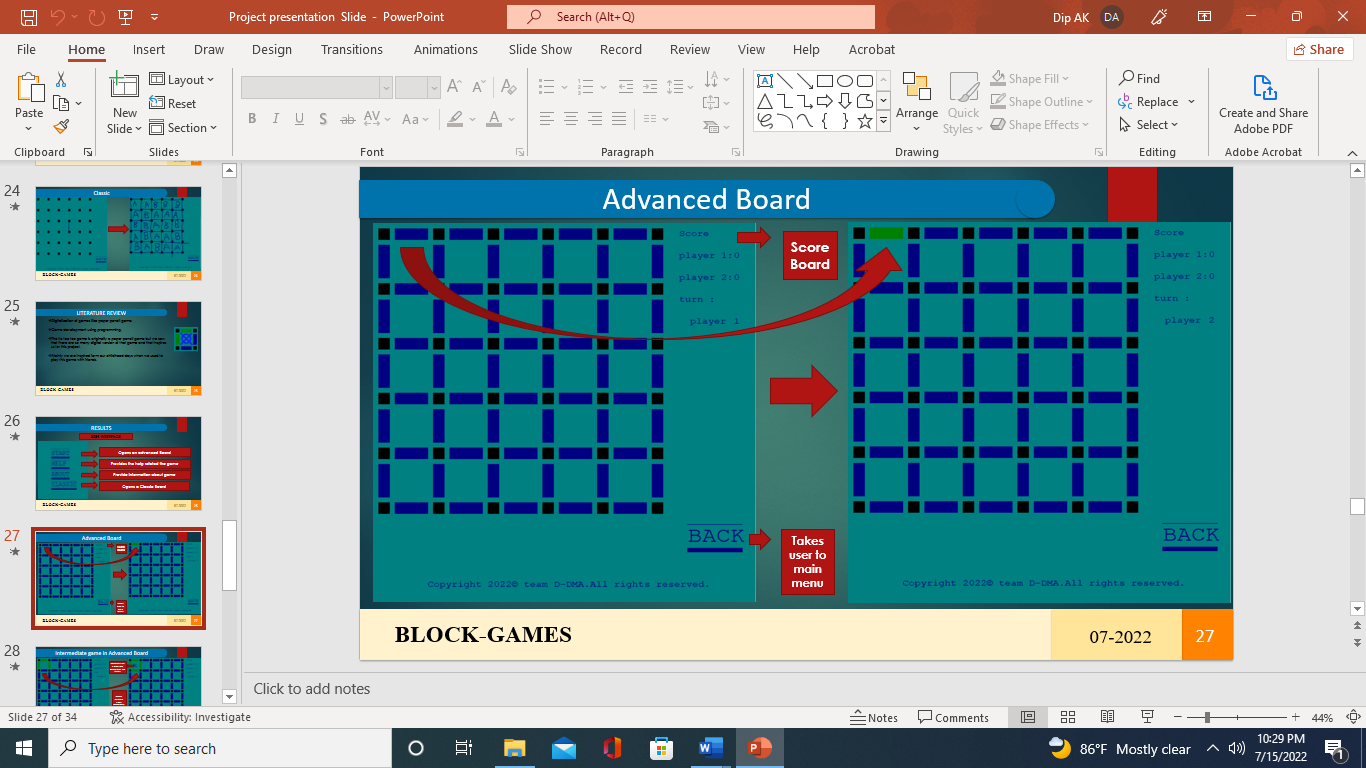
## 4.1 Results

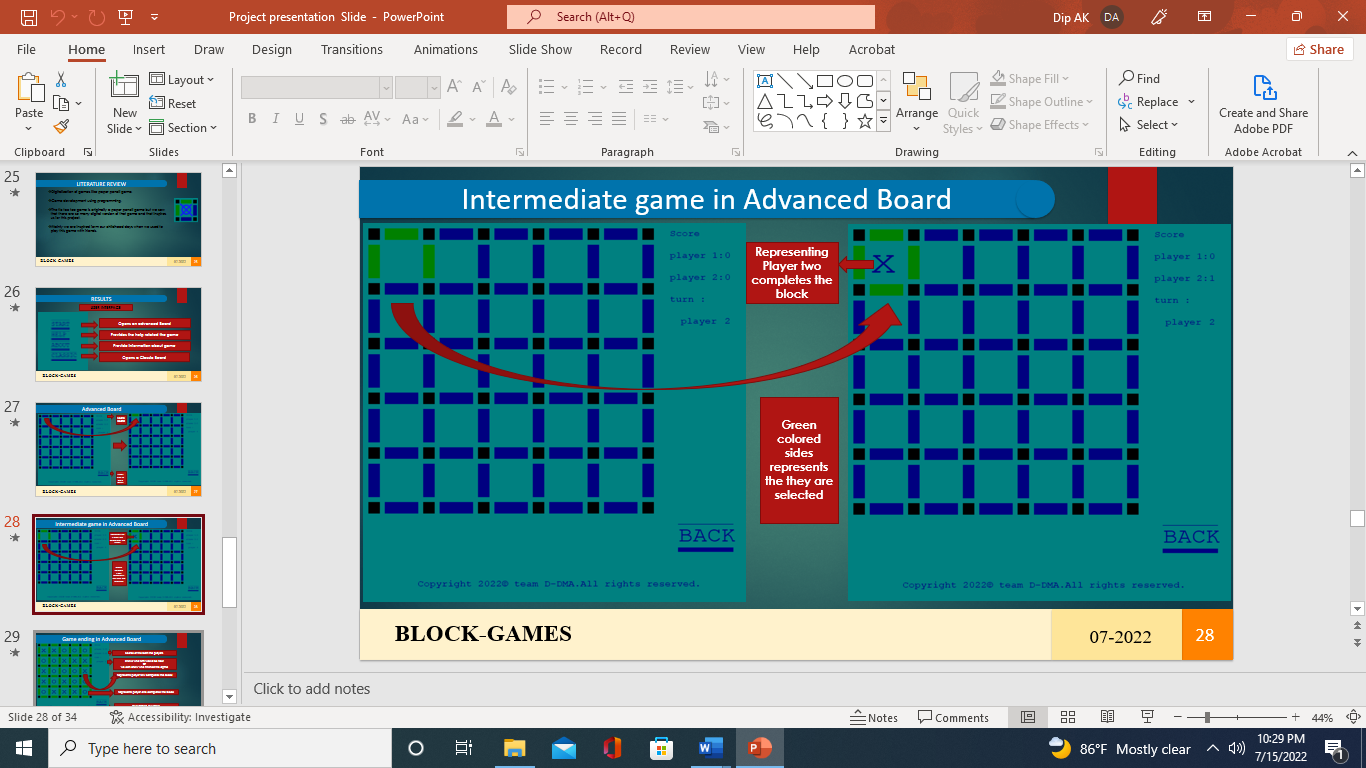
****

**Figure 7 User Interface**

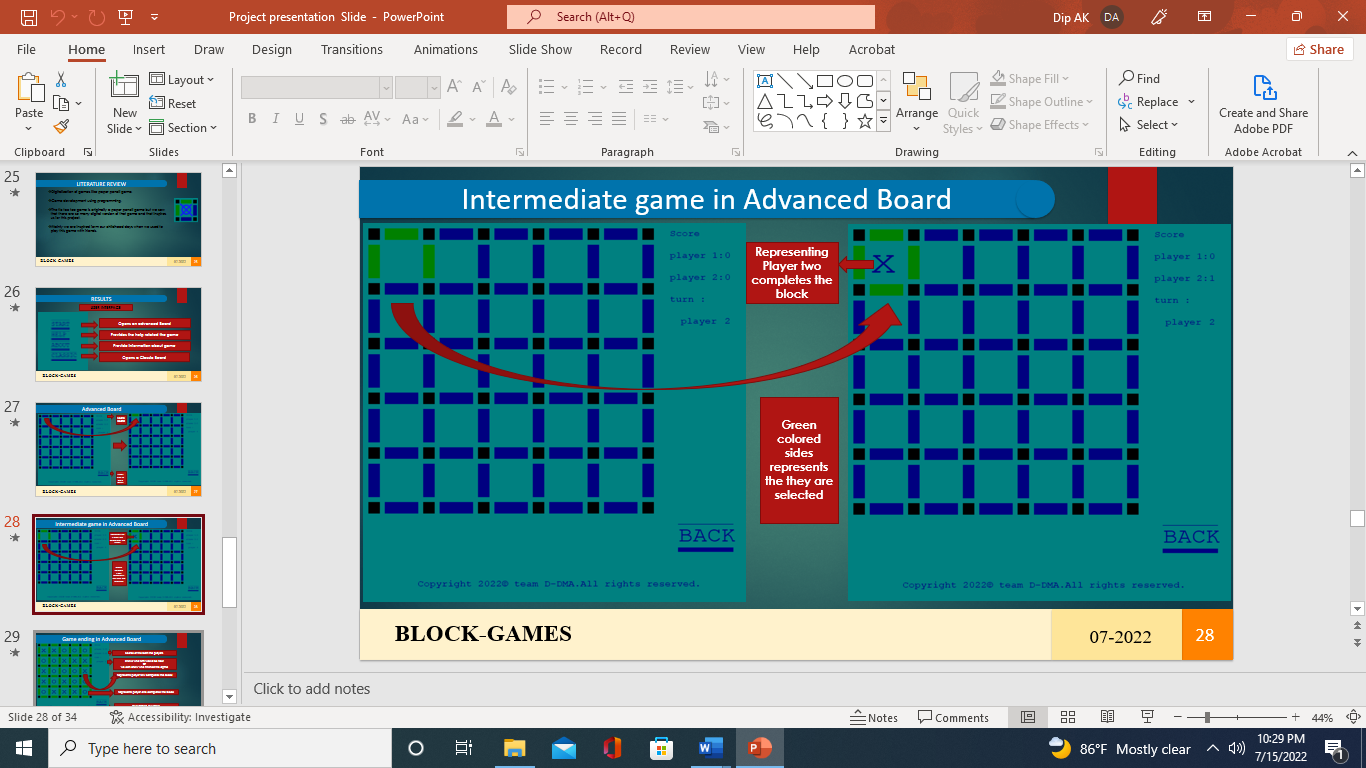
****

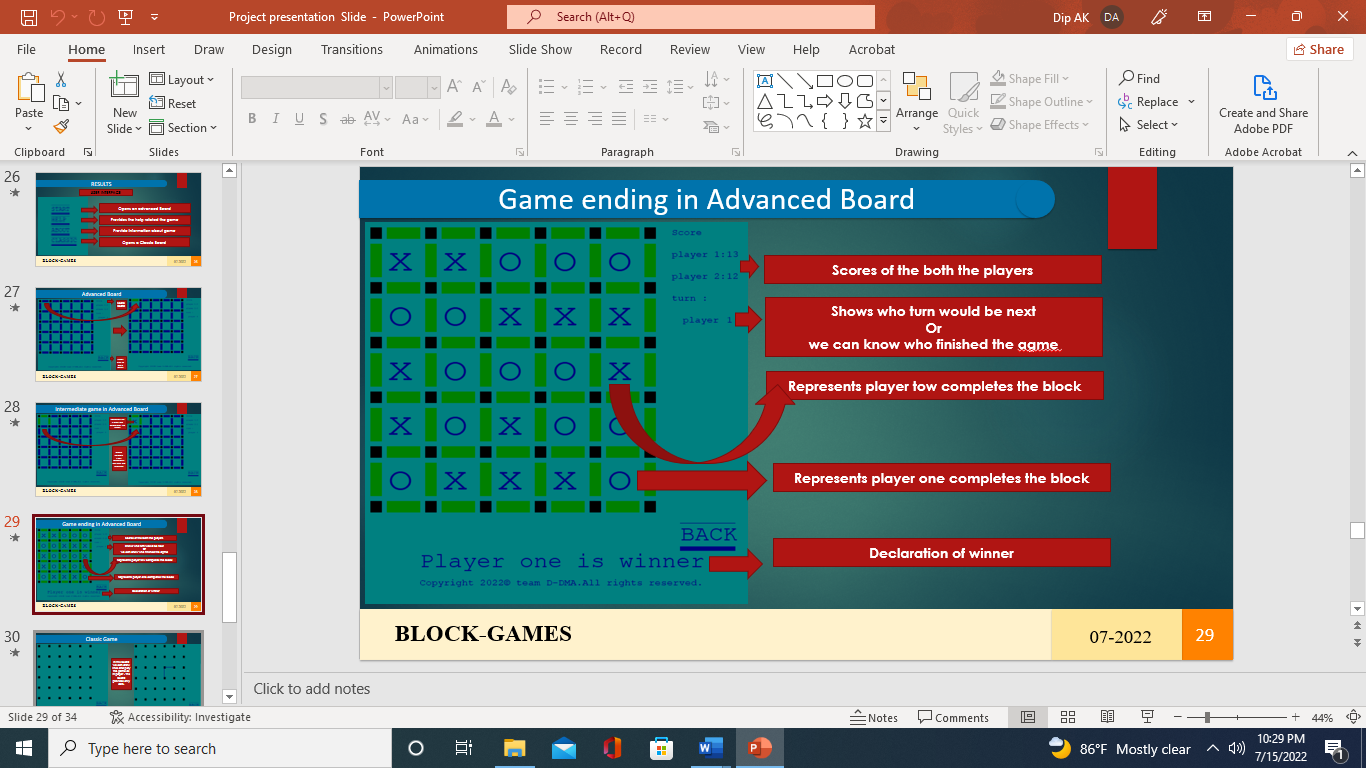
**Figure 8 Advance Game**

****

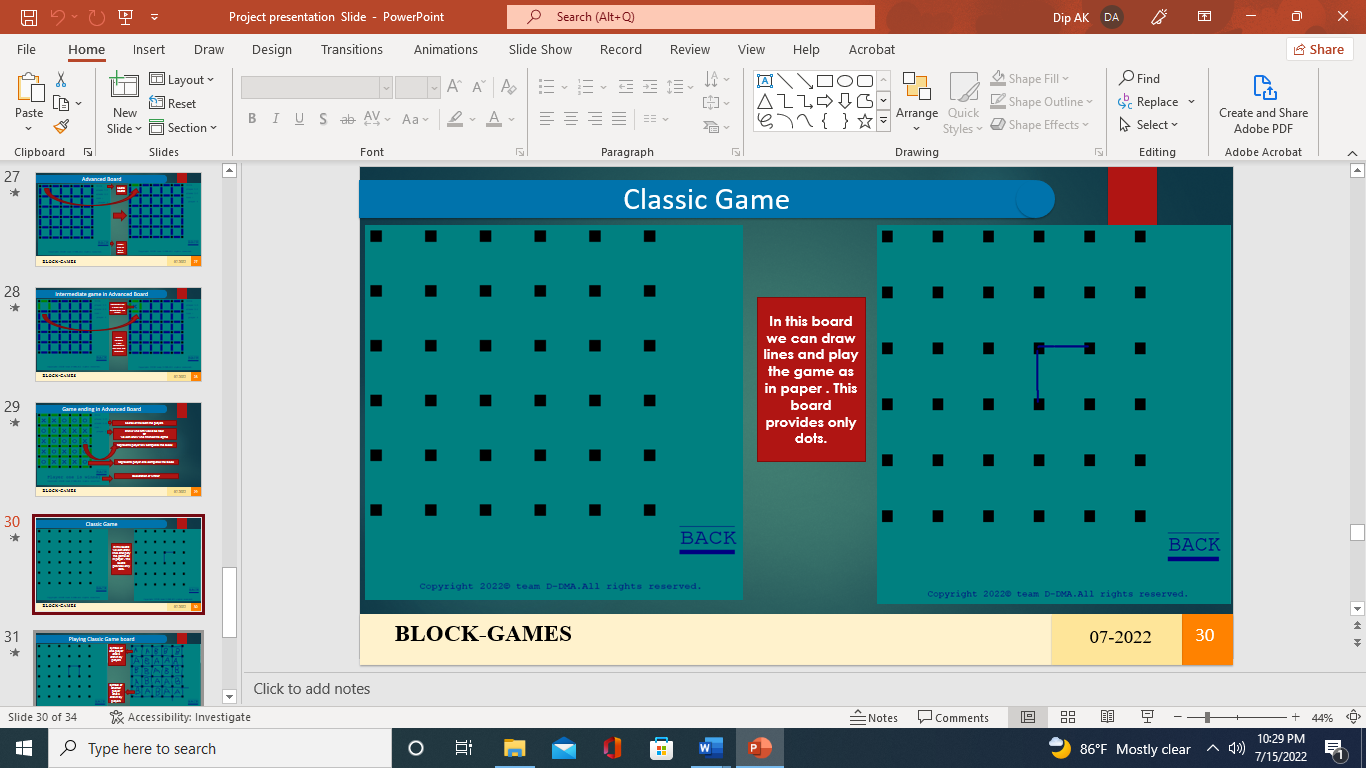
****

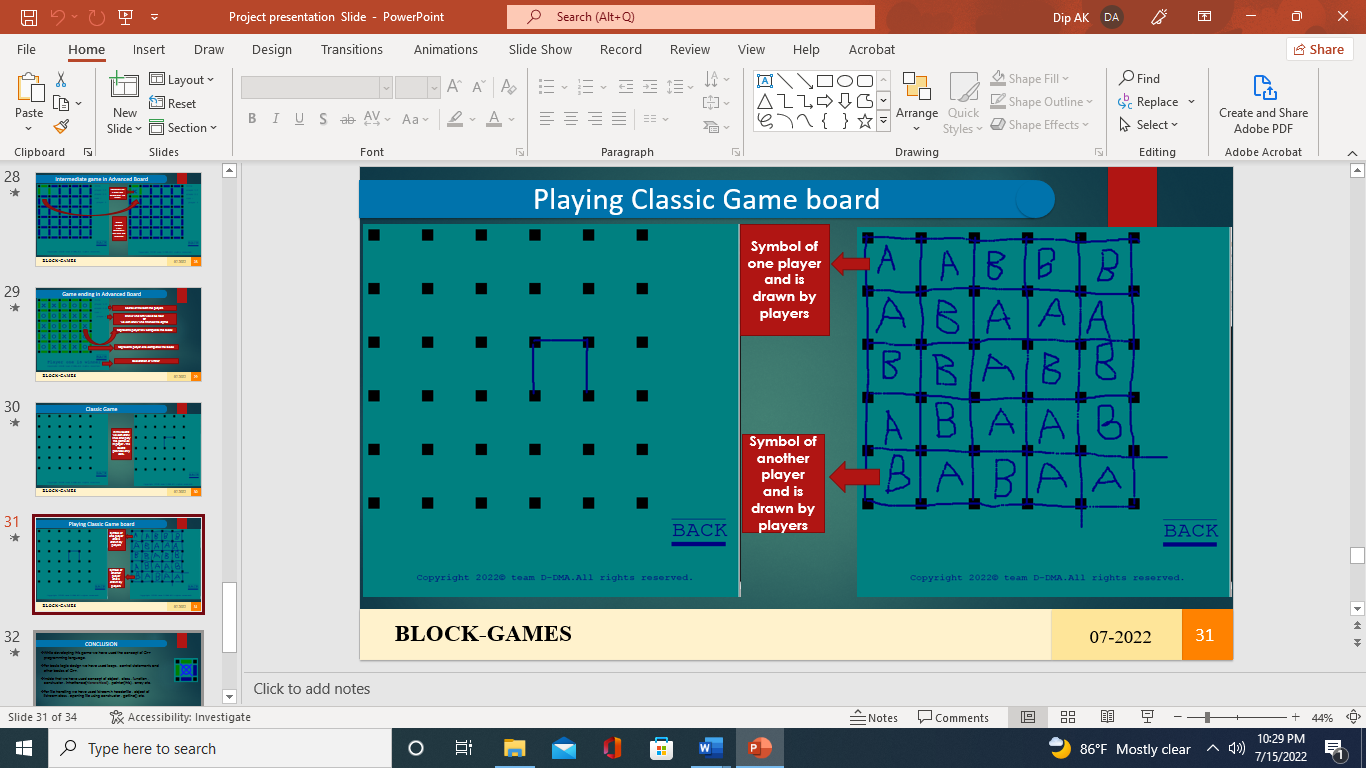
**Figure 9 Intermediate Game**

****

****

**Figure 10 Game Ending**

****

****

**Figure 11 Classic Game**

## 4.2 Limitations

* There is the board size fixed.
* The game doesn’t have sound effects.

## 4.3 Problem Faced

* Due to less knowledge of graphics, we have got some errors while using graphics functions.
* While implementing game logic some logical bugs made difficulties.

## 4.4 Conclusion

* While developing this game we have used the concept of C++ programming language.
* For basic logic design we have used loops, control statements and other basics of C++.
* Inside that we have used concept of object, class, function, constructor, inheritance(hierarchical), pointer(this), array etc.
* For file handling we have used fstream.h headerfile, object of ifstream class, opening file using constructor, getline () etc.

## 4.5 Future Scope and Recommendation

* This game can be extended with new interesting features.
* This game can made playable with more than 2 players too.

# References

Agrawal, D. (n.d.). *setfillstyle() and floodfill() in C*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/setfillstyle-floodfill-c/

*Dots and Boxes*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Dots\_and\_Boxes

Li, S., & Yuan, X. (n.d.). *Technique Analysis of Dots-and-Boxes*. Retrieved from ResearchGate: https://www.researchgate.net/publication/301454149\_Technique\_Analysis\_of\_Dots-and-Boxes

Main, M. (2008, september). *Using the WinBGIm Graphics Library with Dev-C++*. Retrieved from Colorado: https://home.cs.colorado.edu/~main/bgi/dev-c++/

Prasad, R. (2020, May 1). *01. Introduction to C/C++ graphics.h tutorial series | CodeWar*. Retrieved from CodeWar: https://www.youtube.com/watch?v=r7yOMz8HrHk

spielen, K., & puntos, j. c. (2022, February 16). *How to Play Dots and Boxes*. Retrieved from WikiHow: https://www.wikihow.com/Play-Dots-and-Boxes

Computer game is an important research subject of artificial intelligence, and the development of

artificial intelligence itself mainly profit from the development of computer game researches, in

which, the notable events in 1997 of Deep Blue beat Garry Kasparov, the world chess champion,

became one important milestone [1]. Invented in 1891 by Edward Lucas, one French mathematician,

Dots-and-Boxes [2] is a two-player game that using paper and pen. At earlier time, this game is very

popular in European, being taken as the project of international computer Olympic tournament game.

By now, many well-known international institutions have taken part in the development of related

software for Dots-and-Boxes, for example, Elmo Timoteus with the Mathematics Department of

UCLA (University of California at Los Angeles) has developed one software of Dots-and-Boxes,

which not only of high efficiency, but also can adjust the board size at any time during the game [2].

In recent years, with the development of Chinese computer game championship, Dots-and-Boxes has

gradually been known by the computer game enthusiasts. By now, Dots-and-Boxes has been listed as

one of the major computer competition games, and it also is the game has the most participator

number.

At present, only few articles have comprehensive introduce Dots-and-Boxes (fro details please

refer to [3-6]). This paper will mainly introduce the basic concept of Dots-and-Boxes, study the board

representation and winning factor and etc.

Computer game is an important research subject of artificial intelligence, and the development of

artificial intelligence itself mainly profit from the development of computer game researches, in

which, the notable events in 1997 of Deep Blue beat Garry Kasparov, the world chess champion,

became one important milestone [1]. Invented in 1891 by Edward Lucas, one French mathematician,

Dots-and-Boxes [2] is a two-player game that using paper and pen. At earlier time, this game is very

popular in European, being taken as the project of international computer Olympic tournament game.

By now, many well-known international institutions have taken part in the development of related

software for Dots-and-Boxes, for example, Elmo Timoteus with the Mathematics Department of

UCLA (University of California at Los Angeles) has developed one software of Dots-and-Boxes,

which not only of high efficiency, but also can adjust the board size at any time during the game [2].

In recent years, with the development of Chinese computer game championship, Dots-and-Boxes has

gradually been known by the computer game enthusiasts. By now, Dots-and-Boxes has been listed as

one of the major computer competition games, and it also is the game has the most participator

number.

At present, only few articles have comprehensive introduce Dots-and-Boxes (fro details please

refer to [3-6]). This paper will mainly introduce the basic concept of Dots-and-Boxes, study the board

representation and winning factor and etc.

Computer game is an important research subject of artificial intelligence, and the development of

artificial intelligence itself mainly profit from the development of computer game researches, in

which, the notable events in 1997 of Deep Blue beat Garry Kasparov, the world chess champion,

became one important milestone [1]. Invented in 1891 by Edward Lucas, one French mathematician,

Dots-and-Boxes [2] is a two-player game that using paper and pen. At earlier time, this game is very

popular in European, being taken as the project of international computer Olympic tournament game.

By now, many well-known international institutions have taken part in the development of related

software for Dots-and-Boxes, for example, Elmo Timoteus with the Mathematics Department of

UCLA (University of California at Los Angeles) has developed one software of Dots-and-Boxes,

which not only of high efficiency, but also can adjust the board size at any time during the game [2].

In recent years, with the development of Chinese computer game championship, Dots-and-Boxes has

gradually been known by the computer game enthusiasts. By now, Dots-and-Boxes has been listed as

one of the major computer competition games, and it also is the game has the most participator

number.

At present, only few articles have comprehensive introduce Dots-and-Boxes (fro details please

refer to [3-6]). This paper will mainly introduce the basic concept of Dots-and-Boxes, study the board

representation and winning factor and etc.