NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited bv NAAC with 'A' Grade. Accredited bv NBA

nirf TOP ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

A

MINI PROJECT REPORT

ON

"TIC TAC TOE GAME"

Submitted in the partial fulfillment of the requirements in the 3rd semester of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

BY

CHRISEL FERNANDES - [1NH18IS026]

FOR

COURSE NAME: MINI PROJECT

COURSE CODE: 19ISE391

Under the guidance of

Vandana C P
Sr. Professor
Dept. of ISE, NHCE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING
(Autonomous College Permanently Affiliated to VTU, Approved by AICTE, Accredited by NAAC with 'A' Grade & NBA)
Ring Road, Bellandur Post, Near Marathahalli,
Bengaluru-560103, INDIA

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade. Accredited by NBA

nirf TOP ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

CERTIFICATE

Certified that the project work entitled "**Tic Tac Toe game**" carried out by Mr. Chrisel Fernandes, USN: 1NH18IS026, a bonafide student of III sem in partial fulfillment for the award of Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2019-20. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Mrs. Vandana C P	Dr. R.J Anandhii	Dr. Manjunatha

ACKNOWLEDGMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, who’s constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha**, Principal NHCE, for his constant support and encouragement.

I am grateful to **Dr. Prashanth C.S.R**, Dean Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. R.J. Anandhi,** Professor and Head, Department of Information Science and Engineering, for her constant support.

I express my gratitude to **Mrs. Vandana C P,** Senior Professor, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

CHRISEL FERNANDES

ABSTRACT

The game is to be played between two people. One of the player chooses ‘x’ and the other ‘o’ to mark their respective cells. The game starts with one of the player choosing their respective position. Later, the other player chooses their position. Then the game goes as the two players make their moves. The game ends when one of the players has one whole row/column/diagonal/L-shape with his/her respective character(x or o). If no one wins then the game is said to be draw. If both the players play optimally then it is destined that you will never loose. It doesn’t matter whether you play first or second. In other ways, two experts will always draw. There is a strategy to this game , that is, each time the player makes their first available move he/she should choose the following position: Center: A player marks the center, Opposite corner: If the opponent is in the corner, the player plays the opposite corner .

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: - INTRODUCTION

- Tic-tac-toe , noughts and crosses , or X’s and O’s is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid.
- The player who place their own marks three in a row wins the game.
- To judge the winner of the game, it requires marks on the grid and follows the game rule to find out the winner.
- Games are generally used as a source of entertainment, but can additionally serve an educational purpose. Young children can enjoy playing games such as Tic Tac Toe, while naturally developing skills.
- Students can enjoy playing Tic Tac Toe game in the leisure time and can relax and enjoy the game while at the same time increasing their logic and keep their minds active
- Tic tac toe is a paper and pencil game for two players, X and O who marks take turns marking the spaces in 3 X 3 grids. The player who succeeds in placing three respective marks in a horizontal, vertical, diagonal, L-shaped row wins the game.
- The Tic Tac Toe is a great way to pass your free time whether you're standing in a line or spending time with your kids. Stop wasting paper and save trees.
- Because of the simplicity of Tic Tac Toe, it is often used as a pedagogical tool for teaching the concepts of good sportsmanship and the branch of artificial intelligence.

- Here’s how it works:
  1. The input will be an 3x3 matrix consisting only of 0 to 9 . Player 1 should choose numbers between 0 to 9 in order to make his move.
  2. Then, player 2 should follow the similar procedure.
  3. This should be done until a result has been established.

4. If the user has entered the wrong value, then error message should be displayed and the user should be allowed to enter the correct choice again.
5. If either horizontal, vertical, diagonal, L-shaped row has been achieved by one of the players then the respective player has won should be executed. If neither of the above conditions have been established, “The game has been drawn” should be displayed.

1.1 Objective

- To develop a computer-based version of the classic 2-player Tic Tac Toe Game using C.
- The game is developed for full-time entertainment and enthusiasms. It teaches the Gamer to be alert at every situation he/she faces, because if the Gamer is not fully alert and notice the saucer fire he/she must be hit by the saucer-bombs.
- Though the proposed game is an action game, it doesn’t involve direct violence. No zombie killing, animal killing, or human killing is performed in the game. So it can also be viewed as a non-violence game.
- Kids can also play this game, because the design of the game is very simple, controlling the game is very easy – pressing some neighbouring keys of the keyboard.

1.2 Brief Description

- Tic Tac Toe is a quick and easy game for at least two people that requires nothing more than paper, a pencil, and the skill to play. One player chooses X while the other player chooses O . However, every wrong move brings them one step closer to losing.

- The game of tic-tac-toe is played on a 3 by grid, onto which players take turns placing their marks (either X or an O), with the goal of being the first player to get three of their marks in a straight line (horizontally, vertically or diagonally).

- This project implements a human-based version of the 2-player Tic Tac Toe Game. The human player 1 competes against the human player 2 with the goal of being the first player to get three of their marks in a straight line (horizontally, vertically or diagonally).

- In the fast growing field of software engineering and development and even more rapidly growing sector of game development the future is hard to predict. In general software project is a project focusing on the creation of software. Consequently, Success can be measured by taking a look at the resulting software. In a game project, the product is a game. But and here comes the point: A game is much more than just its software. It has to provide content to become enjoyable. Just like a web server: without content the server is useless, and the quality cannot be measured. This has an important effect on the game project as a whole. The software part of the project is not the only one, and it must be considered in connection to all other parts: The environment of the game, the story, characters, game plays, the artwork, and so on.

1.3 Project Features

The application's purpose is to simulate the game "Tic Tac Toe." The major features are:

- Allow the players to select X or O.
- Manage the game play: getting the player think, make the move using logic, and getting three of their marks in a straight line.
- Determining if the game is won or lost and display the necessary output.

CHAPTER 2: - PROJECT PLAN AND DESIGN METHOD

2.1 Resources Planned

- Hardware Requirements: -
    - A system based on 64-bit architecture that supports Intel® Streaming SIMD Extensions 4.2 (Intel® SSE 4.2) instructions or compatible non-Intel® processors
    - 2 GB of free disk space for all product features and architectures

- Software Requirements: -
    - Windows* 7, 8.x, Windows® 10 (SP1 for Windows 7 is required for Intel® Advanced Vector Extensions use)
    - Windows Server* 2008 R2 SP1 and SP2, 2012, 2016
    - C Language

2.2 Software Used

C Language:

- C is a general-purpose programming language that is extremely popular, simple and flexible. It is machine-independent, structured programming language which is used extensively in various applications.

- C is a procedural programming language.

- It was initially developed by Dennis Ritchie in the year 1972.

- It was mainly developed as a system programming language to write an operating system. The main features of C language include low-level access to memory, a simple set of keywords, and clean style, these features make C language suitable for system programmings like an operating system or compiler development.

- Many later languages have borrowed syntax/features directly or indirectly from C language. Like syntax of Java, PHP, JavaScript, and many other languages are mainly based on C language. C++ is nearly a superset of C language (There are few programs that may compile in C, but not in C++).

- C was the basics language to write everything from operating systems (Windows and many others) to complex programs like the Oracle database, Git, Python interpreter and more.

- It is said that ‘C’ is a god’s programming language. One can say, C is a base for the programming. If you know ‘C,’ you can easily grasp the knowledge of the other programming languages that uses the concept of ‘C’.

- It is essential to have a background in computer memory mechanisms because it is an important aspect when dealing with the C programming language.

History of C language

The base or father of programming languages is ‘ALGOL.’ It was first introduced in 1960. ‘ALGOL’ was used on a large basis in European countries. ‘ALGOL’ introduced the concept of structured programming to the developer community. In 1967, a new computer programming language was announced called as ‘BCPL’ which stands for Basic Combined Programming Language. BCPL was designed and developed by Martin Richards, especially for writing system software. This was the era of programming languages. Just after three years, in 1970 a new programming language called ‘B’ was introduced by Ken Thompson that contained multiple features of ‘BCPL.’ This programming language was created using UNIX operating system at AT&T and Bell Laboratories. Both the ‘BCPL’ and ‘B’ were system programming languages.

CHAPTER 3: -CONCEPT USED

This project uses :

3.1 Header Files Inclusion

The first and foremost component is the inclusion of the Header files in a C program. A header file is a file with extension .h which contains C function declarations and macro definitions to be shared between several source files.

Some of C Header files:

- stddef.h – Defines several useful types and macros.
- stdint.h – Defines exact width integer types.
- stdio.h – Defines core input and output functions
- stdlib.h – Defines numeric conversion functions, pseudo-random network generator, memory allocation
- string.h – Defines string handling functions
- math.h – Defines common mathematical functions

Syntax to include a header file in C:

#include

3.2 One Dimensional Array

- Scalar variable is one that can hold only one value at a time. For many problems you need to group data items together. A program that processes exam scores for a class, for example, would be easier to write if all the scores were stored in one area of memory and were able to be accessed as a group. C allows a programmer to group such related data items together into a single composite data structure. A data structure is a grouping of related data items in memory. We now take a look at one such data structure: the Array. An array is a collection of two or more adjacent memory cells that: 1) Store the same type of data values. 2) Are A referenced by the same name (i.e using one variable). These individual cells are called array elements.

- Declaring One-dimensional Arrays: To declare an array, we must declare its name, type of data values it will store and the number of cells associated with it. Example: double x[8]; This instructs C to associate eight memory cells with the

  name x; these memory cells will be adjacent to each other in memory. You can declare arrays along with scalar variables: double temperature[30], maxTemp, rainfall[30], max Rainfall; It is a good practice to define the array size as constant: #define ARRAY_SIZE 12 //... int myArray[ARRAY_SIZE];

- Each element of the array x may contain a single value of type double, so a total of eight such numbers may be stored and referenced using the array name x. The elements are numbered starting with 0 ♣ An array with 8 elements has elements at 0,1,2,3,4,5,6, and 7 The subscripted variable x[0] (read as x sub zero) refers to the initial or 0 th element of the array x, x[1] is the next element in the array, and so on. The integer enclosed in brackets is the array subscript or index and its value must be in the range from 0 to array Size - 1.

3.3 Functions used

1. **int checkwin():** To check the status of the player that is, if he has won or lost.

2. **void board():** To design the tic tac toe game board

3.4 Loops

In looping, a program executes the sequence of statements many times until the stated condition becomes false. A loop consists of two parts, a body of a loop and a control statement. The control statement is a combination of some conditions that direct the body of the loop to execute until the specified condition becomes false.

Types of Loops:

Depending upon the position of a control statement in a program, a loop is classified into two types:

1. Entry controlled loop

2. Exit controlled loop

In an **entry controlled loop,** a condition is checked before executing the body of a loop. It is also called as a pre-checking loop.

In an **exit controlled loop**, a condition is checked after executing the body of a loop. It is also called as a post-checking loop.

The control conditions must be well defined and specified otherwise the loop will execute an infinite number of times. The loop that does not stop executing and processes the statements number of times is called as an **infinite loop**. An infinite loop is also called as an "**Endless loop**." Following are some characteristics of an infinite loop:

1. No termination condition is specified.

2. The specified conditions never meet.

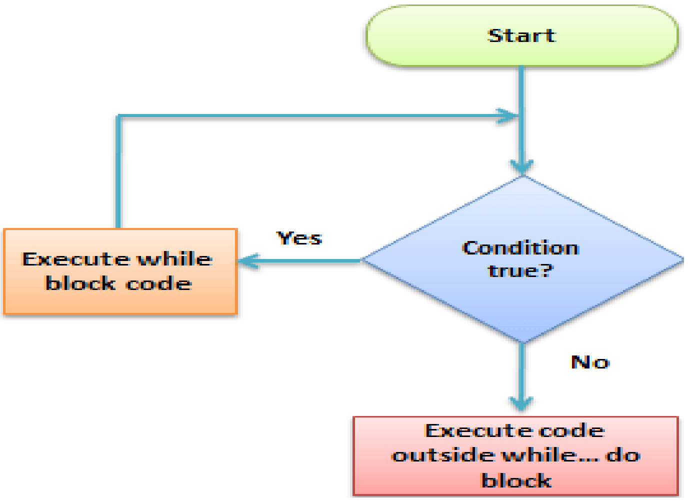
The specified condition determines whether to execute the loop body or not.

'C' programming language provides us with three types of loop constructs:

1. The while loop

2. The do-while loop

3. The for loop



In this project, do-while loop has been made use.
A do-while loop is similar to the while loop except that the condition is always executed after the body of a loop. It is also called an exit-controlled loop.

3.5 Conditional statements

- **Ternary operator** has been made use in this project
  If any operator is used on three operands
  Operator. It can be represented with ? : . It is also called as conditional operator
  The ternary operator is an operator that takes three arguments. The first

Fig 3.4. Loops

argument is a comparison argument, the second is the result upon a true comparison, and the third is the result upon a false comparison. Ternary operator is shortened way of writing an if-else statement.
Ternary operator is a?b:c it say that the condition a is true b will be executed else c will be executed.

- **If-else if** looping statement has been made use in this project.
  Else If statement in C handles multiple statements effectively by executing them

  sequentially. It will check for the first condition. If the condition is TRUE, then it will execute the statements present in that block. If the condition is FALSE, then check the Next one (Else If condition) and so on.

CHAPTER 4: -DESIGN METHOD

4.1 Algorithm

Step 1: Start.

Step 2: Select X or O

Step 3: After choosing X or O, start playing by inserting X or O in the appropriate position from 1 to 9

Step 4:If the position entered by the user is invalid, ask the user to enter the correct position.

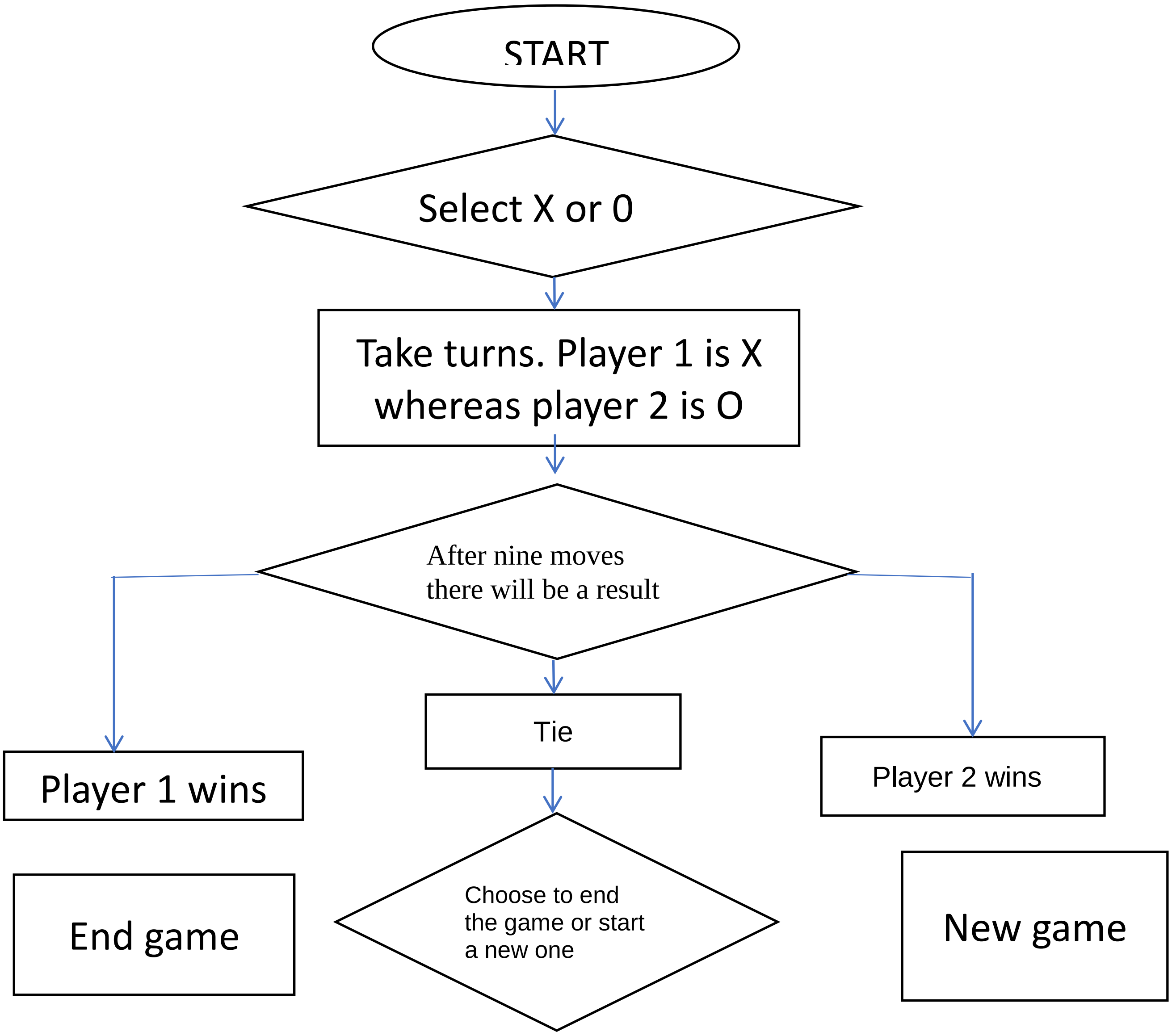
Step 5:Then the choice of position is continuously made until result has been displayed.

Step 6: Once the game has ended either game won, lost or game has been drawn is displayed.

Step 7: Ask user if he/she wants to play again or not.

Step 8: End

4.2 Flowchart


START
Select X or 0
Take turns. Player 1 is X whereas player 2 is O
After nine moves there will be a result
Player 1 wins
Tie
Player 2 wins
End game
Choose to end the game or start a new one
New game

4.3 Code

```
#include <stdio.h>
#include <conio.h>

char square[10] = { 'o', '1', '2', '3', '4', '5', '6', '7', '8', '9'
};

int checkwin();
void board();

int main()
{
    int player = 1, i, choice;

    char mark;
    do
    {
        board();
        player = (player % 2) ? 1 : 2;

        printf("Player %d, enter a number:  ", player);
        scanf("%d", &choice);

        mark = (player == 1) ? 'X' : 'O';

        if (choice == 1 && square[1] == '1')
            square[1] = mark;

        else if (choice == 2 && square[2] == '2')
            square[2] = mark;

        else if (choice == 3 && square[3] == '3')
            square[3] = mark;

        else if (choice == 4 && square[4] == '4')
            square[4] = mark;

        else if (choice == 5 && square[5] == '5')
            square[5] = mark;

        else if (choice == 6 && square[6] == '6')
            square[6] = mark;

        else if (choice == 7 && square[7] == '7')
            square[7] = mark;

        else if (choice == 8 && square[8] == '8')
            square[8] = mark;

        else if (choice == 9 && square[9] == '9')
            square[9] = mark;

        else
```

```
          {
            printf("Invalid move ");

            player--;
            getch();
        }
        i = checkwin();

        player++;
    }while (i ==  - 1);

    board();

    if (i == 1)
        printf("==>\aPlayer %d win ", --player);
    else
        printf("==>\aGame draw");

    getch();

    return 0;
}

/*********************************************

FUNCTION TO RETURN GAME STATUS
1 FOR GAME IS OVER WITH RESULT
-1 FOR GAME IS IN PROGRESS
O GAME IS OVER AND NO RESULT
 **********************************************/

int checkwin()
{
    if (square[1] == square[2] && square[2] == square[3])
        return 1;

    else if (square[4] == square[5] && square[5] == square[6])
        return 1;

    else if (square[7] == square[8] && square[8] == square[9])
        return 1;

    else if (square[1] == square[4] && square[4] == square[7])
        return 1;

    else if (square[2] == square[5] && square[5] == square[8])
        return 1;

    else if (square[3] == square[6] && square[6] == square[9])
        return 1;

    else if (square[1] == square[5] && square[5] == square[9])
        return 1;

    else if (square[3] == square[5] && square[5] == square[7])
```

```
       return 1;

   else if (square[1] == square[2] && square[2] == square[4])
       return 1;

   else if (square[2] == square[3] && square[3] == square[6])
       return 1;

   else if (square[4] == square[7] && square[7] == square[8])
       return 1;

   else if (square[8] == square[9] && square[9] == square[6])
       return 1;

    else if (square[1] != '1' && square[2] != '2' && square[3] !=
'3' &&
        square[4] != '4' && square[5] != '5' && square[6] != '6' &&
square[7]
        != '7' && square[8] != '8' && square[9] != '9')

        return 0;
    else
        return  - 1;
}


/*******************************************************************
FUNCTION TO DRAW BOARD OF TIC TAC TOE WITH PLAYERS MARK

********************************************************************
/


void board()
{
    system("cls");
    printf("\n\n\tTic Tac Toe\n\n");

    printf("Player 1 (X)  -  Player 2 (O)\n\n\n");


    printf("     |     |     \n");
    printf("  %c  |  %c  |  %c \n", square[1], square[2],
square[3]);

    printf("_____|_____|_____\n");
    printf("     |     |     \n");

    printf("  %c  |  %c  |  %c \n", square[4], square[5],
square[6]);

    printf("_____|_____|_____\n");
    printf("     |     |     \n");
printf("  %c  |  %c  |  %c \n", square[7], square[8], square[9]);

    printf("     |     |     \n\n");
}

/*******************************************************************
END OF PROJECT

********************************************************************
/
```

CHAPTER 5: - TEST SCENARIOS AND RESULT

5.1 Test Scenarios

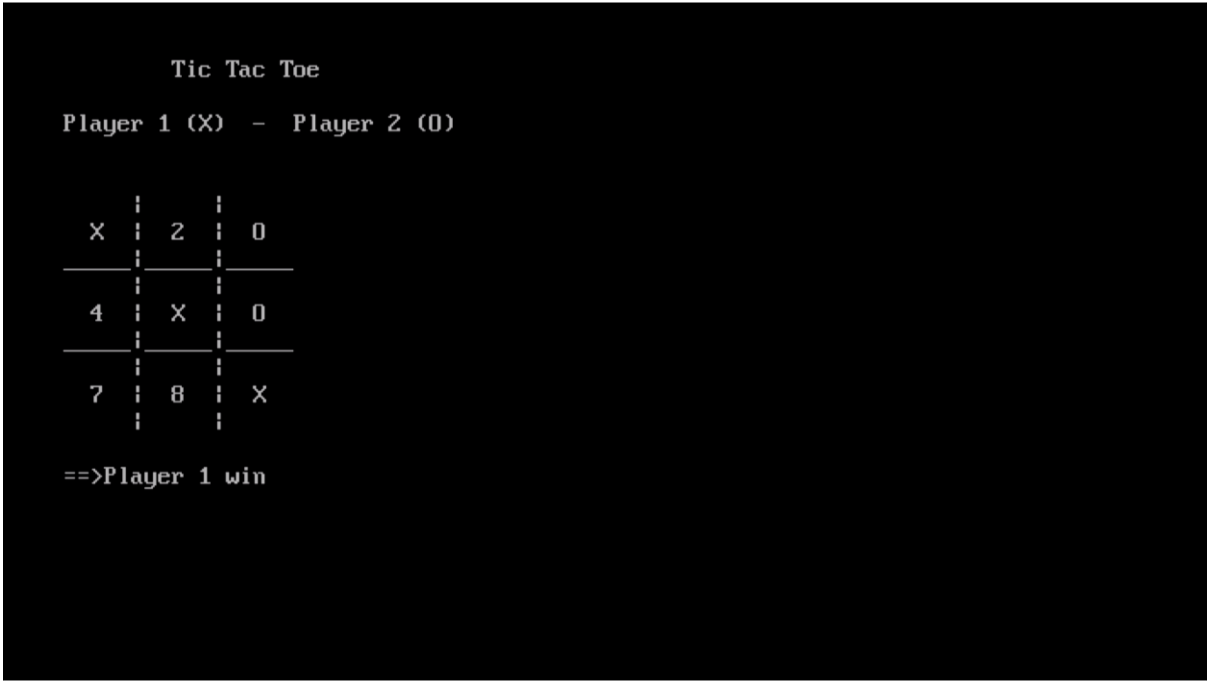
The program can have varying outputs depending on how the user plays the game.
There are primarily three main scenarios:

- The player 1 wins the game
- The player 2 wins the game
- The game ends in a draw

5.2 Result

The program was tested for the above test case scenarios and the result expected as per the test case scenarios was successfully achieved. The result for the various test case scenarios is given as follows:

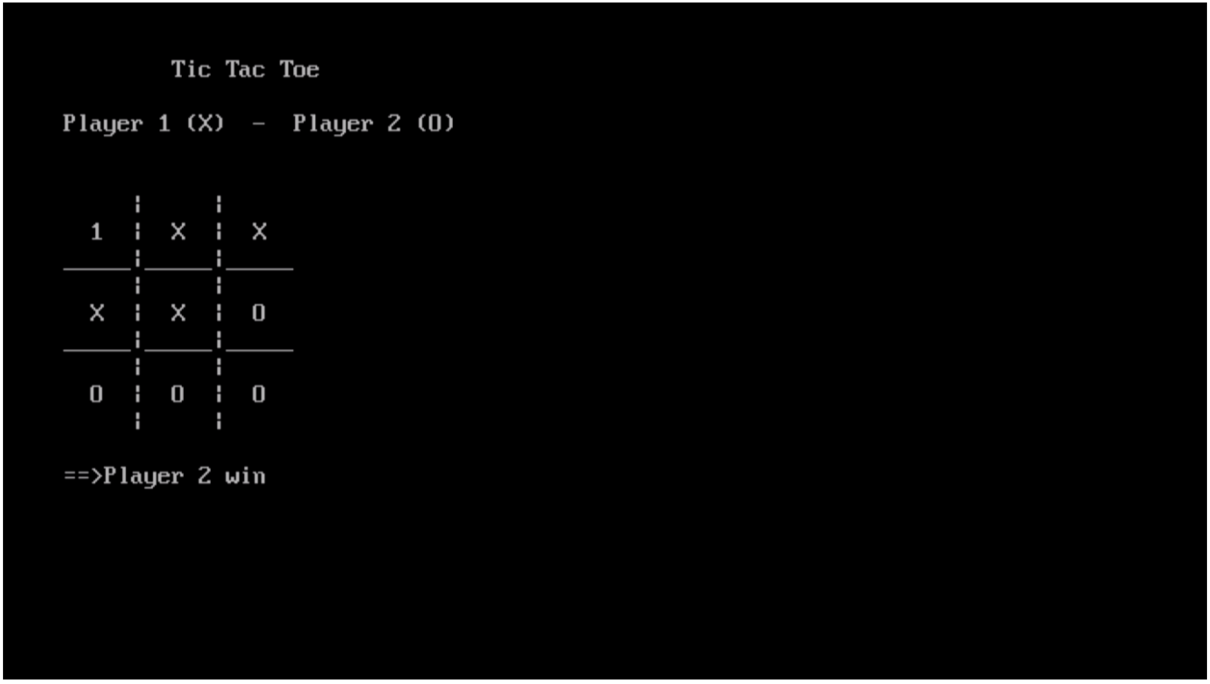
- The player 1 wins the game.



```
        Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  2  |  O
_____|_____|_____
     |     |
  4  |  X  |  O
_____|_____|_____
     |     |
  7  |  8  |  X
     |     |

==>Player 1 win
```

Fig 5.2.1. The player 1 wins the game by choosing X in a diagonal row

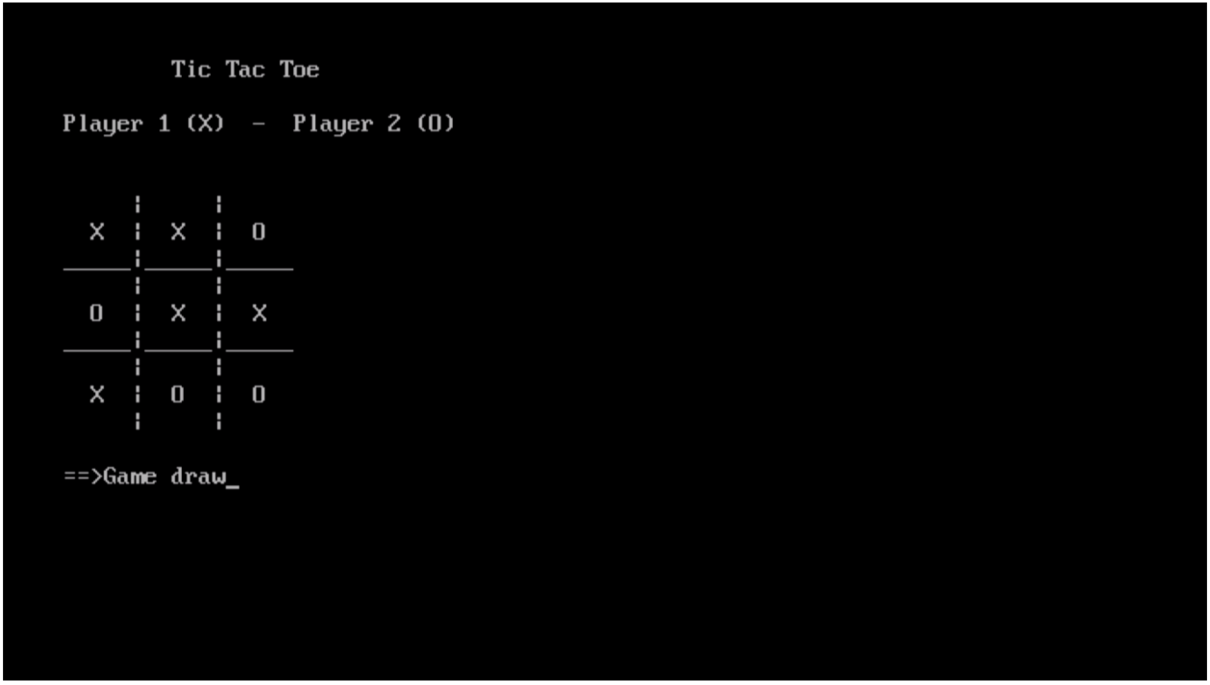
- The player 2 wins the game.



```
        Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  1  |  X  |  X
_____|_____|_____
     |     |
  X  |  X  |  O
_____|_____|_____
     |     |
  O  |  O  |  O
     |     |

==>Player 2 win
```

Fig 5.2.2. The player 2 wins the game by choosing O in a horizontal row

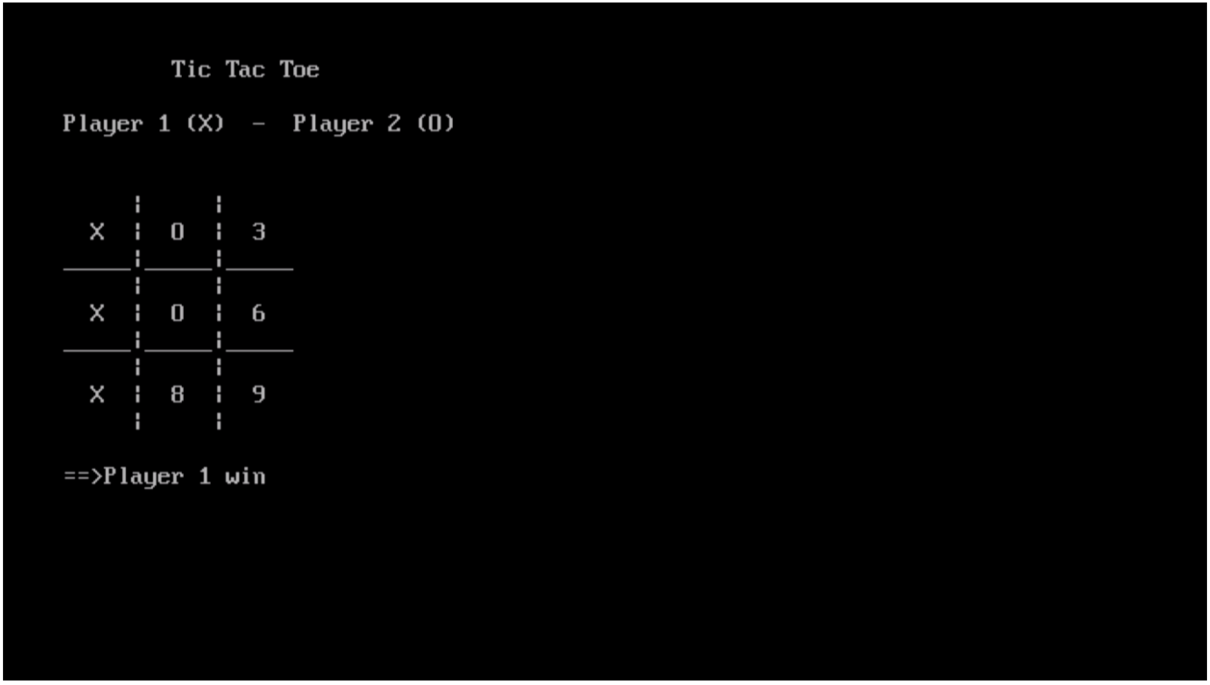
- The game ends in a draw



```
        Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  X  |  O
_____|_____|_____
     |     |
  O  |  X  |  X
_____|_____|_____
     |     |
  X  |  O  |  O
     |     |

==>Game draw_
```

Fig 5.2.3. The game ends in a draw

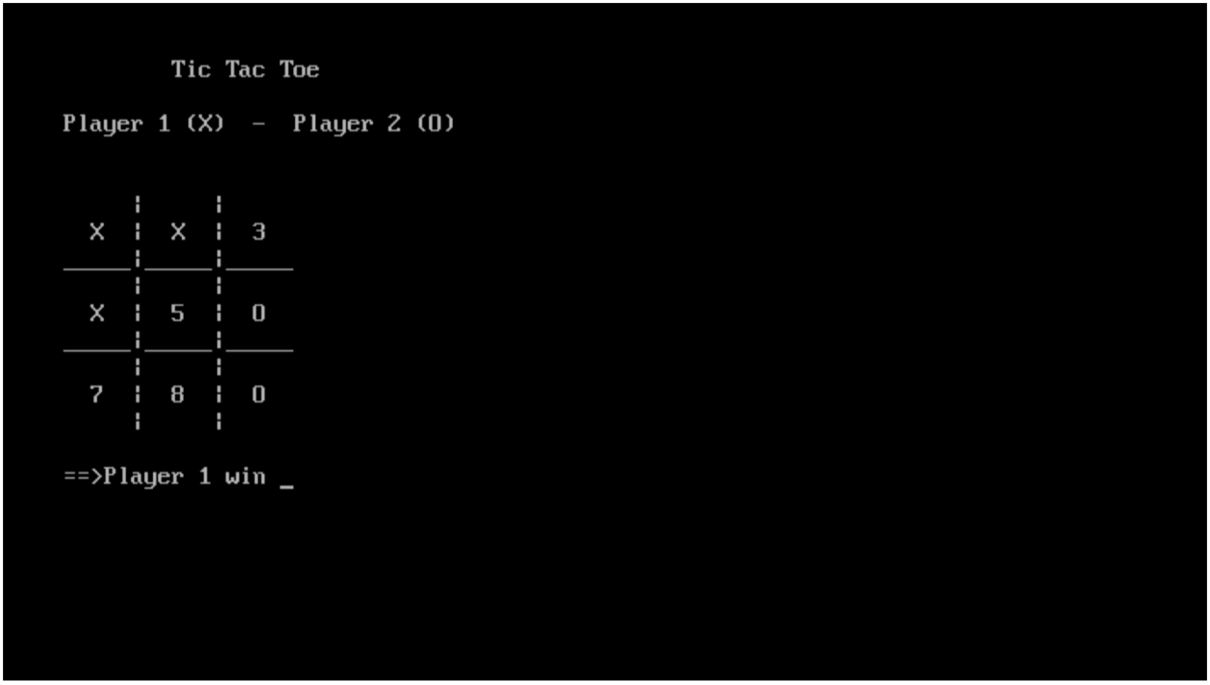
- The player 1 wins the game



```
        Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  O  |  3
_____|_____|_____
     |     |
  X  |  O  |  6
_____|_____|_____
     |     |
  X  |  8  |  9
     |     |

==>Player 1 win
```

Fig 5.2.4. The player 1 wins the game by choosing X in a vertical row

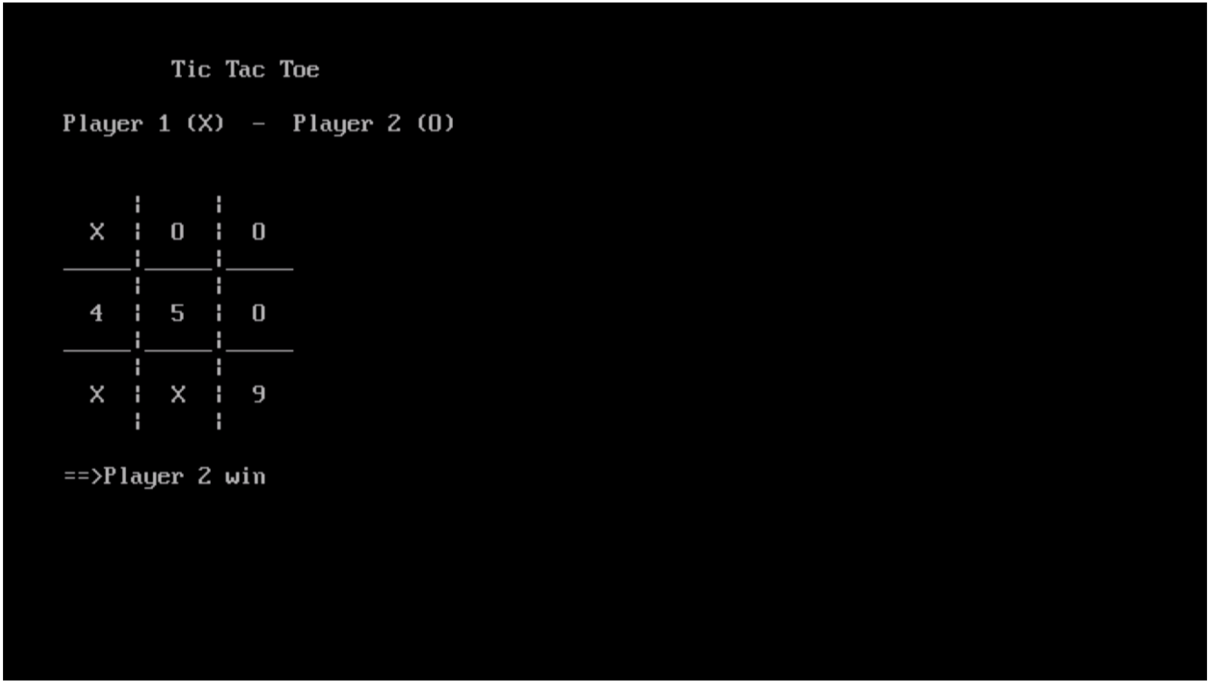
- The player 1 wins the game.



```
        Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  X  |  3
_____|_____|_____
     |     |
  X  |  5  |  O
_____|_____|_____
     |     |
  7  |  8  |  O
     |     |

==>Player 1 win _
```

Fig 5.2.5. The player 1 wins the game by choosing X in the form of L-shape In the top left corner.

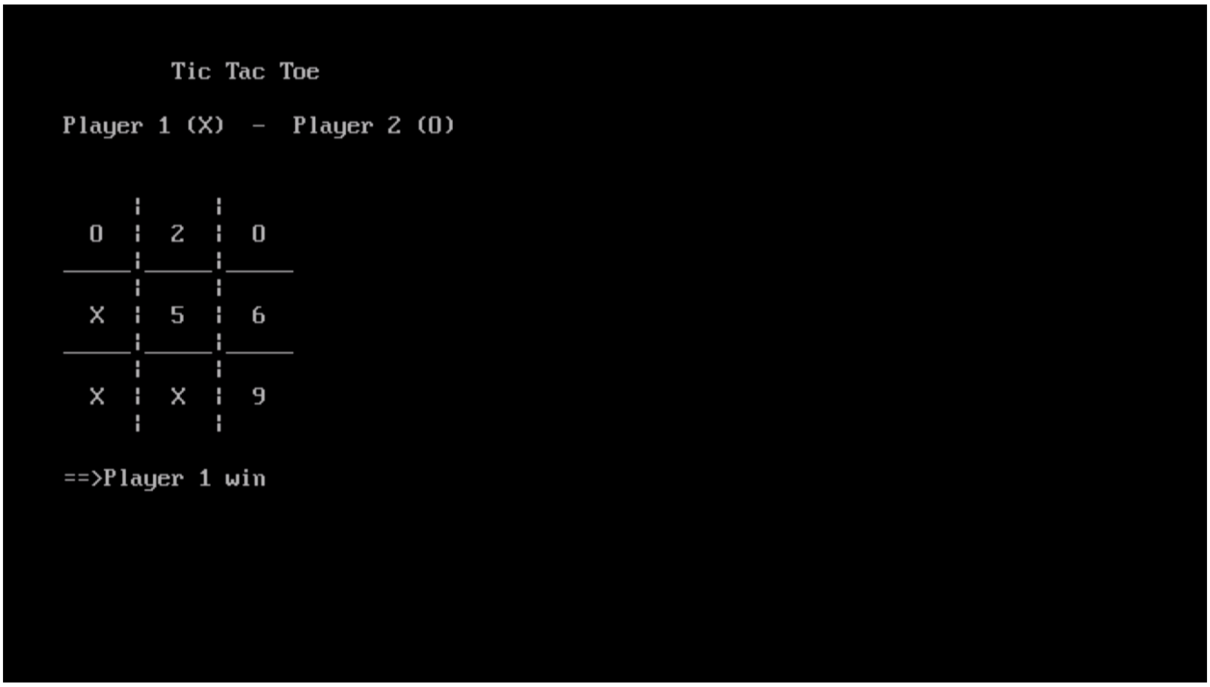
- The player 2 wins the game



```
        Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  O  |  O
_____|_____|_____
     |     |
  4  |  5  |  O
_____|_____|_____
     |     |
  X  |  X  |  9
     |     |

==>Player 2 win
```

Fig 5.2.6. The player 2 wins the game by choosing O in the form of L-shape in the top right corner

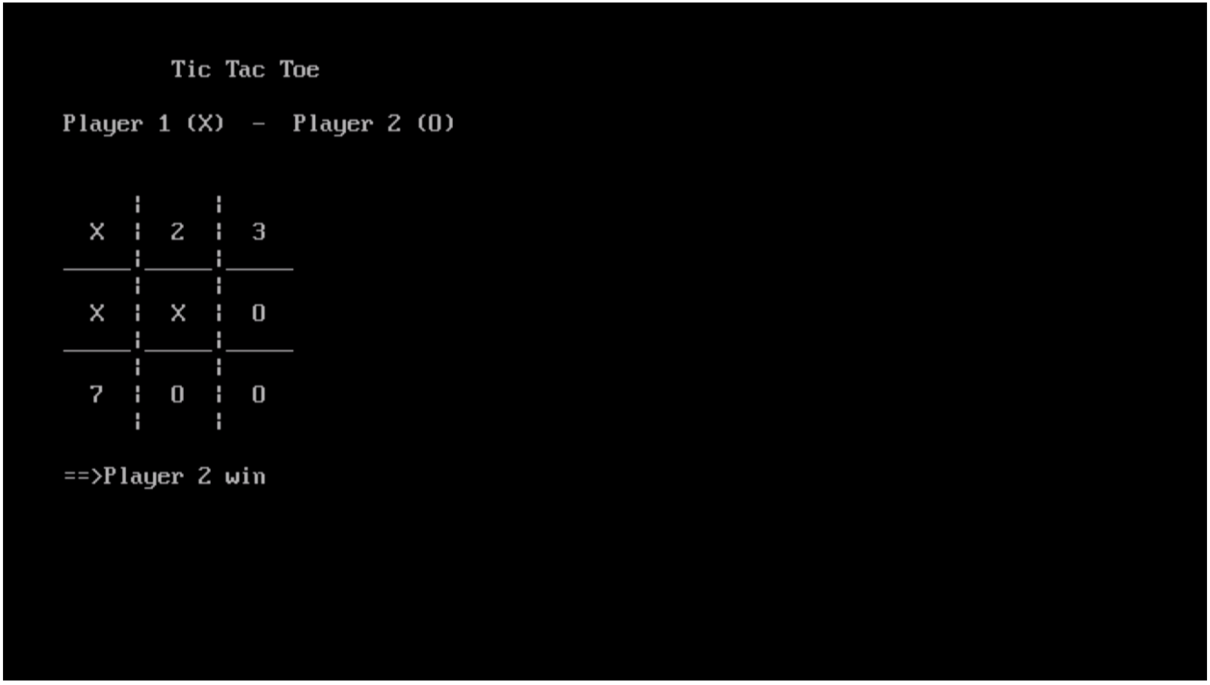
- Player 1 wins the game



```
        Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  O  |  2  |  O
_____|_____|_____
     |     |
  X  |  5  |  6
_____|_____|_____
     |     |
  X  |  X  |  9
     |     |

==>Player 1 win
```

Fig 5.2.7. The player 1 wins the game by choosing X in the form of L-shape in the bottom left corner

- Player 2 wins the game



```
        Tic Tac Toe

Player 1 (X)  -  Player 2 (O)


     |     |
  X  |  2  |  3
_____|_____|_____
     |     |
  X  |  X  |  O
_____|_____|_____
     |     |
  7  |  O  |  O
     |     |

==>Player 2 win
```

Fig 5.2.8. The player 2 wins the game by choosing O in the form of L-shape in the bottom-right corner

CHAPTER 6: - CONCLUSION

The Tic Tac Toe game is most familiar among all the age groups. Intelligence can be a property of any purpose-driven decision maker. This basic idea has been suggested many times. An algorithm of playing Tic Tac Toe has been presented and tested that works in efficient way. Overall the system works without any bugs.

- The project of Tic Tac Toe has been working efficiently for the game player and is enjoyed by the user.
- One can easily play the game with user friendly environment.
- Works very efficiently with available resources and produces results in a very short time span.
- The game is made very interactive so that the user can enjoy playing the game

CHAPTER 7: -BIBLIOGRAPHY

- https://academia.edu
- https://tutorialgateway.org
- https://wikepedia.org