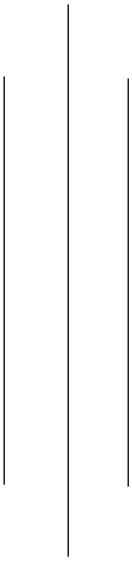
Oxford College of Engineering and Management

Gaindakot-2, Nawalpur



Tic Tac Toe (Game)

Submitted By:

Name: Gaurav Kharal Sharma	Symbol No. : 20530062
Name: Dipak Kumar Kalwar	Symbol No. : 20530059
Name: Amrit Puri	Symbol No. : 20530047
Name: Khem Narayan Khojwar	Symbol No. : 20530066

Under the Guidance of

SOBA RAJ PAUDEL

Submitted To: Pokhara University

Bachelor of Computer Application (BCA)

2020

ROLES AND RESPONSIBILITY

TIC TAC TOE (GAME)

Name	Gaurav Kharal Sharma	Dipak Kumar Kalwar	Amrit Puri	Khem Narayan Khojwar
Roles	Documentation, Asst. Programmer	System Designer, Asst. System Analyst	System Analyst, Asst. Programmer	Programmer, Documentation
Responsibility	Detailed Explanation of the Project, Coding and Debugging	Algorithm and Flowchart, Analysing and supervising the System	Analysing and supervising the System, Coding and Debugging	Coding and Debugging, Detailed Explanation of the Project
Signature				

Project Advisor

Name: Soba Raj Paudel

Signature:

Head Of Department (H.O.D.)

Name: Suresh Baral

Signature:

ABSTRACT

This game is the digital version of the very popular paper-and-pencil game “Tic Tac Toe” also known as “Noughts and Crosses”. This project has a sole purpose of providing entertainment to the users as they can choose and play with a friend or the computer itself as the computer has been given instructions to give a tough competition to the user. This project is designed in an attempt to provide the same level of satisfaction and adrenaline rush to the players that the original physical game would provide. This Game has been given as comprehensible interface as possible to allow anyone play the Game with ease.

ACKNOWLEDGEMENT

As much satisfying and beautiful the completion of this project is, the making of it was equally hard and terrifying. Hence, we leaned onto some great people without whom this project would have never been successful.

We are grateful to our guide throughout this project **Mr. Soba Raj Poudel** for the immense support and guidance showered by him onto our Project. The motivation, suggestions and criticism provided by him has led this project to its destination.

We would also like to thank **Mr. Dilip Poudel** sir for helping us decoding some logical errors that were out of our comprehension. Moreover, we would like to acknowledge the help provided by one of our seniors **Mr. Bans Raj Khojwar**, who helped us understand some of the new aspects of C programming without which this project was not possible.

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

Entertainment is a huge aspect of human existence, without which life would look very dull and boring. Playing Games is one of the many sources of recreation that we have come to discover.

People have recently taken interest into the various digital games which provide hyper-realistic experience to the players. But earlier, games were played only physically either outdoors or indoors. “Tic Tac Toe” is one of those nostalgic games that were played before the computers took over. It is played between two players who take turns marking the spaces in 3x3 grid with X or O. The player who succeeds in marking three diagonal, horizontal or vertical row is considered the winner.

This project is a simple digitized version of the Game “Tic Tac Toe” where users can easily play and enjoy the game with little knowledge of computer operation. With features like “Play with the computer” or “Play with a Friend”, this game is designed to be preferable by a large group of people.

1.1 Objectives

The project “TIC TAC TOE (GAME)” is aimed to be the go-to recreational preference of a huge group of people irrespective of their knowledge about computer or the game itself. Multi-player feature of this game allows users to play this game with their friends or colleagues and Single-player feature allows users to play the game with the computer.

1.2 Purpose, Scope and Applicability

TIC TAC TOE (GAME) is designed for anyone to replace the manual paper-and-pencil system and to have a more portable and easier to use system without any hassle to create the grids for every new game. In any boring situations or, for need of entertainment, users can play this game anywhere anytime for instant recreation. This project is designed to be as less time consuming as possible and be an instant recreation centre for huge number of people irrespective of their knowledge of the game, as detailed instructions are provided to the users.

2. REQUIREMENTS AND ANALYSIS

2.1 Problem Definition

Tic-tac-toe is a game that is traditionally played by being drawn on paper. Most adults and children think it's a simple game to pass the time; that it's a game similar to what playing on the playground means today, it's done when the power's out and there's nothing else to do. Children nowadays learn to be technologically savvy at an early age, using tablets or smartphones to learn their ABCs or hear audios of bedtime rhymes. The game of tic-tac-toe is a game of predictability. This predictability is what helps foster strategic thinking in children. They can learn through observation what their opponents' next move is and think ways on how to block them, a simple but effective version of chess. Since most of the adults and children spend most of their time in digital gadgets, they miss out fun and merits of playing Tic Tac Toe. This application is capable of restoring Tic Tac Toe digitally for adults and for future generation in a simple way.

2.2 Feasibility Analysis

A feasibility study is an analysis that considers all of a project's relevant factors including economic, technical, legal, and scheduling considerations to ascertain the likelihood of completing the project successfully. The main goal of feasibility study is not to solve the problem but to achieve the scope. Feasibility Analysis includes following study:

2.2.1 Technical Feasibility

During technical feasibility study, first of all the system analyst identified the existing computer system (hardware and software) then it was determined that "Tic Tac Toe" can easily run in almost all of the computer system (hardware and software) available. So, the analyst recommended that the proposed system is feasible.

2.2.2 Operational Feasibility

Since our system is technically feasible so the next step is to determine whether it is operationally feasible or not. So, we determined to go ahead towards our goals because we have experienced, skill-full and trained manpower to develop the proposed system. That's why the system is feasible in operational too. We are confident on developing the proposed system that the user really wants and can handle easily.

2.2.3 Social Feasibility

After investigating in the society, we decided that the proposed system would be acceptable to the people or society because after releasing this system, everyone will be able to enjoy paper-and-pencil game “Tic Tac Toe” also known as “Noughts and Crosses” digitally.

2.3 Hardware and Software Requirement

The minimum software requirement for the effective operation of this project:

- Operating system: Windows XP or above versions
- Application: Dev C++, Turbo C, Code::Blocks

Minimum hardware requirement:

- Processor: Pentium or above
- Hard disk: 100MB or above
- Output device: Monitor
- Input device: Keyboard
- RAM :512MB or above

3. SYSTEM DESIGN

3.1 Algorithm

An algorithm is a set of instructions in a sequential order used to solve a particular problem, based on conducting a sequence of specified actions. In mathematics and computer science, an algorithm usually means a small procedure that solve a recurrent problem.

An algorithm of the purpose system “TIC TAC TOE ” is as follow:

Step 1: START

Step 2: Display the following menu

1. Play with X
2. Play with O
3. Play with friend
4. Exit

Step 3: Display “Enter your choice”

Step 4: If choice = 1

Player = X and Computer = O

goto step 8

Step 5: If choice = 2

Player = O and Computer = X

goto step 8

Step 6: If choice = 3

Player 1 = X and Player 2 = O

Goto step 8

Step 7: If choice = 4

Exit the game

Step 8: Draw the board

Step 9: Ask for input.

Step 10: If input = 1, then (row=1, column=1) is marked.

If input = 2, then (row=1, column=2) is marked.

If input = 3, then (row=1, column=3) is marked.

If input = 4, then (row=2, column=1) is marked.

If input = 5, then (row=2, column=2) is marked.

If input = 6, then (row=2, column=3) is marked

If input = 7, then (row=3, column=1) is marked.

If input =8, then (row=3, column=2) is marked.

If input =9, then (row=3, column=3) is marked.

If input > 9 and if input is already marked, display “Invalid Input”

Step 11: If a player succeeds to mark 3 horizontal, vertical or diagonal boxes, goto step 13

If all the boxes are marked goto step 12

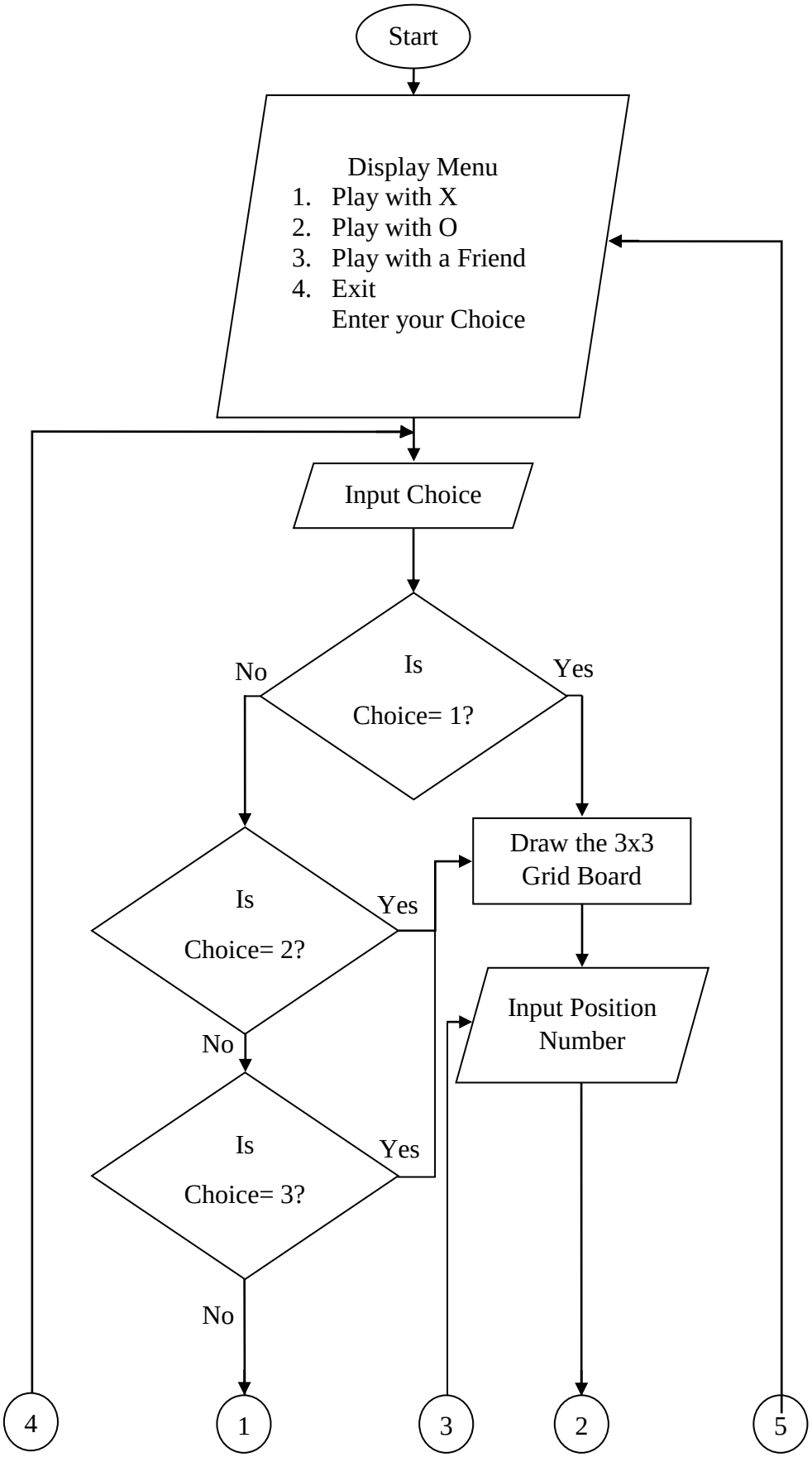
else Alternate turns and goto step 9

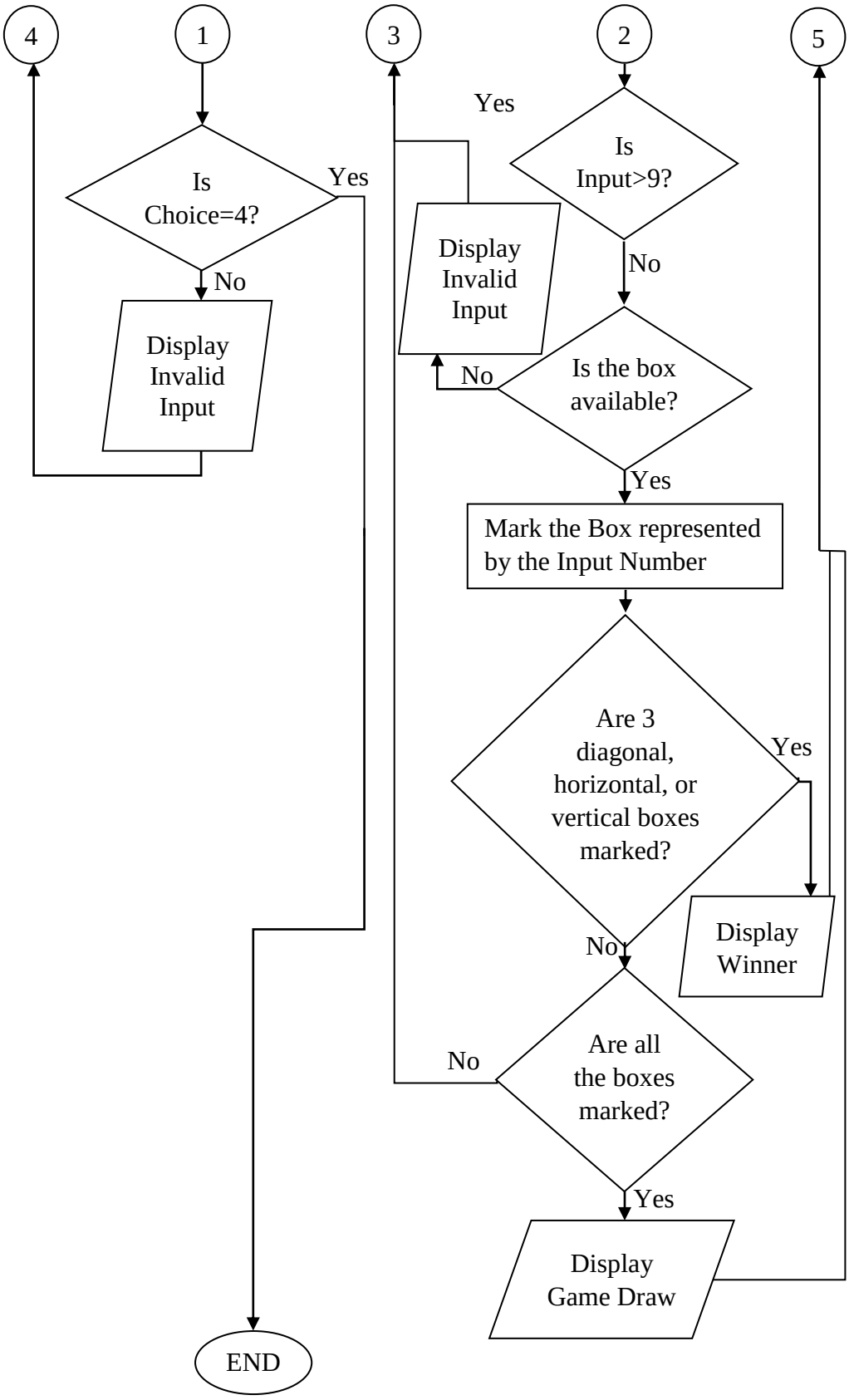
Step 12: Display “Game Draw”, goto step 2

Step 13: The player who succeeds is displayed Winner, goto step 2

Step 14: END

3.2 Flowchart




4
1
3
2
5
Is Choice=4?
Yes
No
Display Invalid Input
Yes
Is Input>9?
Display Invalid Input
No
No
Is the box available?
Yes
Mark the Box represented by the Input Number
Are 3 diagonal, horizontal, or vertical boxes marked?
Yes
Display Winner
No
No
Are all the boxes marked?
Yes
Display Game Draw
END

4. SYSTEM DEVELOPMENT

System Development is the process of defining, designing, testing and implementing a new software application or program. It can include the internal development of customized system, the creation of database or the acquisition of reckoned software.

To start the development of a new computer program, the initial stage is coding. The program finds the desired result with the creation or the design of coding. Simply, this means the construction of the system. So, programmers are fully responsible for the activities of the system. Here, we conduct the coding to run the program and it also provides the required information to the operator.

Coding:

```
#include<stdio.h>
#include<conio.h>
#include <windows.h>
int board[10] = {2,2,2,2,2,2,2,2,2,2};
char square[10] = { ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ' };
int turn = 1,flag = 0;
int player,comp,player1,player2;
void menu();
void go(int n);
void start_game();
void check_draw();
void draw_board();
void player_first();
void two_player();
int checkwin();
void board1();
void clear_board1();
void put_X_O(char ch,int pos);
void clear_board();
COORD coord= {0,0}; // this is global variable
//center of axis is set to the top left cornor of the screen
void gotoxy(int x,int y)
{
   coord.X=x;
   coord.Y=y;

SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),coord);
```

```
}

void main()
{
   system("cls");
   menu();
   getch();
}

void menu()
{
   int choice;
   again:
   clear_board();
   flag=0;
   system("cls");
   printf("\n\t\t\t--------MENU--------");
   printf("\n\t\t\t1 : Play with X");
   printf("\n\t\t\t2 : Play with O");
   printf("\n\t\t\t3 : Play with a Friend");
   printf("\n\t\t\t4 : exit");
   printf("\n\t\t\tEnter your choice:>");
   scanf("%d",&choice);
   turn = 1;
   switch (choice)
   {
   case 1:
      player = 1;
      comp = 0;
      system("cls");
      printf("\n\n\n\t\t\t player = 'X' and computer = 'O'\n");
      player_first();
      break;
   case 2:
      player = 0;
      comp = 1;
      system("cls");
      printf("\n\n\n\t\t\t computer = 'X' and player = 'O'\n");
      start_game();
      break;
   case 3:
```

```
        player1=1;
        player2=0;
        two_player();
        break;
    case 4:
        exit(1);
    default:
        printf("\n\t\t\tInvalid input \n\t\t\t Press any key to choose again\n");
    }
    getch();
    goto again;
}

int make2()
{
    if(board[5] == 2)
        return 5;
    if(board[2] == 2)
        return 2;
    if(board[4] == 2)
        return 4;
    if(board[6] == 2)
        return 6;
    if(board[8] == 2)
        return 8;
    return 0;
}

int make4()
{
    if(board[1] == 2)
        return 1;
    if(board[3] == 2)
        return 3;
    if(board[7] == 2)
        return 7;
    if(board[9] == 2)
        return 9;
    return 0;
}
```

```
int posswin(int p)
{
   int i;
   int check_val,pos;

   if(p == 1)
      check_val = 18;
   else
      check_val = 50;

   i = 1;
   while(i<=9)//row check
   {
      if(board[i] * board[i+1] * board[i+2] == check_val)
      {
         if(board[i] == 2)
            return i;
         if(board[i+1] == 2)
            return i+1;
         if(board[i+2] == 2)
            return i+2;
      }
      i+=3;
   }

   i = 1;
   while(i<=3)//column check
   {
      if(board[i] * board[i+3] * board[i+6] == check_val)
      {
         if(board[i] == 2)
            return i;
         if(board[i+3] == 2)
            return i+3;
         if(board[i+6] == 2)
            return i+6;
      }
      i++;
   }

   if(board[1] * board[5] * board[9] == check_val)
```

```
    {
        if(board[1] == 2)
            return 1;
        if(board[5] == 2)
            return 5;
        if(board[9] == 2)
            return 9;
    }

    if(board[3] * board[5] * board[7] == check_val)
    {
        if(board[3] == 2)
            return 3;
        if(board[5] == 2)
            return 5;
        if(board[7] == 2)
            return 7;
    }
    return 0;
}

void go(int n)
{
    if(turn % 2)
        board[n] = 3;
    else
        board[n] = 5;
    turn++;
}

void player_first()
{
    int pos;

    check_draw();
    draw_board();
    gotoxy(30,18);
    printf("Your Turn :> ");
    scanf("%d",&pos);

    if(board[pos] != 2)
```

```
        player_first();

    if(pos == posswin(player))
    {
        go(pos);
        draw_board();
        gotoxy(30,20);
        printf("Player Wins");
        getch();
        printf("\n\t\t\tPress any key to go menu\n");
        getch();
        menu();
    }

    go(pos);
    draw_board();
    start_game();
}

void start_game()
{
    int pos;
    if(posswin(comp))
    {
        go(posswin(comp));
        flag = 1;
    }
    else if(posswin(player))
        go(posswin(player));
    else if(make2())
        go(make2());
    else
        go(make4());
    draw_board();

    if(flag)
    {
        gotoxy(30,20);
        printf("Computer wins");
        getch();
        printf("\n\t\t\tPress any key to go menu\n");
```

```
        getch();
        menu();
    }
    else
        player_first();
}

void check_draw()
{
    if(turn > 9)
    {
        gotoxy(30,20);
        printf("Game Draw");
        printf("\n\t\t\tPress any key to go menu \n");
        getch();
        menu();
        system("cls");
        getch();
    }
}

void draw_board()
{

    int j;
    for(j=9; j<17; j++)
    {
        gotoxy(35,j);
        printf("|       |");
    }
    gotoxy(28,11);
    printf("-----------------------");
    gotoxy(28,14);
    printf("-----------------------");
    for(j=1; j<10; j++)
    {
        if(board[j] == 3)
            put_X_O('X',j);
        else if(board[j] == 5)
            put_X_O('O',j);
    }
```

```
}

void put_X_O(char ch,int pos)
{
   int m;
   int x = 31, y = 10;
   m = pos;
   if(m > 3)
   {
      while(m > 3)
      {
         y += 3;
         m -= 3;
      }
   }
   if(pos % 3 == 0)
      x += 16;
   else
   {
      pos %= 3;
      pos--;
      while(pos)
      {
         x+= 8;
         pos--;
      }
   }
   gotoxy(x,y);
   printf("%c",ch);
}

void clear_board()
{
   int j;
   for(j=1;j<10;j++)
   {
      board[j]=2;
   }

}
```

```
void two_player()

    {
       int player = 1, i, choice;

       char mark;
       clear_board1();

       do
       {
          board1();
          player = (player % 2) ? 1 : 2;

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

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

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

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

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

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

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

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

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

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

```
            square[8] = mark;

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

        else
        {
            printf("Invalid move ");

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

    board1();

    if (i == 1)
        printf("\t\t\t==>\aPlayer %d win ", --player);
    else
        printf("\t\t\t==>\aGame draw");
    printf("\n\t\t\tEnter any two key to go menu\n");
    getch();
}

int checkwin()
{

  if (!(square[1] == ' ' || square[2] == ' ' || square[3] == ' '))
    {
      if (square[1] == square[2] && square[2] == square[3])
      return 1;
    }

  if (!(square[4] == ' ' || square[5] == ' ' || square[6] == ' '))
    {
      if (square[4] == square[5] && square[5] == square[6])
      return 1;
    }
```

```
if (!(square[7] == ' ' || square[8] == ' ' || square[9] == ' '))
   {
      if (square[7] == square[8] && square[8] == square[9])
      return 1;
   }

if (!(square[1] == ' ' || square[4] == ' ' || square[7] == ' '))
   {
      if (square[1] == square[4] && square[4] == square[7])
      return 1;
   }

if (!(square[2] == ' ' || square[5] == ' ' || square[8] == ' '))
   {
      if (square[2] == square[5] && square[5] == square[8])
      return 1;
   }

if (!(square[3] == ' ' || square[6] == ' ' || square[9] == ' '))
   {
      if (square[3] == square[6] && square[6] == square[9])
      return 1;
   }

if (!(square[1] == ' ' || square[5] == ' ' || square[9] == ' '))
   {
      if (square[1] == square[5] && square[5] == square[9])
      return 1;
   }
if (!(square[3] == ' ' || square[5] == ' ' || square[7] == ' '))
   {
      if (square[3] == square[5] && square[5] == square[7])
      return 1;
   }

if (square[1] != ' ' && square[2] != ' ' && square[3] != ' ' &&
   square[4] != ' ' && square[5] != ' ' && square[6] != ' ' && square[7]
   != ' ' && square[8] != ' ' && square[9] != ' ')
   return 0;
else
   return -1;
```

```
}

void board1()
{
   system("cls");
   printf("\n\n\t\t\tTic Tac Toe\n\n");

   printf("\t\t\tPlayer 1 = 'X'  -  Player 2 = 'O'\n\n\n");


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

   printf("\t\t\t-----------------\n");
   printf("\t\t\t     |     |     \n");

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

   printf("\t\t\t-----------------\n");
   printf("\t\t\t     |     |     \n");

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

void clear_board1()
{

   square[1]= ' ';
   square[2]= ' ';
   square[3]= ' ';
   square[4]= ' ';
   square[5]= ' ';
   square[6]= ' ';
   square[7]= ' ';
   square[8]= ' ';
   square[9]= ' ';
}
```

5. SYSTEM TESTING

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. Once the system is developed, the testing of the system is necessary as it helps us to know whether the program is executable or not. The bugs/errors should be debugged as soon as possible to run the system smoothly.

Screenshot of Output:

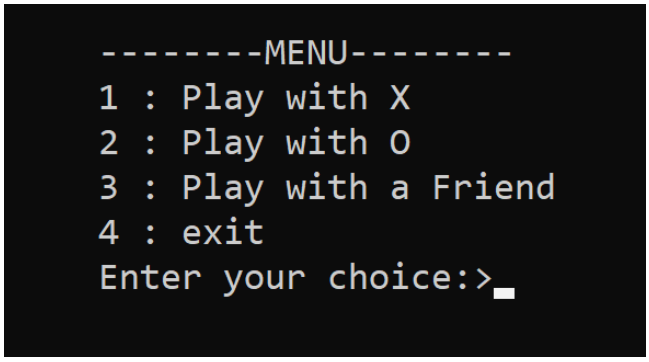


Figure 5.1: Main Menu

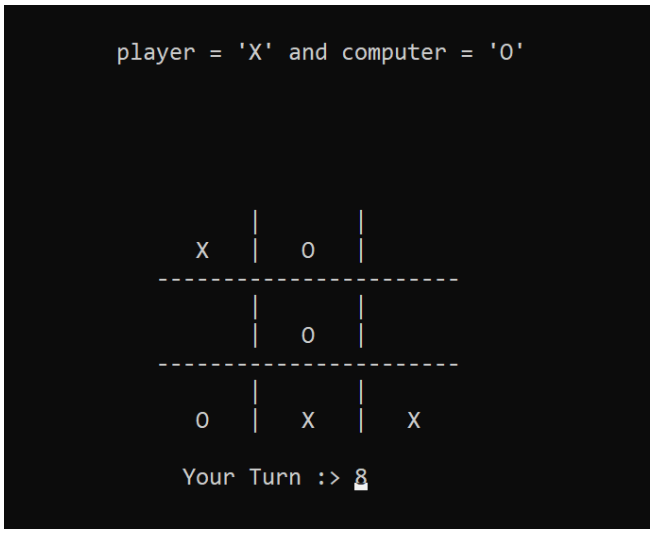


Figure 5.2: Playing Against Computer

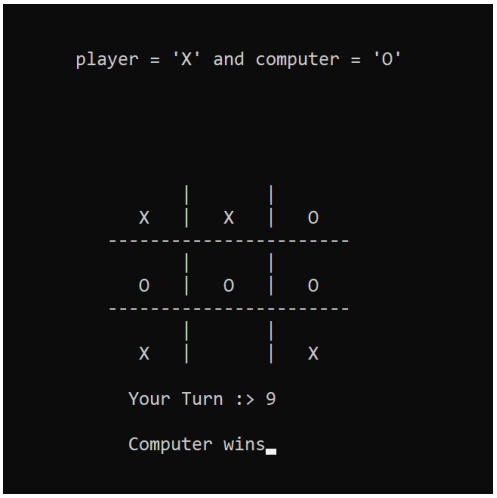


Figure 5.3: Computer Wins

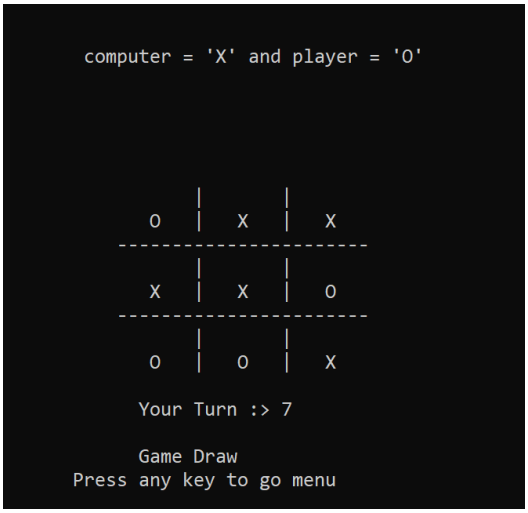


Figure 5.4: Game Draw

Figure 5.5: Playing Against A Friend (Multi-player Mode)

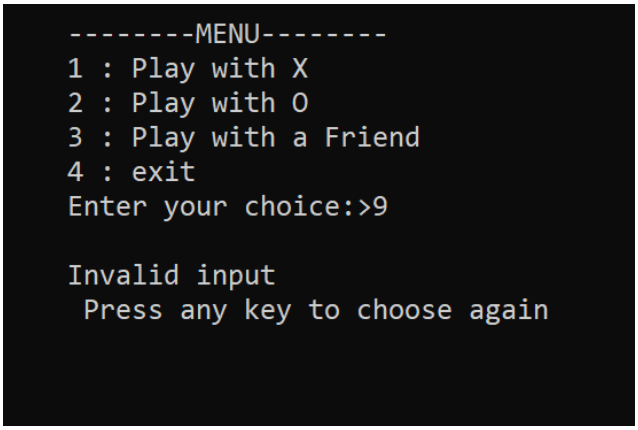


Figure 5.6: Invalid Input

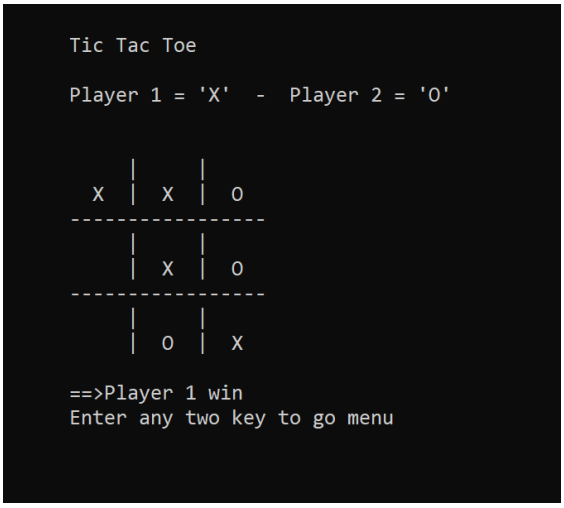


Figure 5.7: Enter any Two Key to go to Menu

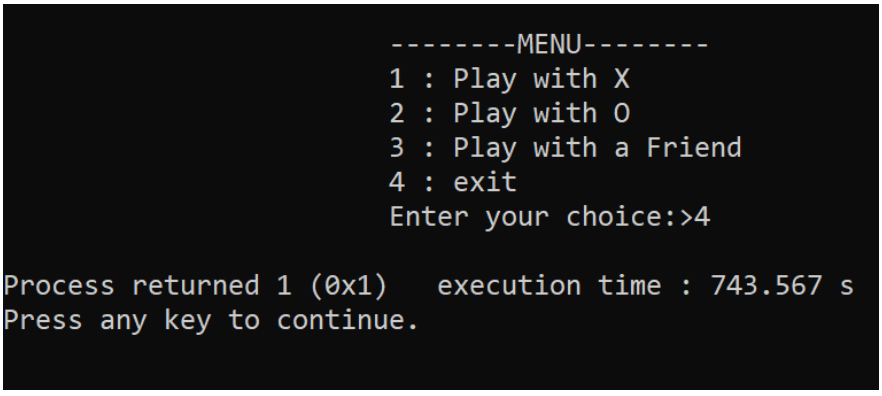


Figure 5.8: Exiting the System

6. SYSTEM IMPLEMENTATION

System implementation is the process of using the designed system in the real use. So, its purpose is to test whether the system is valid or not. As we know, creating software us one thing and the implementation of the created software is another. The process of implementing software is much difficult as compared to the task of creating the project. As the software is developed, the implementation is necessary to prove it as error free and easily workable design. Accordingly, upgradation will be implementing the system.

7. CONCLUSION

The “TIC TAC TOE (GAME)” is familiar among all the age groups of people. We dedicated a huge amount of our time and knowledge into this game to make it as less defective as possible and provide tremendous satisfaction to the user. But like every project ever made, our very own game also inhibits some imperfections which we tried very hard to eradicate.

The final program turned out to be better than we expected. We are confident that this game can be readily used by non-programming personal avoiding human handled chance of error. An algorithm of playing Tic Tac Toe has been presented and tested that works in efficient way. Overall, the system works without any bugs.

7.1 Features

1. Basic AI opponent.
2. Simple Character User Interface (CUI).
3. It has been tested and implemented successfully.
4. Runs almost on all computer systems.
5. User Friendly.
6. Supports single-player and multi-player.

7.2 Limitations

1. It is limited to CUI, drawback of C language.
2. Gameplay is Short and Simple

7.3 Future Scope of the Project

This game is targeted towards the users who want a simple digital version of the popular game- Tic Tac Toe. Simple codes and lack of graphics in this system might not attract a huge amount of people but this game is good enough for being a go-to recreation device. The AI provided can be improved to make the game against the computer harder than now. Introduction of graphics would add a completely new dynamic to the game and would attract more people into the colourful Tic Tac Toe.

REFERENCES

Book Reference:

Bhandari, Hari Prasad. Programming Logics and Techniques.
Gaindakot: Oxford College of Engineering and Management (OCEM), 2018.
Quality Press.

Web Reference:

Programiz: https://www.programiz.com/

Tutorialspoint: https://www.tutorialspoint.com/

Wikipedia: https://computersciencewiki.org/