A Mini Project Report

On

Tic-Tac-Toe

Submitted in partial fulfillment of requirements for the Course CSE18R272
- JAVA PROGRAMMING

Bachelor of Technology

In

Computer Science and Engineering

Submitted By

M Hemanth
9918004064

M Sasi Chandra
9918004074

Under the guidance of

Dr.R. RAMALAKSHMI

Associate Professor

Department Of Computer Science and Engineering

Kalasalingam Academy of Research and Education

Anand Nagar,Krishnankoil-626126
APRIL 2020

ABSTRACT

When it comes to a game which has no end Tic-Tac-Toe is one of such Legendary games. This game has So much History Games ,played on three-in-a-row boards can be traced back to ancient Egypt, where such game boards have been found on roofing tiles dating from around 1300 BCE. In 1952, OXO (or Noughts and Crosses), developed by British computer scientist Sandy Douglas for the EDSAC computer at the University of Cambridge, became one of the first known video games.Tic-Tac-Toe is most often played by young children, who often have not yet discovered the optimal strategy.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 that deals with the searching of game trees. It is straightforward to write a computer program to play tic-tac-toe perfectly or to enumerate the 765 essentially different positions (the state space complexity) or the 26,830 possible games up to rotations and reflections (the game tree complexity) on this space.This is a two player game. Tic-tac-toe is the game where n equals 3 and d equals 2. If played optimally by both players, the game always ends in a draw, making tic-tac-toe a futile game.

DECLARATION

I here by declare that the work presented in this report entitled "**Tic-Tac-Toe** ", in partial fulfilment of the requirements for the course CSE18R272-Java Programing and submitted in **Deemed to be University** is an authentic record of our own work carried out during the period from Jan 2020 under the guidance of mr.**Dr. R. Ramalakshmi** Associate Professor.

The work reported in this has not been submitted by us for the award of any other degree of this has not been submitted by me fir the award of any other degree of this or any other institute.

M HEMANTH
9918004064
M SASI CHANDRA
9918004074

ACKNOWLEDGEMENT

I express my deep gratitude to Mr. **Friends** and Mr.**Family Members** for their valuable guidance throughtout my training.

I would also like to thank Mr."Kalvivallal"Triu. **T.Kalasalingam B.com**,Founder Chairman,Mr"llayavallal"**Dr.k.Sridharan Ph.D**,Vice President (Academic),**Mr.S.Arjun Kalasalingam M.s.**, Vice President(Administration), Mr**Dr.R.Nagaraj**,Vice Chancellor,Mr.**Dr.V.Vasudevan.Ph.D**,Registrar,Ms.**Dr.P.Deepalalakshmi.Ph.D**,Dean (School of Computing).and also special thanks to Mr.Dr.A.Franics saviour Devaraj.Head Department of CSE,Kalasalingam Academy of Research of Education for granting the permission and providing necessary facilities to carry out Project work.

I would like to express my special apppreciation and profound thanks to my enthusisstic Project supervisor Ms.**Dr.R.Ramalakshmi Ph.D**, Associate Professor at Kalasalingam Academy of Research and Education[KARE] for her inspiring guidance,contant encouragement with my work during all stages. I am extremely glad that i had a chance to do my Project under our Guide for all the time he has spent with me in the discussions. and during the most difficult times when writing this report,he gave the moral supprot and the freedom i needed to move on.

M HEMANTH
9918004064
M SASI CHANDRA
9918004074

0.0.1 Objectives

1. INTRODUCTION
    - 1.1 Problem statement
    - 1.2 Scope
    - 1.3 Conventions Uses
    - 1.4 Product Features
    - 2. System Requirements
    - 3. Tic-Tac-Toe game Code
    - 4. IMPLEMENTATION
    - 5. References

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Problem Statement

The main aim of this project to illustrate the Gameplay and Design of Paper Pencil Game Tic-Tac-Toe using Java We want to write an application to play Tic-Tac-Toe (Naughts and Crosses, Tic-Tac-Toe, X's and O's). Starts with a GUI board. First, the player has to click on any box to start the game.It contains the Minimax algorithm; which is a decision rule used for a two-player game.A simple 2D GUI is provided for easy gameplay.The gameplay design is so simple that user won't find it difficult to use and understand.

1.2 Scope Since this game is Unbeatable by any game when it comes to Bore Breaker. You can play this game with your Sibling,Cousins. Especially at Quarantine Times. As we all know the Current Covid-19 Pandemic Situation. By playing in PC,Laptop this makes the game more Fun and Enjoyable.

1.3 Conventions used: This is Two player Game. So as traditionally we create X and O as first and Second player respectively.The one who starts the game first becomes X and the player who plays the next move becomes O.

1.4 Product features

No Internet Connection is Required as this is a Offline game. Can be played any time any where. You just need a Pc or a Laptop. Lightweight code. Very Less memory usage.

2. System Requirements

Just an Working Operating System is required,That's it. Beacause the code barely use the memory(RAM) and Space(Storage-ROM) since it run in the Compiler itsself. jdk is required for the compilation and running. Since the language of code is Java.

3.Tic-Tac-Toe game Code

4. IMPLEMENTATION We tried hardly to reduce the lines of code as much as possible for Faster Compilation and Smooth Running. This program barely uses the Resources. Our Code is Faster, Lighter, Better. About Minimax Minimax is a decision rule used in decision theory, game theory, statistics and philosophy for minimizing the possible loss for a worst case (maximum loss) scenario. Originally formulated for two-player zero-sum game theory, covering both the cases where players take alternate moves and those where they make simultaneous moves, it has also been extended to more complex games and to general decision making in the presence of uncertainty. Rules:- 1. The game is played on a grid that's 3 squares by 3 squares.

2. You are X, your friend (or the computer in this case) is O. Players take turns putting their marks in empty squares.

3. The first player to get 3 of her marks in a row (up, down, across, or diagonally) is the winner.

4. When all 9 squares are full, the game is over. If no player has 3 marks in a row, the game ends in a tie.

Chapter 2

PROJECT DESCRIPTION

Tic-Tac-Toe games is a popular two player games played on a three by three grid, is a pencil-and-paper game. Also is to developed an interacting game using Java programming language, where two users can come together and play on their respective computer a player can play up to three times, there will be a record of every game played by every user, a user is entitled to play only three times so that he will allow another user to come in, Tic-Tac-Toe game enable user to think fast and also enable a user to block an opponent not to win, any user who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the games.

Description of the Tic-Tac-Toe game:-

Tic-Tac-Toe game is a popular two-player game played on a three grid we are going to make a project that plays a decent game of Tic-Tac-Toe Game. First of all, you should understand how a mid-game Tic-Tac-Toe Game board will be represented in this project, as a list of what is in each of the nine squares. We will represent the board as a diagram whose items are in the order of the numbers. The program will allow a human user to play against another user the goal in this project is to develop an application that will determine the users next move based on the current position, Tic-Tac-Toe is not a very challenging game for people). This block of code will report the users next move given the current configuration. It takes two inputs, the current position and whether the user is playing X' or O'.

Importance of Playing Games (Tic-Tac-Toe)

(i) Develop the thinking (logical) abilities of a child (player)

(ii) Develop the predictive abilities of the players,as they can anticipate the thought and more of theother player. This would improve their assessingability.

(iii) It prepares the players for more complex games because it helps them to think of multiple things at once.

Background of the Study

This game is very popular and is fairly simple by itself. It is actually a two player game. In our game, it is 3 x 3 squares. The goal of Local Tic-Tac-Toe is to be one of the players to get three same symbols in a row – horizontally, vertically or diagonally – on a 3 x 3 grid. The game can be played by two players (player x' and player o'). A player can choose between two symbols with his opponent, usual games use “X” and “O”. If first player choose “X” then the second player have to play with “O” and vice versa. A player marks any of the 33 squares with his symbol (may be “X” or “O”) and his aim is to create a straight line horizontally or vertically or diagonally with two intensions:

(i) Create a straight line before his opponent to win the game.

(ii) Restrict his opponent from creating a straight line first.

In this case logically no one can create a straight line with his own symbol, the game results a tie. Hence there are only three possible results – a player wins, his opponent (human) wins or it's a tie.

This game's Speciality:-

Tic-Tac-Toe game is something called a zero-sum game. A game where the outcome always sums up to zero in the end. Furthermore this is a perfect information game where the game state is completely open to everyone. A user knows everything about it, no hidden cards or anything like that. Therefore a user can play perfectly since no luck is involved. (Playing Local Tic-Tac-Toe game perfectly doesn't mean a user will always win). For the purpose this means there will always be a tie.

Expected Contribution to Knowledge

The importance of this project work cannot be over emphasized. The purpose of this project is to understand the game better and a lot easier, the Tic-Tac-Toe game will also help the user to improve in fast thinking everybody will benefit from the game. A lot of people can learn at different time and places. The user can also learn to think fast when playing. Develop a two player game of Tic-Tac-Toe, using Java programming language.

Chapter 3

CONCLUSION

In the conclusion of this project, I would like to say that java is a fun and easy programming language and while creating a project like this, it has been a good experience and also helped in the development of my creativity and logical thinking. I would be more than happy to work on other projects in java because it's just amazing to work with java. The program is working and I hope, it's also bug-free. And coming to this game this game never fells bored. So i hope many people would use my code when they wanted to play in PC or Laptop

*

Appendices

SOURCE CODE

```
import java.util.Scanner;


public class TicTacToe
{
    private int counter;
    private   char posn[]=new char[10];
    private   char player;


    public static void main(String args[])
    {
        String ch;
        TicTacToe Toe=new TicTacToe();
        do{
            Toe.newBoard();
            Toe.play();
            System.out.println ("Would˽you˽like˽to˽play
               ↪ ˽again˽(Enter˽'yes')?˽");
            Scanner in =new Scanner(System.in);
            ch=in.nextLine();
            System.out.println("ch˽value˽is˽˽"+ch);
        }while (ch.equals("yes"));


    }
    public  void newBoard()
    {

        char posndef[] = {'0','1', '2', '3', '4', '5',
           ↪ '6', '7', '8', '9'};
        int i;
        counter = 0;
        player = 'X';
        for (i=1; i<10; i++) posn[i]=posndef[i];
        currentBoard();
    }
    public  String currentBoard()
    {
        System.out.println( "\n\n" );
        System.out.println(  "\n\n" );
        System.out.println(  "\n\n\t\t" + posn [1] + "␣
           ↪ ␣␣|␣" +posn [2]+ "␣␣|␣" +posn [3]);
        System.out.println(  "␣\t\t␣␣␣␣|␣␣␣␣|␣␣␣" );
        System.out.println(  "␣\t\t␣___|____|___␣" );
        System.out.println(  "\n\n\t\t" +posn [4]+ "␣␣␣
           ↪ |␣" +posn [5]+ "␣␣|␣" +posn [6]);
        System.out.println(  "␣\t\t␣␣␣␣|␣␣␣␣|␣␣␣" );
        System.out.println(  "␣\t\t␣___|____|___␣" );
        System.out.println(  "\n\n\t\t" +posn [7]+ "␣␣␣
           ↪ |␣" +posn [8]+ "␣␣|␣" +posn [9]);
        System.out.println(  "␣\t\t␣␣␣␣|␣␣␣␣|␣␣␣" );
        System.out.println(  "␣\t\t␣␣␣␣|␣␣␣␣|␣␣␣" );
        System.out.println(  "\n\n" );
        return "currentBoard";
    }

    public  void play()
    {
        int spot;
        char blank = '␣';

        System.out.println(  "Player␣" + getPlayer() +"
           ↪ ␣will␣go␣first␣and␣be␣the␣letter␣'X'" );

        do {
            currentBoard();                // display
               ↪ current board

            System.out.println(  "\n\n␣Player␣" +
               ↪ getPlayer() +"␣choose␣a␣posn." );

            boolean posTaken = true;
            while (posTaken) {
                // System.out.println( "position is
                   ↪ taken, please enter a valid space
                   ↪ ");
                Scanner in =new Scanner (System.in);
                spot=in.nextInt();
                posTaken = checkPosn(spot);
                if(posTaken==false)
                posn[spot]=getPlayer();
            }

            System.out.println( "Nice␣move." );

            currentBoard();                   // display
               ↪ current board

            nextPlayer();
        }while ( checkWinner() == blank );

    }

    public  char checkWinner()
    {
        char Winner = '␣';

        // Check if X wins
        if (posn[1] == 'X' && posn[2] == 'X' && posn[3]
           ↪  == 'X') Winner = 'X';
        if (posn[4] == 'X' && posn[5] == 'X' && posn[6]
           ↪  == 'X') Winner = 'X';
        if (posn[7] == 'X' && posn[8] == 'X' && posn[9]
           ↪  == 'X') Winner = 'X';
        if (posn[1] == 'X' && posn[4] == 'X' && posn[7]
           ↪  == 'X') Winner = 'X';
        if (posn[2] == 'X' && posn[5] == 'X' && posn[8]
           ↪  == 'X') Winner = 'X';
        if (posn[3] == 'X' && posn[6] == 'X' && posn[9]
           ↪  == 'X') Winner = 'X';
        if (posn[1] == 'X' && posn[5] == 'X' && posn[9]
           ↪  == 'X') Winner = 'X';
        if (posn[3] == 'X' && posn[5] == 'X' && posn[7]
           ↪  == 'X') Winner = 'X';
        if (Winner == 'X' )
```

```
{System.out.println("Player1␣wins␣the␣game." );
    return Winner;
}

// Check if O wins
if (posn[1] == 'O' && posn[2] == 'O' && posn[3]
   ↪ == 'O') Winner = 'O';
if (posn[4] == 'O' && posn[5] == 'O' && posn[6]
   ↪ == 'O') Winner = 'O';
if (posn[7] == 'O' && posn[8] == 'O' && posn[9]
   ↪ == 'O') Winner = 'O';
if (posn[1] == 'O' && posn[4] == 'O' && posn[7]
   ↪ == 'O') Winner = 'O';
if (posn[2] == 'O' && posn[5] == 'O' && posn[8]
   ↪ == 'O') Winner = 'O';
if (posn[3] == 'O' && posn[6] == 'O' && posn[9]
   ↪ == 'O') Winner = 'O';
if (posn[1] == 'O' && posn[5] == 'O' && posn[9]
   ↪ == 'O') Winner = 'O';
if (posn[3] == 'O' && posn[5] == 'O' && posn[7]
   ↪ == 'O') Winner = 'O';
if (Winner == 'O' )
{
    System.out.println( "Player2␣wins␣the␣game.
       ↪ " );
return Winner; }

// check for Tie
for(int i=1;i<10;i++)
{
    if(posn[i]=='X' || posn[i]=='O')
    {
        if(i==9)
        {
            char Draw='D';
            System.out.println("␣Game␣is␣
                ↪ stalemate␣");
            return Draw;
        }
        continue;
        }
        else
        break;

    }

    return Winner;
}

public  boolean checkPosn(int spot)
{


    if (posn[spot] == 'X' || posn[spot] == 'O')
    {
        System.out.println("That␣posn␣is␣already␣
            ↪ taken,␣please␣choose␣another");
        return true;
    }
    else {
        return false;
    }

    //  counter++;
    //    return false;
}



public  void nextPlayer()
{
    if (player == 'X')
    player = 'O';
    else player = 'X';

}

public String getTitle()
{
    return "Tic␣Tac␣Toe" ;
    }

    public  char getPlayer()
    {
        return player;
    }

}
```