****

**BAHRIA UNIVERSITY**

**(Karachi Campus)**

*Department of Software Engineering*

**ARTIFICIAL INTELLIGENCE**

**PROJECT REPORT**

**TIC TAC TOE GAME**

**USING MINMAX ALGORITHM**

**Submitted To:**

**Sir Zia Ul Islam / Sir Rehan Baig**

**BSE 5B**

**Group Member:**

**M Sohail Zakir (51742)**

**Ammara Riaz (51730)**

Contents

[1. Objectives: 3](#_Toc61873029)

[2. Overview: 3](#_Toc61873030)

[2.1 Players: 3](#_Toc61873031)

[2.2 Theory of Game: 4](#_Toc61873032)

[3. Core Logic - AI: 5](#_Toc61873033)

[3.1 First move: 5](#_Toc61873034)

[3.2 Second move: 5](#_Toc61873035)

[Case 1: 6](#_Toc61873036)

[Case 2: 6](#_Toc61873037)

[Case 3: 6](#_Toc61873038)

[Case 4: 7](#_Toc61873039)

[3.3 Third and fourth move: 7](#_Toc61873040)

[4. Core Logic - Humans: 8](#_Toc61873041)

[6. Reference: 8](#_Toc61873042)

[https://inventwithpython.com/chapter10.html 8](#_Toc61873043)

[https://codereview.stackexchange.com/questions/108738/python-tic-tac-toe-game 8](#_Toc61873044)

[https://dev.to/nestedsoftware/tic-tac-toe-with-the-minimax-algorithm-5988 8](#_Toc61873045)

[Books: 8](#_Toc61873046)

**ABSTRACT**

Tic-Tac-Toe is one of the paper-and-pencil games. This game requires two players in 3x3 grid with Player 1 acts as “O” and Player 2 acts as “X”, or vice vers. The objective of this game is to take place of three connecting grids in a horizontal, vertical, or diagonal way/fork. This research findings explains about the common AI algorithm for Tic-Tac-Toe game, including the strategy in order to beat human player.

**INTRODUCTION**

Python is a high-level, interpreted and general-purpose dynamic programming language that focuses on code readability. The syntax in Python helps the programmers to do coding in fewer steps as compared to Java or C++. The Python is widely used in bigger organizations because of its multiple programming paradigms. They usually involve imperative and object-oriented functional programming. It has a comprehensive and large standard library that has automatic memory management and dynamic features.

Our project is to implement the Tic-Tac-Toe game which will features like single player and two player modes. In single player mode we have used game theory logics like minimax algorithms to determine the best move that the computer plays.The project was tested and it turned out to be much accurate.

# Objectives:

Our project name is Tic-Tac-Toe game. This game is very popular and is fairly simple by itself. It is actually a two player game. In this game, there is a board with *n* x *n* squares. In our game, it is 3 x 3 squares.

The goal of 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.

# Overview:

This game can be played in a 3x3 grid (shown in the figure 2.1) .The game can be played by two players. There are two options for players:

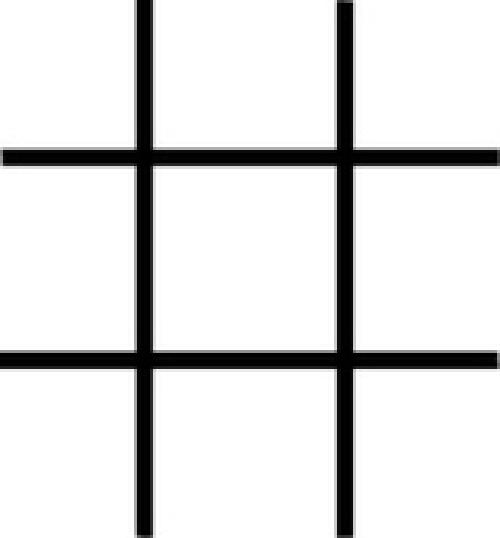
* 1. Human (b) Computer

Figure: 2.1

## Players:

For the option human, both the players are human and for the option computer, the first player is human and the second player is computer

## Theory of Game:

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 3x3 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:

1. Create a straight line before his opponent to win the game.
2. Restrict his opponent from creating a straight line first.

In 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 or computer) wins or it’s a tie.

|  |  |  |
| --- | --- | --- |
| **1** | **2** | **3** |
| **4** | **5** | **6** |
| **7** | **8** | **9** |

Figure: 2.2

If any player is able to draw three Xs or three Os in the following combinations then that player wins. The combinations are:

a) 1, 2, 3 b) 4, 5, 6

c) 7, 8, 9 d) 1, 4, 7

e) 2, 5, 8 f) 3, 6, 9

h) 1, 5, 9 i) 3, 5, 7

# Core Logic - AI:

There are two core logics in this game – when both players are human, and when one is computer. Suppose the player use X and the computer use O . The logic used for the AI is as follows:

## First move:

* + 1. If the center is free, get the center. (Figure: 3.1)
    2. Otherwise, get any of the corners. (Figure: 3.2)

|  |  |  |
| --- | --- | --- |
| **X** |  |  |
|  | **O** |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **O** |  | **O** |
|  | **X** |  |
|  |  |  |

Figure: 3.1

## Second move:

Figure: 3.2

**O**

(Figure: 3.4)

1. Block user from winning. (Figure: 3.3)
2. Option for winning by applying the following logic:

**O**

If the center is occupied by user, get any of the corners.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X** | **X** |  | | |
|  | **O** |  |
|  | **O** |  |  |
|  | | |
|  |  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **O** |  | **O** | |
|  | **X** |  | |
| **X** |  |  | |
|  | **O** |

Figure: 3.3 Figure: 3.4

Otherwise, the following cases happen:

## Case 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X** |  | | |  |
|  | **O** |  |
| **O** | **O** | | | **O** |
|  | **O** | | | **X** |

Figure: 3.5

If any situation arises like the figure 3.5 then the computer sets its symbol any one of the position among 2, 4, 6 and 8.

## Case 2:

|  |  |  |
| --- | --- | --- |
|  | **X** |  |
| **4** | **O** | **6** |
|  | **X** |  |

|  |  |  |
| --- | --- | --- |
| **X** |  |  |
| **4** | **O** | **6** |
|  | **X** |  |

|  |  |  |
| --- | --- | --- |
|  | **X** |  |
| **4** | **O** | **6** |
|  |  | **X** |

Figure: 3.6 Figure: 3.7 Figure: 3.8

If any situation arises like the figure 3.6 or figure 3.7 or figure 3.8 then the computer sets its symbol at any position among 4 and 6.

|  |  |  |
| --- | --- | --- |
|  | **2** |  |
|  | **O** | **X** |
| **X** | **8** |  |

|  |  |  |
| --- | --- | --- |
|  | **2** |  |
| **X** | **O** | **X** |
|  | **8** |  |

|  |  |  |
| --- | --- | --- |
|  | **2** |  |
| **X** | **O** |  |
|  | **8** | **X** |

## Case 3:

Figure: 3.9 Figure: 3.10 Figure: 3.11

If any situation arises like the figure 3.9 or figure 3.10 or figure 3.11 then the computer sets its symbol at any position among 2 and 8.

## Case 4:

|  |  |  |
| --- | --- | --- |
| **1** | **X** | **3** |
|  | **O** | **X** |
| **7** |  | **9** |

|  |  |  |
| --- | --- | --- |
| **1** | **X** | **3** |
| **X** | **O** |  |
| **7** |  | **9** |

Figure: 3.12 Figure: 3.13

|  |  |  |
| --- | --- | --- |
| **1** |  | **3** |
| **X** | **O** |  |
| **7** | **X** | **9** |

|  |  |  |
| --- | --- | --- |
| **1** |  | **3** |
|  | **O** | **X** |
| **7** | **X** | **9** |

Figure: 3.14 Figure: 3.15

If any situation arises like the figure 3.12 or figure 3.13 or figure 3.14 or

3.15 then the computer sets its symbol at any position among 1, 3, 7 and 9.

## Third and fourth move:

* + 1. Option for winning. (Figure: 3.16)
    2. Block user from winning. (Figure: 3.17)
    3. Randomly play a move. (Figure: 3.18)

|  |  |  |
| --- | --- | --- |
| **O** |  | **X** |
| **X** | **O** |  |
| **X** |  | **O** |

|  |  |  |
| --- | --- | --- |
| **O** |  |  |
| **X** | **O** |  |
| **X** |  | **X** |

Figure: 3.16 Figure: 3.17

**O**

Figure: 3.18

|  |  |  |  |
| --- | --- | --- | --- |
| **X** | **X** | | **O** |
| **O** | **O** | | **X** |
| **X** |  | |  |
| **O** |  |
|  |  | |

# Core Logic - Humans:

For each move, check whether any 3 combination is occupied by any player and display the winner accordingly.

1. **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.

# Reference:

# <https://inventwithpython.com/chapter10.html>

# <https://codereview.stackexchange.com/questions/108738/python-tic-tac-toe-game>

# <https://dev.to/nestedsoftware/tic-tac-toe-with-the-minimax-algorithm-5988>

## Books:

*H.M.Deitel and P.J.Deital, Java How to program: Sixth Edition Herbert Schildt, The Complete Reference: Fifth edition*