**Salim Habib University**

**Department of Computer Science**

**Artificial Intelligence**

**Project Report**

**Tic Tac Toe**

**Course Instructor: Sir Waseem Rauf**

**Submitted by: Hasan Rizvi (F18CSC01)**

**Dated: 23/Dec/2022**

**Introduction**

Tic-tac-toe is a very exciting game for human beings. If you’re an admirer, you’ve probably stimulated from the simple game to some modified like three dimensional tic-tac-toe on a larger grid. If you sit down right now to play normal three-by-three tic-tac-toe with a friend, what will happen is that every game will come out a tie. Both you and your friend can play perfectly, never making an error that would allow your opponent to win. But can you define how you know where to move each turn? Most of the time, you probably aren’t even aware of alternate options; you just stare at the board and suddenly know where you want to move. That kind of immediate knowledge is great for human beings, because it makes you a fast player.

But it isn’t much help in writing a computer program. For that, you have to know very clearly what your strategy is.

**Rules of the game**

The game is to be played between two people (in this program between HUMAN and COMPUTER).

One of the player chooses ‘O’ and the other ‘X’ to mark their individual cells.

The game starts with one of the players and the game ends when one of the players has one whole row/ column/ diagonal filled with his/her respective character (‘O’ or ‘X’).

A player can play perfect tic-tac-toe (win or draw) given they move according to the maximum possible move from the following table.

Win: If the player has two in a row, play the third to get three in a row.

Block: If the opponent has two in a row, play the third to block them.

Fork: Produce a chance where you can win in two ways.

Block opponent's fork:

**How to play?**

Use any method to find who will start first.

Then each player taking turn draw their symbol on a space from that nine possible spaces.

The one who met the winning condition first win. If after nine possible spaces are used, but no one wins, that game is a draw

**Algorithm Details**

Backtracking Minimax is a kind of algorithm that is used in decision making and game theory to find the optimal move for a player, assuming that your opponent also plays optimally. It is widely used in two player turn-based games such as Tic-Tac-Toe, Backgammon, Mancala, Chess, etc.

In Minimax the two players are called maximizer and minimizer. The maximizer tries to get the highest score possible while the minimizer tries to do the opposite and get the lowest score possible.

To check whether or not the current move is better than the best move we take the help of minimax () function which will consider all the possible ways the game can go and returns the best value for that move, assuming the opponent also plays optimally The code for the maximizer and minimizer in the minimax () function is similar to find Best Move (), the only difference is, instead of returning a move, it will return a value.