Artificial Intelligence Mini Project

On

“Tic-tac-toe”

Submitted by:

Mani Garg (102003470)

Himanshu Singla (102003474)

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.

1. 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:

(a) Human (b) Computer

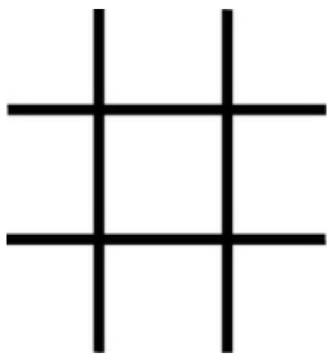

Figure: 2.1

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.

2.2 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:

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

b) 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

2. 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:

3.1 First move:

a) If the center is free, get the center. (Figure: 3.1)
b) Otherwise, get any of the corners. (Figure: 3.2)

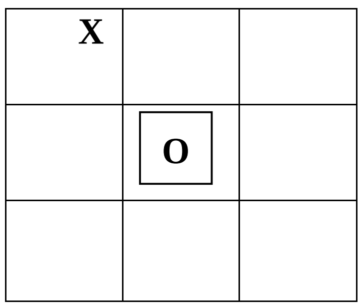

Figure: 3.1

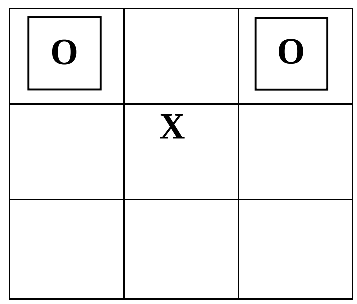

Figure: 3.2

3.2 Second move:

a) Block user from winning. (Figure: 3.3)
b) Option for winning by applying the following logic:
If the center is occupied by user, get any of the corners.
(Figure: 3.4)

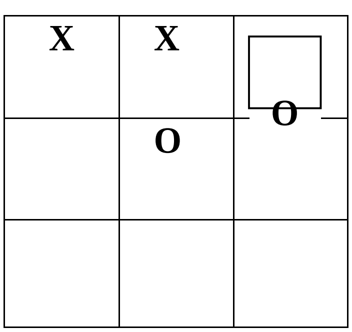

Figure: 3.3

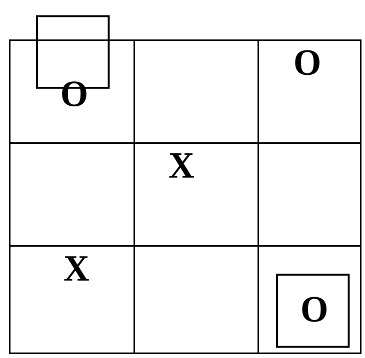

Figure: 3.4

Otherwise, the following cases happen:

Case 1:

X	$\boxed{O}$	
$\boxed{O}$	**O**	$\boxed{O}$
	$\boxed{O}$	**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	

Figure: 3.6

X		
4	**O**	**6**
	X	

Figure: 3.7

	X	
4	**O**	**6**
		X

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.

Case 3:

	2	
	O	X
X	8	

Figure: 3.9

	2	
X	O	X
	8	

Figure: 3.10

	2	
X	O	
	8	X

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

Figure: 3.12

1	X	3
X	O	
7		9

Figure: 3.13

1		3
X	O	
7	X	9

Figure: 3.14

1		3
	O	X
7	X	9

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.

3.3 Third and fourth move:

a) Option for winning. (Figure: 3.16)

b) Block user from winning. (Figure: 3.17)

c) Randomly play a move. (Figure: 3.18)

O		X
X	O	
X		$\boxed{O}$

Figure: 3.16

O		
X	O	
X		X

Figure: 3.17

X	X	O
O	O	X
X	$\boxed{O}$	

$\boxed{O}$

Figure: 3.18

3. Core Logic - Humans:

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

4. ALGORITHMS USED

Minimax Algorithm

Minimax is a recursive algorithm which is used to choose an optimal move for a player assuming that the opponent is also playing optimally. Its objective is to minimize the maximum loss. This algorithm is based on adversarial search technique. In a normal search problem, the optimal solution would be a sequence of actions leading to a goal state. Rather in adversarial search MAX finds the contingent strategy, which specifies the MAX's moves in the initial state, then MAX's moves in the states resulting from every possible response by MIN and continues till the termination condition comes alternately. Furthermore, given a choice, MAX prefers to move to a state of maximum value whereas MIN prefers a state of minimum value.

Alpha-Beta Pruning

The problem with minimax search is that the number of game states it has to examine is exponential in the depth of the tree. The minimax algorithm recursively calls itself until any one of the agent wins or the board is full which takes a lot of computation time and makes it impossible to solve the 4X4 grid usingstandard minimax algorithm.

To solve this issue, we can use alpha-beta pruning algorithm which eliminates large parts of the tree from considerations by pruning the tree. When applied to a standard minimax tree, it returns the same move as minimax would, but it prunes away branches that cannot possibly influence the final decision. Basically, this algorithm applies the principle that there is no use expending search time to find out exactly how bad an alternative is if you have a better alternative. Alpha-beta pruning gets its name from the parameters that bound on the backed-up values that appears anywhere along the path:

α= the value of the best choices (i.e. highest value) so far at any choice point along the path for MAX

β= the value of the best choice (i.e. lowest value) so far at any choice point along the path for MIN

```
Welcome to Tic Tac Toe!

Reference of numbering on the board
1|2|3
-+-+-
4|5|6
-+-+-
7|8|9

Do you want to be 'X' or 'O'?
x
Do you want to go first? (Yes or No)
yes
```

```
The player will go first.
 | |
-+-+-
 | |
-+-+-
 | |
What is your next move? (1-9)
5
O| |
-+-+-
 |X|
-+-+-
 | |
What is your next move? (1-9)
9
O| |O
-+-+-
 |X|
-+-+-
 | |X
What is your next move? (1-9)
2
O|X|O
-+-+-
 |X|
-+-+-
 |O|X
What is your next move? (1-9)
4
O|X|O
-+-+-
X|X|O
-+-+-
 |O|X
What is your next move? (1-9)
7
O|X|O
-+-+-
X|X|O
-+-+-
X|O|X
The game is a tie
Do you want to play again? (Yes or No)
No
```

5. Limitations:

1. GUI is not available. We used Terminal directly.

2. Only keyboard interface is implemented.