PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE

ARTIFICIAL INTELLIGENCE ROBOTICS MINI-PROJECT
REPORT
ON

"TIC TAC TOE GAME USING MINIMAX ALGORITHM"

SUBMITTED BY

Omkar Amilkanthwar	41403
Anirudha Deshmukh	41405
Atharva Deshmukh	41409

Under the guidance of
Prof. Bhumesh Masram

DEPARTMENT OF COMPUTER ENGINEERING
Academic Year 2021-22

Contents

1 Problem Statement

Creating an AI agent in python that plays the classic Tic Tac Toe game to beat a human game.

2 Abstract

In this Project we deduce a mathematical technique to define the winning game Tic-Tac-Toe for an AI. The results were placed in a 3x3 matrix and initial conversions were performed on the rows to find all possible win states. Programming languages were used to find the matrix to determine the diagonal wins. A simulation algorithm is presented to predict the win, or draw of a game by knowing the first move of a player. The game of Tic-Tac-Toe is simulated by using a Min-max algorithm.

3 Harware and Software Requirements

1. Operating system: 64 bit linux operating system- ubuntu or a Windows operating system
2. Ram: 4GB ram is recommended
3. HDD: 50GB hard drive space is recommended
4. Tools/Softwares: Python / Python2 /Python3(recommended)

4 INTRODUCTION

Tic-tac-toe, noughts and crosses, or Xs and Os, is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner. It is a solved game with a forced draw assuming best play from both players.

We will improve the simple computer AI plying at random with an AI agent that takes in case the opponents moves and then play the best possible move to win .We will create an unbeatable AI that plays the classic Tic Tac Toe game. This AI will consider all possible scenarios and make the most optimal move at each step.

In order to solve for the best move the AI makes use of the 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. As its name suggests, its goal is to minimize the maximum loss (minimize the worst case scenario).

MiniMax algorithm is a human representation of predicting what an opponent will play as a consequence of their move.

4.1 Algorithm

A Minimax algorithm can be best defined as a recursive function that does the following things:

1. return a value if a terminal state is found (+1, 0, -1)
2. go through available spots on the board
3. call the minimax function on each available spot (recursion)
4. evaluate returning values from function calls
5. and return the best value

5 OBJECTIVE

- To Build an agent that can play against humans, making the most optimal move at each step by considering all possible predictions of the opponent's moves.
- Learn to implement the MiniMax algorithm in Game Theory to the classic Tic Tac Toe game.

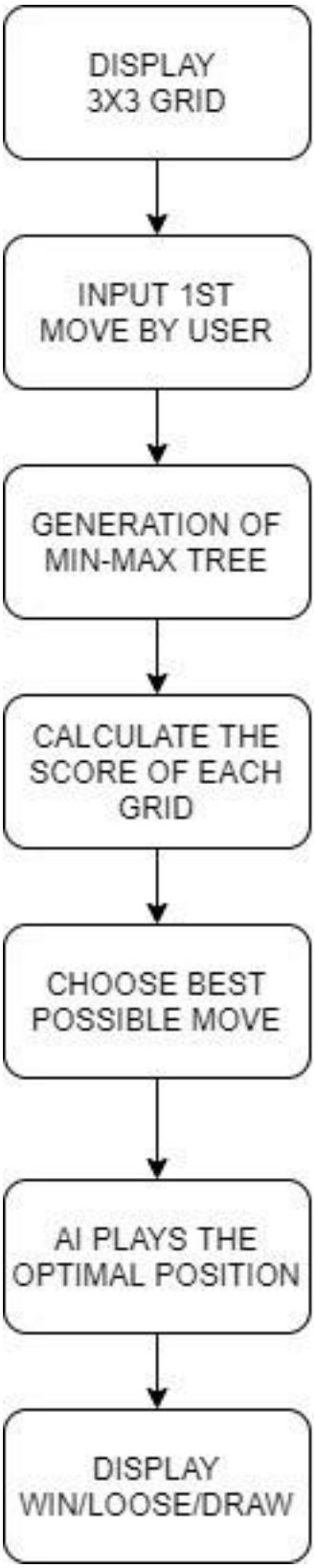
6 Scope

1. For future research study, this game algorithm can be extended to simulate other complicated games like chess and checkers However, in case of chess there are 64 squares with two players. This leads to several possibilities.

2. We can add more to the graphics and add a better Graphical User Interface for the player

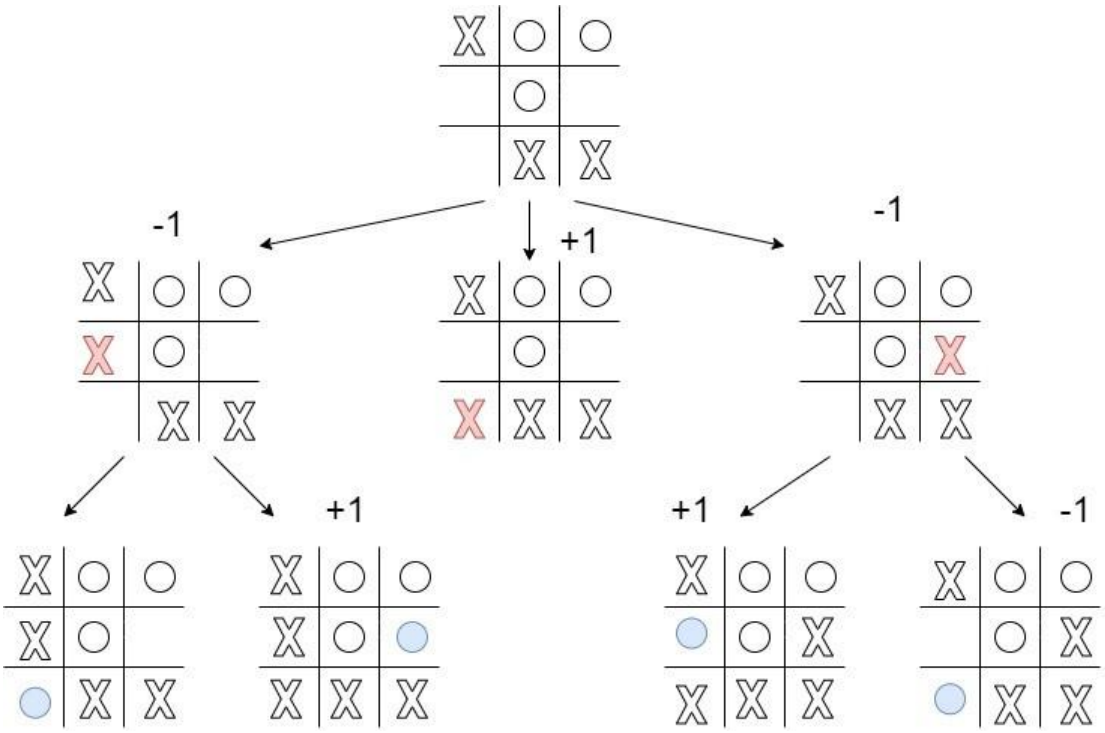
3. Same algorithm can be implemented to a two player Bingo Game to build an AI agent.

7 System Architecture



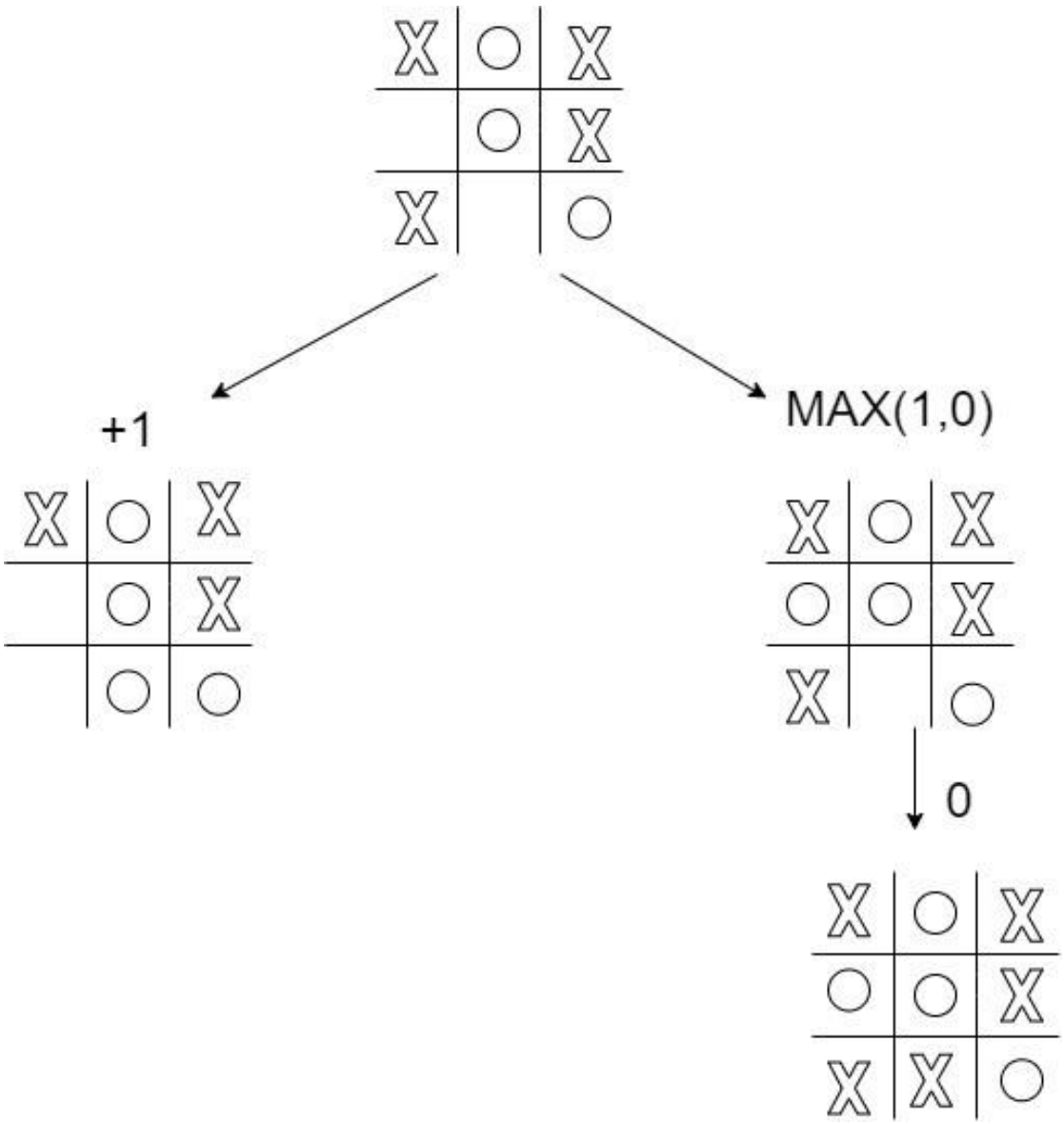
8 Test Cases

8.1 Test Case 1



Final Consequences: X Wins in 2 cases and Y Wins in 2 cases

8.2 Test Case 2



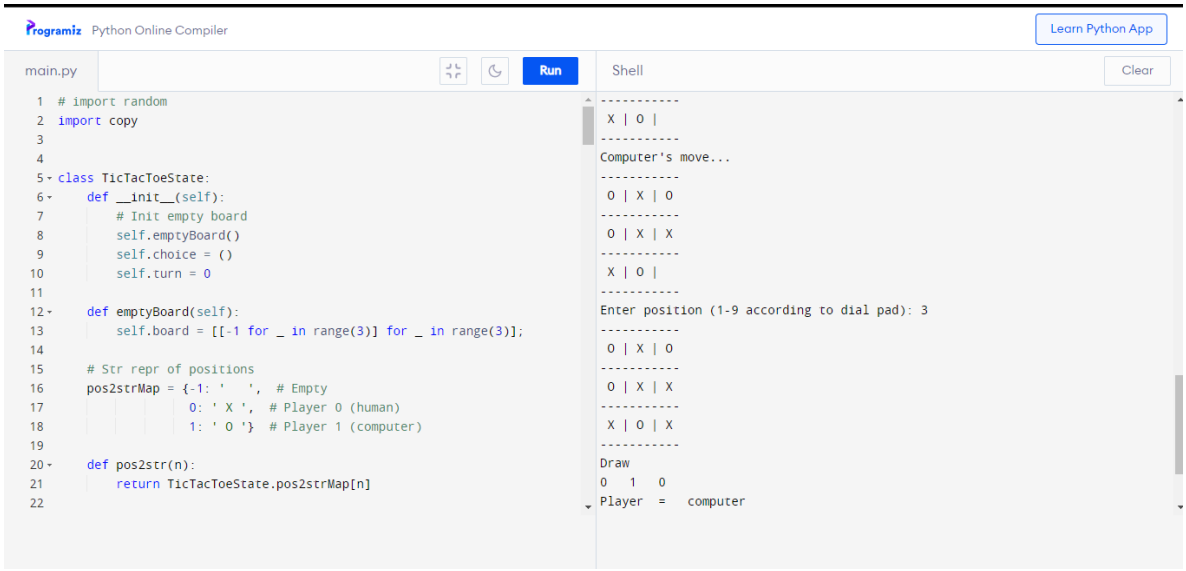
Final Consequences: The game can be won by O or the game will be utmost a draw.

9 Result

In the Tic Tac Toe game there can be a win (+1) lose(-1) or draw(0)
In the first case x can win in two possibilities and y can win in two possibilities
In the second case either the game can be won by O or the game will be utmost a draw.

10 OUTPUT


main.py
Run
Shell
Clear
# import random
import copy
class TicTacToeState:
def __init__(self):
# Init empty board
self.emptyBoard()
self.choice = ()
self.turn = 0
def emptyBoard(self):
self.board = [[-1 for _ in range(3)] for _ in range(3)];
# Str repr of positions
pos2strMap = {-1: ' ', # Empty
0: ' X ', # Player 0 (human)
1: ' O '} # Player 1 (computer)
def pos2str(n):
return TicTacToeState.pos2strMap[n]
Enter position (1-9 according to dial pad): 5
Computer's move...


Programiz Python Online Compiler
Learn Python App
main.py
Run
Shell
Clear
Enter position (1-9 according to dial pad): 3
Draw
0 1 0
Player = computer

11 Conclusion

The Project aims at deducing an intelligent mathematical technique for playing a winning game in Tic-Tac-Toe game. Also, to distort the results into a 3x3 matrix we performed an elementary row operation on the results to establish all possible wins. We depicted the MiniMax algorithm to predict the win cases or draw cases of the Tic-Tac-Toe game the algorithm is implemented in Python using min max algorithm. The concept of graph theory or combinatorial game theory is utilized to implement this game.

This algorithm calculates only one step ahead using a min-max algorithm In an ideal scenario, a player must calculate all the possibilities to ensure the success Tic-Tac-Toe is a small game hence an unbeatable algorithm can be developed because the state space tree generated will be small .

References

[1] Elvis Donkoh, Rebecca Davis, Emmanuel D.J Owusu-Ansah, A. Emmanuel Antwi, Michael Mensah, Application of combinatorial techniques to the Ghanaian Board Game Zaminamina Draft, European Journal of Pure and Applied Mathematics, Vol. 12, No. 1, 2019.

[2] Tac Toe - Creating Unbeatable AI - An Introduction to Minimax Algorithm, Online Available at: https://towardsdatascience.com/tic-tac-toe-creating-unbeatable-ai-with-minimax-algorithm-8af9e52c1e7d