Experiment 05

* Title: Write a program to implement game playing algo for Tic-Tac-Toe using python

* Objectives:
1. To understand mini-max algorithmic approach for Tic-Tac-Toe
2. To implement Tic-Tac-Toe game

* Theory:

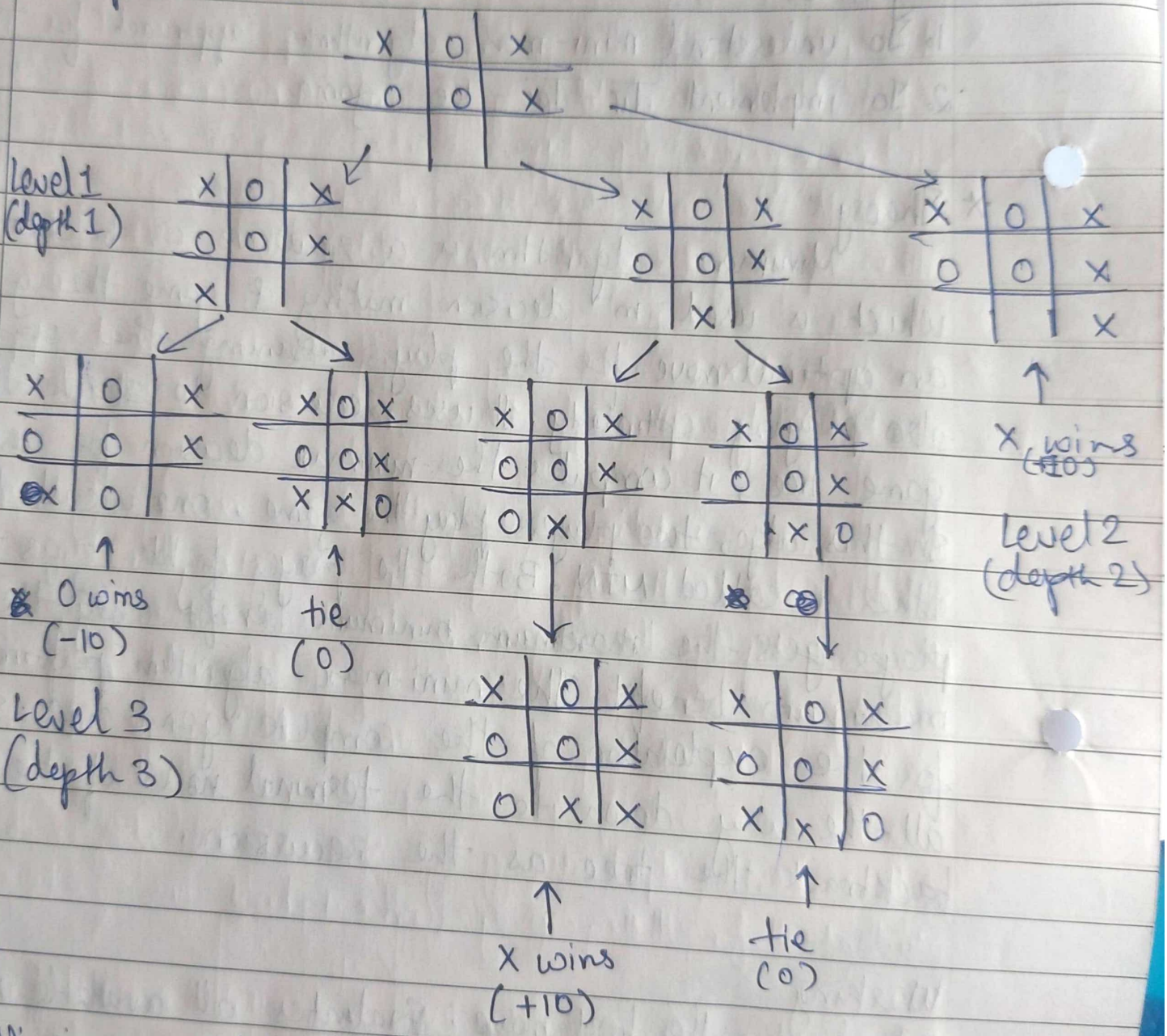
Minimax algorithm is a recursive or backtracking algo which is used in decision making & game theory. It provides an optimal move for the player assuming that opponent is also playing optimally. It uses recursion to search through the game-tree, it computes the minimax decision for current state. In this algo, two players play the game, one is called MAX & other is called MIN. Both the players fight it as the opponent player gets the minimum benefit while they get the maximum benefit. The mini-max algorithm performs a DFS for the exploration of the complete game tree. It proceeds all the way down to the terminal node of the tree, then backtrack the tree as the recursion.

Working
1. Finding the best move: Evaluates all available moves using mini-max & returns best move for maximizes.
2. Mini-max: Checks whether current move is better than best move, assuming opponent plays optimally. Instead of returning a move, it returns a value that describes the current move
3. Checking Game over state: To check whether the game is

over & make sure there are no more moves left

4. Making AI smarter with a heuristic function that can calculate the depth of victory from current state & prefer the least cost path.

Example:



Winning possibilities count = 2 : X(3,3) = +10 - 1 = +9

: X(3,2), O(3,1), X(3,3) = +10 - 3 = +7

Since possibility 1's heuristic cost path is greater, it is achieved faster & game is finished quicker

Pseudo code

```
function findBestMove (board):
    bestmove = NULL
    for each move in board:
        if currentmove is better than bestmove
            bestmove = currentmove
    return bestmove

function mini-max (board, depth, isMaximising Player):
    if current board state is a terminal state
        return value of board

    if isMaximisingPlayer:
        bestVal = -∞
        for each move in board:
            value = minimax(board, depth+1, false)
            bestVal = max( bestVal, value)
        return bestVal
    else
        bestVal = +∞
        for each move in board
            value = minimax (board, depth +1, true)
            bestVal = min ( bestVal, value)
        return bestVal

function isMovesLeft (board):
    for each cell in board
        if current cell is empty
            return true

    return false
```

* Conclusion: Thus, the concept of Mini-Max is understood & implemented in Tic-Tac-Toe game