

[illegible]

Name :- K^untee Omprakash lokiya.

Class :- BE-IT

Roll No :- 32.

Subject: - IS VAB

DOP

DDA

Remark

Sign

* Min Max with Alpha Beta for Tic Tac Toe.

- ii) All nine squares are filled.

→ Programmer created in 'WinningStates' a named set containing a list of all possible win conditions inside "Properties.py", if a player places 'X's or 'O's in any of the list, they are declared winner!

The winning states are:

Winning States = $([0, 1, 2], [3, 4, 5], [6, 7, 8], [0, 3, 6], [1, 4, 7], [2, 5, 8], [0, 4, 8], [2, 4, 6])$.

- Programmer also created a minimax bot which uses Min Max Algorithm with Alpha Beta pruning to decide the best move.

→ The main.py starts by initialization of two objects named of MinMaxBot and DummyBot. The code then creates a variable Judge which called TicTacToeJudge, to which both objects are passed. The TicTacToeJudge.py decides the winner.

→ Programmer also created a helper method, Helper.py which gets the opponent's position to bot and gets the available moves to play. It imports properties.py mentioned earlier.

* Inputs : → No inputs from user.
(As both the bots, DummyBot and MinMaxBot play the game).

* Output : i) Winner Name which can be:
a) Bot One (MinMax Bot)
b) Bot Two (Dummy Bot).
c) Draw (When all positions are filled with no winner).

The winner is decided if the bot's position is in the set of list of WinningStates().

* Analysis of claim by Programmer that it uses MinMax with alpha beta pruning.

- i) This claim comes from the `Bestmove()` method in `MinMaxBot.py` as it uses recursion to find the next best move.
- ii) It starts by getting the `winner()` state and checks if the game already ended by comparing the winner variable with `Self.char`, `Self.Opponent` or Draw state and returns `1, -1, 0` respectively.
- iii) The method then starts a for loop for which iterates through all possible moves in the gameboard.
- iv) After every move, the `Bestmove()` calls itself recursively to figure out next best move by the `MinMaxBot`.
- v) The Bot then places the marker on best move and updates the Alpha, Beta variables.
- vi) The Alpha, Beta variables are checked with Value and are updated accordingly. If Value is greater than Alpha, Alpha is assigned to value & if its lower than Beta, Beta's value is updated to Value.

Thus, the claim by Programmer that it uses MinMax with Alpha Beta Pruning is correct.

Outputs :

i) Bot one

'x'	'o'	'x'
'o'	'o'	None
'x'	'o'	'x'

vi) Bot one

'o'	'x'	'x'
x	o	None
x	o	o

ii) Bot one

'o'	'x'	'x'
'o'	None	None
'o'	None	'x'

vii) Draw

'o'	'x'	'x'
'x'	'o'	'o'
'x'	'o'	'x'

iii) Bot Two

'o'	'x'	None
'o'	'x'	'x'
'o'	'o'	'x'

xiii) Bot one

'o'	'x'	None
None	'x'	'o'
None	'x'	None

iv) Bot Two

'x'	'x'	'x'
'x'	'o'	'o'
'o'	'x'	'o'

ix) Bot Two

'o'	'x'	'x'
'x'	'x'	'o'
'x'	'o'	'o'

v) Bot One

'x'	'o'	None
'o'	'x'	None
'x'	'o'	'x'

x) Draw

'x'	'o'	'x'
'o'	'o'	'x'
'x'	'x'	'o'