

* 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`, `Opponent` or `Draw` state and returns 1, -1, 0 respectively.

iii) The method then starts a for loop which iterates through all possible moves in the game board.

iv) After every move, the `Bestmove()` calls itself recursively to figure out next best move by the `MinMaxBot`.

v) The Bot then places by the `MinMaxBot` Marked 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'

viii) 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'