* Min Max with Alpha Beta For Tic Tac Toe.

1) The goal of Tic-Tac-Toe is to be the first player to get thre in a row on 3x3 grid.
2) 'X' always goes first
3) players alternate playing 'X's and 'O's on board until either:
   i) one player has three in a row horizontally, vertically or digonally
   ii) All nine squares are Filled

4) This A value is associated with each position or state of game.
5) If player A can win in one move, his best move is that winning move.
6) If player B knows that one move will lead to the situation where player A can, at best draw, then player B's best move is the one leading to a draw.
7) player A is trying maximize the chance of A winning.
8) Player B is trying to minimize the chance of A winning
9) program created in 'winningStates' named set containing a list of all possible win conditions inside "propertiespy". If a player place 'X's or 'O's in any of the winning State are:-

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

- It has created a dummy bot which choose position randomly as DummyBot.py
- The GameBoard initialize the Free space to None

⇒ programmer also created a minmaxbot which use MinMax Algorithm with AlphaBeta pruning to decide the best move.

- In main.py start by initialization of two object of minmaxBot and DummyBot
- The code then create a variable judge which called TicTacToeJudge. to which bothe objects are passed the TicTacToeJudge.py decide the winner.
- programmer also created Helper method. Helper.py which get Oppoent's position to bot. and gets the availabe moves to play.

Analysis :-

i) This claim comes from the Bestmove() method in MinMaxBot.py as it uses recursion to find the next be move.
ii) It starts by getting the winner() state and check if the game already end by comparing the winner. Variabe with self.char self.apponent or draw state and return 1, -1, 0 respectively.
iii) The method then starts a for loop which iterates through all possible movies in the gameboard.
iv) After every move, the Bestmove() call itself recursively to figure out next best move by the MinMaxBot.
v) The Bot play the move on best move and update the Alpha, beta Variable.
vi) The alpha beta Vallie check and update accordingly.
vii) If alpha is greater than Alph, Alpha is assgined to value and if it lower than beta, Beta Value is updated.

Thus, the claim by programmer that is uses MinMax with Alpha Beta purning is correct

Inputs :- No inputs from user

Output :- i) Winner Name which can be :-
a) Bot one (MinMax Bot)
b) Bot Two (Dummy Bot.
c) Draw (when all position are Filled with no winner)

The winner is decide if the bot's position is in the set or list winningstates()

~~Outputs :-~~

there are total 6 python files
Dummy.py
Helper.py
Main.py
MinimaxBot.py
Properties.py
TicTacToeJudge.py