class TicTacToe\_board:

def \_init\_(self):

self.board = None

self.mark = None

self.player\_mark = None

def display\_board(self):

print(' | |')

print(' ' + self.board[0] + ' | ' + self.board[1] + ' | ' + self.board[2])

print(' | |')

print('-----------------')

print(' | |')

print(' ' + self.board[3] + ' | ' + self.board[4] + ' | ' + self.board[5])

print(' | |')

print('-----------------')

print(' | |')

print(' ' + self.board[6] + ' | ' + self.board[7] + ' | ' + self.board[8])

print(' | |')

print(' ')

def board\_copy(self, board):

return [x for x in board]

# Check if a person has won

def check\_win(self, player, board): # Player is X or O

## If the player has won then there must be "n" consecutive Player values

# Check Horizontal

board = [board[i:i + 3] for i in range(0, len(board), 3)]

horizontal = [player] \* 3 in board

# Check Vertical

vertical = [player] \* 3 in [list(x) for x in list(zip(\*board))]

# Check Right Diagnol

left = all(board[i][i] == player for i in range(3))

# Left Diagnol

right = all(board[i][2 - i] == player for i in range(3))

return horizontal or vertical or left or right

def check\_draw(self):

return " " not in self.board

def test\_win\_move(self, move, player\_mark, board):

test\_b = self.board\_copy(board)

test\_b[move] = player\_mark

return self.check\_win(player\_mark, test\_b)

def test\_fork\_move(self, move, player\_mark, board):

# Determines if a move opens up a fork

test\_b = self.board\_copy(board)

test\_b[move] = player\_mark

winning\_moves = 0

for i in range(9):

if test\_b[i] and self.test\_win\_move(i, player\_mark, test\_b):

winning\_moves += 1

return winning\_moves >= 2

def final\_stategy(self):

# Play center

if self.board[4] == ' ':

return 4

# Play a corner

for i in [0, 2, 6, 8]:

if self.board[i] == ' ':

return i

# Play a side

for i in [1, 3, 5, 7]:

if self.board[i] == ' ':

return i

def get\_agent\_move(self):

# Check if Agent wins or Sabatoge if Players wins

for i in range(9):

if self.board[i] == ' ':

if self.test\_win\_move(i, self.mark, self.board):

return i

elif self.test\_win\_move(i, self.player\_mark, self.board):

return i

temp = None

count = 0

for i in range(9):

if self.board[i] == ' ':

if self.test\_fork\_move(i, self.mark, self.board):

return i

elif self.test\_fork\_move(i, self.player\_mark, self.board):

temp = i

count += 1

if count == 1:

return temp

elif count == 2:

for i in [1, 3, 5, 7]:

if self.board[i] == ' ':

return i

return self.final\_stategy()

# Player plays

def player\_moves(self):

self.display\_board()

print("Which spot (0-8)")

move = int(input())

while self.board[move] != " ":

self.display\_board()

print("Please pick a valid move (0-8)")

move = int(input())

self.board[move] = self.player\_mark

self.display\_board()

# Agent plays

def agent\_moves(self):

move = self.get\_agent\_move()

self.board[move] = self.mark

self.display\_board()

# Assemble the game

def tictactoe(self):

Playing = True

while Playing:

self.board = [" " for i in range(9)]

# Choose Marks

print('X or O')

self.player\_mark = input()

self.mark = 'X' if self.player\_mark == "O" else "O"

# Choose start

print("Want to go first [y,n]")

flag = '1' if input() == 'y' else '0'

Ingame = True

while Ingame:

if flag == '1':

self.player\_moves()

mark = self.player\_mark

else:

self.agent\_moves()

mark = self.mark

# Check if someone won

if self.check\_win(mark, self.board):

Ingame = False

if mark == self.player\_mark:

print("Player wins!!!")

else:

print("Agent wins")

break

# Check if game Draws

if self.check\_draw():

print('Draw!!!')

break

# Switch to comp or person moves

flag = '0' if flag == '1' else '1'

# Another game

print("Another game [y,n]?")

if input() == 'n':

Playing = False

print('Thank you for playing!')

b = TicTacToe\_board()

b.tictactoe()

X or O

X

Want to go first [y,n]

y

| |

| |

| |

-----------------

| |

| |

| |

-----------------

| |

| |

| |

Which spot (0-8)

0

| |

X | |

| |

-----------------

| |

| |

| |

-----------------

| |

| |

| |

| |

X | |

| |

-----------------

| |

| O |

| |

-----------------

| |

| |

| |

| |

X | |

| |

-----------------

| |

| O |

| |

-----------------

| |

| |

| |

Which spot (0-8)

6

| |

X | |

| |

-----------------

| |

| O |

| |

-----------------

| |

X | |

| |

| |

X | |

| |

-----------------

| |

O | O |

| |

-----------------

| |

X | |

| |

| |

X | |

| |

-----------------

| |

O | O |

| |

-----------------

| |

X | |

| |

Which spot (0-8)

5

| |

X | |

| |

-----------------

| |

O | O | X

| |

-----------------

| |

X | |

| |

| |

X | O |

| |

-----------------

| |

O | O | X

| |

-----------------

| |

X | |

| |

| |

X | O |

| |

-----------------

| |

O | O | X

| |

-----------------

| |

X | |

| |

Which spot (0-8)

7

| |

X | O |

| |

-----------------

| |

O | O | X

| |

-----------------

| |

X | X |

| |

| |

X | O |

| |

-----------------

| |

O | O | X

| |

-----------------

X | X | O

| |

| |

X | O |

| |

-----------------

| |

O | O | X

| |

-----------------

| |

X | X | O

| |

Which spot (0-8)

2

| |

X | O | X

| |

-----------------

| |

O | O | X

| |

-----------------

| |

X | X | O

| |

Draw!!!

Another game [y,n]?