(Part II) Solving Wild Tic-Tac-Toe Using Minimax

BY SIRISH PRABAKAR, DANE JEW

In [1]:
```
import random
import time

current = []


def show(board, player):
    print(player, ":\n")
    for i in range(3):
        for j in range(9):
            if j // 3 == i:
                print(" ", board[j], end="  |")
        print()
        print("------------------")


def check_line(char, pos1, pos2, pos3):
    return pos1 == pos2 == pos3 == char


def check_all(board, char):
    if check_line(char, board[0], board[1], board[2]):
        return True
    elif check_line(char, board[3], board[4], board[5]):
        return True
    elif check_line(char, board[6], board[7], board[8]):
        return True
    elif check_line(char, board[0], board[3], board[6]):
        return True
    elif check_line(char, board[1], board[4], board[7]):
        return True
    elif check_line(char, board[2], board[5], board[8]):
        return True
    elif check_line(char, board[0], board[4], board[8]):
        return True
    elif check_line(char, board[2], board[4], board[6]):
        return True
    else:
        return False


def check_draw(board):
    return set(board) == {"o", "x"}


def check_game_over(board, player):
    if check_all(board, "x") or check_all(board, "o"):
        if player == "P1":
            return -10          # this should be -10 not 10
        elif player == "P2":
            return 10           # this should be 10 not -10
    elif check_draw(board):
        return 0
    else:
        return False


def minimax(board, player):
    global current
    current_score = check_game_over(board, player)

    if current_score is not False:
        return current_score

    scores = []
    moves = []
    p1_win = False
    p2_win = False
    for i in range(9):
    # check all possible moves.
















    if player == "P1":
```

```
    # Find the move with the highest score.   Add that move to current and return that score.







    elif player == "P2":
    # Find the move with the lowest score.   Add that move to current and return that score.
```

Optimal vs Optimal

In [2]:
```
def optimal_vs_optimal():
    global current
    board = [0, 1, 2, 3, 4, 5, 6, 7, 8]
    current = []
    print("Player 1 and Player 2 Both play optimally.\n")
    show(board, "Board")
    curr = ["P1", "P2"]


    #board[0] = 'x'
    #show(board, "P1")
    #print()
    #i = 1


    i = 0
    while True:
        print()
        minimax(board, curr[i])
        show(current[len(current) - 1], curr[i])
        board = current[len(current) - 1]
        print()
        if check_all(board, "x") or check_all(board, "o"):
            print(curr[i] + " Wins!")
            return curr[i]
        elif check_draw(board):
            print("Draw!")
            return "Draw"
        i = (i + 1) % 2

#start_time = time.time()
#optimal_vs_optimal()
#print("\nSeconds Elapsed:", time.time() - start_time)
```

In [3]:
```
optimal_vs_optimal()
```

```
Player 1 and Player 2 Both play optimally.

Board :

  0  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------

P1 :

  0  |  1  |  2  |
------------------
  3  |  x  |  5  |
------------------
  6  |  7  |  8  |
------------------


P2 :

  x  |  1  |  2  |
------------------
  3  |  x  |  5  |
------------------
  6  |  7  |  8  |
------------------


P1 :

  x  |  1  |  2  |
------------------
  3  |  x  |  5  |
------------------
  6  |  7  |  x  |
------------------

P1 Wins!
```

```
Out[3]: 'P1'
```

Random vs Optimal

In [10]:
```
def random_vs_optimal():
    global current
    board = [0, 1, 2, 3, 4, 5, 6, 7, 8]
    current = []
    print("Player 1 plays randomly, and Player 2 plays optimally.\n")
    show(board, "Board")
    print()
    curr = ["P1", "P2"]
    i = 0
    while True:
        print()
        if curr[i] == "P1":
            valid_choice = [i for i in range(9) if board[i] != "x" and board[i] != "o"]
            ran = valid_choice[int(random.random() * len(valid_choice))]
            ran2 = int(random.random() * 2)
            if ran2 == 0:
                board[ran] = "x"
            elif ran2 == 1:
                board[ran] = "o"
            show(board, "P1")
        elif curr[i] == "P2":
            minimax(board, "P2")
            show(current[len(current) - 1], "P2")
            board = current[len(current) - 1]
        print()
        if check_all(board, "x") or check_all(board, "o"):
            print(curr[i] + " Wins!")
            return curr[i]
        elif check_draw(board):
            print("Draw!")
            return "Draw"
        i = (i + 1) % 2


#start_time = time.time()
#random_vs_optimal()
#print("\nSeconds Elapsed:", time.time() - start_time)
```

```
In [12]: random_vs_optimal()
```

```
Player 1 plays randomly, and Player 2 plays optimally.

Board :

  0  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------


P1 :

  o  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------


P2 :

  o  |  x  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------


P1 :

  o  |  x  |  2  |
------------------
  3  |  4  |  5  |
------------------
  o  |  7  |  8  |
------------------


P2 :
  o  |  x  |  2  |
------------------
  o  |  4  |  5  |
------------------
  o  |  7  |  8  |
------------------

P2 Wins!
```

```
Out[12]: 'P2'
```

You vs Optimal

In [13]:
```
def you_vs_optimal():
    global current
    board = [0, 1, 2, 3, 4, 5, 6, 7, 8]
    current = []
    print("You play as Player 1\n")
    show(board, "Board")
    print()
    curr = ["P1", "P2"]
    i = 0
    while True:
        if curr[i] == "P1":
            valid_choice = [str(i) for i in range(9) if i in board]
            while True:
                cell = input("Please enter a valid cell (" + ", ".join(valid_choice) + "): ")
                if cell in valid_choice:
                    break
            while True:
                character = input("Please enter a valid character (x, o): ").lower()
                if character in ["x", "o"]:
                    break
            print()
            board[int(cell)] = character
            show(board, curr[i])
        elif curr[i] == "P2":
            minimax(board, "P2")
            show(current[len(current) - 1], "P2")
            board = current[len(current) - 1]
        print()
        if check_all(board, "x") or check_all(board, "o"):
            print(curr[i] + " Wins!")
            return curr[i]
        if check_draw(board):
            print("Draw!")
            return "Draw"
        i = (i + 1) % 2


#you_vs_optimal()
```