

Tic-Tac-Toe

January 26, 2024

```
import numpy as np
```

using numpy to get more efficient since all lists are only int numbers

```
def ALL_tic_tac_toe(output = False):
    """
    generate all tic-tac-toe games but,
    doesn't care who starts (X or O)
    doesn't care if theres a win, just fills all the nine squares
    impar numbers stand for the starter moves (X or O) and par numbers stand for the other player
    it's justs all the 9! games
    """
    move = 0
    ith_move = np.array([[0,0,0 , 0,0,0 , 0,0,0], [0,0,0 , 0,0,0 , 0,0,0]])

    move = 1
    while move < 9:
        all_games = np.array([0,0,0 , 0,0,0 , 0,0,0])
        list_all_games = [all_games]
        for j in range(1, len(ith_move)):
            for i in range(9): #instead of range(9), range(count(0)) and replace using index of find ?
                if ith_move[j][i] != 0:
                    pass
                else:
                    ith_move[j][i] = move
                    list_all_games.append(ith_move[j].copy())
                    ith_move[j][i] = 0
        ith_move = np.vstack(list_all_games)
        #print("Move " + str(move) + " - OK")
        move += 1

    move = 9
    last_move = np.where(ith_move[1:] == 0, 9, ith_move[1:])
    #print("Move " + str(move) + " - OK")

    np.savetxt("attachment(1)_alltictactoe.csv", last_move, delimiter=",", fmt="%d")

    if output == True:
        return len(last_move), last_move # meter so tipo random lines idk
    else:
        pass
```

```
ALL_tic_tac_toe(output = True)
```

```
## (362880, array([[1, 2, 3, ..., 7, 8, 9],
##           [1, 2, 3, ..., 7, 9, 8],
```

```
##      [1, 2, 3, ..., 8, 7, 9],
##      ...,
##      [9, 7, 8, ..., 3, 2, 1],
##      [8, 9, 7, ..., 3, 2, 1],
##      [9, 8, 7, ..., 3, 2, 1]]))
```

since we now have all the possible games lets see the ones according to the possible win