**A Mini Project Report**

**On**

# TIC TAC TOE game

Submitted to

### InternPe

### 

### PYTHON

### Assignment 4

**Submitted By**

### V. Amrutha - 21691A0504

**Under the Guidance of InternPe**

**PYTHON**



**MADANAPALLE INSTITUTE OF TECHNOLGY & SCIENCE (UGC – AUTONOMOUS)**

**(Affiliated to JNTUA, Anantapuram) Accredited by NBA, Approved by AICTE, New Delhi)**

**AN ISO 9001:2008 Certified Institution**

**P. B. No: 14, Angallu, Madanapalle – 517325 2022-2023**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

This is to certify that the mini project work entitled **“TIC TAC TOE”** is a work carried out by

### V. Amrutha - 21691A0504

And submitted as Assignment 4 for the mini project as a part of Internship with InternPe in the domain PYTHON

Guide

**InternPe**

**PYTHON.**

## LIST OF CONTENTS

|  |  |  |
| --- | --- | --- |
| S.NO | TITLE | PAGE NO |
| 1 | ABSTRACT | 4 |
| 2 | PROCEDURE | 5 |
| 3 | SOURCE CODE | 6-7 |
| 4 | RESULTS | 8 |
| 5 | CONCLUSION | 9 |

# ABSTRACT

**TIC TAC TOE** project is all about a game which is played as a killer of boredom without much effort and noise. It can be played by maximum of two people. The objective of the game is to be the first one to get all the elements of any 1 row or column

get similar. Each of the player will get a chance to place out his coin or symbol he chose at the grid on the board. Here I used random function to give random inputs.

I carried out this project in python language using the packages

numpy and random. The data structure that we used here is an ARRAY (2D array). We used 2D arrays to store the values as per the indices in the from of a grid similar to the structure of matrix.

**PROCEDURE**

 The code starts by importing all the necessary libraries.

 Next, it creates an empty board and checks for empty places on the board.

 The possibilities() function then selects a random place for the player and returns the board.

 The row\_win(), col\_win(), and diag\_win() functions check whether the player has three of their marks in a horizontal row, vertical row, or diagonal row, respectively.

 If so, they return True and win is set to that player.

 If not, they continue checking until either one of these conditions is met.

 Finally, evaluate() determines whether there is a winner or tie based on the results of the other two functions.

 If there is no winner (i.e., all players have zero marks), then no action is taken and the program terminates with an error message stating that there was no game played!

 Otherwise, if both players have at least one mark in each column and row but not in any diagonal line (a situation called a deadlock), then play continues as normal with whoever has more wins being declared the winner.

 In case of a tie, play goes back to evaluating who won last time; this process repeats until somebody wins or somebody loses all their pieces (which ends up being Game Over

 The code creates an empty board and then checks for the player having three marks in a horizontal row, vertical row or diagonal row.

 If the player has achieved this, the code sets the winner variable to be equal to the corresponding value from that row on the board.

 If there is no winner, then all of the players’ pieces are set to 0 and the program ends.

**SOURCE CODE**

import numpy as np

import random

from time import sleep

def create\_board():

return(np.array([[0, 0, 0],

[0, 0, 0],

[0, 0, 0]]))

def possibilities(board):

l = []

for i in range(len(board)):

for j in range(len(board)):

if board[i][j] == 0:

l.append((i, j))

return(l)

def random\_place(board, player):

selection = possibilities(board)

current\_loc = random.choice(selection)

board[current\_loc] = player

return(board)

def row\_win(board, player):

for x in range(len(board)):

win = True

for y in range(len(board)):

if board[x, y] != player:

win = False

continue

if win == True:

return(win)

return(win)

def col\_win(board, player):

for x in range(len(board)):

win = True

for y in range(len(board)):

if board[y][x] != player:

win = False

continue

if win == True:

return(win)

return(win)

def diag\_win(board, player):

win = True

y = 0

for x in range(len(board)):

if board[x, x] != player:

win = False

if win:

return win

win = True

if win:

for x in range(len(board)):

y = len(board) - 1 - x

if board[x, y] != player:

win = False

return win

def evaluate(board):

winner = 0

for player in [1, 2]:

if (row\_win(board, player) or

col\_win(board, player) or

diag\_win(board, player)):

winner = player

if np.all(board != 0) and winner == 0:

winner = -1

return winner

def play\_game():

board, winner, counter = create\_board(), 0, 1

print(board)

sleep(2)

while winner == 0:

for player in [1, 2]:

board = random\_place(board, player)

print("Board after " + str(counter) + " move")

print(board)

sleep(2)

counter += 1

winner = evaluate(board)

if winner != 0:

break

return(winner)

print("Winner is: " + str(play\_game()))

**RESULTS**

[[0 0 0]

[0 0 0]

[0 0 0]]

Board after 1 move

[[1 0 0]

[0 0 0]

[0 0 0]]

Board after 2 move

[[1 0 0]

[2 0 0]

[0 0 0]]

Board after 3 move

[[1 0 0]

[2 0 0]

[1 0 0]]

Board after 4 move

[[1 0 0]

[2 0 0]

[1 2 0]]

Board after 5 move

[[1 1 0]

[2 0 0]

[1 2 0]]

Board after 6 move

[[1 1 0]

[2 2 0]

[1 2 0]]

Board after 7 move

[[1 1 0]

[2 2 1]

[1 2 0]]

Board after 8 move

[[1 1 0]

[2 2 1]

[1 2 2]]

Board after 9 move

[[1 1 1]

[2 2 1]

[1 2 2]]

Winner is: 1

>

**CONCLUSIONS**

I have successfully completed the Mini project which I have selected as my assignment. The output which I received is satisfactory and is acceptable for me in all aspects. I have tested the code with some of my friends asking them to take a game with it. They gave me the positive feedback and they also added saying that the output is very clear.

The code which I developed is based on a simple logic and I made it as simple as I can. I made it in a way which can be easily understood by any person who is aware of some basics in PYTHON programming and having a little exposure towards the Array data structure. Through this project I developed a little knowledge about what a project is and how it is done. On the whole I could definitely say that this **TIC TAC TOE** is a successful execution of the logic as it successfully produced the expected outputs.