# Python implementation of automatic Tic Tac Toe game using random number

Last Updated : 05 Dec, 2022

Tic-tac-toe is a very popular game, so let's implement an automatic Tic-tac-toe game using Python. The game is automatically played by the program and hence, no user input is needed. Still, developing an automatic game will be lots of fun. Let's see how to do this. NumPy and random Python libraries are used to build this game. Instead of asking the user to put a mark on the board, the code randomly chooses a place on the board and put the mark. It will display the board after each turn unless a player wins. If the game gets drawn, then it returns -1.

**Explanation:** play_game() is the main function, which performs the following tasks :

- Calls create_board() to create a 3×3 board and initializes with 0.
- For each player (1 or 2), calls the random_place() function to randomly choose a location on board and mark that location with the player number, alternatively.
- Print the board after each move.
- Evaluate the board after each move to check whether a row or column or diagonal has the same player number. If so, displays the winner's name. If after 9 moves, there is no winner then displays -1.

Below is the code for the above game:

## Python3

```
# Tic-Tac-Toe Program using
# random number in Python

# importing all necessary libraries
import numpy as np
```

```python
import random
from time import sleep

# Creates an empty board


def create_board():
    return(np.array([[0, 0, 0],
                     [0, 0, 0],
                     [0, 0, 0]]))

# Check for empty places on board


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)

# Select a random place for the player


def random_place(board, player):
    selection = possibilities(board)
    current_loc = random.choice(selection)
    board[current_loc] = player
    return(board)

# Checks whether the player has three
# of their marks in a horizontal row


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)
```

```
# Checks whether the player has three
# of their marks in a vertical row


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)

# Checks whether the player has three
# of their marks in a diagonal row


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

# Evaluates whether there is
# a winner or a tie


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

# Main function to start the game


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)


# Driver Code
print("Winner is: " + str(play_game()))
```

## Output:

```
[[0 0 0]
 [0 0 0]
 [0 0 0]]
Board after 1 move
[[0 0 0]
 [0 0 0]
 [1 0 0]]
Board after 2 move
[[0 0 0]
 [0 2 0]
 [1 0 0]]
```

```
Board after 3 move
[[0 1 0]
 [0 2 0]
 [1 0 0]]
Board after 4 move
[[0 1 0]
 [2 2 0]
 [1 0 0]]
Board after 5 move
[[1 1 0]
 [2 2 0]
 [1 0 0]]
Board after 6 move
[[1 1 0]
 [2 2 0]
 [1 2 0]]
Board after 7 move
[[1 1 0]
 [2 2 0]
 [1 2 1]]
Board after 8 move
[[1 1 0]
 [2 2 2]
 [1 2 1]]
Winner is: 2
```

### Code Explanation:

1. The code starts by importing all the necessary libraries.
2. Next, it creates an empty board and checks for empty places on the board.
3. The possibilities() function then selects a random place for the player and returns the board.
4. 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.
5. If so, they return True and win is set to that player.
6. If not, they continue checking until either one of these conditions is met.
7. Finally, evaluate() determines whether there is a winner or tie based on the results of the other two functions.

8. 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!
9. 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.
10. 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
11. The code creates an empty board and then checks for the player having three marks in a horizontal row, vertical row or diagonal row.
12. 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.
13. If there is no winner, then all of the players' pieces are set to 0 and the program ends.

R Richa... 23

Previous Article Next Article

Random Singly Linked List Generator using Python

## Similar Reads

### Tic Tac Toe Game using PyQt5 in Python

In this article , we will see how we can create a Tic Tac Toe game using PyQt5. Tic-tac-toe, noughts, and crosses, or Xs and Os is a paper-and-pencil game for...

5 min read

### Tic Tac Toe game with GUI using tkinter in Python

Tic-tac-toe (American English), noughts and crosses (British English), or Xs and Os is a paper-and-pencil game for two players, X and O, who take turns markin...

15+ min read

### Draw a Tic Tac Toe Board using Python-Turtle

The Task Here is to Make a Tic Tac Toe board layout using Turtle Graphics in Python. For that lets first know what is Turtle Graphics. Turtle graphics In...

2 min read

### Perl | Automatic String to Number Conversion or Casting

Perl has a different way to deal with the operators because here operator defines how the operands will behave but in other programming languages, operands...

4 min read

### Random Walk (Implementation in Python)

Introduction A random walk is a mathematical object, known as a stochastic or random process, that describes a path that consists of a succession of random...

5 min read

### Python Random - random() Function

There are certain situations that involve games or simulations which work on a non-deterministic approach. In these types of situations, random numbers are...

3 min read

### Conway's Game Of Life (Python Implementation)

Conways's Game Of Life is a Cellular Automation Method created by John Conway. This game was created with Biology in mind but has been applied in...

5 min read

**Automatic Birthday mail sending with Python**

Are you bored with sending birthday wishes to your friends or do you forget to send wishes to your friends or do you want to wish them at 12 AM but you...

3 min read

**Making automatic module installer in Python**

Prerequisites: urllibsubprocess Many times the task of installing a module not already available in built-in Python can seem frustrating. This article focuses on...

2 min read

**Automatic News Scraping with Python, Newspaper and Feedparser**

The problem we are trying to solve here is to extract relevant information from news articles, such as the title, author, publish date, and the main content of the...

3 min read

**Article Tags :** Misc Python

**Practice Tags :** Misc python

Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305

### Company

About Us

Legal

In Media

Contact Us

Advertise with us

GFG Corporate Solution

Placement Training Program

GeeksforGeeks Community

### Languages

Python

Java

C++

PHP

GoLang

SQL

R Language

Android Tutorial

Tutorials Archive

### DSA

Data Structures

Algorithms

DSA for Beginners

Basic DSA Problems

DSA Roadmap

Top 100 DSA Interview Problems

DSA Roadmap by Sandeep Jain

All Cheat Sheets

### Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

NLP

Deep Learning

### Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

### Python Tutorial

Python Programming Examples

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Django

### Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

### DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

### System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

### Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

OOAD

System Design Bootcamp

Interview Questions

Puzzles

### School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

### GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects