A

Project on

Tic-Tac-Toe Game using Python Programming

---Submitted By---

Dipak Yashawant Wani

MCS (Ist Year)

---Under Guidance of---

Jyotika Bhati

In Association with Elan & ηvision,

IIT Hyderabad,

Microsoft certified live online training course on

PYTHON

ABSTRACT

Tic-Tac-Toe Game is a very popular game played by two players on the grid of 3 by 3. A special symbol (X or O) is assigned to each participant to indicate that the slot is covered by the respective participant. The winner of the game is the participant who first covers a horizontal, vertical, or diagonal row of the board having only their symbols. Any of the players can play first by their choice. The computation rules ensure selection of the best slot for the computer that will lead to win or prevent the opponent from making a winning move.

Tkinter Python library and in-built random module are used to create the GUI. Two options are available to play the game, along with the system or with another player. A small winning strategy is used to play with the system. The system will try to find the best place to put its naught or cross by which the system will win or try to stop players from winning.

Tic-tac-toe is not a very challenging game for human beings. If you're an enthusiast, you've probably moved from the basic game to some variant like three-dimensional tic-tac-toe on a larger grid. If you sit down right now to play ordinary three-by-three tic-tac-toe with a friend, what will probably happen is that every game will come out a tie.

Both you and your friend can probably play perfectly, never making a mistake that would allow your opponent to win. But can you describe how you know where to move each turn? Most of the time, you probably aren't even aware of alternative possibilities; you just look at the board and instantly know where you want to move. That kind of instant knowledge is great for human beings because it makes you a fast player. But it isn't much help in writing a computer program. For that, you must know very explicitly what your strategy.

CONTENTS

CHAPTER 1

INTRODUCTION

Tic-Tac-Toe Game is a very popular game played by two players on the grid of 3 by 3. When one of the players make a combination of 3 same markers in a horizontal, vertical, or diagonal line the program will display which player has won, whether X or O. Games provide a real source of enjoyment in daily life. Games also are helpful in improving the physical and mental health of human. Apart from daily life physical games, people also play computer games. These games are different than those of physical games in a sense that they do not involve much physical activity rather mental and emotional activities. Getting games to react back to the user of a game has always been long hard question for game programmers. Because let's just face it, a good game that doesn't challenge the user's ability to play the game doesn't keep the user around very long. This idea can be applied to any form of game that is out there. Board games are never fun when the opponent that he or she is playing doesn't learn or catches on. With today's computers always advancing, programmers are always looking for new ways to make a video game more interesting and challenging for the user.

Tic-Tac-Toe game [1] can be played by two players where the square block (3 x 3) can be filled with a cross (X) or a circle (O). The game will toggle between the players by giving the chance for each player to mark their move. When one of the players make a combination of 3 same markers in a horizontal, vertical or diagonal line the program will display which player has won, whether X or O. The Tic-Tac-Toe game is most familiar among all the age groups. The friendliness of Tic-tac-toe games makes them ideal as a pedagogical tool for teaching the concepts of good sportsmanship. The game is a very good brain exercise. It involves looking ahead and trying to figure out what the person playing against you might do next.

CHAPTER 2

OBJECTIVES OF THE PROJECT

The objective of this project is to let the students apply the programming knowledge into a real- world situation/problem and exposed the students how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.
2. Apply object-oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory, and software development.
5. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science

CHAPTER 3

SYSTEM REQUIREMENTS

A. Software Requirements: -

- ➢ Language - Python
- ➢ PyCharm2021.2.3(Community Edition) software
- ➢ Tkinter GUI toolkit

B. Hardware Requirement: -

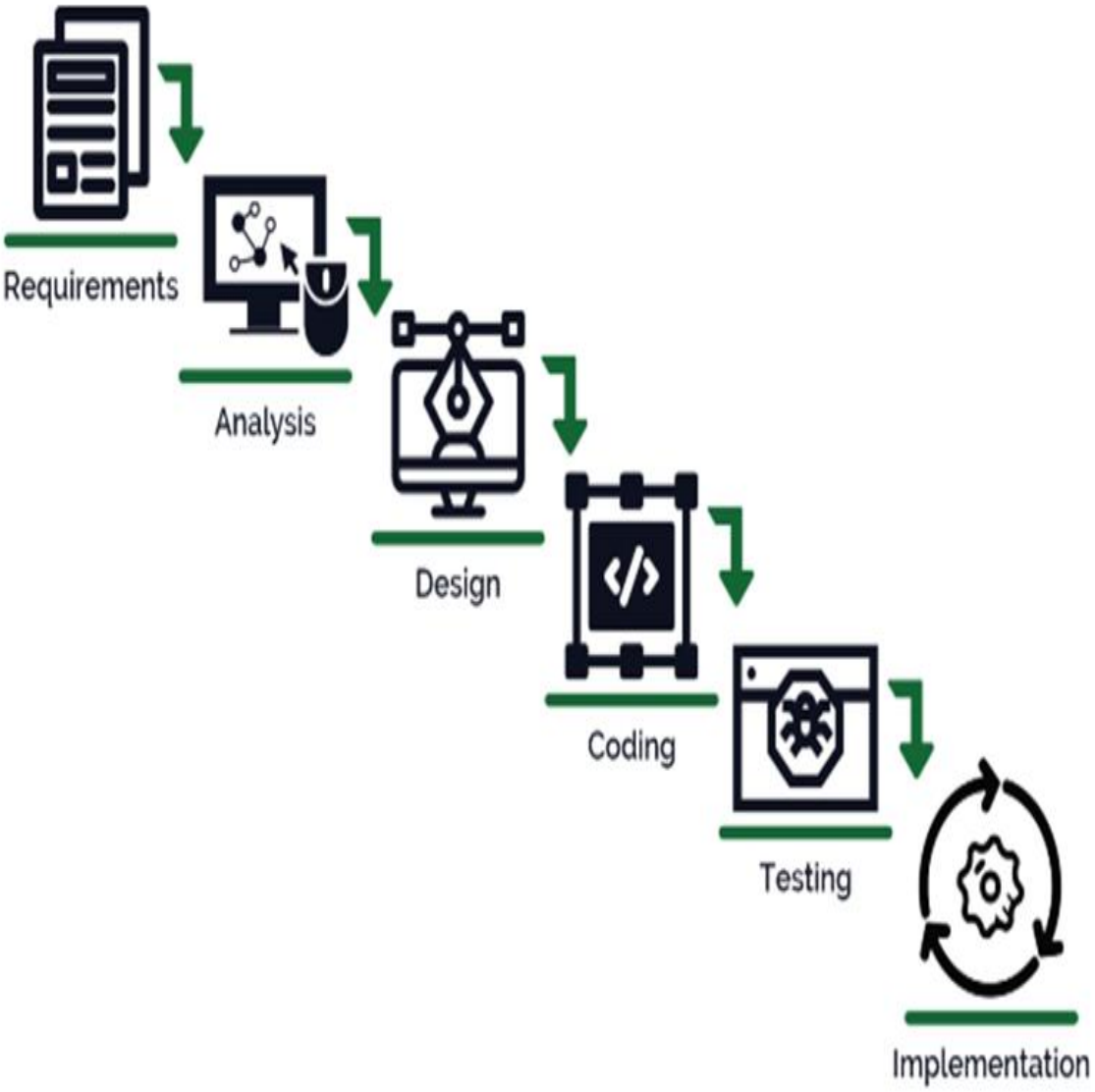
- ➢ Operating system - Window 10
- ➢ Processor - intel core i5
- ➢ Disk space-5 GB
- ➢ RAM- 8 GB

CHAPTER 4

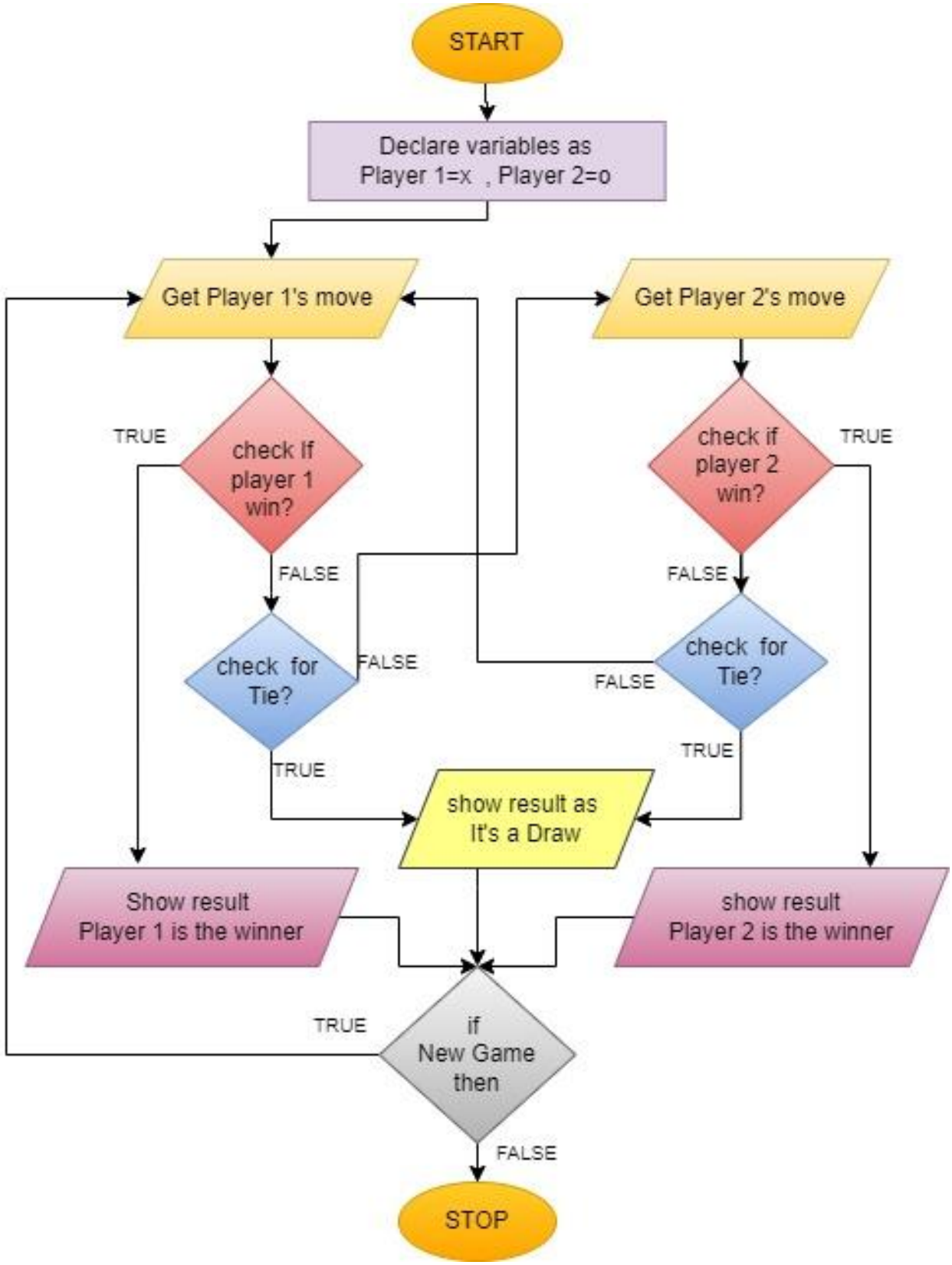
PROJECT LIFE CYCLE

The waterfall model is a classical model used in the software development life cycle to create a system with a linear and sequential approach. It is termed as waterfall because the model develops systematically from one phase to another in downward fashion. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirements. The waterfall approach is the earliest approach that was used for software development.

Software development projects typically include initiation, planning, design, development, testing, implementation, and maintenance phases. However, the phases may be divided differently depending on the organization involved.



➢ Analysis Flowchart: -



CHAPTER 5

WORKING OF PROJECT

The game will toggle between the players by giving the chance for each player to mark their move. When one of the players make a combination of 3 same markers in a horizontal, vertical, or diagonal line the program will display which player has won, whether X or O. Tic-tac-toe (American English), noughts and crosses (British English), or X's and O's is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The winner of the game is the participant who first covers a horizontal, vertical or diagonal row of the board having only their symbols. Any of the players can play first by their choice. The computation rules ensure selection of the best slot for the computer that will lead to win or prevent the opponent from making a winning move.

- **Game Rules: -**

1. Traditionally the first player plays with "X". So you can decide who wants to go with "X" and the computer goes with "O".
2. Only one player can play at a time.
3. If any of the players have filled a square then the other player and the same player cannot override that square.
4. There are only two conditions that match will be draw or may win.
5. The player that succeeds in placing three respective marks (X or O) in a horizontal, vertical, or diagonal row wins the game.

- **Strategy Used in Coding: -**

The highest-priority and the lowest-priority rules seem obvious to me right away. The highest priority is these:1. If I can win on this move, do it.2. If the other player can win on the next move, block that winning square. Here are the lowest-priority rules, used only if there is nothing suggested more strongly by the board position: n-2. Take the center square if it's free n-1. Take a corner square if one is free. Take whatever is available.

The highest priority rules are the ones dealing with the most urgent situations: either I or my opponent can win on the next move. The lowest priority ones deal with the least urgent situations, in which there is nothing special about the moves already made to guide me.

What was harder was to find the rules in between. I knew that the goal of my own tic-tac-toe strategy was to set up a fork, a board position in which I have two winning moves, so my opponent can only block one of them.

CHAPTER 6

IMPLEMENTATION AND RESULT

To implement this game, we will use the basic concepts of Python and Tkinter which is a Python library for building cross-platform GUI. It contains the modules needed for computer graphics. The interesting Tic-Tac-Toe Python project will be built using the tkinter GUI library. We will be explaining all the tkinter object methods that are used in this project. tkinter is a great library that will allow us to create the frames and draw images and shapes on the frame first, we must Create the display window for our game. Then Draw the grid on the canvas where we will play Tic Tac Toe. Then Draw the status bar below the canvas to show which player's turn is it and who wins the game. When someone wins the game, or the game is a draw then we can reset the game.

A. PROJECT CODE: -

```
#Tic-Tac-Toe Game using python code.
from tkinter import Tk,ttk,Button
from tkinter import messagebox
from random import randint

ActivePlayer = 1
p1 = []
p2 = []
mov = 0

def SetLayout(id,player_symbol):
    if id==1:
        b1.config(text= player_symbol)
        b1.state(['disabled'])
    elif id==2:
        b2.config(text= player_symbol)
        b2.state(['disabled'])
    elif id==3:
        b3.config(text= player_symbol)
        b3.state(['disabled'])
    elif id==4:
        b4.config(text= player_symbol)
        b4.state(['disabled'])
    elif id==5:
        b5.config(text= player_symbol)
        b5.state(['disabled'])
    elif id==6:
        b6.config(text= player_symbol)
```

```
            b6.state(['disabled'])
        elif id==7:
            b7.config(text= player_symbol)
            b7.state(['disabled'])
        elif id==8:
            b8.config(text= player_symbol)
            b8.state(['disabled'])
        elif id==9:
            b9.config(text= player_symbol)
            b9.state(['disabled'])

    def CheckWinner():
        global mov
        winner = -1

        if(1 in p1) and (2 in p1) and (3 in p1):
            winner = 1
        if(1 in p2) and (2 in p2) and (3 in p2):
            winner = 2

        if(4 in p1) and (5 in p1) and (6 in p1):
            winner = 1
        if(4 in p2) and (5 in p2) and (6 in p2):
            winner = 2

        if(7 in p1) and (8 in p1) and (9 in p1):
            winner = 1
        if(7 in p2) and (8 in p2) and (9 in p2):
            winner = 2

        if(1 in p1) and (4 in p1) and (7 in p1):
            winner = 1
        if(1 in p2) and (4 in p2) and (7 in p2):
            winner = 2

        if(2 in p1) and (5 in p1) and (8 in p1):
            winner = 1
        if(2 in p2) and (5 in p2) and (8 in p2):
            winner = 2

        if(3 in p1) and (6 in p1) and ( 9 in p1):
            winner = 1
        if(3 in p2) and (6 in p2) and (9 in p2):
            winner = 2

        if(1 in p1) and (5 in p1) and ( 9 in p1):
            winner = 1
        if(1 in p2) and (5 in p2) and (9 in p2):
            winner = 2

        if(3 in p1) and (5 in p1) and ( 7 in p1):
            winner = 1
        if(3 in p2) and (5 in p2) and (7 in p2):
```

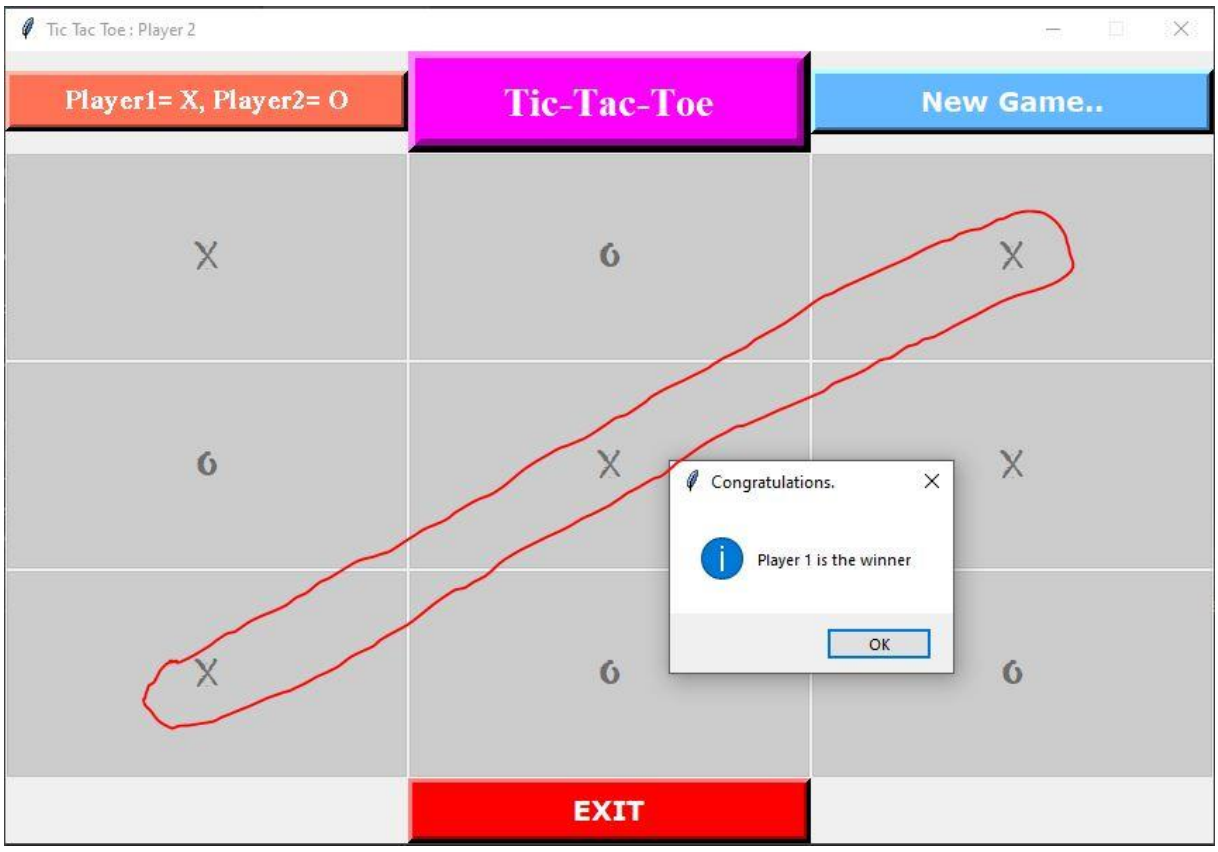
```
            winner = 2

        if winner ==1:
            messagebox.showinfo(title="Congratulations.",
                message="Player 1 is the winner")
        elif winner ==2:
            messagebox.showinfo(title="Congratulations.",
                message="Player 2 is the winner")
        elif mov ==9:
            messagebox.showinfo(title="Match Draw",
                message="It's a Draw!!")

    def ButtonClick(id):
        global ActivePlayer
        global p1,p2
        global mov

        if(ActivePlayer ==1):
            SetLayout(id,"X")
            p1.append(id)
            mov +=1
            root.title("Tic Tac Toe : Player 2")
            ActivePlayer =2

        elif(ActivePlayer==2):
            SetLayout(id,"O")
            p2.append(id)
            mov +=1
            root.title("Tic Tac Toe : Player 1")
            ActivePlayer =1
        CheckWinner()

    def AutoPlay():
        global p1; global p2
        Empty = []
        for cell in range(9):
            if(not((cell +1 in p1) or (cell +1 in p2))):
                Empty.append(cell+1)
        try:
            RandIndex = randint(0,len(Empty) -1)
            ButtonClick(Empty[RandIndex])
        except:
            pass

    def EnableAll():
        b1.config(text= " ")
        b1.state(['!disabled'])
        b2.config(text= " ")
        b2.state(['!disabled'])
        b3.config(text= " ")
        b3.state(['!disabled'])
        b4.config(text= " ")
        b4.state(['!disabled'])
```

```
    b5.config(text= " ")
    b5.state(['!disabled'])
    b6.config(text= " ")
    b6.state(['!disabled'])
    b7.config(text= " ")
    b7.state(['!disabled'])
    b8.config(text= " ")
    b8.state(['!disabled'])
    b9.config(text= " ")
    b9.state(['!disabled'])

def Restart():
    global p1,p2,mov,ActivePlayer
    p1.clear(); p2.clear()
    mov,ActivePlayer = 0,1
    root.title("Tic Tac Toe : Player 1")
    EnableAll()

root = Tk()
root.title("Tic Tac toe : Player 1")
st = ttk.Style()
st.configure("my.TButton", font=('Chiller',24,'bold'))

b1 = ttk.Button(root, text=" ", style="my.TButton")
b1.grid(row=1, column=0, sticky="nwse", ipadx=50,ipady=50)
b1.config(command = lambda : ButtonClick(1))


b2 = ttk.Button(root, text=" ",style ="my.TButton")
b2.grid(row=1, column=1, sticky="snew", ipadx=50, ipady=50)
b2.config(command = lambda : ButtonClick(2))

b3= ttk.Button(root, text=" ",style="my.TButton")
b3.grid(row=1, column=2, sticky="snew", ipadx=50,ipady=50)
b3.config(command = lambda : ButtonClick(3))

b4 = ttk.Button(root, text=" ",style="my.TButton")
b4.grid(row=2, column=0, sticky="snew", ipadx=50,ipady=50)
b4.config(command = lambda : ButtonClick(4))

b5 = ttk.Button(root, text=" ",style="my.TButton")
b5.grid(row=2, column=1, sticky="snew", ipadx=50,ipady=50)
b5.config(command = lambda : ButtonClick(5))

b6 = ttk.Button(root, text=" ",style="my.TButton")
b6.grid(row=2, column=2, sticky="snew", ipadx=50,ipady=50)
b6.config(command = lambda : ButtonClick(6))

b7 = ttk.Button(root, text=" ",style="my.TButton")
b7.grid(row=3, column=0, sticky="snew", ipadx=50,ipady=50)
b7.config(command = lambda : ButtonClick(7))

b8 = ttk.Button(root, text=" ",style="my.TButton")
```

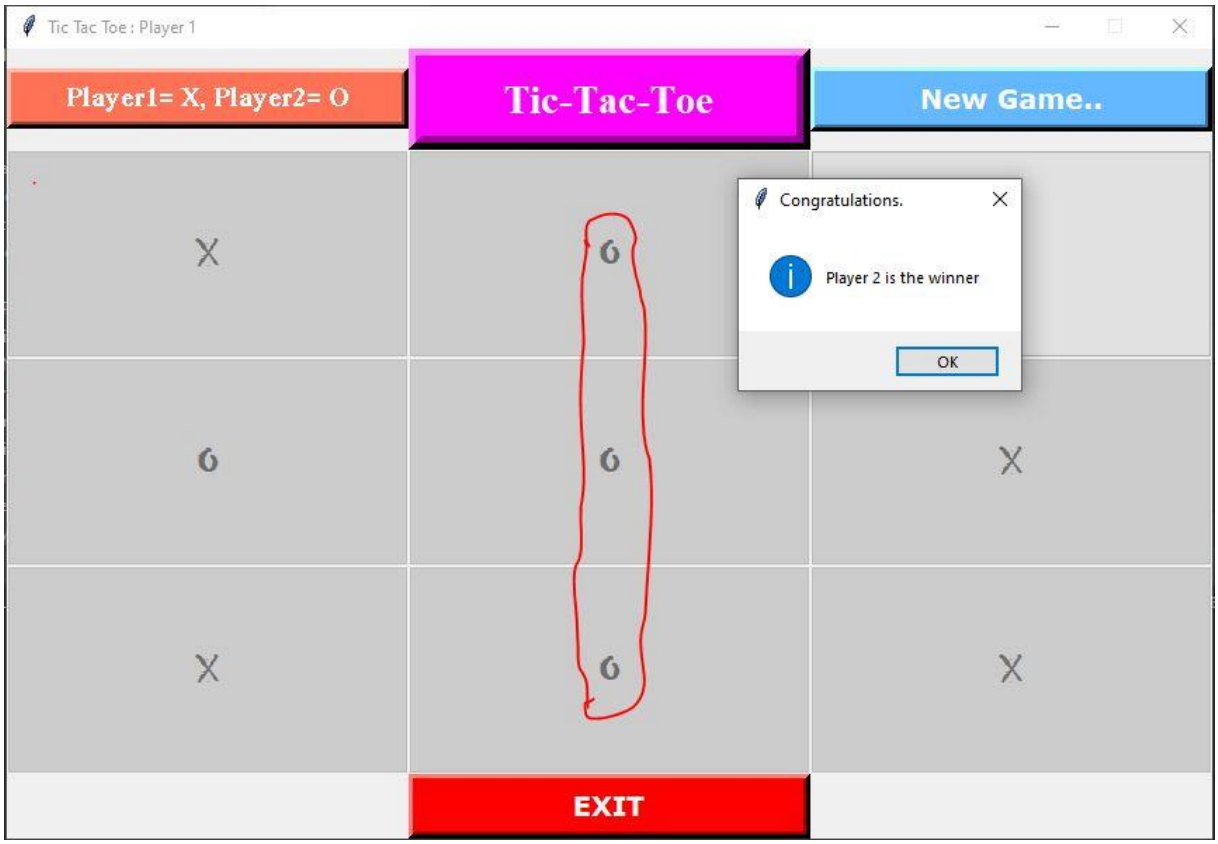
```
b8.grid(row=3, column=1, sticky="snew", ipadx=50,ipady=50)
b8.config(command = lambda : ButtonClick(8))

b9 = ttk.Button(root, text=" ",style="my.TButton")
b9.grid(row=3, column=2, sticky="snew", ipadx=50,ipady=50)
b9.config(command = lambda : ButtonClick(9))

Button(text="Player1= X, Player2= O", font=('MS Serif',
14,'bold'), bg='coral1', fg='white',
            border=5, width=4,).grid(row=0, column=0,
sticky="we")
Button(text="Tic-Tac-Toe", font=('Times new roman', 22,
'bold'), bg='magenta', fg='white',
            border=10, width=5,).grid(row=0, column=1,
sticky="we")
Button(text="New Game..", font=('Verdana', 14,'bold'),
bg='SteelBlue1', fg='white',
            border=5, width=4,command = lambda
:Restart()).grid(row=0, column=2, sticky="we")
Button(text="EXIT", font=('Verdana', 14,'bold'), bg='red',
fg='white',
            border=5, width=4,command = lambda
:quit()).grid(row=4, column=1, sticky="we")
root.resizable(0,0)
root.mainloop()
#End of the code
```

B. RESULT: -

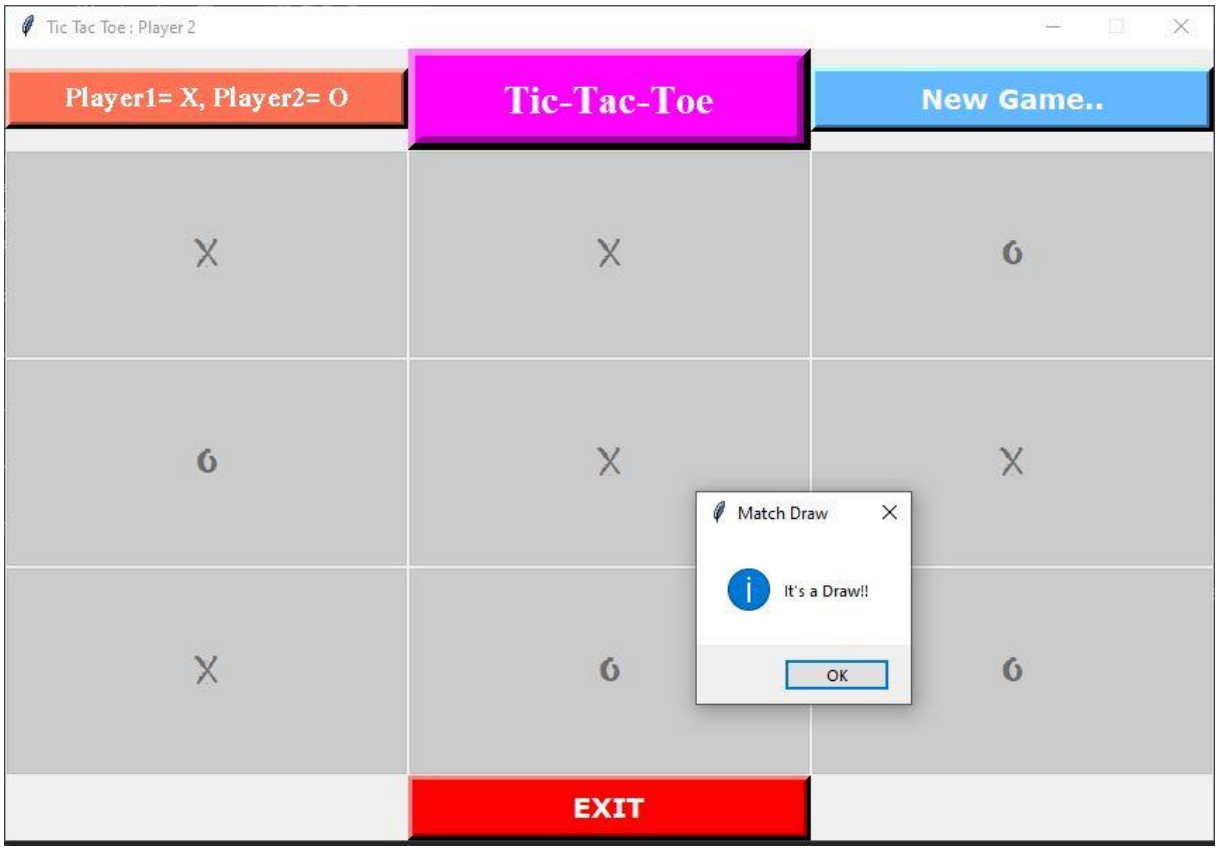
1. If the Player 1 wins, the result be: -



2. If the Player 2 wins, the result be: -



3. If the Neither Player 1 nor Player 2 win, then result be: -



CHAPTER 7

CONCLUSION

With this project in Python, we have successfully made the Tic Tac Toe game. We used the popular tkinter library for rendering graphics on a display window. We learned how to capture events from the keyboard or mouse and trigger a function when the mouse button is pressed. This way we can calculate mouse position, draw X or O on the display and check if the player wins the game or not.

Tic-tac-toe (American English), noughts and crosses (British English), or X's and O's is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The winner of the game is the participant who first covers a horizontal, vertical or diagonal row of the board having only their symbols.

Games provide a real source of enjoyment in daily life. Games also are helpful in improving the physical and mental health of human. The game is a very good brain exercise. I hope you enjoyed building the game.

CHAPTER 7

REFERENCES

1) https://en.wikipedia.org/wiki/Tic-tac-toe
2) https://www.exploratorium.edu/brain_explorer/tictactoe.html
3) https://www.javatpoint.com/tic-tac-toe-in-python
4) www.google.com

Thank you!