Visveswaraya Technological University
Belagavi – 590018, Karnataka

A Mini Project Report on

“Tic-Tac-Toe Game”

Mini Project Report submitted in partial fulfilment of the requirement for Application Development using Python [18CS55]

Bachelor of Engineering
In
Computer Science and Engineering

Submitted By

Shravan K G
1JT19CS086

Rohit Joshi
1JT19CS074

Department of Computer Science and Engineering
Accredited by NBA, New Delhi
Jyothy Institute of Technology,
Tataguni, Bengaluru – 560082

ACKNOWLEDGEMENT

We are very grateful to this esteemed institution **"Jyothy Institute of Technology"** for providing us an opportunity to complete our mini project.

We express our sincere thanks to our **Principal Dr. Gopalakrishna K** for providing us with adequate facilities to undertake this mini project.

We would like to thank **Dr. Prabhanjan S**, **Professor and Head of Computer Science and Engineering Department** for providing for his valuable support.

We would like to thank our guides **Mr. Saravana MK, Assistant Professor** for their keen interest and guidance in preparing this mini project.

Finally, we would thank all our friends who have helped us directly or indirectly in this mini project.

Shravan K G [1JT19CS086]
Rohit Joshi [1JT19CS074]

JYOTHY INSTITUTE OF TECHNOLOGY
Department of Computer Science and Engineering
Accredited by NBA, New Delhi
Jyothy Institute of Technology,
Tataguni, Bengaluru – 560082

CERTIFICATE

Certified that the mini project work entitled **"Tic-Tac-Toe Game"** carried out by **Shravan K G [1JT19CS086] and Rohit Joshi [1JT19CS074]** bonafide students of Jyothy Institute of Technology, in partial fulfilment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** department of the **Visvesvaraya Technological University, Belagavi** during the year **2021-2022**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of Mini project work prescribed for the said degree.

Signature of Guide	Signature of HoD
....	....
Mr. Saravana M. K	**Dr. Prabhanjan. S.**
Assistant Professor	Prof & Head of Dept - CSE
Dept of CSE	JIT, Bengaluru
JIT, Bengaluru	

JYOTHY INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering

Accredited by NBA, New Delhi

Tataguni, Off. Kanakapura Road,

Bengaluru - 560 082

DEPARTMENT VISION AND MISSION

Vison:

To be a centre of excellence in Computer Science and Engineering education, focus on research, innovation and entrepreneurial skill development with professional competency.

Mission:

M1: To provide state of the art ICT infrastructure and innovative, research-oriented teaching-learning environment and motivation for self-learning & problem-solving abilities by recruiting committed faculty.

M2: To encourage Industry Institute Interaction & multi-disciplinary approach for problem-solving and adapt to ever-changing global IT trends.

M3: To imbibe awareness of societal responsibility and leadership qualities with professional competency and ethics.

ABSTRACT

An application built to run and play Games. The game will toggle between the players by giving the chance for each player to mark their move. A Python module Tkinter is used to implement the same. Tkinter is a standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.

Tic-Tac-Toe is one of the paper-and-pencil games. This game requires two players in 3x3 grid with Player 1 acts as “O” and Player 2 acts as “X”, or vice versa. The objective of this game is to take place of three connecting grids in a horizontal, vertical, or diagonal way/fork.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

CHAPTER 1

INTRODUTION

1.1 Introduction to Python

Python is an object-oriented high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python Interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and cam be freely distributed

1.2 Introduction to Tkinter

The Tkinter package the standard Python interface to the TCL/TK GUI toolkit. Both Tk and Tkinter are available on most Unix platforms, including macOS, as well as on Windows systems. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

1.3 Introduction to Tic-Tac-Toe

Tic-Tac-Toe game 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.

1.4 Scope and importance of work

The scope of the project is to create an entertaining game that allows users to play Tic-Tac-Toe using their computer.

CHAPTER 2
IMPLEMENTATION

CHAPTER 2

IMPLEMENTATION

2.1 Modules

I. Tkinter:

Tkinter module is used to create a GUI for the program, allowing user to press buttons to give input.

II. Random:

The random module is used to generate random numbers used in the toss and during batting and bowling.

III. Time:

The Time module is used to suspend execution of the program for a few seconds.

2.2 Functions

I. Turn_choice():

This function toggles between two players X and O

II. Onclick():

This function places the X or O of the player in the table

III. Winner_checker ():

This function checks for all possible ways of winning in a game and shows a pop up when won.

IV. Add_points ():

This function adds 10 points to the player who won in each round 0 if tie.

V. Tie_checker ():

This function checks if the round is tie.

VI. Computer ():

This function is used when player has to play against the computer.

2.3 SOURCE CODE

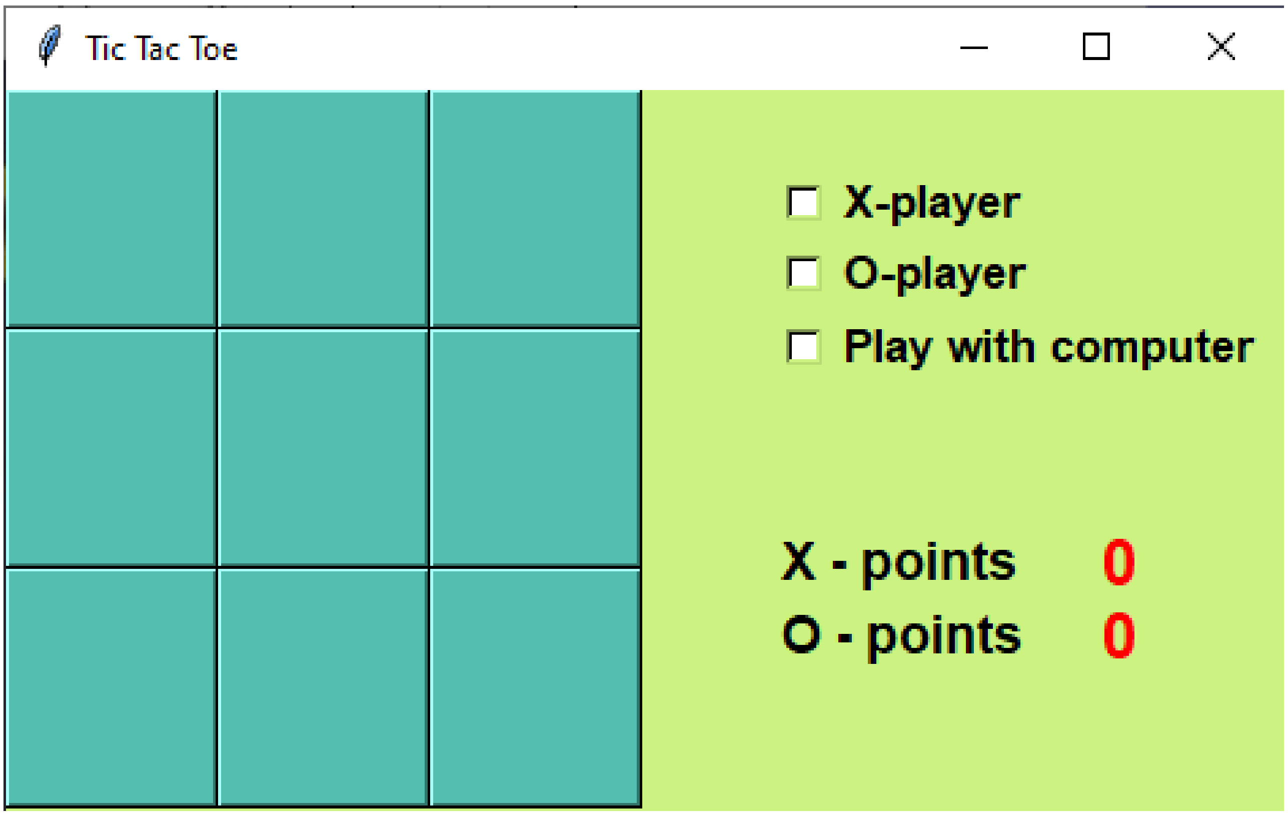
```
from tkinter import *
from tkinter import messagebox
import time, random

class App(Tk):
    is_won = True
    value = []

    def __init__(self):
        super().__init__()
        self.title('Tic Tac Toe')
        self.geometry('470x265')
        self.configure(bg="#CCF381")
        self.turn = ''

        # game frame
        self.game_frame = Frame(self,bg="#CCF381")
        self.game_frame.place(relx=.0, rely=.0, relwidth=.5, relheight=1)

        #button
        self.btn1 = Button(self.game_frame, text='', command=lambda :self.onclick(1),
        height=2, width=4, bg='#54bfb1',font='time 20 bold')
        self.btn1.grid(column=0, row=0)

        self.btn2 = Button(self.game_frame, text='',  command=lambda:self.onclick(2),
        height=2, width=4, bg='#54bfb1',font='time 20 bold')
        self.btn2.grid(column=1, row=0)

        self.btn3 = Button(self.game_frame, text='',  command=lambda
        :self.onclick(3), height=2, width=4, bg='#54bfb1',font='time 20 bold')
        self.btn3.grid(column=2, row=0)

        self.btn4 = Button(self.game_frame, text='',  command=lambda
        :self.onclick(4), height=2, width=4, bg='#54bfb1',font='time 20 bold')
        self.btn4.grid(column=0, row=1)

        self.btn5 = Button(self.game_frame, text='',  command=lambda
        :self.onclick(5), height=2, width=4, bg='#54bfb1',font='time 20 bold')
        self.btn5.grid(column=1, row=1)

        self.btn6 = Button(self.game_frame, text='',  command=lambda
        :self.onclick(6), height=2, width=4, bg='#54bfb1',font='time 20 bold')
        self.btn6.grid(column=2, row=1)

        self.btn7 = Button(self.game_frame, text='',  command=lambda :
```

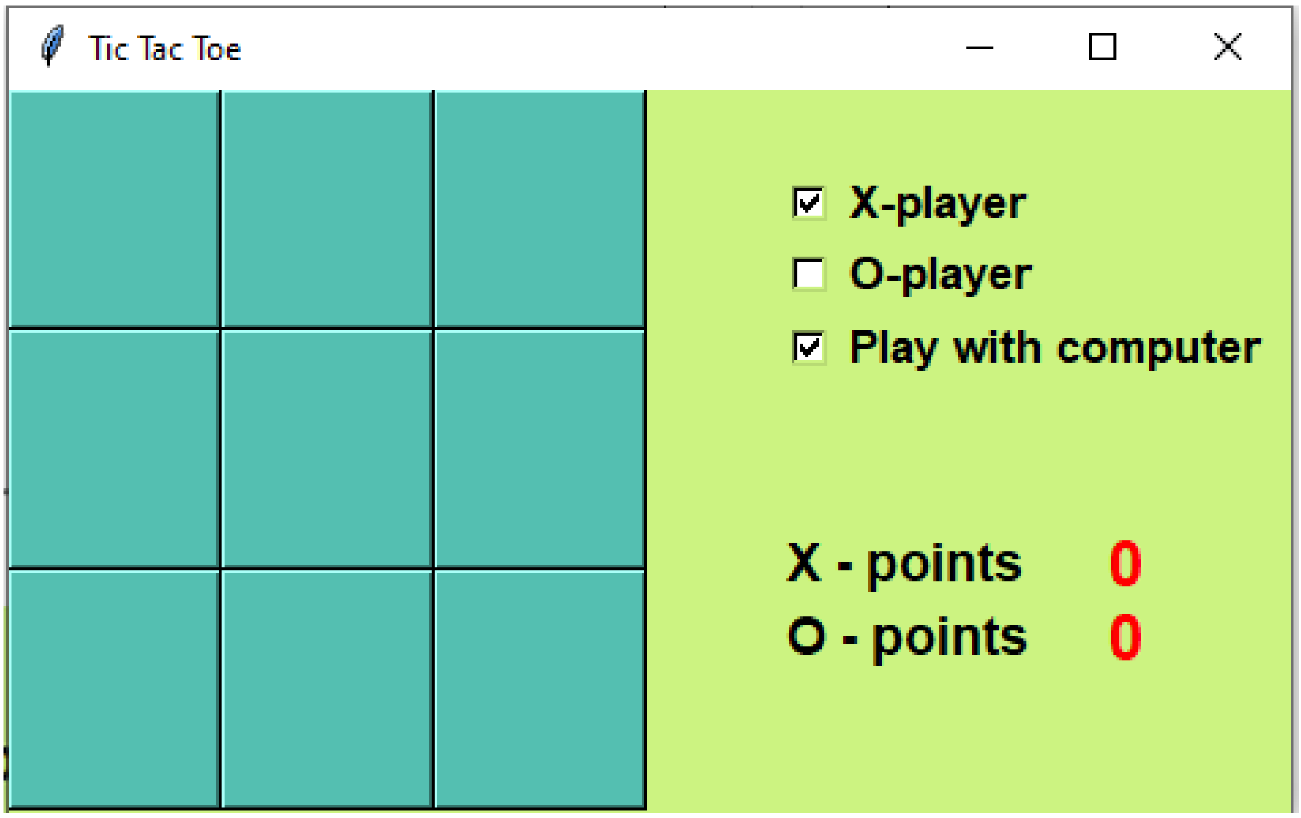
```
self.onclick(7), height=2, width=4,bg='#54bfb1', font='time 20 bold')
self.btn7.grid(column=0, row=2)

self.btn8 = Button(self.game_frame, text='',  command=lambda
:self.onclick(8), height=2, width=4, bg='#54bfb1',font='time 20 bold')
self.btn8.grid(column=1, row=2)

self.btn9 = Button(self.game_frame, text='',  command=lambda
:self.onclick(9), height=2, width=4, bg='#54bfb1',font='time 20 bold')
self.btn9.grid(column=2, row=2)

#output frame
self.output = Frame(self,bg="#CCF381")
self.output.place(relx=.5, rely=.5, relwidth=.5, relheight=1)

#response of turn check box
self.turn_box = Label(self, fg='green', font=20,
justify=CENTER,bg="#CCF381")
self.turn_box.place(relx=.5, rely=.2, relwidth=.4, relheight=.1)

# turn label x
self.check_box1 = IntVar()
self.x_label = Checkbutton(self, variable=self.check_box1, text='X-
player',font='Helvetica 13 bold',bg="#CCF381")
self.x_label.place(relx=.6,rely=.1)
# turn label o
self.check_box2 = IntVar()
self.o_label = Checkbutton(self, variable=self.check_box2, text='O-
player',font='Helvetica 13 bold',bg="#CCF381")
self.o_label.place(relx=.6,rely=.2)

# computer check box
self.computer_box = IntVar()
self.computer_label = Checkbutton(self, variable=self.computer_box,
text='Play with computer',font='Helvetica 13 bold',bg="#CCF381")
self.computer_label.place(relx=.6, rely=.3)

# static name of o and x point
static_x = Label(self, text='X - points =>', font='Helvetica 14 bold', fg='Black',
anchor='nw', justify=CENTER,bg="#CCF381")
static_x.place(relx=.6, rely=.6, relwidth=.2, relheight=.1)
static_o = Label(self, text='O - points =>', fg='Black', font='Helvetica 14 bold',
anchor='nw', justify=CENTER,bg="#CCF381")
static_o.place(relx=.6, rely=.7, relwidth=.2, relheight=.1)

# dynamic x and o points
self.x_points = Label(self, fg='red',text='0', font='Helvetica 18 bold',
anchor='nw', justify=CENTER,bg="#CCF381")
self.x_points.place(relx=.85, rely=.59, relwidth=.1, relheight=.1)

self.o_points = Label(self,text='0', fg='red', font='Helvetica 18 bold',
        anchor='nw', justify=CENTER,bg="#CCF381")
        self.o_points.place(relx=.85, rely=.69, relwidth=.1, relheight=.1)

    def erase_after(self):
        self.btn1['text'] = ''
        self.btn2['text'] = ''
        self.btn3['text'] = ''
        self.btn4['text'] = ''
        self.btn5['text'] = ''
        self.btn6['text'] = ''
        self.btn7['text'] = ''
        self.btn8['text'] = ''
        self.btn9['text'] = ''

    def onclick(self, args):
        def turn_choice():
            if self.computer_box.get() == 0 and (self.check_box1.get() == 1 or
                                                        self.check_box2.get() == 1):
                if self.turn=='X':
                    self.turn='O'
                else: self.turn = 'X'
            else:
                if self.check_box1.get()==1:
                    self.turn='X'
                else:
                    self.turn='O'

        turn_choice()

        if self.check_box2.get() == self.check_box1.get() == self.computer_box.get():
            self.turn_box['text'] = 'Select one player!'
            def rub(text):
                time.sleep(2)
                self.turn_box['text'] = text
            rub("")

        elif self.is_won and (self.check_box2.get()==1 or self.check_box1.get()==1):
            if self.btn1['text'] == '' and args==1: self.btn1['text'] = self.turn
            elif self.btn2['text'] == '' and args==2: self.btn2['text'] = self.turn
            elif self.btn3['text'] == '' and args==3: self.btn3['text'] = self.turn
            elif self.btn4['text'] == '' and args==4: self.btn4['text'] = self.turn
            elif self.btn5['text'] == '' and args==5: self.btn5['text'] = self.turn
            elif self.btn6['text'] == '' and args==6: self.btn6['text'] = self.turn
            elif self.btn7['text'] == '' and args == 7: self.btn7['text'] = self.turn
            elif self.btn8['text'] == '' and args == 8: self.btn8['text'] = self.turn
            elif self.btn9['text'] == '' and args == 9: self.btn9['text'] = self.turn

            # grab all data
            self.value = [self.btn1['text'], self.btn2['text'], self.btn3['text'],
            self.btn4['text'], self.btn5['text'],
            self.btn6['text'], self.btn7['text'], self.btn8['text'], self.btn9['text']]
            self.all_call(self.value)

    def all_call(self, value):
        self.computer(value)
        self.winner_checker(self.turn_box, value)



    def winner_checker(self, turn_box, btn_value):
        check1 = self.btn1['text'] == self.btn2['text'] == self.btn3['text'] != ''
        check2 = self.btn4['text'] == self.btn5['text'] == self.btn6['text'] != ''
        check3 = self.btn7['text'] == self.btn8['text'] == self.btn9['text'] != ''

        check4 = self.btn1['text'] == self.btn4['text'] == self.btn7['text'] != ''
        check5 = self.btn2['text'] == self.btn5['text'] == self.btn8['text'] != ''
        check6 = self.btn3['text'] == self.btn6['text'] == self.btn9['text'] != ''

        check7 = self.btn1['text'] == self.btn5['text'] == self.btn9['text'] != ''
        check8 = self.btn3['text'] == self.btn5['text'] == self.btn7['text'] != ''

        def add_points(value):
            o = self.o_points['text']
            x = self.x_points['text']
            win_x = str(int(x)+10)
            win_o = str(int(o)+10)
            if int(win_x)==30:
                messagebox.showinfo('Tic-Tac-Toe', 'X-Wins!')
                exit(0)
            elif int(win_o)==30:
                messagebox.showinfo('Tic-Tac-Toe', '0-Wins!')
                exit(0)
            if check1:
                if 'O' in [self.btn1['text'] , self.btn2['text'] ,self.btn3['text']]:
                    self.o_points['text'] = win_o
                else: self.x_points['text'] = win_x
            elif check2:
                if 'O' in [self.btn4['text'], self.btn5['text'], self.btn6['text']]:
                    self.o_points['text'] = win_o
                else: self.x_points['text'] = win_x
            elif check3:
                if 'O' in [self.btn7['text'], self.btn8['text'], self.btn9['text']]:
                    self.o_points['text'] = win_o
                else: self.x_points['text'] = win_x
            elif check4:
                if 'O' in [self.btn1['text'], self.btn4['text'], self.btn7['text']]:
                    self.o_points['text'] = win_o
                else: self.x_points['text'] = win_x
            elif check5:
                if 'O' in [self.btn2['text'], self.btn5['text'], self.btn8['text']]:
                    self.o_points['text'] = win_o
                else: self.x_points['text'] = win_x
            elif check6:
                if 'O' in [self.btn3['text'], self.btn6['text'], self.btn9['text']]:
                    self.o_points['text'] = win_o
                else: self.x_points['text'] = win_x
            elif check7:
                if 'O' in [self.btn1['text'], self.btn5['text'], self.btn9['text']]:
                    self.o_points['text'] = win_o
                else: self.x_points['text'] = win_x
            elif check8:
                if 'O' in [self.btn3['text'], self.btn5['text'], self.btn7['text']]:
                    self.o_points['text'] = win_o
                else: self.x_points['text'] = win_x




        if self.is_won:

            if check1:
                add_points(btn_value)
                messagebox.showinfo('Tic-Tac-Toe', 'Wins!')

                self.erase_after()
            elif check2:
                add_points(btn_value)
                messagebox.showinfo('Tic-Tac-Toe', 'Wins!')

                self.erase_after()
            elif check3:
                add_points(btn_value)
                messagebox.showinfo('Tic-Tac-Toe', 'Wins!')

                self.erase_after()
            elif check4:
                add_points(btn_value)
                messagebox.showinfo('Tic-Tac-Toe', 'Wins!')

                self.erase_after()
            elif check5:
                add_points(btn_value)
                messagebox.showinfo('Tic-Tac-Toe', 'Wins!')

                self.erase_after()
            elif check6:
                add_points(btn_value)
                messagebox.showinfo('Tic-Tac-Toe', 'Wins!')

                self.erase_after()
            elif check7:
                add_points(btn_value)
                messagebox.showinfo('Tic-Tac-Toe', 'Wins!')

                self.erase_after()
             elif check8:
                add_points(btn_value)
                messagebox.showinfo('Tic-Tac-Toe', 'Wins!')

                self.erase_after()

          self.tie_checker(self.value)

       def tie_checker(self, check):
          if self.is_won:
             if ('X' in check) and ('O' in check) and ('' not in check):
                messagebox.showinfo('Tic-Tac-Toe', 'Tie!')
                self.erase_after()

       def computer(self, value):
          if self.computer_box.get()==1 and
(self.check_box2.get()!=self.check_box1.get()):
         comp_turn = ''
         if True:
            if self.turn=='X': comp_turn = 'O'
            else: comp_turn = 'X'

         if (value[0] == value[1] != '') and value[2] == '': self.btn3['text'] = comp_turn
         elif (value[1] == value[2] != '') and value[0] == '': self.btn1['text'] =
         comp_turn
         elif (value[0] == value[2] != '') and value[1] == '': self.btn2['text'] =
         comp_turn
         elif (value[3] == value[4] != '') and value[5] == '': self.btn6['text'] =
         comp_turn
         elif (value[4] == value[5] != '') and value[3] == '': self.btn4['text'] =
         comp_turn
         elif (value[3] == value[5] != '') and value[4] == '': self.btn5['text'] =
         comp_turn

         elif (value[5] == value[7] != '') and value[8] == '': self.btn9['text'] =
         comp_turn
         elif (value[7] == value[8] != '') and value[6] == '': self.btn7['text'] =
         comp_turn
         elif (value[5] == value[8] != '') and value[7] == '': self.btn8['text'] =
         comp_turn

         elif (value[0] == value[3] != '') and value[6] == '': self.btn7['text'] =
         comp_turn
         elif (value[3] == value[6] != '') and value[0] == '': self.btn1['text'] =
         comp_turn
         elif (value[0] == value[6] != '') and value[3] == '': self.btn4['text'] =
         comp_turn
```

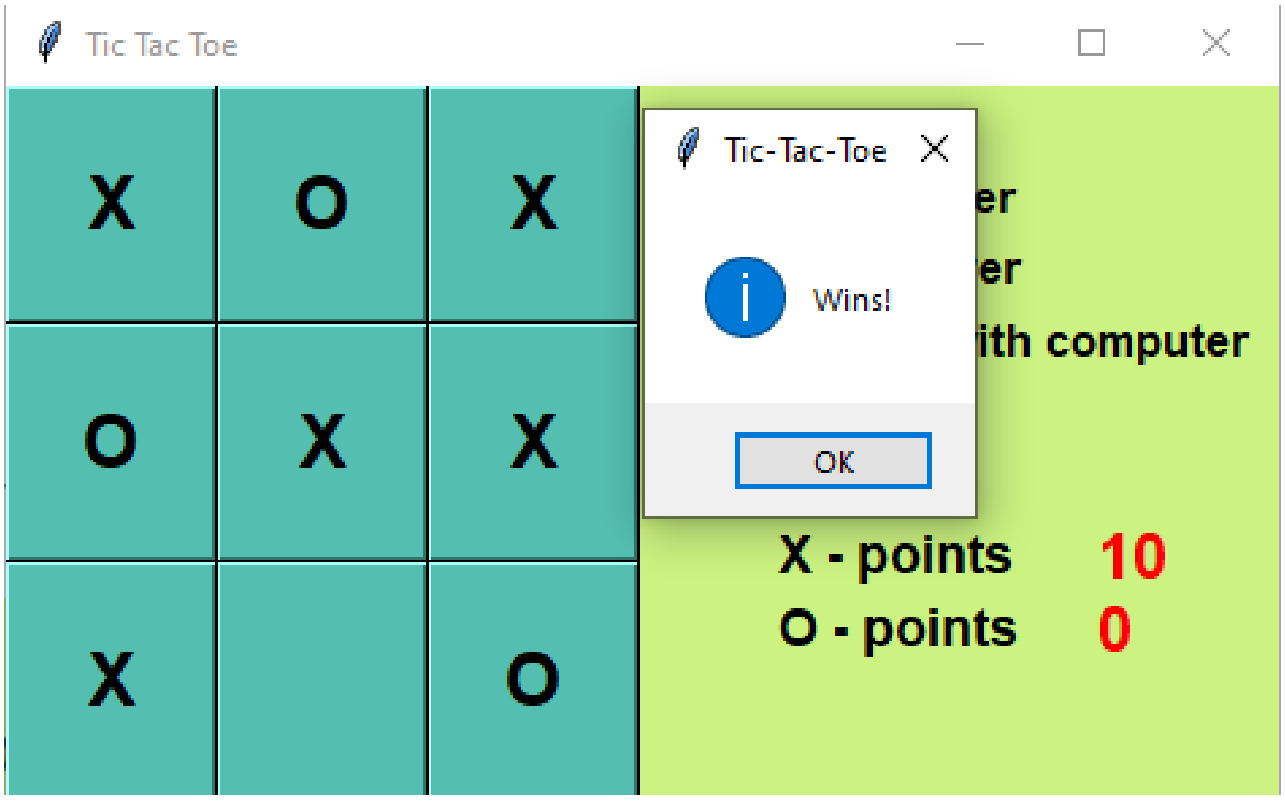
```
elif (value[1] == value[4] != '') and value[7] == '': self.btn8['text'] =
comp_turn
elif (value[4] == value[7] != '') and value[1] == '': self.btn2['text'] =
comp_turn
elif (value[1] == value[7] != '') and value[4] == '': self.btn5['text'] =
comp_turn

elif (value[2] == value[5] != '') and value[8] == '': self.btn9['text'] =
comp_turn
elif (value[5] == value[8] != '') and value[2] == '': self.btn3['text'] =
comp_turn
elif (value[2] == value[8] != '') and value[5] == '': self.btn6['text'] =
comp_turn

elif (value[0] == value[4] != '') and value[8] == '': self.btn9['text'] =
comp_turn
elif (value[4] == value[8] != '') and value[0] == '': self.btn1['text'] =
comp_turn
elif (value[0] == value[8] != '') and value[4] == '': self.btn5['text'] =
comp_turn

elif (value[2] == value[4] != '') and value[6] == '': self.btn7['text'] =
comp_turn
elif (value[4] == value[6] != '') and value[2] == '': self.btn3['text'] =
comp_turn
elif (value[2] == value[6] != '') and value[4] == '': self.btn5['text'] =
comp_turn

else:
    try:
        index =  value.index('X')
    except ValueError:
        index = value.index('O')

    if index == 0:
        select = random.choice([2, 4, 3])
        if select == 2 and value[2] == '':
            self.btn3['text'] = comp_turn
        elif select == 4 and value[4] == '':
            self.btn5['text'] = comp_turn
        elif select==3 and value[3]=='':
            self.btn4['text'] = comp_turn
    elif index == 1:
        select = random.choice([6, 4, 8])
        if select == 6 and value[6]=='':
            self.btn7['text'] = comp_turn
        elif select == 4 and value[4]=='':
            self.btn5['text'] = comp_turn
        elif select==8 and value[8]=='':
            self.btn9['text'] = comp_turn
    elif index == 2:
        select = random.choice([8, 4, 6])
        if select == 8 and value[8]=='':
            self.btn8['text'] = comp_turn
        elif select == 4 and value[4]=='':
            self.btn5['text'] = comp_turn
        elif select==6 and value[6]=='':
            self.btn7['text'] = comp_turn
    elif index == 3:
        select = random.choice([6, 4, 2])
        if select == 6 and value[6]=='':
            self.btn7['text'] = comp_turn
        elif select == 4 and value[4]=='':
            self.btn5['text'] = comp_turn
        elif select==2 and value[2]=='':
            self.btn3['text'] = comp_turn
    elif index == 4:
        select = random.choice([3, 5, 7])
        if select == 3 and value[3]=='':
            self.btn4['text'] = comp_turn
        elif select == 5 and value[5]=='':
            self.btn6['text'] = comp_turn
        elif select==7 and value[7]=='':
            self.btn8['text'] = comp_turn
    elif index == 5:
        select = random.choice([6, 7, 2])
        if select == 6 and value[6]=='':
            self.btn7['text'] = comp_turn
        elif select == 7 and value[7]=='':
            self.btn5['text'] = comp_turn
        elif select==2 and value[2]=='':
            self.btn3['text'] = comp_turn
    elif index == 6:
        select = random.choice([0, 4, 6])
        if select == 0 and value[0]=='':
            self.btn1['text'] = comp_turn
        elif select == 4 and value[4]=='':
            self.btn5['text'] = comp_turn
        elif select==8 and value[8]=='':
            self.btn9['text'] = comp_turn
    elif index == 7:
        select = random.choice([0, 2, 4])
        if select == 0 and value[0]=='':
            self.btn1['text'] = comp_turn
        elif select == 2 and value[2]=='':
            self.btn3['text'] = comp_turn
        elif select==4 and value[4]=='':
            self.btn5['text'] = comp_turn
    elif index == 8:
        select = random.choice([3, 4, 5])
        if select == 3 and value[3]=='':
            self.btn4['text'] = comp_turn
        elif select == 4 and value[4]=='':
            self.btn5['text'] = comp_turn
          elif select==5 and value[5]=='':
            self.btn6['text'] = comp_turn




if __name__ == '__main__':
    app = App()
    app.mainloop()
```

CHAPTER 3
RESULT AND SNAPSHOTS

CHAPTER 3

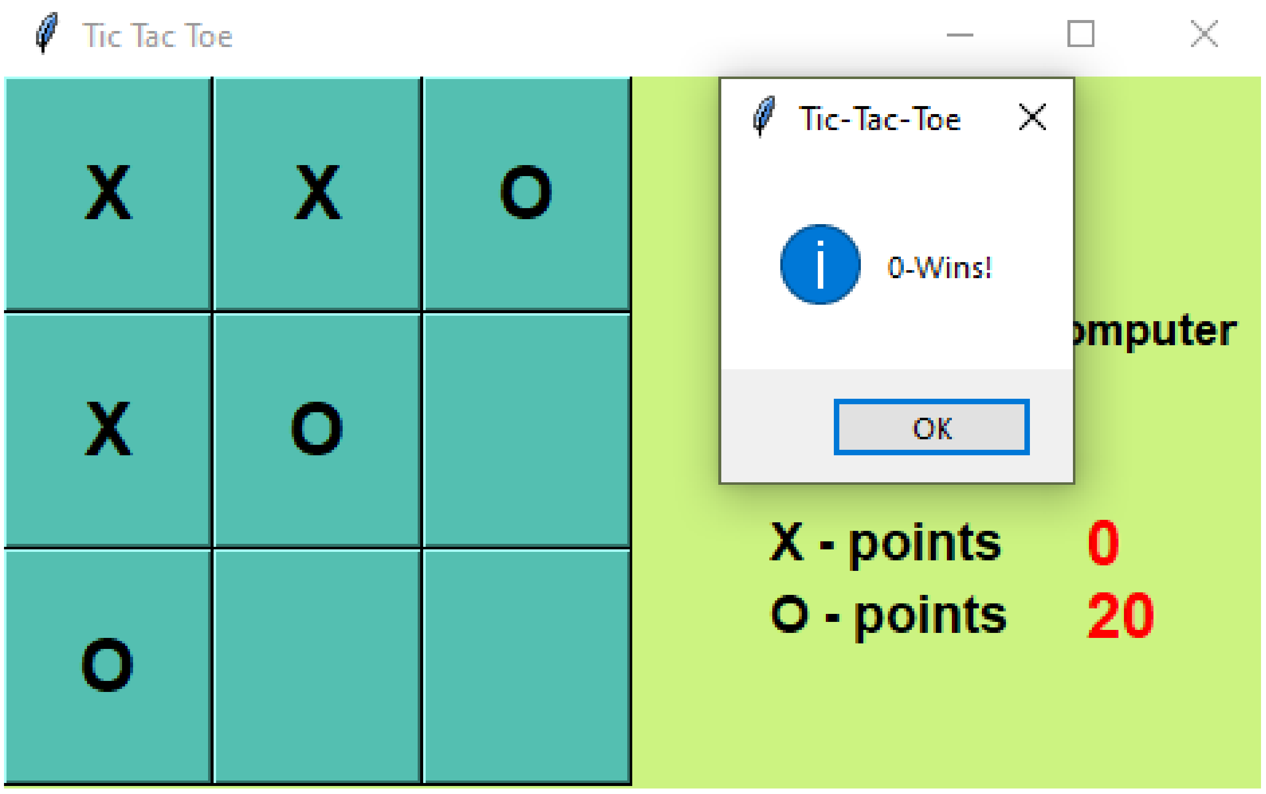

Tic Tac Toe
X-player
O-player
Play with computer
X - points 0
O - points 0


Tic Tac Toe
☑ X-player
☐ O-player
☑ Play with computer
X - points 0
O - points 0



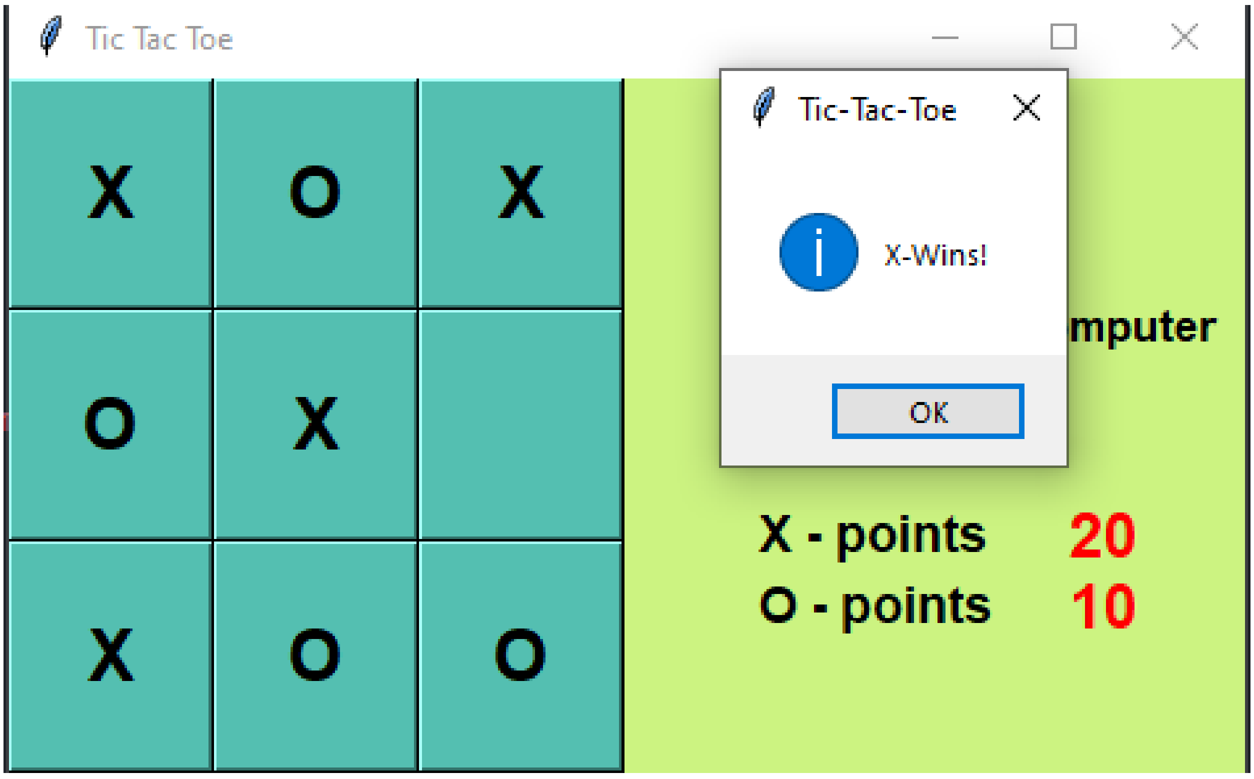
Tic-Tac-Toe

Wins!



Tic-Tac-Toe

0-Wins!



X-Wins!

X - points 20
O - points 10

CHAPTER 4
CONCLUSION

CHAPTER 4
CONCLUSION

We have successfully implemented the Tic-Tac-Toe game in python and demonstrated the functionality of the Tkinter module in a neat and presentable GUI. Buttons are used to take input from the user instead of manually typing their choices. Using random function ensures that no game will be the same.

Features:
1.Interactive GUI program
2.Fun and entertaining game built with Python
3.Behind the scenes randomization and widget placement

REFERENCES

[1] https://www.python.org/doc/

[2] https://docs.python.org/3/library/tkinter.html

[3] https://docs.python.org/3/library/random.html

[4] https://stackoverflow.com/

[5] https://www.tutorialspoint.com/python/