## Visvesvaraya Technological University Belagavi-590 018, Karnataka



A Mini Project Report on

#### “GAMES FOR YOU”

##### Mini Project Report submitted in partial fulfilment of the requirement for the

**Application Programming Using Python with Mini Project [18CS55]**

**Bachelor of Engineering**

**in**

**Computer Science and Engineering**

**Submitted by Shravan K G[1JT19CS086] Rohit Joshi [1JT19CS074]**



## Department of Computer Science and Engineering Jyothy Institute of Technology Tataguni, Bengaluru-560082

**Jyothy Institute of Technology**

### Tataguni, Bengaluru-560082 Department of Computer Science and Engineering



**CERTIFICATE**

Certified that the mini project work entitled **“Tic-Tac-Toe ”** 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 project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

**Mr. SARAVANA M K Dr. Prabhanjan S**

Guide, Asst. Professor Professor & HOD

Dept. of CSE Dept. of CSE

External Viva Examiner Signature with Date : 1.

2.

## ACKNOWLEDGEMENT

Firstly, we are very grateful to this esteemed institution **“Jyothy Institute of Technology**” for providing us an opportunity to complete our project.

We express our sincere thanks to our **Principal Dr. Gopalakrishna K** for providing us with adequate facilities to undertake this 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 M K, Assistant Professor** for their keeninterest and guidance in preparing this work.

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

**Shravan K G [1JT19CS086] Rohit JOshi [1JT19CS074]**

**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. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit

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

|  |  |  |
| --- | --- | --- |
| **SL NO** | **DESCRIPTION** | **PAGE NO** |
| 1. | INTRODUCTION | 06-08 |
| 2. | IMPLEMENTATION | 09-13 |
| 3. | RESULTS AND SNAPSHOTS | 14-17 |
| 4. | CONCLUSION | 18-19 |
| 5. | REFERENCES | 20-21 |

***CHAPTER 1 INTRODUCTION***

###  INTRODUCTION

##### Introduction to Python

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, lets 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 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..

##### Introduction to Tkinter

Tkinter is an open source, portable graphical user interface (GUI) library designed for use in Python scripts.It relies on the Tk library, the GUI library used by Tcl/Tk and Perl, which is in turn implemented in C. Therefore, Tkinter can be said to be implemented using multiple layers. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Tkinter in Python helps in creating GUI Applications with a minimum hassle. Among various GUI Frameworks, Tkinter is the only framework that is built-in into Python's Standard Library. An important feature in favor of Tkinter is that it is cross-platform, so the same code can easily work on Windows, macOS, and Linux.

Python offers multiple options for developing GUI(Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

##### Tic-Tac-Toe

Games for you is an Application implemented using Python. The GUI of the application has been developed using the Tkinter module of python.

Games for you Application is the main window which provides the users an option to select the game they wish to play. A user can select either Color Game or Match Making Game. On selecting the desired game, the user is navigated to that game and can play the game.

##### Scope and importance

This system is developed in visual basic and can run in any windows platform It provides the flexibility and user interface. So briefly this is facilitated with good user interface.

**Scope**: Help with populating your game world

* + 1. Quick start: Helps you build your first rocks and crates using Tkinter.
    2. Understanding Color Tags – explains how you color your Game’s text.
    3. This introduces the full (if small) solo-adventure game that comes with the Tkinter GUI. It is useful both as an example of building and of coding.
    4. Building a Giant Mech – this starts as a building tutorial and transitions into writing code

# CHAPTER 2 IMPLEMENTATION

### CODE:

from tkinter import \*

from threading import Thread

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, 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 == 6:

                    select = random.choice([0, 4, 8])

                    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

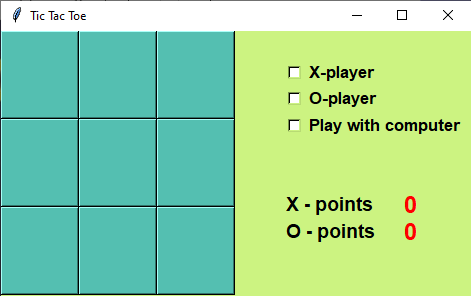
if \_\_name\_\_ == '\_\_main\_\_':

    app = App()

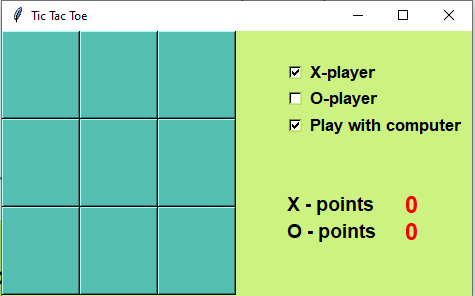
    app.mainloop()

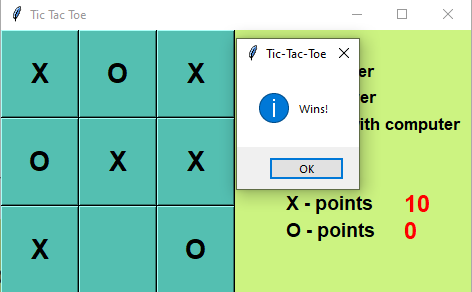
# CHAPTER 3 RESULTS AND SNAPSHOTS

#### Tic-Tac-Toe

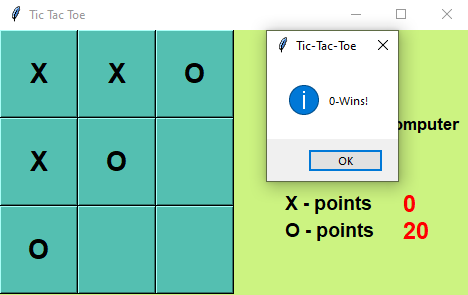


##### Choice Of Players

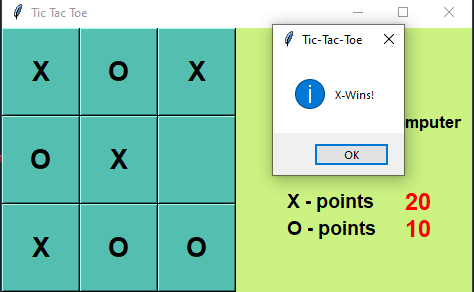


**When Wins**

**When 0 wins at the end**



**When 0 wins at the end**



***CHAPTER 4 CONCLUSION***

**Conclusion**

We have successfully developed color game and matchmaker game using python-tkinter. Tkinter is excellent for small, quick GUI applications, and since it runs on more platforms than any other Python GUI toolkit, it is a good choice where portability is the prime concern.

Obviously, we haven’t been able to give Tkinter the depth of discussion it warrants, but it’s fair to say that almost anything that can be done using the C language can be done using Python and Tkinter. One example is the Python mega widgets (PMW) package mentioned previously; this is a pure Python package that creates an excellent widget set by building on the core Tkinter widgets.

##### Features

1. This is a very flexible approach, compared to some other GUI toolkits.
2. Coding versatility –GUI programming with Qt is built around the idea of signals and slots for creating contact between objects. This allows versatility in dealing with GUI incidents which results in a smoother code base.
3. More than a framework: Qt uses a broad variety of native platform APIs for networking, database development, and more. It provides primary access to them through a special API.

# CHAPTER 5 REFERENCES

REFERENCES:

<https://docs.python.org/>

<https://www.youtube.com/>

<https://stackoverflow.com/>

<https://code.visualstudio.com/docs/>

<https://www.tutorialspoint.com/python/>