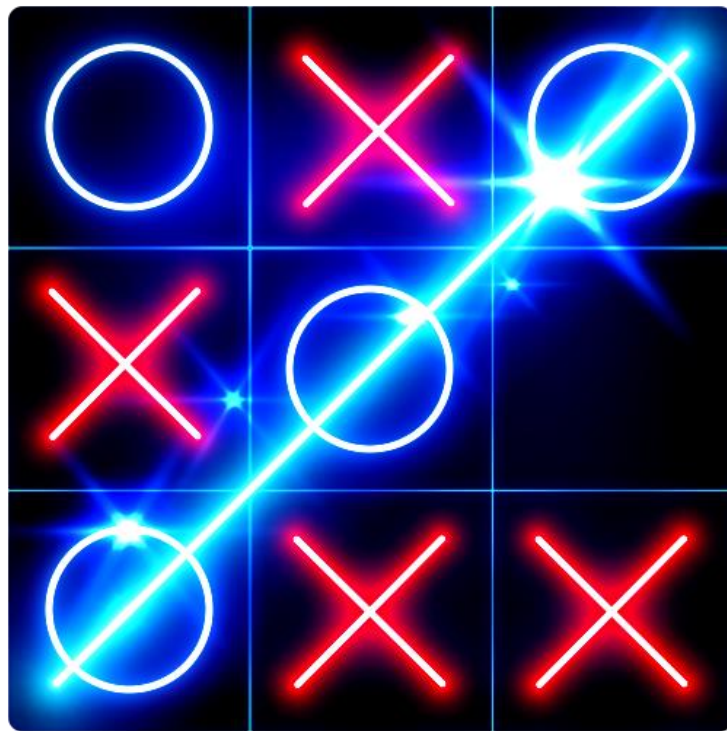


PROJECT ON TIK TAC TOE GAME



BY
R.KEERTHI SRI
12F37

CERTIFICATE

This is to certify that R.Keerthi Sri has successfully completed Computer Project towards partial completion of Practical Examination of AISSCE 2021 as prescribed by CBSE.

Signatures:

Internal Examner

External Examiner

PRINCIPAL

ACKNOWLEDGEMENT

I wish to express my deep gratitude and sincere thanks to all my teachers for encouragement and the management for providing all facilities to successfully complete the project work.

I extend my sincere thanks to my principal; and my Computer Science teacher, whose valuable guidance helped to successfully complete the project .

I extend my gratitude to my parents and classmates for their valuable support and time.

S.NO	CONTENT	PAGE.NO
1	Introduction	5
2	Software and hardware requirement	8
3	Python	9
4	Design works	12
5	Code	15
6	Output	31
7	Bibliography	35

INTRODUCTION

Tic-tac-toe (*also known as noughts and Xs* and Os), is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner. It is a solved game with a forced draw assuming best play from both players.

In order to win the game, a player must place three of their marks in a horizontal, vertical, or diagonal row.

Players soon discover that the best play from both parties leads to a draw. Hence, tic-tac-toe is most often played by young children, who often have not yet discovered the optimal strategy.

1	2	3
4	5	6
7	8	9

Incidence structure for tic-tac-toe. because of the simplicity of tic-tac-toe, it is often used as a pedagogical tool for teaching the concepts of good sportsmanship and the branch of artificial intelligence that deals with the

searching of game trees. It is straightforward to write a computer program to play tic-tac-toe perfectly or to enumerate the 765 essentially different positions (the state space complexity) or the 26,830 possible games up to rotations and reflections (the game tree complexity) on this space.[1] If played optimally by both players, the game always ends in a draw, making tic-tac-toe a futile game.

Each of the cells in the grid was labeled with a number from 1 to 9, so that the players could indicate cell position by number. The number was the critical target word used in the analysis.

There are two reasons why this variant of Tic Tac Toe is useful for separating effects of importance and predictability on acoustic prominence. First, defining importance within the context of Tic Tac Toe is straightforward. An utterance that introduces a game move that wins or blocks the win of a game can be defined as more important than an utterance introducing a move that does not win or block the win of a game. This is particularly advantageous because defining importance is difficult in most conversations. Importance can vary depending on the task, conversation, and intentions of the interlocutor--a point that Bolinger aptly summarized with the title of his classic article: "Accent is predictable (if you are a mind reader)". Within the context of Tic Tac Toe, however, importance is easily operationalized.

Second, Tic Tac Toe allows us to separate contributions of predictability from contributions of importance in acoustic prominence because moves that are important are highly predictable. An importance-based account predicts that a move that is important should have relatively high acoustic prominence. In contrast, a predictability-based account predicts that such a move should have relatively low acoustic prominence because it is highly predictable.

ADVANTAGES or BENEFITS

1. It teaches good sportsmanship.
2. It helps children apply their logic and develop strategy at an early age.
3. It prepares children for more complex games because they have to think of multiple things at one time.
4. Tic-tac-toe helps develop coordination, fine motor skills and visual skills.
5. It helps children learn how to follow rules and take turns.
6. It can help improve your child's concentration.

SYSTEM SOFTWARE AND HARDWARE SOFTWARE

The Software used to run the Python program are:

1. Windows 10 / Windows 7
2. Python 3.6 or above

HARDWARE

The hardware used to run the project are:

1. 2 GHz Dual Core Processor
2. 2 GB RAM
3. 250 GB HDD
4. Monitor with VGA of 1024x768 Screen Resolution. 5

PYTHON

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions- Python 2 and Python 3. Both are quite different. Python 3 is used for this project.

Reason for increasing popularity

- 1.Emphasis on code readability, shorter codes, ease of writing
- 2.Programmers can express logical concepts in fewer lines of code in comparison to languages such as C++ or Java.
- 3.Python supports multiple programming paradigms, like object-oriented, imperative and functional programming or procedural.
- 4.There exists inbuilt functions for almost all of the frequentlyused concepts.
- 5.Philosophy is “Simplicity is the best”.

LANGUAGE FEATURES

- Interpreted
 - There are no separate compilation and execution steps like C and C++.
 - Directly run the program from the source code.
 - Internally, Python converts the source code into an intermediate form called bytecodes which is then translated into native language of specific computer to run it.

- No need to worry about linking and loading with libraries, etc.
- Platform Independent
 - Python programs can be developed and executed on multiple operating system platforms.
 - Python can be used on Linux, Windows, Macintosh, Solaris and many more.
- Free and Open Source; Redistributable
- High-level Language
 - In Python, no need to take care about low-level details such as managing the memory used by the program.
- Simple
 - Closer to English language; Easy to Learn
 - More emphasis on the solution to the problem rather than the syntax
- Embeddable
 - Python can be used within C/C++ program to give scripting capabilities for the program's users.
- Robust:
 - Exceptional handling features
 - Memory management techniques in built
- Rich Library Support
 - The Python Standard Library is very vast.
 - Known as the “batteries included” philosophy of Python ;It can help do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, WAV files, cryptography, GUI and many more.
 - Besides the standard library, there are various other high-quality libraries such as the Python Imaging Library which is an amazingly simple image manipulation library.

LIBRARIES USED

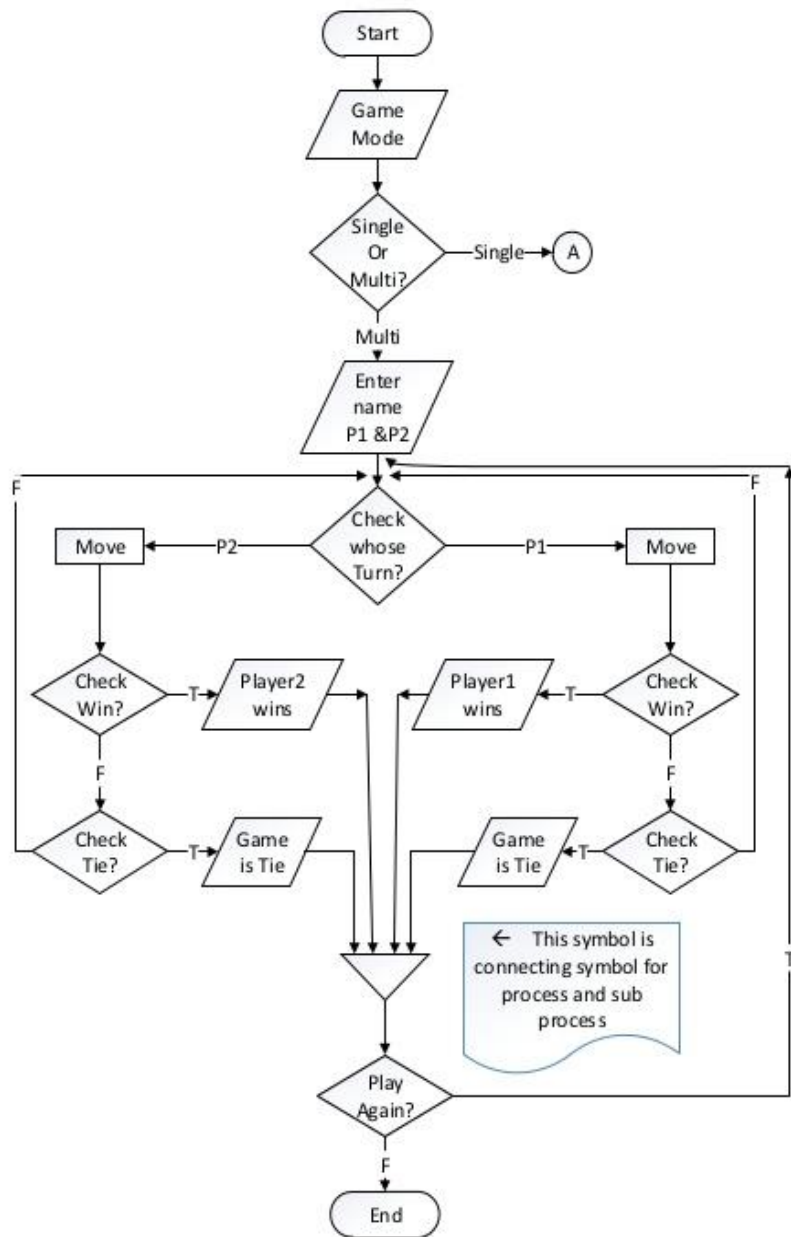
1. tkinter : Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. GUI is nothing but a desktop app that provides you with an interface that helps you to interact with the computers and enriches your experience of giving a command (command-line input) to your code. They are used to perform different tasks in desktops, laptops, and other electronic devices, etc.

Methods used

- i. geometry() : This method is used to set the dimensions of the Tkinter window as well as it is used to set the
- ii. position of the main window on the user's desktop.
- iii. Frame() : It works like a container, which is responsible for arranging the position of other widgets. It uses
- iv. rectangular areas in the screen to organize the layout
- v. and to provide padding of these widgets.
- vi. grid() : This geometry manager organizes widgets in a table-like structure in the parent widget.
- vii. pack () : This geometry manager organizes widgets in blocks before placing them in the parent widget.
- viii. Label () : This widget implements a display box where you can place text or images. The text displayed by this
- ix. widget can be updated at any time you want.
- x. Button () : The Button widget is used to add buttons in a Python application. These buttons can display text or images that convey the purpose of the buttons. You can attach a function or a method to a button which is called automatically when you click the button.
- xi. Entry () : The Entry widget is used to accept single-line text strings from a user.

3. METHODOLOGY

3.1 FLOWCHART



3.2 USE CASE DIAGRAM

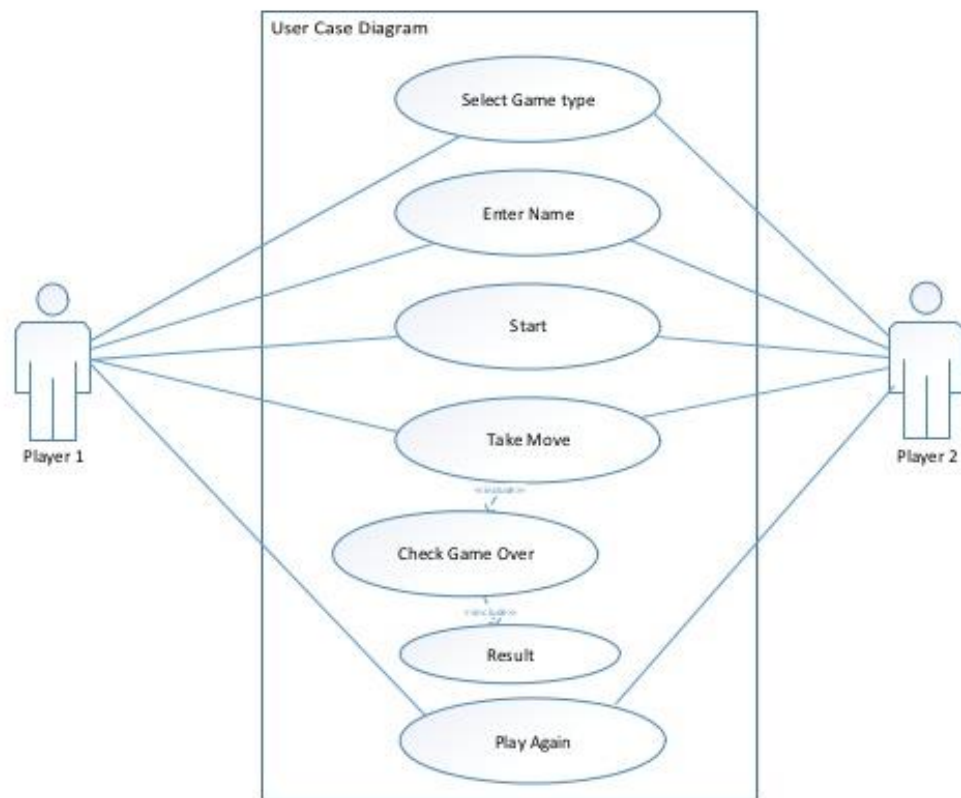
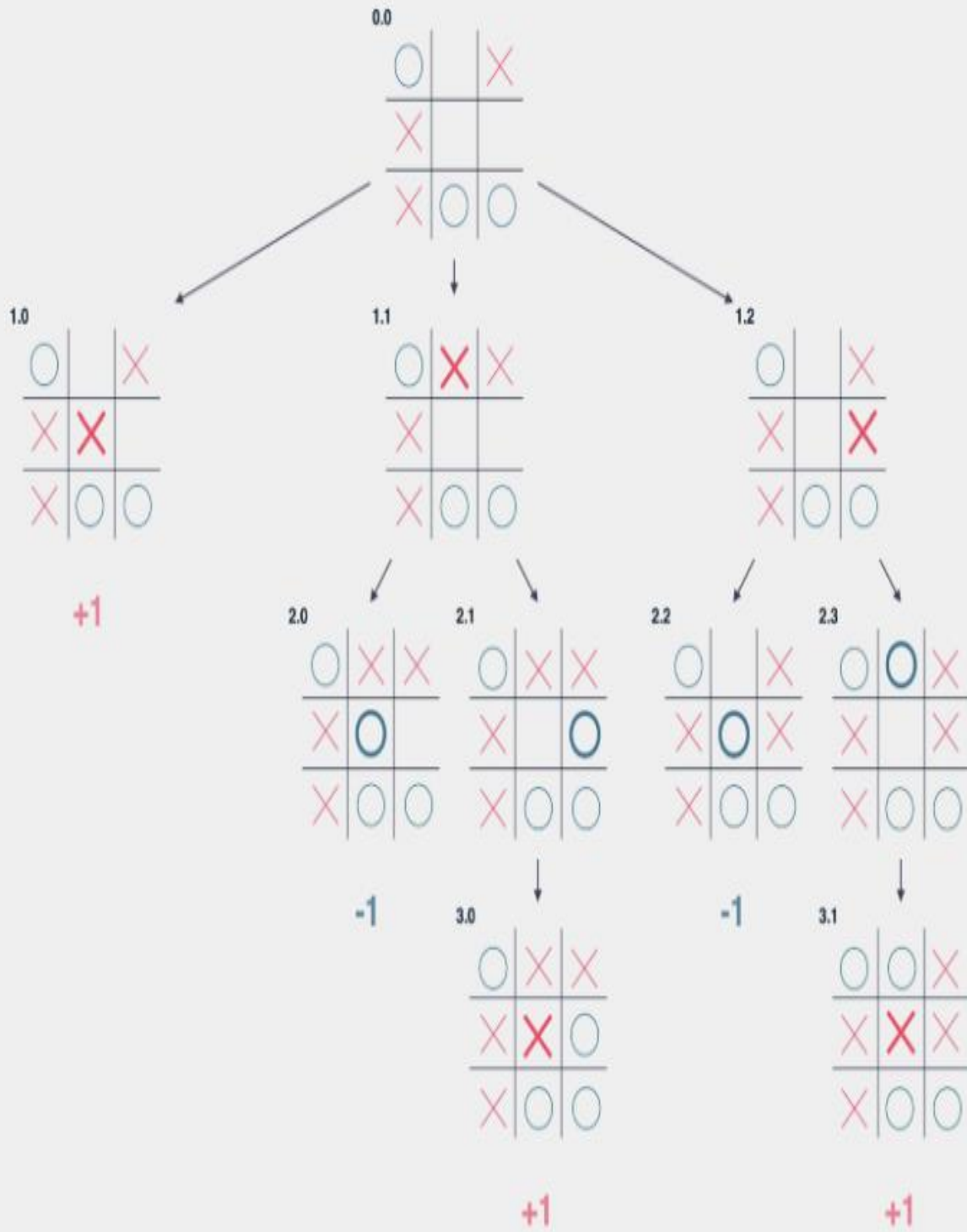


Fig 1.2.a: Use case diagram of Player VS Player



CODE

```
import tkinter.messagebox

from tkinter import*

root=Tk()

root.geometry("1350x750+0+0")

root.title("tic tac toe")

root.configure(background ='Light Blue')


Tops=Frame(root ,bg='Light Blue' , pady= 2, width =1350 , height=100,
relief =RIDGE)

Tops.grid(row=0 ,column=0)


lblTitle= Label(Tops , font =( 'arial',50,'bold'), text='Tic Tac Toe Game',
bd=25, bg='Light Blue' ,fg='cornsilk',justify= CENTER )

lblTitle.grid(row=0 ,column=0)


MainFrame=Frame(root ,bg= 'Dark Blue', pady= 2, width =1350 ,
height=600 , relief = RIDGE)
```

```
MainFrame.grid(row=1 , column=0)
```

```
LeftFrame= Frame(MainFrame , bd=10 ,width=750 ,height=500,  
pady=2 , padx=10, bg='Light Blue', relief= RIDGE)
```

```
LeftFrame.pack(side=LEFT)
```

```
RightFrame= Frame(MainFrame , bd=10 ,width=560 ,height=500  
,pady=2 , padx=10, bg='Light Blue', relief= RIDGE)
```

```
RightFrame.pack(side=RIGHT)
```

```
RightFrame1 = Frame(RightFrame , bd=10 ,width=560 ,height= 200,  
pady=2 , padx=10, bg='Light Blue', relief= RIDGE)
```

```
RightFrame1.grid(row=0 , column=0)
```

```
RightFrame2 = Frame(RightFrame , bd=10 ,width=560 ,height= 200,  
pady=2 , padx=10, bg='Light Blue', relief= RIDGE)
```

```
RightFrame2.grid(row=1, column=0)
```

```
PlayerX= IntVar()
```

```
PlayerO= IntVar()
```



```
PlayerX.set(0)
```

```
PlayerO.set(0)
```

```
buttons =StringVar()
```

```
click=True
```

```
def checker(buttons):
```

```
    global click
```

```
    if buttons['text']==' ' and click== True :
```

```
        buttons['text']="X"
```

```
        click=False
```

```
        scorekeeper()
```

```
    elif buttons['text']==' ' and click== False :
```

```
        buttons['text']="O"
```

```
        click=True
```

```
        scorekeeper()
```

```
def scorekeeper():
```

#X BUTTONS

#Horizontal

```
if(button1['text']=='X' and button2['text']=='X' and  
button3['text']=='X'):
```

```
    button1.configure(background = "Light Green")
```

```
    button2.configure(background = "Light Green")
```

```
    button3.configure(background = "Light Green")
```

```
    n= int(PlayerX.get())
```

```
    score=(n+1)
```

```
    PlayerX.set(score)
```

```
    tkinter.messagebox.showinfo('Winner X','YOU HAVE WON A  
GAME')
```

```
if(button4['text']=='X' and button5['text']=='X' and  
button6['text']=='X'):
```

```
    button4.configure(background = "Light Green")
```

```
    button5.configure(background = "Light Green")
```

```
    button6.configure(background = "Light Green")
```

```
    n= int(PlayerX.get())
```

```
score=(n+1)
```

```
PlayerX.set(score)
```

```
tkinter.messagebox.showinfo('Winner X','YOU HAVE WON A  
GAME')
```

```
if(button7['text']=='X' and button8['text']=='X' and  
button9['text']=='X'):
```

```
button7.configure(background = "Light Green")
```

```
button8.configure(background = "Light Green")
```

```
button9.configure(background = "Light Green")
```

```
n= int(PlayerX.get())
```

```
score=(n+1)
```

```
PlayerX.set(score)
```

```
tkinter.messagebox.showinfo('Winner X','YOU HAVE WON A  
GAME')
```

```
#DIAGONAL
```

```
if(button1['text']=='X' and button5['text']=='X' and  
button9['text']=='X'):
```

```
button1.configure(background = "Light Green")
```

```
button5.configure(background = "Light Green")
```

```
button9.configure(background = "Light Green")
```

```
n= int(PlayerX.get())  
  
score=(n+1)  
  
PlayerX.set(score)  
  
tkinter.messagebox.showinfo('Winner X','YOU HAVE WON A  
GAME')
```

```
if(button3['text']=='X' and button5['text']=='X' and  
button7['text']=='X'):
```

```
    button5.configure(background = "Light Green")  
  
    button7.configure(background = "Light Green")  
  
    button3.configure(background = "Light Green")  
  
    n= int(PlayerX.get())  
  
    score=(n+1)  
  
    PlayerX.set(score)  
  
    tkinter.messagebox.showinfo('Winner X','YOU HAVE WON A  
GAME')
```

```
#VERTICAL
```

```
if(button1['text']=='X' and button4['text']=='X' and  
button7['text']=='X'):
```

```
    button1.configure(background = "Light Green")
```

```
button4.configure(background = "Light Green")

button7.configure(background = "Light Green")

n= int(PlayerX.get())

score=(n+1)

PlayerX.set(score)

tkinter.messagebox.showinfo('Winner X','YOU HAVE WON A
GAME')
```

```
if(button2['text']=='X' and button5['text']=='X' and
button8['text']=='X'):
```

```
    button5.configure(background = "Light Green")

    button2.configure(background = "Light Green")

    button8.configure(background = "Light Green")

    n= int(PlayerX.get())

    score=(n+1)

    PlayerX.set(score)

    tkinter.messagebox.showinfo('Winner X','YOU HAVE WON A
GAME')
```

```
if(button3['text']=='X' and button6['text']=='X' and button9['text']=='X'):
```

```
button9.configure(background = "Light Green")  
button6.configure(background = "Light Green")  
button3.configure(background = "Light Green")  
  
n= int(PlayerX.get())  
  
score=(n+1)  
  
PlayerX.set(score)  
  
tkinter.messagebox.showinfo('Winner X','YOU HAVE WON A  
GAME')
```

#O BUTTONS

#horizontal

```
if(button1['text']=='O' and button2['text']=='O' and  
button3['text']=='O'):
```

```
    button1.configure(background = "Yellow")
```

```
    button2.configure(background = "Yellow")
```

```
    button3.configure(background = "Yellow")
```

```
    n= int(PlayerO.get())
```

```
    score=(n+1)
```

```
    PlayerO.set(score)
```

```
tkinter.messagebox.showinfo('Winner O','YOU HAVE WON A  
GAME')
```

```
if(button4['text']=='O' and button5['text']=='O' and  
button6['text']=='O'):
```

```
    button4.configure(background ="Yellow")
```

```
    button5.configure(background ="Yellow")
```

```
    button6.configure(background ="Yellow")
```

```
    n= int(PlayerO.get())
```

```
    score=(n+1)
```

```
    PlayerO.set(score)
```

```
tkinter.messagebox.showinfo('Winner O','YOU HAVE WON A  
GAME')
```

```
if(button7['text']=='O' and button8['text']=='O' and  
button9['text']=='O'):
```

```
    button7.configure(background ="Yellow")
```

```
    button8.configure(background ="Yellow")
```

```
    button9.configure(background ="Yellow")
```

```
    n= int(PlayerO.get())
```

```
    score=(n+1)
```

```
PlayerO.set(score)
```

```
tkinter.messagebox.showinfo('Winner O','YOU HAVE WON A  
GAME')
```

```
#VERTICAL
```

```
if(button1['text']=='O' and button4['text']=='O' and  
button7['text']=='O'):
```

```
    button1.configure(background ="Yellow")
```

```
    button4.configure(background ="Yellow")
```

```
    button7.configure(background ="Yellow")
```

```
    n= int(PlayerO.get())
```

```
    score=(n+1)
```

```
    PlayerO.set(score)
```

```
    tkinter.messagebox.showinfo('Winner O','YOU HAVE WON A  
GAME')
```

```
if(button2['text']=='O' and button5['text']=='O' and  
button8['text']=='O'):
```

```
    button2.configure(background ="Yellow")
```

```
    button5.configure(background ="Yellow")
```

```
    button8.configure(background ="Yellow")
```

```
    n= int(PlayerO.get())
```



```
score=(n+1)
```

```
PlayerO.set(score)
```

```
tkinter.messagebox.showinfo('Winner O','YOU HAVE WON A  
GAME')
```

```
if(button3['text']=='O' and button6['text']=='O' and  
button9['text']=='O'):
```

```
button3.configure(background ="Yellow")
```

```
button6.configure(background ="Yellow")
```

```
button9.configure(background ="Yellow")
```

```
n= int(PlayerO.get())
```

```
score=(n+1)
```

```
PlayerO.set(score)
```

```
tkinter.messagebox.showinfo('Winner O','YOU HAVE WON A  
GAME')
```

```
#DIAGONAL
```

```
if(button1['text']=='O' and button5['text']=='O' and  
button9['text']=='O'):
```

```
button1.configure(background ="Yellow")
```

```
button5.configure(background ="Yellow")
```

```
button9.configure(background = "Yellow")

n= int(PlayerO.get())

score=(n+1)

PlayerO.set(score)

tkinter.messagebox.showinfo('Winner O','YOU HAVE WON A
GAME')
```

```
if(button3['text']=='O' and button5['text']=='O' and
button7['text']=='O'):
```

```
    button3.configure(background = "Yellow")

    button5.configure(background = "Yellow")

    button7.configure(background = "Yellow")

    n= int(PlayerO.get())

    score=(n+1)

    PlayerO.set(score)

    tkinter.messagebox.showinfo('Winner O','YOU HAVE WON A
GAME')
```

```
def reset():
```

```
    button1['text']=' '
```

```
button2['text']=' '
```

```
button3['text']=' '
```

```
button4['text']=' '
```

```
button5['text']=' '
```

```
button6['text']=' '
```

```
button7['text']=' '
```

```
button8['text']=' '
```

```
button9['text']=' '
```

```
button1.configure(background ='gainsboro')
```

```
button2.configure(background ='gainsboro')
```

```
button3.configure(background ='gainsboro')
```

```
button4.configure(background ='gainsboro')
```

```
button5.configure(background ='gainsboro')
```

```
button6.configure(background ='gainsboro')
```

```
button7.configure(background ='gainsboro')
```

```
button8.configure(background ='gainsboro')
```

```
button9.configure(background ='gainsboro')
```

```
def NewGame():
```

```
    reset()
```

```
    PlayerX.set(0)
```

```
    PlayerO.set(0)
```

```
lblPlayerX= Label(RightFrame1, font=('arial',40, "bold"), text= "Player  
X:" ,padx=2, pady=2, bg='Light Blue')
```

```
lblPlayerX.grid(row=0,column=0 , sticky=W)
```

```
txtPlayerX=Entry(RightFrame1, font=('arial',40, "bold"), bd=2,  
fg='black', textvariable= PlayerX, width=14,  
justify=LEFT).grid(row=0,column=1)
```

```
lblPlayerO= Label(RightFrame1, font=('arial',40, "bold"), text= "Player  
O:" ,padx=2, pady=2, bg='Light Blue')
```

```
lblPlayerO.grid(row=1,column=0 , sticky=W)
```

```
txtPlayerO=Entry(RightFrame1, font=('arial',40, "bold"), bd=2,  
fg='black', textvariable= PlayerO, width=14,
```

```
justify=LEFT).grid(row=1,column=1)
```

```
btnReset = Button(RightFrame2, text= "RESET " , font=('arial',40,  
"bold"), height = 1 , width=20, bg ='gainsboro', command= reset )
```

```
btnReset.grid(row=2, column=0, padx=6 , pady=11)
```

```
btnNewGame = Button(RightFrame2, text= "NEW GAME " ,  
font=('arial',40, "bold"), height =1 , width=20, bg='gainsboro'  
,command= NewGame)
```

```
btnNewGame.grid(row=3, column=0 , padx=6 , pady=10)
```

```
button1 = Button(LeftFrame, text= " " , font=('Times 26 bold'), height =  
3 ,width=8,bg ='gainsboro',command= lambda:checker(button1))
```

```
button1.grid(row=1, column=0, sticky=S+N+E+W)
```

```
button2 = Button(LeftFrame, text= " " , font=('Times 26 bold'), height  
=3 , width=8, bg='gainsboro' ,command= lambda:checker(button2))
```

```
button2.grid(row=1, column=1, sticky=S+N+E+W)
```

```
button3 = Button(LeftFrame, text= " " , font=('Times 26 bold'), height  
=3 , width=8, bg='gainsboro' ,command= lambda:checker(button3))
```

```
button3.grid(row=1, column=2, sticky=S+N+E+W)
```

```
button4 = Button(LeftFrame, text= " " , font=('Times 26 bold'), height  
=3 , width=8, bg='gainsboro',command= lambda:checker(button4))
```

```
button4.grid(row=2, column=0, sticky=S+N+E+W)
```

```
button5 = Button(LeftFrame, text= " ", font=('Times 26 bold'), height
=3 , width=8, bg='gainsboro' ,command= lambda:checker(button5))
```

```
button5.grid(row=2, column=1, sticky=S+N+E+W)
```

```
button6 = Button(LeftFrame, text= " ", font=('Times 26 bold'), height
=3 , width=8, bg='gainsboro' ,command= lambda:checker(button6))
```

```
button6.grid(row=2, column=2, sticky=S+N+E+W)
```

```
button7 = Button(LeftFrame, text= " ", font=('Times 26 bold'), height
=3 , width=8, bg='gainsboro' ,command= lambda:checker(button7))
```

```
button7.grid(row=3, column=0, sticky=S+N+E+W)
```

```
button8 = Button(LeftFrame, text= " ", font=('Times 26 bold'), height
=3 , width=8, bg='gainsboro' ,command= lambda:checker(button8))
```

```
button8.grid(row=3, column=1, sticky=S+N+E+W)
```

```
button9 = Button(LeftFrame, text= " ", font=('Times 26 bold'), height
=3 , width=8, bg='gainsboro' ,command= lambda:checker(button9))
```

```
button9.grid(row=3, column=2, sticky=S+N+E+W)
```

```
root.mainloop()
```

OUTPUT









BIBLIOGRAPHY

1. <https://automatetheboringstuff.com>
2. <https://www.python.org>
3. <https://docs.python.org> › library › tkinter
4. <https://www.geeksforgeeks.org> › python-gui-tkinter