**Bingo Game**

**A PROJECT REPORT**

*Submitted by*

Dharineesh J

A David Vedha Jerome

R Dhruv

K Aswin

*in partial fulfilment for the award of the degree*

*of*

BACHELOR OF ENGINEERING

*in*

COMPUTER SCIENCE AND ENGINEERING

PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH,

COIMBATORE 641 062

ANNA UNIVERSITY: CHENNAI - 600 025

FEBRUARY 2023

**ABSTRACT**

Software development using Python programming

language And Website.

This report is about our project titled Bingo Game using python programming language. Our team of four started the project on 12/01/2023. We started discussing on various ideas and finally came up with an idea of creating a simple game.

We started studying and researching about the tkinter module and its methods which was helpful in creating the Graphical User Interface (GUI) for our game. Then we took some references from the internet about the logic of our game.

This game is playable in two modes. One is (Computer vs Player) mode and (Player 1 vs Player 2)mode. The logic of our game is in a square grid of 5x5 containing 25 numbers, the first to strike all the five numbers horizontally, vertically or diagonally wins the game.

We have used tkinter module for the window generation, grid formation, background color, adjusting title and geometry of the window. Time module has been used for the loading screen generation.

This game can be used as model to learn about tkinter and its applications in an interactive manner.

For the website, we used HTML Language to create it. In the creation of website, we used CSS(CASCADING STYLE SHEETS) for styling purposes and image also been displayed. For the taskbar, we used Favicons.

**IMPLEMENTATION**

We decided to make a simple game using a simple logic. We started from scratch. We came up with the idea and started working on the algorithm. Some references from the internet were useful in making progress with our work.

Then after completing the logical part, we started working on the GUI of the Bingo Game. On researching further, we came to know about tkinter module which was helpful in creating the window with desirable size, number of grids, background color of the window, title as well as geometry. Random module is also used in order to generate random numbers each time.

Apart from many functions in tkinter, we have used functions like tk() for creating a window, title() for the title of the window, grid() to form grids into the created window, Font() for the font style, Button() for creating functional buttons, the Frame() method is used in organizing the group of widgets, the pack () method sizes the frame so that all the contents are at or above their preferred sizes and so on.

Other modules like random, time, bootstrap have also been used in this project.

In website, we have published our codes of python projects, which contains codes of both mode (Computer vs Player) and (Player vs Player2). We have included Our Contact Details and Feedback section. It Contains rules and description of the game. In Feedback section, People can give their feedback by entering their details and it will be sent directly to our mail.

**CODING:**

from tkinter import \*

import time

import tkinter.ttk as p

import tkinter.font as font

import random

import ttkbootstrap as s

import tkinter as tk

global coml,cl,btlist1,btlist2,nownumber,l1,l2,l3,l4,load

coml=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25]

cl=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25]

nownumber=1

art ='''¯\\_(ツ)\_/¯'''

art2 = '''

@(\/)

(\/)-{}-)@

@(={}=)/\)(\/)

(\/(/\)@| (-{}-)

(={}=)@(\/)@(/\)@

(/\)\(={}=)/(\/)

@(\/)\(/\)/(={}=)

(-{}-)""""@/(/\)

|: |

/::' \\

/::: \\

|::' |

|:: |

\::. /

':\_\_\_\_\_\_.'

`""""""`'''

output\_image\_size = 0

Root=Tk()

Root.title('BINGO GAMES')

Root.geometry('500x600')

Root.resizable(width =False, height=False)

f = Frame(Root)

title = Label(f,text='Bingo Game')

title.config(font=('courier',33))

title.grid(pady=10)

b\_encode = Button(f,text="Two player Game",command= lambda :(New()), padx=14)

b\_encode.config(font=('courier',14))

b\_decode = Button(f, text="Single player Game",padx=14,command=lambda: (main()))

b\_decode.config(font=('courier',14))

b\_decode.grid(pady = 12)

ascii\_art = Label(f,text=art)

# ascii\_art.config(font=('MingLiU-ExtB',50))

ascii\_art.config(font=('courier',60),bg='#00FFFF')

ascii\_art2 = Label(f,text=art2)

# ascii\_art.config(font=('MingLiU-ExtB',50))

ascii\_art2.config(font=('courier',12,'bold'),bg='#00FFFF')

Root.grid\_rowconfigure(1, weight=10)

Root.grid\_columnconfigure(0, weight=10)

f.grid()

f.configure(bg='#00FFFF')

title.grid(row=1)

b\_encode.grid(row=2)

b\_decode.grid(row=3)

ascii\_art.grid(row=4,pady=10)

ascii\_art2.grid(row=5,pady=5)

def main():

f.destroy()

tk=Tk()

tk.geometry("500x500")

tk.title("Single Player Game")

myFont=font.Font(family='Helvetica',size=120,weight='bold')

but1=Button(tk,text='Start The Game',command=lambda:(randboard(tk)))

but1['font']=myFont

but1.pack(pady=30)

def randboard(t):

f.destroy()

global butlist,l

btk=Tk()

btk.title("Single Player Game")

btk.geometry("470x440")

but1\_1=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but1\_1))

but1\_2=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but1\_2))

but1\_3=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but1\_3))

but1\_4=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but1\_4))

but1\_5=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but1\_5))

but2\_1=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but2\_1))

but2\_2=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but2\_2))

but2\_3=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but2\_3))

but2\_4=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but2\_4))

but2\_5=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but2\_5))

but3\_1=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but3\_1))

but3\_2=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but3\_2))

but3\_3=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but3\_3))

but3\_4=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but3\_4))

but3\_5=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but3\_5))

but4\_1=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but4\_1))

but4\_2=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but4\_2))

but4\_3=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but4\_3))

but4\_4=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but4\_4))

but4\_5=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but4\_5))

but5\_1=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but5\_1))

but5\_2=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but5\_2))

but5\_3=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but5\_3))

but5\_4=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but5\_4))

but5\_5=Button(btk,text='',height=5,width=12,command=lambda:game(btk,but5\_5))

butlist=[but1\_1,but1\_2,but1\_3,but1\_4,but1\_5,

but2\_1,but2\_2,but2\_3,but2\_4,but2\_5,

but3\_1,but3\_2,but3\_3,but3\_4,but3\_5,

but4\_1,but4\_2,but4\_3,but4\_4,but4\_5,

but5\_1,but5\_2,but5\_3,but5\_4,but5\_5]

l=list(range(1,26))

random.shuffle(l)

for i in range(25):

butlist[i]['text']=l[i]

but1\_1.grid(row=1,column=1)

but1\_2.grid(row=1,column=2)

but1\_3.grid(row=1,column=3)

but1\_4.grid(row=1,column=4)

but1\_5.grid(row=1,column=5)

but2\_1.grid(row=2,column=1)

but2\_2.grid(row=2,column=2)

but2\_3.grid(row=2,column=3)

but2\_4.grid(row=2,column=4)

but2\_5.grid(row=2,column=5)

but3\_1.grid(row=3,column=1)

but3\_2.grid(row=3,column=2)

but3\_3.grid(row=3,column=3)

but3\_4.grid(row=3,column=4)

but3\_5.grid(row=3,column=5)

but4\_1.grid(row=4,column=1)

but4\_2.grid(row=4,column=2)

but4\_3.grid(row=4,column=3)

but4\_4.grid(row=4,column=4)

but4\_5.grid(row=4,column=5)

but5\_1.grid(row=5,column=1)

but5\_2.grid(row=5,column=2)

but5\_3.grid(row=5,column=3)

but5\_4.grid(row=5,column=4)

but5\_5.grid(row=5,column=5)

def custboard(t):

t.destroy()

def game(t,b):

n=b['text']

if b['text']!='\*':

b['text']='\*'

compplay(t,n)

def compplay(t,n):

global coml,cl,butlist,l

l[l.index(int(n))]='\*'

coml[coml.index(int(n))]='\*'

cl.pop(cl.index(int(n)))

if statement(t)== False:

a=random.choice(cl)

cl.pop(cl.index(a))

coml[coml.index(a)]='\*'

butlist[l.index(a)]['text']='\*'

l[l.index(a)]='\*'

statement(t)

def plcheck():

global l

s=0

if l[0]==l[1]==l[2]==l[3]==l[4]:

s+=1

if l[5]==l[6]==l[7]==l[8]==l[9]:

s+=1

if l[10]==l[11]==l[12]==l[13]==l[14]:

s+=1

if l[15]==l[16]==l[17]==l[18]==l[19]:

s+=1

if l[20]==l[21]==l[22]==l[23]==l[24]:

s+=1

if l[0]==l[5]==l[10]==l[15]==l[20]:

s+=1

if l[1]==l[6]==l[11]==l[16]==l[21]:

s+=1

if l[2]==l[7]==l[12]==l[17]==l[22]:

s+=1

if l[3]==l[8]==l[13]==l[18]==l[23]:

s+=1

if l[4]==l[9]==l[14]==l[19]==l[24]:

s+=1

if l[0]==l[6]==l[12]==l[18]==l[24]:

s+=1

if l[4]==l[8]==l[12]==l[16]==l[20]:

s+=1

if s>=5:

return True

else:

return False

def compcheck():

global coml

s=0

if coml[0]==coml[1]==coml[2]==coml[3]==coml[4]:

s+=1

if coml[5]==coml[6]==coml[7]==coml[8]==coml[9]:

s+=1

if coml[10]==coml[11]==coml[12]==coml[13]==coml[14]:

s+=1

if coml[15]==coml[16]==coml[17]==coml[18]==coml[19]:

s+=1

if coml[20]==coml[21]==coml[22]==coml[23]==coml[24]:

s+=1

if coml[0]==coml[5]==coml[10]==coml[15]==coml[20]:

s+=1

if coml[1]==coml[6]==coml[11]==coml[16]==coml[21]:

s+=1

if coml[2]==coml[7]==coml[12]==coml[17]==coml[22]:

s+=1

if coml[3]==coml[8]==coml[13]==coml[18]==coml[23]:

s+=1

if coml[4]==coml[9]==coml[14]==coml[19]==coml[24]:

s+=1

if coml[0]==coml[6]==coml[12]==coml[18]==coml[24]:

s+=1

if coml[4]==coml[8]==coml[12]==coml[16]==coml[20]:

s+=1

if s>=5:

return True

else:

return False

def statement(t):

if plcheck()==True and compcheck()==True:

t.destroy()

tk=Tk()

Label(tk,text='Draw').pack()

Label(tk,text=compboard()).pack()

Label(tk,text=plboard()).pack()

elif plcheck()==True and compcheck()==False:

t.destroy()

tk=Tk()

Label(tk,text='Win').pack()

Label(tk,text=compboard()).pack()

Label(tk,text=plboard()).pack()

elif plcheck()==False and compcheck()==True:

t.destroy()

tk=Tk()

Label(tk,text='Lose').pack()

Label(tk,text=compboard()).pack()

Label(tk,text=plboard()).pack()

else:

return False

def compboard():

global coml

x=0

s='''Computer Board

'''

for i in range(5):

s+='| '

for j in range(5):

s+=str(coml[x])+' | '

x+=1

s+='\n'

return s

def plboard():

global l

x=0

s='''Player board

'''

for i in range(5):

s+=' | '

for j in range(5):

s+=str(l[x])+' | '

x+=1

s+='\n'

return s

def New():

f.destroy()

root=s.Window(themename="cyborg")

root.geometry("500x500")

root.title("TWO PLAYER BINGO GAMES")

load=p.Progressbar(root,orient=HORIZONTAL,length=300,mode="determinate")

frame=Frame(root,width=10,height=20).pack()

myFont=font.Font(family='Sofia Sans',size=20,weight='bold')

but1=Button(frame,text='Start the game',command=lambda:(bar(root)))

but1['font']=myFont

but1.pack(pady=30)

def bar(root):

load['value']=40

root.update\_idletasks()

time.sleep(0.3)

load['value']=60

root.update\_idletasks()

time.sleep(0.3)

load['value']=8

root.update\_idletasks()

time.sleep(0.3)

load['value']=100

root.update\_idletasks()

time.sleep(0.3)

load['value']=140

root.update\_idletasks()

time.sleep(0.3)

load['value']=160

root.update\_idletasks()

time.sleep(0.3)

load['value']=180

root.update\_idletasks()

time.sleep(0.3)

load['value']=200

root.update\_idletasks()

time.sleep(0.3)

load['value']=240

root.update\_idletasks()

time.sleep(0.3)

load['value']=260

root.update\_idletasks()

time.sleep(1)

load['value']=280

root.update\_idletasks()

time.sleep(0.3)

load['value']=300

root.update\_idletasks()

time.sleep(0.3)

load.pack(pady=30)

board1(root)

board2(root)

#inserting button for the player1

def board1(t):

global btlist1,l1,l3

f.destroy()

btk=Tk()

btk.geometry("510x518")

btk.title('Player 1')

but1\_1=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but1\_1))

but1\_2=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but1\_2))

but1\_3=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but1\_3))

but1\_4=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but1\_4))

but1\_5=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but1\_5))

but2\_1=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but2\_1))

but2\_2=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but2\_2))

but2\_3=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but2\_3))

but2\_4=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but2\_4))

but2\_5=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but2\_5))

but3\_1=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but3\_1))

but3\_2=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but3\_2))

but3\_3=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but3\_3))

but3\_4=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but3\_4))

but3\_5=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but3\_5))

but4\_1=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but4\_1))

but4\_2=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but4\_2))

but4\_3=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but4\_3))

but4\_4=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but4\_4))

but4\_5=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but4\_5))

but5\_1=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but5\_1))

but5\_2=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but5\_2))

but5\_3=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but5\_3))

but5\_4=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but5\_4))

but5\_5=Button(btk,text='',height=5,width=12,command=lambda:game1(btk,but5\_5))

btlist1=[but1\_1,but1\_2,but1\_3,but1\_4,but1\_5,

but2\_1,but2\_2,but2\_3,but2\_4,but2\_5,

but3\_1,but3\_2,but3\_3,but3\_4,but3\_5,

but4\_1,but4\_2,but4\_3,but4\_4,but4\_5,

but5\_1,but5\_2,but5\_3,but5\_4,but5\_5]

l1=list(range(1,26))

random.shuffle(l1)

l3=l1[:]

for i in range(len(l1)):

btlist1[i]['text']=l1[i]

but1\_1.grid(row=1,column=1)

but1\_2.grid(row=1,column=2)

but1\_3.grid(row=1,column=3)

but1\_4.grid(row=1,column=4)

but1\_5.grid(row=1,column=5)

but2\_1.grid(row=2,column=1)

but2\_2.grid(row=2,column=2)

but2\_3.grid(row=2,column=3)

but2\_4.grid(row=2,column=4)

but2\_5.grid(row=2,column=5)

but3\_1.grid(row=3,column=1)

but3\_2.grid(row=3,column=2)

but3\_3.grid(row=3,column=3)

but3\_4.grid(row=3,column=4)

but3\_5.grid(row=3,column=5)

but4\_1.grid(row=4,column=1)

but4\_2.grid(row=4,column=2)

but4\_3.grid(row=4,column=3)

but4\_4.grid(row=4,column=4)

but4\_5.grid(row=4,column=5)

but5\_1.grid(row=5,column=1)

but5\_2.grid(row=5,column=2)

but5\_3.grid(row=5,column=3)

but5\_4.grid(row=5,column=4)

but5\_5.grid(row=5,column=5)

#inserting button for player2

def board2(t):

global btlist2,l2,l4

f.destroy()

atk=Tk()

atk.geometry("510x518")

atk.title('Player 2')

but1\_1=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but1\_1))

but1\_2=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but1\_2))

but1\_3=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but1\_3))

but1\_4=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but1\_4))

but1\_5=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but1\_5))

but2\_1=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but2\_1))

but2\_2=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but2\_2))

but2\_3=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but2\_3))

but2\_4=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but2\_4))

but2\_5=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but2\_5))

but3\_1=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but3\_1))

but3\_2=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but3\_2))

but3\_3=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but3\_3))

but3\_4=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but3\_4))

but3\_5=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but3\_5))

but4\_1=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but4\_1))

but4\_2=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but4\_2))

but4\_3=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but4\_3))

but4\_4=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but4\_4))

but4\_5=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but4\_5))

but5\_1=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but5\_1))

but5\_2=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but5\_2))

but5\_3=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but5\_3))

but5\_4=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but5\_4))

but5\_5=Button(atk,text='',height=5,width=12,command=lambda:game2(atk,but5\_5))

btlist2=[but1\_1,but1\_2,but1\_3,but1\_4,but1\_5,

but2\_1,but2\_2,but2\_3,but2\_4,but2\_5,

but3\_1,but3\_2,but3\_3,but3\_4,but3\_5,

but4\_1,but4\_2,but4\_3,but4\_4,but4\_5,

but5\_1,but5\_2,but5\_3,but5\_4,but5\_5]

l2=list(range(1,26))

random.shuffle(l2)

l4=l2[:]

for i in range(len(l2)):

btlist2[i]['text']=l2[i]

but1\_1.grid(row=1,column=1)

but1\_2.grid(row=1,column=2)

but1\_3.grid(row=1,column=3)

but1\_4.grid(row=1,column=4)

but1\_5.grid(row=1,column=5)

but2\_1.grid(row=2,column=1)

but2\_2.grid(row=2,column=2)

but2\_3.grid(row=2,column=3)

but2\_4.grid(row=2,column=4)

but2\_5.grid(row=2,column=5)

but3\_1.grid(row=3,column=1)

but3\_2.grid(row=3,column=2)

but3\_3.grid(row=3,column=3)

but3\_4.grid(row=3,column=4)

but3\_5.grid(row=3,column=5)

but4\_1.grid(row=4,column=1)

but4\_2.grid(row=4,column=2)

but4\_3.grid(row=4,column=3)

but4\_4.grid(row=4,column=4)

but4\_5.grid(row=4,column=5)

but5\_1.grid(row=5,column=1)

but5\_2.grid(row=5,column=2)

but5\_3.grid(row=5,column=3)

but5\_4.grid(row=5,column=4)

but5\_5.grid(row=5,column=5)

def game1(t,b):

n=b['text']

b['text']='\*'

l1[l1.index(int(n))]='\*'

confirm1(t,n)

if int(n) in l3:

l3.remove(int(n))

for i in range(len(btlist1)):

btlist1[i]['state']='disabled'

btlist2[i]['state']='normal'

def game2(t,b):

n=b['text']

b['text']='\*'

l2[l2.index(int(n))]='\*'

confirm2(t,n)

if int(n) in l4:

l4.remove(int(n))

for i in range(len(btlist2)):

btlist2[i]['state']='disabled'

btlist1[i]['state']='normal'

def confirm1(t,b):

if l2[l2.index(int(b))]!='\*':

l2[l2.index(int(b))]='\*'

for i in range(len(btlist2)):

if btlist2[i]['text']==b:

btlist2[i]['text']='\*'

if statement(t)==False:

statement(t)

def confirm2(t,b):

if l1[l1.index(int(b))]!='\*':

l1[l1.index(int(b))]='\*'

for i in range(len(btlist1)):

if btlist1[i]['text']==b:

btlist1[i]['text']='\*'

if statement(t)==False:

statement(t)

#checking

def plcheck():

global l1

s=0

if l1[0]==l1[1]==l1[2]==l1[3]==l1[4]:

s+=1

if l1[5]==l1[6]==l1[7]==l1[8]==l1[9]:

s+=1

if l1[10]==l1[11]==l1[12]==l1[13]==l1[14]:

s+=1

if l1[15]==l1[16]==l1[17]==l1[18]==l1[19]:

s+=1

if l1[20]==l1[21]==l1[22]==l1[23]==l1[24]:

s+=1

if l1[0]==l1[5]==l1[10]==l1[15]==l1[20]:

s+=1

if l1[1]==l1[6]==l1[11]==l1[16]==l1[21]:

s+=1

if l1[2]==l1[7]==l1[12]==l1[17]==l1[22]:

s+=1

if l1[3]==l1[8]==l1[13]==l1[18]==l1[23]:

s+=1

if l1[4]==l1[9]==l1[14]==l1[19]==l1[24]:

s+=1

if l1[0]==l1[6]==l1[12]==l1[18]==l1[24]:

s+=1

if l1[4]==l1[8]==l1[12]==l1[16]==l1[20]:

s+=1

if s>=5:

return True

else:

return False

def p2check():

global l2

s=0

if l2[0]==l2[1]==l2[2]==l2[3]==l2[4]:

s+=1

if l2[5]==l2[6]==l2[7]==l2[8]==l2[9]:

s+=1

if l2[10]==l2[11]==l2[12]==l2[13]==l2[14]:

s+=1

if l2[15]==l2[16]==l2[17]==l2[18]==l2[19]:

s+=1

if l2[20]==l2[21]==l2[22]==l2[23]==l2[24]:

s+=1

if l2[0]==l2[5]==l2[10]==l2[15]==l2[20]:

s+=1

if l2[1]==l2[6]==l2[11]==l2[16]==l2[21]:

s+=1

if l2[2]==l2[7]==l2[12]==l2[17]==l2[22]:

s+=1

if l2[3]==l2[8]==l2[13]==l2[18]==l2[23]:

s+=1

if l2[4]==l2[9]==l2[14]==l2[19]==l2[24]:

s+=1

if l2[0]==l2[6]==l2[12]==l2[18]==l2[24]:

s+=1

if l2[4]==l2[8]==l2[12]==l2[16]==l2[20]:

s+=1

if s>=5:

return True

else:

return False

def statement(t):

if plcheck()==True and p2check()==True:

tk=Tk()

tk.geometry('500x500')

tk.title("RESULT")

Label(tk,text='Draw').pack()

Label(tk,text=p1()).pack()

Label(tk,text=p2()).pack()

elif plcheck()==True and p2check()==False:

tk=Tk()

Label(tk,text='Player 1 wins').pack()

Label(tk,text=p1()).pack()

Label(tk,text=p2()).pack()

elif plcheck()==False and p2check()==True:

tk=Tk()

Label(tk,text='Player 2 wins').pack()

Label(tk,text=p2()).pack()

Label(tk,text=p1()).pack()

else:

return False

def p1():

global coml

x=0

s='''player 1

'''

for i in range(5):

s+='| '

for j in range(5):

s+=str(l1[x])+' | '

x+=1

s+='\n'

return s

def p2():

global l

x=0

s='''player 2

'''

for i in range(5):

s+=' | '

for j in range(5):

s+=str(l2[x])+' | '

x+=1

s+='\n'

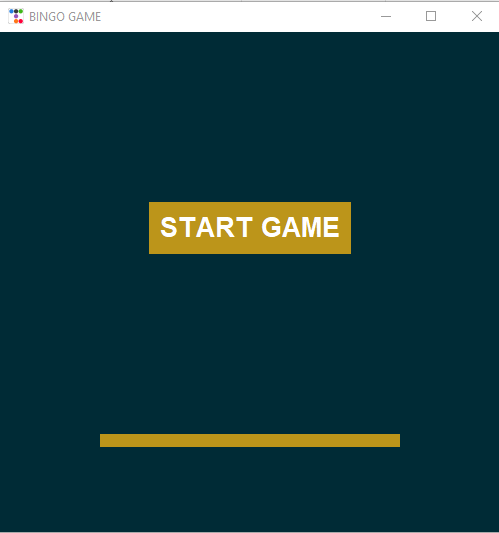
return s

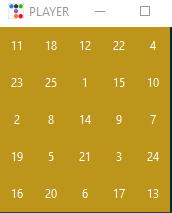
root.mainloop()

**RESULT:**

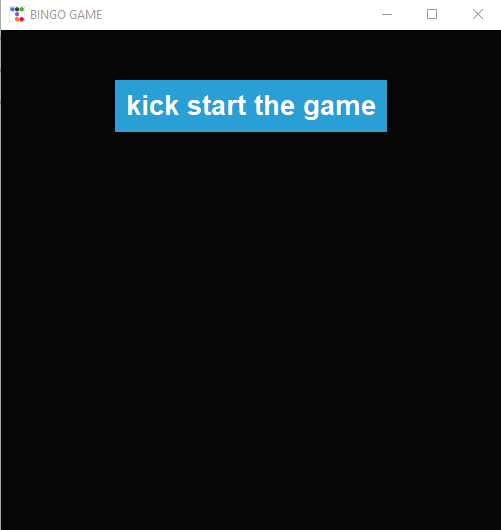
The project was completed and became fully functional on 26/01/2023. The final output of the game and the website is shown below.

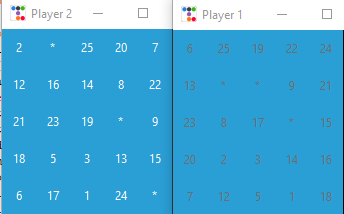
1. **Computer Vs Player mode:**

****



1. **Player 1 Vs Player 2 mode:**





**REFERENCE:**

**1.IMAGES-ONLINE**

**2.OUR WEBSITE:**

[**https://idealisticpelican.static.app/jeromeme**](https://idealisticpelican.static.app/jeromeme)

**3.Bingo Concept-** [**https://youtu.be/0ctKOlgciUA**](https://youtu.be/0ctKOlgciUA)

**4.Tkinder Module-** **https://youtu.be/iVHwTYiYciA**