Abstract

The project titled “ Gameitrator ” is a GUI based application which is written in python programming language .This is a simple GUI based project which is very easy to use and very entertaining also. Talking about the system it contains all the functions to play the game .We have included three GUI based application using python programing . The project file contains the gui based applications which can access through one particular GUI interface.

This project file contains all required function to play games available in the project and know the score .

Gameitrator provides the simple way to reach on a particular GUI based game and play it in a simple word one can access the GUI based applications easily just with a click on button and play, after completion user will get game over message and he can simple quit.

In this project we have included one live match scorer for entertainment perpose .one can know the score of the match by visiting on project.

We have also included database in our project so one can know his previous score and improve it. taking about the requirement it is compulsory to have python installed in user’s system to used this GUI based project.

Acknowledgment

We are very grateful that we managed to complete our "Pyterator Game "project on time this would not be happened without contribution and co-operation from our group member .

We would sincerely thank to Mr.Sagar Pande (Ass.Pro LPU) for his help and support for this project and for his techings
Last but not the least ,we would like to express our gratitude to our friends and responds for the support .

Bharat Garg(11912251)
Shubham Mishra(11912287)
Bikky kumar pandey(11318508)

Content

Introduction

Gameiterator is GUI based project written in python programing language. We created one interface through which we have linked three GUI based applications .two of the three are game which are working with the help of different module/library available in python and third one is score predicator which will work online ,the third application has been made for entertainment purpose and it is also linked with main interface.
Talking about the functionalities, the first GUI based game is ‘KILL CORONA” which is working with the help of pygame and GUI of this game has been created with the help of Tkinter library .

Second one is Tic toe game which is a number game for this game we have used Tkinter for implementation of GUI. Last but not the least the third one is Scorer which is fetching the score online .

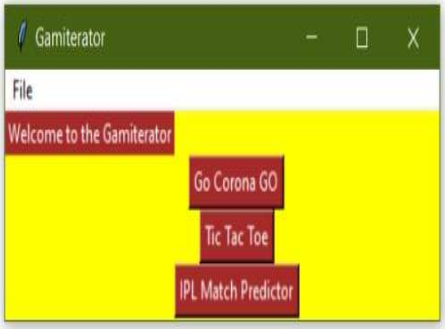
Interface

We have created a user interface to intrect with user .user can access all the three applications available in the GUI based window .

The code for the same is given below…

```
#GUI Building
l0=Label(win,text="Welcome to the Gamiterator",fg="White",bg="Brown"
)
l0.grid(row=2,column=2)
win.geometry("400x100")
win.title("Gamiterator")
win.configure(background="Yellow")

#Button Is Decleared
button=Button(win,text="Go Corona
GO",command=game1,fg="White",bg="Brown")
button2=Button(win,text="Tic Tac
Toe",command=game2,fg="White",bg="Brown")
button3=Button(win,text="IPL Match
Predictor",command=game3,fg="White",bg="Brown")
#Button GUI Drawing
button.grid(row=3,column=3)
button2.grid(row=4,column=3)
button3.grid(row=5,column=3)
#MenuBar Layout Design
```

```
mymenu=Menu(win)
m1=Menu(mymenu,tearoff=0)

#m1.add_command(label="Save",command=foolFunction)
m1.add_command(label="Exit",command=quit)
m1.add_separator()
win.config(menu=mymenu)
mymenu.add_cascade(label="File",menu=m1)
#Last Packing Of the GUI LAyout
win.mainloop()
```


Gamiterator
File
Welcome to the Gamiterator
Go Corona GO
Tic Tac Toe
IPL Match Predictor

Create Game “Go Corona”

Go corona is a gui based game in which corona virus will be died when vaccine drop will touch it .

This game is a gui based game which is implemented with the help of Pygame library and Tkinter library

The source code for same is given below…

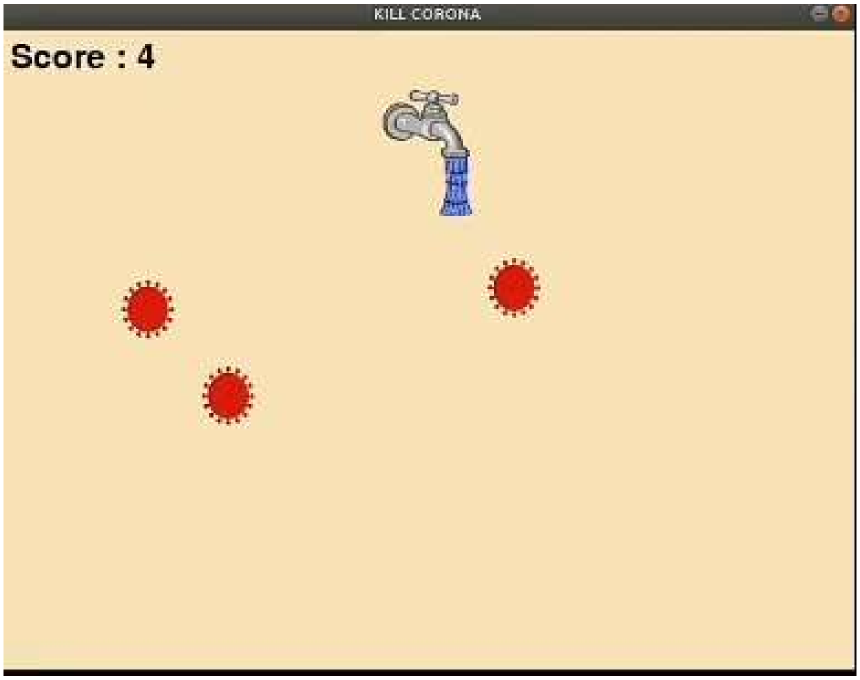
```
import pygame
import random
import math
from pygame import mixer

# initialize the pygame
pygame.init()

# Create the screen
screen = pygame.display.set_mode((800, 600))

# background
#mixer.music.load('background.wav')
#mixer.music.play(-1)


# Title and Icon
pygame.display.set_caption("KILL CORONA")
icon = pygame.image.load('icon.jpeg')
pygame.display.set_icon(icon)

# tap
```

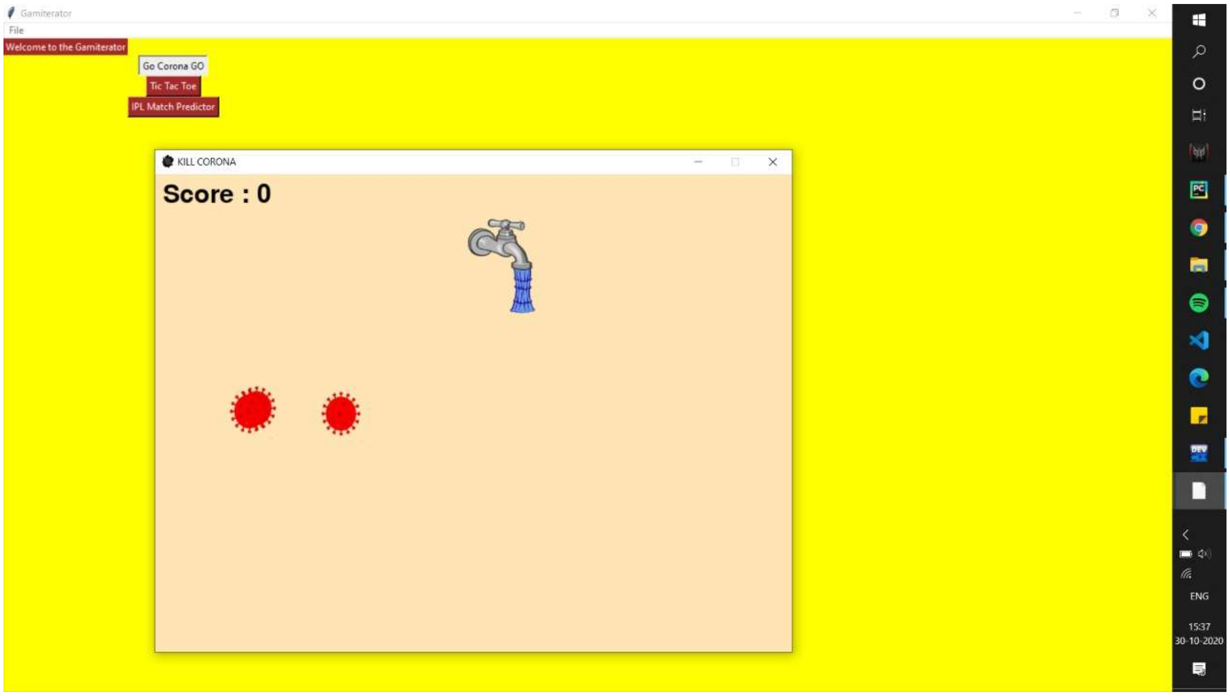
```
tapImg = pygame.image.load('tap.jpeg')
tapX = 370
tapY = 50
tapX_change = 0

# corona
coronaImg = []
coronaX = []
coronaY = []
coronaX_change = []
coronaY_change = []
num_of_corona = 3

for i in range(num_of_corona):
    coronaImg.append(pygame.image.load('corona.jpeg'))
    coronaX.append(random.randint(20, 780))
    coronaY.append(random.randint(200, 220))
    coronaX_change.append(4)
    coronaY_change.append(4)

# drop
dropImg = pygame.image.load('drop.jpeg')
dropX = 0
dropY = 150
dropX_change = 0
dropY_change = 1
drop_state = "ready"  # Ready - No drop on screen

# Score
score_value = 0
font = pygame.font.Font('freesansbold.ttf', 32)
textX = 10
textY = 10
```

```
# Game over
over_text = pygame.font.Font('freesansbold.ttf', 72)


def show_score(x, y):
    score = font.render("Score : " + str(score_value), True, (0, 0, 0))
    screen.blit(score, (x, y))


def game_over_text():
    over_text = font.render("Game over ", True, (0, 0, 0))
    screen.blit(over_text, (300, 250))


def tap(x, y):
    screen.blit(tapImg, (x, y))


def corona(x, y, i):
    screen.blit(coronaImg[i], (x, y))


def fire_drop(x, y):
    global drop_state
    drop_state = "fire"
    screen.blit(dropImg, (x + 16, y + 10))


def isCollision(coronaX, coronaY, dropX, dropY):
    distance = math.sqrt((math.pow(coronaX - dropX, 2)) +
(math.pow(coronaY - dropY, 2)))
    if distance < 27:
        return True


# Game Loop
running = True
while running:
    # Adding colour to screen
    screen.fill((255, 228, 181))
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

        # if keystroke is pressed check whether its right or left
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                tapX_change = -5
            if event.key == pygame.K_RIGHT:
                tapX_change = 5
            if event.key == pygame.K_UP or event.key ==
pygame.K_SPACE:
                if drop_state is "ready":
                    dropX = tapX
                    fire_drop(dropX, dropY)

        if event.type == pygame.KEYUP:
            if event.key == pygame.K_LEFT or event.key ==
pygame.K_RIGHT:
                tapX_change = 0

    # tap movement
    tapX += tapX_change
    if tapX <= 0:
        tapX = 0
```

```
elif tapX >= 690:
    tapX = 690

# corona movement
for i in range(num_of_corona):

    # Game Over
    if coronaY[i] > 400:
        for j in range(num_of_corona):
            coronaY[j] = 2000
        game_over_text()
        break
    coronaX[i] += coronaX_change[i]
    if coronaX[i] <= 0:
        coronaX_change[i] = 1
        coronaY[i] += coronaY_change[i]
    elif coronaX[i] >= 736:
        coronaX_change[i] = -1
        coronaY[i] += coronaY_change[i]
    # Collision
    collision = isCollision(coronaX[i], coronaY[i], dropX, dropY)
    if collision:
        dropY = 20
        drop_state = "ready"
        score_value += 1
        coronaX[i] = random.randint(0, 736)
        coronaY[i] = random.randint(200, 220)
    corona(coronaX[i], coronaY[i], i)

# drop movement
if dropY >= 600:
    dropY = 150
    drop_state = "ready"
```

```
if drop_state is "fire":
    fire_drop(dropX, dropY)
    dropY += dropY_change

tap(tapX, tapY)
show_score(textX, textY)
pygame.display.update()
```


KILL CORONA
Score : 4


Gamiterator
File
Welcome to the Gamiterator
Go Corona GO
Tic Tac Toe
IPL Match Predictor
KILL CORONA
Score : 0
ENG
15:37
30-10-2020

Create the Tic Tac Toe

Tic Tac Toe is a GUI based game in which corona virus will be died when vaccine drop will touch it .

This game is a gui based game which is implemented with the help of Pygame library and Tkinter library

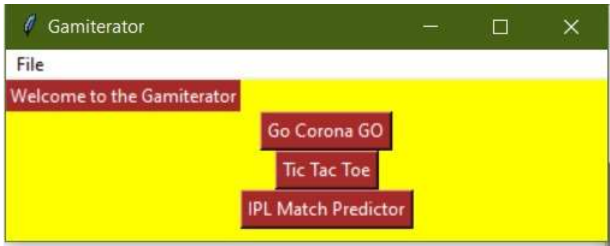
```
from tkinter import *
from tkinter import messagebox
import mysql.connector
a=0
b="ok"
tk1 = Tk()

pa = StringVar()
playerb = StringVar()
p1 = StringVar()
p2 = StringVar()
bclick = True
flag = 0



player1_name = Entry(tk1, textvariable=p1, bd=5)
player1_name.insert(END, 'PLAYER 1')
player1_name.grid(row=1, column=1, columnspan=8)
player2_name = Entry(tk1, textvariable=p2, bd=5)
player2_name.insert(END, 'PLAYER 2')
```

```
player2_name.grid(row=2, column=1, columnspan=8)



def disableButton():
    button1.configure(state=DISABLED)
    button2.configure(state=DISABLED)
    button3.configure(state=DISABLED)
    button4.configure(state=DISABLED)
    button5.configure(state=DISABLED)
    button6.configure(state=DISABLED)
    button7.configure(state=DISABLED)
    button8.configure(state=DISABLED)
    button9.configure(state=DISABLED)



def btnClick(buttons):
    global bclick, flag, player2_name, player1_name, playerb, pa,a,b
    if buttons["text"] == " " and bclick == True:
        buttons["text"] = "X"
        bclick = False
        playerb = p2.get() + " Wins! in " + str(flag) + " moves"
        pa = p1.get() + " Wins! in " + str(flag) + " moves"
        checkForWin()
        flag += 1


    elif buttons["text"] == " " and bclick == False:
        buttons["text"] = "O"
        bclick = True
        checkForWin()
        flag += 1
    else:
        messagebox.showinfo("Tic-Tac-Toe", "Button already
Clicked!")

def checkForWin():
    if (button1['text'] == 'X' and button2['text'] == 'X' and
button3['text'] == 'X' or
        button4['text'] == 'X' and button5['text'] == 'X' and
button6['text'] == 'X' or
        button7['text'] == 'X' and button8['text'] == 'X' and
button9['text'] == 'X' or
        button1['text'] == 'X' and button5['text'] == 'X' and
button9['text'] == 'X' or
        button3['text'] == 'X' and button5['text'] == 'X' and
button7['text'] == 'X' or
        button1['text'] == 'X' and button4['text'] == 'X' and
button7['text'] == 'X' or
        button2['text'] == 'X' and button5['text'] == 'X' and
button8['text'] == 'X' or
        button3['text'] == 'X' and button6['text'] == 'X' and
button9['text'] == 'X'):
        global a,b
        a=1
        b=p1.get()
        disableButton()
        messagebox.showinfo("Tic-Tac-Toe", pa)
        tk1.after(50, tk1.destroy)
    elif(flag == 8):
        messagebox.showinfo("Tic-Tac-Toe", "It is a Tie")
        tk1.after(50, tk1.destroy)
    elif (button1['text'] == 'O' and button2['text'] == 'O' and
button3['text'] == 'O' or
```

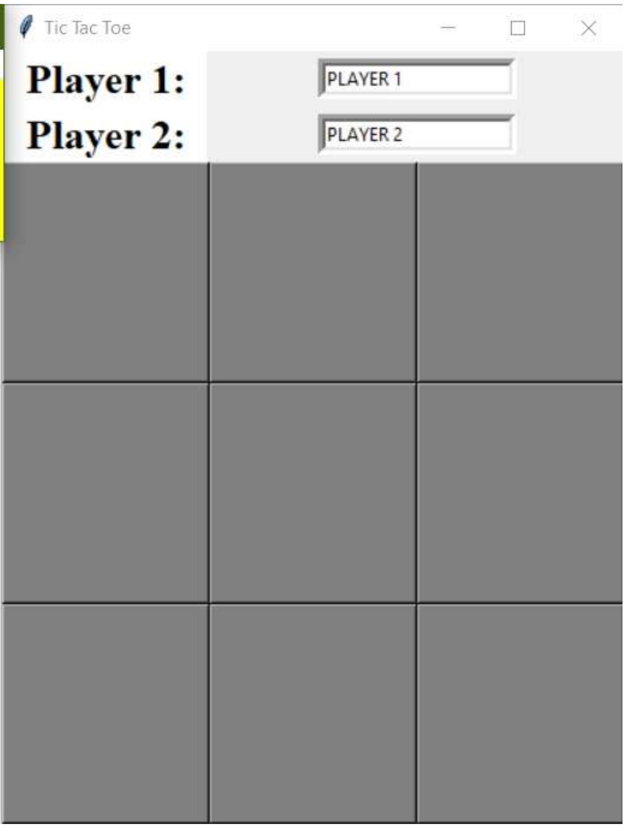
```
        button4['text'] == 'O' and button5['text'] == 'O' and
button6['text'] == 'O' or
        button7['text'] == 'O' and button8['text'] == 'O' and
button9['text'] == 'O' or
        button1['text'] == 'O' and button5['text'] == 'O' and
button9['text'] == 'O' or
        button3['text'] == 'O' and button5['text'] == 'O' and
button7['text'] == 'O' or
        button1['text'] == 'O' and button4['text'] == 'O' and
button7['text'] == 'O' or
        button2['text'] == 'O' and button5['text'] == 'O' and
button8['text'] == 'O' or
        button3['text'] == 'O' and button6['text'] == 'O' and
button9['text'] == 'O'):
        disableButton()
        messagebox.showinfo("Tic-Tac-Toe", playerb)
        tk1.after(50, tk1.destroy)
        a=1
        b=p2.get()



label = Label( tk1, text="Player 1:", font='Times 20 bold',
bg='white', fg='black', height=1, width=8)
label.grid(row=1, column=0)


label = Label( tk1, text="Player 2:", font='Times 20 bold',
bg='white', fg='black', height=1, width=8)
label.grid(row=2, column=0)
```

```
button1 = Button(tk1, text=" ", font='Times 20 bold', bg='gray',
fg='white', height=4, width=8, command=lambda:
btnClick(button1))
button1.grid(row=3, column=0)

button2 = Button(tk1, text=' ', font='Times 20 bold', bg='gray',
fg='white', height=4, width=8, command=lambda:
btnClick(button2))
button2.grid(row=3, column=1)

button3 = Button(tk1, text=' ',font='Times 20 bold', bg='gray',
fg='white', height=4, width=8, command=lambda:
btnClick(button3))
button3.grid(row=3, column=2)

button4 = Button(tk1, text=' ', font='Times 20 bold', bg='gray',
fg='white', height=4, width=8, command=lambda:
btnClick(button4))
button4.grid(row=4, column=0)

button5 = Button(tk1, text=' ', font='Times 20 bold', bg='gray',
fg='white', height=4, width=8, command=lambda:
btnClick(button5))
button5.grid(row=4, column=1)

button6 = Button(tk1, text=' ', font='Times 20 bold', bg='gray',
fg='white', height=4, width=8, command=lambda:
btnClick(button6))
button6.grid(row=4, column=2)

button7 = Button(tk1, text=' ', font='Times 20 bold', bg='gray',
fg='white', height=4, width=8, command=lambda:
btnClick(button7))
```

```
button7.grid(row=5, column=0)

button8 = Button(tk1, text=' ', font='Times 20 bold', bg='gray',
fg='white', height=4, width=8, command=lambda:
btnClick(button8))
button8.grid(row=5, column=1)

button9 = Button(tk1, text=' ', font='Times 20 bold', bg='gray',
fg='white', height=4, width=8, command=lambda:
btnClick(button9))
button9.grid(row=5, column=2)

tk1.title("Tic Tac Toe")
tk1.mainloop()

print(a,b)



if a==1:
    try:
        mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            password="admin",
        )
        mycursor = mydb.cursor()
        sql = "CREATE DATABASE GAME"
        mycursor.execute(sql)
        print("done")
        mydb2 = mysql.connector.connect(
        host="localhost",
        user="root",
        password="admin",
        database="GAME"
        )
        mycursor2 = mydb2.cursor()
        mycursor2.execute("CREATE TABLE OXGAME(SNo
INTEGER AUTO_INCREMENT PRIMARY KEY,PlayerName
VARCHAR(100),Number_of_moves_win int(10) )")
        print("table is created successfully")
        a=0
    except Exception:
        print("database and table already exist")
        a=1

    mydb3 = mysql.connector.connect(
    host="localhost",
    user="root",
    password="admin",
    database="GAME"
    )
    mycursor3 = mydb3.cursor()
    sql = "INSERT INTO OXGAME (  PlayerName ,
Number_of_moves_win  ) VALUES (%s,%s)"
    val =   (b,str(flag))
    mycursor3.execute(sql, val)
    mydb3.commit()
    print(mycursor3.rowcount, "record inserted.")
```



Welcome to the Gamiterator



Player 1: PLAYER 1

Player 2: PLAYER 2

Create Scorer

In this GUI based application we are using pil library .through this we are fetching the score for any match online from any url.

Code for the same has been given below….

```
import tkinter as tk
from PIL import ImageTk,Image
import os
from bs4 import BeautifulSoup  #webscrapping
import urllib.request          #for fetching url
score_page='https://static.cricinfo.com/rss/livescores.xml'
#url for scrap the score
page=urllib.request.urlopen(score_page)          #to open that url
soup=BeautifulSoup(page,'html.parser')                    #intially it will be on
html form to convert it to readible format ,we are pasing it
result=soup.find_all('description')
ls=[]                                   #empt list for live score
for match in result:
   ls.append(match.get_text())

def score():
   T.insert(tk.END,ls)
def clear():
   T.delete(1.0,tk.END)
#GUI work start
```

```
root=tk.Toplevel()
root.geometry('1200x675')

img=ImageTk.PhotoImage(Image.open("matches.jpg"))
panel=tk.Label(root,image=img)
panel.place(x=0,y=0)

T=tk.Text(root)                #text area creation
T.place(x=30,y=250,height=250,width=300)

l=tk.Label(root,text="Live Score",fg="white",bg="black")
l.place(x=30,y=400,height=100,width=300)


b1=tk.Button(root,text="Score",bg="black",fg="red",command=score)
b1.place(x=800,y=200,height=100,width=250)

b2=tk.Button(root,text="Clear",bg="black",fg="red",command=clear)
b2.place(x=800,y=400,height=100,width=100)

root.mainloop()
```

Contribution

We are three in a group and each and every team member contributed their part and helped in others (when needed).
Every group member make a Individual game after that put it in a single surface.
Bharat Garg –Tic Toe
Shubham Mishra –Kill Corona
Vikky kumar Pandey –Score predictor

Start Date	**End Date**	**Project states abd objective**
09 sep 2020	16 sep 2020	Project proposal
24 sep 2020	4 oct 2020	Planning thinking about games
7 oct 2020	17 oct 2020	Start implementing
21 oct 2020	22 oct 2020	Testing
30 oct 2020		Project submission

We have found the planning of this project here which now leads us to completion of the project.

Conclusion

With the completion of this project we learnt new things

1.Now we know much more about game implementation

2.We come to know about the full process like implementation testing etc.

3.learnt co-operation between group members

4.Learnt to complete the task before dead-lines

5.come to know about future of python in gaming.