

MINI ARCADE



Submitted As

COMPUTER SCIENCE PROJECT

CLASS XII, 2023-24

Name: Tirth Patel, Bhargav Makwana, Pravin Thakor

Class: 12th – Science

Guided by: Tarun Baria Sir
(PGT CS)

Subject: Computer Science

School: JNV Hathijan , Ahmedabad

Acknowledgement

We would like to express our special thanks of gratitude to our teacher Tarun Baria Sir as well as our principal Ravindrakumar Dixit Sir who gave us the golden opportunity to do this wonderful project titled **MINI ARCADE**, which also helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to them.

Secondly we would also like to thank our parents and friends who helped me a lot in finalizing this project within the limited time frame.

Certificate

This is to certify that **Tirth Patel, Bhargav Makwana, Pravin Thakor** of class XII from JNV Hathijan ,Ahmedabad who were allowed to do work on this project titled **MINI ARCADE** have faithfully carried out this project under my guidance and supervision and the accompanying report is completely their genuine work. They have completed the necessary preliminaries and have submitted the Project in allotted time.

Teacher Sign

Principal Sign

INTRODUCTION

We have created this “MINI ARCADE” program for entertainment purpose. Whenever you are free and offline you can play these games if you have our program in your pc.

Our Mini Arcade consists of three games and one game in under development phase. Games which you can play are:

- 1.Tic-Tac- Toe: You know this game very well. Difference is you can play this game without wasting your papers.
- 2.Hungary Snake: This game is under development
- 3.Stone paper scissor: You don't need a friend to play it, you can play this with computer.
- 4.Dice: If you don't have a real dice, “Don't Worry”, we have created a dice program for you.

CONTENTS OF THE PROJECT

We have used 4 modules in the software.

(i) tkinter

(ii) pygame

(iii) turtle

(iv) random

(I)**tkinter:** tkinter is used to make all the interfaces of windows used in the project.

(II)**pygame:** pygame is used to make tic-tac-toe.

(III)**turtle:** turtle is used in making of dice.

(IV)**random:** random module is used to make stone paper scissors.

SOURCE CODE

```
import tkinter
from tkinter import messagebox
import pygame as pg
from pygame.locals import *
import turtle
import random
def mainext():
    mainwindow.destroy()
def mainext2():
    mainwindow.destroy()
    extwar.destroy()
def war():
    def dest():
        extwar.destroy()
    global extwar
    extwar = tkinter.Tk()
    extwar.title("Confirm Quit")
    extwar.geometry("250x100")
    extwar.configure(bg="black")
    l2 = tkinter.Label(extwar, text="Are you sure to
quit", font=("Arial", 15), bg="black", fg="white").grid(columnspan=5, column=0, row=0)

    lbl3= tkinter.Label(extwar, text="", bg=z).grid(row=1)
    ok= tkinter.Button(extwar, text="Quit",
command=mainext2, width=7, height=1, bg=y, fg=x).grid(column=2, row=2)
    cancel= tkinter.Button(extwar, text="Cancel",
command=dest, width=7, height=1, bg=y, fg=x).grid(column=3, row=2)
def exths():
    w2.destroy()
    main()
def extrps():
    rpswin.destroy()
    main()
def extdice():
```

```

w4.destroy()
main()
def ttt():
    global board , game, winner
    mainwindow.destroy()
    pg.init()

    #defining nessasary variable value
    pos=[]
    markers=[]
    width = 600
    height = 700
    line_width = 5
    clicked = False
    player = 1
    winner = 0
    game = False
    Blue = (0,0,255)
    Red = (255,0,0)
    Green = (0,255,0)
    Turquoise=(10,171,222)
    text_font = pg.font.SysFont("freesansbold.ttf",50)

    #create empty 3 x 3 list to represent the grid
    for x in range(3):
        row = [0]*3
        markers.append(row)

    #creating interface
    board=pg.display.set_mode((width,height)) #defining display size
    pg.display.set_caption("Tic-Tac-Toe")# defining title

    #creating rectangle for play again button
    Rect = pg.draw.rect(board,Green,(360,620,200,60))

    #Drawing grid lines
    def grid():
        B = (0,0,0)
        grid_color = (255,255,255)
        board.fill(B)
        pg.draw.line(board,grid_color, (200,0), (200,600),line_width)
        pg.draw.line(board,grid_color, (400,0), (400,600),line_width)
        pg.draw.line(board,grid_color, (0,200), (600,200),line_width)
        pg.draw.line(board,grid_color, (0,400), (600,400),line_width)
        pg.draw.line(board,grid_color, (0,600), (600,600),line_width)
        pg.draw.line(board,grid_color, (0,0), (600,0),line_width)

```

```

pg.draw.line(board,grid_color, (0,0), (0,700),line_width)
pg.draw.line(board,grid_color, (600,0), (600,700),line_width)
pg.draw.line(board,grid_color,(0,700),(700,700),line_width)

#drawing player symbol wherever they click
def draw_markers():
    x_pos=0
    for x in markers:
        y_pos=0
        for y in x:
            if y==1:

pg.draw.line(board,Blue,(x_pos*200+20,y_pos*200+20),(x_pos*200+180,y_pos*200+180)
,line_width)

pg.draw.line(board,Blue,(x_pos*200+180,y_pos*200+20),(x_pos*200+20,y_pos*200+180)
,line_width)
            if y== -1:

pg.draw.circle(board,Red,(x_pos*200+100,y_pos*200+100),90,line_width)
            y_pos=y_pos+1
            x_pos=x_pos+1


#Checking who is winner and drawing line in row,column or cross
def winner_check():
    global winner
    global game
    #Row 1
    if markers[0][0]+markers[1][0]+markers[2][0]==3 :
        winner=1
        game=True
        pg.draw.line(board,Green,(15,100),(585,100),line_width)

    elif markers[0][0]+markers[1][0]+markers[2][0]==-3 :
        winner=2
        game=True
        pg.draw.line(board,Green,(15,100),(585,100),line_width)

    #Row 2
    elif markers[0][1]+markers[1][1]+markers[2][1]==3:
        winner = 1
        game=True
        pg.draw.line(board,Green,(15,300),(585,300),line_width)

    elif markers[0][1]+markers[1][1]+markers[2][1]==-3:
        winner = 2
        game=True

```



```

        pg.draw.line(board,Green,(15,300),(585,300),line_width)
#Row 3
elif markers[0][2]+markers[1][2]+markers[2][2]==3:
    winner = 1
    game = True
    pg.draw.line(board,Green,(15,500),(585,500),line_width)

elif markers[0][2]+markers[1][2]+markers[2][2]==-3:
    winner = 2
    game = True
    pg.draw.line(board,Green,(15,500),(585,500),line_width)
#Column1
elif markers[0][0]+markers[0][1]+markers[0][2]==3:
    winner= 1
    game = True
    pg.draw.line(board,Green,(100,15),(100,585),line_width)

elif markers[0][0]+markers[0][1]+markers[0][2]==-3:
    winner= 2
    game = True
    pg.draw.line(board,Green,(100,15),(100,585),line_width)
#Column2
elif markers[1][0]+markers[1][1]+markers[1][2]==3:
    winner=1
    game=True
    pg.draw.line(board,Green,(300,15),(300,585),line_width)

elif markers[1][0]+markers[1][1]+markers[1][2]==-3:
    winner=2
    game=True
    pg.draw.line(board,Green,(300,15),(300,585),line_width)
#Column3
elif markers[2][0]+markers[2][1]+markers[2][2]==3:
    winner= 1
    game= True
    pg.draw.line(board,Green,(500,15),(500,585),line_width)

elif markers[2][0]+markers[2][1]+markers[2][2]==-3:
    winner= 2
    game= True
    pg.draw.line(board,Green,(500,15),(500,585),line_width)

#cross left to right
elif markers[0][0]+markers[1][1]+markers[2][2]==3:
    winner = 1
    game = True
    pg.draw.line(board,Green,(15,15),(585,585),line_width)

```

```

elif markers[0][0]+markers[1][1]+markers[2][2]==-3:
    winner = 2
    game = True
    pg.draw.line(board,Green,(15,15),(585,585),line_width)

#cross right to left
elif markers[0][2]+markers[1][1]+markers[2][0]==3:
    winner = 1
    game = True
    pg.draw.line(board,Green,(585,15),(15,585),line_width)

elif markers[0][2]+markers[1][1]+markers[2][0]==-3:
    winner = 2
    game = True
    pg.draw.line(board,Green,(585,15),(15,585),line_width)

#check for tie
elif
markers[0][0]+markers[1][0]+markers[2][0]+markers[0][1]+markers[1][1]+markers[2][1]+markers[0][2]+markers[1][2]+markers[2][2]==1:
    if (markers[0][0] == 1 or markers[0][0] == -1) and\
        (markers[1][0]==1 or markers[1][0] ==-1) and\
        (markers[2][0]==1 or markers[2][0]==-1) and\
        (markers[0][1]==1 or markers[0][1]==-1) and\
        (markers[1][1]==1 or markers[1][1]==-1) and\
        (markers[2][1]==1 or markers[2][1]==-1) and\
        (markers[0][2]==1 or markers[0][2]==-1) and\
        (markers[1][2]==1 or markers[1][2]==-1)
and\
        (markers[2][2]==1 or markers[2][2]==-
1):
    winner = 3
    game = True

#displaying who is winner text and play again text when game is over
def winner_text():

    if winner==1:
        text=text_font.render("Winner is Player 1!",True,Turquoise)
        board.blit(text,(30,630))

    if winner ==2:
        text=text_font.render("Winner is Player 2!",True,Turquoise)
        board.blit(text,(30,630))

    if winner == 3:
        text=text_font.render("You have Tied!",True,Turquoise)
        board.blit(text,(30,630))

```

```

if game == True:
    pg.draw.rect(board,Green,(370,620,200,60))
    restart = text_font.render("Play Again",True,Red)
    board.blit(restart,(380,630))

#Rendering interface
run = True
while run:

    grid()
    draw_markers()
    winner_text()
    winner_check()
    for event in pg.event.get():

        if event.type == pg.QUIT:
            pg.quit()
            main()

            run=False

#changing player at every click and updating markers
if game == False:

    #check for mouse click
    if event.type == MOUSEBUTTONDOWN and clicked == False:
        clicked = True
    if event.type == MOUSEBUTTONUP and clicked == True:
        clicked == False

    #taking click position
    pos = pg.mouse.get_pos()
    cell_x= pos[0]
    cell_y= pos[1]
    if markers[cell_x//200][cell_y//200]==0:
        markers[cell_x//200][cell_y//200]=player

    #switching player
    player = player*-1

#creating a "play again" button
if game==True:

    #checking mouse click
    if event.type == pg.MOUSEBUTTONDOWN and clicked == False:

```

```

        clicked = True
    if event.type == pg.MOUSEBUTTONUP and clicked == True:
        clicked = False

    #taking mouse click position
    pos = pg.mouse.get_pos()

    #checking if the click is done on play again button
    if Rect.collidepoint(pos):
        #resetting variables when play again button is clicked
        game = False
        player = 1
        pos = []
        markers = []
        winner = 0
        #create empty 3 x 3 list to represent the grid
        for x in range (3):
            row = [0] * 3
            markers.append(row)

        #updating display
        pg.display.update()
    #exiting pygame once the program is executed
def hs():
    global w2
    mainwindow.destroy()
    w2 = tkinter.Tk()
    w2.title("Hungry snack!")
    w2.geometry("550x400")
    w2.configure(bg="black")
    l2 = tkinter.Label(w2, text="Hungry snake",bg="black",fg="light
green",font=("Arial bold",25)).pack()
    l3 = tkinter.Label(w2, text="(Game Under development)",bg="black",fg="light
green",font=("Arial bold",25)).pack()
    l4 = tkinter.Label(w2, text="",bg="black",fg="light green",font=("Arial
bold",25)).pack()
    l5 = tkinter.Label(w2, text="",bg="black",fg="light green",font=("Arial
bold",25)).pack()
    ext= tkinter.Button(w2, text="Quit", command=exths, font=("Arial
bold",13),bg=y,fg=x,height=2 , width=15).pack()
def rps():
    global rpswin
    mainwindow.destroy()
    opt = ['Rock', 'Paper', 'Scissor']
    b = "black"
    w = "white"
    m = "maroon"
    def rock():
        comp = random.choice(opt)

```

```

        if comp == 'Rock':
            l = tkinter.Label(rpswin, text="        That's a tie.
",bg=b,fg=w,font=("Arial bold",17)).grid(row=6,column=3)
        elif comp == 'Paper':
            l = tkinter.Label(rpswin, text="        Oops! You lose.
",bg=b,fg=w,font=("Arial bold",17)).grid(row=6,column=3)
        else:
            l = tkinter.Label(rpswin, text="Congrats! You
won",bg=b,fg=w,font=("Arial bold",17)).grid(row=6,column=3)
    def paper():
        comp = random.choice(opt)
        if comp == 'Rock':
            l = tkinter.Label(rpswin, text="Congrats! You
won",bg=b,fg=w,font=("Arial bold",17)).grid(row=6,column=3)
        elif comp == 'Paper':
            l = tkinter.Label(rpswin, text="        That's a tie.
",bg=b,fg=w,font=("Arial bold",17)).grid(row=6,column=3)
        else:
            l = tkinter.Label(rpswin, text="        Oops! You lose.
",bg=b,fg=w,font=("Arial bold",17)).grid(row=6,column=3)
    def scissor():
        comp = random.choice(opt)
        if comp == 'Rock':
            l = tkinter.Label(rpswin, text="        Oops! You lose.
",bg=b,fg=w,font=("Arial bold",17)).grid(row=6,column=3)
        elif comp == 'Paper':
            l = tkinter.Label(rpswin, text="Congrats! You
won.",bg=b,fg=w,font=("Arial bold",17)).grid(row=6,column=3)
        else:
            l = tkinter.Label(rpswin, text="        That's a tie.
",bg=b,fg=w,font=("Arial bold",17)).grid(row=6,column=3)
    rpswin = tkinter.Tk()
    rpswin.geometry("650x400")
    rpswin.title("Rock-Paper-Scissor")
    rpswin.configure(bg = b)
    lbl00= tkinter.Label(rpswin,text="",bg=b).grid(row=0)
    lbl01= tkinter.Label(rpswin,text="",bg=b).grid(row=1)
    lbl1= tkinter.Label(rpswin,text="        ",bg=b,font=("Arial
bold",25)).grid(column=0,row=2)
    rockbtn = tkinter.Button(rpswin, text="Rock",font=("Arial
bold",13),command=rock,bg=m,fg=w,height=2 , width=15).grid(column=1,row=2)
    lbl2= tkinter.Label(rpswin,text="",bg=b).grid(column=2,row=2)
    paperbtn = tkinter.Button(rpswin, text="Paper",font=("Arial
bold",13),command=paper,bg=m,fg=w,height=2 , width=15).grid(column=3,row=2)
    lbl3= tkinter.Label(rpswin,text="",bg=b).grid(column=4,row=2)
    scissorbtn = tkinter.Button(rpswin, text="Scissor",font=("Arial
bold",13),command=scissor,bg=m,fg=w,height=2 , width=15).grid(column=5,row=2)
    lbl4= tkinter.Label(rpswin,text="",bg=b).grid(row=3)

```

```

lbl5= tkinter.Label(rpswin,text="",bg=b).grid(row=4)
lbl6= tkinter.Label(rpswin,text="",bg=b).grid(row=5)
lbl6= tkinter.Label(rpswin,text="",bg=b).grid(row=7)
lbl6= tkinter.Label(rpswin,text="",bg=b).grid(row=8)
ext= tkinter.Button(rpswin, text="Quit", command=extrps,font=("Arial
bold",13),bg=m,fg=w,height=2 , width=15).place(x=235,y=300)
rpswin.mainloop()
def subdice():
    global dicewin
    dicewin = turtle.Screen()
    dicewin.setup(width=300, height=300)
    dicewin.bgcolor('black')
    dicewin.title('Dice Play')
    turtle.setpos(0,-120)
    turtle.color("white")
    ws=turtle.write("Click spcebar to roll dice" ,
move=True,align='center',font=('Arial',10))
    dicewin.tracer(1)
    turtle.hideturtle()
    dot_positions = [[(0, 0, 'red'), (-100, 100, 'black'), (-100, 0, 'black'), (-
100, -100, 'black'), (100, 100, 'black'), (100, 0, 'black'), (100, -100,
'black')],
                    [(0, 0, 'black'), (-100, 100, 'red'), (-100, 0, 'black'), (-
100, -100, 'black'), (100, 100, 'black'), (100, 0, 'black'), (100, -100, 'red')],
                    [(0, 0, 'red'), (-100, 100, 'red'), (-100, 0, 'black'), (-
100, -100, 'black'), (100, 100, 'black'), (100, 0, 'black'), (100, -100, 'red')],
                    [(0, 0, 'black'), (-100, 100, 'red'), (-100, 0, 'black'), (-
100, -100, 'red'), (100, 100, 'red'), (100, 0, 'black'), (100, -100, 'red')],
                    [(0, 0, 'red'), (-100, 100, 'red'), (-100, 0, 'black'), (-
100, -100, 'red'), (100, 100, 'red'), (100, 0, 'black'), (100, -100, 'red')],
                    [(0, 0, 'black'), (-100, 100, 'red'), (-100, 0, 'red'), (-
100, -100, 'red'), (100, 100, 'red'), (100, 0, 'red'), (100, -100, 'red')]]
    dot = [turtle.Turtle() for _ in range(7)]
    def click():
        global num
        num = random.randint(1, 6)
        for i in range(7):
            dot[i].shape('circle')
            dot[i].color(dot_positions[num-1][i][2])
            dot[i].penup()
            dot[i].goto(dot_positions[num-1][i][0], dot_positions[num-1][i][1])
            dot[i].dot()
    dicewin.listen()
    dicewin.onkeypress(click, 'space')
    dicewin.mainloop()
def dice():
    global w4
    mainwindow.destroy()

```

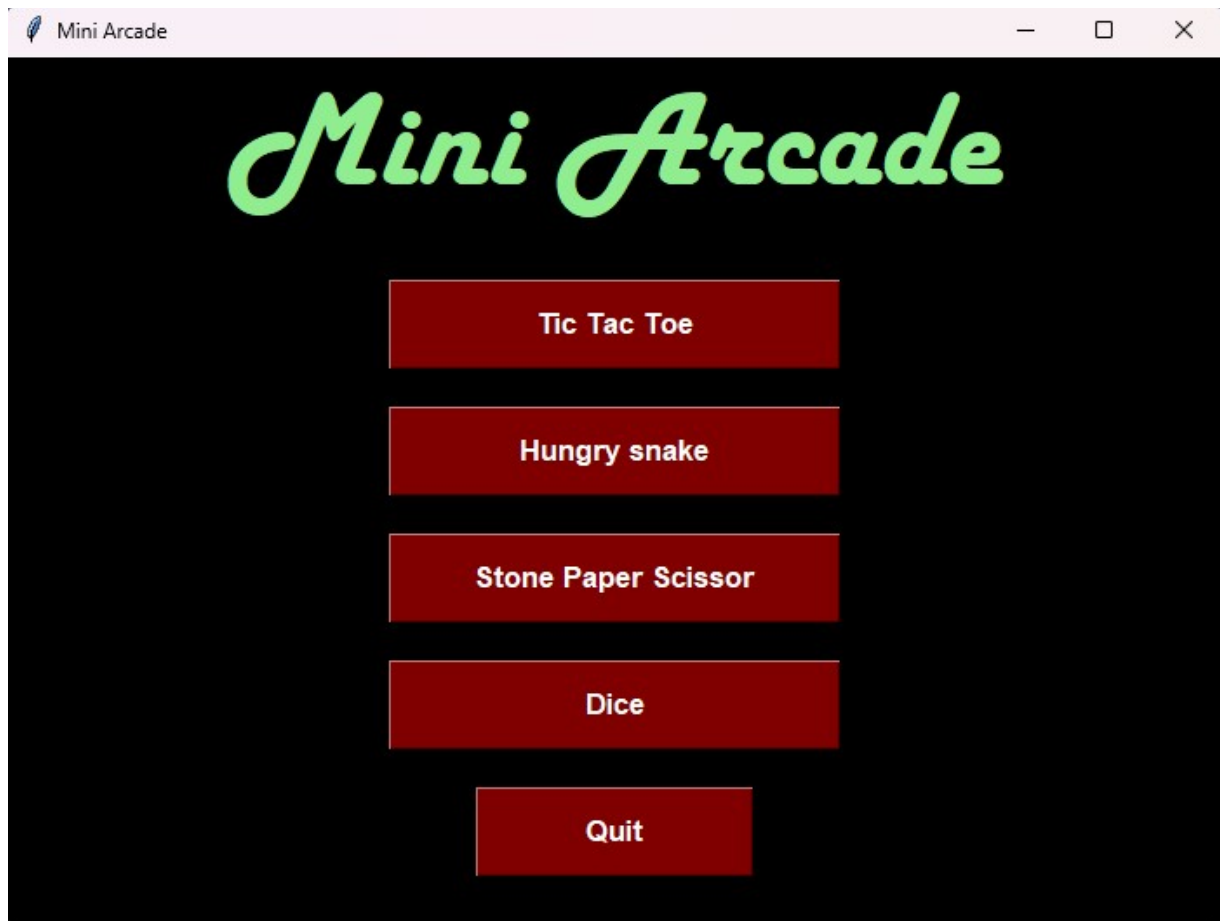
```

w4 = tkinter.Tk()
w4.title("Dice")
w4.geometry("550x400")
w4.configure(bg=z)
l4= tkinter.Label(w4, text="Dice",font=("Harlow Solid
Italic",70),bg=z,fg="lightgreen").pack(fill="both")
lbl1= tkinter.Label(w4,text="",bg=z).pack()
btndice = tkinter.Button(w4, text="Roll dice", font=("Arial
bold",13),bg=y,fg=x,command=subdice,height=2 , width=25).pack(fill="y")
lbl2= tkinter.Label(w4,text="",bg=z).pack()
ext= tkinter.Button(w4, text="Quit", font=("Arial bold",13),bg=y,fg=x,
command=extdice, height=2 , width=25).pack(fill="y")
def main():
    global mainwindow,x,y,z
    x="white"
    y="maroon"
    z="black"
    mainwindow = tkinter.Tk()
    mainwindow.title("Mini Arcade")
    mainwindow.geometry("700x500")
    mainwindow.configure(bg=z)
    mainlable = tkinter.Label(mainwindow, text = "Mini Arcade",font=("Harlow
Solid Italic",60),fg="lightgreen",bg=z).pack(fill="both")
    lbl0= tkinter.Label(mainwindow,text="",bg=z).pack(fill="both")
    b1=tkinter.Button(mainwindow, text="Tic Tac Toe",font=("Arial
bold",13),bg=y,fg=x,height=2 , width=25,command=ttt).pack(fill="y")
    lbl1= tkinter.Label(mainwindow,text="",bg=z).pack(fill="both")
    b2=tkinter.Button(mainwindow, text="Hungry snake",font=("Arial
bold",13),bg=y,fg=x,command=hs,height=2 , width=25).pack(fill="y")
    lbl2= tkinter.Label(mainwindow,text="",bg=z).pack(fill="both")
    b3=tkinter.Button(mainwindow, text="Rock Paper Scissor",font=("Arial
bold",13),bg=y,fg=x,command=rps,height=2 , width=25).pack(fill="y")
    lbl3= tkinter.Label(mainwindow,text="",bg=z).pack(fill="both")
    b4=tkinter.Button(mainwindow, text="Dice",font=("Arial
bold",13),bg=y,fg=x,command=dice,height=2 , width=25).pack()
    lbl4= tkinter.Label(mainwindow,text="",bg=z).pack(fill="both")
    bext = tkinter.Button(mainwindow, text="Quit", font=("Arial
bold",13),bg=y,fg=x,command=war,height=2 , width=15).pack()
    mainwindow.mainloop()
main()

```

OUTPUT

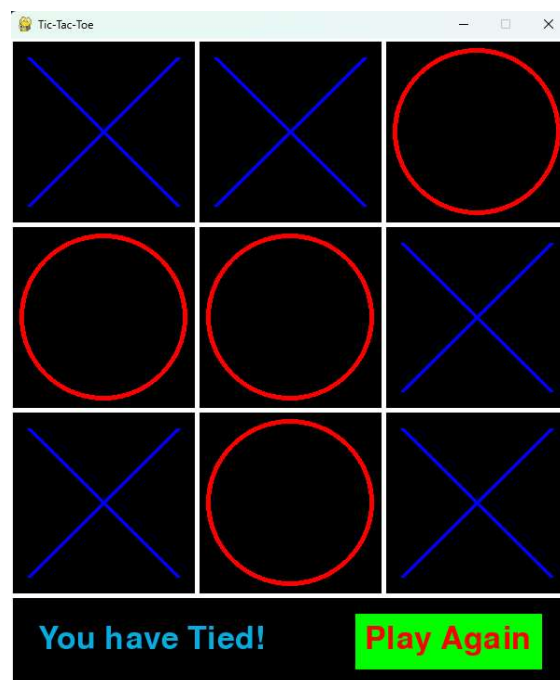
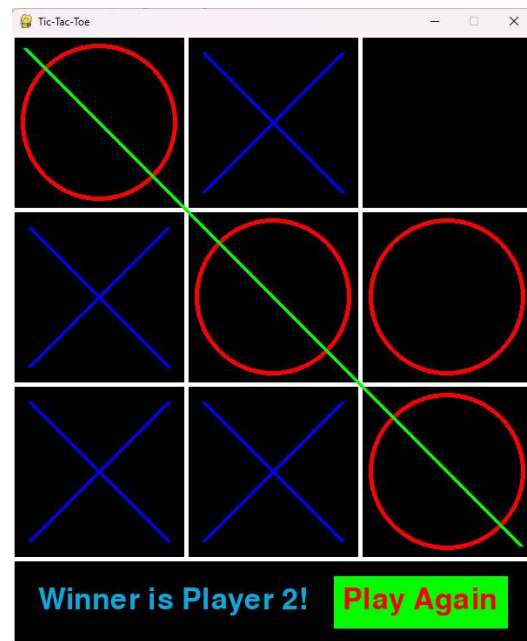
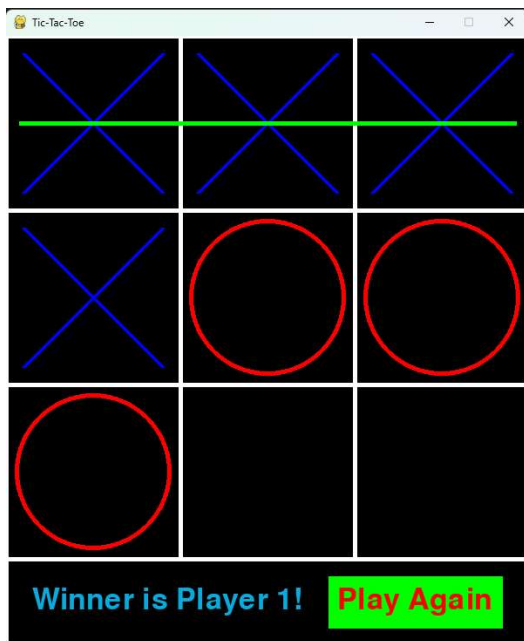
On running the program



On clicking quit button a small window will pop up
Asking permission to close the game.

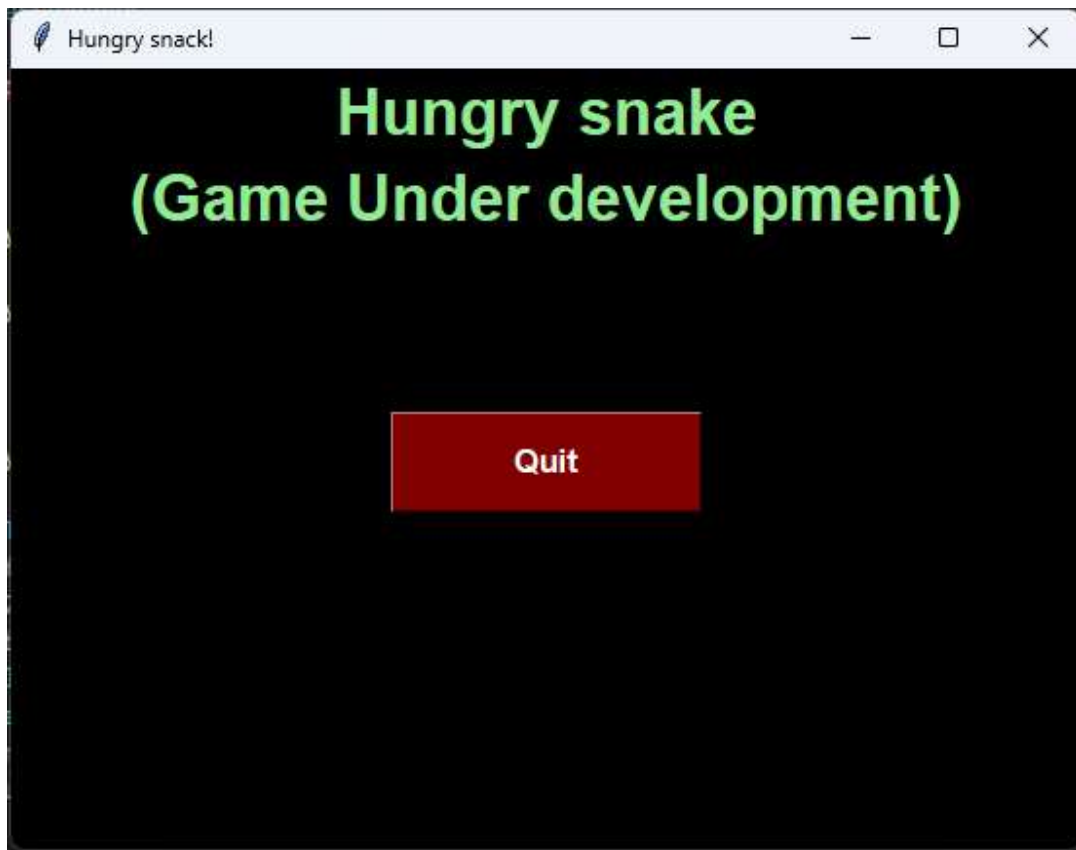


On clicking the tic-tac-toe button and playing it



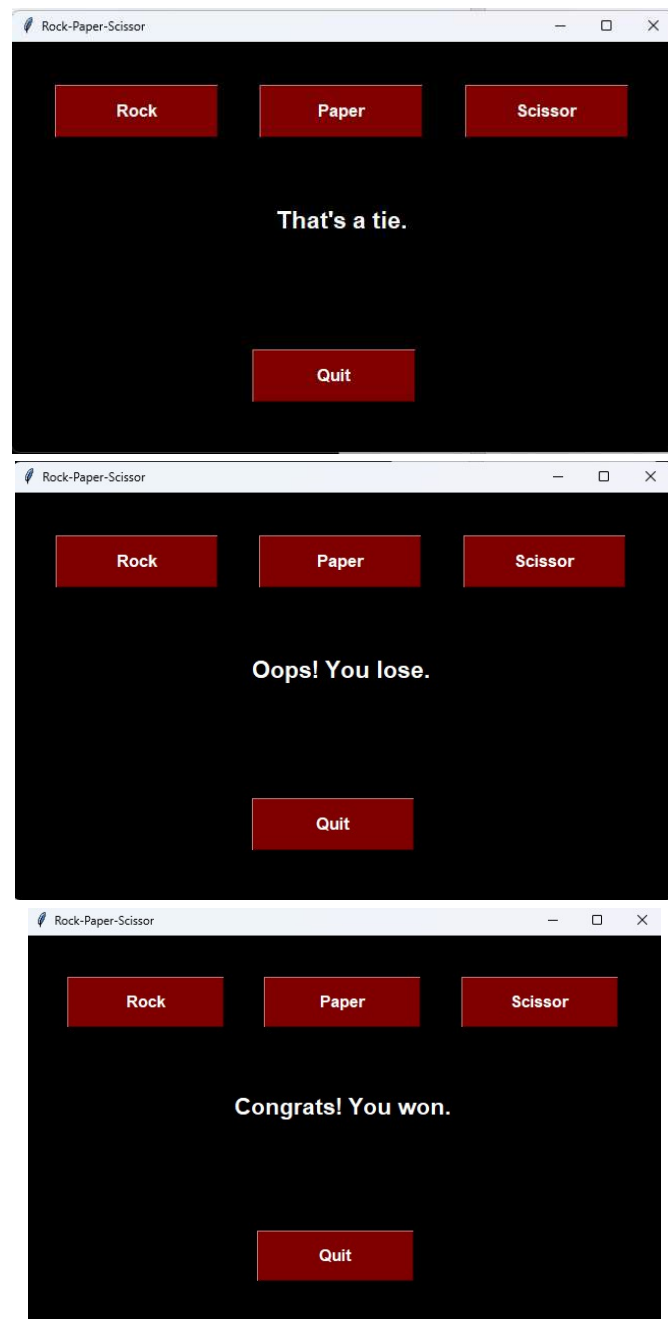
On closing this window, main window will reopen

On clicking hungry snake following window will pop up



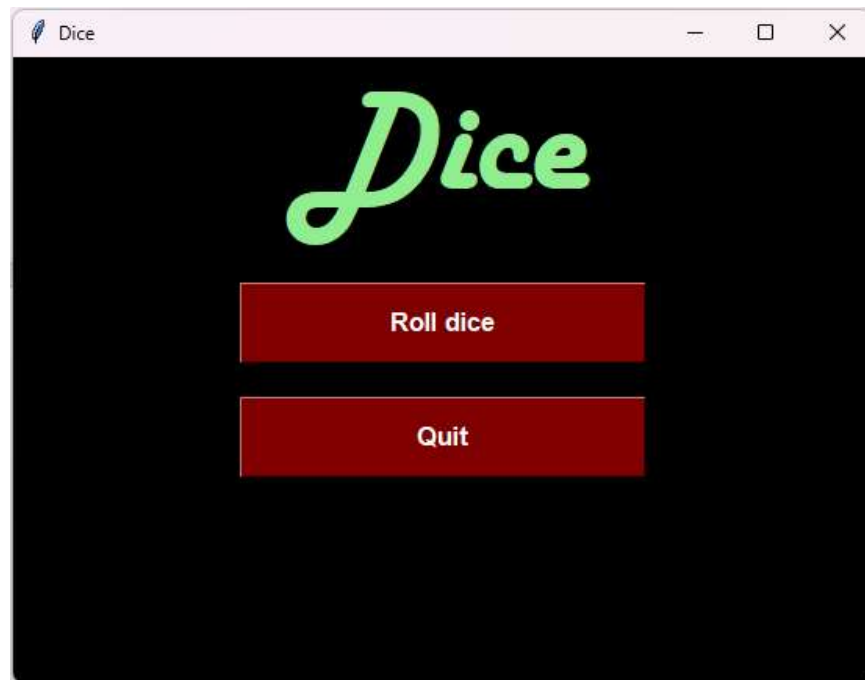
On clicking quit button, main window will reopen

On clicking rock paper scissor, following window will pop up



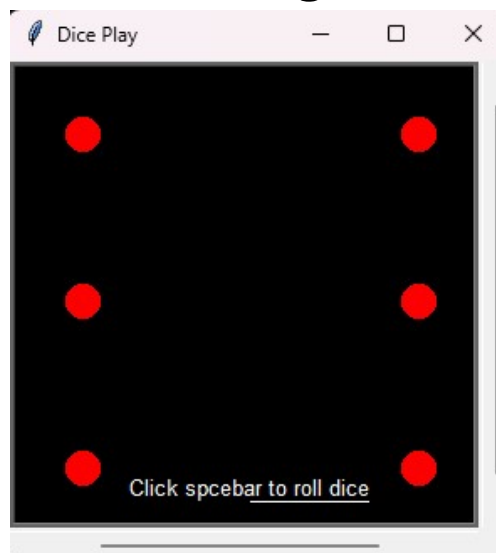
On clicking quit button main window will reopen.

On clicking dice button this window will pop up



On clicking quit button main window will reopen.

On clicking roll dice, following window will pop up



On closing this window previous window will pop up

Scope of the project

We have plans for the future scope of the project.

We are planning to add more games but because of the limitation of time we are not able to do much on the project. But we will try to modify this project in future.

Bibliography

- **stackoverflow.com**
- **w3schools.com**
- **python.com**
- **geeksforgeek.com**
- **CS class 12 Python (Sumita Arora)**

*Thank
you!*