 Language used :- Python,Tkinter

**Project : Arcade Games**

Made by:-

Ayush Gautam(UE213023)

Ananya Trinach(UE213007)

Akshat Verma(UE213005)

Gitansh Sehgal(UE213037)

Ritish Bhatt(UE213077)

**Source Code**

from tkinter import \*

from PIL import ImageTk, Image

root = Tk()

root.title("UIET ARCADE")

root.geometry("650x535")

label=Label(root,text="WELCOME TO UIET ARCADE ",font="Arial 20 bold")

label.pack(side=TOP)

root.maxsize(650,535)

def guess():

    # Imports

    from tkinter import messagebox

    from random import randint

    # Screen

    root = Tk()

    root.geometry("500x500")

    root.title("Number Guessing Game")

    # Generate Number Function

    def GenerateNumberFunc():

        global Number

        # Generate Number

        Number = randint(1, 100)

        # MessageBox to show that a number was generated

        messagebox.showinfo("A Number was Generated!", "Please Guess the Number")

    # Guess Number Function

    def GuessNumberFunc():

        global Number

        # Get Value from Answer Entry Box

        UserResponse = AnswerEntry.get()

        # Convert Value from Answer Entry Box to a Number

        UserResponse = int(UserResponse)

        # Check if the User Response was higher, lower, or equal to the correct number

        if UserResponse > Number:

            ResultLabel.config(text="Incorrect! Please Guess Lower", fg="Red")

        elif UserResponse < Number:

            ResultLabel.config(text="Incorrect! Please Guess Higher", fg="Red")

        else:

            ResultLabel.config(text="You Guess Correctly! The Number was {}".format(Number), fg="Green")

            AnswerEntry.delete(0, "end")

    # Title

    Title = Label(root, text="Number Guessing Game", font=("Arial", 30))

    Title.pack()

    # Main Frame

    MainFrame = Frame(root)

    MainFrame.pack(pady=60)

    # Guess the Number Label

    GuessNumLabel = Label(MainFrame, text="Guess a number from 1 to 100:", font=("Arial", 20))

    GuessNumLabel.pack()

    # Answer Entry

    AnswerEntry = Entry(MainFrame, font=("Arial", 16))

    AnswerEntry.pack(pady=10)

    # Generate Number Button

    GenerateNumberBtn = Button(MainFrame, text="Generate Number", width=16, font=("Arial", 16), background="Dodgerblue",

                               command=GenerateNumberFunc)

    GenerateNumberBtn.pack()

    # Guess Button

    GuessBtn = Button(MainFrame, text="Guess", width=16, font=("Arial", 16), background="#15e650",

                      command=GuessNumberFunc)

    GuessBtn.pack(pady=5)

    # Result Label

    ResultLabel = Label(MainFrame, text="", font=("Arial", 16))

    ResultLabel.pack()

    # Mainloop

    root.mainloop()

def Tic():

    import random

    global player

    def next\_turn(row, column):

        global player

        if buttons[row][column]['text'] == "" and check\_winner() is False:

            if player == players[0]:

                buttons[row][column]['text'] = player

                if check\_winner() is False:

                    player = players[1]

                    label.config(text=(players[1] + " turn"))

                elif check\_winner() is True:

                    label.config(text=(players[0] + " wins"))

                elif check\_winner() == "Tie":

                    label.config(text="Tie!")

            else:

                buttons[row][column]['text'] = player

                if check\_winner() is False:

                    player = players[0]

                    label.config(text=(players[0] + " turn"))

                elif check\_winner() is True:

                    label.config(text=(players[1] + " wins"))

                elif check\_winner() == "Tie":

                    label.config(text="Tie!")

    def check\_winner():

        for row in range(3):

            if buttons[row][0]['text'] == buttons[row][1]['text'] == buttons[row][2]['text'] != "":

                buttons[row][0].config(bg="green")

                buttons[row][1].config(bg="green")

                buttons[row][2].config(bg="green")

                return True

        for column in range(3):

            if buttons[0][column]['text'] == buttons[1][column]['text'] == buttons[2][column]['text'] != "":

                buttons[0][column].config(bg="green")

                buttons[1][column].config(bg="green")

                buttons[2][column].config(bg="green")

                return True

        if buttons[0][0]['text'] == buttons[1][1]['text'] == buttons[2][2]['text'] != "":

            buttons[0][0].config(bg="green")

            buttons[1][1].config(bg="green")

            buttons[2][2].config(bg="green")

            return True

        elif buttons[0][2]['text'] == buttons[1][1]['text'] == buttons[2][0]['text'] != "":

            buttons[0][2].config(bg="green")

            buttons[1][1].config(bg="green")

            buttons[2][0].config(bg="green")

            return True

        elif empty\_spaces() is False:

            for row in range(3):

                for column in range(3):

                    buttons[row][column].config(bg="yellow")

            return "Tie"

        else:

            return False

    def empty\_spaces():

        spaces = 9

        for row in range(3):

            for column in range(3):

                if buttons[row][column]['text'] != "":

                    spaces -= 1

        if spaces == 0:

            return False

        else:

            return True

    def new\_game():

        global player

        player = random.choice(players)

        label.config(text=player + " turn")

        for row in range(3):

            for column in range(3):

                buttons[row][column].config(text="", bg="#F0F0F0")

    window = Tk()

    window.title("Tic-Tac-Toe")

    players = ["x", "o"]

    player = random.choice(players)

    buttons = [[0, 0, 0],

               [0, 0, 0],

               [0, 0, 0]]

    label = Label(text=player + " turn", font=('consolas', 40))

    label.pack(side="top")

    reset\_button = Button(text="restart", font=('consolas', 20), command=new\_game)

    reset\_button.pack(side="top")

    frame = Frame(window)

    frame.pack()

    for row in range(3):

        for column in range(3):

            buttons[row][column] = Button(frame, text="", font=('consolas', 40), width=5, height=2,

                                          command=lambda row=row, column=column: next\_turn(row, column))

            buttons[row][column].grid(row=row, column=column)

    window.mainloop()

def snake():

    import random

    GAME\_WIDTH = 600

    GAME\_HEIGHT = 600

    SPEED = 90

    SPACE\_SIZE = 20

    BODY\_PARTS = 3

    SNAKE\_COLOR = "#00FF00"

    FOOD\_COLOR = "#FF0000"

    BACKGROUND\_COLOR = "#000000"

    class Snake:

        def \_init\_(self):

            self.body\_size = BODY\_PARTS

            self.coordinates = []

            self.squares = []

            for i in range(0, BODY\_PARTS):

                self.coordinates.append([0, 0])

            for x, y in self.coordinates:

                square = canvas.create\_rectangle(x, y, x + SPACE\_SIZE, y + SPACE\_SIZE, fill=SNAKE\_COLOR, tag="snake")

                self.squares.append(square)

    class Food:

        def \_init\_(self):

            x = random.randint(0, (GAME\_WIDTH / SPACE\_SIZE) - 1) \* SPACE\_SIZE

            y = random.randint(0, (GAME\_HEIGHT / SPACE\_SIZE) - 1) \* SPACE\_SIZE

            self.coordinates = [x, y]

            canvas.create\_oval(x, y, x + SPACE\_SIZE, y + SPACE\_SIZE, fill=FOOD\_COLOR, tag="food")

    def next\_turn(snake, food):

        x, y = snake.coordinates[0]

        if direction == "up":

            y -= SPACE\_SIZE

        elif direction == "down":

            y += SPACE\_SIZE

        elif direction == "left":

            x -= SPACE\_SIZE

        elif direction == "right":

            x += SPACE\_SIZE

        snake.coordinates.insert(0, (x, y))

        square = canvas.create\_rectangle(x, y, x + SPACE\_SIZE, y + SPACE\_SIZE, fill=SNAKE\_COLOR)

        snake.squares.insert(0, square)

        if x == food.coordinates[0] and y == food.coordinates[1]:

            global score

            score += 1

            label.config(text="Score:{}".format(score))

            canvas.delete("food")

            food = Food()

        else:

            del snake.coordinates[-1]

            canvas.delete(snake.squares[-1])

            del snake.squares[-1]

        if check\_collisions(snake):

            game\_over()

        else:

            window.after(SPEED, next\_turn, snake, food)

    def change\_direction(new\_direction):

        global direction

        if new\_direction == 'left':

            if direction != 'right':

                direction = new\_direction

        elif new\_direction == 'right':

            if direction != 'left':

                direction = new\_direction

        elif new\_direction == 'up':

            if direction != 'down':

                direction = new\_direction

        elif new\_direction == 'down':

            if direction != 'up':

                direction = new\_direction

    def check\_collisions(snake):

        x, y = snake.coordinates[0]

        if x < 0 or x >= GAME\_WIDTH:

            return True

        elif y < 0 or y >= GAME\_HEIGHT:

            return True

        for body\_part in snake.coordinates[1:]:

            if x == body\_part[0] and y == body\_part[1]:

                return True

        return False

    def game\_over():

        canvas.delete(ALL)

        canvas.create\_text(canvas.winfo\_width() / 2, canvas.winfo\_height() / 2,

                           font=('consolas', 70), text="GAME OVER", fill="red", tag="gameover")

    window = Tk()

    window.title("Snake game")

    window.resizable(False, False)

    score = 0

    direction = 'down'

    label = Label(window, text="Score:{}".format(score), font=('consolas', 40))

    label.pack()

    canvas = Canvas(window, bg=BACKGROUND\_COLOR, height=GAME\_HEIGHT, width=GAME\_WIDTH)

    canvas.pack()

    window.update()

    window\_width = window.winfo\_width()

    window\_height = window.winfo\_height()

    screen\_width = window.winfo\_screenwidth()

    screen\_height = window.winfo\_screenheight()

    x = int((screen\_width / 2) - (window\_width / 2))

    y = int((screen\_height / 2) - (window\_height / 2))

    window.geometry(f"{window\_width}x{window\_height}+{x}+{y}")

    window.bind('<Left>', lambda event: change\_direction('left'))

    window.bind('<Right>', lambda event: change\_direction('right'))

    window.bind('<Up>', lambda event: change\_direction('up'))

    window.bind('<Down>', lambda event: change\_direction('down'))

    snake = Snake()

    food = Food()

    next\_turn(snake, food)

    window.mainloop()

def rock():

    import random

    root = Tk()

    root.geometry("600x600")

    root.title("Rock,Paper and Scissors")

    computer\_value = {

        "0": "Rock",

        "1": "Paper",

        "2": "Scissor"

    }

    def reset\_game():

        b1["state"] = "active"

        b2["state"] = "active"

        b3["state"] = "active"

        l1.config(text="Player              ")

        l3.config(text="Computer")

        l4.config(text="")

    def button\_disable():

        b1["state"] = "disable"

        b2["state"] = "disable"

        b3["state"] = "disable"

    def isrock():

        c\_v = computer\_value[str(random.randint(0, 2))]

        if c\_v == "Rock":

            match\_result = "DRAW (\*￣０￣)"

        elif c\_v == "Scissor":

            match\_result = "You WIN (≧∇≦)!"

        else:

            match\_result = "You LOSE ಥ\_ಥ"

        l4.config(text=match\_result)

        l1.config(text="Rock            ")

        l3.config(text=c\_v)

        button\_disable()

    def ispaper():

        c\_v = computer\_value[str(random.randint(0, 2))]

        if c\_v == "Paper":

            match\_result = "DRAW (\*￣０￣)"

        elif c\_v == "Scissor":

            match\_result = "You LOSE ಥ\_ಥ"

        else:

            match\_result = "You WIN (≧∇≦)!"

        l4.config(text=match\_result)

        l1.config(text="Paper           ")

        l3.config(text=c\_v)

        button\_disable()

    def isscissor():

        c\_v = computer\_value[str(random.randint(0, 2))]

        if c\_v == "Rock":

            match\_result = "You LOSE ಥ\_ಥ"

        elif c\_v == "Scissor":

            match\_result = "DRAW (\*￣０￣)"

        else:

            match\_result = "You WIN (≧∇≦)!"

        l4.config(text=match\_result)

        l1.config(text="Scissor         ")

        l3.config(text=c\_v)

        button\_disable()

    Label(root,

          text="ROCK PAPER & SCISSORS (👉ﾟヮﾟ)👉",

          font="normal 25 bold",

          fg="Black").pack(pady=20)

    frame = Frame(root)

    frame.pack()

    l1 = Label(frame,

               text="Player              ",

               font=24)

    l2 = Label(frame,

               text="VS             ",

               font="normal 15 bold")

    l3 = Label(frame, text="Computer", font=24)

    l1.pack(side=LEFT)

    l2.pack(side=LEFT)

    l3.pack()

    l4 = Label(root,

               text="",

               font="normal 20 bold",

               bg="white",

               width=15,

               borderwidth=2,

               relief="solid")

    l4.pack(pady=20)

    frame1 = Frame(root)

    frame1.pack()

    b1 = Button(frame1, text="🪨",

                font=10, fg="white", width=7,

                bg="black", command=isrock)

    b2 = Button(frame1, text="📃",

                font=10, fg="white", width=7,

                bg="black", command=ispaper)

    b3 = Button(frame1, text="✂",

                font=100, fg="white", width=7,

                bg="black", command=isscissor)

    b1.pack(side=LEFT, padx=10)

    b2.pack(side=LEFT, padx=10)

    b3.pack(padx=10)

    Button(root, text="Reset Game",

           font=10, fg="white",

           bg="black", command=reset\_game).pack(pady=20)

    root.mainloop()

def colour():

    import tkinter.font as font

    import random

    colors = ["Red", "Orange", "White", "Black", "Green", "Blue", "Brown", "Purple", "Cyan", "Yellow", "Pink"]

    global timer,score,word\_color

    timer = 40

    score = 0

    word\_color = ''

    def CountDown():

        global timer

        if (timer >= 0):

            time\_left.config(text="Game Ends in : " + str(timer) + "s")

            timer -= 1

            time\_left.after(1000, CountDown)

            if (timer == -1):

                time\_left.config(text="Game Over!!!")

    def startGame():

        global word\_color

        if (timer == 40):

            CountDown()

            word\_color = random.choice(colors).lower()

            display\_words.config(text=random.choice(colors), fg=word\_color)

            color\_entry.bind('<Return>', displayNextWord)

    def resetGame():

        global timer, score, word\_color

        timer = 40

        score = 0

        word\_color = ''

        game\_score.config(text="Your Score : " + str(score))

        display\_words.config(text='')

        time\_left.config(text="Game Ends in : -")

        color\_entry.delete(0, END)

    def displayNextWord(event):

        global word\_color

        global score

        if (timer > 0):

            if (word\_color == color\_entry.get().lower()):

                score += 1

                game\_score.config(text="Your Score : " + str(score))

            color\_entry.delete(0, END)

            word\_color = random.choice(colors).lower()

            display\_words.config(text=random.choice(colors), fg=word\_color)

    window = Tk()

    window.geometry('600x300')

    window.title("Color Game")

    window.geometry("500x200")

    app\_font = font.Font(family='Helvetica', size=12)

    game\_desp = "Game Description: Enter the color of the word,not the word itself !!"

    myFont = font.Font(family='Helvetica')

    game\_description = Label(window, text=game\_desp, font=app\_font, fg="black")

    game\_description.pack()

    game\_score = Label(window, text="Your Score : " + str(score), font=(font.Font(size=16)), fg="green")

    game\_score.pack()

    display\_words = Label(window, font=(font.Font(size=28)), pady=10)

    display\_words.pack()

    time\_left = Label(window, text="Game Ends in : -", font=(font.Font(size=14)), fg="orange")

    time\_left.pack()

    color\_entry = Entry(window, width=30)

    color\_entry.pack(pady=10)

    btn\_frame = Frame(window, width=80, height=40, bg='red')

    btn\_frame.pack(side=BOTTOM)

    start\_button = Button(btn\_frame, text="Start", width=20, fg="black", bg="pink", bd=0, padx=20, pady=10,

                          command=startGame)

    start\_button.grid(row=0, column=0)

    reset\_button = Button(btn\_frame, text="Reset", width=20, fg="black", bg="light blue", bd=0, padx=20, pady=10,

                          command=resetGame)

    reset\_button.grid(row=0, column=1)

    window.geometry('600x300')

    window.mainloop()

frame = Frame(root, borderwidth=6,bg="grey", relief=SUNKEN)

frame.pack(side=LEFT,anchor="nw")

b1 = Button(frame,fg="red",text="Snake Game",command=snake)

b1.pack(side=TOP,padx=25,pady=40)

b2 = Button(frame,fg="red",text="Guess The number",command=guess)

b2.pack(side=TOP,padx=30,pady=40)

b3 = Button(frame,fg="red",text="Guess Colour",command=colour)

b3.pack(side=TOP,padx=30,pady=40)

b4 = Button(frame,fg="red",text="Rock Paper scissors",command=rock)

b4.pack(side=TOP,padx=20,pady=40)

b5 = Button(frame,fg="red",text="Tic Tac Toe",command=Tic)

b5.pack(side=TOP,padx=20,pady=20)

photo=Image.open("img.png")

img=photo.resize((500,500))

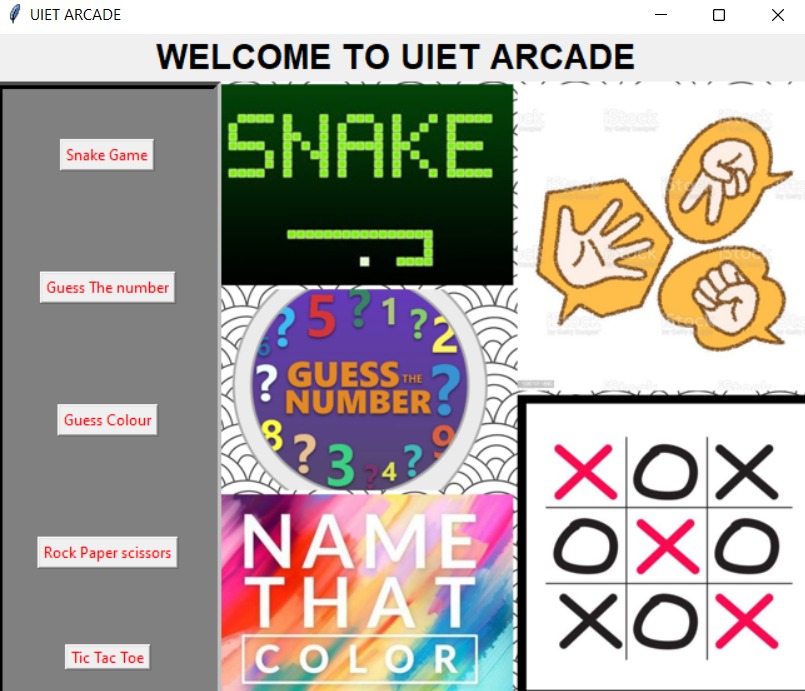
my=ImageTk.PhotoImage(img)

rit=Label(image=my)

rit.pack()

root.mainloop()

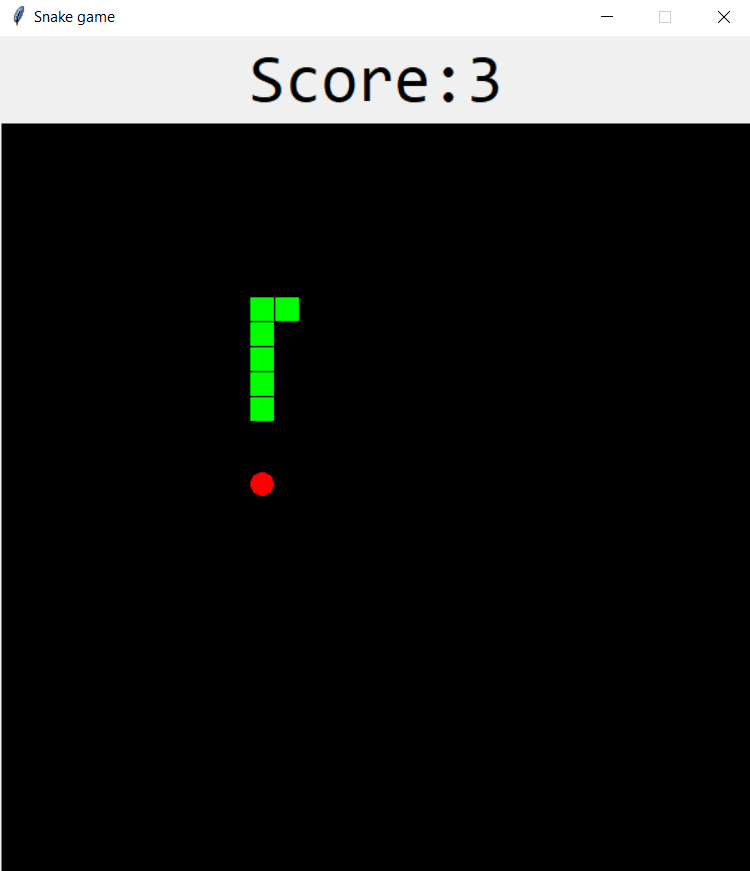
*Output*



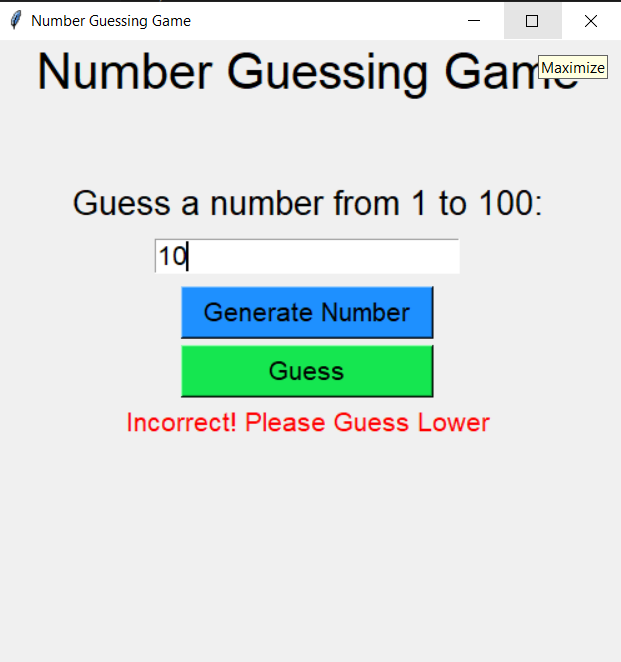
*Main user interface*

Snake game

* FollowinG GUI will appear when you click on the “SNAKE GAME” button.



**Guess The Number**

* Following GUI will appear when you click on the “GUESS THE NUMBER” button.

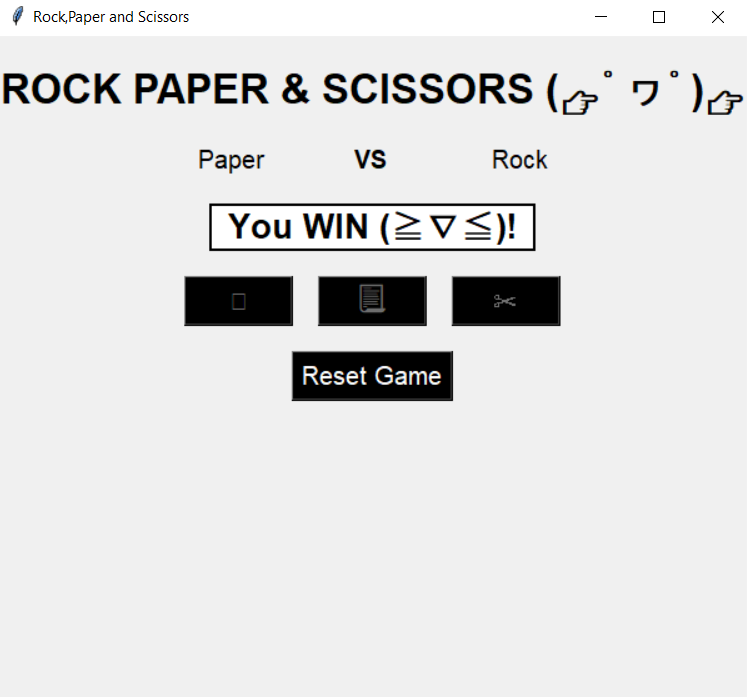
**Name the colour**

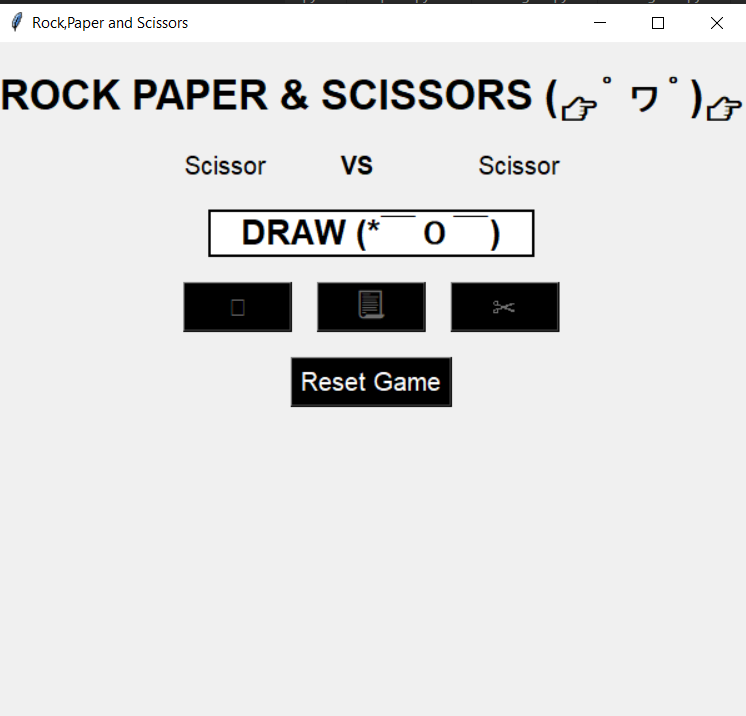
* Following GUI will appear when you click on the “NAME THE COLOUR” button.



**Rock,Paper,scissor**

* Following GUI will appear when you click on the “ROCK,PAPER,SCISSOR” button.





**Tic-tac-toe**

* Following GUI will appear when you click on the “TIC-TAC-TOE” button.

