**TASK4-ROCK PAPER SCISSORS GAME**

**PROJECT TITLE:**

**Rock Paper Scissors Game  
  
DESCRIPTION:  
 This is a GUI-based Rock-Paper-Scissors game where a user plays against the computer. The application displays the choices, maintains scores, and shows game history.  
  
FEATURES:  
1.CustomTkinter GUI with buttons for Rock, Paper, and Scissors.  
2.Random choice generation for computer opponent.  
3.Score tracking, result display, and history.  
4.Option to save game history to a text file.  
  
TECHNOLOGIES USED:  
1.Python  
2.CustomTkinter  
3.random module  
  
TARGET USERS:  
1.General users, casual gamers, students.  
2.Learning aid for understanding GUI programming and basic game logic.**

**APPENDIX:**

**RPSGame.py**

**import customtkinter as ctk**

**import random**

**#Creating a class named RPSGame**

**class RPSGame(ctk.CTk):**

**def \_\_init\_\_(self):**

**super().\_\_init\_\_()**

**self.title("Rock Paper Scissors")**

**self.configure(fg\_color="#1e1e1e")**

**self.resizable(False,False)**

**#Setting the initial Scores and round!**

**self.user\_score=0**

**self.com\_score=0**

**self.round\_no=1**

**self.history\_lines=[]**

**self.create\_widgets()**

**#Creating the widgtes!**

**def create\_widgets(self):**

**self.title\_label=ctk.CTkLabel(self,text="Rock Paper Scissors",font=("Arial",28,'bold'),text\_color="white")**

**self.title\_label.pack(pady=20)**

**#creating the buttons rock,paper,scissors**

**self.button\_frame = ctk.CTkFrame(self,fg\_color="transparent")**

**self.button\_frame.pack(pady=30)**

**self.rockbtn = ctk.CTkButton(self.button\_frame,text="Rock",width=120,command=lambda:self.play("rock"))**

**self.rockbtn.grid(row=0,column=0,padx=15)**

**self.paperbtn=ctk.CTkButton(self.button\_frame,text="Paper",width=120,command=lambda:self.play("paper"))**

**self.paperbtn.grid(row=0,column=1,padx=15)**

**self.scissorsbtn=ctk.CTkButton(self.button\_frame,text="Scissors",width=120,command=lambda:self.play("scissors"))**

**self.scissorsbtn.grid(row=0,column=2,padx=15)**

**self.result\_label = ctk.CTkLabel(self,text="",font=("Arial",20),text\_color='white')**

**self.result\_label.pack(pady=10)**

**self.score\_label=ctk.CTkLabel(self,text="YOU: 0 || COMPUTER: 0",font=("Arial",18),text\_color='white')**

**self.score\_label.pack(pady=10)**

**self.history\_label= ctk.CTkLabel(self,text="Game History",font=("Arial",16,'bold'),text\_color='white')**

**self.history\_label.pack(pady=(10,0))**

**self.hist\_box=ctk.CTkTextbox(self,width=700,height=150,font=("Arial",14))**

**self.hist\_box.pack(pady=10)**

**self.hist\_box.configure(state="disabled")**

**self.bottomframe=ctk.CTkFrame(self,fg\_color="transparent")**

**self.bottomframe.pack(pady=10)**

**self.resetbtn =ctk.CTkButton(self.bottomframe,text="Play Again",command=self.reset,width=150)**

**self.resetbtn.grid(row=0,column=0,padx=10)**

**self.savebtn=ctk.CTkButton(self.bottomframe,text="Save History to file",command=self.save\_history,width=200)**

**self.savebtn.grid(row=0,column=1,padx=10)**

**def play(self,user\_choice):**

**options=['rock','paper','scissors']**

**comp\_choice=random.choice(options)**

**if user\_choice == comp\_choice:**

**result = "It's a Tie!!"**

**color ="#cccccc"**

**elif (user\_choice == 'rock' and comp\_choice == 'scissors') or \**

**(user\_choice == 'scissors' and comp\_choice == 'paper') or \**

**(user\_choice == 'paper' and comp\_choice == 'rock'):**

**result ="You Win!!"**

**self.user\_score+=1**

**color = "#00cc66"**

**else:**

**result = "Computer Wins!!"**

**self.com\_score+=1**

**color = "#ff4444"**

**self.result\_label.configure(text=f"You Chose : {user\_choice.capitalize()} | Computer Chose: {comp\_choice.capitalize()}\n{result}",text\_color=color)**

**self.score\_label.configure(text=f"You: {self.user\_score} | Computer: {self.com\_score}")**

**history\_entry = f"Round {self.round\_no}: You Chose: {user\_choice.capitalize()},Computer Chose: {comp\_choice.capitalize()} --> {result}"**

**self.history\_lines.append(history\_entry)**

**self.hist\_box.configure(state = 'normal')**

**self.hist\_box.insert('end',history\_entry+"\n")**

**self.hist\_box.see('end')**

**self.round\_no+=1**

**def reset(self):**

**self.user\_score = 0**

**self.com\_score = 0**

**self.round\_no = 1**

**self.result\_label.configure(text="",text\_color='white')**

**self.score\_label.configure(text="You: 0 | Computer: 0")**

**self.history\_lines=[]**

**self.hist\_box.configure(state= 'normal')**

**self.hist\_box.delete("0.0","end")**

**self.hist\_box.configure(state="disabled")**

**def save\_history(self):**

**with open("rps\_game.txt","w",encoding="utf-8") as file:**

**for line in self.history\_lines:**

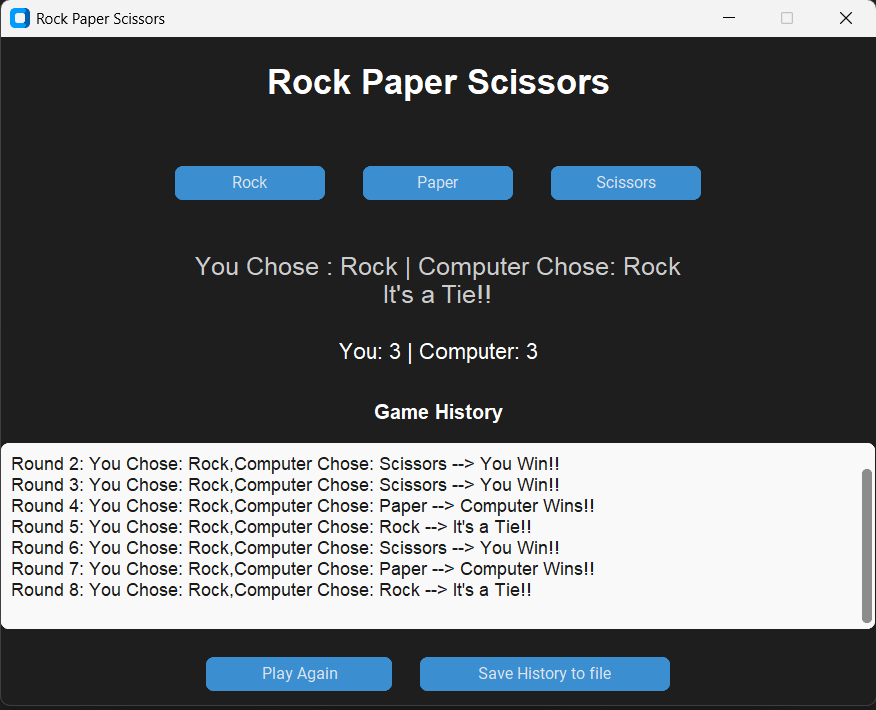
**file.write(line + "\n")**

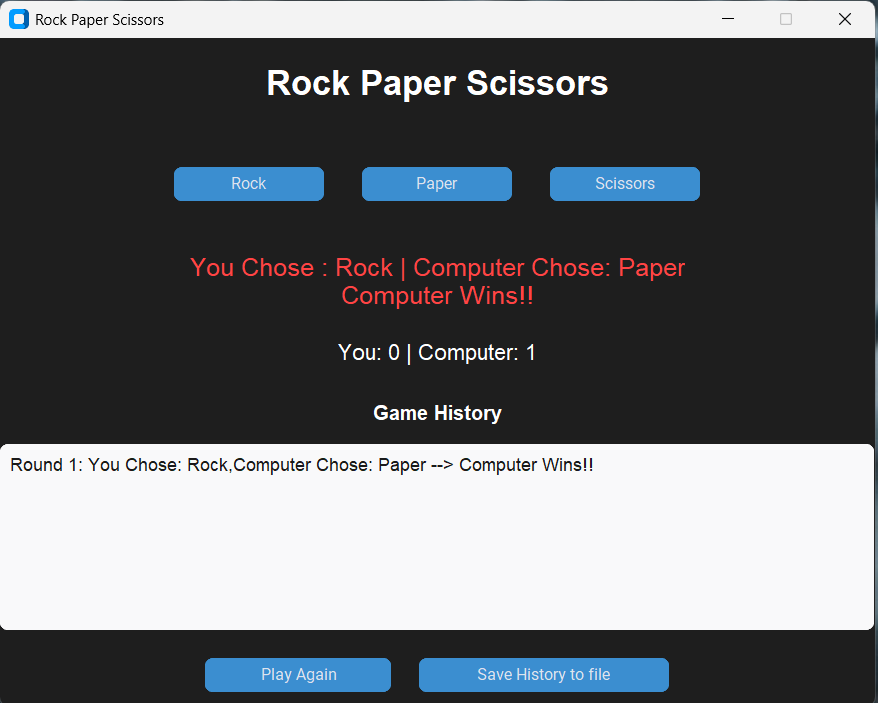
**self.result\_label.configure(text="History Saved to rps\_game.txt",text\_color="yellow")**

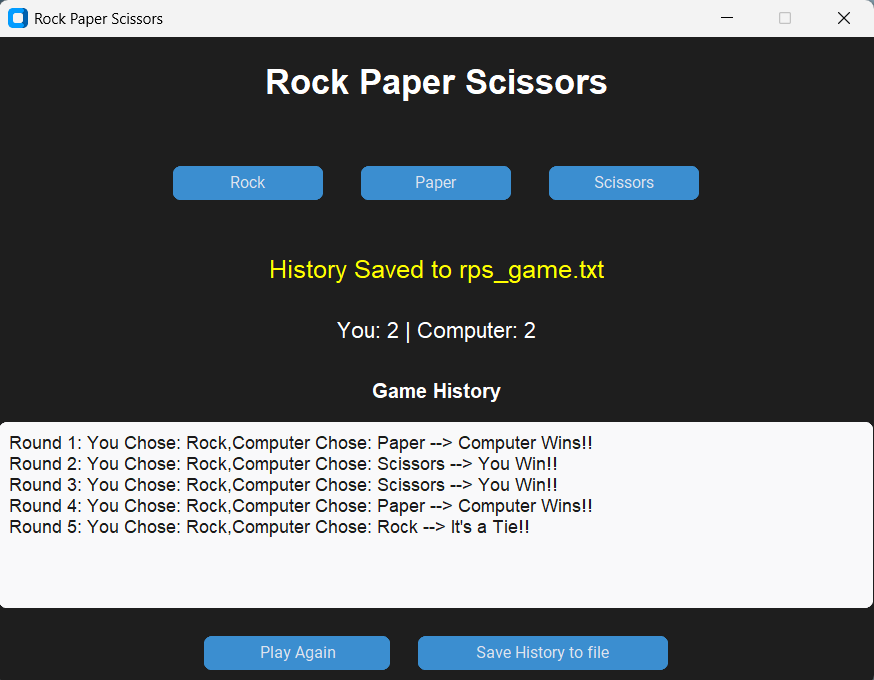
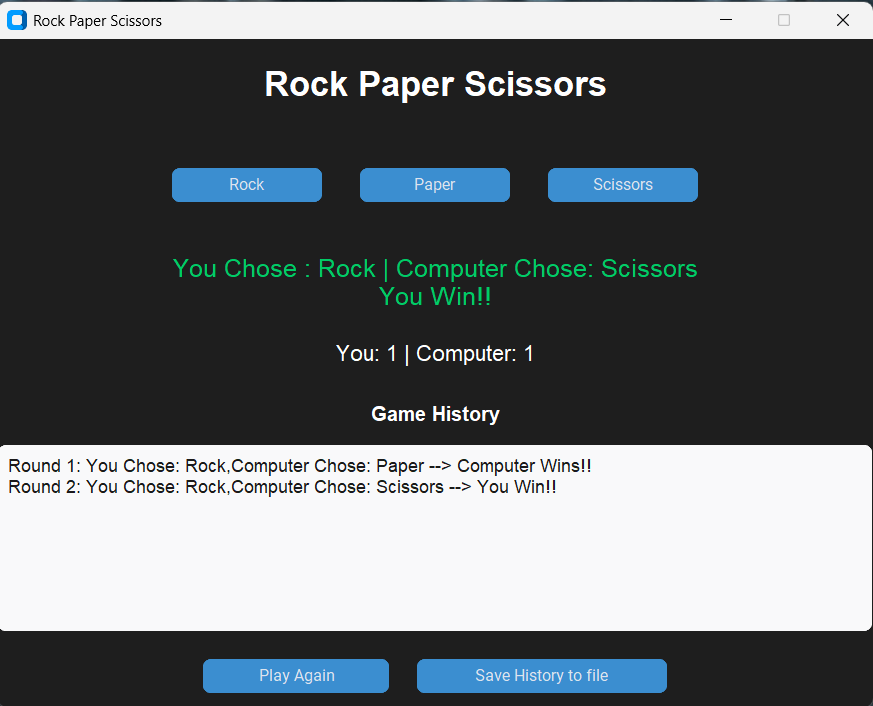
**#launching the game.**

**app=RPSGame()**

**app.mainloop()**

**OUTPUTS:**

****

****