ROCK PAPER AND SCISSOR

```
import tkinter as tk
 from tkinter import messagebox
import pygame

# Initialize pygame mixer
pygame.init()
pygame.mixer.init()

# Play default beep as background music substitute
 def play_music():
  try:
  pygame.mixer.Sound.play(pygame.mixer.Sound(pygame.mixer.Sound(buffer=b'\x00' *
 1000)))
  except:
  pass

# Main Game Class
 class RPSMultiplayerGame:
  def _init_(self, root):
  self.root = root
  self.root.title("Rock Paper Scissors - Multiplayer ��")
  self.root.geometry("400x400")
  self.root.resizable(False, False)

  self.player1_choice = None
  self.player2_choice = None
  self.player1_score = 0
  self.player2_score = 0

  play_music()

  self.create_widgets()

  def create_widgets(self):
  # Labels
  self.label = tk.Label(self.root, text="Multiplayer Rock Paper Scissors", font=("Arial", 14))
self.label.pack(pady=10)
```

```
 self.result = tk.Label(self.root, text="Waiting for choices...", font=("Arial", 12))
 self.result.pack(pady=10)

 self.score = tk.Label(self.root, text=self.get_score_text(), font=("Arial", 12))
self.score.pack(pady=5)

 # Buttons for Player 1
 tk.Label(self.root, text="Player 1").pack()
 frame1 = tk.Frame(self.root)
 frame1.pack()
 for choice in ["Rock", "Paper", "Scissors"]:
 tk.Button(frame1, text=choice, width=10, command=lambda c=choice: self.set_choice(1,
c)).pack(side=tk.LEFT, padx=5)

 # Buttons for Player 2
 tk.Label(self.root, text="Player 2").pack()
 frame2 = tk.Frame(self.root)
 frame2.pack()
 for choice in ["Rock", "Paper", "Scissors"]:
 tk.Button(frame2, text=choice, width=10, command=lambda c=choice: self.set_choice(2,
c)).pack(side=tk.LEFT, padx=5)

 # Reset Button
 tk.Button(self.root, text="Reset Scores", command=self.reset).pack(pady=10)

 def set_choice(self, player, choice):
 if player == 1:
 self.player1_choice = choice
 self.result.config(text="Player 1 locked in.")
 else:
 self.player2_choice = choice
 self.result.config(text="Player 2 locked in.")

 if self.player1_choice and self.player2_choice:
 self.evaluate_winner()

 def evaluate_winner(self):
 p1, p2 = self.player1_choice, self.player2_choice
```

```
 if p1 == p2:
 winner_text = "It's a Tie ��"
 elif (p1 == "Rock" and p2 == "Scissors") or \
 (p1 == "Paper" and p2 == "Rock") or \
 (p1 == "Scissors" and p2 == "Paper"):
 self.player1_score += 1
 winner_text = "Player 1 Wins ��"
 else:
 self.player2_score += 1
 winner_text = "Player 2 Wins ��"

 self.result.config(text=winner_text)
 self.score.config(text=self.get_score_text())

 self.player1_choice = None
 self.player2_choice = None

 def get_score_text(self):
 return f"Score: Player 1 = {self.player1_score} | Player 2 = {self.player2_score}"

 def reset(self):
 self.player1_score = 0
 self.player2_score = 0
 self.score.config(text=self.get_score_text())
 self.result.config(text="Scores reset! Make new choices.")

# Run the Game
if _name_ == "_main_":
 root = tk.Tk()
 game = RPSMultiplayerGame(root)
 root.mainloop()
```