# PROGRAM

import random import time import tkinter as tk

MAX\_TIME = 30

leaderboard\_file = "leaderboard.txt" word\_file = "words.txt"

def load\_words():

try:

with open(word\_file, "r") as file:

return [word.strip().lower() for word in file if word.strip()] except:

print(f"Could not read '{word\_file}'.") return []

def load\_leaderboard():

scores = {} try:

with open(leaderboard\_file, "r") as file: for line in file:

parts = line.strip().split(',') if len(parts) == 2:

name, score = parts scores[name] = float(score)

except:

pass

return scores

def save\_leaderboard(scores):

with open(leaderboard\_file, "w") as file:

sorted\_scores = sorted(scores.items(), key=lambda x: x[1], reverse=True) for name, score in sorted\_scores:

file.write(f"{name},{score}\n")

def show\_leaderboard(scores):

leaderboard\_text.delete('1.0', tk.END) leaderboard\_text.insert(tk.END, "Final Leaderboard:\n")

sorted\_scores = sorted(scores.items(), key=lambda x: x[1], reverse=True) position = 1

for name, score in sorted\_scores:

leaderboard\_text.insert(tk.END, f"{position} {name} - {score} point(s)\n") position += 1

def play\_multiplayer\_game():

global all\_words, players, scores, current\_round, total\_rounds, current\_player\_idx, current\_word, start\_time

all\_words = load\_words() if not all\_words:

leaderboard\_text.insert(tk.END, "No words loaded. Exiting game.\n") return

scores = {} try:

num\_players = int(num\_players\_entry.get()) if num\_players <= 0:

leaderboard\_text.insert(tk.END, "Invalid number of players.\n") return

except ValueError:

leaderboard\_text.insert(tk.END, "Invalid number of players.\n") return

players.clear()

if len(player\_entries) < num\_players:

leaderboard\_text.insert(tk.END, "Please enter all player names first.\n") return

for i in range(num\_players):

name = player\_entries[i].get().strip() if not name:

leaderboard\_text.insert(tk.END, f"Player {i+1} name is missing.\n") return

players.append(name) scores[name] = 0.0

try:

total\_rounds = int(rounds\_entry.get()) except:

leaderboard\_text.insert(tk.END, "Invalid number of rounds.\n") return

current\_round = 1

current\_player\_idx = 0 leaderboard\_text.delete('1.0', tk.END) game\_frame.pack()

next\_turn()

def next\_turn():

global current\_word, start\_time, current\_player\_idx, current\_round

if current\_player\_idx >= len(players): current\_player\_idx = 0

current\_round += 1

if current\_round > total\_rounds: save\_leaderboard(scores) show\_leaderboard(scores) game\_frame.pack\_forget() return

player = players[current\_player\_idx] word = random.choice(all\_words) shuffled = list(word)

attempts = 0 while True:

random.shuffle(shuffled)

if ''.join(shuffled) != word or attempts >= 5: break

attempts += 1

current\_word = word

word\_label.config(text=f"{player}'s Turn: Unscramble → {''.join(shuffled)}") guess\_entry.delete(0, tk.END)

start\_time = time.time()

def submit\_guess():

global current\_player\_idx

guess = guess\_entry.get().strip().lower() elapsed = time.time() - start\_time player = players[current\_player\_idx]

if elapsed > MAX\_TIME:

leaderboard\_text.insert(tk.END, f"{player} took {round(elapsed, 2)}s. Time's up! No points.\n") elif guess == current\_word:

points = max(30 - round(elapsed), 1) scores[player] += points

leaderboard\_text.insert(tk.END, f"{player} guessed correctly in {round(elapsed, 2)}s! +{points} points.\n") else:

leaderboard\_text.insert(tk.END, f"{player} guessed wrong. Word was: {current\_word}\n")

current\_player\_idx += 1 next\_turn()

root = tk.Tk() root.title("Word Game")

root.configure(bg="#1e1e1e")

style\_settings = { "bg": "#1e1e1e",

"fg": "#f0f0f0",

"font": ("Helvetica", 12),

"padx": 10,

"pady": 5,

}

# Main frame (everything except leaderboard) main\_frame = tk.Frame(root) main\_frame.pack(fill=tk.BOTH, expand=True)

# Bottom frame (for leaderboard only) bottom\_frame = tk.Frame(root) bottom\_frame.pack(fill=tk.X, side=tk.BOTTOM)

# Global variables player\_entries = [] players = []

scores = {} all\_words = [] current\_round = 1

total\_rounds = 1

current\_player\_idx = 0 current\_word = "" start\_time = 0

def show\_player\_name\_entry(index=0):

if index >= expected\_num\_players:

# All names entered, show rounds entry rounds\_label.pack() rounds\_entry.pack() start\_button.pack()

return

label = tk.Label(player\_frame, text=f"Player {index+1} name:") label.pack()

entry = tk.Entry(player\_frame) entry.pack() player\_entries.append(entry)

# Automatically show next after entry def on\_return(event):

if entry.get().strip() == "":

leaderboard\_text.insert(tk.END, "Please enter a name before continuing.\n") return

show\_player\_name\_entry(index + 1)

entry.bind("<Return>", on\_return) entry.focus\_set()

def proceed\_to\_names():

global expected\_num\_players

try:

expected\_num\_players = int(num\_players\_entry.get())

if expected\_num\_players <= 0 or expected\_num\_players>10: raise ValueError

except ValueError:

leaderboard\_text.insert(tk.END, "Invalid number of players.\n") return

# Hide number of players input num\_players\_label.pack\_forget() num\_players\_entry.pack\_forget() num\_players\_button.pack\_forget()

show\_player\_name\_entry(0)

num\_players\_label = tk.Label(main\_frame, text="Enter number of players:", \*\*style\_settings) num\_players\_label.pack()

num\_players\_entry = tk.Entry(main\_frame, font=("Helvetica", 12), width=30) num\_players\_entry.pack(pady=5)

num\_players\_entry.bind("<Return>", lambda event: proceed\_to\_names())

num\_players\_button = tk.Button(main\_frame, text="Next", command=proceed\_to\_names, bg="#4CAF50", fg="white", font=("Helvetica", 12), padx=10, pady=5)

num\_players\_button.pack(pady=10)

# --- Step 2: Player Names --- player\_frame = tk.Frame(main\_frame) player\_frame.pack()

# --- Step 3: Rounds & Start ---

rounds\_label = tk.Label(main\_frame, text="Enter number of rounds:") rounds\_entry = tk.Entry(main\_frame)

rounds\_entry.bind("<Return>", lambda event: play\_multiplayer\_game())

start\_button = tk.Button(main\_frame, text="Start Game", command=play\_multiplayer\_game)

# --- Game Area ---

game\_frame = tk.Frame(main\_frame)

word\_label = tk.Label(game\_frame, text="", font=('Arial', 14)) word\_label.pack()

guess\_entry = tk.Entry(game\_frame) guess\_entry.pack()

guess\_entry.bind("<Return>", lambda event: submit\_guess())

submit\_button = tk.Button(game\_frame, text="Submit Guess", command=submit\_guess) submit\_button.pack()

leaderboard\_text = tk.Text(bottom\_frame, height=10, bg="#121212", fg="lime", font=("Courier", 11)) leaderboard\_text.pack(fill=tk.X, padx=10, pady=5)

root.mainloop()