import tkinter as tk

from tkinter import messagebox

import random

categories = {

"fruits": ["apple", "banana", "mango", "orange"],

"animals": ["elephant", "giraffe", "tiger", "kangaroo"],

"tech": ["computer", "python", "keyboard", "monitor"]

}

hints = {

"apple": "A fruit that keeps the doctor away",

"banana": "A long yellow fruit",

"mango": "King of fruits",

"orange": "A citrus fruit",

"elephant": "Largest land animal",

"giraffe": "Tallest land animal",

"tiger": "Big striped cat",

"kangaroo": "Animal that jumps with a pouch",

"computer": "Used to code or browse",

"python": "Programming language or a snake",

"keyboard": "Used to type on a computer",

"monitor": "Displays visuals from a computer"

}

MAX\_WRONG = 6 # Maximum wrong guesses

# ---------------- Main Hangman Game Class ----------------

class HangmanGame:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("🎯 Hangman Game")

self.root.geometry("800x500")

self.root.config(bg="#F0F4F8")

# Canvas for Hangman

self.canvas = tk.Canvas(root, width=250, height=300, bg="#E8EAF6", highlightthickness=0)

self.canvas.place(x=520, y=50)

# Category selection

self.category\_label = tk.Label(root, text="Choose Category: fruits, animals, tech", font=("Helvetica", 12, "bold"), bg="#F0F4F8")

self.category\_label.place(x=20, y=20)

self.category\_entry = tk.Entry(root, font=("Helvetica", 12))

self.category\_entry.place(x=300, y=20)

self.start\_btn = tk.Button(root, text="Start Game", command=self.start\_game, bg="#4CAF50", fg="white", font=("Helvetica", 12, "bold"), width=12)

self.start\_btn.place(x=550, y=17)

# Word display

self.word\_label = tk.Label(root, text="", font=("Helvetica", 28, "bold"), bg="#F0F4F8", fg="#1A237E")

self.word\_label.place(x=20, y=80)

# Hint

self.hint\_label = tk.Label(root, text="", font=("Helvetica", 12, "italic"), fg="#D32F2F", bg="#F0F4F8")

self.hint\_label.place(x=20, y=140)

# Letter buttons

self.buttons\_frame = tk.Frame(root, bg="#F0F4F8")

self.buttons\_frame.place(x=20, y=180)

self.letter\_buttons = {}

# Replay Button

self.replay\_btn = tk.Button(root, text="Replay", command=self.replay\_game, bg="#FF5722", fg="white", font=("Helvetica", 12, "bold"), width=12)

self.replay\_btn.place(x=320, y=450)

self.reset\_game\_variables()

# ---------------- Reset Variables ----------------

def reset\_game\_variables(self):

self.word = ""

self.display\_word = []

self.guessed\_letters = []

self.wrong\_guesses = 0

self.game\_active = False

self.canvas.delete("all")

self.word\_label.config(text="")

self.hint\_label.config(text="")

# Destroy previous buttons

for btn in self.letter\_buttons.values():

btn.destroy()

self.letter\_buttons = {}

# ---------------- Start Game ----------------

def start\_game(self):

category = self.category\_entry.get().lower()

if category not in categories:

messagebox.showinfo("Info", "Invalid category! Defaulting to 'fruits'.")

category = "fruits"

self.word = random.choice(categories[category])

self.display\_word = ["\_"] \* len(self.word)

self.guessed\_letters = []

self.wrong\_guesses = 0

self.game\_active = True

self.update\_word\_label()

self.hint\_label.config(text="")

# Create letter buttons with colors

letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

for i, letter in enumerate(letters):

btn = tk.Button(self.buttons\_frame, text=letter, width=4, height=2,

command=lambda l=letter: self.guess\_letter(l.lower()),

bg="#1976D2", fg="white", font=("Helvetica", 10, "bold"))

btn.grid(row=i // 9, column=i % 9, padx=2, pady=2)

btn.bind("<Enter>", lambda e, b=btn: b.config(bg="#64B5F6"))

btn.bind("<Leave>", lambda e, b=btn: b.config(bg="#1976D2"))

self.letter\_buttons[letter] = btn

# ---------------- Handle Letter Guess ----------------

def guess\_letter(self, letter):

if not self.game\_active:

return

if letter in self.guessed\_letters:

return

self.guessed\_letters.append(letter)

self.letter\_buttons[letter.upper()].config(state="disabled", bg="#B0BEC5")

if letter in self.word:

for i, l in enumerate(self.word):

if l == letter:

self.display\_word[i] = letter

self.update\_word\_label()

if "\_" not in self.display\_word:

messagebox.showinfo("🎉 Hangman", f"Congratulations! You guessed the word: {self.word}")

self.game\_active = False

else:

self.wrong\_guesses += 1

self.draw\_hangman()

if self.wrong\_guesses == MAX\_WRONG // 2 and self.word in hints:

self.hint\_label.config(text=f"Hint: {hints[self.word]}")

if self.wrong\_guesses >= MAX\_WRONG:

messagebox.showinfo("💀 Hangman", f"Game Over! The word was: {self.word}")

self.game\_active = False

# ---------------- Update Word Display ----------------

def update\_word\_label(self):

self.word\_label.config(text=" ".join(self.display\_word))

# ---------------- Draw Hangman ----------------

def draw\_hangman(self):

self.canvas.delete("all")

# Gallows

self.canvas.create\_line(50, 250, 150, 250, width=3, fill="#3E2723")

self.canvas.create\_line(100, 250, 100, 50, width=3, fill="#3E2723")

self.canvas.create\_line(100, 50, 200, 50, width=3, fill="#3E2723")

self.canvas.create\_line(200, 50, 200, 70, width=3, fill="#3E2723")

# Hangman based on wrong\_guesses

if self.wrong\_guesses >= 1:

self.canvas.create\_oval(180, 70, 220, 110, width=3, outline="#D32F2F") # Head

if self.wrong\_guesses >= 2:

self.canvas.create\_line(200, 110, 200, 170, width=3, fill="#D32F2F") # Body

if self.wrong\_guesses >= 3:

self.canvas.create\_line(200, 120, 170, 150, width=3, fill="#D32F2F") # Left arm

if self.wrong\_guesses >= 4:

self.canvas.create\_line(200, 120, 230, 150, width=3, fill="#D32F2F") # Right arm

if self.wrong\_guesses >= 5:

self.canvas.create\_line(200, 170, 170, 200, width=3, fill="#D32F2F") # Left leg

if self.wrong\_guesses >= 6:

self.canvas.create\_line(200, 170, 230, 200, width=3, fill="#D32F2F") # Right leg

# ---------------- Replay Game ----------------

def replay\_game(self):

self.reset\_game\_variables()

# ---------------- Main Program ----------------

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

game = HangmanGame(root)

root.mainloop()