

#Morse Code Trainer for touchscreen written by ChatGPT 4 AI

import tkinter as tk

import random

import time

from playsound import playsound

from threading import Thread

Morse code dictionary

```
MORSE_CODE_DICT = {
    'A': '-.-', 'B': '-...', 'C': '-.-.', 'D': '-..', 'E': '.',
    'F': '..-.', 'G': '--.', 'H': '....', 'I': '..', 'J': '---',
    'K': '-.-.', 'L': '-..', 'M': '--', 'N': '-.', 'O': '---',
    'P': '-.-.', 'Q': '--.-', 'R': '-.-', 'S': '...', 'T': '-',
    'U': '..-', 'V': '...-', 'W': '-.-', 'X': '-.-.', 'Y': '-.-.',
    'Z': '--..', '0': '-----', '1': '.-----', '2': '..---', '3': '...--',
    '4': '....-', '5': '.....', '6': '-....', '7': '--...', '8': '---..',
    '9': '----.'
}
```

Initialize variables

selected_characters = list(MORSE_CODE_DICT.keys())

correct_answers = {char: 0 for char in selected_characters}

wrong_answers = {char: 0 for char in selected_characters}

Play Morse code sound

```
def play_morse_code(morse_code):
    for symbol in morse_code:
        if symbol == '.':
            playsound('dot.mp3') # Replace with actual path to dot sound
            time.sleep(0.1) # Short gap between sounds
        elif symbol == '-':
            playsound('dash.mp3') # Replace with actual path to dash sound
            time.sleep(0.1) # Short gap between sounds
    time.sleep(0.5) # Gap between letters
```

def start_morse_thread(morse_code):

thread = Thread(target=play_morse_code, args=(morse_code,))

thread.start()

Function to start a new round

def new_round():

global current_char

current_char = random.choice(selected_characters)

start_morse_thread(MORSE_CODE_DICT[current_char])

Function to check user's answer

def check_answer(char):

if char == current_char:

correct_answers[char] += 1

else:

wrong_answers[char] += 1

```

update_scoreboard()
new_round()

# Update scoreboard
def update_scoreboard():
    for char in selected_characters:
        correct_label = scoreboard_labels[char]['correct']
        wrong_label = scoreboard_labels[char]['wrong']
        correct_label.config(text=str(correct_answers[char]))
        wrong_label.config(text=str(wrong_answers[char]))

# Reset the game
def reset_game():
    global correct_answers, wrong_answers
    correct_answers = {char: 0 for char in selected_characters}
    wrong_answers = {char: 0 for char in selected_characters}
    update_scoreboard()
    new_round()

# Quit the game
def quit_game():
    root.destroy()

# Initialize main application window
root = tk.Tk()
root.title("Morse Code Trainer")

# Create keyboard buttons
keyboard_frame = tk.Frame(root)
keyboard_frame.pack()

for char in selected_characters:
    btn = tk.Button(keyboard_frame, text=char, font=("Helvetica", 18), width=4, height=2,
                    command=lambda c=char: check_answer(c))
    btn.pack(side=tk.LEFT, padx=5, pady=5)

# Create scoreboard
scoreboard_frame = tk.Frame(root)
scoreboard_frame.pack(pady=20)

scoreboard_labels = {}
for char in selected_characters:
    frame = tk.Frame(scoreboard_frame)
    frame.pack(side=tk.LEFT, padx=10)
    tk.Label(frame, text=char, font=("Helvetica", 18)).pack()
    tk.Label(frame, text="Right:", font=("Helvetica", 12)).pack()
    correct_label = tk.Label(frame, text="0", font=("Helvetica", 12))
    correct_label.pack()
    tk.Label(frame, text="Wrong:", font=("Helvetica", 12)).pack()
    wrong_label = tk.Label(frame, text="0", font=("Helvetica", 12))
    wrong_label.pack()
    scoreboard_labels[char] = {'correct': correct_label, 'wrong': wrong_label}

# Create control buttons

```

```
control_frame = tk.Frame(root)
control_frame.pack(pady=20)

reset_button = tk.Button(control_frame, text="Restart", font=("Helvetica", 18), command=reset_game)
reset_button.pack(side=tk.LEFT, padx=20)

quit_button = tk.Button(control_frame, text="Quit", font=("Helvetica", 18), command=quit_game)
quit_button.pack(side=tk.LEFT, padx=20)

# Start the first round
new_round()

# Start the main loop
root.mainloop()
```