import os  
import pickle  
import tkinter as tk  
from tkinter import filedialog  
from tkinter import PhotoImage  
from pygame import mixer  
  
class Player(tk.Frame):  
 def \_\_init\_\_(self, master=None):  
 super().\_\_init\_\_(master)  
 self.master = master  
 self.pack()  
 mixer.init()  
  
 if os.path.exists(**'songs.pickle'**):  
 with open(**'songs.pickle'**, **'rb'**) as f:  
 self.playlist = pickle.load(f)  
 else:  
 self.playlist=[]  
  
 self.current = 0  
 self.paused = True  
 self.played = False  
  
 self.create\_frames()  
 self.track\_widgets()  
 self.control\_widgets()  
 self.tracklist\_widgets()  
  
 def create\_frames(self):  
 self.track = tk.LabelFrame(self, text=**'Song Track'**,  
 font=(**"GEORGIA"**,15,**"bold"**),  
 bg=**"grey"**,fg=**"white"**,bd=5,relief=tk.GROOVE)  
 self.track.config(width=410,height=300)  
 self.track.grid(row=0, column=0, padx=10)  
  
 self.tracklist = tk.LabelFrame(self, text=**f'PlayList -** {str(len(self.playlist))}**'**,  
 font=(**"georgia"**,15,**"bold"**),  
 bg=**"grey"**,fg=**"white"**,bd=5,relief=tk.GROOVE)  
 self.tracklist.config(width=190,height=400)  
 self.tracklist.grid(row=0, column=1, rowspan=3, pady=5)  
  
 self.controls = tk.LabelFrame(self,  
 font=(**"times new roman"**,15,**"bold"**),  
 bg=**"white"**,fg=**"white"**,bd=2,relief=tk.GROOVE)  
 self.controls.config(width=410,height=80)  
 self.controls.grid(row=2, column=0, pady=5, padx=10)  
  
 def track\_widgets(self):  
 self.canvas = tk.Label(self.track, image=img)  
 self.canvas.configure(width=400, height=240)  
 self.canvas.grid(row=0,column=0)  
  
 self.songtrack = tk.Label(self.track, font=(**"times new roman"**,16,**"bold"**),  
 bg=**"white"**,fg=**"dark blue"**)  
 self.songtrack[**'text'**] = **'Musicxy MP3 Player'** self.songtrack.config(width=30, height=1)  
 self.songtrack.grid(row=1,column=0,padx=10)  
  
 def control\_widgets(self):  
 self.loadSongs = tk.Button(self.controls, bg=**'purple'**, fg=**'yellow'**, font=10)  
 self.loadSongs[**'text'**] = **'Load Songs'** self.loadSongs[**'command'**] = self.retrieve\_songs  
 self.loadSongs.grid(row=0, column=0, padx=10)  
  
 self.prev = tk.Button(self.controls, image=prev, border=0)  
 self.prev[**'command'**] = self.prev\_song  
 self.prev.grid(row=0, column=1)  
  
 self.pause = tk.Button(self.controls, image=pause, border=0)  
 self.pause[**'command'**] = self.pause\_song  
 self.pause.grid(row=0, column=2)  
  
 self.next = tk.Button(self.controls, image=next\_, border=0)  
 self.next[**'command'**] = self.next\_song  
 self.next.grid(row=0, column=3)  
  
 self.volume = tk.DoubleVar(self)  
 self.slider = tk.Scale(self.controls, from\_ = 0, to = 10, orient = tk.HORIZONTAL)  
 self.slider[**'variable'**] = self.volume  
 self.slider.set(5)  
 mixer.music.set\_volume(0.8)  
 self.slider[**'command'**] = self.change\_volume  
 self.slider.grid(row=0, column=4, padx=5)  
  
  
 def tracklist\_widgets(self):  
 self.yscrollbar = tk.Scrollbar(self.tracklist, orient=tk.VERTICAL)  
 self.yscrollbar.grid(row=0,column=1, rowspan=5, sticky=**'ns'**)  
  
 self.xscrollbar = tk.Scrollbar(self.tracklist, orient=tk.HORIZONTAL)  
 self.xscrollbar.grid(row=5, column=0, rowspan=5, sticky=**'we'**)  
  
  
 self.list = tk.Listbox(self.tracklist, selectmode=tk.SINGLE,  
 yscrollcommand=self.yscrollbar.set,xscrollcommand=self.xscrollbar.set, selectbackground=**'sky blue'**,bg=**'violet'**)  
 self.enumerate\_songs()  
 self.list.config(height=22)  
 self.list.bind(**'<Double-1>'**, self.play\_song)  
  
 self.yscrollbar.config(command=self.list.yview)  
 self.xscrollbar.config(command=self.list.xview)  
 self.list.grid(row=0, column=0, rowspan=5)  
  
 def retrieve\_songs(self):  
 self.songlist = []  
 directory = filedialog.askdirectory()  
 for root\_, dirs, files in os.walk(directory):  
 for file in files:  
 if os.path.splitext(file)[1] == **'.mp3'**:  
 path = (root\_ + **'/'** + file).replace(**'**\\**'**,**'/'**)  
 self.songlist.append(path)  
  
 with open(**'songs.pickle'**, **'wb'**) as f:  
 pickle.dump(self.songlist, f)  
 self.playlist = self.songlist  
 self.tracklist[**'text'**] = **f'PlayList -** {str(len(self.playlist))}**'** self.list.delete(0, tk.END)  
 self.enumerate\_songs()  
  
 def enumerate\_songs(self):  
 for index, song in enumerate(self.playlist):  
 self.list.insert(index, os.path.basename(song))  
  
  
 def play\_song(self, event=None):  
 if event is not None:  
 self.current = self.list.curselection()[0]  
 for i in range(len(self.playlist)):  
 self.list.itemconfigure(i, bg=**"white"**)  
  
 print(self.playlist[self.current])  
 mixer.music.load(self.playlist[self.current])  
 self.songtrack[**'anchor'**] = **'w'** self.songtrack[**'text'**] = os.path.basename(self.playlist[self.current])  
  
 self.pause[**'image'**] = play  
 self.paused = False  
 self.played = True  
 self.list.activate(self.current)  
 self.list.itemconfigure(self.current, bg=**'purple'**)  
  
 mixer.music.play()  
  
 def pause\_song(self):  
 if not self.paused:  
 self.paused = True  
 mixer.music.pause()  
 self.pause[**'image'**] = pause  
 else:  
 if self.played == False:  
 self.play\_song()  
 self.paused = False  
 mixer.music.unpause()  
 self.pause[**'image'**] = play  
  
 def prev\_song(self):  
 if self.current > 0:  
 self.current -= 1  
 else:  
 self.current = 0  
 self.list.itemconfigure(self.current + 1, bg=**'white'**)  
 self.play\_song()  
  
 def next\_song(self):  
 if self.current < len(self.playlist) - 1:  
 self.current += 1  
 else:  
 self.current = 0  
 self.list.itemconfigure(self.current - 1, bg=**'white'**)  
 self.play\_song()  
  
 def change\_volume(self, event=None):  
 self.v = self.volume.get()  
 mixer.music.set\_volume(self.v / 10)  
  
*# ----------------------------- Main -------------------------------------------*root = tk.Tk()  
root.geometry(**'600x415'**)  
root.wm\_title(**'Musicxy'**)  
root.iconbitmap(**"D:\SEM 4\Mini Project 1B\images**\\**1.ico"**)  
  
img = PhotoImage(file=**'images/image.gif'**)  
next\_ = PhotoImage(file = **'images/next.png'**)  
prev = PhotoImage(file=**'images/previous.png'**)  
play = PhotoImage(file=**'images/pause.png'**)  
pause = PhotoImage(file=**'images/play.png'**)  
  
app = Player(master=root)  
app.mainloop()