

Software Report

Vevek Manda

May 18, 2023

1 Introduction

The provided code represents a simple music player application developed using the Pygame library in Python. The application allows users to select a folder containing music files, create a playlist from the files, and control music playback, including features like play/pause, next song, shuffle, and quit.

2 Functionality

The code consists of several functions that perform specific tasks:

- `choose_music_folder()`: Opens a file dialog box for the user to select a folder containing music files and returns the selected folder path.
- `get_music_files(folder_path)`: Retrieves all the music files (specifically, files with the ".mp3" extension) from the provided folder path and returns a list of file paths.
- `play_next_song()`: Plays the next song in the playlist. It updates the current song index, loads the corresponding song file, and plays it using the Pygame mixer module.
- `shuffle_playlist()`: Shuffles the order of songs in the playlist. It creates a shuffled copy of the original playlist, resets the current song index, loads the first song from the shuffled playlist, and plays it.
- `update_current_song_label()`: Updates the current song label in the GUI window to display the name of the currently playing song.
- `play_pause()`: Toggles between play and pause states of the music player. It unpauses the playback if paused, and vice versa. It also updates the label of the play/pause button accordingly.
- `next_song()`: Invokes the `play_next_song()` function and updates the current song label.
- `shuffle_songs()`: Invokes the `shuffle_playlist()` function and updates the current song label.
- `quit_music_player()`: Stops the music playback and closes the application window.

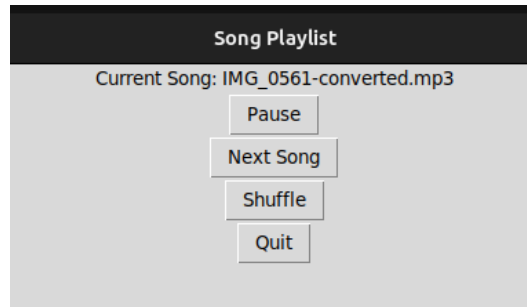


Figure 1: GUI window

3 Graphical User Interface (GUI)

The code uses the Tkinter library to create a GUI window for the music player. The window is titled "Song Playlist" and has dimensions of 400x200 pixels. It contains the following GUI elements:

- `current_song_label`: A label that displays the name of the current song being played.
- `play_pause.button`: A button that allows the user to play or pause the music playback.
- `next_song.button`: A button that skips to the next song in the playlist.
- `shuffle.button`: A button that shuffles the order of songs in the playlist.
- `quit.button`: A button that stops the music playback and closes the application.

4 Initialization and Execution

The code initializes the Pygame library and prompts the user to select a folder containing music files. It then retrieves the music files from the selected folder and shuffles the playlist. The GUI window is created, and the initial current song label is updated. The Pygame mixer module is initialized, and the first song from the shuffled playlist is loaded and played. Finally, the code enters the main loop of the GUI window, allowing the user to interact with the music player until they choose to quit.

5 Conclusion

The provided code demonstrates a basic music player application with essential playback functionalities. Users can select a folder of music files, control playback, skip songs, shuffle the playlist, and quit the application. This code can serve as a starting point for further enhancements, such as adding additional features, improving the user interface, or integrating with online music services.