Lab Report: Exploring a Simple Music Player

Ch.Kushwanth
AI22BTECH11006

Introduction:

The purpose of this lab report is to explore the inner workings of a simple music player created using Python. The music player allows users to play a collection of music tracks from a specified folder in a randomized order. It provides options to skip to the next song, replay the current order, reshuffle the playlist, or exit the program. This report delves into the implementation details and functionality of the music player.

1) Implementation Details:

a) Library Dependencies:
   i) The code utilizes the numpy, os, pydub, and pydub.playback libraries.
   ii) The random library is used for shuffling the playlist.
   iii) The os library is used for handling file and folder operations.
   iv) The pydub library is used for audio file manipulation and playback.

b) Function Definitions:
   i) play-playlist(playlist): This function forms the core of the music player. It handles the main logic for shuffling and playing the playlist, as well as user interactions and input handling.
   ii) get-audio-files-from-folder(folder-path): This function retrieves a list of audio files from a specified folder. It filters files based on supported formats and returns a tuple containing the file path and its corresponding AudioSegment object.

2) Workflow of the Music Player:

a) Playlist Creation:
   i) The user specifies the folder path containing the music tracks.
   ii) The program scans the folder and compiles a list of compatible audio files using the get-audio-files-from-folder() function.
   iii) Each audio file is represented as a tuple containing the file path and its corresponding AudioSegment object.

b) Shuffling and Playback:

i) The program shuffles the original playlist randomly using the np.random.shuffle() function.

ii) It enters a continuous loop to play the songs in the shuffled order.

iii) For each song, the file name is extracted from the path using the os.path.basename() function.

iv) The song is played using the play() function from the pydub.playback library.

c) User Interactions:

i) While the songs are playing, the program prompts the user for input to control the playback.

ii) If the current song is the last in the playlist, the program offers options to skip to the next song, replay the current order, reshuffle the playlist, or exit the program.

iii) If the current song is not the last, the user can choose to skip to the next song or continue playing.

3) **Conclusion:**

The simple music player provides an interactive way to enjoy a collection of music tracks. It allows users to shuffle the playlist, skip songs, replay the current order, reshuffle the playlist, or exit the program. By leveraging libraries such as pydub, the player handles audio file manipulation and playback seamlessly. This lab report has provided an overview of the music player's implementation details, workflow, and user interactions, offering insights into its functionality.

4) **outcome:**

These are some of pictures of working code.

Fig. 1. this shows the playing of songs

Fig. 2. this shows option to skip a song or continue

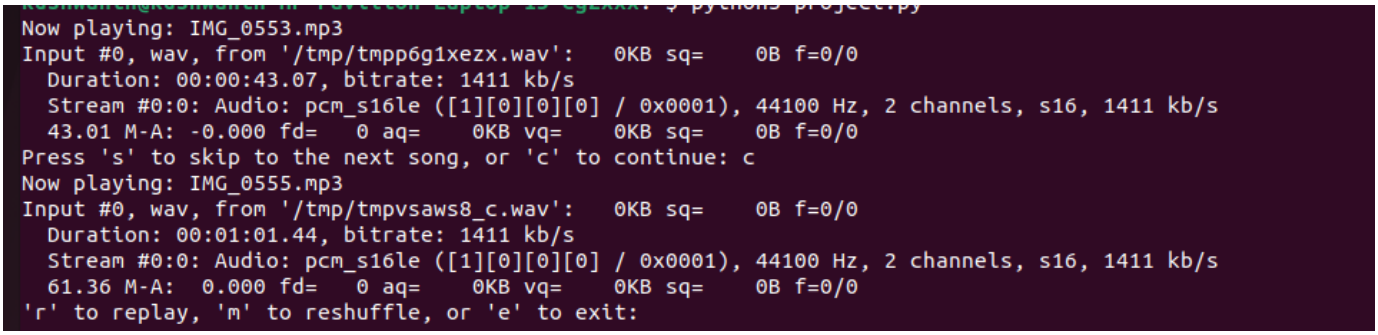

Fig. 3. this shows option that appear after completing whole list