

# Project Report - Random Audio Playlist Player

AI22BTECH11022, RUVVA SURAJ KUMAR

## 1 ABSTRACT

This lab report presents a program that plays audio files randomly from a given playlist directory using the Pygame library in Python. The program allows the user to control the playback, next songs, repeat the playlist and quit the playlist. The playlist is shuffled using the NumPy library to provide a random order of songs.

## 2 INTRODUCTION

The goal of this project is to create a random audio playlist player that provides a seamless and interactive experience for the user. The program utilizes Pygame, a popular multimedia library in Python, to handle audio playback. The playlist directory is specified by the user, and the program shuffles the songs randomly using the NumPy library. The user can control the playback, skip songs, pause and resume, and exit the playlist.

## 3 METHODS

### 3.1 Initializing Pygame

`pygame.mixer.init()` is called to initialize the Pygame mixer module, which handles audio playback.

### 3.2 Loading and Playing Songs

The program retrieves a list of audio files in the playlist directory using `os.listdir()`. The playlist is shuffled using `np.random.shuffle(playlist)` to randomize the order of songs. The program iterates over the shuffled playlist, loads each audio file using `pygame.mixer.music.load()`, and plays it using `pygame.mixer.music.play()`. The name of the current song is printed to the console using `print(f"Now playing: {audio_file}")`.

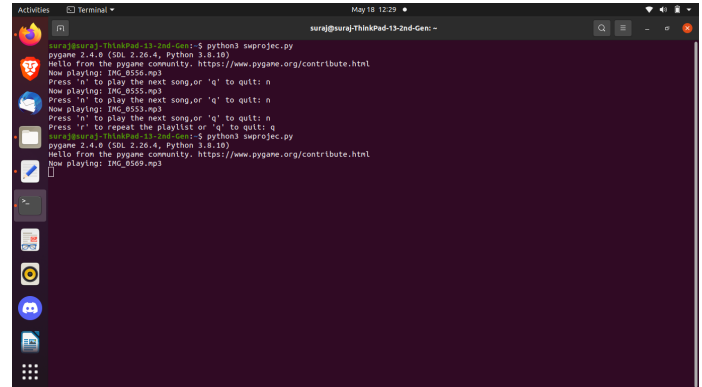


Fig. 1: Playing songs randomly

### 3.3 Handling User Input

During song playback, the program waits until the audio playback is finished using `pygame.mixer.music.get_busy()`. The program prompts the user for input, allowing them to control the playback:

- Press 'n' to proceed to the next song.
- Press 'q' to exit the playlist, stopping the playback.

If the user chooses to proceed to the next song, the program breaks out of the inner while loop and proceeds to the next iteration.

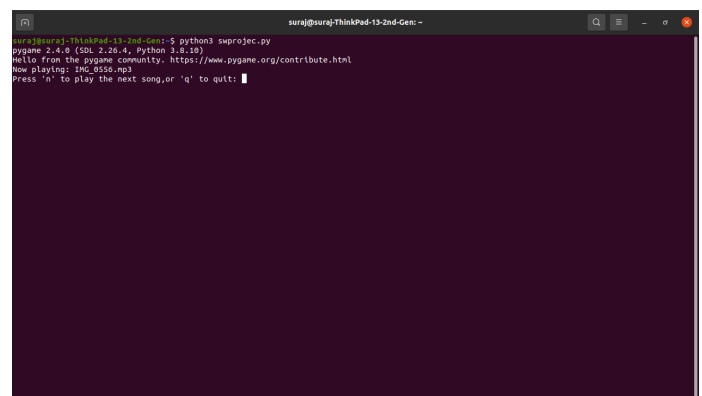


Fig. 2: Displaying next and quit options

### 3.4 Repeating or Quitting the Playlist

After the playlist ends, the program prompts the user to repeat the playlist or quit. If the user chooses to repeat, the `play_random_playlist()` function is called recursively, restarting the playback with a shuffled playlist. If the user chooses to quit, the program exits.

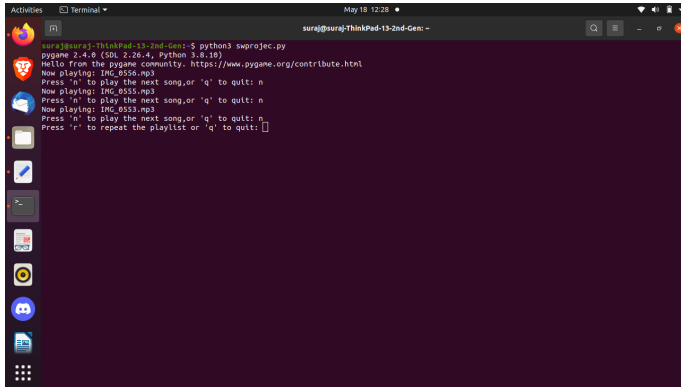


Fig. 3: Displaying repeat and quit options

## 4 RESULTS

The program successfully plays audio files randomly from the specified playlist directory. The songs are shuffled using NumPy, ensuring a random order for each playback. The user can control the playback, skip songs, pause and resume, and exit the playlist as desired.

## 5 CONCLUSION

The random audio playlist player program provides an interactive and enjoyable experience for users who want to listen to their music playlists in a random order. It utilizes the Pygame library for audio playback, allowing for seamless and efficient handling of audio files. The program's functionality to control the playback, skip songs, pause and resume, and exit the playlist enhances the user experience. Overall, the program fulfills its purpose of playing audio files randomly from a given playlist directory.