

# AI1110 Software Project Report

K.SaiTeja  
(AI22BTECH11014)

## I. INTRODUCTION

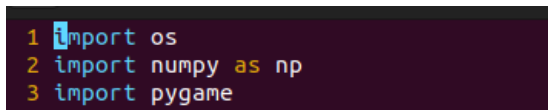
The objective of this project is to develop a simple audio player using Python and the Pygame library. The audio player allows users to play a collection of MP3 files stored in a specified directory. It offers basic playback controls such as play, pause, resume, next, and quit. Additionally, the order of the audio files is shuffled to provide variation during playback.

## II. IMPLEMENTATION

The project implementation can be summarized as follows:

### A. Importing Required Libraries

The necessary libraries, including `os`, `numpy`, and `pygame`, are imported to enable file operations, shuffling, and audio playback, respectively.

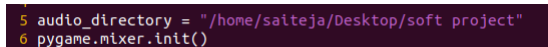


```
1 import os
2 import numpy as np
3 import pygame
```

Fig. 1. Playing songs randomly

### B. Setting the Audio Directory

The audio directory path is specified, which represents the location where the MP3 files are stored.



```
5 audio_directory = "/home/saiteja/Desktop/soft project"
6 pygame.mixer.init()
```

Fig. 2. Audio Directory and Python Mixer

### C. Initializing Pygame Mixer

The Pygame mixer module is initialized to enable audio playback functionality.

### D. Creating a List of Audio Files

The program retrieves a list of files in the audio directory using `os.listdir`, filters for MP3 files using the `.endswith(".mp3")` condition, and creates a list containing the full paths to these audio files.

### E. Iterating Through the Audio Files

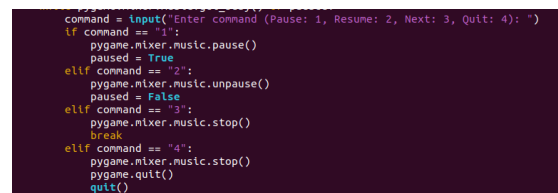
The program loops through the audio files. Before each iteration, the list of audio files is shuffled using `np.random.shuffle` to randomize the order.

### F. Audio Playback

For each audio file, it is loaded using `pygame.mixer.music.load` and played using `pygame.mixer.music.play`. The name of the currently playing song is displayed.

### G. User Interaction

While the audio is playing or paused, the program waits for user input. The user can enter commands such as "1" to pause, "2" to resume, "3" to skip to the next song, or "4" to quit the application. The program responds accordingly to the user's commands by invoking the appropriate Pygame mixer functions.



```
command = input('Enter Command (Pause: 1, Resume: 2, Next: 3, Quit: 4): ')
if command == '1':
    pygame.mixer.music.pause()
    paused = True
elif command == '2':
    pygame.mixer.music.unpause()
    paused = False
elif command == '3':
    pygame.mixer.music.stop()
    break
elif command == '4':
    pygame.mixer.music.stop()
    pygame.quit()
    quit()
```

Fig. 3. User Inputs

### H. Shuffling the Audio Files

After playing through the entire playlist, the program shuffles the list of audio files again using `np.random.shuffle` to provide a different order for the next iteration.

```
np.random.shuffle(audio_files)
```

Fig. 4. Shuffling Playlist

### III. CONCLUSION

In conclusion, the developed audio player project allows users to play a collection of MP3 files from a specified directory. It provides basic playback controls and shuffles the order of the songs for variation. This project demonstrates the use of Python and the Pygame library to create a simple yet functional audio player. Further enhancements can be made to include features like volume control, playlist management, and graphical user interface (GUI) integration.

```
saiteja@saitejas-laptop:~$ nvim project.py
saiteja@saitejas-laptop:~$ python3 project.py
pygame 2.4.0 (SDL 2.26.4, Python 3.10.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
Now playing: IMG_0565.mp3
Enter command (Pause: 1, Resume: 2, Next: 3, Quit: 4): 3
Now playing: IMG_0562.mp3
Enter command (Pause: 1, Resume: 2, Next: 3, Quit: 4): 3
Now playing: IMG_0561.mp3
Enter command (Pause: 1, Resume: 2, Next: 3, Quit: 4): 1
Enter command (Pause: 1, Resume: 2, Next: 3, Quit: 4): 2
Enter command (Pause: 1, Resume: 2, Next: 3, Quit: 4): 3
Now playing: IMG_0553.mp3
Enter command (Pause: 1, Resume: 2, Next: 3, Quit: 4): 4
saiteja@saitejas-laptop:~$
```

Fig. 5. Overall Output Interface