Software Assignment Report

Anagha Balaji
EE22BTECH11204

1 Aim

To make a python script which can make a playlist of songs and shuffle them.

2 Introduction

The provided code implements an audio player using the Pygame library in Python. It allows users to play, pause, skip to the next track, go back to the previous track, and quit the application. The player loads audio files from a specified directory and supports playback of MP3 files.

3 Overview

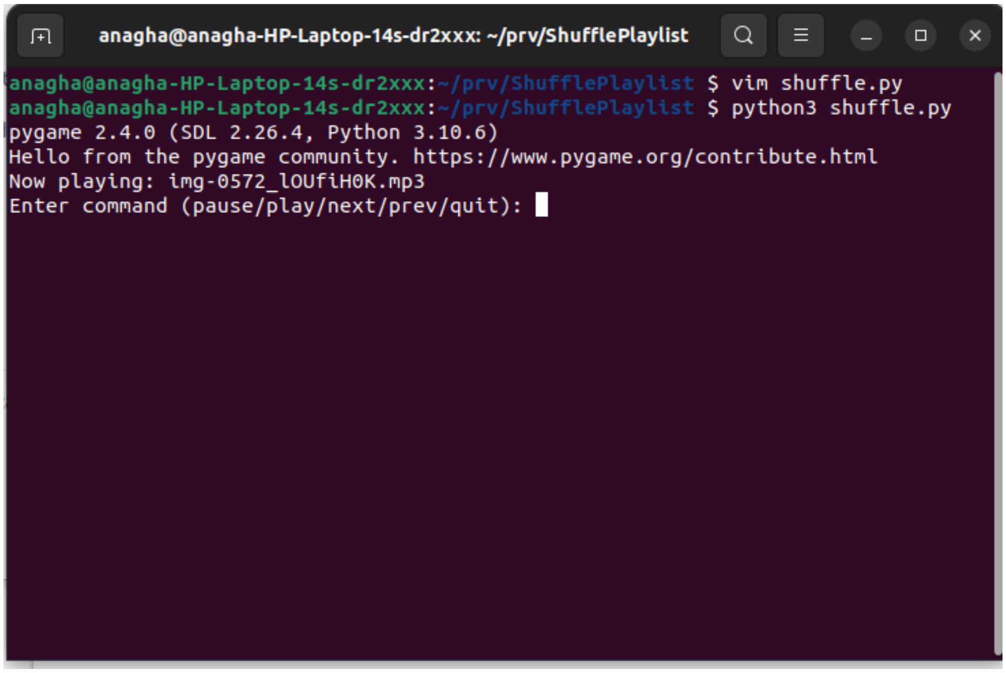
The code begins by importing the necessary modules, including os and pygame. It then sets the path to the directory containing the audio files and initializes the Pygame mixer.
Next, it retrieves a list of audio files in the specified directory, filtering only the ones with the ".mp3" extension. The code also initializes a variable for the current song index.
The playcurrentsong() function is defined to load and play the current song. It utilizes the pygame.mixer.music module to load the audio file and start playback. The function also prints the name of the current song.
The initial song is played by calling the playcurrentsong() function. Afterward, a while loop is used to repeatedly prompt the user for commands via the input() function.

Inside the loop, the user's command is obtained and matched against several possibilities. If the command is "pause", the player pauses the currently playing song using pygame.mixer.music.pause(). If the command is "play", the player resumes playback using pygame.mixer.music.unpause(). For "next" and "prev" commands, the current song index is incremented or decremented, respectively, and the player stops the current song and calls playcurrentsong() again with the updated index.
The loop continues until the user enters the "quit" command. In this case, the current song is stopped, and the loop is exited, ending the program.

4 Conclusion

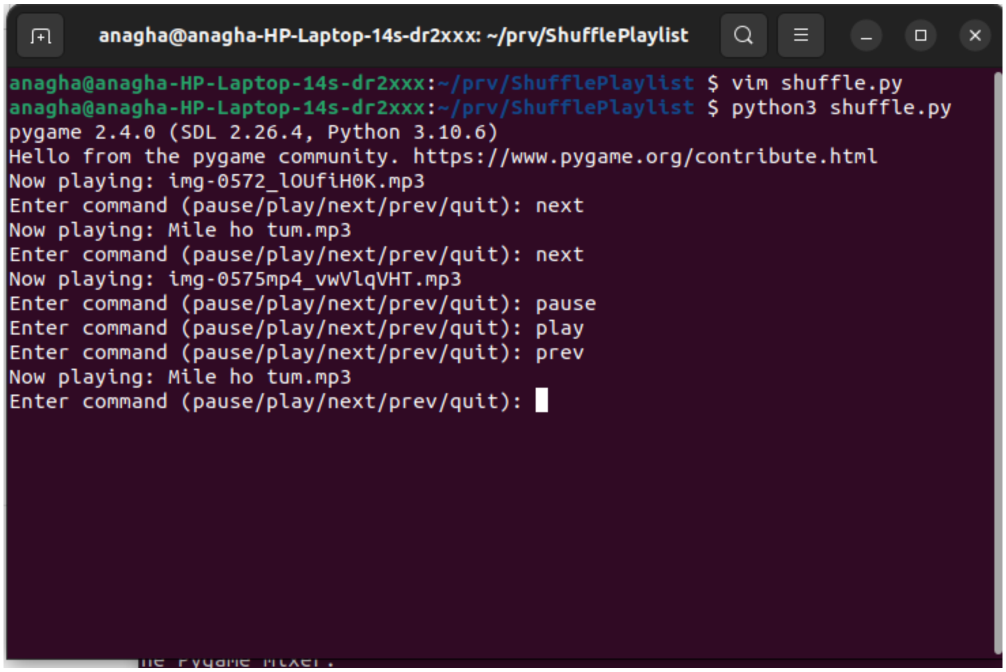
The provided code offers a basic audio player that allows users to control playback, skip between tracks, and quit the application. It utilizes the Pygame library to handle audio playback in an efficient and straightforward manner. The code can be extended and customized further to include additional features, such as shuffle or repeat modes, volume control, or a graphical user interface.

5 Terminal



```
anagha@anagha-HP-Laptop-14s-dr2xxx:~/prv/ShufflePlaylist $ vim shuffle.py
anagha@anagha-HP-Laptop-14s-dr2xxx:~/prv/ShufflePlaylist $ python3 shuffle.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-0572_lOUfiH0K.mp3
Enter command (pause/play/next/prev/quit): 
```

Figure 1:



```
anagha@anagha-HP-Laptop-14s-dr2xxx:~/prv/ShufflePlaylist $ vim shuffle.py
anagha@anagha-HP-Laptop-14s-dr2xxx:~/prv/ShufflePlaylist $ python3 shuffle.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-0572_lOUfiH0K.mp3
Enter command (pause/play/next/prev/quit): next
Now playing: Mile ho tum.mp3
Enter command (pause/play/next/prev/quit): next
Now playing: img-0575mp4_vwVlqVHT.mp3
Enter command (pause/play/next/prev/quit): pause
Enter command (pause/play/next/prev/quit): play
Enter command (pause/play/next/prev/quit): prev
Now playing: Mile ho tum.mp3
Enter command (pause/play/next/prev/quit): 
```

Figure 2: