Project Report: Random Audio Playlist

Apar Saxena (bt22btech11003)

May 18, 2023

1 Introduction

The goal of this project was to develop a Python script that generates a random audio playlist and plays the audio files in a random order. The script allows users to listen to their collection of audio files in a fresh and unpredictable sequence, adding excitement to the listening experience.

2 Implementation

The implementation involved the following steps:

- **Audio Directory:** The script begins by specifying the directory where the audio files are located. Users need to provide the actual path to the audio directory.
- **File Retrieval:** Using the `os.listdir` function, the script retrieves a list of all files within the specified audio directory. It then filters this list to include only the files with the '.mp3' extension.
- **Randomization:** The filtered list of audio files is shuffled using the `random.shuffle` function. This randomization step ensures that the playlist is unique and unpredictable for each run.
- **Audio Playback:** The script uses the `playsound` library to play each audio file in the shuffled list. The `playsound` function is called within a `for` loop, sequentially playing the audio files.

3 Usage

To use the script, follow these steps:

1. Install the `playsound` library by running `pip install playsound` in the terminal or command prompt.
2. Modify the `audio_dir` variable in the script to specify the path to the directory containing your audio files.

3. Run the script using Python: `python audio_playlist.py`.

4 Conclusion

The random audio playlist script provides a simple and effective solution for generating random playback sequences of audio files. By shuffling the playlist, users can enjoy their audio collection in a fresh and exciting way, adding variety and unpredictability to their listening experience.

5 Further Enhancements

The project can be further improved in the following ways:

- Implement a graphical user interface (GUI) to provide a user-friendly interface for selecting the audio directory and controlling the playback.
- Support multiple audio file formats to cater to different user preferences.
- Allow users to customize the playlist length or specify the number of times the playlist should be played.
- Implement a feature to skip to the next audio file during playback.
- Enable the script to remember the last played position and resume from that point in the next session.

Overall, this project serves as a foundation for building more advanced audio playlist management systems and provides a starting point for adding additional features to enhance the user experience.