CA PROJECT

MUSIC-SYSTEM-PLAYER

CSE228

Lovely Professional University
Jalandhar, Punjab, India.

By: Pranshul Thakur
Roll no: 10
12219336

ABSTRACT

This report presents the design and implementation of a music player system that utilizes queues to optimize playlist management and song transitions. We discuss the rationale behind using queues in a music player, emphasizing their advantages in dynamic playlist control.

We delve into the system's architecture, including the user interface, audio playback engine, and queue data structure, explaining how these components work together to ensure seamless music playback.

Our implementation section focuses on Java programming and libraries, describing queue operations, audio file handling, and user interaction through the graphical interface.

We explore user experience features, such as adding/removing songs, shuffling, and looping options. We also touch on visual playlist representation.

Acknowledgement

A student of a university will have to write the acknowledgement for their project or research paper stating that the submission is done and is not copied. The acknowledgement sample for the university project is mostly attached after the dedication page thanking all the faculty members of the department, HOD, the Dean and the mentor.

I would like to express my special thanks to our mentor Mr. Shubham Sharma for his time and efforts he provided throughout the year. Your useful advice and suggestions were really helpful to me during the project's completion. In this aspect, I am eternally grateful to you.

I would like to acknowledge that this project was completed entirely by me and not by someone else.

Table of Contents

INTRODUCTION

In today's digital landscape, music player applications have become essential tools for experiencing and enjoying music. These applications have evolved from simple audio players to sophisticated systems designed to enhance our auditory journeys. This report is a journey into the creation of an innovative music player that exploits queue data structures to redefine the art of music playback. Music is a universal language, capable of evoking emotions, memories, and reshaping our perception of the world. A music player is more than just a tool for playing audio files; it is the conduit between the listener and the world of harmonious sounds. Our journey begins with a fundamental question: How can we make music listening more engaging and dynamic? The answer lies in utilizing data structures, specifically queues, as the foundation of our music player system. Queues, with their First-In-First-Out (FIFO) principle, offer dynamic playlist management, ensuring seamless transitions between tracks. This report explores the nuances and technical aspects of our music system player. We will outline the motivation behind using queues in a music player and the numerous advantages they bring to playlist management and song transitions. We will also examine the system's architecture, which includes user-friendly interfaces, a robust audio playback engine, and efficient queue management.

Objective and Scope of Project

The project aims to create a Java-based music player that utilizes a queue for managing playlists. The primary objectives are to provide a user-friendly interface for:

1. Adding and removing songs from the playlist.
2. Efficiently playing music tracks.
3. Persisting playlists to enable users to resume sessions.
4. Handling errors gracefully.
5. Ensuring cross-platform compatibility.

The project scope encompasses the following functionalities:

1. Queue-based playlist management.
2. Basic music playback controls (play, pause, skip).
3. Song addition and removal from the playlist.
4. Saving and loading playlists.
5. User-friendly interface.
6. Error handling.
7. Cross-platform compatibility.

Application Tools

1. Java Development Kit (JDK)
2. Integrated Development Environment
3. JavaFX or Swing (GUI Development)
4. Audio Playback Library
5. Metadata Extraction Library
6. Data Structures
7. File I/O
8. External Libraries
9. Testing Frameworks
10. Documentation Tools
11. Build Tools

Flowchart

1. Now Playing (A song selected by user) Assume that a song ends only when user selects Play Next in the menu
2. Add Songs to Application (Use a LinkedList to store songs)
3. Queue

   a. Create Queue by adding songs from the app by their titles (search by keywords containing phrases entered by user)

   b. Edit

      1) Add Song (by searching through title)

      2) Remove Song

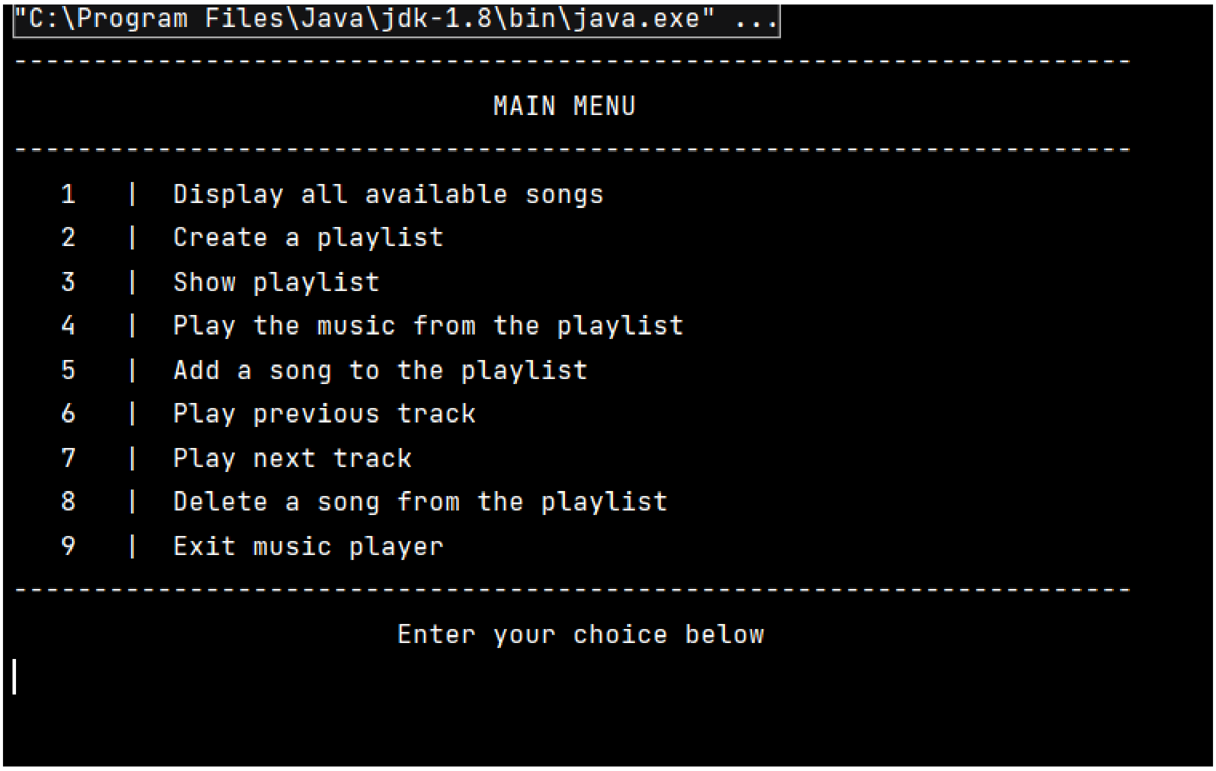
   c. Mode Select

      1) Repeat one (Play Next causes the same song to play)

      2) Repeat All

      3) Shuffle

SCREENSHOTS



```
"C:\Program Files\Java\jdk-1.8\bin\java.exe" ...
----------------------------------------------------------------------
                              MAIN MENU
----------------------------------------------------------------------
   1   |  Display all available songs
   2   |  Create a playlist
   3   |  Show playlist
   4   |  Play the music from the playlist
   5   |  Add a song to the playlist
   6   |  Play previous track
   7   |  Play next track
   8   |  Delete a song from the playlist
   9   |  Exit music player
----------------------------------------------------------------------
                        Enter your choice below
```

Summary

The Music System Player implemented in Java, which utilizes queues as its underlying data structure, offers a comprehensive solution for managing and enjoying music. This application employs a First-In-First-Out queue to organize music tracks, enabling users to seamlessly add, remove, and reorder songs in the queue. It provides standard playback controls, such as play, pause, stop, and skip, as well as the ability to create and save playlists, offering a personalized music experience. Users can also shuffle tracks and repeat single songs or the entire queue for a tailored listening experience. The user-friendly interface displays song metadata, including title, artist, album, and duration. Additionally, robust error handling ensures smooth operation, even with issues like missing files. This Java-based Music System Player enhances the management and playback of music while accommodating various audio file formats, makes it a versatile choice for music enthusiasts

Bibliography

- GitHub
- Codespeedy
- Notes made in Shubham sir’s class
- Upgrads ppts
- Data-Flair