

DJ Application Project Report

Student Name: Azha Balouch

Project Overview:

The DJ Application project was undertaken with the goal of creating a versatile and feature-rich DJ software using the JUCE library. The project encompassed a wide range of functionalities inspired by real-world DJ applications, with a focus on customizing the user interface (UI), enhancing audio playback capabilities, and introducing a unique DJ-related feature. Throughout the development process, I conducted thorough research to ensure that the application met specific requirements outlined as follows:

R1: Basic Functionality

R1A - Loading Audio Files: The application successfully implemented the ability to load audio files into audio players. This feature allowed DJs to import their music library seamlessly.

R1B - Playing Multiple Tracks: The application excelled in this requirement by allowing users to play two or more tracks simultaneously, mimicking the experience of traditional DJ setups with multiple decks.

R1C - Track Mixing: To provide DJs with precise control over their mixes, the application enabled users to adjust the volumes of each track independently. This feature allowed for seamless blending and mixing of songs.

R1D - Speed Control: Another vital aspect of DJing is the ability to alter track speed. The application allowed users to speed up or slow down tracks, enabling creative tempo adjustments during performances.

R2: Customized User Interface (UI)

R2A - GUI Layout: The user interface of the DJ Application significantly deviated from the basic DeckGUI seen in class. I designed a unique and visually appealing layout that catered to the specific needs of DJs. The UI featured enhanced graphics and layout elements, ensuring a modern and engaging user experience.

R2B - Event Listeners: The GUI code incorporated additional event listeners beyond the original codebase seen in class. These event listeners enriched user interaction and enabled dynamic responses to user actions, enhancing the overall functionality and usability of the application.

R3: Implementing a New DJ-Related Feature

In accordance with Requirement R3, I conducted in-depth research into existing DJ applications, utilizing resources such as YouTube and other online references. During this research, I identified a compelling feature to implement, inspired by real DJ programs. A pitch effect and reverb.

Project Components:

1. Application Framework:

- I chose to use the JUCE library as the foundation for the application due to its versatility and cross-platform compatibility.

2. User Interface:

- I designed a custom UI using the Bango KIT from a licensed website (elements.envato). This provided a visually appealing and user-friendly interface.
- Implemented various themes for UI components, including buttons, sliders, and waveform displays, as listed in the provided files.

3. Audio Player:

- The core of the application was the DJAudioPlayer class, responsible for loading and playing audio tracks.
- I integrated pitch and reverb sound effects into the player to enhance the mixing capabilities.

4. Deck Management:

- The application featured deck management functionality, allowing users to load and manipulate multiple tracks simultaneously.

5. Playlist Management:

- I implemented a PlaylistComponent to organize and manage playlists, enabling users to create, edit, and play custom playlists.

6. Pads to Speakers:

- Initially, I attempted to implement pads for triggering sounds, but it encountered technical issues.
- To ensure a seamless user experience, I switched to using speakers for audio playback.

7. Meter and Position Controls:

- Included meter and position slider themes to provide users with precise control over audio levels and track positioning.

GitHub Repository:

The project's codebase is hosted on GitHub for transparency and collaboration. You can access the code and detailed documentation at <https://github.com/aazhaa/DJ-Application>.

Challenges and Learning:

- One of the main challenges I encountered was the transition from pads to speakers, which required significant adjustments to the codebase.
- Learning to integrate audio effects like pitch and reverb was an enriching experience and added valuable skills to my repertoire.

Future Improvements:

Future enhancements to the DJ Application could include:

- Advanced mixing and crossfade capabilities.
- Integration with external MIDI controllers for more tactile control.
- Enhanced waveform visualization for precise track manipulation.

Conclusion:

In conclusion, the DJ Application project successfully created a versatile and visually appealing DJ software using the JUCE library. It incorporated essential DJ functionalities, sound effects, and a user-friendly interface. While facing some challenges along the way, the project provided valuable learning experiences and room for future improvements.