Music Player Project Progress Report - Group 17

Group Members and contribution distribution

- 1155157839 NG Yu Chun Thomas
  - Wav decoding, music format capability, progress bar
- 1155125979 Kong Kwai Man
  - UI, music control
- 1155143519 Woo Pok
  - LRC displayer, music visualizer
- 1155157719 Leung Kit Lun
  - Data management, play mode

Introduction

Github repository: https://github.com/alexkkm/CSCI3280Project

This project aims to create a music player with a web interface using React library. React is a popular front-end library that provides the ability to create complex user interfaces using components. This project leverages the advantages of React, such as the ability to create reusable components, fast rendering, and easy debugging.

Currently, the project includes the following features:

- Audio player component with play, pause, next and previous track functionalities. (Basic feature)
- Wav file decoding feature. (Basic feature)
- Support ogg and aac format. (Enhanced feature)
- Lyrics (LRC) display feature. (Enhanced feature)
- Music visualization feature. (Enhanced feature)
- Switching play mode (single, loop, random). (Enhanced feature)

Parts that are in progress:

- A database system, now the music library information (audio file path, artist, album, cover photo path and lyrics path) is stored in a JSON file.
- A better GUI.
- Data uplaoding and deleting interface.
- Phase 2 features.

File Tree

```
CSCI3280Project
├ .git
├ node_modules
├ public
|  ├ Cover
|  ├ Lyrics
|  ├ Music
```

```
│   └ index.html
├ src
│   ├ AudioScript
│   │   ├ AudioPlayer.js
│   │   ├ LrcDisplayer.js
│   │   ├ MusicVisualizer.js
│   │   └ WavDecoder.js
│   ├ MusicDatabase
│   │   └ musicList.json
│   ├ asserts
│   │   └ background_pic.png
│   ├ index.css
└   └ index.jsx
```

Important Implementation Files Description

- index.html: The HTML file that serves as the entry point for the application.

- index.jsx: The JavaScript file that renders the React components to the DOM.

- index.css: The CSS file that provides styling to the components.

- AudioPlayer.js: This React component renders the audio player which is responsible for playing music. It utilizes the HTML5 audio API for playing and controlling audio tracks. It also provides the basic controls for the audio player such as play, pause, and volume, using buttons and a progress bar.

- LrcDisplayer.js: This React component reads the LRC lyrics file and displays the lyrics synchronized with the audio. It uses regular expressions to parse the LRC file and calculate the timing for each line of the lyrics.

- MusicVisulizer.js: This React component creates and displays the music visualization using the Web Audio API. It creates a frequency analyzer and a canvas element to visualize the waveform of the audio track. The music visualization is synchronized with the audio playback.

- WavDecoder.js: This JavaScript file decodes the audio file and returns its waveform. It utilizes the Web Audio API to load and decode the audio file. The decoded waveform data can be used for creating music visualizations and other audio-related applications.

Conclusion

The Music Player Project, implemented essential features such as an audio player component, LRC lyric display, music visualization, and WAV file decoding. These features have been developed using React's powerful and efficient library to provide a fast, reusable and easy-to-debug interface. However, there are still some parts in progress, such as the implementation of a database for storing music library information and improving the GUI. Overall, this project has provided a valuable learning experience in developing a music player application using React and the Web Audio API.