Peer-to-Peer Music Sharing System

CSCI3280 Group Project Phase I: 13.02.2023 - 13.03.2023 (deadline at 23:59)

Late submission penalty: 10% daily deduction (maximum 30%).

DO NOT directly copy code from ***the internet / other groups*** if you want to pass the course.

Overview

In this project, you are required to achieve **two** main goals.

- ***(Phase I)*** Firstly, you are required to implement a graphical music player to play WAV audio files, control the playback, display music lyrics, and manage a music library.
- ***(Phase II)*** Secondly, you are required to build a Peer-to-Peer (P2P) system for playing music (in the form of streaming) from remote computers. Each computer works as both client and server, which means you may get the audio data from other computers and share audio data for others to be downloaded.

Language & Environment: You are welcome to use any programming language, such as C/C++, C#, Java, or python, or use different languages to handle different parts. You have to explicitly state what packages or libraries your programs are based on when presenting your work. We do not limit the application's operating system (OS) to work, but we suggest Microsoft Windows.

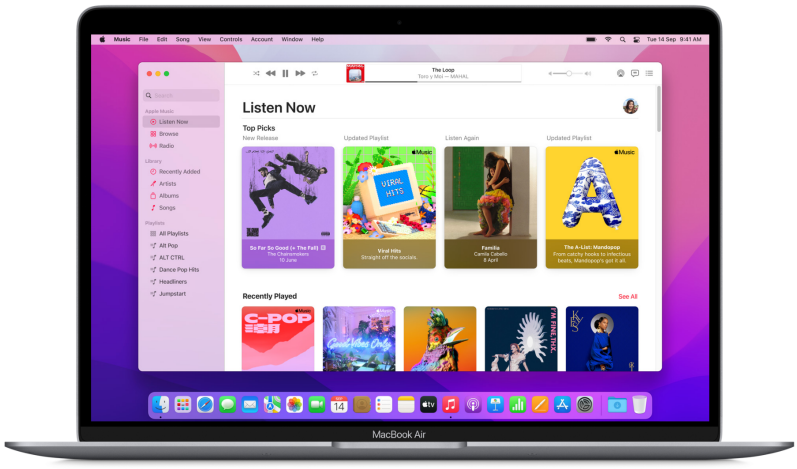
Grouping: The project should be carried out by a group of exactly four students. You can find teammates via Piazza. A group should be formed ***before Feb. 28th***, each group is required to send a message to Yuechen (zhangyc@link.cuhk.edu.hk), including all members' names and student IDs.

Discussions: Discussions are encouraged. Besides discussion within the group, you can ask and answer questions about the project on the Piazza page. As it is an open project in terms of programming language, TA may not help you to debug concrete coding problems.

Evaluation: Your system should fulfill the basic requirements and have enough enhanced features. In Phase I, you need to submit your code and a two-page progress report. After implementing the whole system (Phase II), you are required to upload the latest version of the code and report, demonstrate your system, and give a 12-minute demonstration of your project.

Evaluation Metrics	***Score (100% in total)***
Basic Requirements: Phase I	30%
Basic Requirements: Phase II	30%
Enhanced Features (Phase I & II)	25%
Demonstration & Report	15%

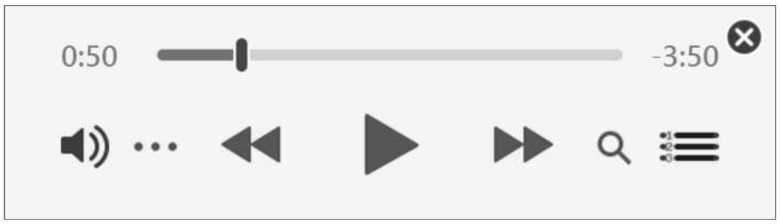
Basic Requirements



Apple Music, https://www.apple.com/hk/en/itunes, is a well-designed music player.

Basic User Interface

Your program shall have a basic user interface. The interface should at least include a play/stop button and a list control of audio files. The user can select the music in the list control to play the audio file. You also need to provide an interface for users to edit the information of audio files.



A demonstration of playback controls. You can add any function as you wish, but the controls should contain at least a play/stop (or a play/pause) button.

Music Decoding and Playback

You are required to understand the inside structure of wave format and write your own codes that can open, analyze, and playback a WAV file. ***The fmt sub-chunk and the data sub-chunk of the WAV file must be read and extracted by yourselves, which means you cannot use any third-party libraries/programs***. The sound data should be played fluently and bring the users beautiful music.

Music Management

Your program should have a database that stores music information (e.g., album, title, length) such that the program can detect the music files in the database and then display them in the playlist. Your program should also have a database (or a text file) to store the information of the audio files. The information of the audio files should be manually input and removed by the user or automatically generated.

```
'1.wav' 'Years' 'Eason Chan' 'Black , White & Grey'
'2.wav' 'Unfortunately, not you' 'Jasmine Liang' 'Silk Road'
'3.wav' 'Our Song' 'Leehom Wang' 'Change their'
'4.wav' 'Unfortunately, not you' 'HU XIA' 'NONE'
```

A demonstration of music info management using a simple .TXT file.

Information Display

Your program shall be able to display the information of the music, including the music title, singer, and album name according to your database. The corresponding information should be displayed when the user plays a song in the list control. The program should display "None" in certain places if certain information is unavailable.

Name	Time	Artist	Album
WHITE ALBUM Live at Campus Fes	4:40	小木曽雪菜	WHITE ALBUM2 Or...
White As Snow	4:12	CAPSULE	WAVE RUNNER
White As Snow	4:12	CAPSULE	Wave Runner (Delu...
White As Snow (extended mix)	5:11	CAPSULE	WAVE RUNNER
White Garden	3:08	Another Infinity fea...	Sakura Luminance
White Is Right	2:00	Pink Guy	Pink Season
WHITE LOVE	6:55	JUJU	Request
White Love / MOMENT	5:38	Speed	Xiami Compilations
White Peak	4:13	xi	RADIAL

A demonstration of the music info display interface.

Music Searching

Users can type in keywords to search music based on your database. Your program can search the music from the music database according to the keywords. The results should be displayed in the list control of your program. What's more, the user could search from any properties of the music, including music title, singer, and the album as the keywords.

Lyrics Display

Your program should be able to play music and show lyrics. The lyrics file can be simple text files or in LRC format. The location of the lyric file can be maintained by the database, or simply place the file in the same folder as the music file with similar names.

Sample Enhanced Features

You are welcome to come up with more creative ideas to enhance your project!

Support for other music formats

You can use any 3rd-party libs/programs or implement them by yourself to extend your music player to support other audio formats, such as mp3, aac, ogg, and others.

Progress bar

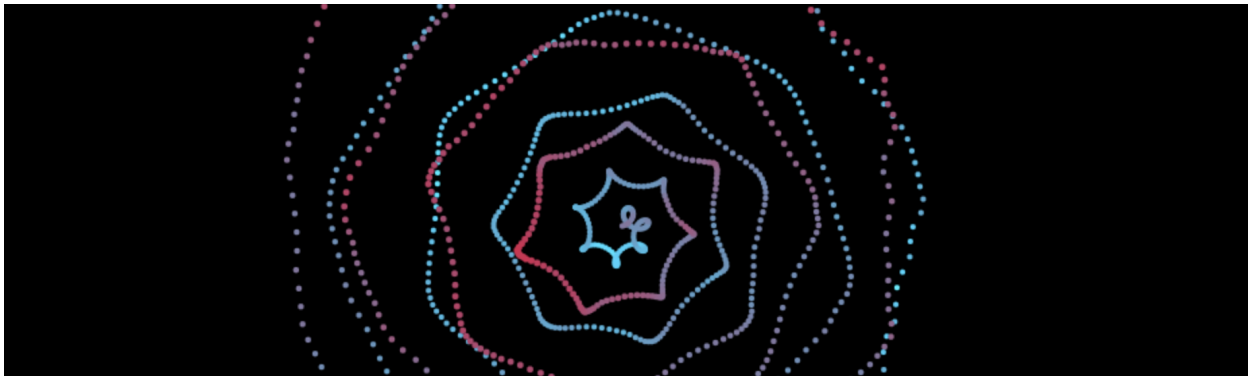
You may implement a progress bar on the GUI to support fast seeking for playback.

Synchronized lyric display

LRC format contains lyrics with the addition of timing information. You may play the music and show the lyrics synchronously. You should download a .LRC file of your favorite song from the internet or type the lyrics text without time-info, and then edit it by yourself.

Visualization

You may create beautiful music visualization effects based on the music. For example, display music spectrum based on Fourier analysis. You may grab some inspiration from here: https://github.com/willianjusten/awesome-audio-visualization



Source: https://soniaboller.github.io/audible-visuals/

Submission Guideline (Phase I)

Each group will get a group number after Feb 28th, you can check your group number on the Blackboard.

Each group is required to submit one .ZIP file (CSCI3280_Group_X_Project_Phase_I.zip), including your source code (with an executable file if available), and a two-page progress report via blackboard **before Mar. 13th 23:59**. In the report, please specify the concrete contribution of each group member, a simple introduction to your system, and describe your implemented features.