# C++ Project Plan—Media Player

Qianqian Qin, Xiaopu Li, Yang Xiao, Zhicun Xu

November 17, 2017

## 1 Introduction

### 1.1 Background

A media player is a computer application used to play multimedia files. Different media players may have different goals and feature sets. While many media players can play both audio and vedio, some others focus on only one media type. Currently there are plenty of media players available for different operating systems, which can be choosen by consumers in the market. Many of them are of rich functions atrracting people. But on the other hand, they usually contain more than users need.

### 1.2 Project Objective

The purpose of this project is to design and implement a media player with the specified features. The player will have the following features:

- Basic features:
    1. User-friendly graphical user interface
    2. Loading .mp3 audio file (possibly include other file formats)
    3. Controlling the audio playback
- Advanced features:
    1. Managing filelist
    2. displaying audio information
    3. Audio visualization
    4. Filters and effects

### 1.3 Project Tools

The group plans to use the Qt library which is often used to develop user interface. It provides a multimedia module which is possibly helpful to develop a media player, but it still needs further research. Some other libraries maybe useful as well, and the group will do some different tries.
The build tool the group will use is CMake. No group member has experience with CMake, so it will be a challenge.

## 2 Architecture

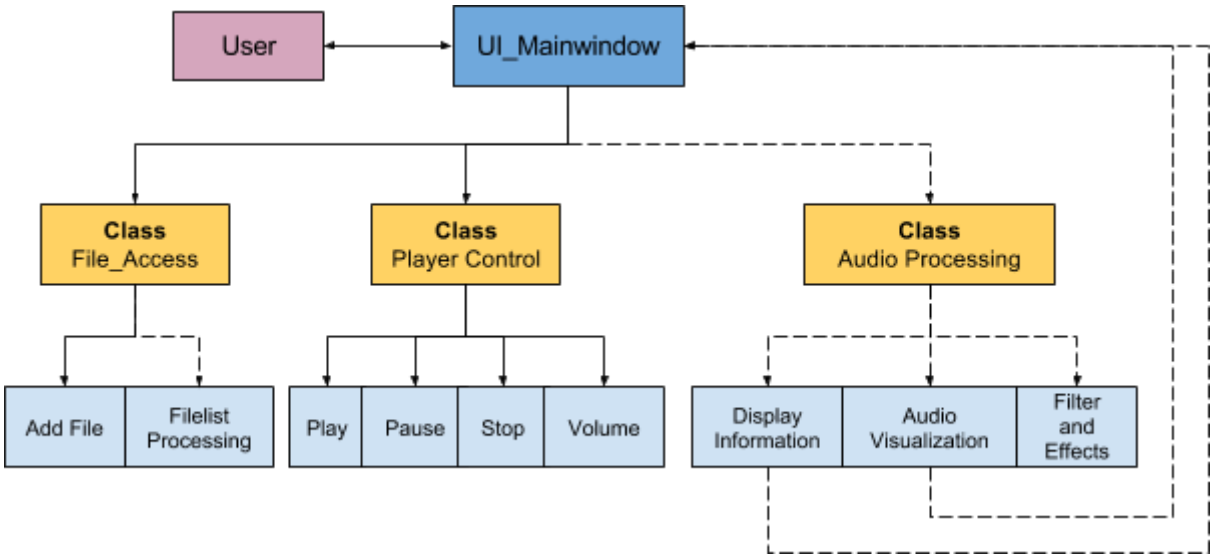


Figure 1: Basic Architecture (Dashline parts are advanced functions)

### 2.1 Graphical User Interface

The Graphical User Interface (GUI) is a space, where users can interact with applications through graphical elements. The team plan to develop a user-friendly GUI using Qt application framework. The connections between GUI and the other classes are realized based on the signals and slots mechanism, which simplifys handling of events, and helps in the development of UI which receives a set of event information and should process them accordingly.

The basic GUI elements of the designed media player includes: File-adding button, play button, pause button, stop button and volume adjustment bar.
The advanced GUI elements contains: information messagebox, graph viewer, playlist table, and some elements related to the class Filter and Effects. whether to add the advanced elements will be decided according to the practical condition.

## 2.2 File access

### 2.2.1 Add file

An audio file can be added to wait for playing. A way to add a file is inputing file name through a input box. It is also possible to load files from external directories, which is a advanced function.

### 2.2.2 Filelist processing

Filelist function can list loaded audio files in an index. It may have grouping capabilities which can group songs by some keywords, such as singers, albums or album art. In other words, filelist function can find the tracks you want, when you want them.

## 2.3 Player control

As the name indicates, player control for controlling the music while playing. The main modules for this section will be play, pause and stop modules, as well as module for volume control just like most media players. After you select your audio file from local storage, you can choose to play the music or maybe a continuous loop function can be choose. And it is possible that you can also choose to play the previous or next file when you click playNext button when the music file is in a list. Of course you can pause and stop playing when click the corresponding buttons.

## 2.4 Audio Processing

### 2.4.1 Display information

This module is mainly for displaying the audio information by reading meta-data such as the artist, album, genre and so on. The major work includes the ways to read meta-data and build a window block to display the information in a certain pattern(sliding through or fixed display).

### 2.4.2 Audio visualization

The visualization is displaying the spectrum or other graphical pattern using the spectrum as the audio plays. The major work is as follows:

- The algorithm to obtain the spectrum information of the audio in real time. Such as using STFT(Short-Time-Fourier-transform)Divide the audio data into relatively small section and apply FFT to get the spectrum information for display. The selection of windowing functions should also be taken into account.
- The display pattern setting: Pure spectrogram is relatively easy by using bars with different height representing the amplitude for the respective frequencies. If time allows, we can use the information to form other graphical patterns or even including interactive modules.

### 2.4.3 Filter and other effects

In this part, we might implement a 10-band equalizer for the media player. Each band representing different frequency from low to high, and the user can adjust the amplitude of each band for his own preference. This might need the algorithm for the band-pass filters. Some preset equalization can also be helpful.

For other effects, we are thinking about implementing some reverb effects with certain presets and basic tuning parameters such as the dry/wet ratio.

# 3 Distribution of Work

| **Task** | **Member(s) Responsible** |
|---|---|
| **Basic** | |
| Architecture Design | ***Qin*** |
| GUI Design | ***Qin*** & ***Xiao*** |
| Add_file function | ***Xiao*** & ***Xu*** |
| Play&Stop functions | ***Li*** & ***Qin*** |
| Pause&Volume functions | ***Li*** & ***Xu*** |
| **Advanced** | |
| Filelist function | ***Xiao*** & ***Qin*** & ***Li*** |
| Display_information function | ***Xu*** & ***Li*** |
| Audio Visualization function | ***Xu*** & ***Qin*** |
| Filters&Effects functions | ***Xu*** & ***Xiao*** |
| **Final Stage** | |
| Debugging&Testing | ***All*** |

Table 1: Workload Distribution

# 4 Schedule

- 10 Nov–17 Nov: Making plan and Researching topic.
- 18 Nov–20 Nov: A prototype of UI and main architecture design will be done.
- 20 Nov–27 Nov: Build basic UI and realize basic functions, such as: **loading files**, **play** and **stop**.
- 27 Nov–1 Dec: Mid-term meeting and feedback. We will have all the basic functions and modules ready for the midterm review.

- 2 Dec–9 Dec: The advanced module will be implemented such as:**audio visualization**, **filters& effects**. Starting to write some parts of the final document.
- 9 Dec–15 Dec: The finish touch of the media player and debugging& testing process for the final demonstration. Final document will be finished before 15 Dec.

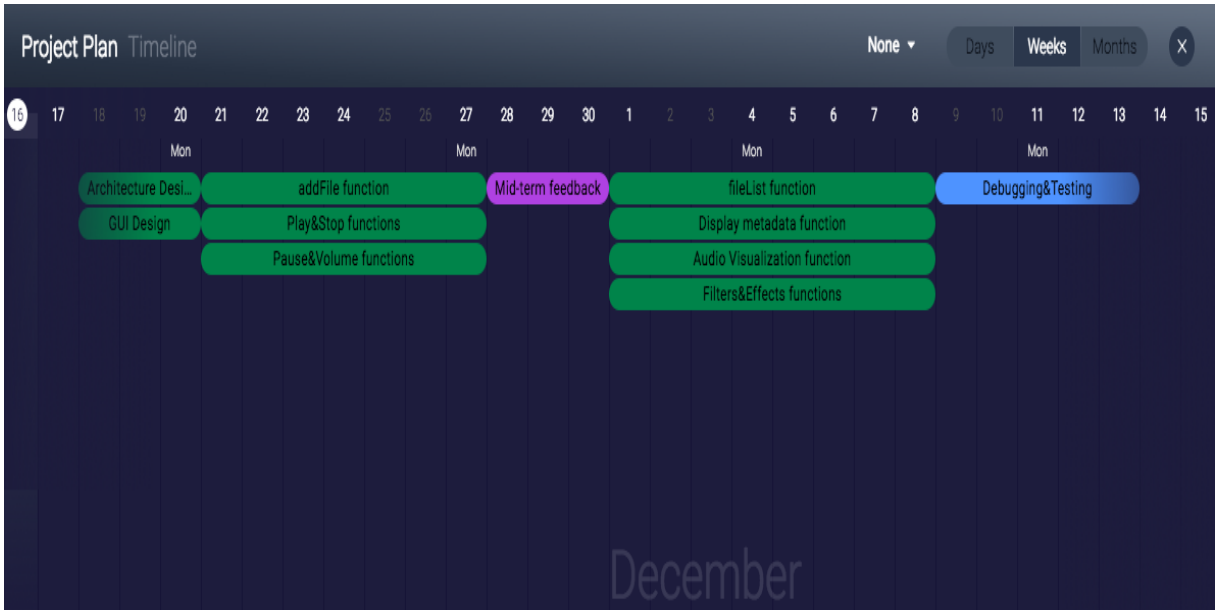


Figure 2: working timeline

# 5 Practical Arrangement

- **Project meeting:** once or twice a week, time and place decided separately among the group members, but frequency can be changed if required. The purpose of meetings include: summarizing previous week's work, tracking progress of the project, sharing information and making necessary adjustments based on actual condition.
- **Materials-sharing tools:** Google Drive, Overleaf, Gitlab
- **Comminication tools:** Wechat