Aalto University

C++ Programming

ELEC-A7150

Media Player

Author:	*Student Number:*
Qianqian Qin	601098
Zhicun Xu	602301
Xiaopu Li	600167
Yang Xiao	598211

December 15, 2017

Contents

1 Overview

In this C++ programming project, our group implemented a media player. The project has been finished quite well following the project plan. Most of the functions planned have been implemented successfully.

The library adopted to develop the project is Qt5. The QMediaPlayer class in the Multimedia module is used to build the player.

The media player has a user-friendly GUI and basic features. The basic functions include Play, Pause, Stop, Next, Previous, Seeking, Volume adjustment and Playback modes selection. The player supports multiple audio formats.

We also add some advanced features including Managing filelist, Real-time audio visualization and Meta-data display. When audio files are added to the player, they will be stored in the play list that is easy to be fetched next time. Of course, user can also manage the play list by adding and removing. The real-time sound visualization is realized by using QAudioProbe class, QAudioBuffer class, relevant FFT class and adopting related algorithm. The meta-data of the audio can be displayed by QMediaMetaData Namespace.

Compared with the project plan, the function not implemented is Filters and effects. The reason why it can not be realized is Qt does not allow to modify audio buffer.

2 Software structure

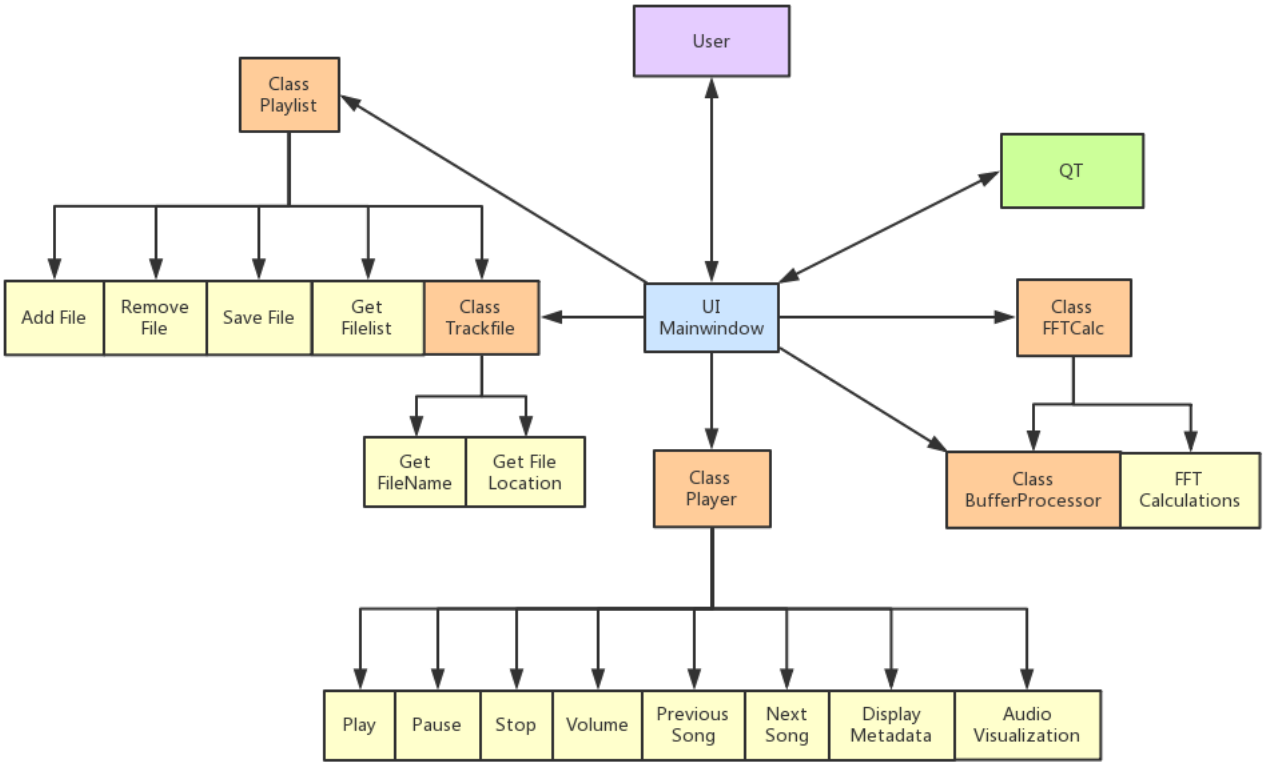

Figure 1: Software Architecture

As can be seen from the figure, the software architecture of the media player consists of three functional modules: the file access module, the song playback module and the audio processing module. The first module is file access which mainly achieves the function of add file, remove file, save file, get file list and load sound track. The song playback module achieves the main function of the software, including play, pause, stop, volume control and two advanced features, display music meta-data and audio visualization. The last module is the audio processing module which is an advanced module, mainly achieving FFT calculations and user can see the audio visualization from the UI.

3 Instructions

3.1 How to run

The needed external library is Qt5. The version used to develop the project is 5.5.1. After downloading the tar.gz file of the project and unzipping it:

```
$ cd src/qt-project/
$ make
$ ./player
```

3.2 User guide

- Button "add"
  - Adding audio file(s) to the playlist
  - If the playlist is not empty, the file(s) will be added below the file currently selected in the list
  - If the file has been added to the playlist, an warning will be given
- Button "remove"
  - Removing the selected file from the playlist.
  - An warning will be given if the file is under playing or paused
- Button "Play"
  - Playing the current selected file in the playlist
  - If the selected file is not playing, the file will be played soon
  - If the selected file is under playing, there is no effect
  - An warning will be given if the playlist is empty.
  - An warning will be given if the file can not be found (e.g. the file has been deleted from local machine or changed its path). (Then you can remove this file from the playlist by clicking "remove")
  - The file name of the current media will be showed in the window title
- Button "Pause"
  - The audio under playing is paused
  - It is disabled when the audio has been paused or stopped
- Button "Stop"

  - The audio under playing is stopped
  - It is disabled when the audio has been stopped
- Button "Next"
  - Playing the next file of the file currently selected
  - If the selected file is the last in the playlist, the first file in the list will be played
- Button "Previous"
  - Playing the previous file of the file currently selected
  - If the selected file is the first in the playlist, the last file in the list will be played
- Slider Volume
  - Moving the slider to adjust volume
  - The default value is 50/100
- Slider Progress_bar
  - Showing the progress of the audio under playing
  - Moving it to seek position
- Combo-box Mode
  - Select one among 5 playback modes: List loop, Single loop, Play in order, Single once, Random
  - The default mode is List loop
- Listwidget Meta-data_list
  - Display the meta-data of the current audio
  - The displayed information includes: Title, Album, Author, Genre
- Visualization_bar group
  - Real-time visualize the audio under playing

4 Testing

4.1 Make

The make can be done successfully by the procedures given. Then a file called 'player' can be run as shown in the third section.

In addition to that, the UI can adjust accordingly when we change the size of the window. The maximum/minimize/close window also works correctly.

4.2 File Access

For the file access, we tested the following aspects:

- Hit the button when there is no sound track in the list:

  The exception is handled well, a window would pop up and declares the error.

- Load and remove sound track

  These commands work successfully, and the loaded sound track can be seen in the small window following the loaded order.

- Load existing file

  The exception is handled well, a window would pop up say there is already a same file.

4.3 Player Control

- Play, Pause, Stop button:

  They all worked successfully, and sound can be heard through headsets.

- Volume slider:

  We can hear that the volume changes correctly as we move the slider for the volume.

- Previous and next song button and play mode:

  Player consists of five modes. They all worked correctly as we testing the previous and next button.

4.4 Audio Signal Processing

- Meta-data visualization:

  The meta-data can be seen in the respective region including the tile, album, author, and genre. We test several songs and they all worked greatly.

- Spectrum Visualization

  The spectrum is shown in real time. The amplitudes from low frequencies to high frequencies all change accordingly. Some bars might frozen for a while, that is due to the scaling function and sensitivity of the signal.

4.5 Valgrind Test

There still existing some valgrind errors but mainly concerning of Qt. There are also some existing memory errors concerning the usage of containers that we haven't resolved yet.

5 Work log

Detailed description of division of work and everyone's responsibilities For each week, description of what was done and roughly how many hours were used, for each project member

5.1 Distribution of work

The distribution of work can be seen below and the respective hours used are shown after the name.

- Qianqian Qin(around 70 hours):

  Software structure

  GUI design

  Implementing player controls and meta-data display in the class Player, including: Play(cooperating with Li), Pause, Stop, Next, Previous, Volume adjustment, Progress_bar showing and seeking, Next, Previous, Playback modes, Meta-data display.

  Making some modification on the function "add" of the class Playlist, making the file can be added in specified location, rather than only can be added at the end of the list.

- Xiaopu Li(around 50 hours):

  Implement the basic file access function, including add file, remove file, save file and read file list.

  Incorporate the playlist class to our implementation.

- Yang Xiao(around 50 hours):

Implementation of the file access and playlist management functions with Li. And adding some extra algorithms like media files cannot be loaded to the list more then once.

Incorporated the playlist class to implementation as well.

- Zhicun Xu(around 55 hours)

  Incorporating the relevant FFT classes to to our implementation.

  The real time spectrum visualization functions. It includes reading buffer from QAudioBuffer class in real time, applying FFT on these time domain signals to get the spectrum information, choosing suitable octaves, and applying scaling functions so that the spectrum can be shown relatively approximately.

5.2 Progress along week

- Week 1(20 Nov. – 26 Nov.): Basic gui design was done and a rough template was created including basic file adding function and play, pause and stop button.
- Week 2(27 Nov. – 3 Dec.): Basic Playlist class was implemented, the file adding and removing were in place. We further improve the player class, adding the previous and next song button into our implementation. We managed to read audio buffer in real time through Qt and sketched out basic algorithm for the spectrum visualization.
- Week 3(4 Dec. – 10 Dec.): Audio Visualization now was implemented successfully. Further UI improvements were made including replacing the text button with certain images. Some bugs were fixed in this week.
- Week 4(11 Dec. – 15 Dec.): We did some final tuning of our implementations. Meta-data visualization was made. Some bugs was fixed. Then we finished our presentation and other relevant documentations.