Aalto University

C++ Programming

ELEC-A7151

Media Player

Author:	*Student Number*
Linming Pan	292096
Yi Zhang	602466
Kejin Zhou	
Yongyu Huang	739627

December 14, 2018

Contents

1. Overview

We have created a cross-platform multimedia player, called Melön Player, with full-fledged functionalities. We build our Melön Player based on Qt to achieve cross-platform-ness. To further reduce the dependence on the tool-chain, we choose cmake as the building tool.

Melön Player contains basic functions including playback control, volume control, playlist management, etc. Melön player supports popular media formats including mp4.

Firstly, we use Qt design the shape and basic function buttons of the player, such as the main screen, playlist, progress control bar, volume control bar, play and pause buttons. Then we use FFmpeg to decode video and audio. Finally, Qt to display video raw image and SDL to play the audio. And because of the heavy workload, this media player can only play video file.

2. Software Architecture

2.1 Graphical User Interface

We plan to use Qt application for GUI. Qt have many ready-made functionalities that we will utilize, like button and slider functionalities. Additionally some of our team members have experiences with Qt, and they can help other members when implementing Qt functionalities.

2.2 Video and Audio processing

FFmpeg is a very powerful SDK, with a set of open source audio and video tools for recording, converting, and streaming multimedia content. It supports nearly every digital format and codec known, from the old and obsolete to the cutting edge.

We use it to decode video and audio as well as adding functions to the player, for example, metadata and media management.

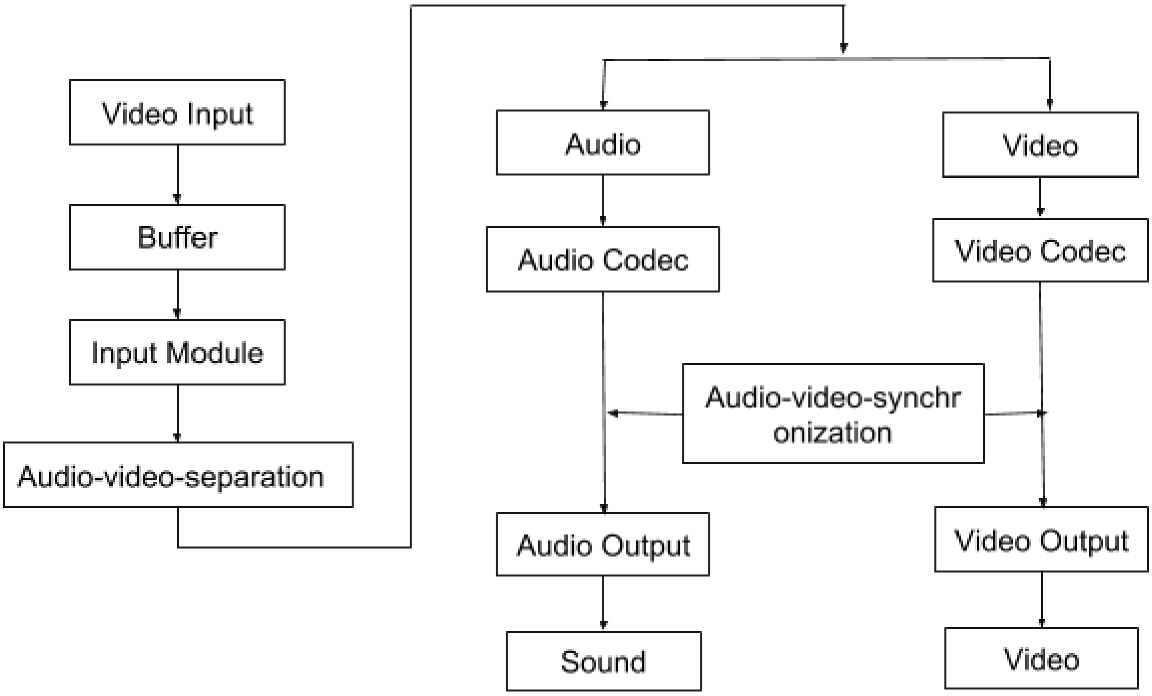

Figure 1 Flow chart of the software architecture

The basic design idea is that we will use multiple threads for the media processing, one thread is for reading the video and audio, one is for decode the video, and SDL will handle the audio play. So the Media processing includes play, pause, stop, seek, sound control etc.

2.3 Media Playlist

File loading, metadata editing, sorting, drag and drop functions are included in this module.

2.4 UML Class Diagram

A UML class diagram of our architecture of the media player is presented in Figure 1

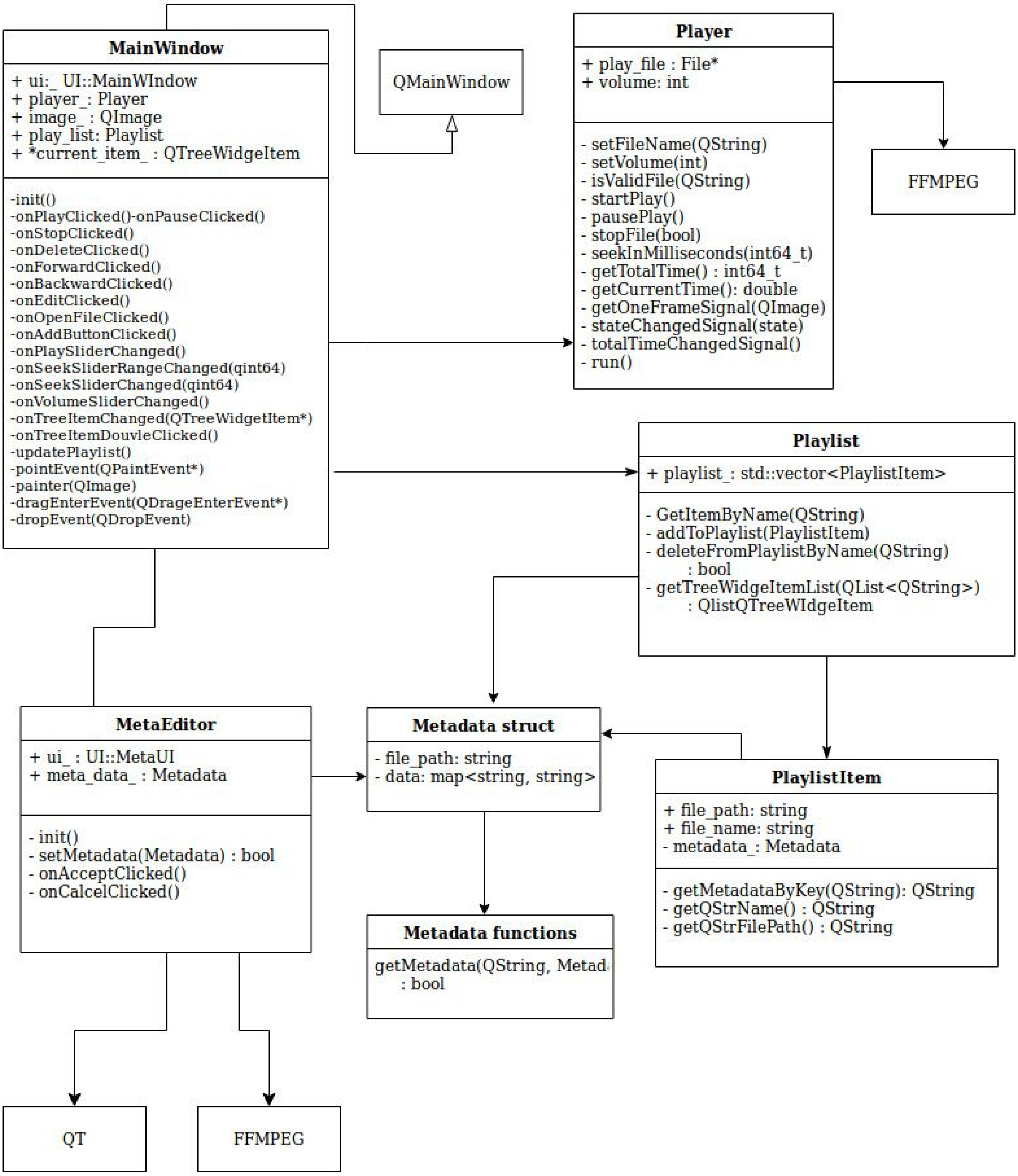

Figure 2 UML of the media player

3. The Dependencies

Qt SDK is a popular choice for implementing user interfaces. It contains a multimedia module sufficient for developing a media player with graphical interface. We base our melon player on Qt.

- FFMPEG

FFmpeg is a free software project consisting of a vast software suite of libraries and programs for handling video, audio, and other multimedia files and streams. We use it to decode video and audio.

- Simple DirectMedia Layer

Simple DirectMedia Layer (SDL2) is a cross-platform library designed to provide low level access to audio, keyboard, mouse, joystick, and graphic. We use it to display the raw image and audio.

- Gtest

Google Tests is a good library for unit testing.

4. How to Build

4.1 Installing Prerequisite Libraries

Other libraries

4.2 Building from Source

We include the paths of the dependencies into CMakeList.txt. Here is a part of the content in the file:

After navigating into the project directory, run the following commands to build the project from the source

```
$ cmake .
$ make
$ ./Gui
```

5. User Guide

- Add button
  - Adding audio file(s) to the playlist
- Delete button
  - Removing the selected item from the playlist.
- Open button
  - Add media file to the playlist
  - Start playing the file
- Volume slider
  - The default value is 50/100
  - Moving the slider to adjust volume
- Seek slider
  - Seek to desired position
- Play button
  - Start playing the file if not playing yet
- Pause button
  - Pauses the player if it is playing
- Stop button
  - Stops the player if it is playing
- Forward button
  - Moves 3 second forwards
- Backward button
  - Moves 3 second backwards
- Drag and drop
  - Support only one file
  - File start playing immediately

6. Testing

6.1 Test Cases

6.1.1. File handling

Test Case	State	Notes
Click open button and open file	ok	Media file plays immediately
Close the opened file without selecting anything	ok	
Open another file when playing a file	ok	
Open file when previous file ended	ok	
Open the same file that is playing	fault	Duplicate the file in the playlist, but player is working correctly
Open the same file that has ended	fault	duplicate the file in the playlist, but player is working correctly
Open a text file	ok	the file is ignored

6.1.2. Player

Test Case	State	Notes
Click buttons when no file is loaded yet (backwards, play, pause, stop, forward)	ok	Nothing happens
Click buttons multiple times when no file is loaded yet (backwards, play, pause, stop, forwards)	ok	
Start playing by opening a file	ok	
Click playing when a file is playing	ok	Continues playing
Click pause when playing	ok	Paused

Click pause when paused	ok	Still paused
Click forward/backward when paused	bug	Not working
Click stop when a file is playing	ok	Stopped
Click play button when a file is stopped	ok	Start playing the same file again
Click buttons when file stopped (backward, pause, stop, forward)	ok	Nothing happens
Seek when no file is loaded	ok	Slider move, but nothing happens
Seek while playing	ok	
Seek while paused	bug	Not possible
Seek while stopped	ok	Slider move, but nothing happens
While playing drag slider from start to end multiple times	ok	Continues from where slider is left
Test volume slider	ok	
Move volume slider to min and start playing new file	ok	No sound
Move volume slider to max and start playing new file	ok	Plays with max volume

6.1.3. Playlist

Test Case	**State**	**Notes**
Click add button and add one file	ok	
Add same file that is already in playlist	fault	Same file can be added multiple times
Close add file without selecting anything	ok	
Add multiple files at same time	fault	Duplicate files possible
Add multiple not working files	ok	No file added to playlist
Add multiple working and not working files together	ok	Not-working files skipped

Sort playlist	ok	Tested all possible options
Click delete while file selected	ok	
Click delete while no file selected	ok	Nothing happens
Delete file from playlist that is playing	ok	Current playing file keeps playing
Start playing file by double clicking it	ok	
Double click list item that do not exist anymore (or name changed)	not tested	

6.1.4. Drag and Drop

Test Case	**State**	**Notes**
Drag and drop one valid file (anywhere inside player)	ok	
Drag and drop one non-valid file to (txt file or other non-valid file)	ok	
Drag and drop multiple files (working and not working)	ok	First valid file is added to playlist and start playing
Drag and drop folder	ok	Nothing happens
Drag and drop same file multiple times	fault	Duplicate file in playlist

6.1.5. Metadata

Test Case	**State**	**Notes**
Click edit button while nothing selected	ok	Nothing happens
Click edit button while file selected	ok	Meta Editor opens
Edit metadata in meta editor and click save	ok	Metadata edited, but may take some time, also playlist item is not updated
Play file after edit	ok	
Click cancel when Meta Editor is opened	ok	Closed without any problem

Click Save without editing any metadata	fault?	Create new file started
Click edit button again without closing previous one	fault?	New Meta Editor is opened
Close main window while Meta Editor is still opened	fault?	Meta Editor is still open
Close main window while Meta Editor is still opened and file playing	fault?	Meta Editor is still open and file keeps playing until it ends

6.2. Memory leak

Valgrind shows memory leak in Qt. The leaks are possibly related to Qt framework itself. We have run empty Qt window with valgrind and is shows memory leaks. Based on this we assume that leaks are related to Qt and we could not detect is there is actual memory leak in out program.

6.3. Playing Audio Files

We do not support playing audio files (mp3 etc.) at the moment. Some audio files may be playable, but there will be some bugs (not able to seek, duration not shown correctly etc.).

7. Work Log

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

7.1 Distribution of Workload

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

Linming Pan:

- Player class
- MainWindow class
- Playlist class
- PlaylistItem class
- MetaEditor class (read metadata)

Yi Zhang:

- Player class
- MediaProcess.h
- MainWindow class
- MetaEdtior class (save metadata)

Yongyu Huang:

- UI design
- Icon design

Kejin Zhou: