Project Plan

Media Player 5

Linming Pan, Yi Zhang, Kejin Zhou, Yongyu Huang

1. Introduction

1.1 Project features

top priority

- audio playback
- video playback
- audio video sync
- GUI with
  1. buttons that control playback: pause, seeking
  2. slider that controls volume

secondary priority

- read metadata and display
- playlist
  - sorting by name
  - Sorting by album name

- **tertiary**

1.2 Implementation outline

We might not follow this steps, it depends on the research in the following days, then we will decide which function to add.

Iteration 1

- the first functional version that can play audio files
- minimalistic, placeholder aesthetics, e.g. all square buttons
- functional file loading, file type recognition
- buttons include load, play, stop, open file
    - error checking for file, does not crash the program if file disappears during playback
- placeholder text display that indicates operation status, e.g. success or failure of loading file, standby/stopped, metadata of file (in a later implementation) ...

iteration 2

- slider for volume control
- progress slider for seeking
- the rest of the buttons
    - pause button, combined with play, icon toggles
    - forward and backward buttons, left-clicking seeks

iteration

- basic playlist functionality
    - Add file to the playlist
    - Delete file from playlist

- metadata reading and display of metadata on the text display

1.3 Project Tools

- QT
  The group plans to use the Qt SDK which is often used to develop user interface. It provides a multimedia module which is enough for developing a media player interface.

- 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.

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

2. 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. 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 obscure to the cutting edge.

We plan to use it decode video and audio as well as add some functions to the player, for example, return metadata and audio visualization.

2.3. File Access and Management

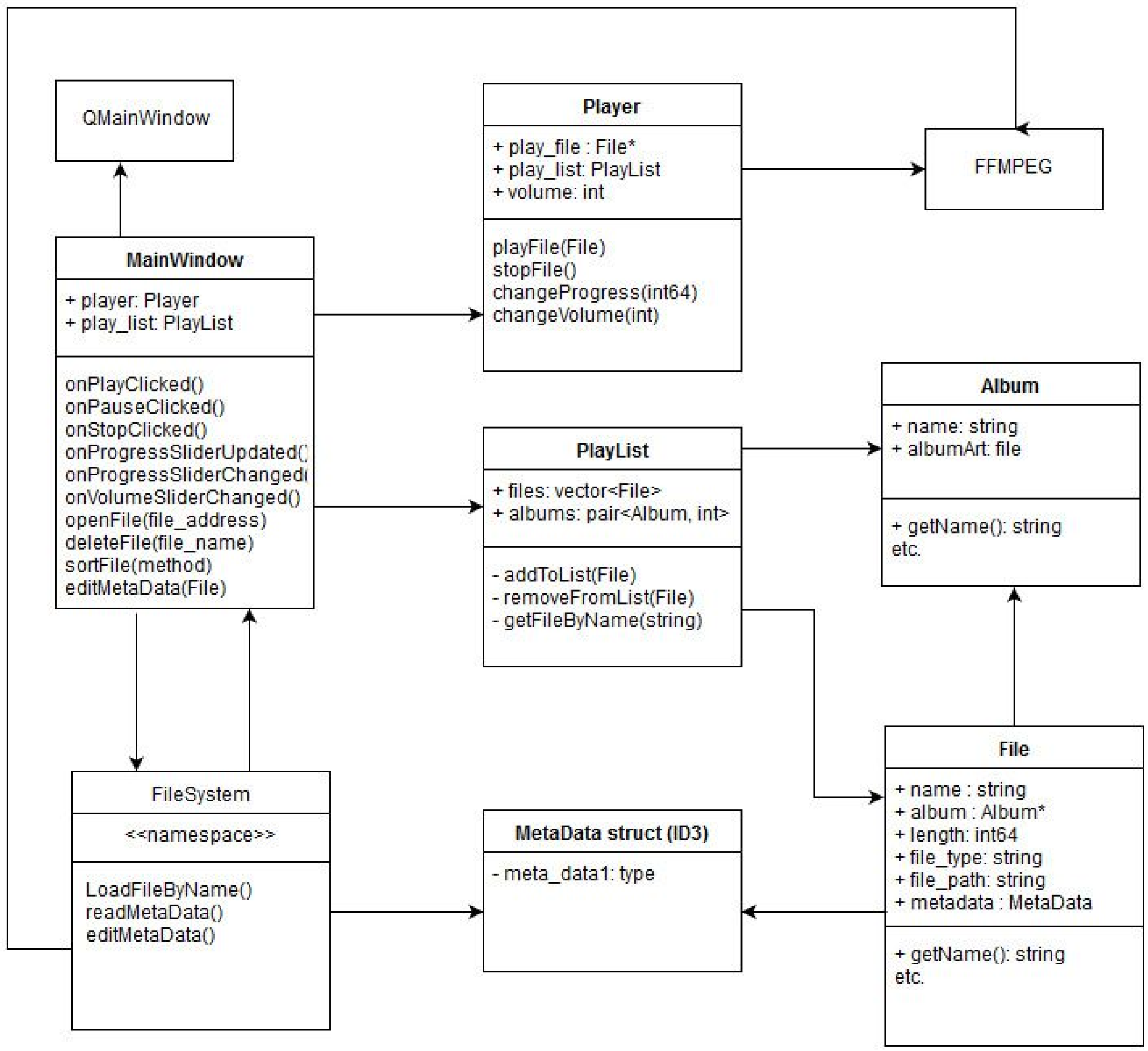
C++ has its own filesystems and we plan to use that to access files. We plan to create a PlayList class to store all files loaded to the program. This class also works as a list for showing all played and loaded files.

We will also create File and Album classes. File class will have information about file: name, length, type etc. If the file is an audio file, a song, it will have an Album class linked to it. Using these classes we will have better management of loaded files.

For modifying and saving audio information (metadata), we will create separate interface for it using ffmpeg library.

2.4. UML Diagram

A tentative version of our architecture of the media player is presented in figure 1. We may change some structure during implementation.



3. Work Distribution

- UI design(Yongyu)
- Qt, buttons and connections(Yongyu)

- Player & ffmpeg & visualization(Yi)
- Metadata & ffmpeg(Yi)

- PlayList, File, Album(Kejin)
- Filesystem & open file in Qt(Kejin)

- Showing metadata in Qt(Pan)
- Video/audio visualization in Qt(Pan)

4. Testing

We will do unit testing for PlayList, File and Album classes. For UI functionalities we have to do testing manually. We will also do memory leak tests.

5. Schedule

9th Nov	*project plan DL*
(2 weeks dev time)	
16th	version 0.5
23rd	version 1.0, possibly version 1.1
19th - 23rd Nov	*mid-term meeting for Q&A*
(2 weeks dev time)	
30th	version 1.4
7th	version 2.1
10th - 14th Dec	demos and finalization
14th Dec	**DL for final commit**
19th Dec	DL for evaluation forms

6. Others

If we are ahead of our schedule and have implement and test the first version of our media player, we will move to second phase and add additional features.

Features we may implement in second phase are:

- Drag and drop support
- Mini-player