

Media Player Pro 2000

Project members:

Kyösti Alkio, 543567

Tomi Panula, 527127

Juhana Autti, 605502

Jussi Impiö, 591946

Scope of the project

Our scope is to have some extra features in addition to the basic features. The amount of extra features will depend on how long the basic ones end up taking time. We hope to have at least a few extra features finished.

Here is a list of the things that we aim to do after we have the basic features done in a priority order:

- Audio file browsing
- Displaying information based on metadata
- Try to get support for a few formats: mp3, wav, ogg
- Audio visualization
- Modifying & saving audio information
- Drag & drop support
- Custom start and end position when looping song?

Major architectural decisions

We have decided to use Qt as our UI framework as it looks most promising. None of us are familiar with it beforehand, but we feel like we will be able to learn it fast enough.

We have not yet decided on what exact audio library we will be using for this project, but here are a few of the options that we have come across.

- SoX
- FFmpeg
- OpenAL
- SDL & SDL_Mixer
- SFML
- FFTW
- Qt's own audio library

We will probably end up using Qt's audio library for basic features, but we will probably try a few different ones if we end up doing audio visualization (as it will need different kind of libraries for example for fast fourier transform).

We will not be doing test driven development. If we do add unit or integration tests, they will probably be added at the end of the project and the test framework will be researched then.

We are not quite sure what classes and templates we will need as we are not sure what is included in Qt. We might end up writing some wrappers for the Qt's audio classes if needed, but we will try to avoid unnecessary layers.

Preliminary schedule

Our goal is to first implement the basic features for the media player. After that, we will start prototyping and testing different kind of features on our TODO list. We aim to have the basic features, the ability to load and play an audio file and basic GUI

with playback controls, finished before the mid-term meetings. After that, we will start working on the extra features.

We will be trying to follow some sort of agile development, where we will try to implement a new feature in a week or two. We will try to have some sort of retrospective each week, where we will look back what each of us has done and re-evaluate if we want to continue implementing current feature or take a new one, and what new feature we want to start implementing next.

Distribution of roles in the group

At the beginning, we will divide in two groups. One of the groups will focus on the graphical user interface and the other will focus on loading and playing audio files. After having the basic features implemented, we will most likely select additional features that we will implement individually or in groups of two, depending on the feature, which we will keep repeating until the project ends. None of the members in our team had a strong desire to work on some specific part of the project, like the graphical user interface, so we will most likely have all members of the team try working on different kinds of features.

We're going to spread the workload of writing the documentation on all the members of the team, so everyone has to do it, or at least one person in each group of two. Because everyone wants to do more of the actual development and not just writing documentation the tasks for writing parts of the documentation, like the overview, software structure and usage instructions, are going to be spread evenly amongst the team members.

We don't currently have a plan to write too many tests for our project because writing tests for UI heavy projects is often quite difficult. We might however have some tests for some of the back-end parts of the project. Since we're at least going to try using some form of agile development to manage our team, we'll most likely be meeting at least once or twice each week for the means of the project. One of these sessions

will probably be used to discuss what was done that week, and what is going to be the plan next week, the work log for that weeks work will be written by each of the members based on that meeting.

Design rationale

We hope to make the design as intuitive as possible. We have some experience from the course Human-computer interaction, but will probably not do any kind of thorough research or PACT-analysis on this project. We will be mostly focusing on making the software work as well as possible.

The project will be focused on object oriented programming. We will not be doing mainly functional programming, but the idea is to have the functions and classes designed in a way, that they can follow functional programming at least in some ways.

Work flow

We will be using git as our version control. Each new feature will be developed in its own branch, and when it's ready a merge request will be made. Another group member will look through the merge request and test it before accepting it. With this kind of peer review, we will be able to maintain project quality. Also if we do have unit tests or integration tests, they will also have to pass for the merge request to be merged.

We will have develop branch which all the features will be merged to first. At certain points, the develop branch will be merged into master. Master branch would be protected, but we don't have permissions to do that on the git repository.

