

## Self review for project

# Media-player 6

Project members:

Kyösti Alkio, 543567

Tomi Panula, 527127

Juhana Autti, 605502

Jussi Impiö, 591946

### **1. Overall design and functionality (0-6p)**

1.1: The implementation corresponds to the selected topic and scope. The extent of project is large enough to accommodate work for everyone (2p/2p)

The implementation corresponds well to the selected topic and has many useful features. The original plan had more than enough features so that everyone in the team could work on something. Audio visualization was changed to video support.

1.2: The software structure is appropriate, clear and well documented. e.g. class structure is justified, inheritance used where appropriate, information hiding is implemented as appropriate. (2p/2p)

Software structure is clear and well documented. Inheritance is used appropriately throughout the whole project with many of the Qt classes and a lot of the methods are hidden as appropriate.

1.3: Use of external libraries is justified and well documented. (2p/2p)

Qt's media library has been used extensively throughout the project and its use has been documented in the project report.

## **2. Working practices (0-6p)**

2.1: Git is used appropriately (e.g., commits are logical and frequent enough, commit logs are descriptive). (2p/2p)

Commits have been frequent when working, with meaningful commit messages. Branches have been used in a clear and efficient manner allowing for a clean, working master branch in repository.

2.2: Work is distributed and organised well. Everyone contributes to the project and has a relevant role that matches his/her skills. The distribution of roles is described well enough. (1p/2p)

The final workload between all team members was not evenly spread.

2.3: Quality assurance is appropriate. Implementation is tested comprehensively and those testing principles are well documented. (1p/2p)

Implementation doesn't have working unit tests which would have helped the testing a lot. A lot of manual testing has been done to ensure all parts are working correctly.

## **3. Implementation aspects (0-8p)**

3.1: Building the software is easy and well documented. CMake or such tool is highly recommended. (2p/2p)

The project documentation has clear instructions for how to build and run the project. The documentation also contains information about how to get rid of the most common problems.

3.2: Memory management is robust, well-organised and coherent. E.g., smart pointers are used where appropriate or RO3/5 is followed. The memory management practices should be documented. (0p/2p)

Memory management is missing and there is memory leakages.

3.3: C++ standard library is used where appropriate. For example, containers are used instead of own solutions where it makes sense. (2p/2p)

The software uses mainly Qt:s own data models. The C++ standard library is not used much as it is not needed.

3.4: Implementation works robustly also in exceptional situations. E.g., functions can survive invalid inputs and exception handling is used where appropriate. (2p)

The software recovers rather well from user mistakes. It doesn't crash when loading invalid files. Qt does some exception handling itself.

#### **4. Project extensiveness (0-10p)**

Project contains features beyond the minimal requirements: Most of the projects list additional features which can be implemented for more points. Teams can also suggest their own custom features, though they have to be in the scope of the project and approved by the course assistant who is overseeing the project. (10p/10p)

Implemented extra features:

- Video support (3p)
- Drag & drop support (2p)
- Displaying audio information by reading metadata (2p)
- Support for many file types like .mp3, .wav, ... (2p)
- Audio browsing & loading from external directories (2p)