Review form for project XXX

Media-player-6: Media Player Pro 2000

**Names of reviewers:
Petri Jehkonen,
Aleksi Liljemark,
Ragnar Lindqvist,
Denis Yeboah**

Provide short comments (2-4 sentences) for each item below.

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)

Project plan is extensive, and scope is large enough to accommodate work for everyone.

* 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)

Structure is directly derived from the Qt and has no implementation of class inheritances. All used objects are derived from Qt.

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

The objectives of project can be achieved with selected libraries of Qt. Documentation is yet not enough, but it is understandable in this phase of the project.

2. Working practices (0-6p)

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

We observed that visible use of git commits had only project plan before 27th Nov. 27th November there were number of commits almost descriptive.

* 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. (2p)

The team under review explained working using “pair-work”. There one pair creates the new code, such that the less experienced developer is key contributor and more experienced acts as an advisor. Then the other pair reviews the code from the first pair. The work flow was well documented.

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

Quality assurance is implemented via pair code development. Testing was not yet planned but at least memory leakages will be debugged.

3. Implementation aspects (0-8p)

* 3.1: Building the software is easy and well documented. CMake or

such tool is highly recommended. (2p)

The design is based on QtCreator and build environment works well between teams.

* 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. (2p)

No memory leakages were observed, as dynamic memory handling implementation did not exist.

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

No C++ standard libraries were used.

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

Implementation was robust, though seek-slider didn't work yet.

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. (0-10p)

On the review the base line of functionality was there, no additional features were implemented yet.