Respondent: **Qianqian Qin** Submitted on: Thursday, 7 December 2017, 11:27 PM

Mid-term review of C++ project

Name of the project group evaluated

media-player-2

C1.1: The implementation corresponds to the selected topic and scope. The extent of project is large enough to accommodate work for everyone (2 p)

The implementation corresepondes to the topic well.basic functions are nicely done, especially the metadata information. It is a good point to realize a playlist which can show metadata and can be sorted.

C1.2: The class structure, information hiding and modularization is appropriate, and it is explained and justified in documentation. The file structure corresponds to the class structure (2 p)

Nice, The class structure is clear. The information hiding and modularization is appropriate.

C1.3: Use of at least one external library (in addition to C++ standard library). Comment the appropriateness of libraries and their use. (2 p)

Qt is well used. The UI design is good. And the Qt internal objects are used approapriately.

There is another library used to build a playlist with extended functionalities. it is a good implementation, even though it needs an extra installation when building the program in Aalto Linux.

C2.1: Git is used appropriately (e.g., commits are logical and frequent enough, commit logs are descriptive) (2 p)

The commits are used approapriately and frequently enough. And the messages are descriptive enough.

C2.2: Make or Cmake (recommended) is used appropriately. The software should build easily using these tools without additional tricks. Nevertheless, instructions for building the project should be provided (1 p)

Make is used and explained clearly in readme file.

It will be more clear to add some linebreaks when introducing how to install taglib.

C2.3: Work is distributed and organised well, everyone has a relevant role that matches his/her skills and contributes project (the distribution of roles needs to be described) (1 p)

The given table explained the distribution of work well enough. Work is distrubted and organised well.

C2.4: Issue tracker is used appropriately to assign new features and bug fixes (1 p)

The issue tracker are good used and informative.

C2.5: Testing and quality assurance is appropriately done and documented. There should be a systematic method to ensure functionality (unit tests, valgrind for memory safety, separate test software and/or something else.) (1 p)

Currently the test stage hasn't begined yet. This part can not be evaluated currently.

C3.1: C++ containers are used appropriately (including appropriate use of iterators), and justified (e.g., why certain type of container over another) (2 p)

There seems that no tranditional containers are used. But Qt library has many its own specific features. The 'Qhash' is used greatly for the playlist functionality. And the Qlist over Qhash is also used.

C3.2: Smart pointers are used in memory management, describe how (1 p)

Seems that no smart pointers are used in the current version.

C3.3: C++ exception handling is used appropriately, describe how (1 p)

The exception handing is used in 'player.cpp' to verify if the media is loaded correctly.

C3.4: Rule of three / rule of five is followed, describe how (1 p)

Destructor is used properly for each module. There is no copy constructor and copy assignment operator. (But maybe they are not necessary in this program)

C3.5: Dynamic binding and virtual classes/functions are used, describe how (1 p)

Vitual classes are used in the 'playlistmodel' class. There are several inherited methods in this class.

Other comments and feedback to the evaluated project group.

The plan is really nice and clean, especially the graphs you made. And you satisfy the majority of the evaluation criteria. Good job!

If you did this review together with (some of) your group members, list the names of the group members here. Everyone needs to turn in a review, either separately or as a group.

Xiaopu Li,

Qianqian Qin,

Yang Xiao,

Zhicun Xu