Respondent: **Matti Hyyrynen** Submitted on: Saturday, 16 December 2017, 10:26 AM

Mid-term review of C++ project

Name of the project group evaluated

q-learning-6

This is a project self-evaluation

☐ Yes

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)

Yes 2/2

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)

Ok 2p

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)

Yes 2p

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

Yes 2p

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)

Ok 1p

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)

Ok 1p

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

Ok 1p

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)

Ok 1p

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)

The vector of raw pointers thing was fishy. It might be demanded by box2d by design though. Otherwise ok.

Also std::vector<std::vector<xyz>> is icky. Do not use it in high perf real time software. Run a performance analysis and you see why. There is a standard way to squash a n-dimensional vector into 1d vector.

2p

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

I did see pointers but not smart pointers. 0p

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

Not used. There is file I/O which definitely requires exception handling. 0p

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

Not followed. cpu_time.hpp for example has destructor but not the other two. 0p

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

Not used. Not needed in this case. 1p

Other comments and feedback to the evaluated project group.

The outcome is excellent considering this was a short sprint of a school project on a cpp basic course.

In the next project please be super careful with pointers, const correctness, right use of references and other c++ creme.

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.

q-learning-7