Review form for project traffic-simulator

Name of project to be reviewed:

traffic-simulator-2019-1

Names of reviewers:

Tommi Gröhn, Kaisa Kärkkäinen, Akseli Oinaanoja, Stas Tatun

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)

The scope of the project seems to have been thought-out carefully so that there was clearly enough work for everyone. The graphical interface is very well implemented. The implementation corresponds very well to the selected topic.

* 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)

The software structure is justified, but there could have been an explanation of what the classes do and what information they inherit and use from other classes in the project documentation in addition to the graph. The class structure makes sense on account of how the classes are inherited.

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

Use of external libraries is justified and well documented in the project documentation.

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)

The commit logs are always descriptive and logical. They have made commits very frequently and regularly and even so all the commits are logical to an outsider as well.

  * 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 distribution of the roles is very well documented and described in the project documentation. Good use of graphs and tables in the documentation of the work log. Everyone clearly had a relevant role in the making of the software.

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

Testing was started near the end of the project, which is easily justified. It was a good idea giving the code to be tested by a different person to whom wrote the code.

3. Implementation aspects (0-8p)

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

Building the software is documented well and on top of that very easy with the help of CMake.

* 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)

Smart pointers were extensively utilized and used appropriately. The destructors seem to be a bit forgotten. There is no clear documentation of the memory management practises.

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

Yes. Containers are used in a sensible matter.

* 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 works robustly in the sense that it doesn't seem to really allow invalid user inputs. So not much need for exception handling.

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)

Project contains many features beyond the minimal requirements. The minimal requirements are also fulfilled. For example, the traffic lights work, the possibility to create your own city map works, the zooming possibility works, and the analysis tools (the heatmap and the drawing of the histogram) are implemented well.