Code Reading Assignment

A good way to learn and improve our programming skills is to read and understand others' programs. Please download the following source code and read the code by yourself. Then write the code-reading report (i.e., CRReport.pdf) about what you have learned during the code reading process, including the background knowledge related to the source code and the programming skills used in the source code. The programming skills may include good OOP design styles, design patterns, template-based generic programming, and any other coding skills that you enjoyed during the code reading process. The report should include two parts, i.e., Part No. 1 and Part No. 2. Please note the following deadline:

(a) Submission deadline: May 21th (Sunday)

(b) Part No. 1: The basic part should include how to compile and run the code, basic OOP-related programming methods, and any specific coding skills (特有的编程技巧).

(c) Part No. 2: The advanced part should include more advanced programming skills, key design principles (设计原理), and what you have learned during code reading. Program flow charts (程序流程图), program structure diagrams (程序结构图), and UML class diagrams may also be included for better explaining the source codes. It is suggested to find some software tools, and learn to use the tools for drawing these charts and diagrams. Please see the attached "Software Design Document.pdf" for more detailed information on how to write a formal document (Please note that some parts are marked as optional, i.e., they are not mandatory).

Mandatory code reading assignment:

Choose one from the following projects (从如下项目中任意选择一个)

(1) Google open source project "LiquidFun": 2D rigid-body and fluid simulation C++ library

(a) Introduction: http://google.github.io/liquidfun/

(b) Download: https://github.com/google/liquidfun/releases

(c) Build & Run: http://google.github.io/liquidfun/Building/html/index.html

(2) Google open source project "protocol buffers"

(a) Introduction: https://developers.google.com/protocol-buffers/

(b) Download:

https://developers.google.com/protocol-buffers/docs/downloads

(3) Boost C++ Libraries

(a) http://www.boost.org

(b) As Boost library is huge, you may first write an overview of each Boost library, and then choose one section of the source code that interests you most. When the interesting section is determined, you may start to read, understand, and then compile and run some examples using the library.

For example, you may choose "Boost Pool Library":

http://www.boost.org/doc/libs/1_60_0/libs/pool/doc/html/index.html

or "Boost.Python":

http://www.boost.org/doc/libs/1_60_0/libs/python/doc/index.html

or "In Place Factory, Typed In Place Factory":

http://www.boost.org/doc/libs/1_60_0/libs/utility/in_place_factories.html

(4) Open source project “TensorFlow” by Google Brain Team:

(a) Website: https://www.tensorflow.org/

(b) Only the reading on the C++ part is needed. The Python part is not required.

(5) Deep learning framework “Caffe” from Berkeley Vision and Learning Center:

(a) Introduction: http://caffe.berkeleyvision.org/

(b) Download: https://github.com/BVLC/caffe

(c) Only the reading on the C++ part is needed. Python or CUDA parts are not required.

(d) Please try to search and find some open-source demo example using Caffe framework. Then compile and run the example to have fun.

TIPS (提示): You may want to install some software to aid the code reading, e.g., Understand, SourceInsight, etc. Please search the software using Google/baidu, then download and install the software by yourself.