

CPPLII : S1 : DÉMINEUR : MÉTIER ET CONSOLE : NVS

Le présent document spécifie les contraintes supplémentaires à celles déjà émises dans le texte de l'énoncé du [projet Démineur](#) que doivent suivre les étudiants dont j'ai la charge pour ce qui concerne la partie métier du projet ainsi que l'application console demandée.

Date et modalités de remise des sources finaux et du rapport

Les classes métier et la partie console (2^e remise) du projet Démineur doivent me parvenir pour le vendredi 30 mars 2018 à 18h00 au plus tard. La remise doit se faire par courrier électronique. Utilisez mon adresse nvansteenkiste@he2b.be. Le plus simple est de me fournir un lien vers une archive dans votre Google Drive @he2b.be.

Cette archive contient :

- une archive 7z du projet Qt Creator avec les classes métier et l'application console ;
- une documentation html générée par Doxygen1 ou un outil alternatif2 ;
- un rapport au format pdf.

Contenu du rapport

Le rapport doit contenir :

- une liste des écarts, ajouts, précisions par rapport à l'énoncé. Veuillez également à *indiquer précisément tous les éventuels bogues restant dans les classes métier ou dans l'application console*. En effet, un bogue non signalé reflète soit un travail de test et de débogage insuffisant, soit une tentative de dissimulation d'un vice caché ;
- le relevé des problèmes rencontrés lors de la mise au point des classes métier et de l'application et les solutions qui ont été implémentées ;
- une justification pour chaque avertissement (*warning*) produit par gcc ou cppcheck (voir plus bas) restant dans le projet ;
- une estimation du temps passé sur cette partie de l'énoncé.

Contraintes sur les sources

Dans les fichiers d'en-têtes, favorisez les [déclarations anticipées](#)¹ et n'incluez les fichiers d'en-têtes que dans les cas où c'est strictement nécessaire : héritage, modèle de classe, attribut objet et pointeur intelligent.

Veillez à ce que votre code soit portable d'un système d'exploitation à un autre :

- faites attention à la [casse dans les noms de fichiers](#)² ;
- utilisez [le séparateur « / »](#)³ entre les répertoires dans les chemins d'accès d'inclusion de fichiers.

Votre code :

1 <https://stackoverflow.com/q/2059665>

2 <https://stackoverflow.com/a/1951969>

3 <https://stackoverflow.com/a/5790259>

- ne doit provoquer aucun avertissement (*warning*) quand compilé avec [gcc](https://gcc.gnu.org/)⁴ et l'option [-Wall](https://gcc.gnu.org/onlinedocs/gcc-6.3.0/gcc/Warning-Options.html#Warning-Options)⁵ ;
- doit passer la validation statique de code par [Cppcheck](http://cppcheck.sourceforge.net/)⁶ avec les [options](http://cppcheck.sourceforge.net/manual.pdf)⁷ :
`--enable=all --inconclusive --language=c++ --std=c++11`

N'hésitez pas à utiliser le plugin [QtCppcheck](https://sourceforge.net/projects/qt-cppcheck/)⁸.

Réalisez des tests unitaires. Il existe de nombreuses usines à tests unitaires en C++. Utilisez celle de votre choix, sachant que [Catch](https://github.com/philsquared/Catch)⁹ est une des plus populaires et [doctest](https://github.com/onqtam/doctest)¹⁰ semble sympathique. Vous devez remettre les tests que vous réalisez.

Vous ne pouvez utiliser que la bibliothèque standard du C++17, telle que prise en charge sur les machines de l'école, à l'exception, éventuellement, pour ce qui concerne l'extraction des données des fichiers de sauvegarde. Vous pouvez en effet adopter un format de donnée standard ([XML](https://stackoverflow.com/questions/9387610/what-xml-parser-should-i-use-in-c)¹¹, [JSON](https://www.reddit.com/r/cpp/comments/2xlowj/im_looking_for_json_parser_c14_header_only_no/)¹², etc.) et utiliser un parser tout prêt, à condition qu'il soit libre.

Points sensibles

Lors de la lecture des sources, je serai particulièrement attentif :

- à la bonne gestion des ressources système au sein de l'application, c'est-à-dire de la mémoire, principalement ; utilisez [Valgrind](http://valgrind.org/)¹³ pour vous assurer que tout est OK de ce côté ;
- à la bonne construction des classes métier : il doit être impossible de tricher ou de se tromper lors de l'utilisation des modificateurs du jeu ; chacun d'eux doit réaliser l'entièreté de la tâche qui lui incombe : vérifier que l'action est permise, la réaliser, la terminer.

4 <https://gcc.gnu.org/>

5 <https://gcc.gnu.org/onlinedocs/gcc-6.3.0/gcc/Warning-Options.html#Warning-Options>

6 <http://cppcheck.sourceforge.net/>

7 <http://cppcheck.sourceforge.net/manual.pdf>

8 <https://sourceforge.net/projects/qt-cppcheck/>

9 <https://github.com/philsquared/Catch>

10 <https://github.com/onqtam/doctest>

11 <https://stackoverflow.com/questions/9387610/what-xml-parser-should-i-use-in-c>

12 https://www.reddit.com/r/cpp/comments/2xlowj/im_looking_for_json_parser_c14_header_only_no/

13 <http://valgrind.org/>