Team P code review report

Ondrej Mosnáček, Lenka Kuníková and Ľubomír Obrátil

17.12.2015

Style issues

- **Player.h**
  - Methods used only by class are public (should be private).
- **Player.cpp**
  - Use of exit(1) - parser can't be used with other interface/inside other application.
  - Code duplication in methods checkString and readInt - same methods are in Team.cpp.
  - Heavy use of hardcoded strings and magic constants - bad readability and prone to errors.
- **Team.cpp**
  - Use of exit(1), code duplication, magic numbers.
  - All error messages are almost same, need for manual stepping over application to find out where an error is issued.
  - Members of class are initialized in the body of constructor instead of initializer section.
  - Duplicate comparisons in elaborate if's
- **Match.cpp**
  - Same issues as in Player.cpp and Team.cpp.
- **main.cpp**
  - Hardcoded output file name.

Performance and portability issues

- **General**
  - Using #pragma once even though it's not standard.
  - Extensive unnecessary substring copying.
- **Team.h**
  - Wrong case in #include directives, can't be built on *nix system.
  - Wrong usage of angle brackets vs. quotes in includes.
- **Team.cpp**
  - Use of function std::stoi (part of C++11 standard), yet project report doesn't mention use of C++11
- **main.cpp**
  - Program will work only for files, not for other streams or I/O devices.

Bugs and crashes

- **UTF.h**
  - Some byte sequences that should be rejected are accepted (2 byte characters can be padded to 3 bytes and are still accepted)
  - IsForbiddenUTF8Char method has no purpose as all comparisons will be always false. Char variable is compared to constants that are bigger than 0x7F.
- **Player.cpp**
  - In method findAndCheck - end of string is never checked, some inputs will cause reading behind buffer, parser will either crash or raise exception.
  - Method parse will reject any valid input that has no whitespace before curly brackets
- **Team.cpp**
  - Again, no bound checking for index and direct access of string elements through this index - causes crashes and exceptions in method findAndCheck.
  - Method checkString should reject unescaped control character but won't do so. Will accept binary zero (or other characters) in any string, will accept newline character if it is preceeded by backslash - this has no effect, newline is still there. The \uXXXX escape defined by JSON standard is completely ignored. Forward slash is rejected but shouldn't be.
- **main.cpp**
  - bad_alloc is not handled if it is raised during parsing

Overall

Even though parser is fairly short (+-500 lines), it contains a lot of potentially dangerous bugs. Big portions of this bugs could be avoided by more thorough testing on malformed inputs. Code alone is rather hard to read as it features a lot of inconsistencies, conditions and hardcoded values.