Sprite Editor Code Standards

By Allen Shumway, Michael Sullivan, Soli Yan, Emile Goulard, Stefan Mada
Date: 11/4/2022

Format Standards:
- Classes and namespaces are named with Pascal Case i.e. `ThisIsMyClassName` or `ThisIsTheNameSpaceWeWant`
- Methods will be named with camel case format like: `methodName`
- Consistently used "StudlyCaps" aka `camelCase` to indicate separations within a word rather than using `snake_case` or `kebab-case`. The only exception to this rule being C or C++ `MACRO_DEFINITIONS` and global constants which should always be uppercase with underlines separating the words as shown above.
- Opening brackets for classes, methods, loops, if / else statements are present on their own line, unless the loop or if check is contained within one line (including nested statements), in which case there are no brackets.
- Indentation will be four spaces (one tab) and whitespace standards are currently subject to code review by the team.
- Space between methods will be two lines apart. Subject to code review by the team.

Files:
- Files should be the same name as the class name.
1) Header Files *(.h)*
    - Should always **start with a header guard** (`#ifndef HEADER_FILENAME_H`) below the comment block.
    - Should only `#include "other_header_files.h")`
        - And **should contain** `#include <QTypes> or <directives>`
2) Source Code Files *(.cpp)*
    - **Should only** `#include "other_header_files.h"`
    - **Should not** `#include "any_source_file.cpp"` or `<QType>`

Comments:
- Files: Comment block at the top with the following topics in this order…
    - File name, author, assignment name, group name, date of project submission, course name, copyright, honor pledge, summary of file.
- Methods: Comment block above that explains the following topics in this order…
    - Brief summary, parameter 1, parameter 2, etc parameters, return, thrown exceptions.
    - Note: This is the same notation that's autogenerated by QT
- Code: Every method should have at-least some in-text comments, no comments are a red flag. (exceptions are made for particularly small/short methods)

Subjective/Miscellaneous:
- `const` all variables by default, unless not possible
- `const` all member functions by default, unless not possible
- Objects (exclude primitives) are passed `const &reference` by default, unless not possible