Code Standards

By Robert Li, Wenlin Li, Qingwen Bao, Julian Zhou, Qimeng Chao
Created: Wednesday, March 30, 2022
Updated: Friday, April 1, 2022
A7: Sprite Editor Implementation

Goals of study guide:

The goal of code style guide standards is not only to allow coders to have a standard code style to make code clear but also to allow the codes to be more readable and easy to understand since it is a teamwork project, by this way, we can reduce the coder's **friction and maintainability**.

The most important goal of implementing code standards is to write code more **consistently** to make the code **robust and have low defects**, codes should keep the code simple, necessary and working. The standards also should be used when building new source files. If files are not broken, do not "fix it", always **keep it working**.

General Rules:

- No Compiler Error or Warnings!
- Eliminate duplicate code by refactoring as necessary
- Keep the consistency. Keep the code readable and easy to follow as needed. If code cannot be written as clearly, write clear comments to explain. The helper method should be written as needed.
- After you finish any implementation, always review the code and source file.

Format Standard:

- classesarenamedlikethis
- variableLookLikeThis;
- methodName()
- Use tab for indentation
- Only one extra blank between each methods

Naming:

Always choose the name that describes the purpose, never use something like var, foo(), class1. The choosing the easy-to-remember and descriptive to avoid unnecessary misunderstanding, also to fit the C++ style.

Comments:

Comments are generally a good thing. If you think your code needs a comment, it probably does. Use "/**" for multiple lines and "//" for single-line comments. Delete the commented code at the final version or after discussed.

The rule of comments doesn't have duplicate comments and too many detailed comments. Comments should be concise, powerful, and easy to understand.

Source files Check-List		
Files Name	Author(s)	Checkers/Modifiers
drawingbroad.cpp and header file	Wenlin, Robert	All other members
mainwindow.cpp and header file	Wenlin	All other members
previewwindow.cpp and header file	Qingwen	All other members
model.cpp and header file	Qimeng, Julian	All other members
main.cpp and header file	Robert	All other members
framearea.cpp and header file	Qimeng, Julian	All other members
sizedialog.cpp and header file	Qingwen	All other members