CODE STYLE SPECIFICATION 28th March 2023 15% Sprite Editor

OVERVIEW

Our goal is to write our goods in a straightforward and consistent way. To do this, our code will be descriptive and not redundant. Along with the code, our descriptions of the code will be concise.

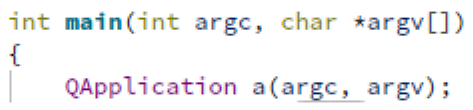
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

General Rules

1. Method comments go in .cpp, not .h. Comments should explain code that is not self-explanatory.
2. DRY! Put duplicate code into helper methods
3. All variables and methods should be self-explanatory and not vague
4. Code should be maintainable
5. Adhere to QT creator auto format

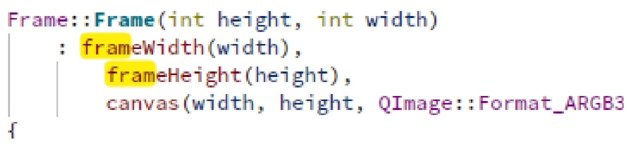
Code formatting

- Indents SHOULD be 4 spaces (per Qt Creator's default autoformat)
- Opening curly braces SHOULD be placed on a new line, for example:



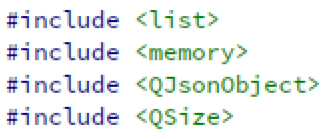
```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
```

- Initializer lists SHOULD be split into one member variable per line with commas and colons as shown:



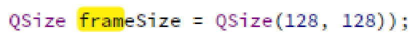
```
Frame::Frame(int height, int width)
    : frameWidth(width),
      frameHeight(height),
      canvas(width, height, QImage::Format_ARGB3
{
```

Includes SHOULD be alphabetized. Optionally, includes with quotes ("") and angles (<>) can be in two separately alphabetized lists



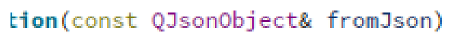
```
#include <list>
#include <memory>
#include <QJsonObject>
#include <QSize>
```

- Default parameters SHOULD be spaced out as such:



```
QSize frameSize = QSize(128, 128));
```

- "const" and "&" modifiers SHOULD be spaced out as such:



```
tion(const QJsonObject& fromJson)
```

Identifier formatting

- ClassesAreNamedLikeThis
- methodsAreNamedLikeThis()
- variablesAreNamedLikeThis