Custom Code Style Document for Sprite Editor Project

General Rules:

1. Refactoring reused code to helper functions.
2. Show and comment hard to understand part in you code with comments.
3. Clear use of Const if possible for safty

Comments:

Using /\*\* \*\*/ for file header

Using good naming to replace most comments.

Using // for comments, aligned then if possible especially in .h file.

In method Comment come after code first unless no space, method comments directly above method.

Format Standards:

Using initialization list for constructors if possible (with indent on each paremeter)

Spacing: using Tab for indent first.

Space a new line between code chunks for better readability

For Pointers use -> as . no space needed

Ex:

Function();

for (const QPixmap &frame : frames) {

}

File Header put all includes in .h files if possible

Ex:

/\*\*

\* File Comment

\*\*/

#ifndef CLASS\_H

#define CLASS\_H

#include <QClass>

#include "CustomClass.h"

#endif // CLASS\_H

In .h file list sequence as public then private variable, variables then functions then slots, and signals:

class Sprite

{

private:

vector<QPixmap> frames;

public:

Sprite();

};

Slot name using on\_clearButton\_clicked(); for same style look.

Variables: int currentAnimateFrameIndex;

Functions: void addOnFrameBar();

Multiple line phrase:

spriteEditor->getCanvas().scaled(

ui->canvasLabel->size(),

Qt::KeepAspectRatio,

Qt::FastTransformation

)

If else:

If(condition) {

return;

}

else {

return;

}