Qt Smart Graphics

Documented by KKyang

Introduction

A simple library that improves the user experience when displaying images by Qt's QLabel and QGraphicsView. It also supports OpenCV Mat array type.

qsmartGraphicsView supports show image functions which supports QImage and Mat (in BGR format). It also supports mouse clicks, moves, and wheel events.

qsmartLabel supports show image functions which supports QImage and Mat (in BGR format). It also supports mouse clicks events.

Author

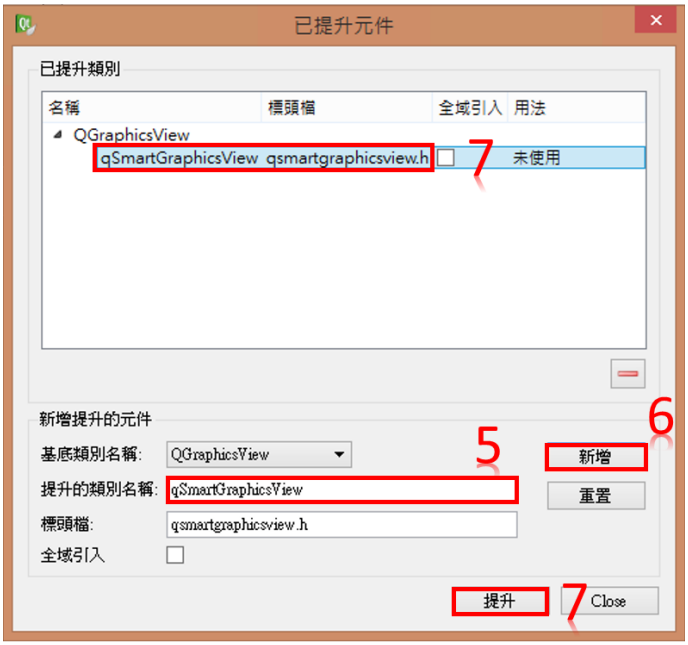
Original by Shih-Jhong Yu, Revised by KKyang. If there is any bug found, feel free to open an issue at Github.

License

Usage

The tutorial shows how to use the library by Qt Creator

1. Put qsmartgraphicsview.h, qsmartgraphicsview.cpp, qsmartlabel.h, qsmartlabel.cpp into your Qt project.
2. Go to mainwindow.ui
3. Add a new QGraphicsView or QLabel in Creator.
4. Right click the widget and choose “promote to…”
5. Add the promote widget by entering the name of the class in the line edit at the button.



6. Click “Add”
7. Click the added class then press promote.
8. Open main.cpp
9. Add #include <QReadWriteLock> and QReadWriteLock lock;

   It should look like this:

```
#include "mainwindow.h"
#include <QApplication>
#include <QReadWriteLock>
QReadWriteLock lock;
```

10. If want to use with OpenCV, simply add **DEFINE += HAVE_OPENCV** to .pro file.
    (Currently only qSmartGraphicsView needs to add this line.)
11. Done.

Functions

qSmartGraphicsView

public

`explicit qSmartGraphicsView(QWidget *parent = 0);`

Constructor of the class.

`void initialize(const int _img_num, const int width, const int height, int changeRow = 4);`

Before using this class, you must initialize it first. You also have to initialize when you change the number of images input. The last parameter changeRow lets user able to change row if the input images are too much. Default number is 4.

`void setImage(const cv::Mat &img);`

Put image into QGraphicsView by a single mat.

`void setImage(const std::vector<cv::Mat> &imgs);`

Put a set of images into QGraphicsView by several mat arrays.

`void setImagefromQImage(const QImage &qimg);`

Put image into QGraphicsView by a single QImage.

`void setImagefromQImage(const std::vector<QImage> &qimgs);`

Put a set of images into QGraphicsView by several QImages.

`int getImgNum(){return img_num;}`

Get current images number displayed in the widget.

signals

`void sendItemMouXY(const double x, const double y);`

Send the mouse coordinate of the image when double clicked the image.

`void sendMousePress();`

Send mouse signal when the widget is pressed.

qSmartLabel

public

`explicit qSmartLabel(QWidget *parent = 0);`

Constructor of the class.

signals

`void sendMouXY(double x, double y);`

Send the mouse coordinate of the image when clicked the image.

qSmartLabel currently still use setPixmap to pass an image file to QLabel.

Both class supports mouse right-click to save the images shown in the widget.