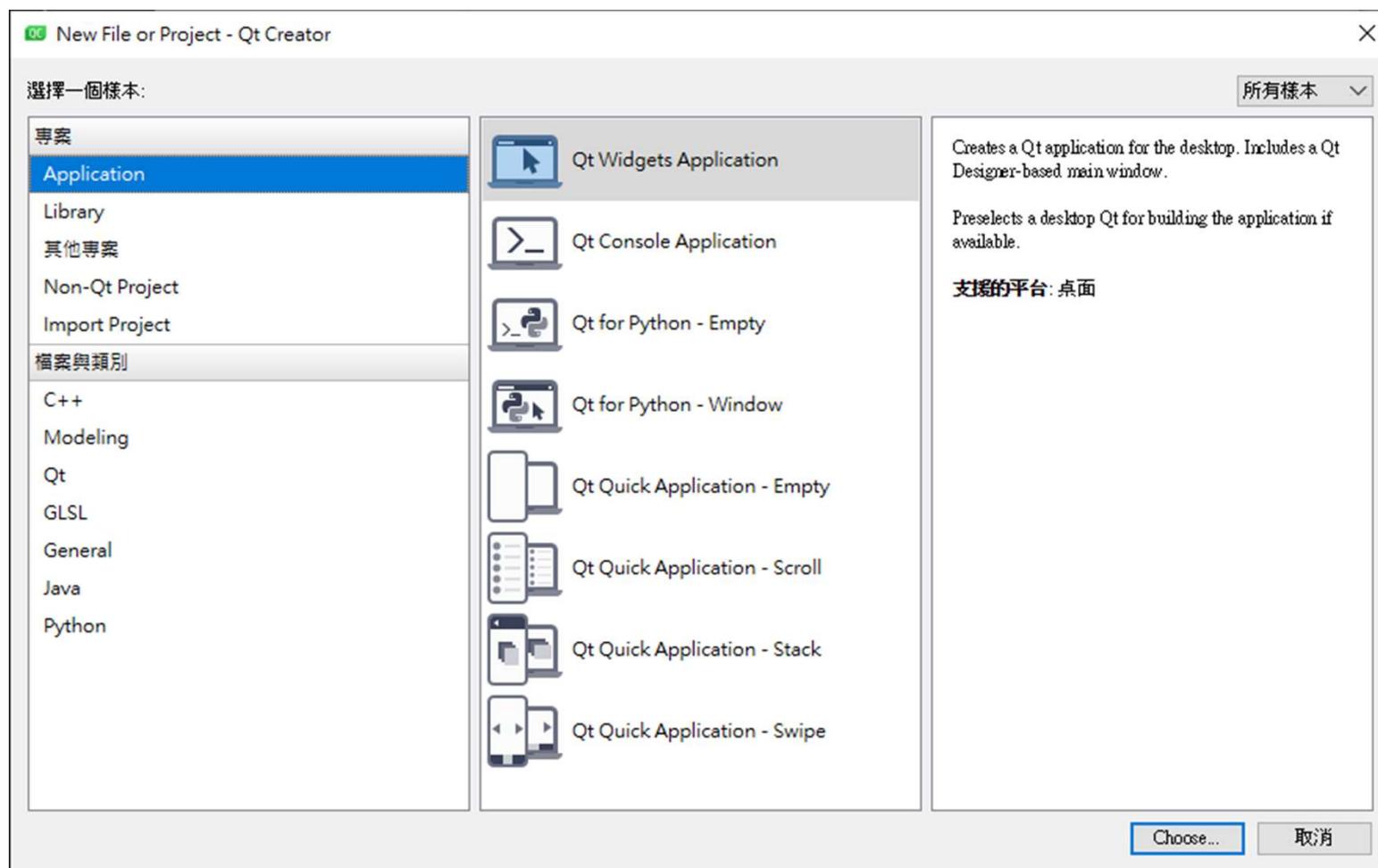


# QT 影像處理

利用QT建立一個基本的影像處理程式

## 1. 建立專案



X

## Qt Widgets Application

### Location

Build System  
Details  
Translation  
Kits  
Summary

### Project Location

This wizard generates a Qt Widgets Application project. The application derives by default from QApplication and includes an empty widget.

名稱 :

建立於 :

做為預設的專案位置

/course/101001001/c/course\_qt\_ImagePcocessing

×

←  Qt Widgets Application

Define Build System

Location

 Build System

Build system: qmake



Details

Translation

Kits

Summary

下一個(N)

取消

X

← Qt Widgets Application

Class Information

Location

Build System

Details

Translation

Kits

Summary

Specify basic information about the classes for which you want to generate skeleton source code files.

Class name:

Base class:  ▾

Header file:

Source file:

Generate form

Form file:

下一個(N)

取消

X

← Qt Widgets Application

Translation File

Location

Build System

Details

Translation

Kits

Summary

If you plan to provide translations for your project's user interface via the Qt Linguist tool, please select a language here. A corresponding translation (.ts) file will be generated for you.

Language: <none>

Translation file: <none>.ts



下一個(N)

取消

X

← Qt Widgets Application

Location

Build System

Details

Translation

Kits

Summary

Kit Selection

The following kits can be used for project **qfImageProcessing**

Type to filter kits by name...

Select all kits

Desktop Qt 5.14.0 MinGW 32-bit

More

詳情 ▾

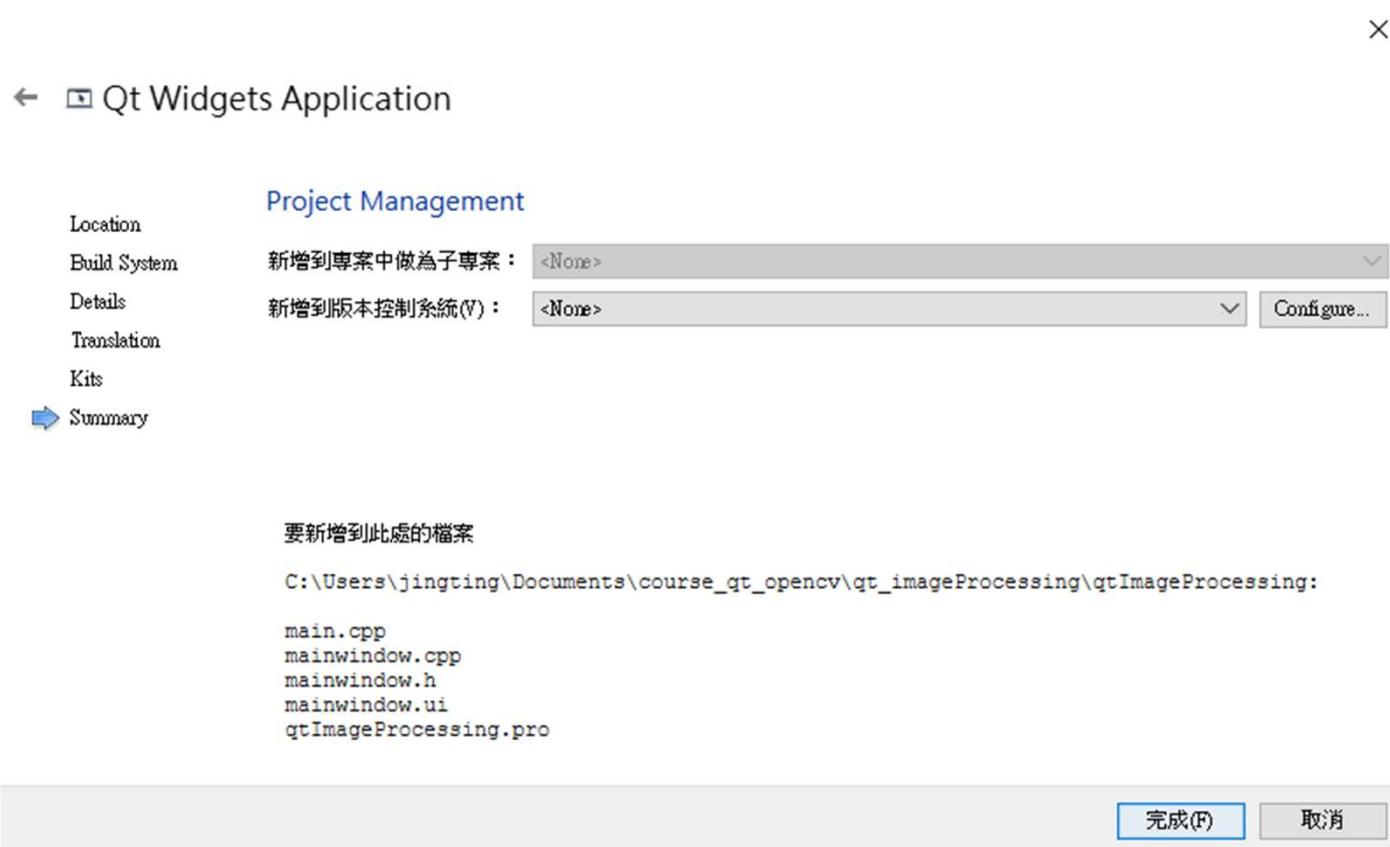
Desktop Qt 5.14.0 MinGW 64-bit

More

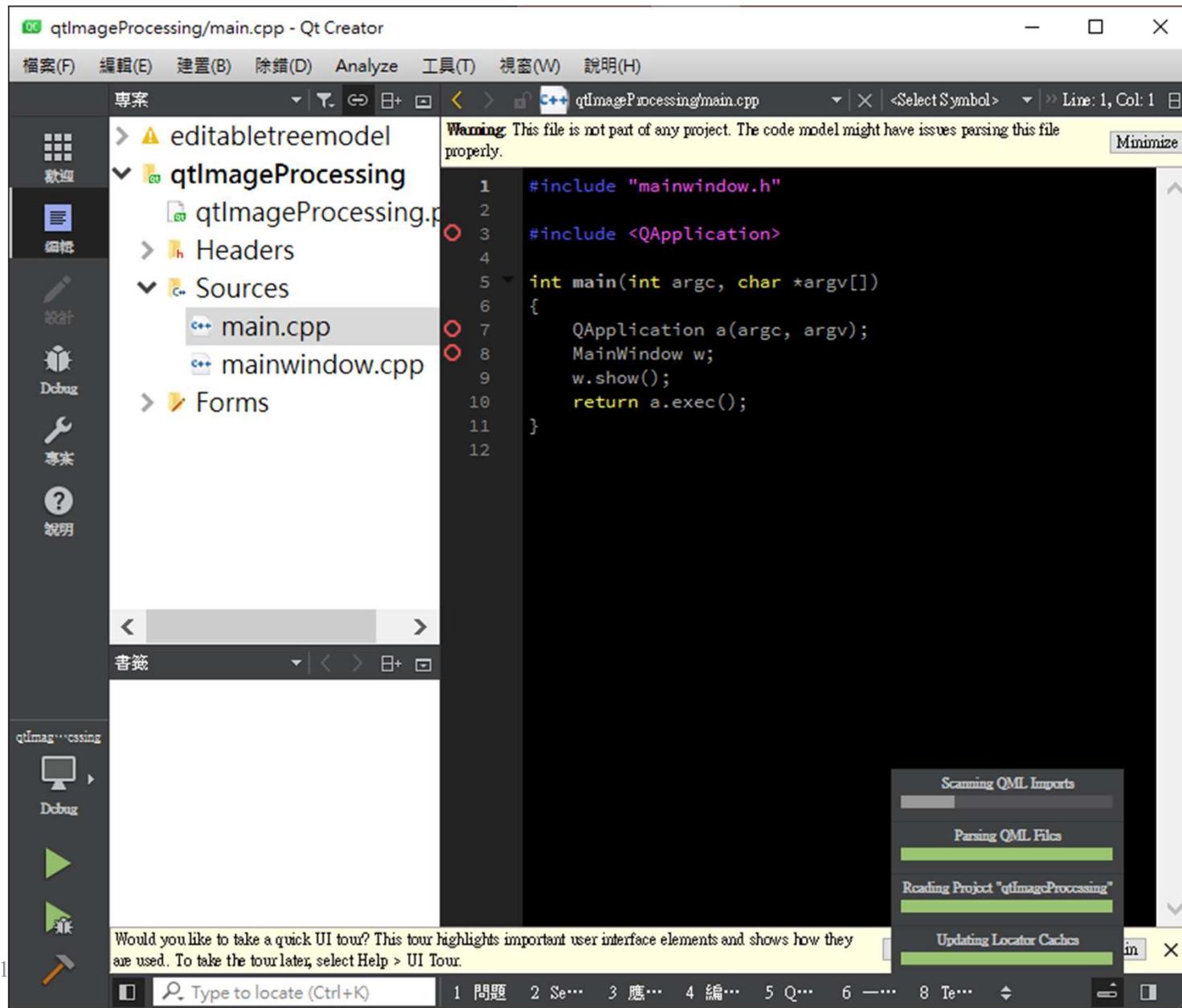
詳情 ▾

下一個(N)

取消



/course/101001001/c/course\_qt\_ImagePcocessing



mainwindow.ui @ qtImageProcessing - Qt Creator

檔案(F) 編輯(E) 建置(B) 除錯(D) Analyze 工具(T) 視窗(W) 說明(H)

mainwindow.ui

過濾器 在此輸入

物件 類別

- MainWindow QMainWindow
- centralwidget QWidget
- menubar QMenuBar
- statusbar QStatusBar

過濾器

MainWindow : QMainWindow

屬性	數值
X	0
Y	0
寬度	1024
高度	768
sizePolicy	[Preferred, Preferred, 0, 0]
minimumSize	0x0
maximumSize	16777215x16777215
sizeIncrement	0x0

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.

Type to locate (Ctrl+K)

1 問題 2 Search Results 3 應用程式輸出 4 編譯輸出 5 QML Debugger Console 6 一般訊息 7 Signals Slots E... 8 Test Results

mainwindow.ui @ qtImageProcessing - Qt Creator

檔案(F) 編輯(E) 建置(B) 除錯(D) Analyze 工具(T) 視窗(W) 說明(H)

mainwindow.ui\*

在此輸入

過濾器

物件 類別

MainWindow QMainWindow  
centralwidget QWidget  
labelPic QLabel  
menubar QMenuBar  
statusbar QStatusBar

object Name: labelPic  
frame Shape: Box  
Goometry: 寬:800, 高:600

過濾器

屬性 數值

windowOpacity 1.000000  
toolTip  
toolTipDuration -1  
statusTip  
whatsThis  
accessibleName  
accessibleDescription  
layoutDirection LeftToRight

名稱 已使用 文字 捷徑 可勾選 工具提示

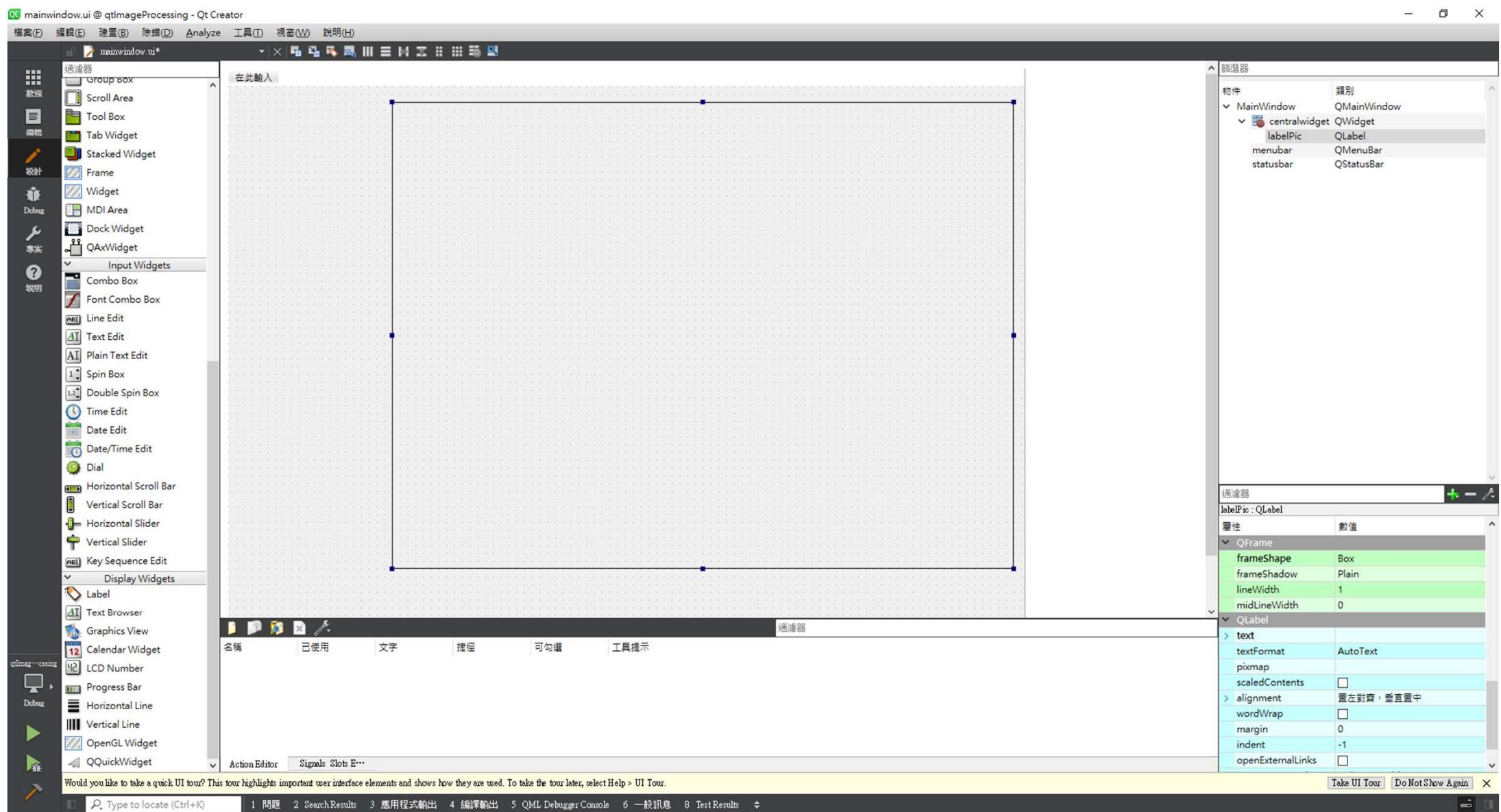
Action Editor Signals Slots E...

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.

Take UI Tour Do Not Show Again

Type to locate (Ctrl+K)

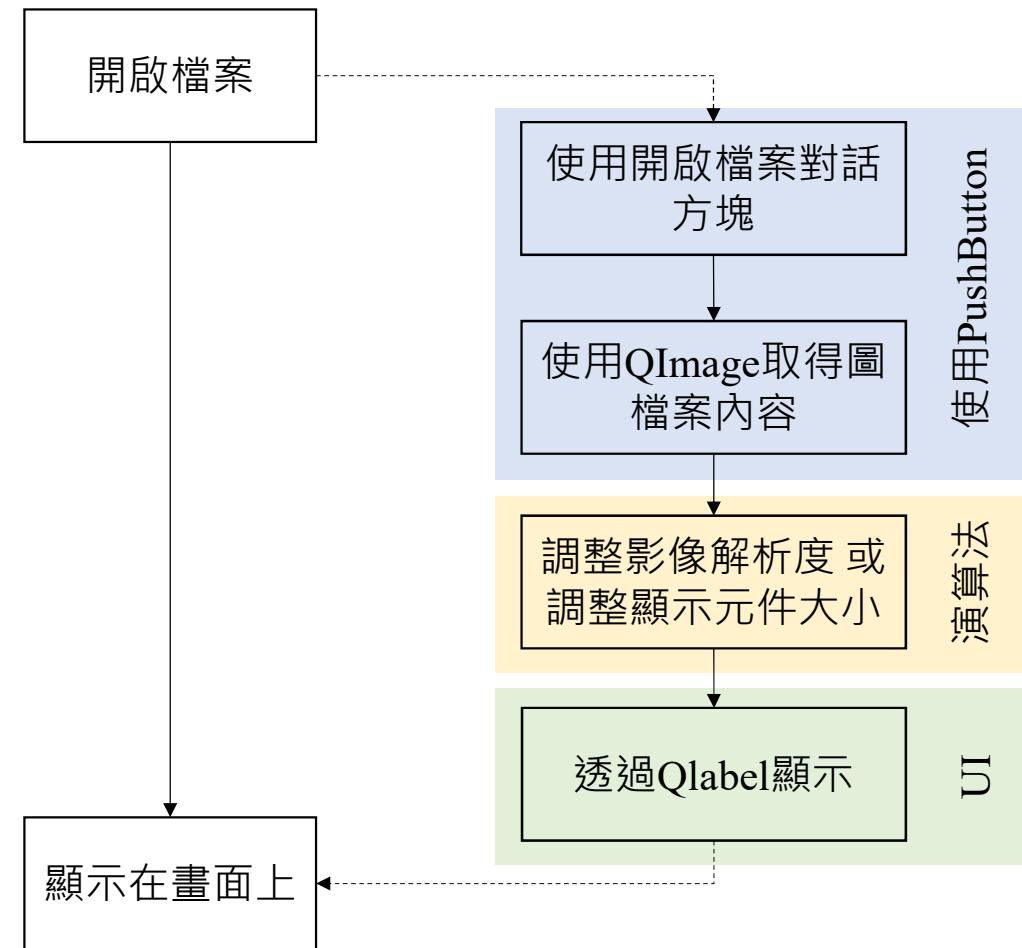
1 問題 2 Search Results 3 應用程式輸出 4 編譯輸出 5 QML Debugger Console 6 一般訊息 7 Test Results





/course/101001001/c/course\_qt\_ImagePcocessing

利用QT元件QImage開啟影像並顯示在畫面上



mainwindow.ui @ qtImageProcessing - Qt Creator

檔案(F) 編輯(E) 建置(B) 除錯(D) Analyze 工具(T) 視窗(W) 說明(H)

mainwindow.ui\*

過濾器

在此輸入

開啟影像

加入一個按鈕

設計

歡迎

編輯

分析

Debug

專案

說明

qtImageProcessing

Debug

Item Widgets (Item-Based)

Layouts

- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Spacers

- Horizontal Spacer
- Vertical Spacer

Buttons

- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Item Views (Model-Based)

- List View
- Tree View
- Table View
- Column View
- Undo View

pushButton : QPushButton

屬性	數值
focusPolicy	StrongFocus
contextMenuPolicy	DefaultContextMenu
acceptDrops	<input type="checkbox"/>
toolTip	
toolTipDuration	-1
statusTip	
whatsthis	
accessibleName	

過濾器

名稱 已使用 文字 捷徑 可勾選 工具提示

Action Editor Signals Slots E...

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.

Take UI Tour Do Not Show Again

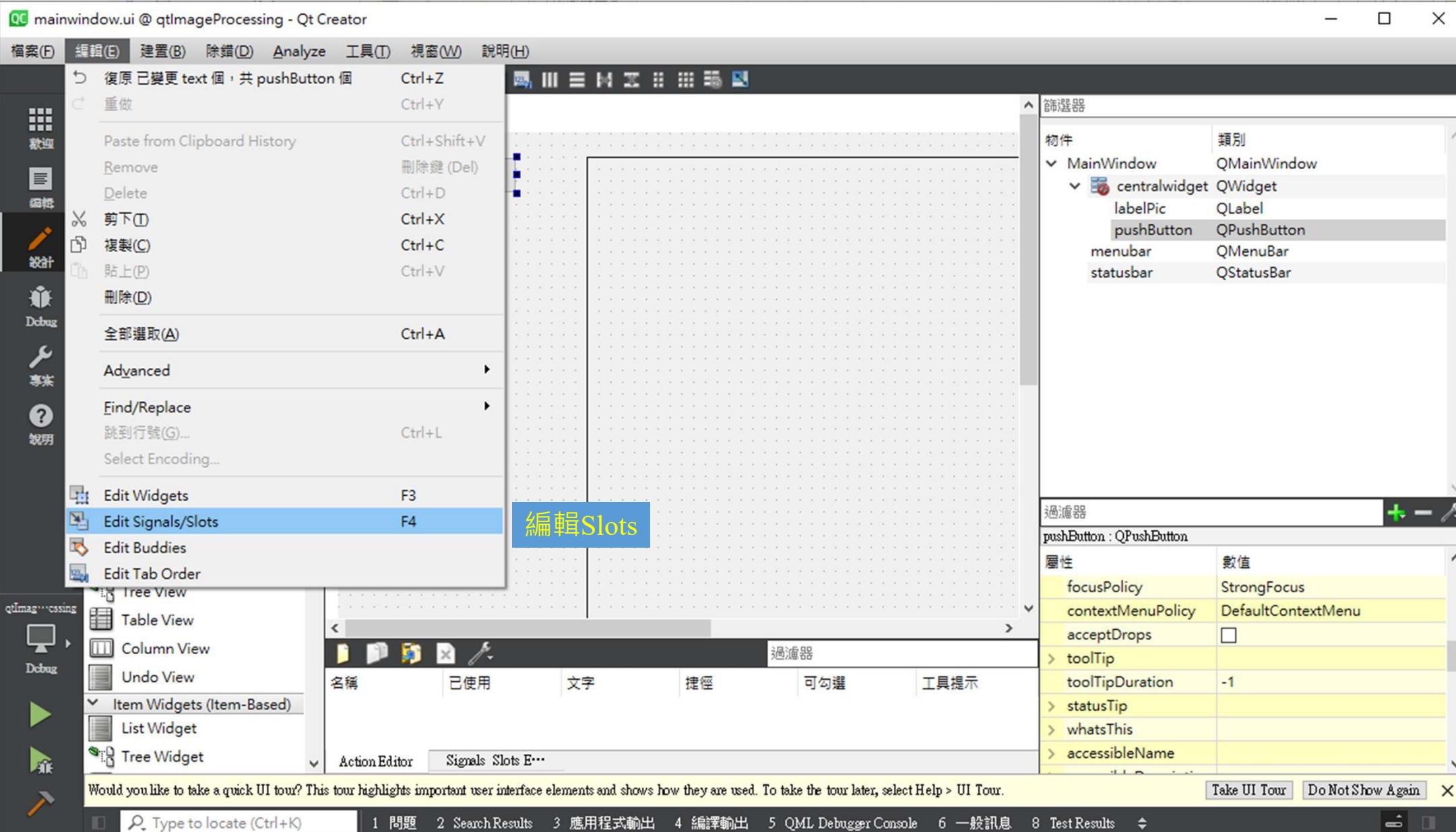
Type to locate (Ctrl+K)

1 問題 2 Search Results 3 應用程式輸出 4 編譯輸出 5 QML Debugger Console 6 一般訊息 7 Test Results

最大化

最小化

關閉



mainwindow.ui @ qtImageProcessing - Qt Creator

檔案(F) 編輯(E) 建置(B) 除錯(D) Analyze 工具(I) 視窗(W) 說明(H)

mainwindow.ui\*

在此輸入

開啟影像

將按鈕拉到空白處

過濾器

物件 類別

- MainWindow QMainWindow
- centralwidget QWidget
  - labelPic QLabel
  - pushButton QPushButton
- menubar QMenuBar
- statusbar QStatusBar

過濾器

MainWindow : QMainWindow

屬性	數值
focusPolicy	NoFocus
contextMenuPolicy	DefaultContextMenu
acceptDrops	<input type="checkbox"/>
windowTitle	MainWindow
可翻譯	<input checked="" type="checkbox"/>
澄清	
註解	
windowIcon	

過濾器

名稱 已使用 文字 捷徑 可勾選 工具提示

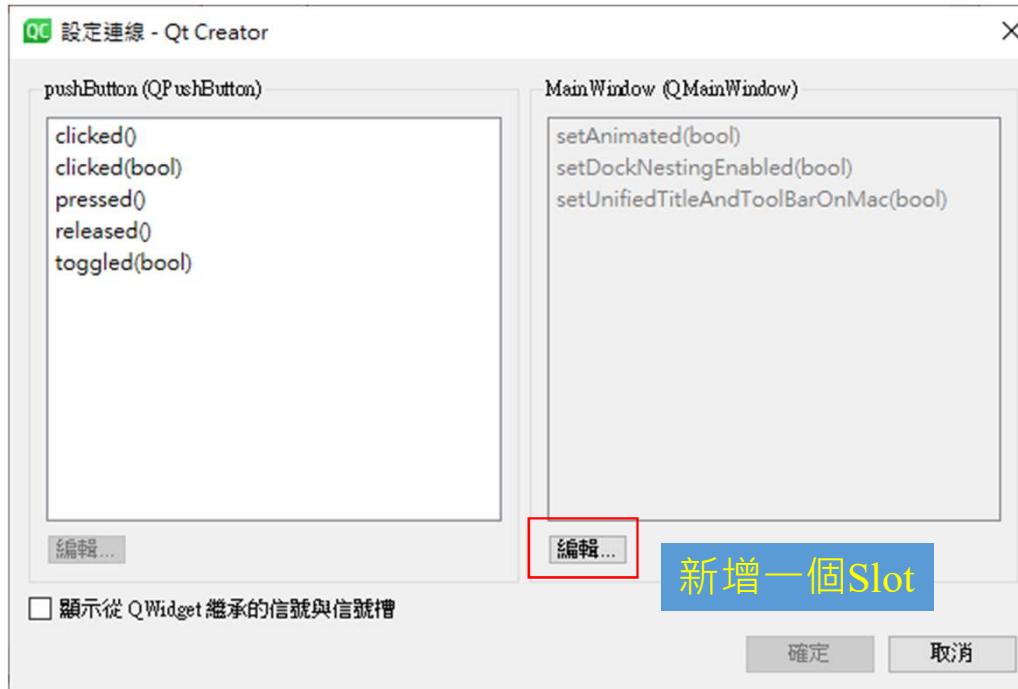
Action Editor Signals Slots E...

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.

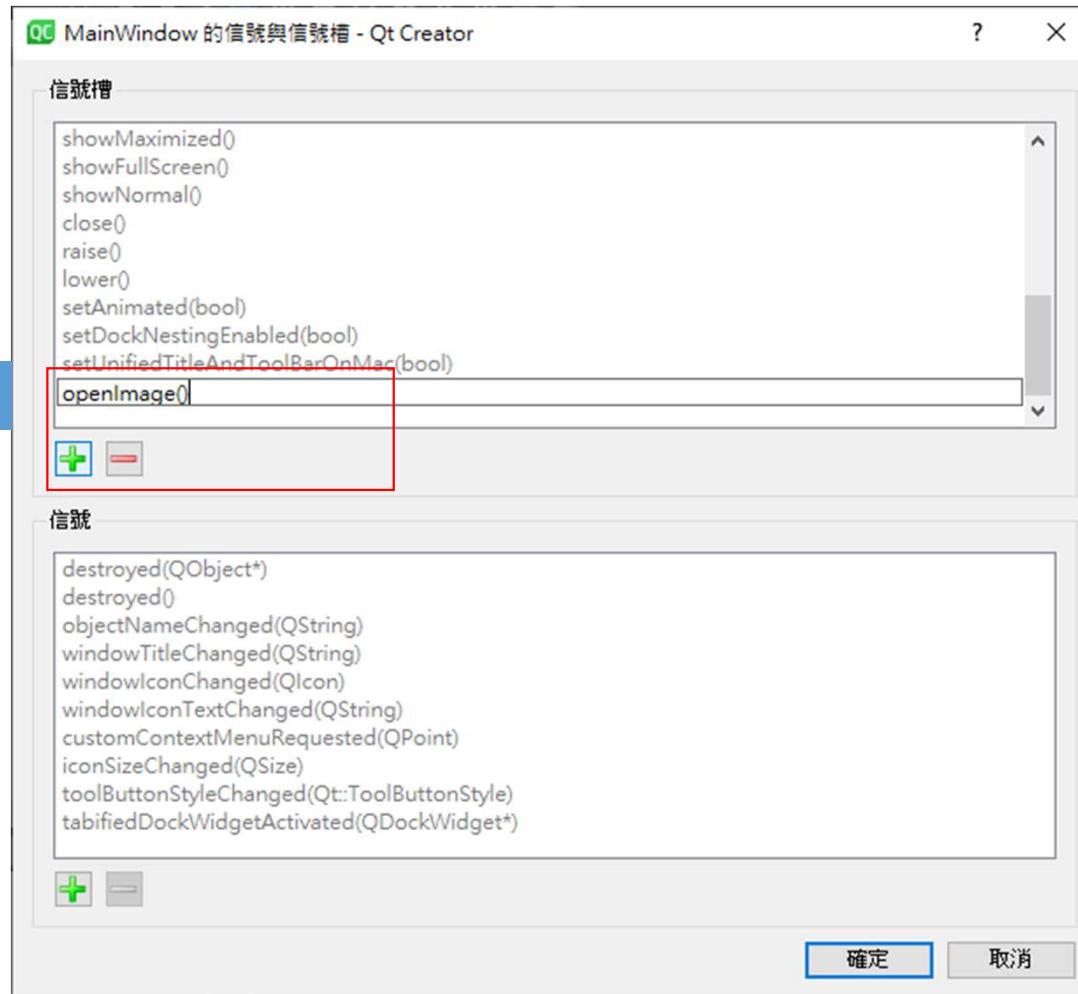
Type to locate (Ctrl+K)

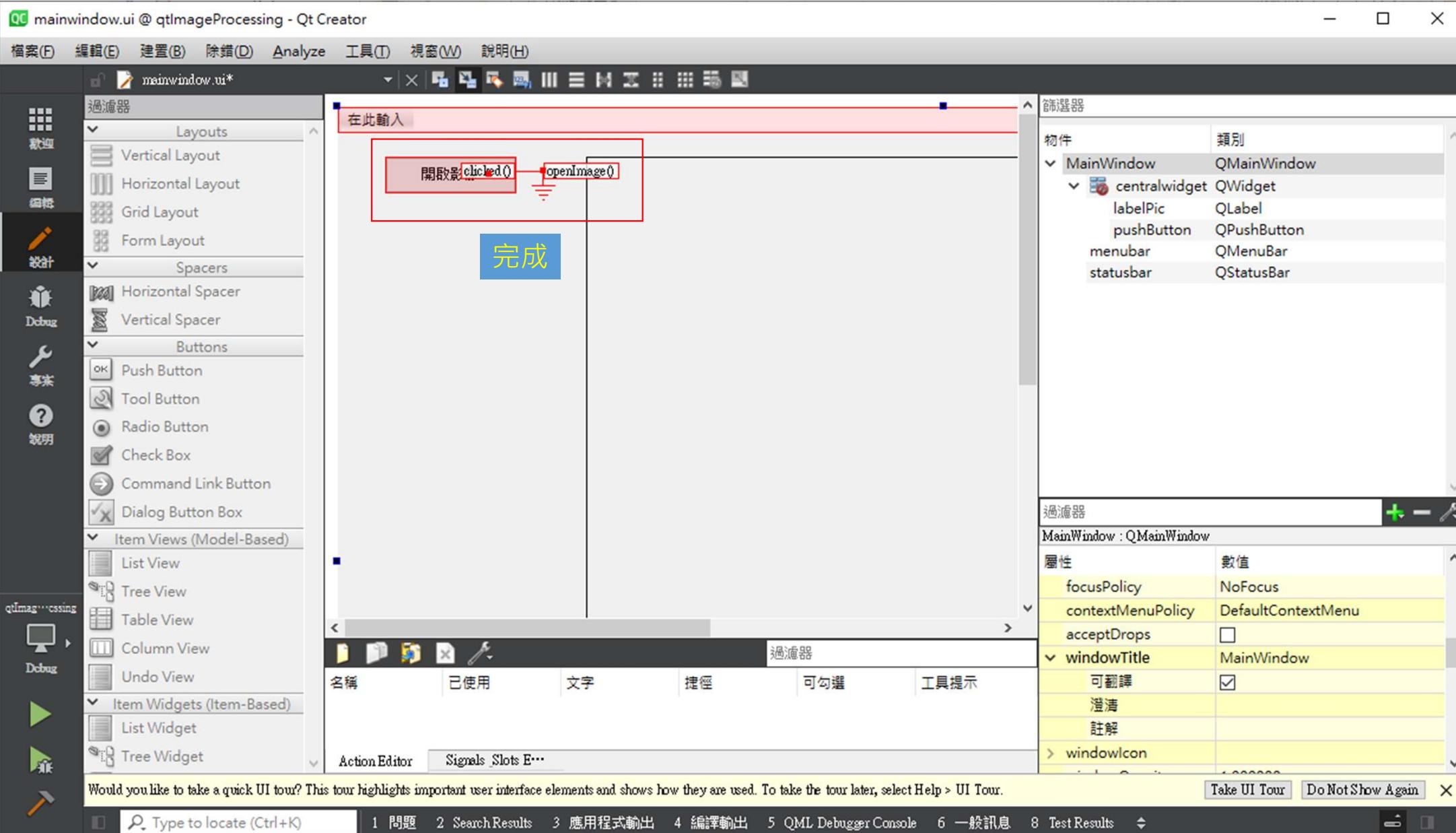
1 問題 2 Search Results 3 應用程式輸出 4 編譯輸出 5 QML Debugger Console 6 一般訊息 7 Test Results 8 Test Results

Take UI Tour Do Not Show Again



新增一個 openImage()





mainwindow.h @ qtImageProcessing - Qt Creator

檔案(F) 編輯(E) 建置(B) 除錯(D) Analyze 工具(T) 視窗(W) 說明(H)

專案 qtImageProcessing Headers mainwindow.h MainWindow Windows (CRLF) Line: 19, Col: 22

歡迎 編輯 設計 誰是 說明

qtImageProcessing Headers mainwindow.h

Sources main.cpp mainwindow.cpp

Forms mainwindow.ui

```
1 #ifndef MAINWINDOW_H
2 #define MAINWINDOW_H
3
4 #include <QMainWindow>
5
6 QT_BEGIN_NAMESPACE
7 namespace Ui { class MainWindow; }
8 QT_END_NAMESPACE
9
10 class MainWindow : public QMainWindow
11 {
12     Q_OBJECT
13
14 public:
15     MainWindow(QWidget *parent = nullptr);
16     ~MainWindow();
17
18 public slots:
19     void openImage();
20
21 private:
22     Ui::MainWindow *ui;
23 };
24
25 #endif // MAINWINDOW_H
```

宣告 public slots

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.

Take UI Tour Do Not Show Again

Type to locate (Ctrl+K)

1 問題 2 Search Results 3 應用程式輸出 4 編譯輸出 5 QML Debugger Console 6 一般訊息 7 Test Results

mainwindow.cpp @ qtImageProcessing - Qt Creator

檔案(F) 編輯(E) 建置(B) 除錯(D) Analyze 工具(T) 視窗(W) 說明(H)

專案 | mainwindow.cpp | <Select Symbol>

Windows (CRLF) | Line: 5, Col: 18

歡迎  
編輯  
設計  
Debug  
專案  
說明

qtImageProcessing  
  qtImageProcessing.pro  
  Headers  
    mainwindow.h  
  Sources  
    main.cpp  
    mainwindow.cpp  
  Forms

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

#include <QFileDialog>
#include <QImage>

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::openImage()
{
    QString ImageFileName;
    QImage *showImg;
    QFileDialog *d = new QFileDialog();
    if(d->exec()==QDialog::Accepted)
    {
        ImageFileName = d->selectedFiles()[0];
        showImg = new QImage();
        showImg->load(ImageFileName);

        ui->labelPic->setPixmap(QPixmap::fromImage(*showImg));
    }
}
```

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.

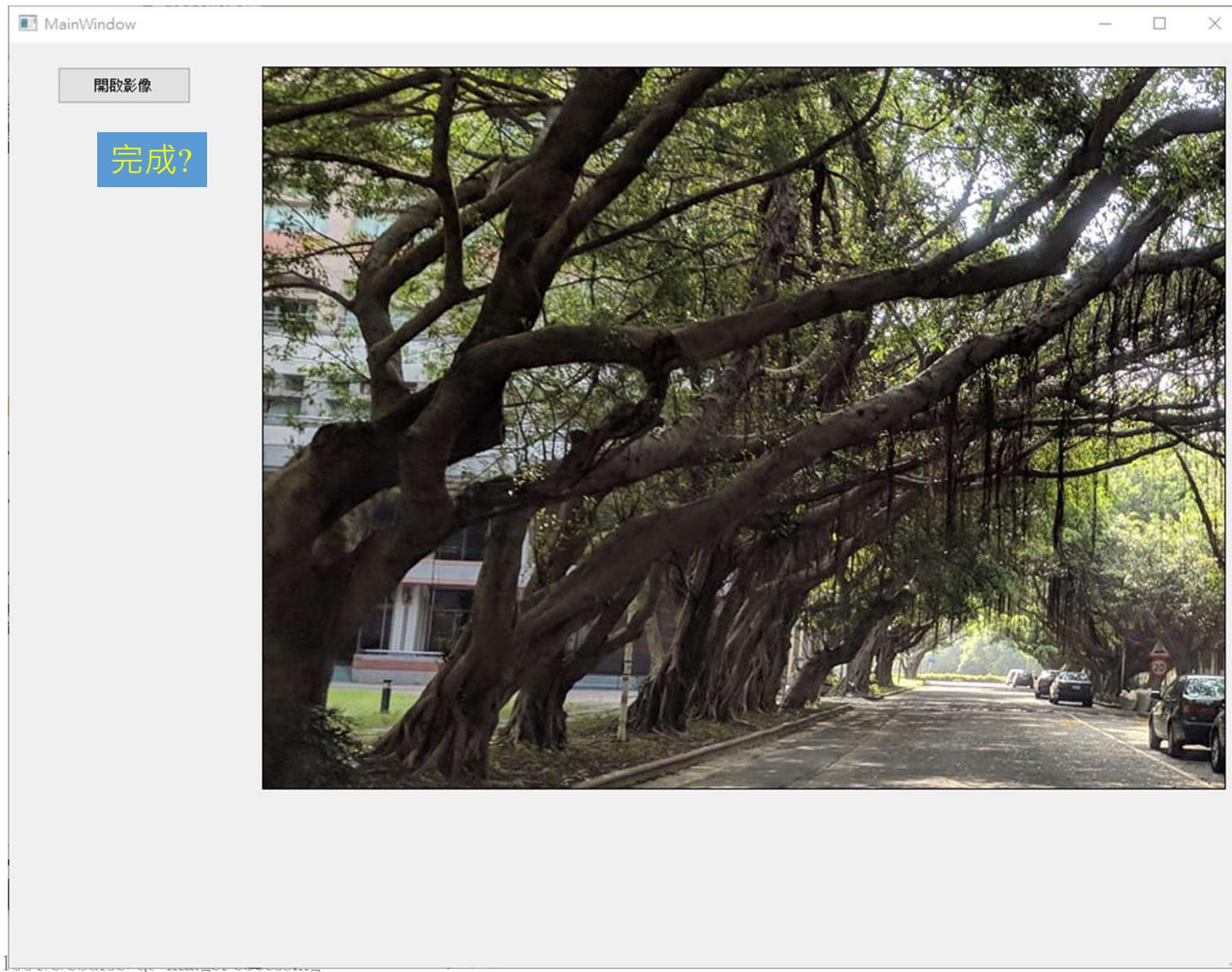
Take UI Tour | Do Not Show Again | X

Type to locate (Ctrl+K)

1 問題 2 Search Results 3 應用程式輸出 4 編譯輸出 5 QML Debugger Console 6 一般訊息 7 Test Results 8

```
3  
4 #include <QFileDialog>  
5 #include <QImage>
```

```
18  
19 void MainWindow::openImage()  
20 {  
21     QString ImageFileName;  
22     QImage *showImg;  
23     QFileDialog *d = new QFileDialog();  
24     if(d->exec()==QDialog::Accepted)  
25     {  
26         ImageFileName = d->selectedFiles()[0];  
27         showImg = new QImage();  
28         showImg->load(ImageFileName);  
29  
30         ui->labelPic->setPixmap(QPixmap::fromImage(*showImg));  
31     }  
32 }
```





/course/101001001/c/course\_qt\_ImagePcocessing

mainwindow.cpp @ qtImageProcessing - Qt Creator

檔案(F) 編輯(E) 建置(B) 除錯(D) Analyze 工具(T) 視窗(W) 說明(H)

專案

歡迎

編輯

設計

Debug

專案

說明

書籤

qtImageProcessing

  qtImageProcessing.pro

  Headers

  mainwindow.h

  Sources

  main.cpp

  mainwindow.cpp

  Forms

  mainwindow.ui

mainwindow.cpp

```
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 #include <QFileDialog>
5 #include <QImage>
6
7 MainWindow::MainWindow(QWidget *parent)
8     : QMainWindow(parent)
9     , ui(new Ui::MainWindow)
10 {
11     ui->setupUi(this);
12 }
13
14 MainWindow::~MainWindow()
15 {
16     delete ui;
17 }
18
19 void MainWindow::openImage()
20 {
21     QString ImageFileName;
22     QImage *showImg;
23     QFileDialog *d = new QFileDialog();
24     if(d->exec() == QDialog::Accepted)
25     {
26         ImageFileName = d->selectedFiles()[0];
27         showImg = new QImage();
28         showImg->load(ImageFileName);
29
30         *showImg = showImg->scaled(ui->labelPic->size());
31
32         ui->labelPic->setPixmap(QPixmap::fromImage(*showImg));
33     }
34 }
35
36 |
```

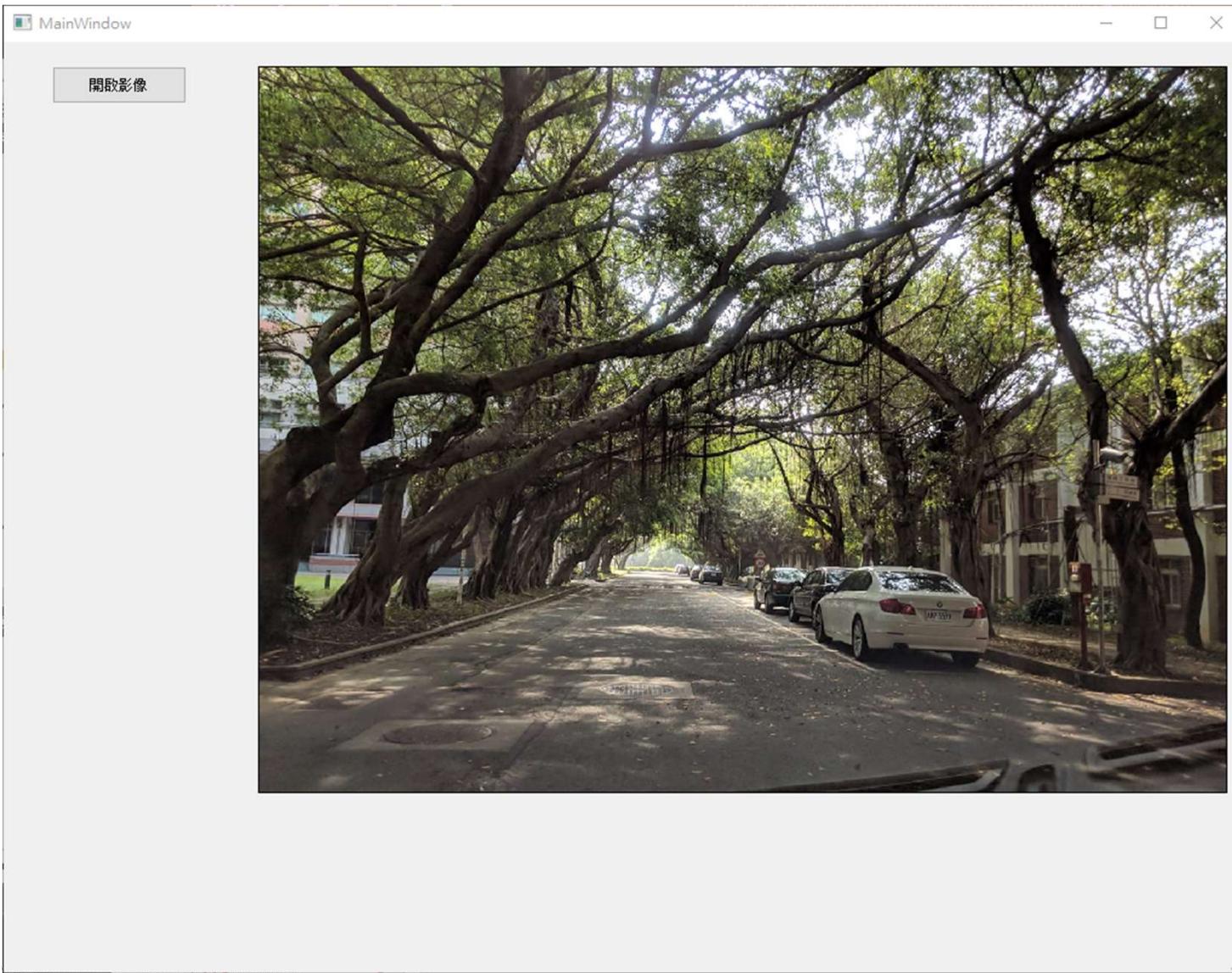
將showImg的尺寸轉換為labelPic的尺寸

Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.

Take UI Tour Do Not Show Again X

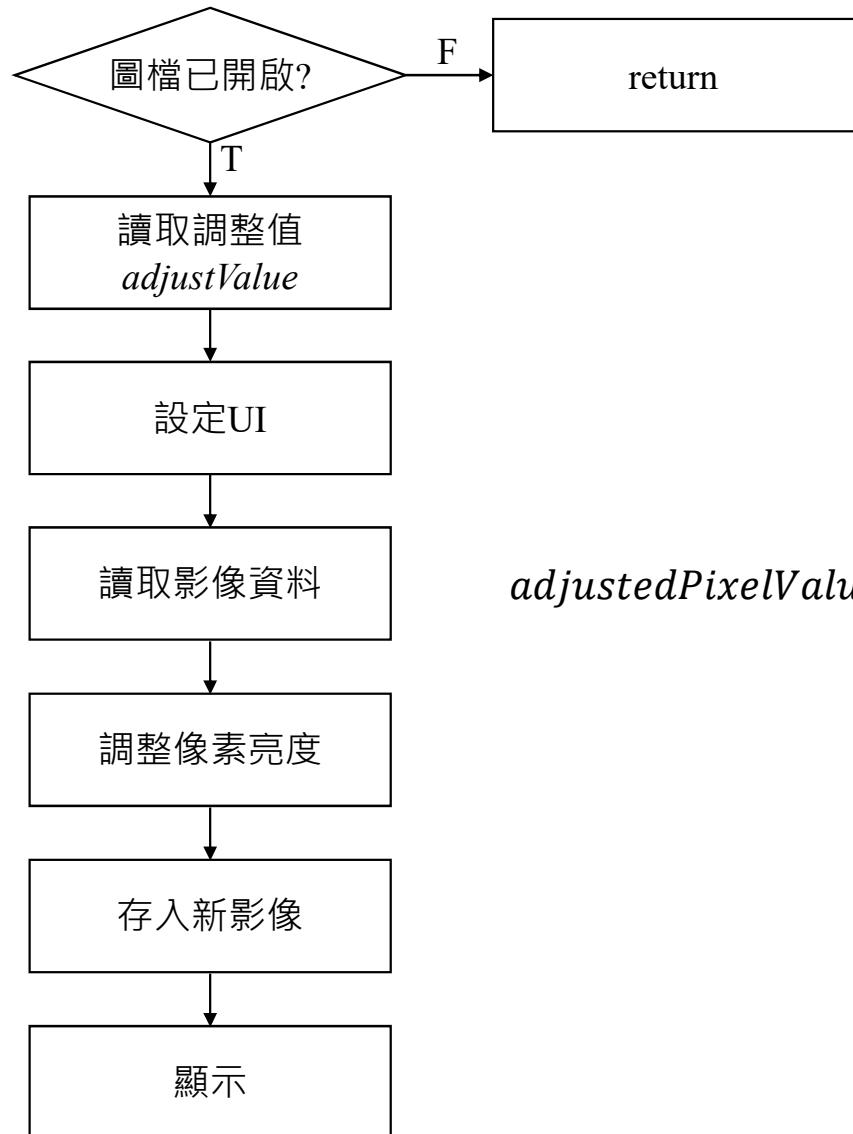
Type to locate (Ctrl+K)

1 問題 2 Search Results 3 應用程式輸出 4 編譯輸出 5 QML Debugger Console 6 一般訊息 8 Test Results



/course/10100100/course\_qt\_imagerprocessing

調整影像亮度(by Pixel)



$$adjustedPixelValue = originalPixelValue \times \left(1 + \frac{adjustValue}{100}\right)$$

## 開始之前—調整變數的視野(Scope)

```
19 void MainWindow::openImage()
20 {
21     QString ImageFileName;
22     QImage *showImg;          改成全域變數
23     QFileDialog *d = new QFileDialog();
24     if(d->exec() == QDialog::Accepted)
25     {
26         ImageFileName = d->selectedFiles()[0];
27         showImg = new QImage();
28         showImg->load(ImageFileName);
29
30         *showImg = showImg->scaled(ui->labelPic->size());
31
32         ui->labelPic->setPixmap(QPixmap::fromImage(*showImg));
33     }
34 }
```

The screenshot shows the Qt Creator IDE interface. On the left is the project tree for 'qtImageProcessing'. It contains a 'Headers' folder with 'mainwindow.h', a 'Sources' folder with 'main.cpp' and 'mainwindow.cpp', and a 'Forms' folder with 'mainwindow.ui'. The 'mainwindow.h' file is currently selected and shown in the main editor area.

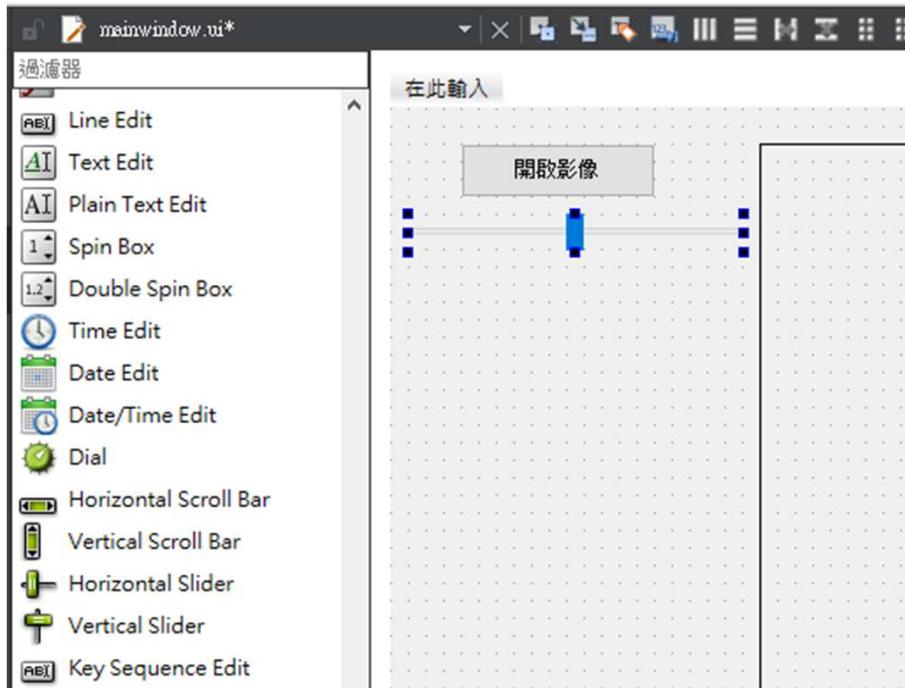
```
1 #ifndef MAINWINDOW_H
2 #define MAINWINDOW_H
3
4 #include <QMainWindow>
5
6 QT_BEGIN_NAMESPACE
7 namespace Ui { class MainWindow; }
8 QT_END_NAMESPACE
9
10 class MainWindow : public QMainWindow
11 {
12     Q_OBJECT
13
14 public:
15     MainWindow(QWidget *parent = nullptr);
16     ~MainWindow();
17
18     QImage *orgImg;
19     QImage *showImg;
20     QImage *adjustedImg;
21     bool imageOpened;
22
23 public slots:
24     void openImage();
25
26 private:
27     Ui::MainWindow *ui;
28 };
29 #endif // MAINWINDOW_H
```

A red rectangular box highlights the declarations of `QImage` pointers and the `bool` variable `imageOpened`. To the right of this box, there is explanatory text in Chinese:

宣告全域變數  
併加上控制變數

The screenshot shows the Qt Creator IDE interface. On the left is the Project Explorer pane, displaying the project 'qtImageProcessing' with files like 'qtImageProcessing.pro', 'mainwindow.h', 'mainwindow.cpp', 'main.cpp', and 'mainwindow.h'. On the right is the Code Editor pane, showing the 'mainwindow.cpp' file with the following code:

```
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 #include <QFileDialog>
5 #include <QImage>
6
7 MainWindow::MainWindow(QWidget *parent)
8     : QMainWindow(parent)
9     , ui(new Ui::MainWindow)
10 {
11     ui->setupUi(this);
12
13     imageOpened = false; 程式開始時的預設值
14 }
15
16 MainWindow::~MainWindow()
17 {
18     delete ui;
19 }
20
21 void MainWindow::openImage()
22 {
23     QString ImageFileName;
24     QFileDialog *d = new QFileDialog();
25     if(d->exec() == QDialog::Accepted)
26     {
27         ImageFileName = d->selectedFiles()[0];
28         showImg = new QImage();
29         showImg->load(ImageFileName);
30
31         *showImg = showImg->scaled(ui->labelPic->size());
32
33         ui->labelPic->setPixmap(QPixmap::fromImage(*showImg));
34
35         imageOpened = true; 圖檔開啟後的控制項
36     }
37 }
38 }
```



minimum: -100  
maximum: 100  
singleStep: 1  
value: 0

屬性	數值
QObject	
objectName	sliderImageBrightness
QWidget	
enabled	<input checked="" type="checkbox"/>
geometry	[(10, 60), 191x22]
X	10
Y	60
寬度	191
高度	22
sizePolicy	[Expanding, Fixed, 0, 0]

屬性	數值
locale	Chinese : Taiwan
inputMethodHints	ImhNone
QAbstractSlider	
minimum	-100
maximum	100
singleStep	1
pageStep	10
value	0
sliderPosition	0
tracking	<input checked="" type="checkbox"/>

mainwindow.ui @ qtImageProcessing - Qt Creator

檔案(F) 編輯(E) 建置(B) 除錯(D) Analyze 工具(I) 視窗(W) 說明(H)

mainwindow.ui\*

過濾器

在此輸入

開啟影像

TextLabel

Input Widgets

- Combo Box
- Font Combo Box
- LineEdit
- TextEdit
- Plain Text Edit
- SpinBox
- Double Spin Box
- Time Edit
- Date Edit
- Date/Time Edit
- Dial
- Horizontal Scroll Bar
- Vertical Scroll Bar
- Horizontal Slider
- Vertical Slider
- Key Sequence Edit

Display Widgets

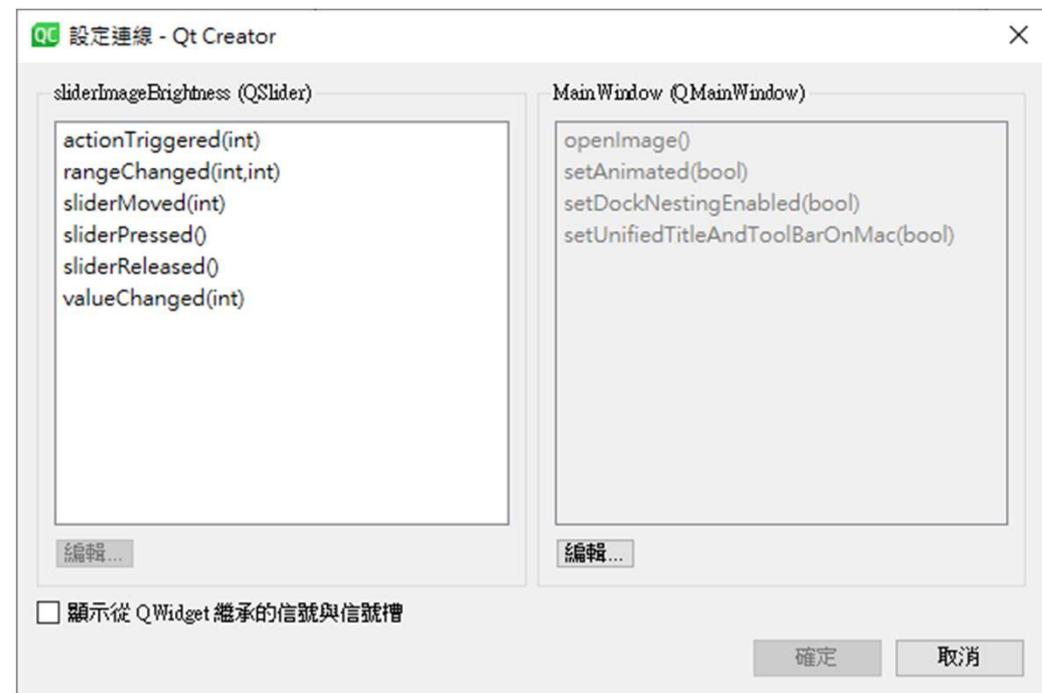
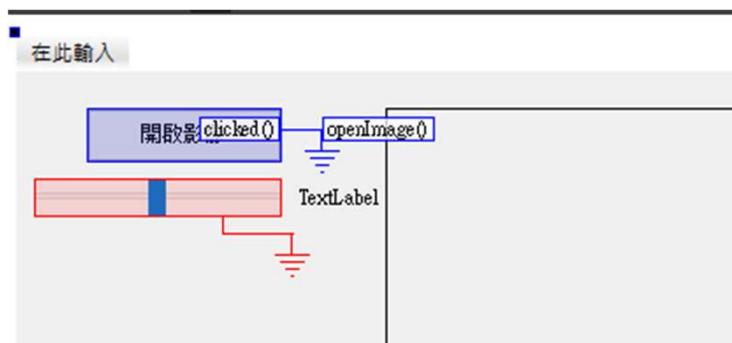
- Label
- Text Browser
- Graphics View
- Calendar Widget
- LCD Number
- Progress Bar
- Horizontal Line

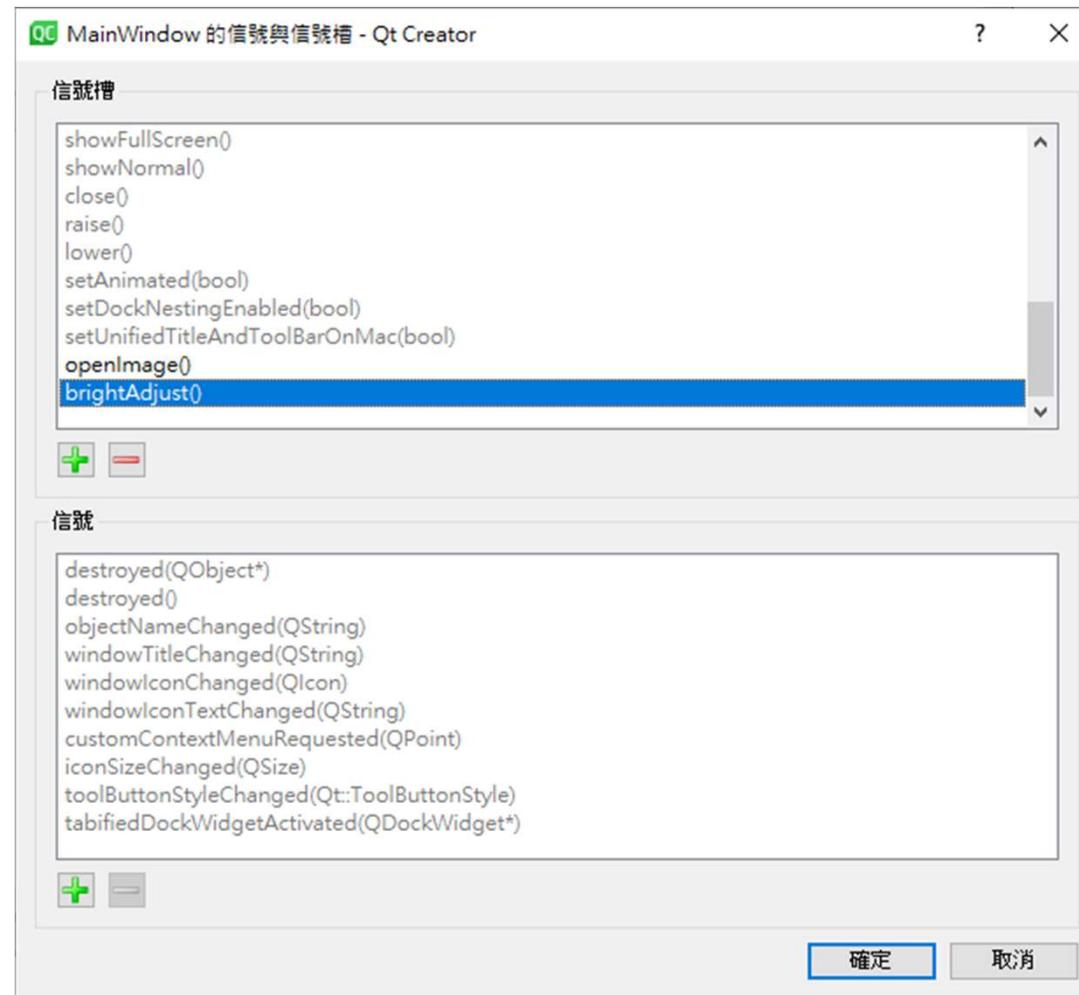
labelBrightAdjustValue : QLabel

屬性	數值
QObject	
objectName	labelBrightAdjustValue
QWidget	
enabled	<input checked="" type="checkbox"/>
geometry	[(160, 61), 47x21]
X	160
Y	61
寬度	47
高度	21
sizePolicy	[Preferred, Preferred, 0, 0]

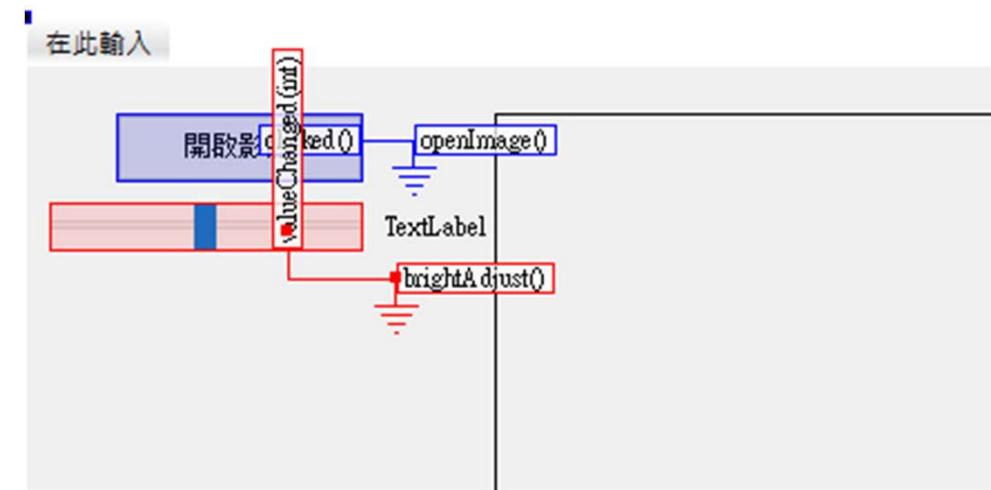
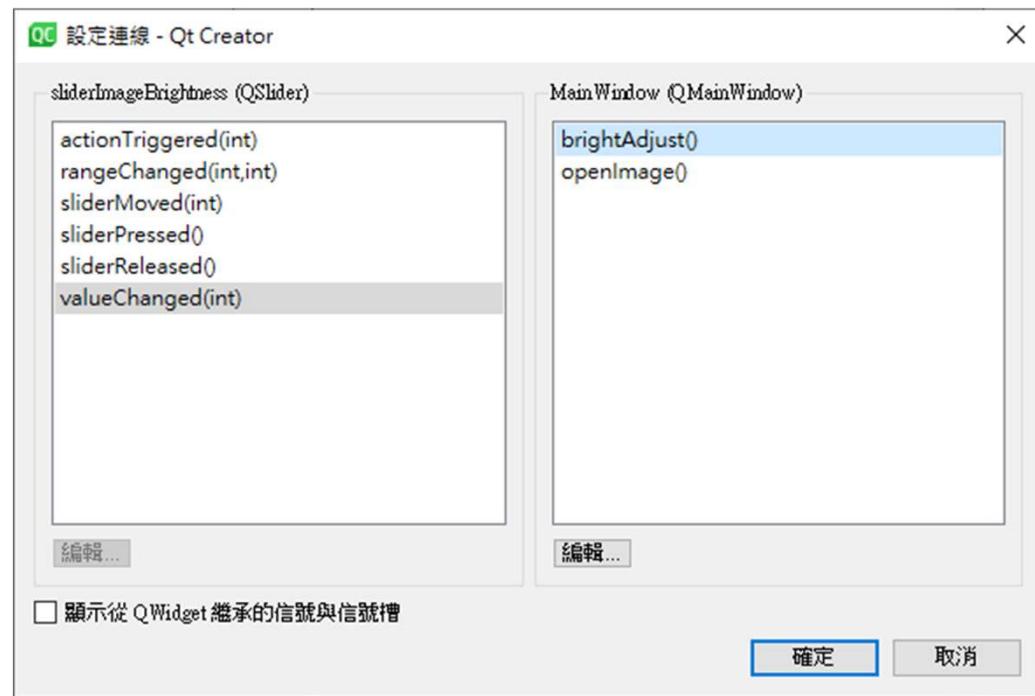
名稱 已使用 文字 捷徑 可勾選 工具提示 過濾器

objectName: labelBrightAdjustValue





/course/101001001/c/course\_qt\_ImagePcocessing



The screenshot shows the Qt Creator IDE interface. On the left is the project tree for 'qtImageProcessing'. It contains a 'Headers' folder with 'mainwindow.h', a 'Sources' folder with 'main.cpp' and 'mainwindow.cpp', and a 'Forms' folder with 'mainwindow.ui'. The 'mainwindow.h' file is selected and shown in the main editor area.

```
1 #ifndef MAINWINDOW_H
2 #define MAINWINDOW_H
3
4 #include <QMainWindow>
5
6 QT_BEGIN_NAMESPACE
7 namespace Ui { class MainWindow; }
8 QT_END_NAMESPACE
9
10 class MainWindow : public QMainWindow
11 {
12     Q_OBJECT
13
14 public:
15     MainWindow(QWidget *parent = nullptr);
16     ~MainWindow();
17
18     QImage *orgImg;
19     QImage *showImg;
20     QImage *adjustedImg;
21     bool imageOpened;
22
23 public slots:
24     void openImage();
25     void brightAdjust(); 加上 slots
26
27 private:
28     Ui::MainWindow *ui;
29 };
30 #endif // MAINWINDOW_H
```

/course/101001001/c/course\_qt\_ImagePcocessing

The screenshot shows the Qt Creator IDE interface. On the left is the project tree for 'qtImageProcessing' containing files like 'mainwindow.h', 'main.cpp', and 'mainwindow.ui'. The main window displays the source code for 'mainwindow.cpp'.

```
41 void MainWindow::brightAdjust()
42 {
43     unsigned char *rgba;
44     double adjustValue = ui->sliderImageBrightness->value();
45
46     if(!imageOpened)
47         return;
48
49     adjustValue /= 100.0;
50     ui->labelBrightAdjustValue->setText(QString::number(adjustValue));
51
52     rgba = orgImg->bits();
53     int w = orgImg->width();
54     int h = orgImg->height();
55
56     for(int y=0;y<h;y++)
57         for(int x=0;x<w*4;x+=4)
58         {
59             int r,g,b;
60             r = rgba[y*w*4+x+2]*(1+adjustValue);
61             (r>255)?r=255:r=r;
62             (r<0)?r=0:r=r;
63             g = rgba[y*w*4+x+1]*(1+adjustValue);
64             (g>255)?g=255:g=g;
65             (g<0)?g=0:g=g;
66             b = rgba[y*w*4+x]*(1+adjustValue);
67             (b>255)?b=255:b=b;
68             (b<0)?b=0:b=b;
69
70             adjustedImg->setPixel(x/4,y,qRgb(r,g,b));
71         }
72
73     *showImg = adjustedImg->scaled(ui->labelPic->width(),ui->labelPic->height());
74     ui->labelPic->setPixmap(QPixmap::fromImage(*showImg));
75
76 }
77 }
```

```
44     unsigned char *rgba;
45     double adjustValue = ui->sliderImageBrightness->value();
```

宣告 `unsigned char *rgba` 為像素資料的指標

宣告 `double adjustValue` 為亮度調整值 → 數值會介於 -100 ~ +100 之間(原始值為整數)

```
46
47     if(!imageOpened)
48         return;
```

檢查影像是否開啟，沒有開啟的話就return

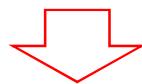
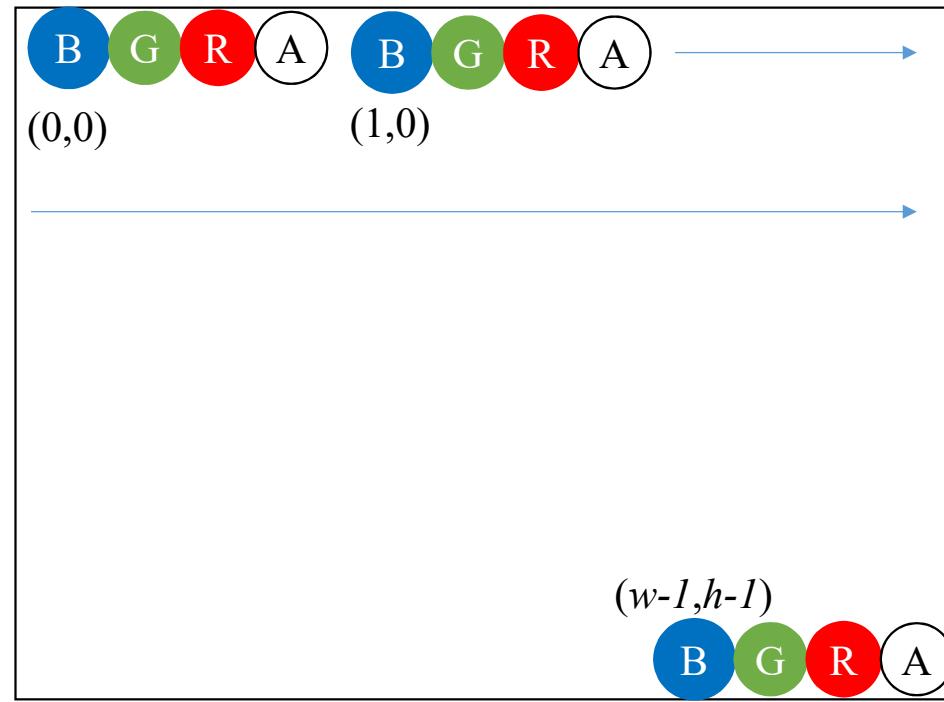
```
49
50     adjustValue /=100.0;
51     ui->labelBrightAdjustValue->setText(QString::number(adjustValue));
```

正規化調整值，使其介於-1~+1之間 (浮點數)

設定UI

```
52
53     rgba = orgImg->bits();
54     int w = orgImg->width();
55     int h = orgImg->height();
```

將 `rgba` 指向原始影像的資料位址，並將寬與高宣告成為變數



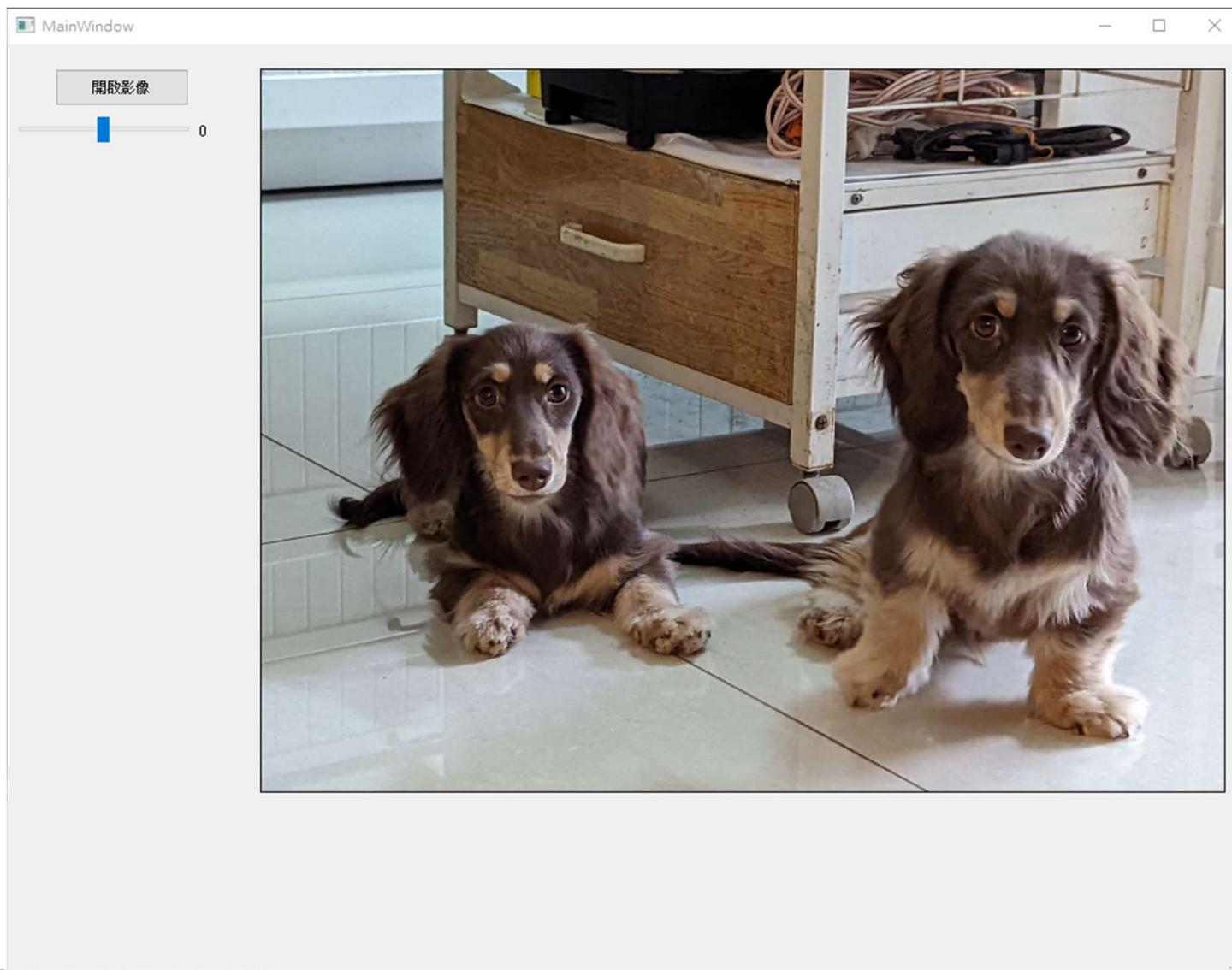
```
57     for(int y=0;y<h;y++)
58         for(int x=0;x<w*4;x+=4)
59     {
60         int r,g,b;
61         r = rgba[y*w*4+x+2]*(1+adjustValue);
62         (r>255)?r=255:r=r;
63         (r<0)?r=0:r=r;
64         g = rgba[y*w*4+x+1]*(1+adjustValue);
65         (g>255)?g=255:g=g;
66         (g<0)?g=0:g=g;
67         b = rgba[y*w*4+x]*(1+adjustValue);
68         (b>255)?b=255:b=b;
69         (b<0)?b=0:b=b;
70
71         adjustedImg->setPixel(x/4,y,qRgb(r,g,b));
72     }
```

找到像素值並乘以調整值  
同時限制像素介於0-255之間

將處理後的像素值透過  
qRgb()函數存到新的影  
像當中

```
13  
74     *showImg = adjustedImg->scaled(ui->labelPic->width(),ui->labelPic->height());  
75     ui->labelPic->setPixmap(QPixmap::fromImage(*showImg));  
76 
```

處理後的影像調整解析度後顯示到畫面上



/course/101001001/course\_qt\_imageresprocessing

