

## 25 绘制事件

### 25.1 处理绘制事件

当有下列情况之一发生时，将触发窗口的绘制事件，届时窗口的paintEvent虚函数会被调用：

- 窗口创建后首次显示
- 窗口由隐藏变为可见
- 窗口由最小化变为正常或最大化
- 窗口因大小改变需要呈现更多内容
- 人为调用窗口的update或repaint方法

程序设计者可以通过覆盖基类的paintEvent虚函数，自己决定在窗口中绘制的内容。

### 25.2 画家类

Qt提供了名为QPainter的画家类，用于实现二维图形图像的绘制和渲染。在paintEvent虚函数的覆盖版本中，可以借助QPainter类的对象，完成在窗口中绘制图形或渲染图像的操作。

### 25.3 案例

#### 25.3.1 创建项目

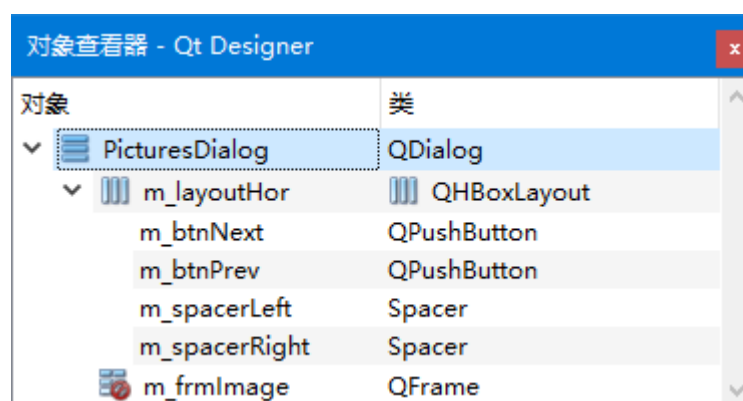
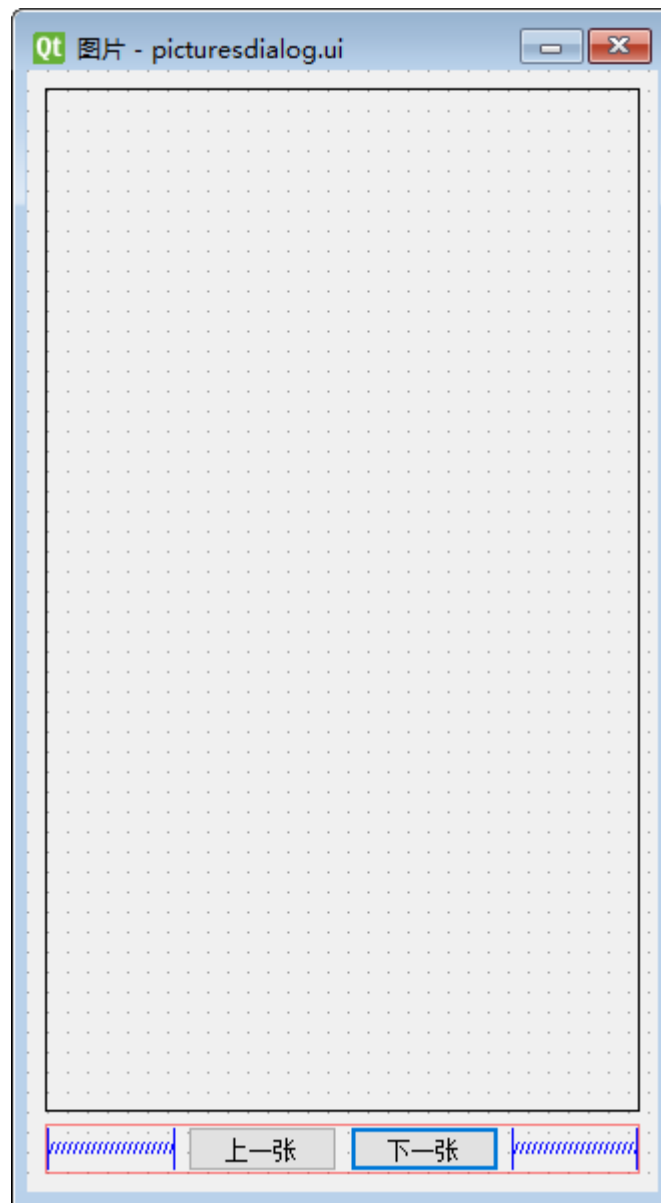
通过QtCreator，在C:\Users\Minwei\Projects\Qt路径下，创建名为Pictures的项目。

#### 25.3.2 添加资源

C:\Users\Minwei\Projects\Qt\Pictures\Pictures.qrc：

```
1 <RCC>
2     <qresource prefix="/">
3         <file>images/0.jpg</file>
4         <file>images/1.jpg</file>
5         <file>images/2.jpg</file>
6         <file>images/3.jpg</file>
7         <file>images/4.jpg</file>
8         <file>images/5.jpg</file>
9         <file>images/6.jpg</file>
10        <file>images/7.jpg</file>
11    </qresource>
12 </RCC>
```

### 25.3.3 设计界面



C:\Users\Minwei\Projects\Qt\Pictures\picturesdialog.ui:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ui version="4.0">
3   <class>PicturesDialog</class>
4   <widget class="QDialog" name="PicturesDialog">
5     <property name="geometry">
6       <rect>
7         <x>0</x>
8         <y>0</y>
```

```
9     <width>315</width>
10    <height>560</height>
11    </rect>
12  </property>
13  <property name="windowTitle">
14    <string>图片</string>
15  </property>
16  <layout class="QVBoxLayout" name="m_layoutVer">
17    <item>
18      <widget class="QFrame" name="m_frmImage">
19        <property name="sizePolicy">
20          <sizepolicy hsize="Preferred" vsizetype="Expanding">
21            <horstretch>0</horstretch>
22            <verstretch>0</verstretch>
23          </sizepolicy>
24        </property>
25        <property name="frameShape">
26          <enum>QFrame::Box</enum>
27        </property>
28        <property name="frameShadow">
29          <enum>QFrame::Plain</enum>
30        </property>
31        <property name="linewidth">
32          <number>1</number>
33        </property>
34      </widget>
35    </item>
36    <item>
37      <layout class="QHBoxLayout" name="m_layoutHor">
38        <item>
39          <spacer name="m_spacerLeft">
40            <property name="orientation">
41              <enum>Qt::Horizontal</enum>
42            </property>
43            <property name="sizeHint" stdset="0">
44              <size>
45                <width>40</width>
46                <height>20</height>
47              </size>
48            </property>
49          </spacer>
50        </item>
51        <item>
52          <widget class="QPushButton" name="m_btnPrev">
53            <property name="text">
54              <string>上一张</string>
55            </property>
56          </widget>
57        </item>
58        <item>
59          <widget class="QPushButton" name="m_btnNext">
60            <property name="text">
61              <string>下一张</string>
62            </property>
63            <property name="default">
64              <bool>true</bool>
```

```

65     </property>
66 </widget>
67 </item>
68 <item>
69     <spacer name="m_spacerRight">
70     <property name="orientation">
71         <enum>Qt::Horizontal</enum>
72     </property>
73     <property name="sizeHint" stdset="0">
74         <size>
75             <width>40</width>
76             <height>20</height>
77         </size>
78     </property>
79 </spacer>
80 </item>
81 </layout>
82 </item>
83 </layout>
84 </widget>
85 <resources/>
86 <connections/>
87 </ui>

```

### 25.3.4 实现功能

C:\Users\Minwei\Projects\Qt\Pictures\picturesdialog.h:

```

1  #ifndef PICTURESIALOG_H
2  #define PICTURESIALOG_H
3
4  #include <QDialog>
5
6  QT_BEGIN_NAMESPACE
7  namespace Ui { class PicturesDialog; }
8  QT_END_NAMESPACE
9
10 class PicturesDialog : public QDialog
11 {
12     Q_OBJECT
13
14 public:
15     PicturesDialog(QWidget *parent = nullptr);
16     ~PicturesDialog();
17
18 protected:
19     void paintEvent(QPaintEvent*);
20
21 private slots:
22     void on_m_btnPrev_clicked();
23     void on_m_btnNext_clicked();
24
25 private:
26     void enableButtons(void);
27

```

```

28 private:
29     Ui::PicturesDialog *ui;
30     int m_imageIndex;
31 };
32
33 #endif // PICTURES_DIALOG_H

```

C:\Users\Minwei\Projects\Qt\Pictures\picturesdialog.cpp:

```

1  #include <QPainter>
2
3  #include "picturesdialog.h"
4  #include "ui_picturesdialog.h"
5
6  PicturesDialog::PicturesDialog(QWidget *parent)
7      : QDialog(parent)
8      , ui(new Ui::PicturesDialog)
9      , m_imageIndex(0)
10 {
11     ui->setupUi(this);
12
13     enableButtons();
14 }
15
16 PicturesDialog::~PicturesDialog()
17 {
18     delete ui;
19 }
20
21 void PicturesDialog::paintEvent(QPaintEvent*)
22 {
23     QPainter painter(this);
24
25     QRect frameRect = ui->m_frmImage->frameRect();
26     frameRect.translate(ui->m_frmImage->pos());
27
28     QImage image(":/images/" + QString::number(m_imageIndex) + ".jpg");
29     painter.drawImage(frameRect, image);
30 }
31
32 void PicturesDialog::on_m_btnPrev_clicked()
33 {
34     --m_imageIndex;
35
36     enableButtons();
37     update();
38 }
39
40 void PicturesDialog::on_m_btnNext_clicked()
41 {
42     ++m_imageIndex;
43
44     enableButtons();
45     update();
46 }

```

```
47
48 void PicturesDialog::enableButtons(void)
49 {
50     ui->m_btnPrev->setEnabled(m_imageIndex != 0);
51     ui->m_btnNext->setEnabled(m_imageIndex != 6);
52 }
```

### 25.3.5 测试验证

运行效果如图所示：

