第12篇 (2D绘图) 绘制图片

代码地址：https://github.com/Lornatang/QtStartQuicklyTutorial/tree/main/Painter02

目录

一、简单绘制图片

1．这次我们重新创建一个Qt Gui应用，项目名称为`painter_2`，在类信息页面，将基类更改为`QDialog`，类名使用默认的`Dialog`即可。

2．然后在源码目录中复制一张图片，比如这里是一张背景透明的`logo.png`图片，如下图所示。

3．在`dialog.h`文件中添加重绘事件处理函数的声明：

```
protected:
    void paintEvent(QPaintEvent *);
```

4．到`dialog.cpp`文件中先添加头文件包含`#include <QPainter>`，然后添加函数的定义：

```
void Dialog::paintEvent(QPaintEvent *)
{
    QPainter painter(this);
    QPixmap pix;
    pix.load("../Painter02/beacon_logo.png");
    painter.drawPixmap(0, 0, 230, 24, pix);
}
```

这里使用了相对路径，因为Qt Creator默认是使用影子构建，即编译生成的文件在`painter_2-build-desktop-Debug`这样的目录里面，而这个目录就是当前目录，所以源码目录就是其上级目录了。大家可以根据自己的实际情况来更改路径，也可以使用绝对路径，不过最好使用资源文件来存放图片。`drawPixmap()`函数在给定的矩形中来绘制图片，这里矩形的左上角顶点为`(0, 0)`点，宽129，高66，如果这个跟图片的大小不相同，默认会拉伸图片。运行效果如下图所示。

（注意：下面的操作涉及到了坐标系统，这里不再详细讲解，大家先进行操作查看效果，具体的坐标内容将在下一节讲解。）

二、平移图片

`QPainter`类中的`translate()`函数实现坐标原点的改变，改变原点后，此点将会成为新的原点`(0, 0)`。下面来看一个例子。

在`paintEvent()`函数里面继续添加如下代码：

```
painter.translate(100, 100); //将（100，100）设为坐标原点
painter.drawPixmap(0, 0, 230, 24, pix);
```

运行程序，效果如下图所示。

这里将`(100, 100)`设置为了新的坐标原点，所以下面在`(0, 0)`点贴图，就相当于在以前的`(100, 100)`点贴图。

三、缩放图片

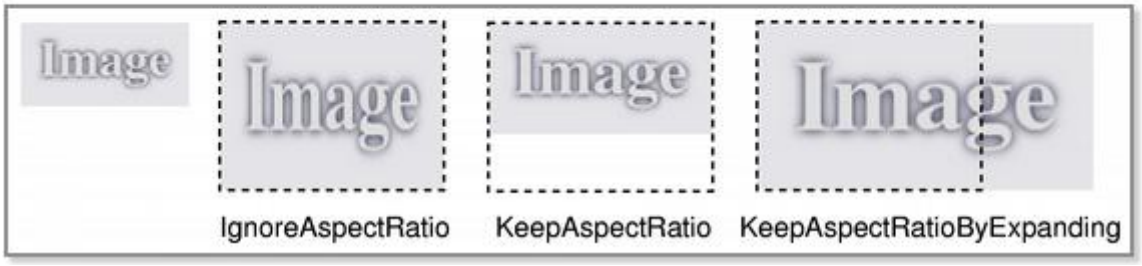
我们可以使用`QPixmap`类中的`scaled()`函数来实现图片的放大和缩小。

在`paintEvent()`函数中继续添加代码：

```
qreal width = pix.width(); //获得以前图片的宽和高
qreal height = pix.height();
//将图片的宽和高都扩大两倍，并且在给定的矩形内保持宽高的比值不变
pix = pix.scaled(width*2, height*2, Qt::KeepAspectRatio);
painter.drawPixmap(70, 70,pix);
```

其中参数`Qt::KeepAspectRatio`，是图片缩放的方式。我们可以查看其帮助。将鼠标指针放到该代码上，当出现F1提示时，按下F1键，这时就可以查看其帮助了。当然我们也可以直接在帮助里查找该关键字。如下图所示。

Scales the pixmap to the given *size*, using the aspect ratio and transformation modes spe
transformMode.



这里有三个值，只看其图片就可大致明白，`Qt::IgnoreAspectRatio`是不保持图片的宽高比，`Qt::KeepAspectRatio`是在给定的矩形中保持宽高比，最后一个也是保持宽高比，但可能超出给定的矩形。这里给定的矩形是由我们显示图片时给定的参数决定的，例如`painter.drawPixmap(0,0,100,100,pix);`就是在以`(0, 0)`点为起始点的宽和高都是100的矩形中。

运行程序效果如下图所示。

四、旋转图片

旋转使用的是`QPainter`类的`rotate()`函数，它默认是以原点为中心进行旋转的。我们要改变旋转的中心，可以使用前面讲到的`translate()`函数完成。

在`paintEvent()`函数中继续添加如下代码：

```
painter.translate(64, 33); //让图片的中心作为旋转的中心
painter.rotate(90); //顺时针旋转90度
painter.translate(-64,-33); //使原点复原
painter.drawPixmap(100, 100, 129, 66, pix);
```

这里必须先改变旋转中心，然后再旋转，然后再将原点复原，才能达到想要的效果。运行程序，效果如图所示。

五、扭曲图片

实现图片的扭曲，是使用的`QPainter`类的`shear(qreal sh, qreal sv)`函数完成的。它有两个参数，前面的参数实现横行变形，后面的参数实现纵向变形。当它们的值为0时，表示不扭曲。

在`paintEvent()`中继续添加如下代码：

```
painter.shear(0.5, 0); //横向扭曲
painter.drawPixmap(100, 0, 129, 66, pix);
```

运行效果如下图所示。


Dialog
Qter
开源社区
Qter
开源社区
Qter
开源社区
Qter
开源社区
Qter
开源社区