



# 第11篇 (2D绘图) 绘制路径

代码地址：<https://github.com/Lornatang/QtStartQuicklyTutorial/tree/main/Painter01>

## 目录

目录

一、简单的使用路径

二、复制图形

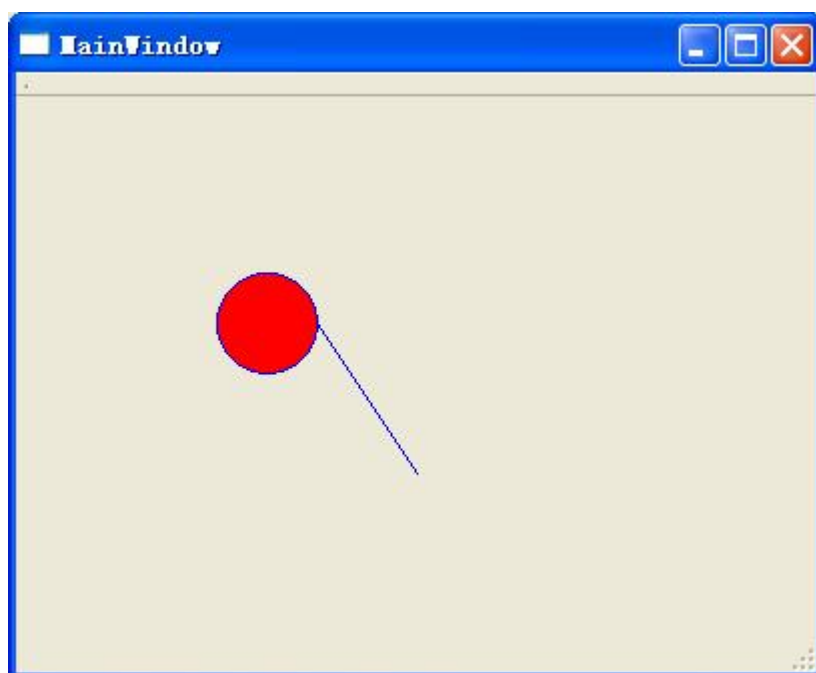
三、绘制图形时的当前位置

## 一、简单的使用路径

依然在前面的项目中进行讲解。更改 `paintEvent()` 函数如下：

```
void MainWindow::paintEvent(QPaintEvent *)
{
    QPainterPath path;
    path.addEllipse(100, 100, 50, 50);
    path.lineTo(200, 200);
    QPainter painter(this);
    painter.setPen(Qt::blue);
    painter.setBrush(Qt::red);
    painter.drawPath(path);
}
```

当创建一个 `QPainterPath` 对象后，可以使用 `lineTo()`、`arcTo()`、`cubicTo()` 和 `quadTo()` 等函数将直线或者曲线添加到路径中。运行程序，效果如下图所示。

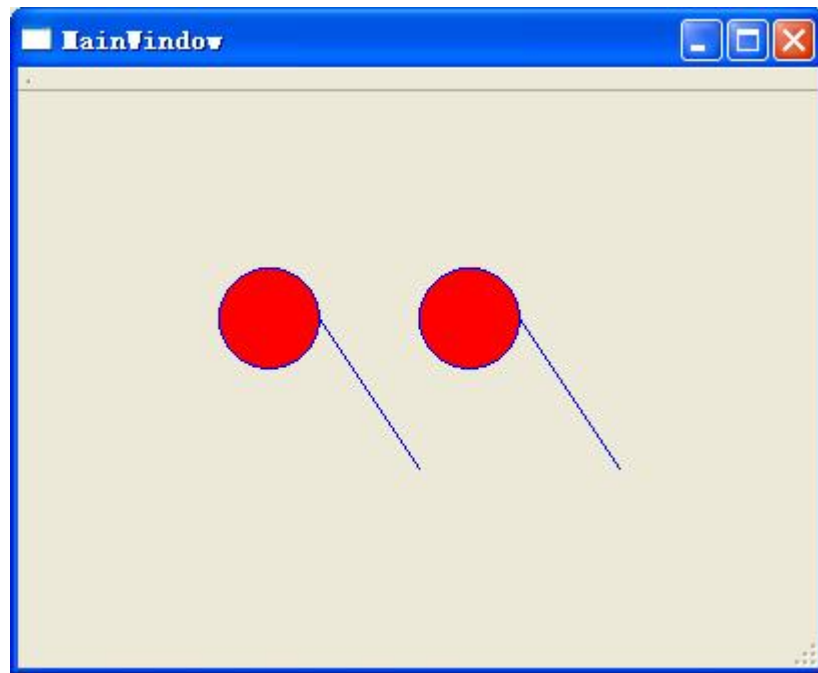


## 二、复制图形

如果只是简单的将几个图形拼接在一起，其实完全没有必要用路径，之所以要引入路径，就是因为它的一个非常有用的功能：复制图形路径。我们在 `paintEvent()` 函数中继续添加下面几行代码：

```
QPainterPath path2;  
path2.addPath(path);  
path2.translate(100,0);  
painter.drawPath(path2);
```

现在运行程序，效果如下图所示。



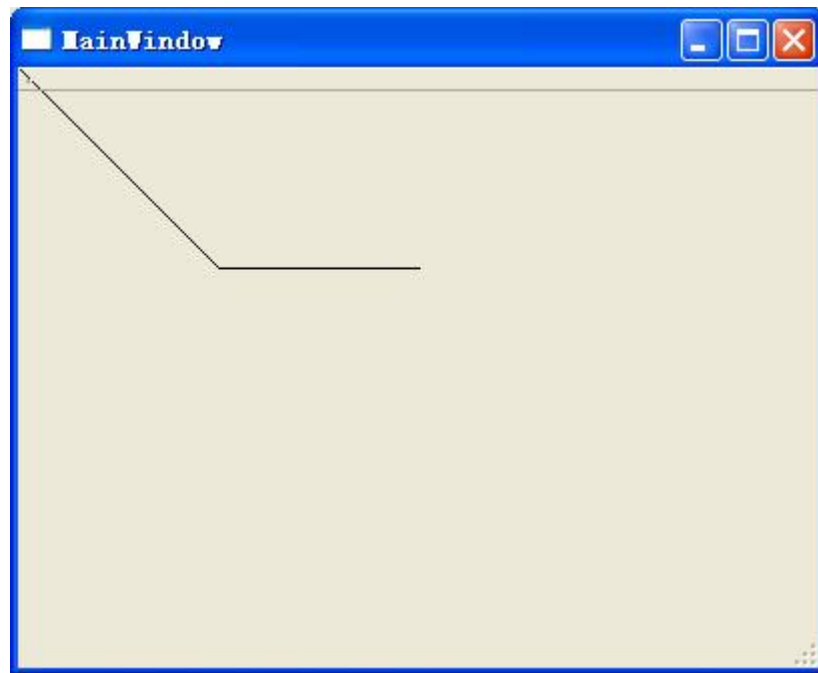
可以看到，对于已经绘制好的路径，可以非常简单的进行重复绘制。

### 三、绘制图形时的当前位置

1. 我们先来看个例子，将 `paintEvent()` 函数更改如下：

```
void MainWindow::paintEvent(QPaintEvent *)
{
    QPainterPath path;
    path.lineTo(100,100);
    path.lineTo(200,100);
    QPainter painter(this);
    painter.drawPath(path);
}
```

程序运行效果如下图所示。

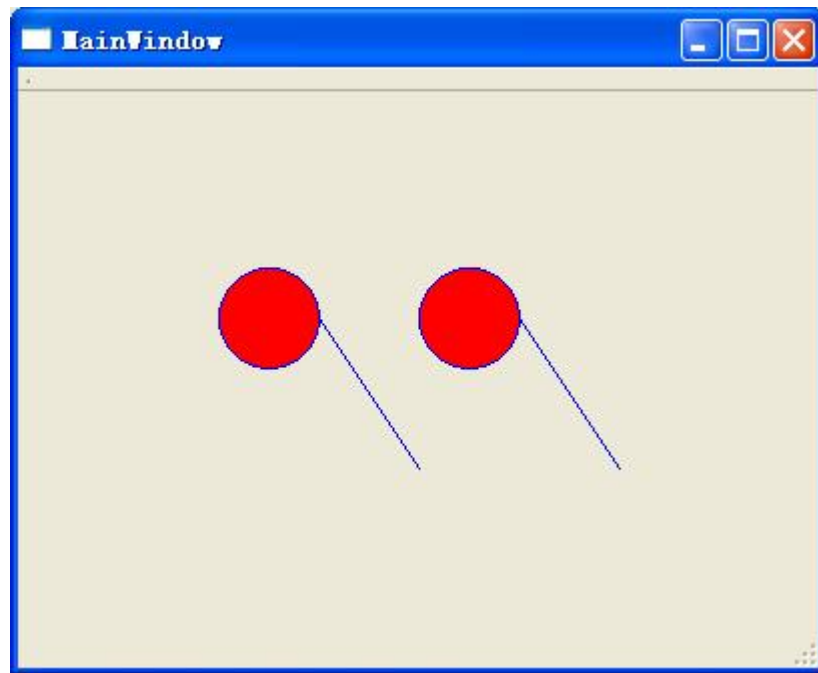


可以看到，创建路径后，默认是从 `(0, 0)` 点开始绘制的，当绘制完第一条直线后当前位置是 `(100, 100)` 点，从这里开始绘制第二条直线。绘制完第二条直线后，当前位置是 `(200, 100)`。

2. 再来看一个例子。将 `paintEvent()` 函数的内容更改如下：

```
void MainWindow::paintEvent(QPaintEvent *)
{
    QPainterPath path;
    path.addRect(50, 50, 40, 40);
    path.lineTo(200, 200);
    QPainter painter(this);
    painter.drawPath(path);
}
```

运行程序，效果如下图所示。



可以发现，当绘制完矩形后，当前位置在矩形的左上角顶点，然后从这里开始绘制后面的直线。

4. 我们也可以使用 `moveTo()` 函数来改变当前点的位置。例如将上面的代码更改为：

```
void MainWindow::paintEvent(QPaintEvent *)
{
    QPainterPath path;
    path.addRect(50, 50, 40, 40);
    //移动到(100, 100)点
    path.moveTo(100, 100);
    path.lineTo(200, 200);
    QPainter painter(this);
    painter.drawPath(path);
}
```

这样当绘制完矩形以后，就会移动到 `(100, 100)` 点进行后面的绘制。程序运行效果如下图所示。

