



ーキ、一二甲

Qing



博客等级: 18

博客积分: 78篇

博客访问: 403,304

关注人气: **378**

获赠金笔: 0支

赠出金笔: 0支

荣誉徽章: 🛐





相关博文

更多〉〉

推荐博文

- 2550mAh电
- ■日本雾霾之战: 民众与政府的博弈
- ■南非猎豹冒死捕杀豪猪被扎满嘴刺
- ■别光顾着把枪口对准柴静
- 马来西亚男子与眼镜王蛇接吻(图
- ■【早评】降息利好将会在后市中逐
- ■哪些人能从柴静的"穹顶之下"赚
- "深圳机场撞人事件"不是一个人
- ■政采剔除国外品牌: 早该说不了!
- 【DIY】做一枚宫灯迎元宵





青蛙冒险坐鳄鱼 广西北海银滩 鼻子 景

Qt之二维绘图 (2013-07-03 11:19:31) [编辑][删除]

+ 转 载 ▼

标签: qt 二维绘图 圆角窗体 qt圆角 it 分类: Qt

学习Qt,那么二维绘图必不可少!而且在开发界面的时候很多样式(点、线、矩形、弧形、饼状图、多边形、贝塞尔弧线等)都会用到,所以建议认真学习二维绘图!

Qt的二维图形引擎是基于QPainter类的,QPainter既可以绘制几何图形,也可以绘制像素映射、图像和文字。此外,QPainter也支持一些高级特性,例如反走样(针对文字和图形边缘)、像素混合、渐变填充和矢量路径等,QPainter也支持线性变换,例如平移、旋转、错切和缩放。

QPainter可以画在"绘图设备"上,例如: QWidget、QPixmap、QIamge或者QSvgGenerator。QPainter也可以与QPrinter一起使用来打印文件盒创建PDF文档。这意味着通常可以用相同的代码在屏幕上显示数据,也可以生成打印形式的报告。

如果要在绘图设备(一般为窗口部件)上绘图,只需创建一个QPainter,再将指针传到该设备中。

例如:

void MyWidget::paintEvent(QPaintEvent *event)

QPainter painter(this);

使用QPainter的draw...()函数,可以绘制各种各样的形状。图8.1列出了其中最重要一些函数。绘制效果取决于QPainter的设置,一些是从设备中取得的,然而有些被初始化成默认值。三个主要设置是画笔、画刷、字体:

画笔: 用来画线和边缘。它包含颜色、宽度、线性、拐点风格以及连线风格。

画刷:用来填充几何图形的图案。它一般由颜色和风格组成,但同时也可以是纹理(一个不断重复的图像)或者是一个渐变。

字体: 用来绘制文字。字体有很多属性,包括字体族和磅值大小。

可以随时调用QPen、QBrush或者QFont对象的setPen()、setBrush()和setFont()来修改这些设置。

draw...,见明知义。绘制点drawPoint()、绘制直线drawLine()、绘制折线drawPolyLine()、绘制多点drawPoints()、绘制多直线drawLines()、绘制矩形区域drawRect()、绘制圆角区域drawRoundRect()、绘制椭圆drawEllipse()、绘制背景图片drawPixmap()等!

举例:

绘制直线:

1、在当前窗口绘制

void myWidget::paintEvent(QPaintEvent *event)

QPainter painter(this);

painter. setPen(Qt::gray); //设置画笔为灰色

painter.drawLine(0, 0, 100, 100); //(0, 0)为初始坐标, (100, 100)为最终坐标

2、在当前窗体上的子组件绘制

Qt之二维绘图_一去、二三里_新浪博客





监狱

萨尔瓦多城铁笼 狮子金沙禁猎区 旅馆





潜水员勇喂巨型 双髻鲨

俄罗斯山中隐士 生活

查看更多>>

谁看过这篇博文

	自然度	3月2日
	Hujun-whu	3月1日
	todaysky	2月27日
	右右youyou	2月24日
	Cissy	2月18日
	等待中Wai…	2月10日
	小妞妞	2月6日
2	卫城	2月5日

amgkun0720 1月28日 1月27日 nhf_hit

☆ 蒾茫 1月27日 🙎 lbdream 1月26日

```
{
}
```

```
paintEvent()可以实现图形的绘制,前提是绘制当前窗体!如果界面上有其它组件,如何来绘制呢?
   (1) 对子组件自定制,可以重新实现一个类,实现paintEvent()
   (2) 添加监听器line label->installEventFilter(this), 实现eventFilter()。
   关于(1)就不再多讲,同1,(2)代码如下:
line label->installEventFilter(this);
bool myWidget::eventFilter(QObject *obj, QEvent *event)
 if(obj == line_label)
 if(event->type() == QEvent::Paint)
  int label_height = line_label->height();
   int label_width = line_label->width();
  QPainter painter(line_label);
  painter.setPen(QPen(Qt::gray, 1, Qt::DashLine));
  painter.drawLine(label_width/2, 0, label_width/2, label_height);
 return QWidget::eventFilter(obj, event);
   这样就可以实现在myWidget窗体上的QLabel的绘制!
   优劣性:如果窗口子部件较多,若每个部件的绘制相同,则可采用(1),若不相同,那么根据(1)就会实
现较多的类,而(2)只需要添加多个监听器即可,建议采用方式(2)!
绘制背景图片:
void myWidget::paintEvent(QPaintEvent *event)
QPainter painter(this);
painter.drawPixmap(rect(), QPixmap(skin_name));
绘制矩形:
void myWidget::paintEvent(QPaintEvent *event)
QPainter painter (this);
QLinearGradient linear(rect().topLeft(), rect().bottomLeft());
linear.setColorAt(0, Qt::red);
 linear.setColorAt(0.5, Qt::green);
linear.setColorAt(1, Qt::blue); //设置红、绿、蓝变化
painter. setPen(Qt::gray); //设定画笔颜色, 到时侯就是边框颜色
painter.setBrush(linear); //设置画笔, 到时候就是区域颜色
 painter.drawRect(QRect(0, 30, this->width(), this->height()-30)); //设置绘制区域
绘制折线:
void myWidget::paintEvent(QPaintEvent *event)
QPainter painter(this);
painter.setPen(Qt::gray);
static const QPointF points[4] = {QPointF(0, 30), QPointF(0, this->height()-1), QPointF(this-
\width()-1, this-\height()-1), \QPointF(this-\width()-1, \dot{30});
painter.drawPolyline(points, 4); //由4个点连成的折线
绘制圆角:
setWindowFlags(Qt::FramelessWindowHint); //去掉标题栏
setAttribute(Qt::WA_TranslucentBackground); //不被绘制上的部分设置透明
void myWidget::paintEvent(QPaintEvent *event)
QPainter painter (this);
```

```
QBrush brush;
brush.setTextureImage(QImage(background_image));
painter.setBrush(brush); //设置画刷,绘制背景图片
painter.setPen(Qt::black); //设置画笔, 绘制边框色
painter.drawRoundedRect(QRect(0, 0, this->width()-1, this->height()-1), 5, 5); //绘制圆角, 像素为
5
  就我所知,setAttribute(Qt::WA_TranslucentBackground)有一定的弊病,当窗体最小化
(showMinimized())后,再次显示时,窗体上的组件就会失去焦点!
  好了,二维绘图基本就介绍到这里,代码实现可以不尽相同,只要掌握原理,实现起来就会游刃有余!
注:
  技术在于交流、沟通、转载请注明出处并保持作品的完整性。
  作者: └☆奋斗ing♥孩子` 原文: http://blog. sina. com. cn/s/blog_a6fb6cc90101e91m. html.
                           貪喜欢
分享: 🗐 微米 🚨 🖂 👩 😘 🚇 人 🙀 豆 🛨
阅读 | 评论 | 收藏 | 已有8人转载▼ | 喜欢▼ | 打印
                                                        已投稿到: 🚧 排行榜
前一篇: Qt之360安全卫士(换肤)
后一篇: Qt之界面切换(360、新浪、人人界面,多语化切换随心所欲)
评论 重要提示:警惕虚假中奖信息
                                                               [发评论]
  新浪网友
  很好的内容对我有帮助
  1月19日 11:20
                                                              回复(0)
发评论
□ 😽 分享到微博 📠
                                                              ■匿名评论
验证码: 请点击后输入验证码 收听验证码
                               发评论
                 以上网友发言只代表其个人观点,不代表新浪网的观点或立场。
〈前一篇
                                                               后一篇 >
Qt之360安全卫士(换肤)
                                   Qt之界面切换(360、新浪、人人界面,多语化切换随
                                                               心所欲)
```

新浪BLOG意见反馈留言板 不良信息反馈 电话: 4006900000 提示音后按1键(按当地市话标准计费) 欢迎批评指正 新浪简介 | About Sina | 广告服务 | 联系我们 | 招聘信息 | 网站律师 | SINA English | 会员注册 | 产品答疑

> Copyright © 1996 - 2014 SINA Corporation, All Rights Reserved 新浪公司 版权所有