





萨尔瓦多城铁笼监狱



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
	today'sky	2月27日
	右右youyou	2月24日
	Cissy	2月18日
	等待中Wai...	2月10日
	小姐姐	2月6日
	卫城	2月5日
	wangkun0720	1月28日
	mhf_hit	1月27日
	莲茫	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) 对子组件自定义，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，到时候就是边框颜色
    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, 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](http://blog.sina.com.cn/s/blog_a6fb6cc90101e91m.html)。



分享：

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 不良信息举报 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