

# muzi9\_17的专栏

目录视图

摘要视图

RSS 订阅

个人资料



muzi9\_17

访问： 123111次  
积分： 1497  
等级：   
排名： 第17584名

原创： 35篇    转载： 17篇  
译文： 0篇    评论： 17条

文章搜索

文章分类

鸟哥的Linux私房菜 (0)  
Linux (1)

文章存档

2011年12月 (3)  
2011年09月 (14)  
2011年08月 (10)  
2011年07月 (3)  
2011年05月 (11)

阅读排行

MFC画图（画线、画矩形  
(31685)  
MFC控件（三）（进度条  
(17154)  
MFC控件（四）（列表框  
(12531)  
MFC控件（一） (5211)  
MFC字体（二）（控件字 (5156)  
MFC控件（四）（树视图 (4673)  
默认构造函数和默认复制 (4063)  
MFC控件（四）（图像列 (4042)  
VC++深入解析（六）（ (3575)  
MFC字体（一）（控件文 (3196)

2016攒课第二期之你听课我买单，快来攒你想听的课！  
架构免费公开课报名  
【UDN沙龙】Intel大拿分享HTML5时代的跨平台开发解

## MFC画图（画线、画矩形、画刷画笔的使用）

标签： mfc windows null 文档

2011-04-30 22:01    31687人阅读    评论(5)    收藏    举报

版权声明：本文为博主原创文章，未经博主允许不得转载。

绘制任意线条：

1、创建基于单文档工程：Draw

CDrawView类中添加成员变量：

CPoint m\_ptOrigin;

BOOL m\_bDraw;

2、CDrawView() 中初始化：

m\_ptOrigin = point;

m\_bDraw = false;

3、添加消息响应函数：

WM\_LBUTTONDOWN    WM\_LBUTTONUP    WM\_MOUSEMOVE

4、左键按下时：

m\_ptOrigin = point;

m\_bDraw = true;

5、左键抬起：

m\_bDraw = false;

6、鼠标移动时：

//画连续线条

CClientDC dc(this);

if(m\_bDraw)

{

dc.MoveTo(m\_ptOrigin);

dc.LineTo(point);

m\_ptOrigin = point;

}

//画扇形线条

CClientDC dc(this);

if(m\_bDraw)

{

dc.MoveTo(m\_ptOrigin);

dc.LineTo(point);

}

//画带外包线的扇形线条

析：画带外包线的扇形线条，不光记录初始位置，还要记录鼠标移动上一位置。

添加成员变量：

CPoint m\_ptOld;

在鼠标左键按下时初始化：

评论排行

<a href="#">MFC画图（画线、画矩形</a>	(5)
<a href="#">MFC控件（四）（列表视</a>	(4)
<a href="#">MFC控件（四）（树视图</a>	(3)
<a href="#">MFC控件（三）（进度条</a>	(2)
<a href="#">VC++深入解析（六）（</a>	(1)
<a href="#">MFC控件（四）（图像列</a>	(1)
<a href="#">作为程序员的出路</a>	(1)
<a href="#">【转】VC++技术内幕（</a>	(1)
<a href="#">VC++技术内幕（五_1）</a>	(0)
<a href="#">x86, x86-64, i386, IA32,</a>	(0)

推荐文章

- [\\*Android从启动到程序运行发生的事情](#)
- [\\*Android杂谈之RadioGroup+ViewPager制作的底部导航栏](#)
- [\\*浅谈Storm流式处理框架](#)
- [\\*Oculus Rift, HTC Vive, SONY PSVR的全面对比](#)
- [\\*Android移动应用开发中常见的经验技巧总结](#)
- [\\*Rxjava原理探索：切换线程，变换](#)

最新评论

- [MFC画图（画线、画矩形、画刷](#)  
[csdnqixiaoxin](#): 挺好的
- [MFC控件（四）（列表视图控件](#)  
[yeqingbo2010](#): 很有用，谢谢你了，楼主！资料归档能力太强了！向你学习！
- [MFC控件（四）（列表视图控件](#)  
[古往今来](#): 在视图列表控件中添加滚动条
- [MFC控件（四）（列表视图控件](#)  
[古往今来](#): 很有用
- [MFC控件（三）（进度条控件和](#)  
[hj974957894](#):  
[@xzjlcyz](#):SetRange32这个函数对应的成员变量的声明有问题，你看看是不是用的CProg...
- [MFC控件（四）（树视图控件Tr](#)  
[小静静静静静](#):[m\\_SellItemList](#) 你这个东西是怎么定义的呢？
- [MFC控件（四）（树视图控件Tr](#)  
[小静静静静静](#): 好几年了。。我也找到这里来了。
- [MFC控件（四）（列表视图控件](#)  
[haz2yy](#): 很有用，可以收藏。。。
- [作为程序员的出路](#)  
[Killer\\_PatrickJane](#): 我是一名从业1年多的程序员，都说是青春饭，也很担忧自己的将来发展。很好奇博主的现在选择。
- [【转】VC++技术内幕（一）](#)  
[xy\\_dl](#): 整理得很不错

m\_ptOld = point;

在鼠标移动时进行改变:

```
CClientDC dc(this);

if(m_bDraw)

{

    dc.MoveTo(m_ptOrigin);

    dc.LineTo(m_ptOld);

    dc.MoveTo(m_ptOld);

    dc.LineTo(point);

    m_ptOld = point;

}
```

画矩形:

[画无边框的矩形](#)

方法说明：利用FillRect函数填充矩形。填充内容为画刷的内容，即画刷对象的指针。

```
CBrush brush( RGB(5,255,255) );

CClientDC dc(this);

dc.FillRect( CRect(m_ptOrigin,point), &brush );
```

[以位图作为画刷的资源](#)

```
CBitmap bitmap;

bitmap.LoadBitmap( IDB_BITMAP1 );

CBrush brush( &bitmap );

CClientDC dc(this);

dc.FillRect( CRect(m_ptOrigin,point), &brush );
```

[画有边框的矩形](#)

方法说明：利用Rectangle方法来画矩形。其中CBrush与CPen类分别负责矩形内部和矩形边框的属性

```
//画中心为白色的矩形

CClientDC dc(this);

dc.Rectangle( CRect(m_ptOrigin,point) );


//画中心为透明的矩形

CClientDC dc(this);

CBrush *pBrush=CBrush::FromHandle( (HBRUSH)

GetStockObject( NULL_BRUSH ) );

CBrush *pOldBrush=dc.SelectObject( pBrush );

dc.Rectangle( CRect(m_ptOrigin,point) );

pOldBrush=pBrush;


//画中心为透明边为任意颜色的矩形

CClientDC dc(this);

CPen pen( PS_SOLID,1,RGB(234,23,53) );

CPen *pOldPen=dc.SelectObject( &pen );

CBrush *pBrush=CBrush::FromHandle( (HBRUSH)

GetStockObject( NULL_BRUSH ) )

CBrush *pOldBrush=dc.SelectObject( pBrush );

dc.Rectangle( CRect(m_ptOrigin,point) );

dc.SelectObject( pOldPen );

dc.SelectObject( pOldBrush );
```

使用画笔、画刷

在OnLButtonUp函数中:

```
////////////////////////////////////

CPen pen( PS_SOLID,1,RGB(234,23,53) );

//定义一个CPen类对象，并利用RGB宏改变其颜色

CClientDC dc(this);
```

```
CPen *pOldPen=dc.SelectObject(&pen); //将其选入设备表

dc.MoveTo(m_ptOrigin);

dc.LineTo(point);

dc.SelectObject(pOldPen);

////////////////////////////////////

CBrush brush( RGB(0,255,0));

CClientDC dc(this);

CRect rect(m_ptOrigin,point);

dc.FillRect(&rect,&brush);

////////////////////////////////////
```

使用位图创建画刷：

插入->资源->位图->引入, 可以引入位图，初始一般命名为：IDB\_BITMAP1

```
CBitmap bitmap;

bitmap.LoadBitmap(IDB_BITMAP1);

CBrush brush(&bitmap);

CClientDC dc(this);

CRect rect(m_ptOrigin,point);

dc.FillRect(&rect,&brush);

////////////////////////////////////

CBrush *brush;

brush = CBrush::FromHandle( (HBRUSH)GetStockObject(NULL_BRUSH));

CClientDC dc(this);

CBrush *pOldBrush =dc.SelectObject(brush) ;

dc.Rectangle(CRect(m_ptOrigin,point));

dc.SelectObject(pOldBrush) ;
```

注：

常用的绘图设备类对象的创建和将其引入到设备环境中的方法

在选择绘图设备类对象（GDI对象）进行绘图时，一般可按下列步骤进行：

- （1）定义一个GDI对象，然后调用相应的函数（如CreatePen、CreateSolidBrush等）创建此GDI对象。
- （2）将已构造出的GDI对象利用设备环境类对象的成员函数SelectObject选入到当前环境中，并同时 将原来的GDI对象保存起来。
- （3）绘图结束后，恢复当前设备环境中原来的GDI对象。

注意：在Windows系统中包含了一些库存的GDI对象，在第（3）步中可直接利用设备环境类（CDC）的成员函数SelectStockObject将一个Windows库存的GDI对象选入，系统可自动将原来选入的GDI对象从设备环境中分离出来。

顶 踩  
2 0

上一篇 CDC,CPaintDC,CClientDC,CWindowDC  
下一篇 MFC字体（一）（控件文字字体）

### 猜你在找

- |                                    |                             |
|------------------------------------|-----------------------------|
| Windows Server 2012 AD RMS 文档安全管理  | iOS小实例画字画线矩形画图动画iOS学习笔记从零开始 |
| Windows Server 2012 R2 高可用性管理      | Gdiplus学习2画笔与画刷             |
| 移动APP测试基础到进阶                       | MFC实现画图工具中的画线功能一            |
| Windows Server 2012 DNS Server 管理  | MFC实现画图工具中的画线功能二            |
| Windows Server 2012 DHCP Server 管理 | MFC实现画图工具中画线功能三             |



查看评论

3楼 [csdnqixiaoxin](#) 2015-10-27 14:34发表



挺好的

2楼 [lili13897741554](#) 2013-06-06 15:00发表



太简单

1楼 [q2423423](#) 2013-03-05 11:07发表



非常有用

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服

杂志客服

微博客服

webmaster@csdn.net

400-600-2320

| 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

