**暨南大学本科实验报告专用纸**

课程名称 Windows编程实验 指导教师 刘晓翔 成绩

实验项目名称 windows标准图形界面程序 实验项目编号

实验项目类验证实验地点三楼机房1学院智能科学与工程学院 专业 人工智能

学生姓名 王志涛 学号 2021102259 实验时间 2023 年 4 月 22 日

**1.实验目的**

1.1了解MFC消息映射机制；

1.2掌握如何在窗口中绘图和输出文本；

1.3掌握菜单编程（静态菜单与动态菜单）；

1.4掌握界面元素（标题栏、窗口背景、鼠标形状、浮动工具栏、状态栏、启动界面）的编程方法。

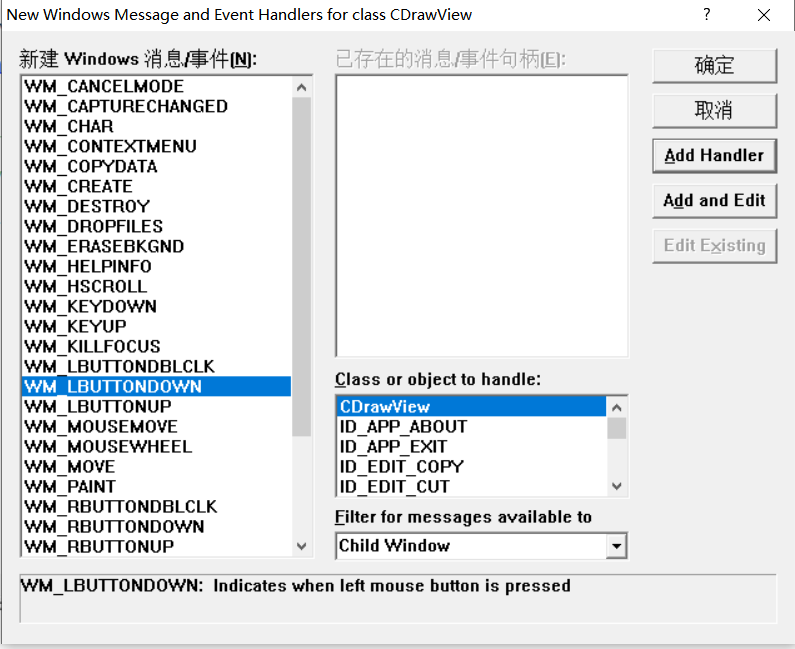
**2.实验要求**

利用Visual C++6.0软件开发工具编写程序，在实验二生成的VC工程基础上，完成一个具备图形/文本输出、菜单、自制工具栏、定制状态栏的Windows标准图形界面程序。

**3.实验过程**

**3.1消息映射机制**

新建一个工程名为Draw,在CDrawView中添加一个WM\_LBUTTONDOWN的响应函数。

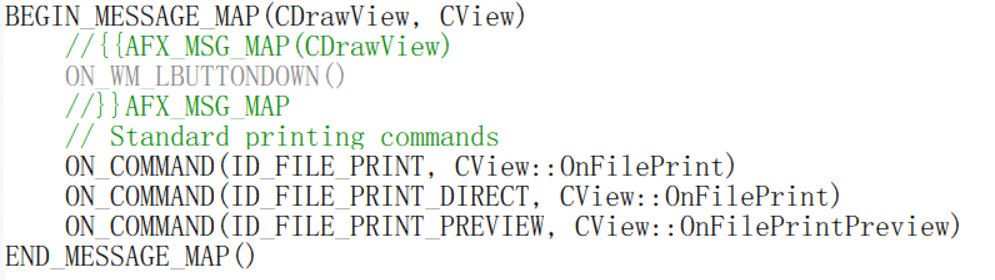


添加后，会在这三个地方增加代码

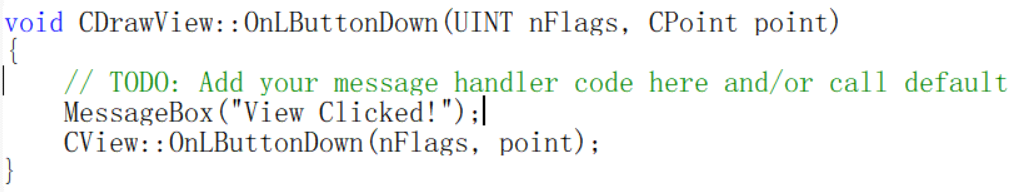
·在CDrawView类的头文件中，增加了灰色处的代码。在DECLARE\_MESSAGE\_MAP()之上有两个AFX\_MSG注释宏，位于这两个宏直接有一个OnLButtonDown函数原型，这个函数前面有一个afx\_msg限定符，这也是一个宏。该宏表示这个函数是一个消息响应函数的声明。



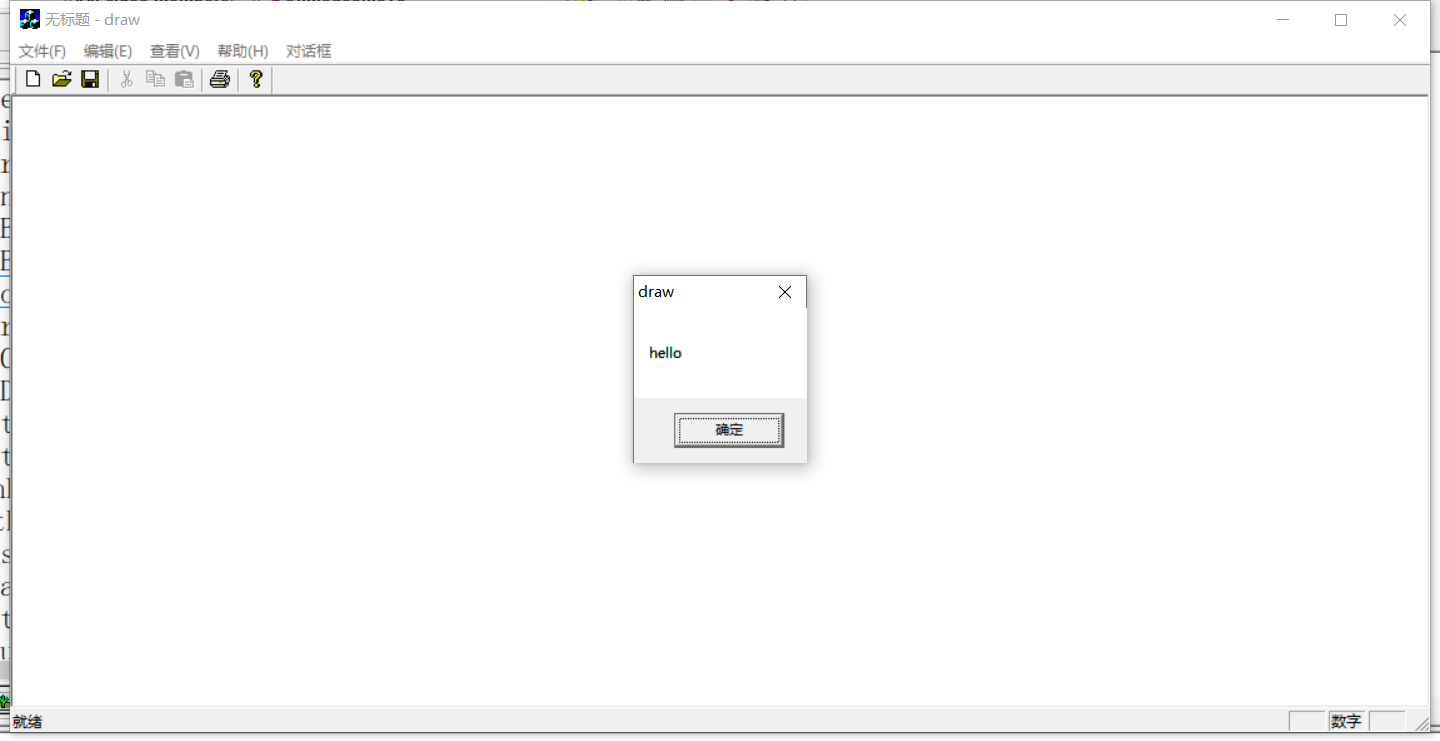
·在CDrawView类的源文件中，增加了灰色处的代码。BEFIN\_MESSAGE\_MAP和END\_MESSAGE\_MAP这两个宏之间定义了CDrawView类的消息映射表。本例中就是把WM\_LBUTTONDOWN和OnLButtonDown函数关联起来。



·在CDrawView类的源文件中，有着消息响应函数的定义（OnLButtonDown）



运行程序，点击右键，成功对鼠标右键消息进行响应

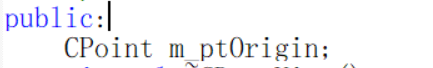


在第1章中，我们介绍了消息循环的概念：当有消息产生时，操作系统会将该消息添加到应用程序的消息队列中。应用程序通过消息循环来获取消息、翻译消息，并将消息提交给窗口过程函数。在窗口过程函数中，使用switch-case结构对消息进行处理。然而，在MFC程序中，只需按照上述步骤进行操作，并定义与消息相关的三个部分，就能实现消息的响应处理。这种消息处理机制被称为MFC消息映射机制。

3.2 在窗口中绘图

3.2.1 利用SDK全局函数实现画线功能

在CDrawView中添加了一个CPoint类型的成员变量，存储上一个点



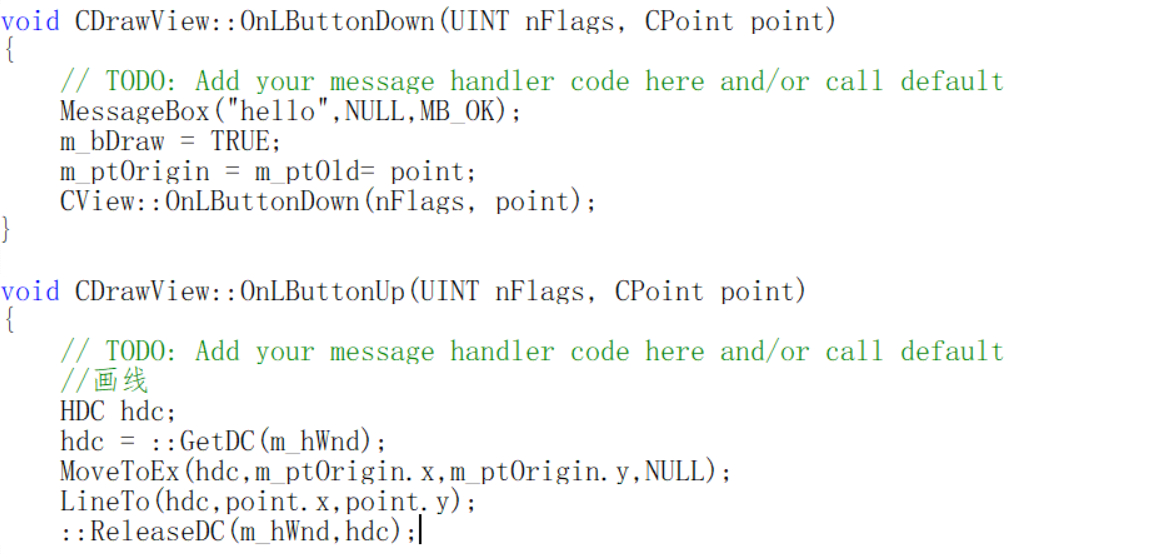
3.2窗口绘图和输出文本

3.2.1窗口绘图

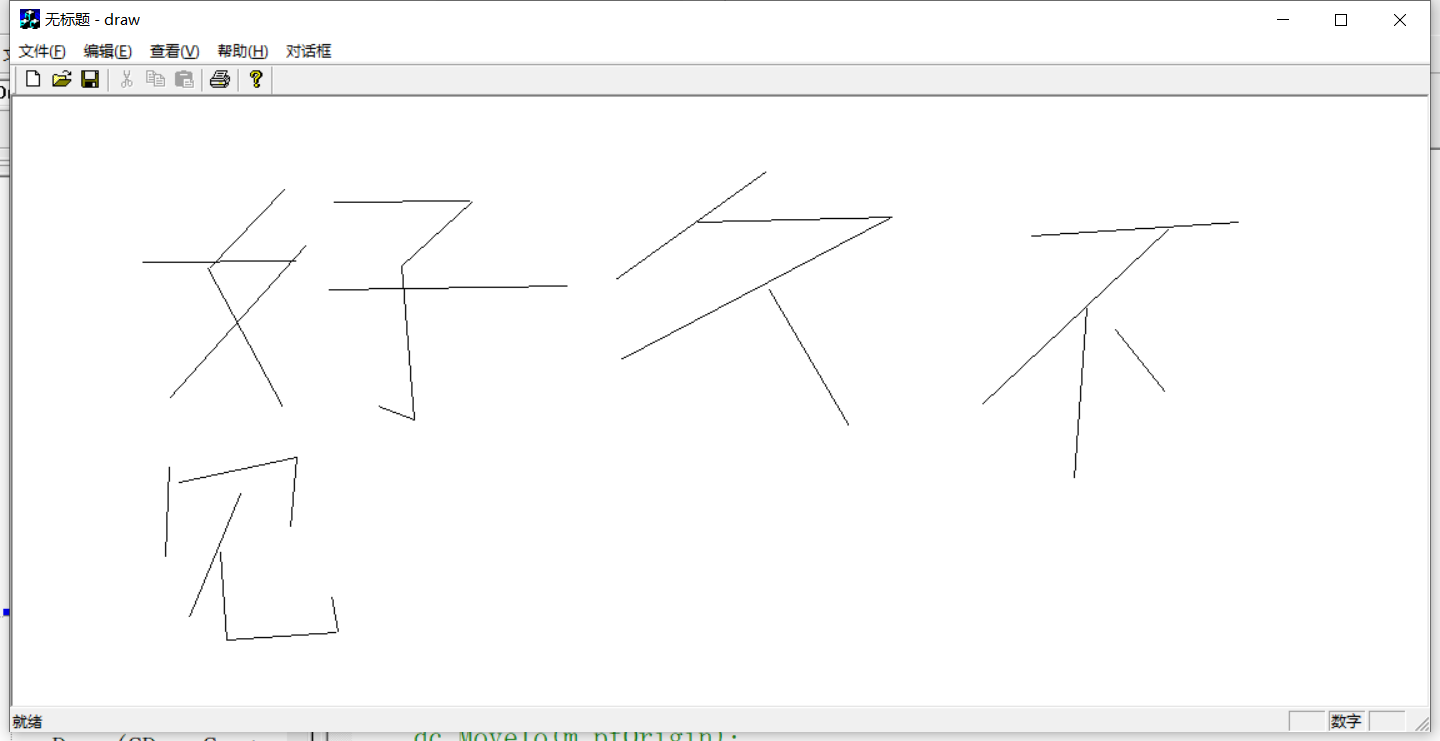
绘制线条

添加鼠标左键按下消息响应，将鼠标的坐标点存储起来。

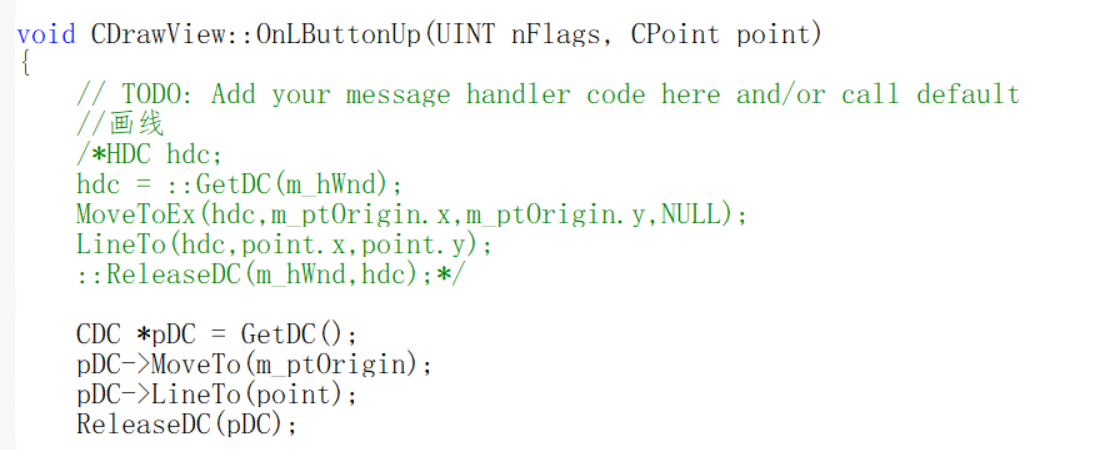
添加鼠标左键松开的消息，将刚刚的按下的坐标点和现在的坐标点作为线的两端



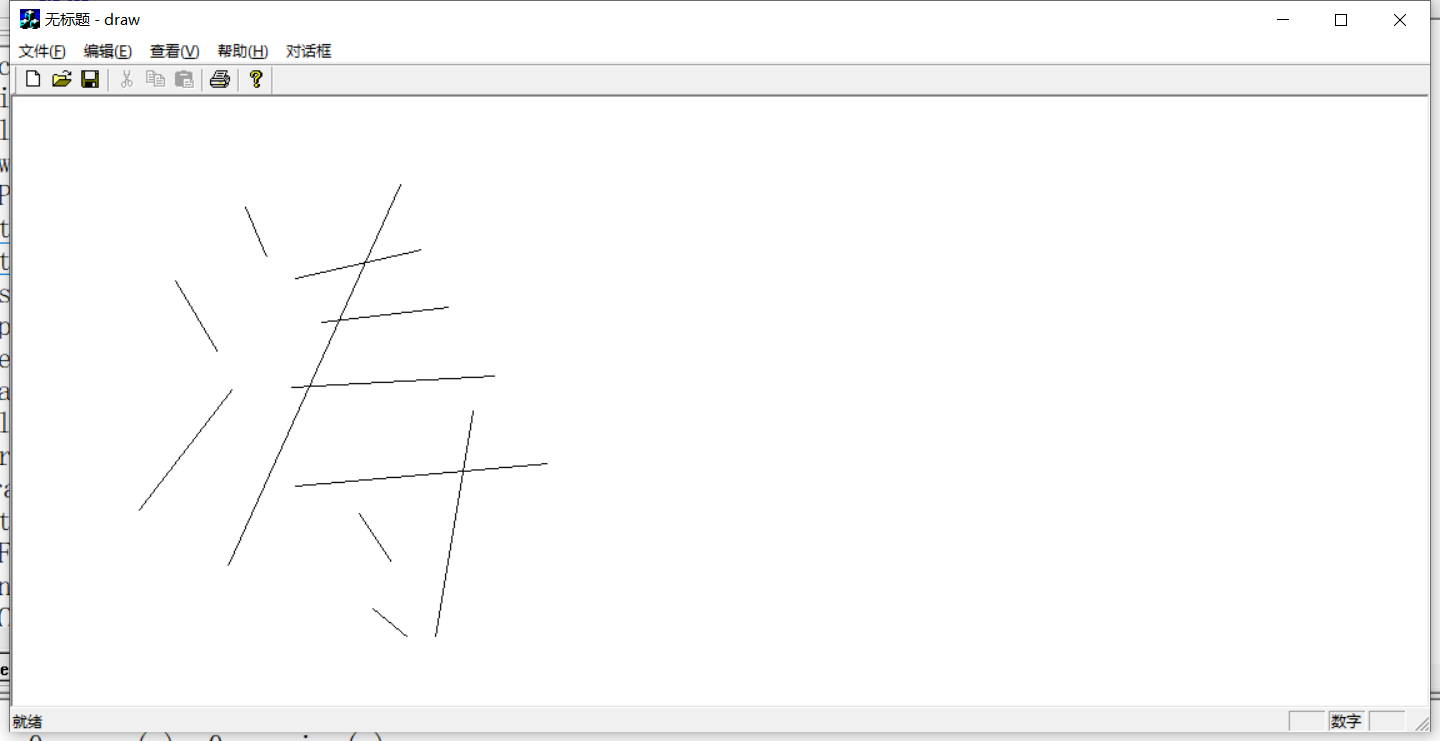
运行如图所示：



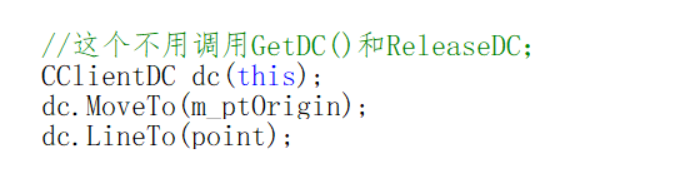
利用MFC的CDC类实现画线功能



画线效果如图：

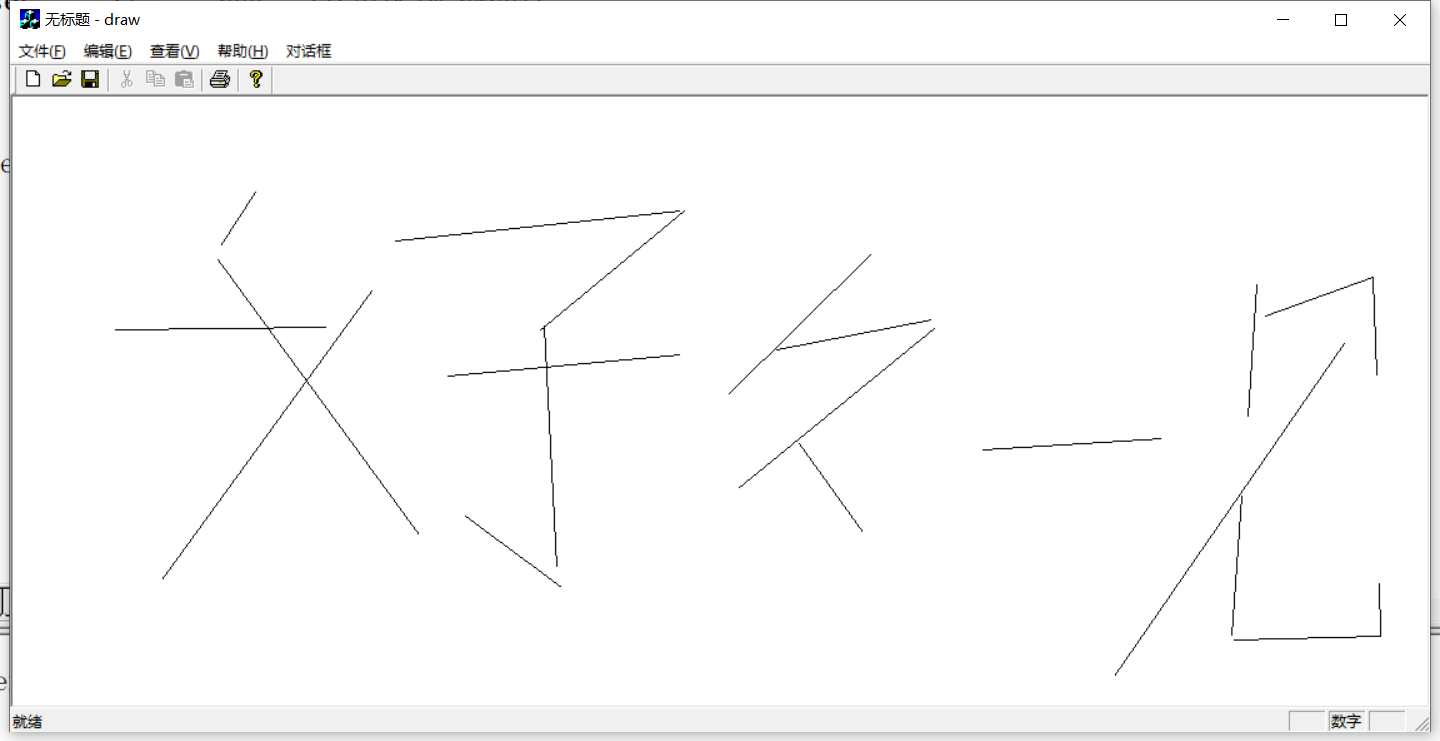


利用MFC的CClinetDC类实现画线功能

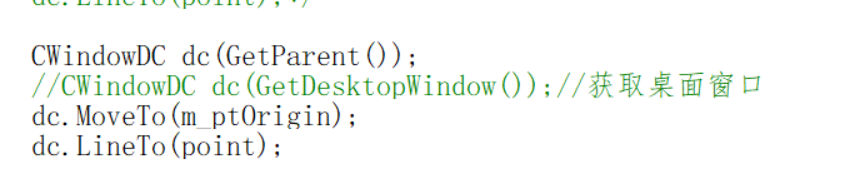


CClientDC是在窗口的客户区画线，并且该类对象可以自行析构，不需要显式释放。

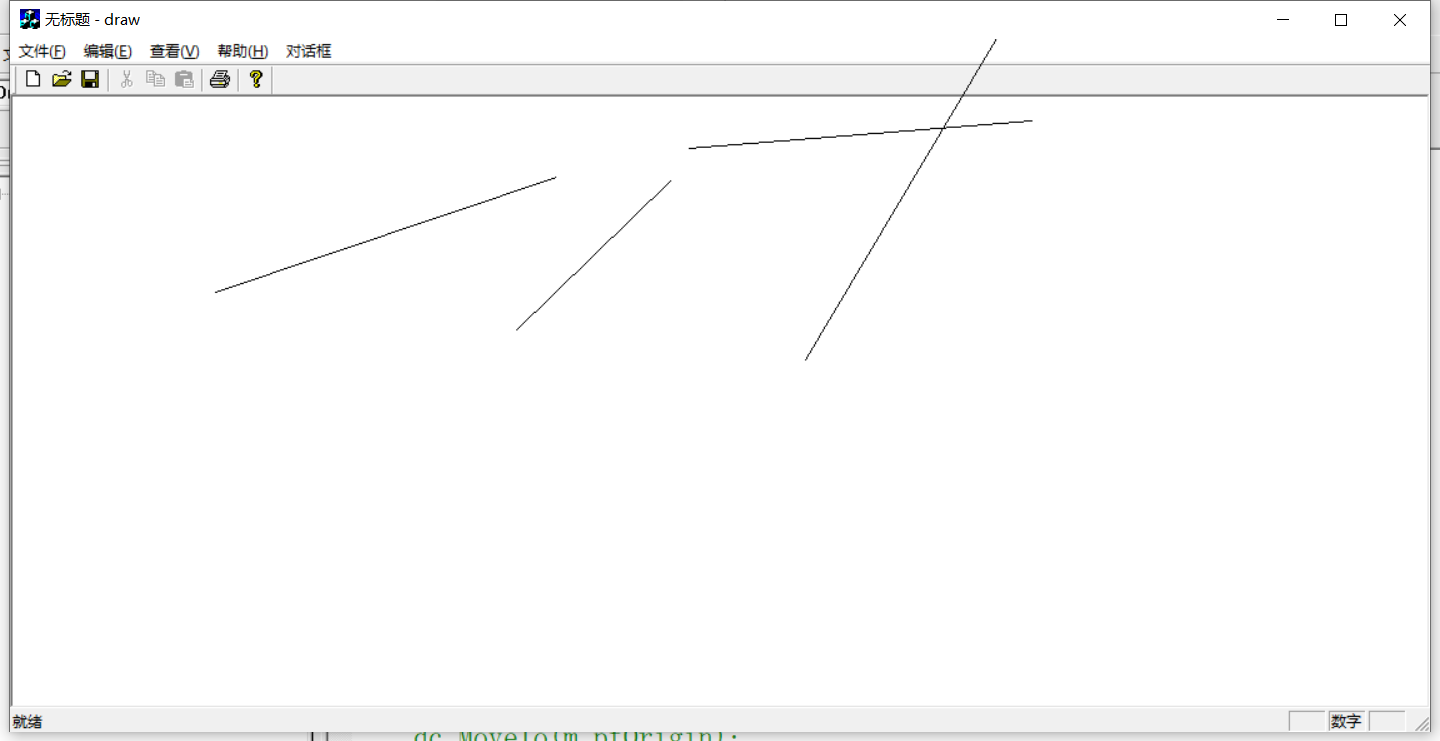
画图结果如下：



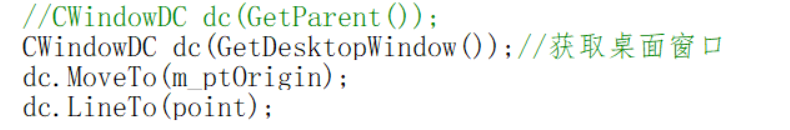
利用MFC的CWindowDC类实现画线功能



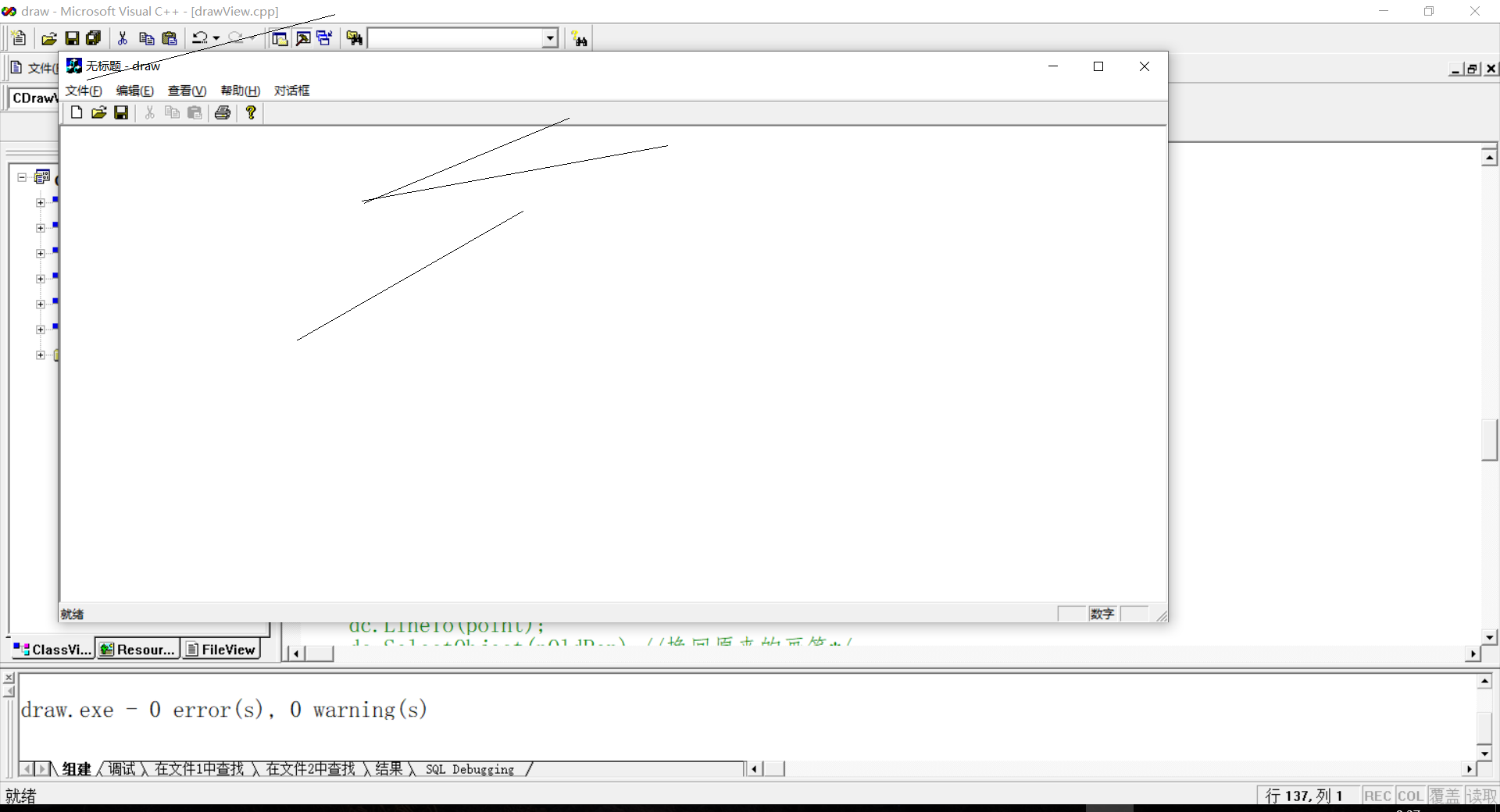
可以在非客户区和客户区画线



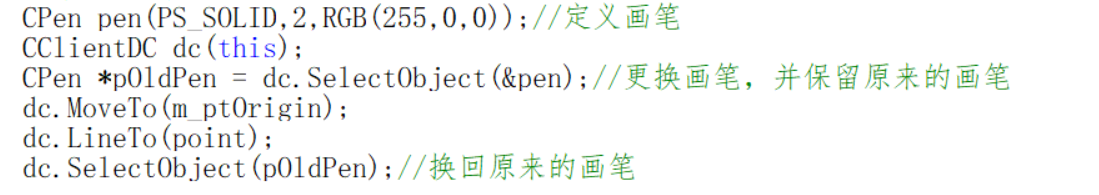
在桌面上画线



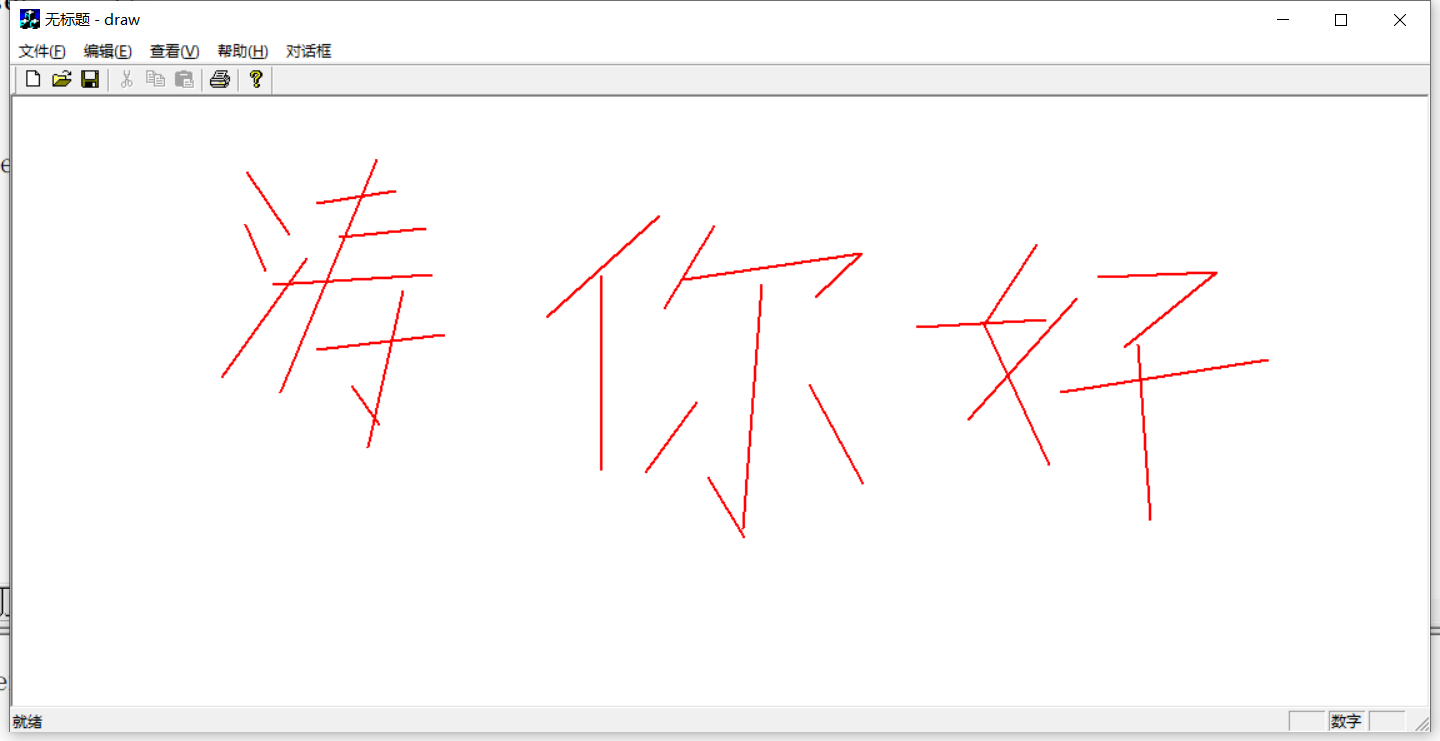
画线如下图所示：



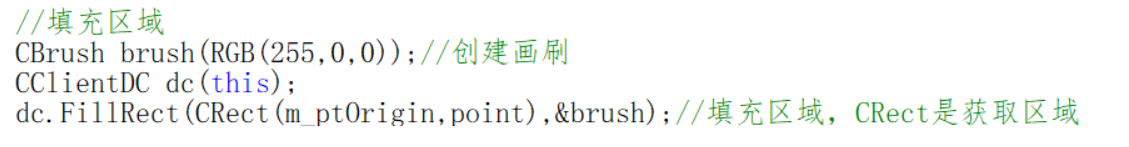
绘制彩色线条



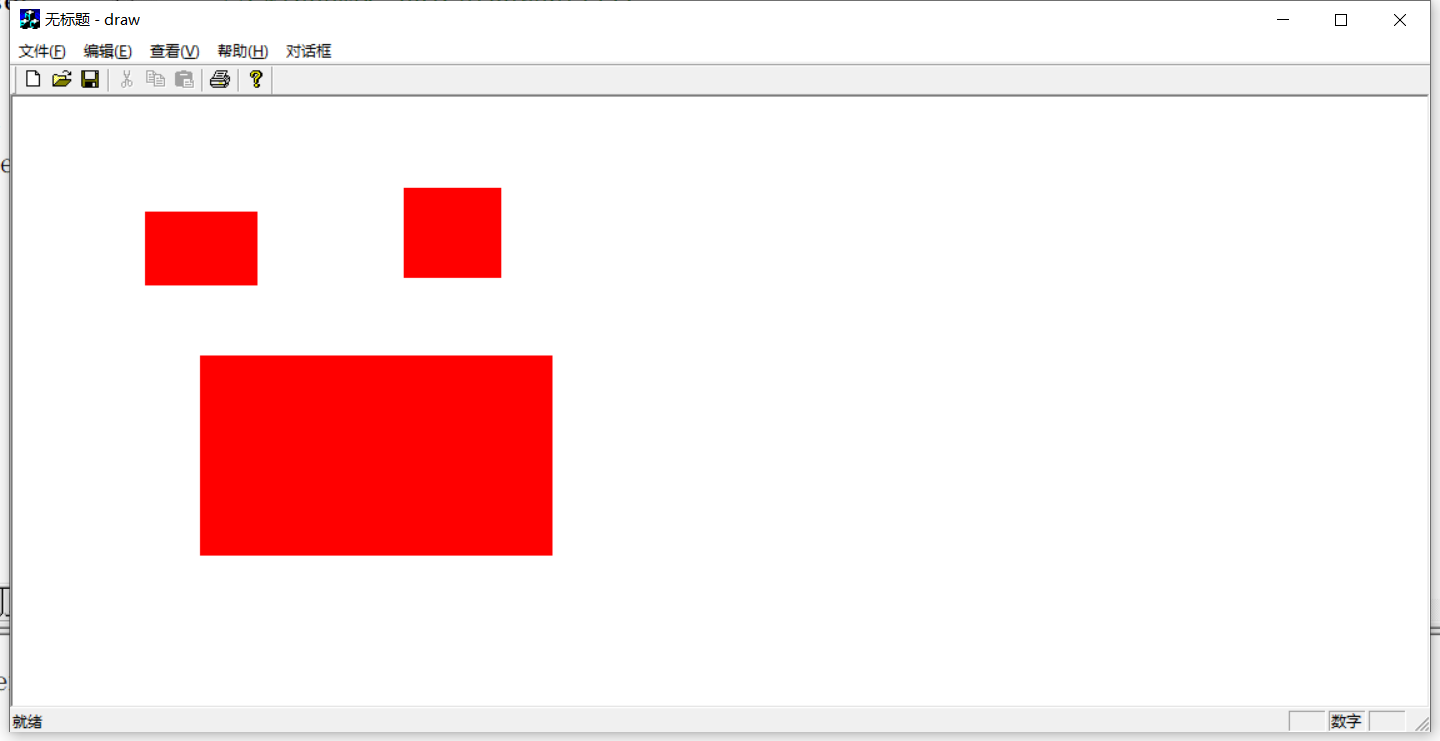
将画笔颜色改变就可以绘制彩色线条了



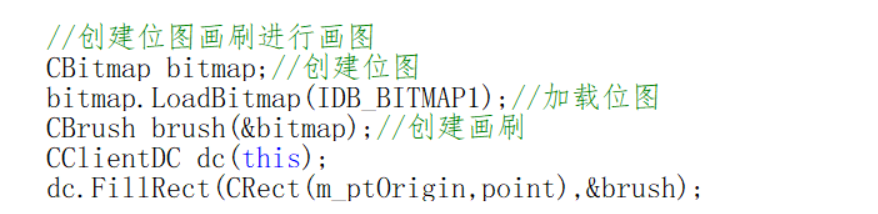
使用笔刷画线



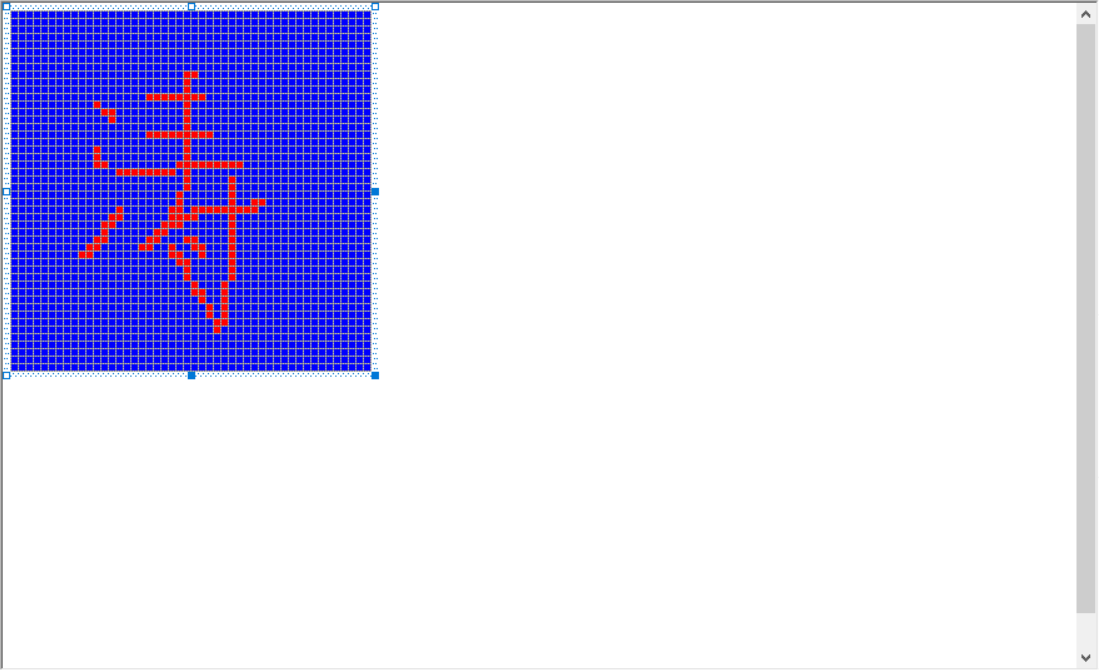
根据代码注释可以知道具体细节，我这里是创建矩形画刷,图形如下：



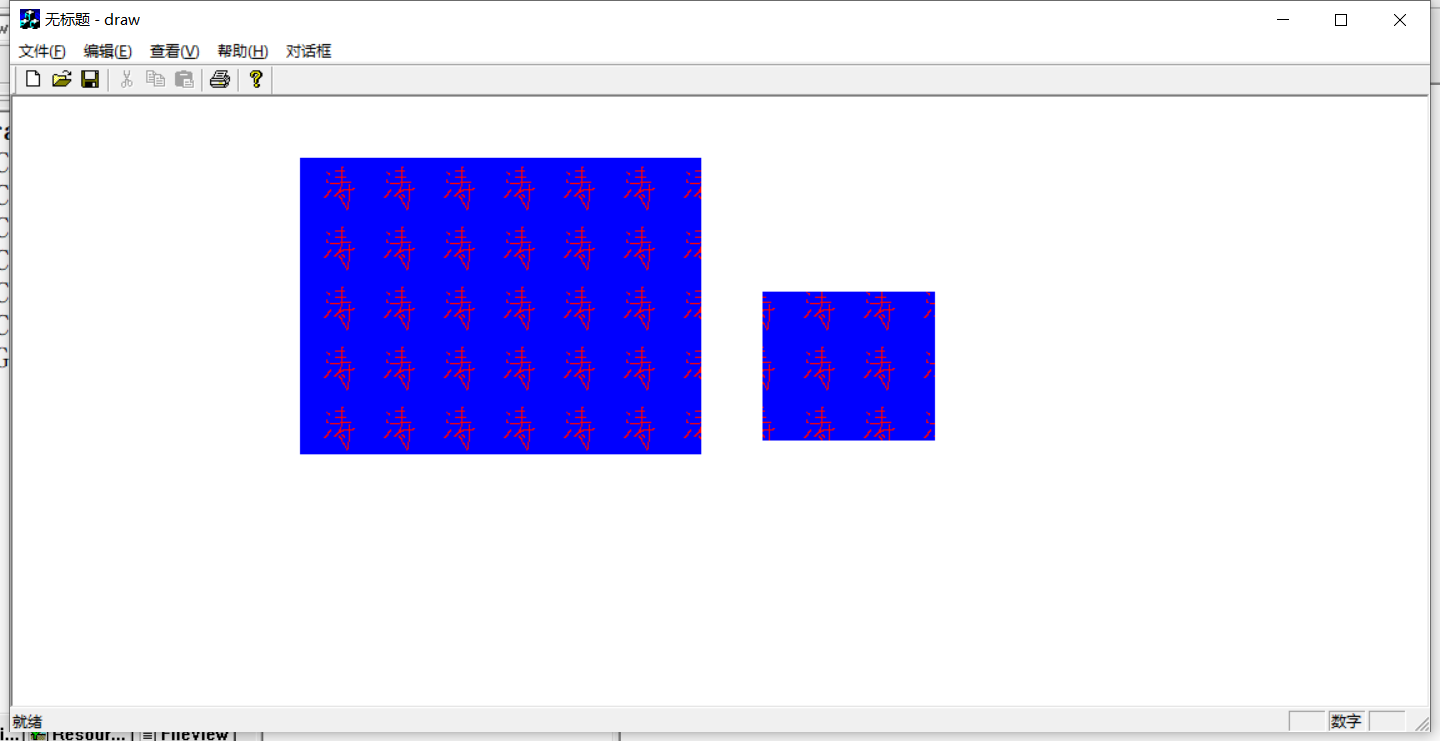
指定自己设置的位图填充



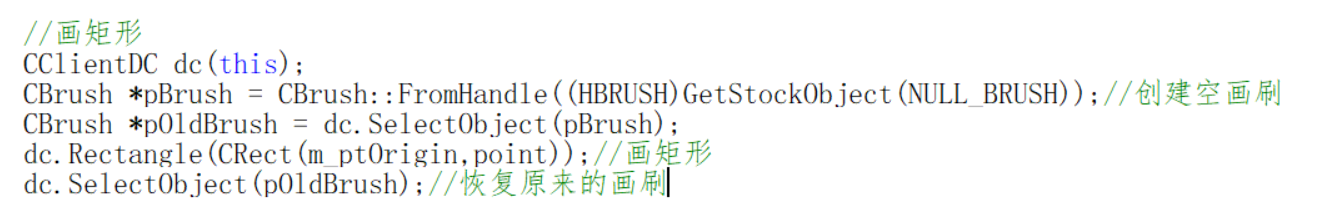
我的位图资源



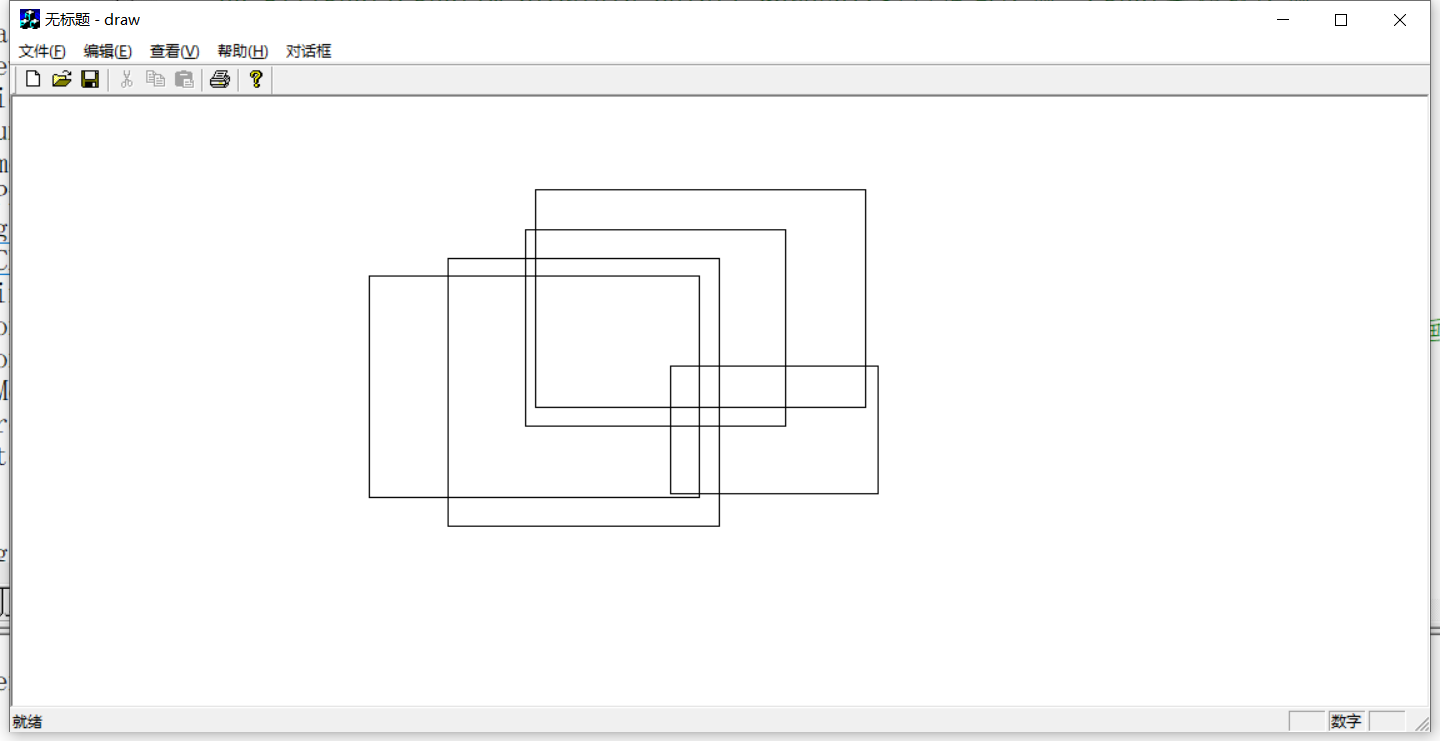
运行程序，画画



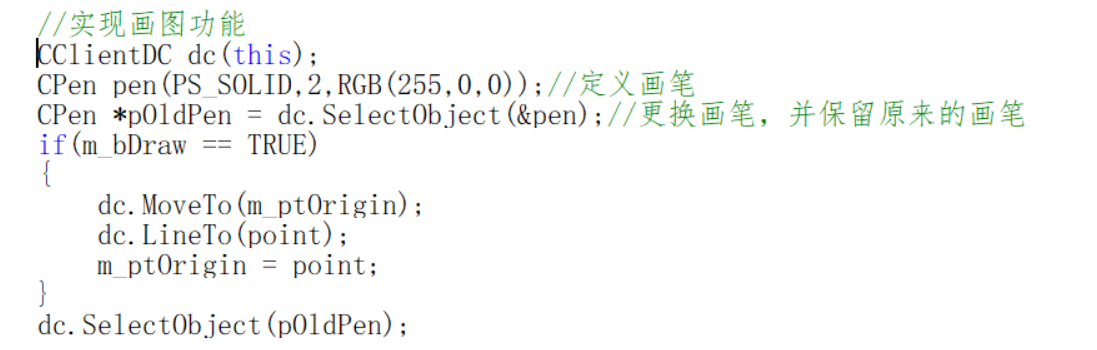
绘制矩形框



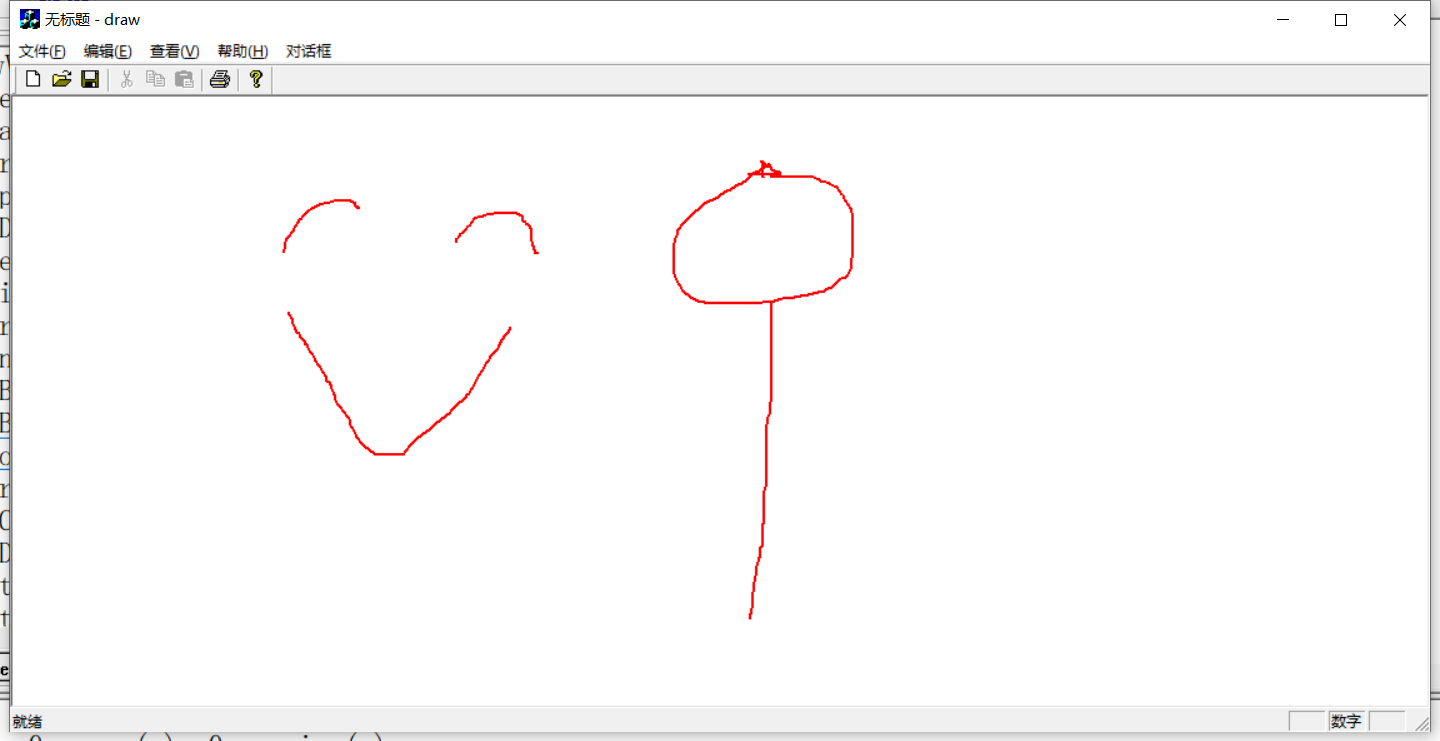
运行代码，画图：



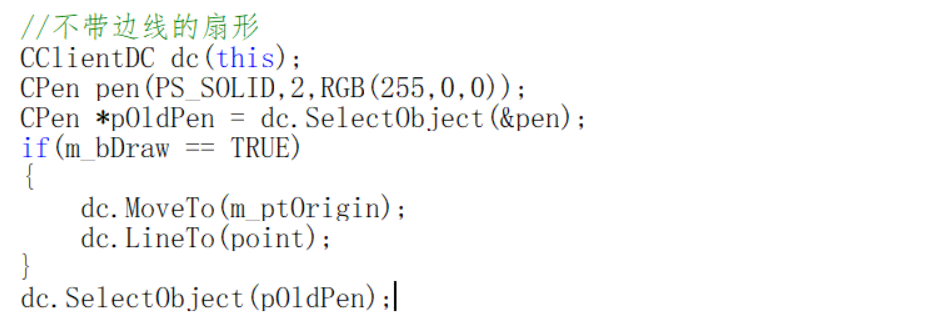
绘制连续曲线



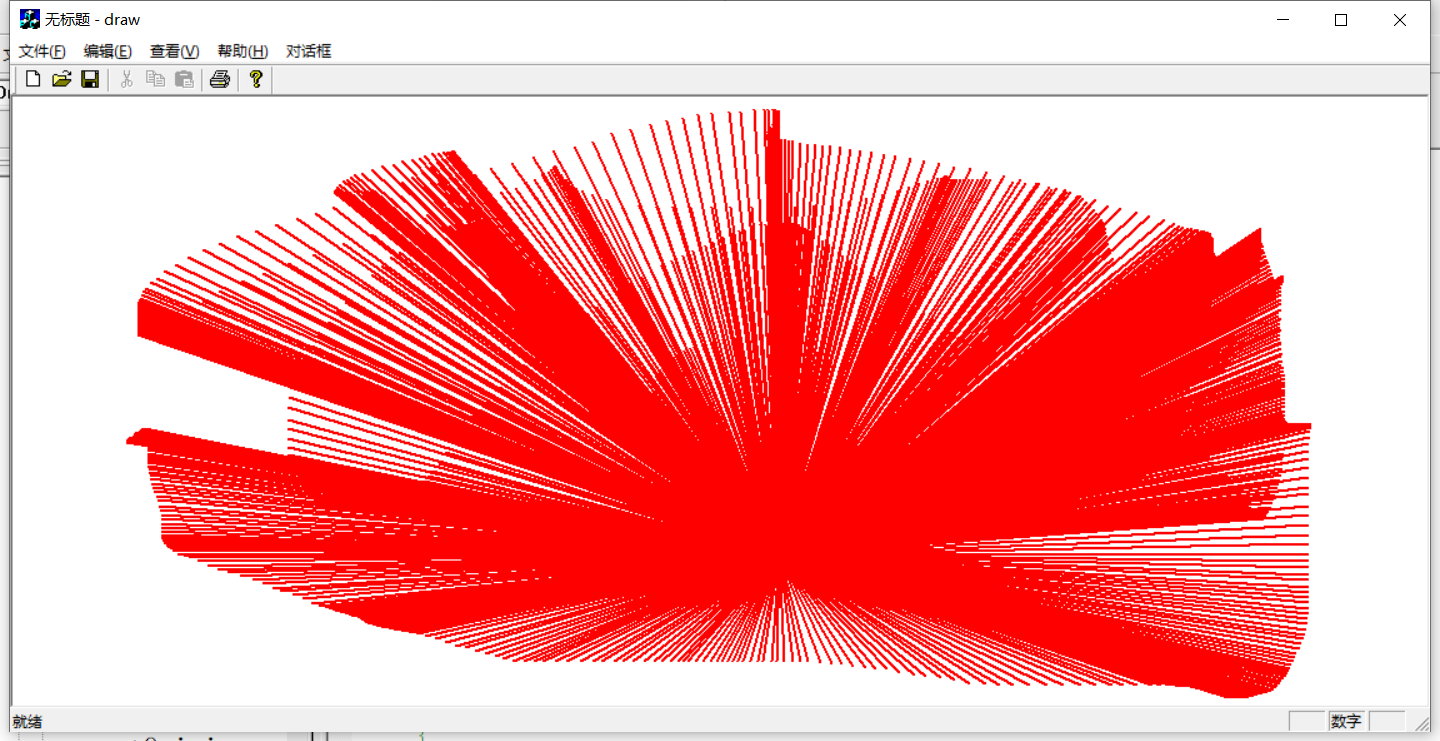
这个可以画连续曲线，画图如下：



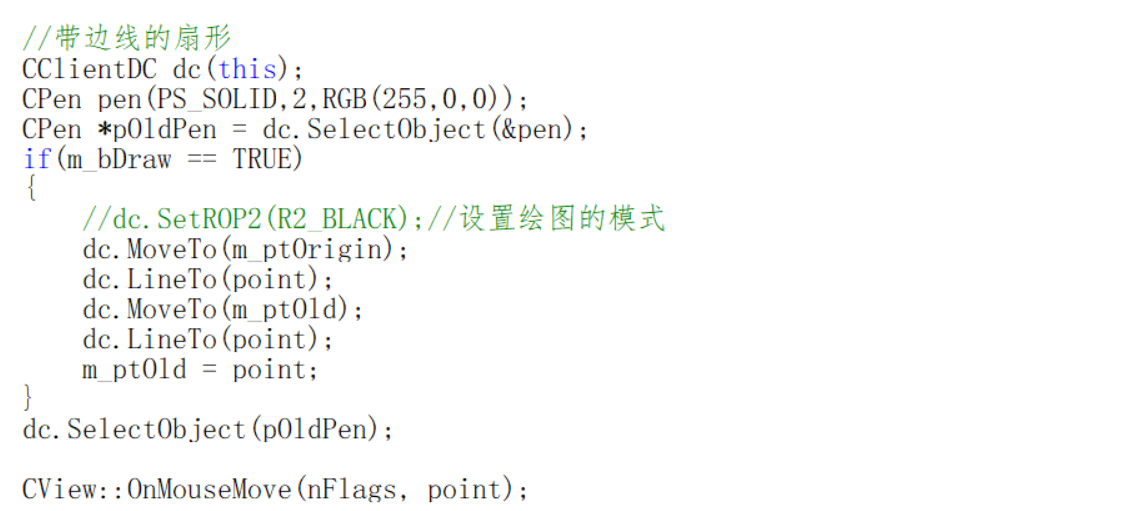
绘制不太边线的扇形



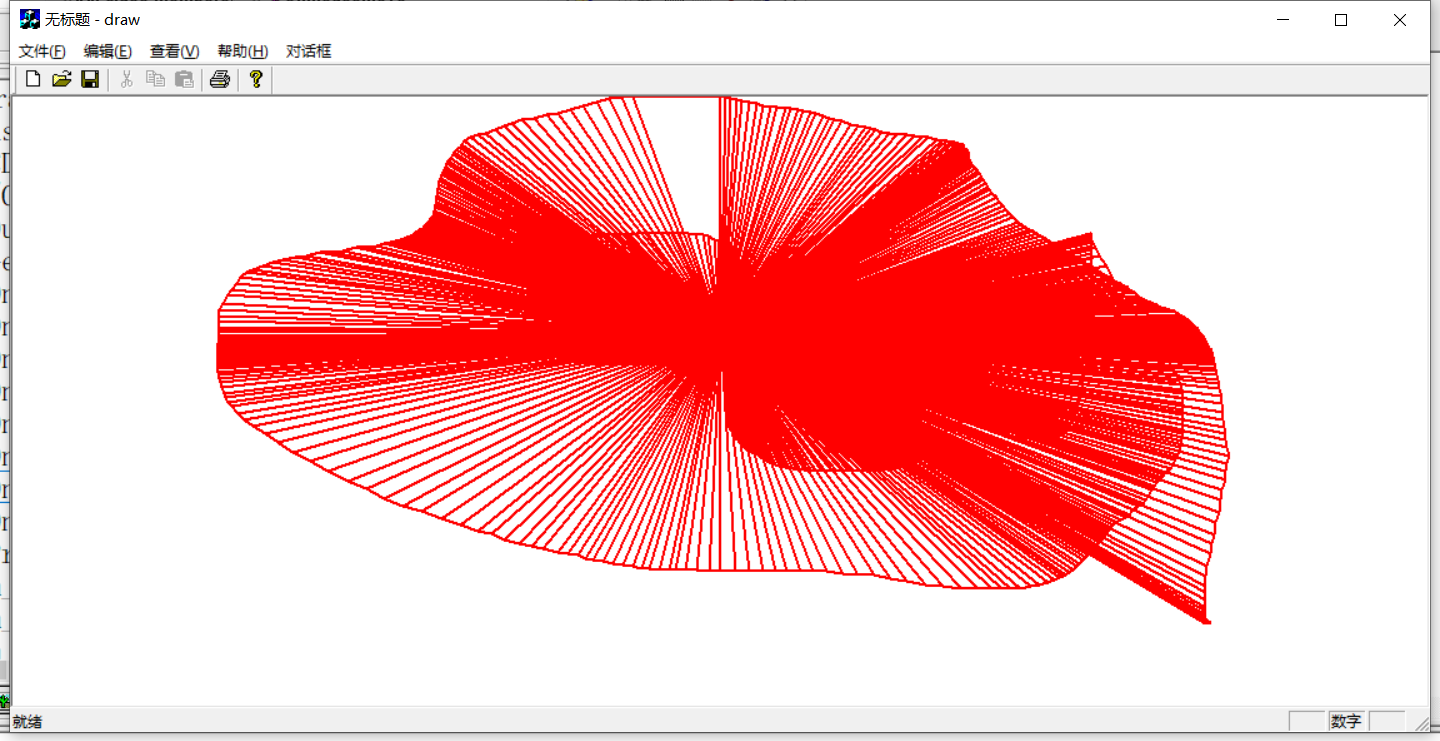
画图效果如下：



绘制带边线的扇形：



画图效果如下：



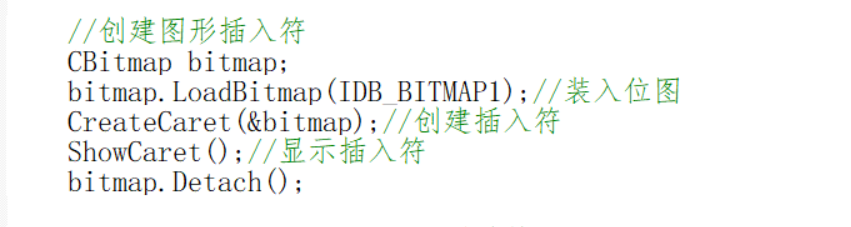
创建插入符



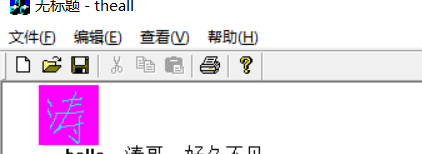
运行输出如下



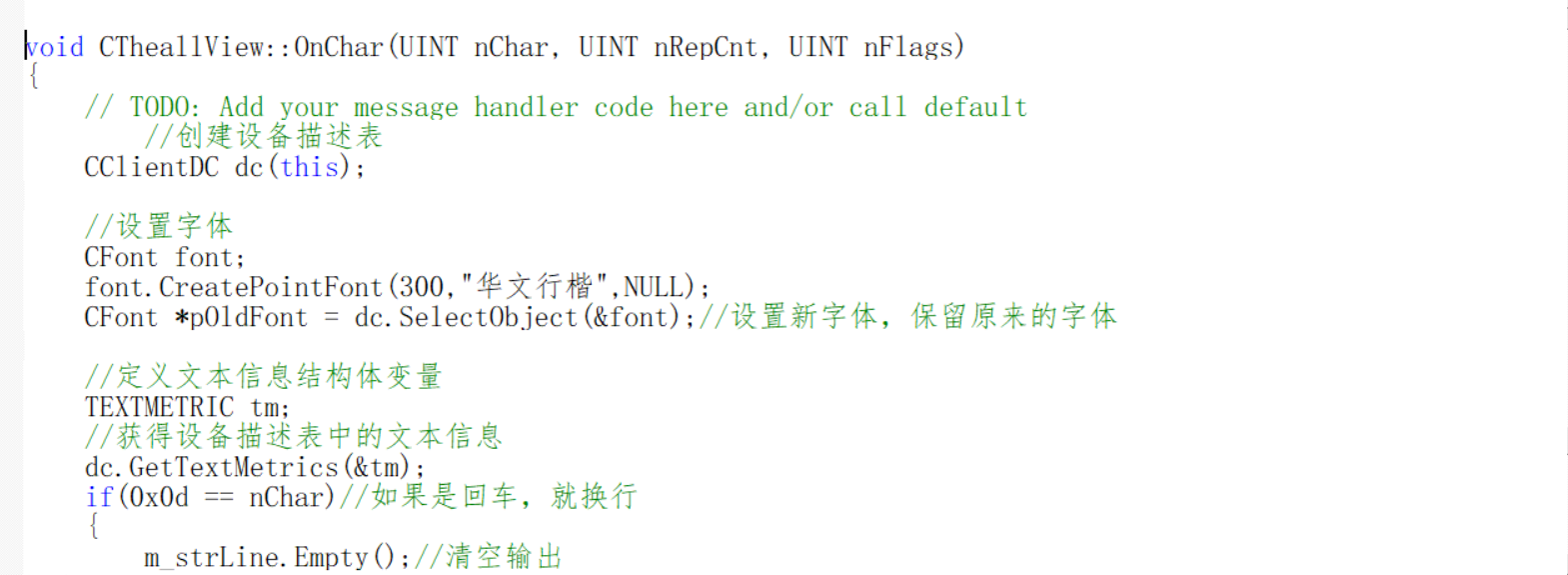
创建图形插入符



运行输出如下

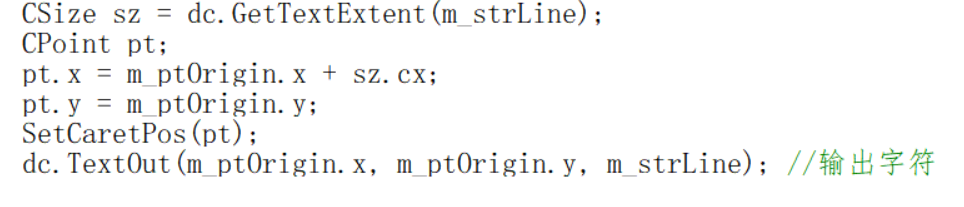


实现字符输入



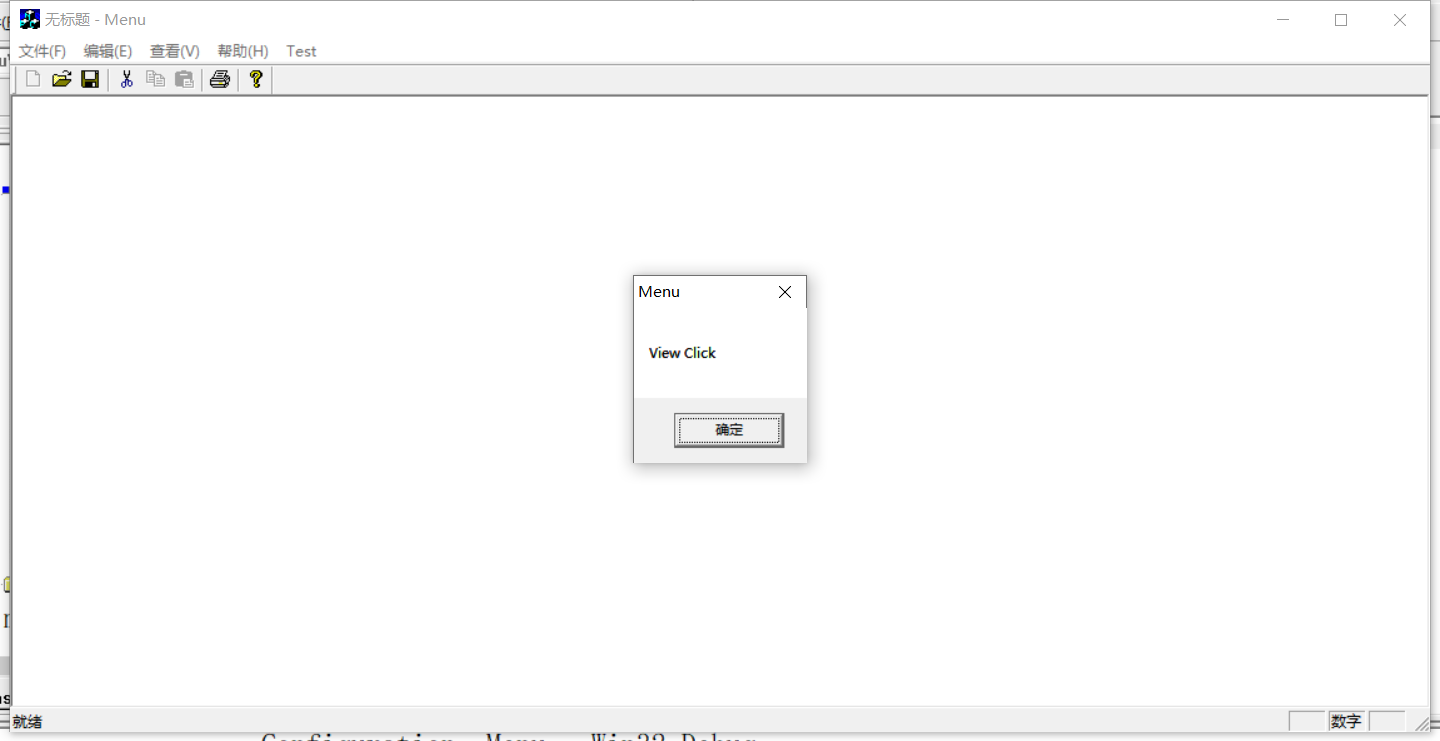
运行结果如下：

标识符位置跟随字符移动

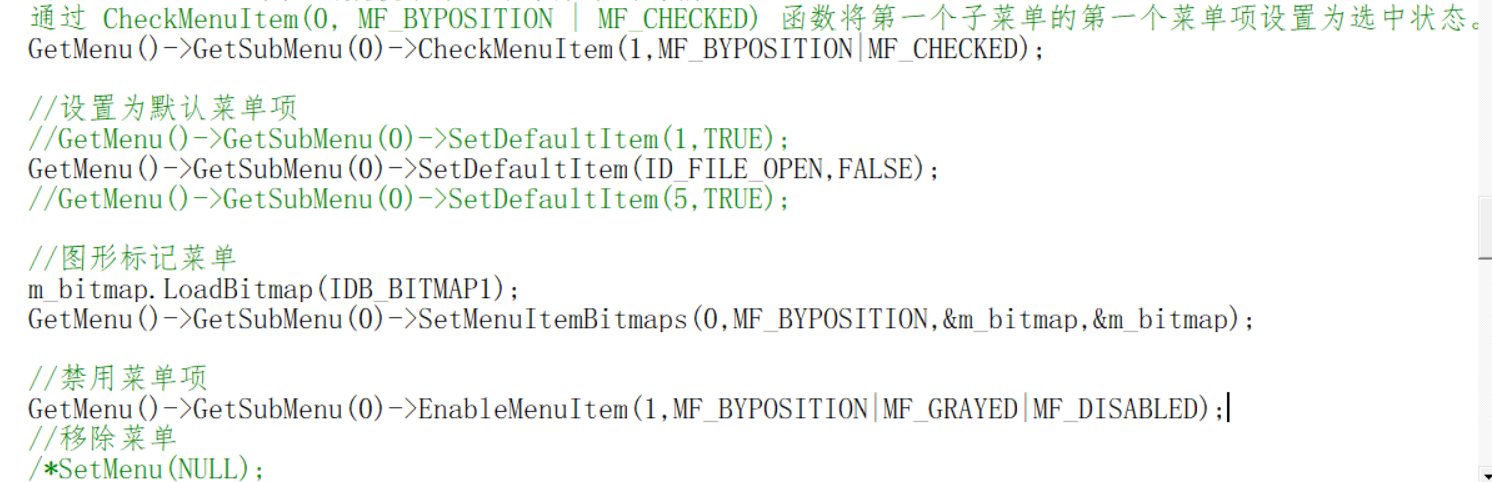
输出：

  
菜单编程

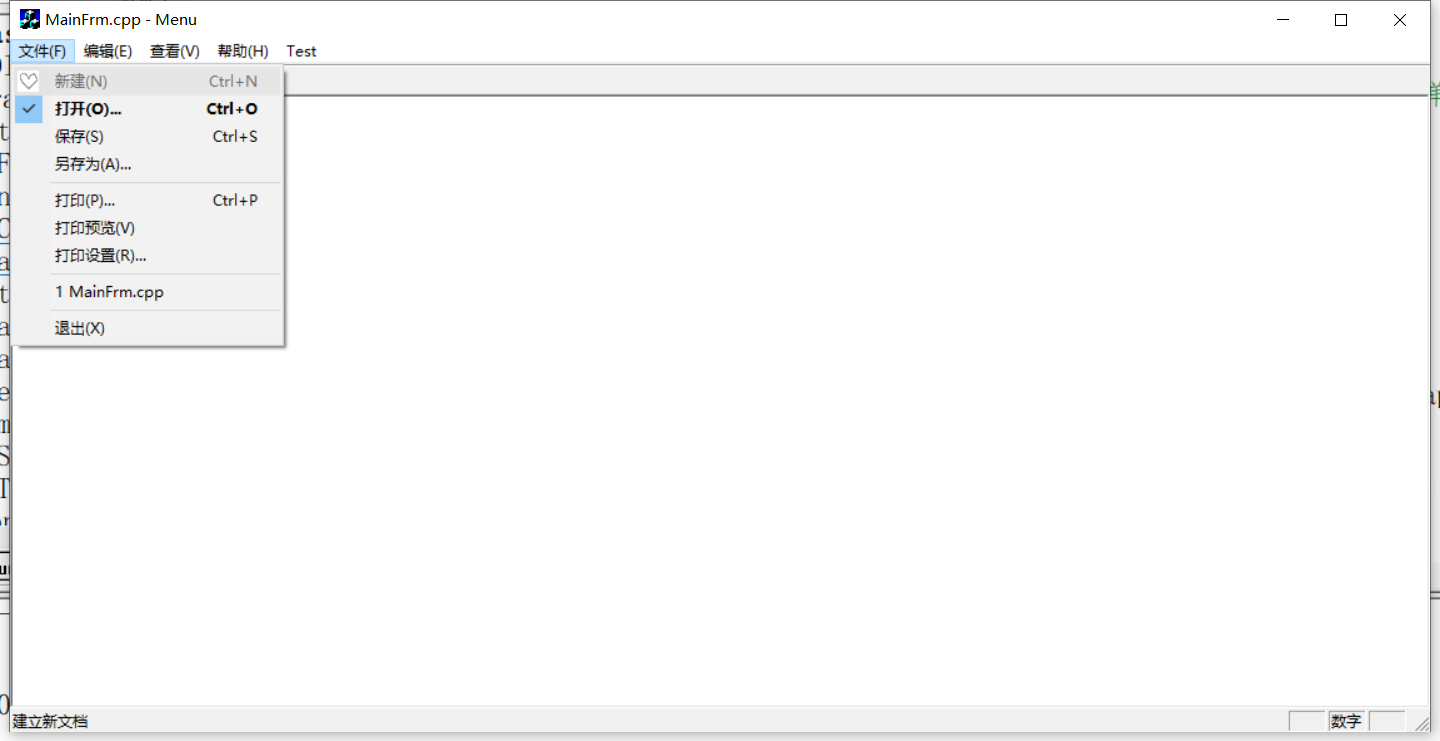
运行程序，点击Test,弹出一个信息框



基本菜单操作

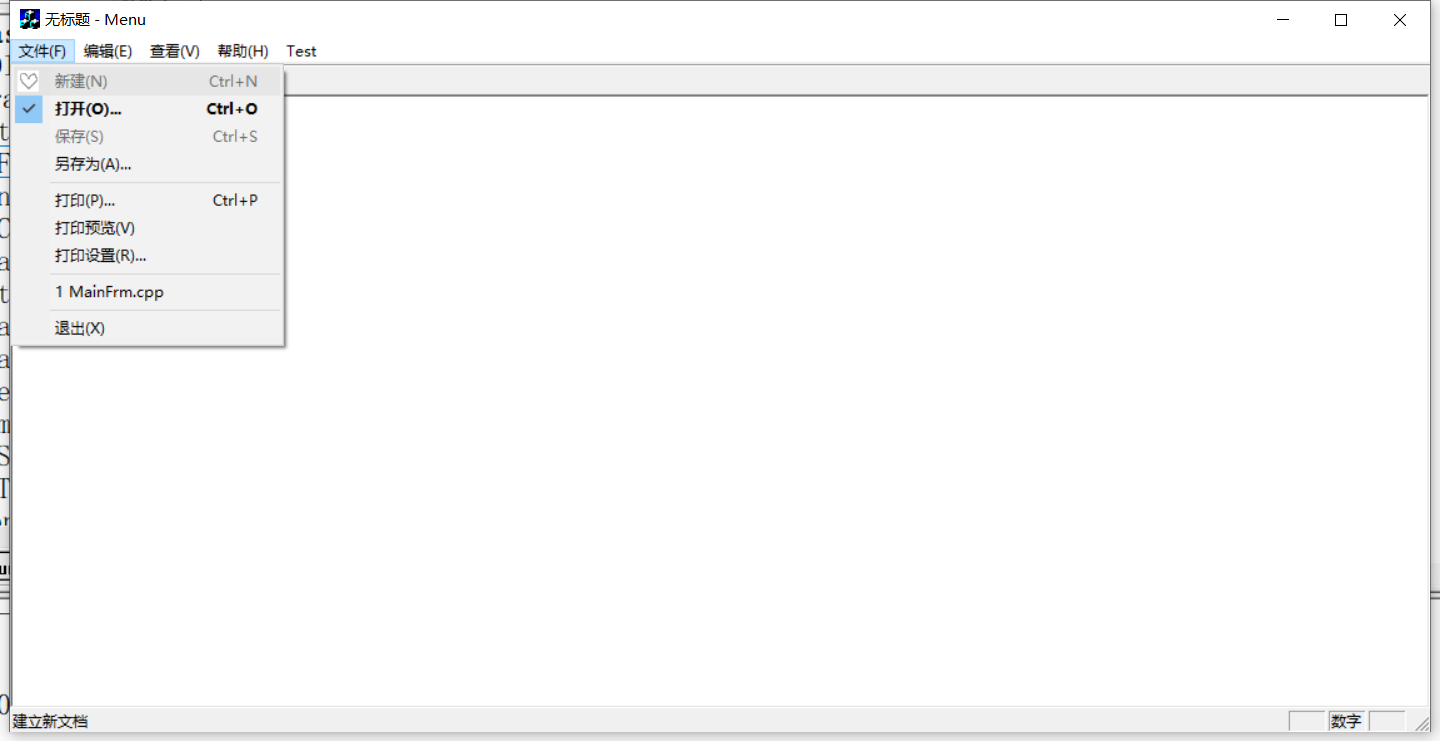


可以发现下面的效果图

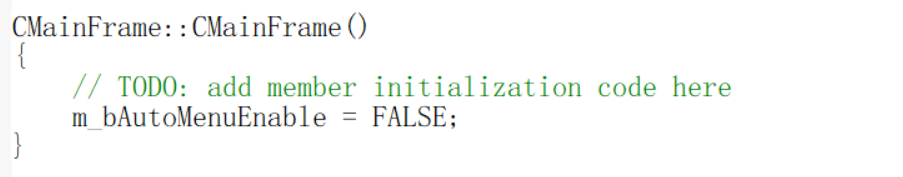


新建的那里多了我自己定义的图形

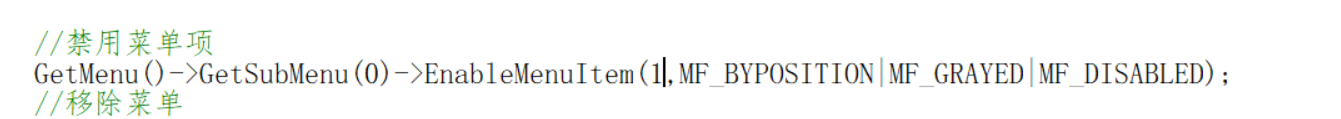
禁用菜单项



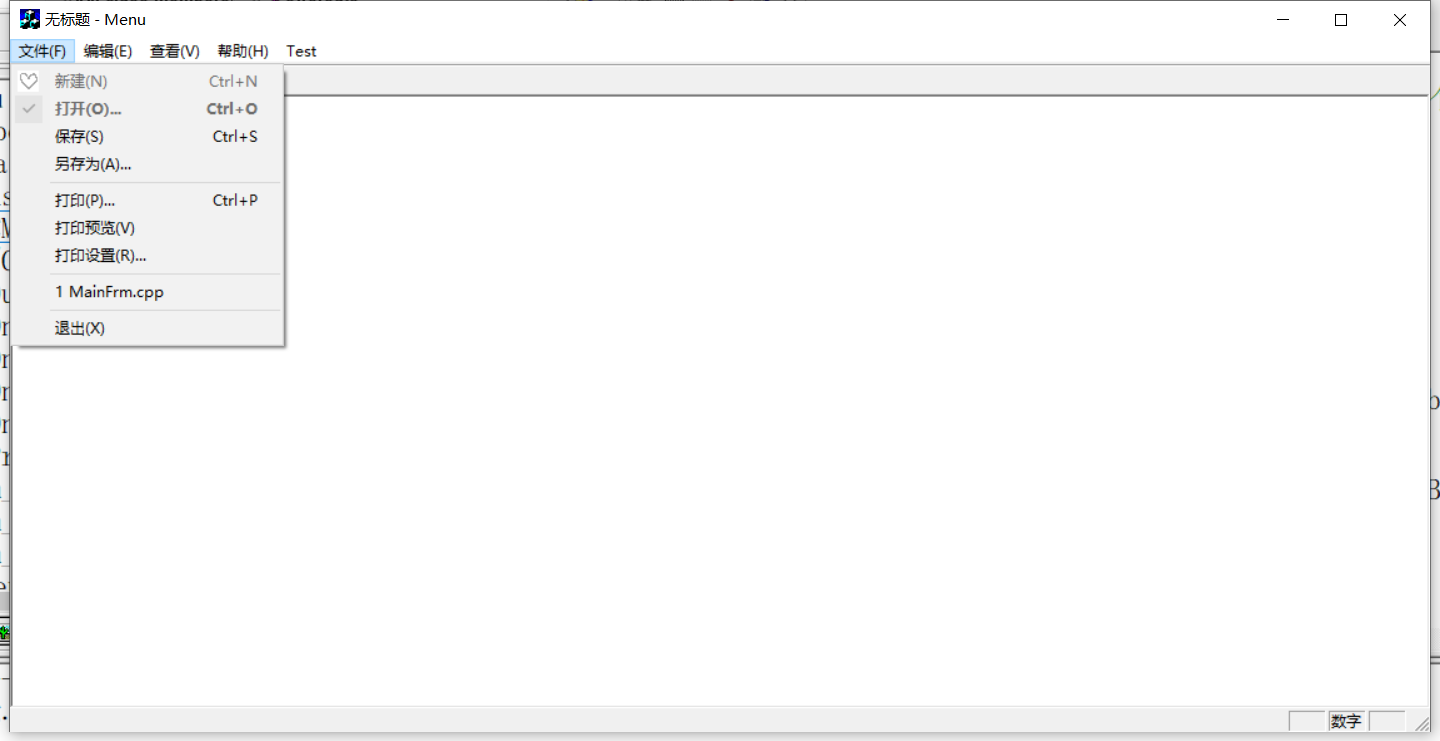
现在菜单项是可以使用的



自己操作菜单项需要这句代码



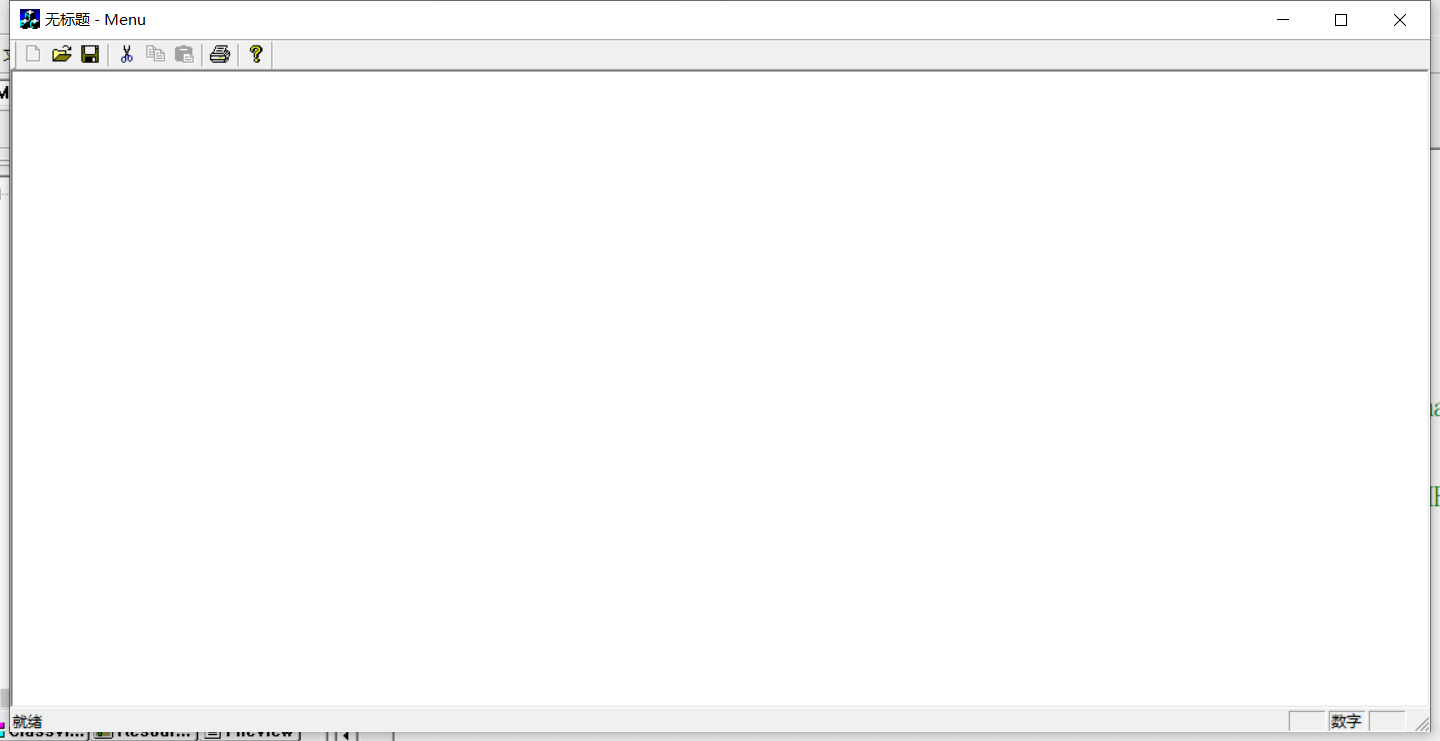
运行代码



发现打开的按钮灰了，表示禁用菜单栏

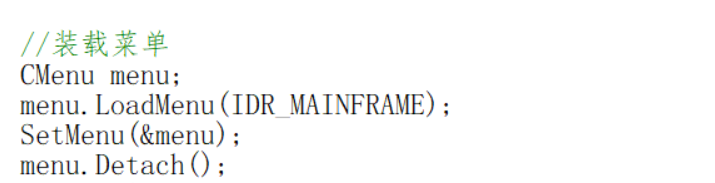
移除和装载菜单

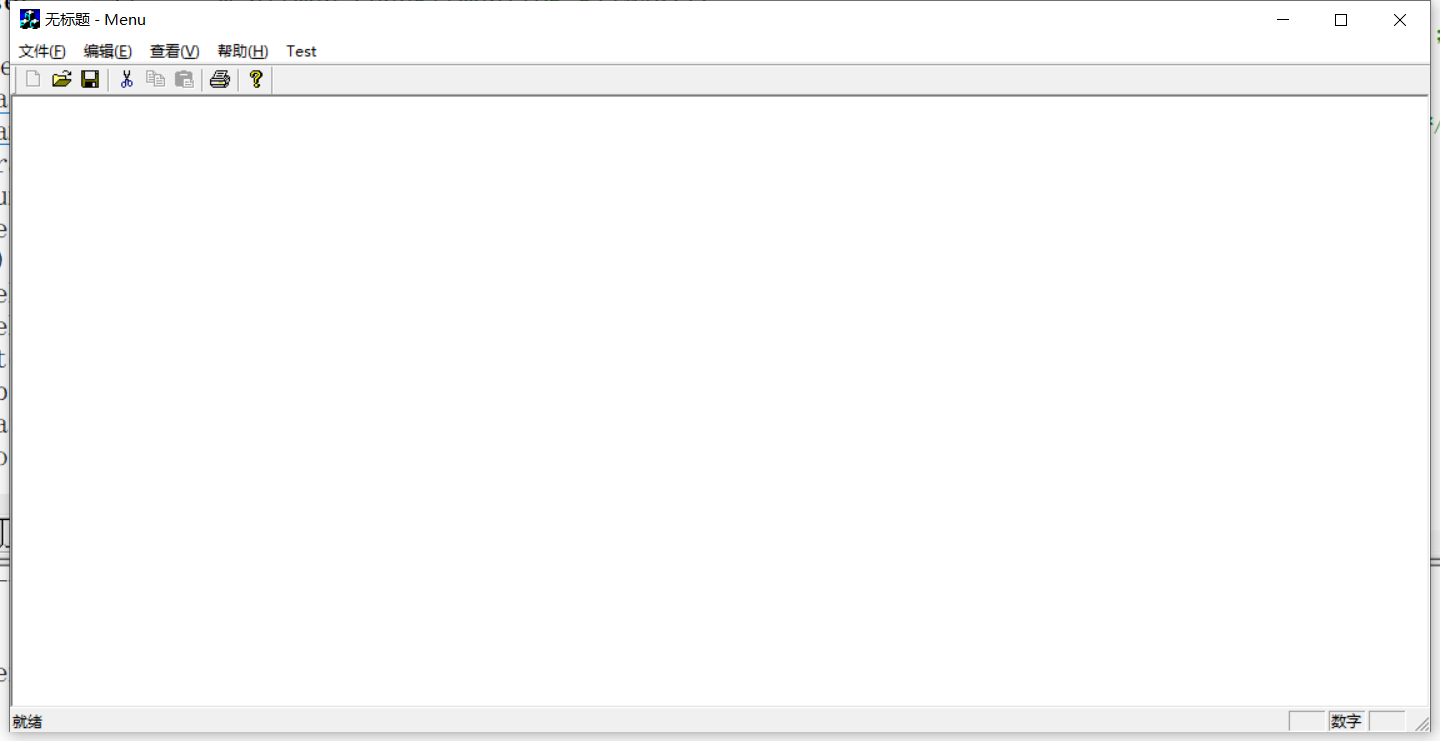




发现菜单项没有了。

再加以下代码

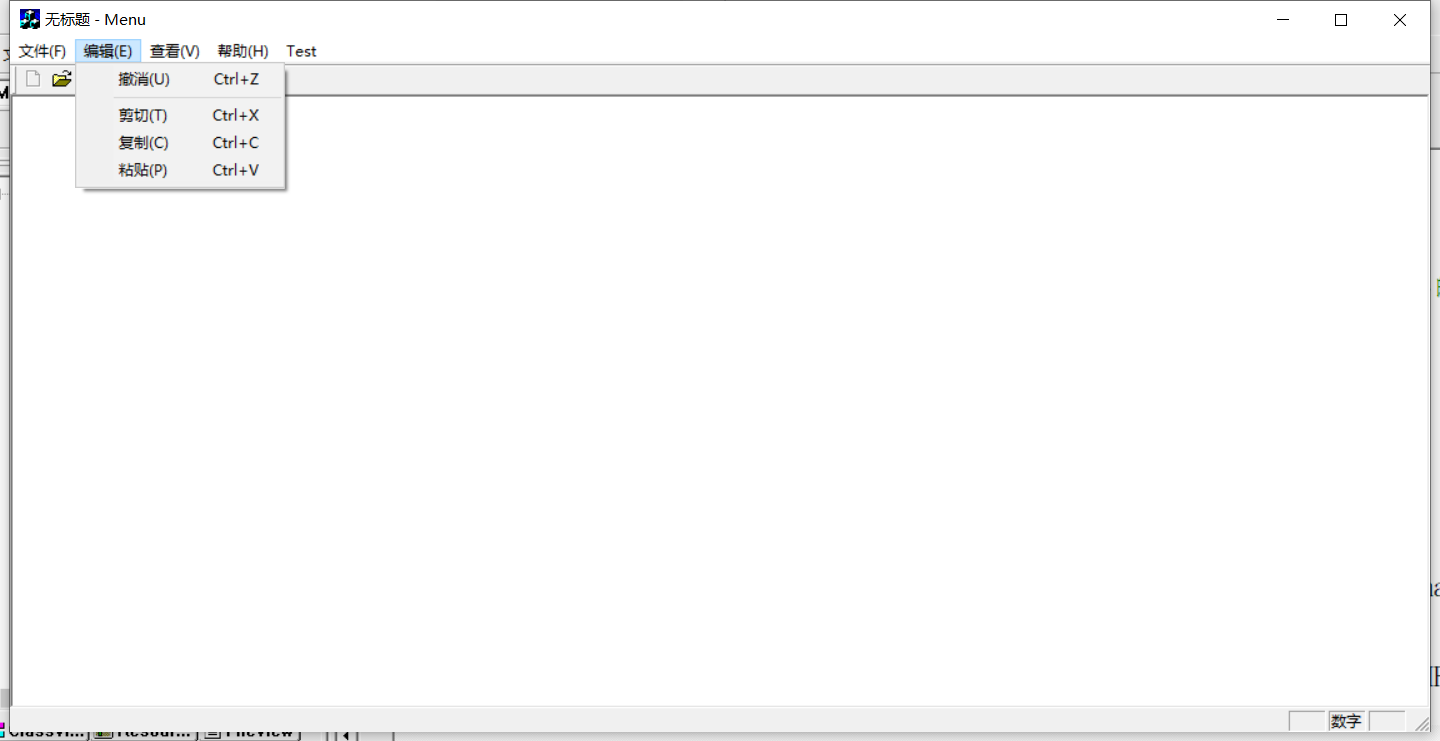




菜单项又回来了。

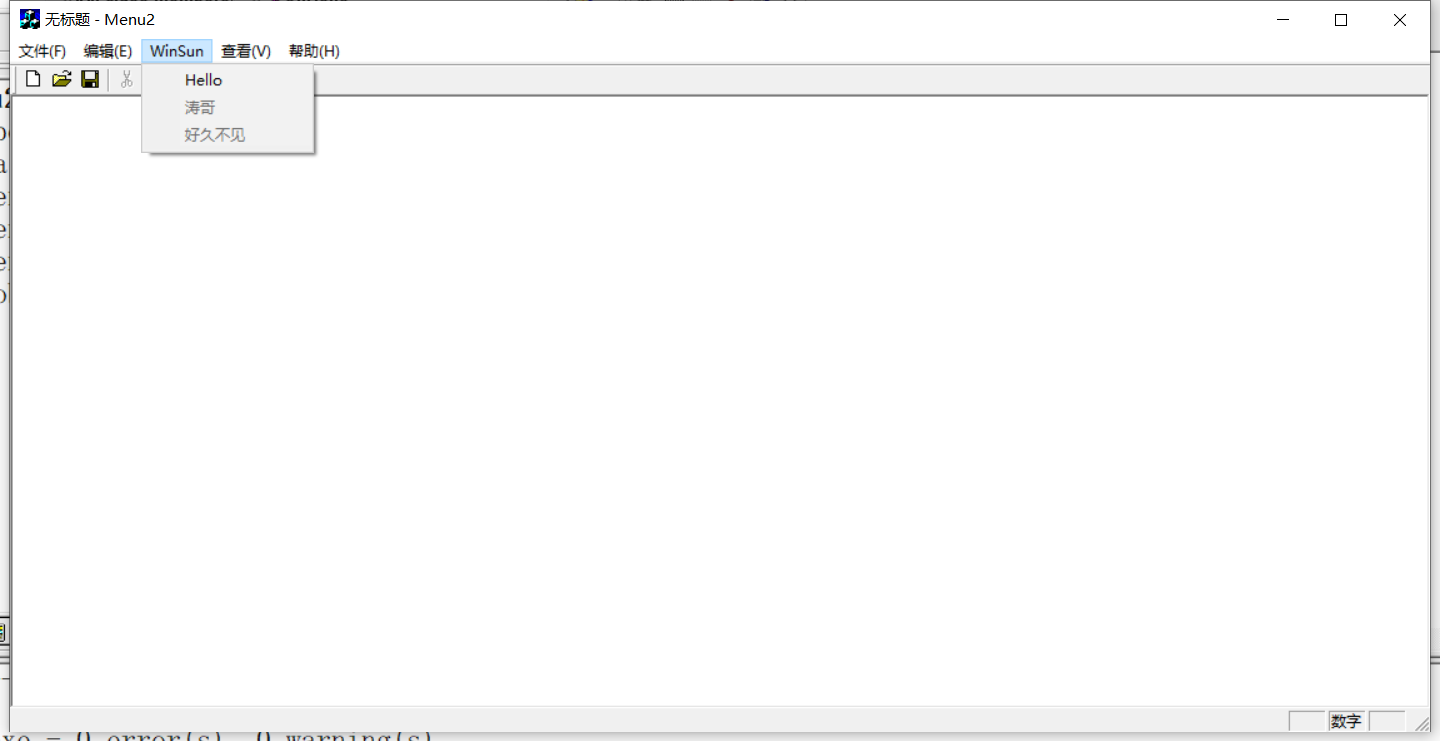
MFC菜单命令更新机制

使用MFC菜单命令更新机制让编辑下的剪切变得有效

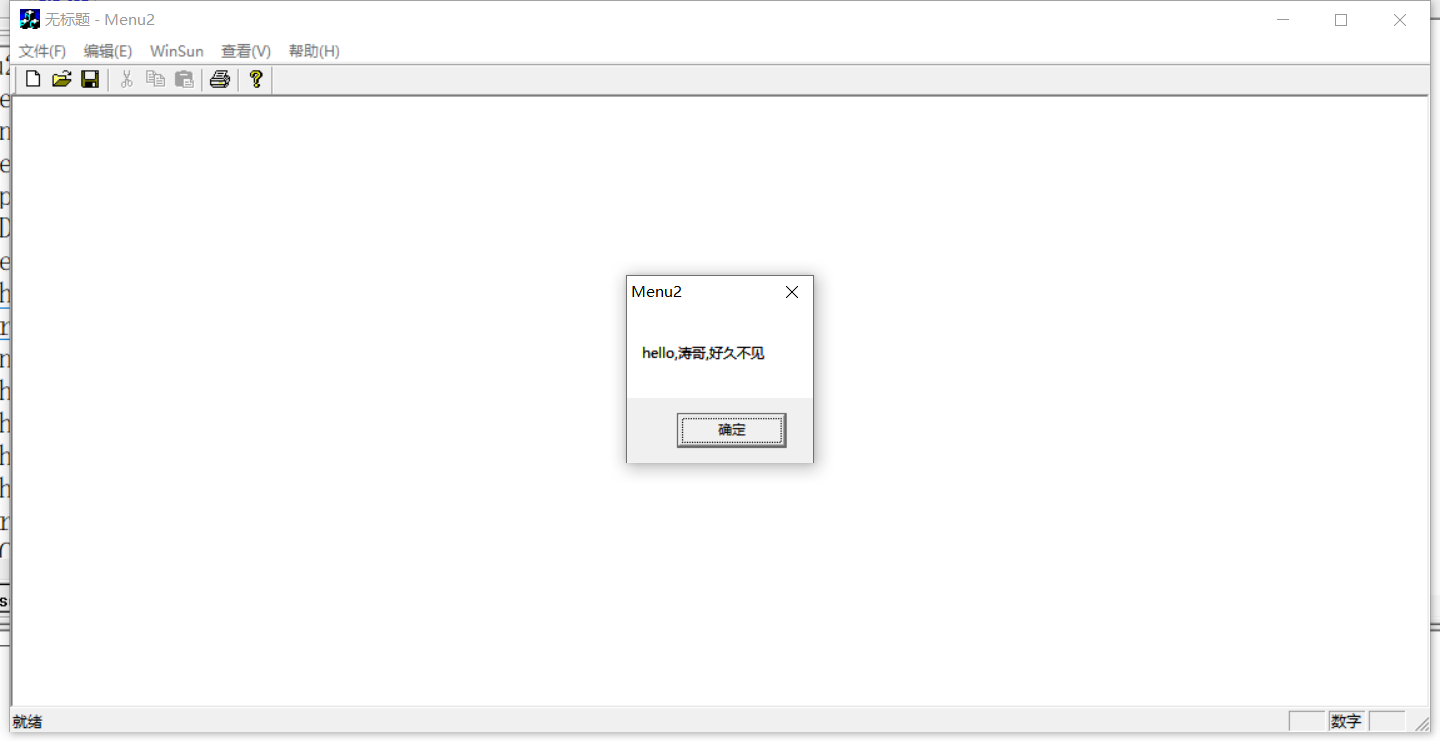


快捷菜单

我们平时在使用程序时，经常会用到单击鼠标右键显示快捷菜单（也称为上下文菜单或右键菜单）这一功能。现在我们尝试实现这个功能。

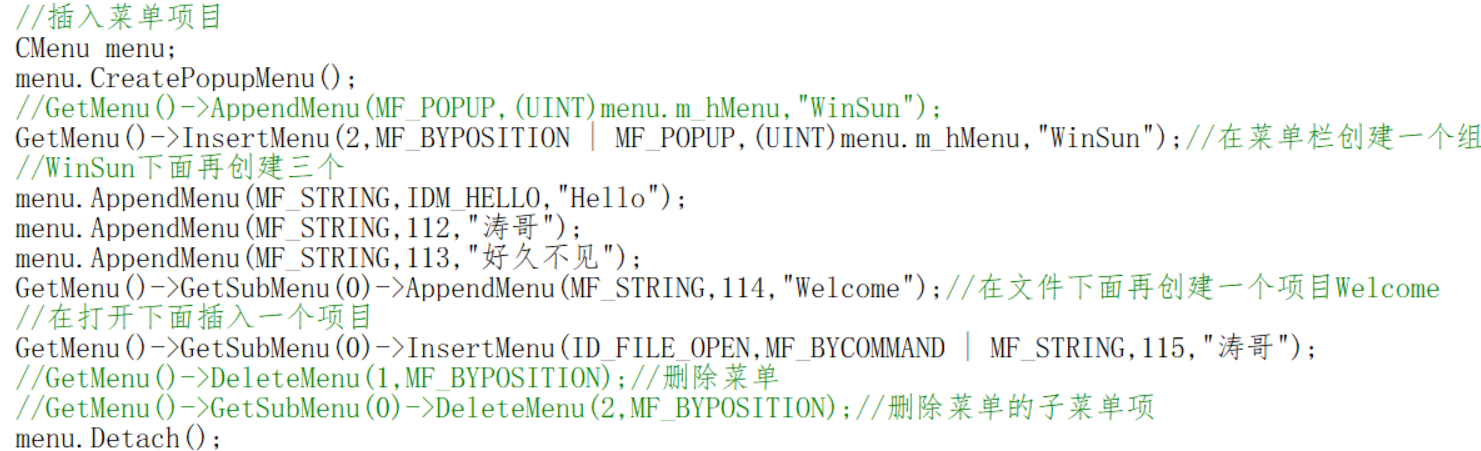


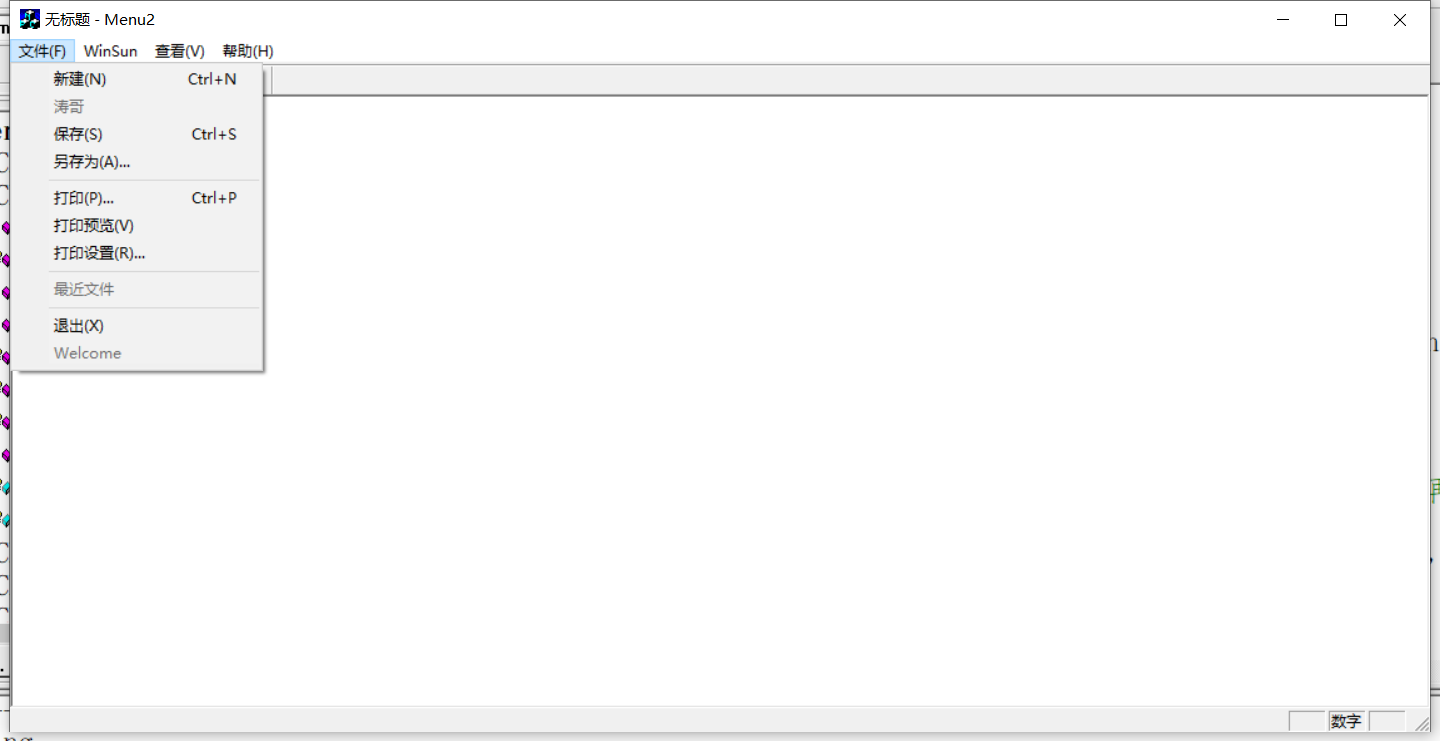
运行，右键打开快捷菜单，点击菜单项，显示如下



动态菜单操作

在主框架类的OnCreate里添加以下代码





在文件上添加了我新加的菜单。