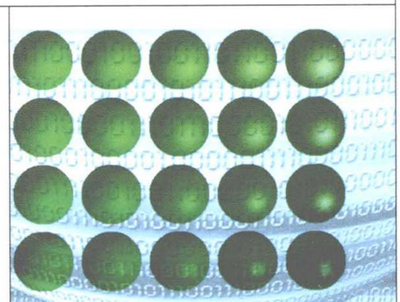
中国地质大学(武汉)实验教学系列教材

计算机图形学实习教程

JISUANJI TUXINGXUE SHIXI JIAOCHENG



郭际元
曾　文 ◎编著
龚君芳

中国地质大学出版社有限责任公司
ZHONGGUO DIZHI DAXUE CHUBANSHE YOUXIAN ZEREN GONGSI

中国地质大学(武汉)实验教学系列教材

计算机图形学实习教程

JISUANJI TUXINGXUE SHIXI JIAOCHENG

郭际元　曾文　龚君芳　编著

中国地质大学出版社有限责任公司
ZHONGGUO DIZHI DAXUE CHUBANSHE YOUXIAN ZEREN GONGSI

图书在版编目(CIP)数据

计算机图形学实习教程/郭际元,曾文,龚君芳编著.—武汉:中国地质大学出版社有限责任公司,2012.12

中国地质大学(武汉)实验教学系列教材

ISBN 978-7-5625-2889-0

Ⅰ.①计…

Ⅱ.①郭…②曾…③龚…

Ⅲ.①计算机图形学-高等学校-教材

Ⅳ.①TP391.41

中国版本图书馆 CIP 数据核字(2012)第 203102 号

计算机图形学实习教程	郭际元　曾文　龚君芳　编著
责任编辑:潘　娜	责任校对:张咏梅
出版发行:中国地质大学出版社有限责任公司(武汉市洪山区鲁磨路 388 号)	邮政编码:430074
电　话:(027)67883511　　传真:67883580	E-mail:cbb @ cug.edu.cn
经　销:全国新华书店	http://www.cugp.cug.edu.cn
开本:787 毫米×1 092 毫米 1/16	字数:323 千字　印张:12.5
版次:2012 年 12 月第 1 版	印次:2012 年 12 月第 1 次印刷
印刷:武汉教文印刷厂	印数:1—1 000 册
ISBN 978-7-5625-2889-0	定价:28.00 元

前　言

计算机图形学是研究利用计算机来处理图形的原理、方法和技术的学科，它的出现是计算机应用史上的一次重大变革。计算机图形学是目前应用最广泛、发展最迅速的计算机分支学科之一，也是学习图像处理、模式识别、CAD、CAM、CAI、计算机视觉、多媒体技术等各门课程的技术基础。

我国对计算机图形学和CAD的研究始于20世纪60年代中后期，进入80年代后已取得可喜的成果。硬件方面，我国研制并批量生产出多种高分辨率显示器、图形图像处理卡、图形终端和图形交互设备；软件方面，国家攻关项目、863计划和国家自然科学基金项目中有不少图形软件研究课题，其中二维交互绘图系统已进入商品化阶段，足以与国际市场上的著名软件媲美，中国地质大学（武汉）信息工程学院研制的彩色地图编辑出版系统MAPCAD就是一个典型的例子。所有这些，都预示着计算机图形学和CAD技术在我国有着更加广阔的发展前景。

根据我们多年的教学经验看，计算机图形学是一门实践性很强的课程，如果希望学生通过几十个学时的学习，较好地掌握计算机图形学的基本理论并有一定的实际编程能力，就需要有一本具有系统性和实用性的实习教材，用它来配合高校计算机图形学教材的使用。本教材正是出于这个目的而编写。

本实习教材分八章编写。主要内容分别是：第一章，实习导引；第二章，基本图形的生成；第三章，区域填充；第四章，图形变换；第五章，图形裁剪；第六章，曲线；第七章，交互技术；第八章，消隐；书末有两个附录，分别介绍C语言环境下画像素的方法和OpenGL环境下画图的方法。第一章的内容包括：实习要求与实习报告、VC＋＋环境下实现图形显示、VC＋＋环境下画像素的方法和VC＋＋中基本绘图函数。其他各章在介绍计算机图形学常用算法思想的基础上，较为详细地阐述了各种算法的实现步骤和方法，给出实现提示，提出实习要求和提高要求，并针对实现过程中常遇到的问题，给出思考题。每一章还针对典型算法给出实现实例。实例中既有操作过程，又有完整的程序代码，可以使学生通过该实例的学习，举一反三地实现其他算法。

本实习教材的特点：一是具有适当的理论性和系统性，在配合理论教材学习的基础上，帮助学生更好地掌握计算机图形学理论和方法；二是针对计算机图形学课程实践性强的特点，在内容组织上，尽可能让学生容易上手，较为顺利地完成实习内容，从而提高学生编写计算机图形软件的能力；三是通过附录简要介绍了C语言绘图方法和OpenGL环境中实现画图的方法，能给在其他环境实现画图的人员以参考。

本书中，郭际元负责第六、第七、第八章及附录一的撰写，曾文负责第二、第三、第四章的撰写，龚君芳负责第一、第五章及附录二的撰写。本书的撰写还要感谢敖卿鑫、潘永婷、柯尊林、李雄、宇政专等同学，他们分别参与了部分计算机图形学实例的编程实现工作。

本书是我们多年来科研和教学经验的总结。由于水平有限，错误和不当之处在所难免，敬请读者批评指正。

编　者

2012年3月

目　录

第一章 实习导引

1.1 实习教程简介

计算机图形学是一门实践性很强的学科，如果仅仅是通过课上介绍原理和算法，学生很难真正学好这门课程，在今后的实践过程中也很难学以致用。

多年教学经验告诉我们，要想使学生通过几十个学时的学习较好地掌握计算机图形学的基本理论，并具备一定的实际编程能力，就需要一本具有系统性和实用性的实习教材来配合高校计算机图形的教学。本教材正是出于这个目的而编写。

本书在第一部分共安排了8个实习，实质体现了计算机图形学的几个研究方面。每个实习都针对其研究方面介绍了一些常用算法，并对读者提出了实习要求。读者可在本教材的提示下完成实习内容，我们也提倡读者在本教材的基础上，提出自己的算法思想以及实现方法。

本书的第二部分是附录部分，该部分的内容包括C语言环境下画像素的方法以及Open GL环境下画图的方法。附录部分是对理论教材的一些补充，以帮助学生提高实际动手能力。

1.2 实习要求

本实习教程一共有8个实习(其中包括实习导引)，每个实习有2～5个小题。教学过程中，可根据不同层次、不同专业学生的实际情况选做4～5个实习，每个实习又可只选做其中的1～2个题目的基本要求或提高要求。实习过程中具体要求如下。

(1)问题分析。充分地分析和理解问题本身，弄清要求做什么，用什么算法。

(2)程序设计。根据所采用的算法，设计数据结构，画出流程图并编程。最后准备调试程序的数据及测试方案。

(3)上机调试。对程序进行编译，纠正程序中可能出现的语法错误。调试前，先运行一遍程序看看究竟将会发生什么。如果情况很糟，根据事先设计的测试方案并结合现场情况进行错误跟踪，包括打印执行路径、输出中间变量值等手段。

(4)整理实习报告。实习报告内容包括如下几方面。

1)问题描述：包括目标、任务、条件约束描述等；

2)设计：数据结构设计和核心算法设计，主要功能模块的输入，处理(算法框架)和输出；

3)测试范例：显示测试结果(包括图和数据)，对测试结果进行分析讨论，测试过程中遇到的主要问题及所采用的解决措施；

4)心得：包括程序的改进设想、经验和体会；

5)程序清单:应包括详细的注释。

1.3　实习环境介绍

1.3.1　VC++环境下实现图形显示

【目标】

1. 了解如何用 MFC 函数来绘图;
2. 熟悉 MFC 编程基本知识,包括消息映射,鼠标、菜单的使用;
3. 熟悉类的定义、构造、编写;
4. 熟悉数据的保存。

【内容】

1. 用鼠标在窗口上绘制一组直线段;
2. 自己实现直线类 Line,用来保存每条直线段的起点、终点,及自身的绘制;
3. 实现菜单上的 Undo 功能;
4. 完成所绘图的保存。

【实现步骤】

一、建立绘图程序框架

在 VC++环境下开发绘图程序,实现图形显示的第一步是使用 MFC Application Wizard 来建立绘图程序的基本框架,具体步骤如下:

(1)从"File"菜单选择"New"菜单项,再选择"Project",弹出"New Project"对话框,如图 1.1 所示;

(2)在图 1.1 所示对话框中,左侧"Project types"选项中选择"MFC",右侧"Templates"选项中选择"MFC Application",在"Name"编辑框中输入绘图程序的名字,这里设定为 Graphics Exp,其他采用默认值;

(3)单击"OK"按钮,弹出如图 1.2 所示"MFC Application Wizard"对话框,缺省情况下是多文档类型;

(4)在图 1.2 所示对话框中单击左侧的"Application Type",得到图 1.3 所示对话框,选择右侧的"Single document"单选钮,表示要生成一个单文档(SDI)绘图程序,单击"Finish"按钮,表示使用随后的各项默认设置,MFC Application Wizard 自动生成绘图程序的各项源文件。

Application Wizard 自动生成了完整的应用程序的基本框架,你可以立即单击"Build"菜单的"Build Graphic Exp"菜单项,进行编译和链接,然后运行该应用程序,你可以看到该程序是一个标准的 Win 32 应用程序,它包含有标题、菜单栏、工具栏和状态栏,之后可以操作许多命令。当然,因为你还没有给这个程序添加任何自己的代码,所以它还不能作出任何有实际意义的操作。

二、在屏幕上画图

为了能用鼠标在屏幕上作图(画线),当然必须要控制鼠标。控制鼠标的方法就是对鼠标消息进行映射,在鼠标消息的处理函数中加上画线的代码,这是实现用鼠标在屏幕上画图的基

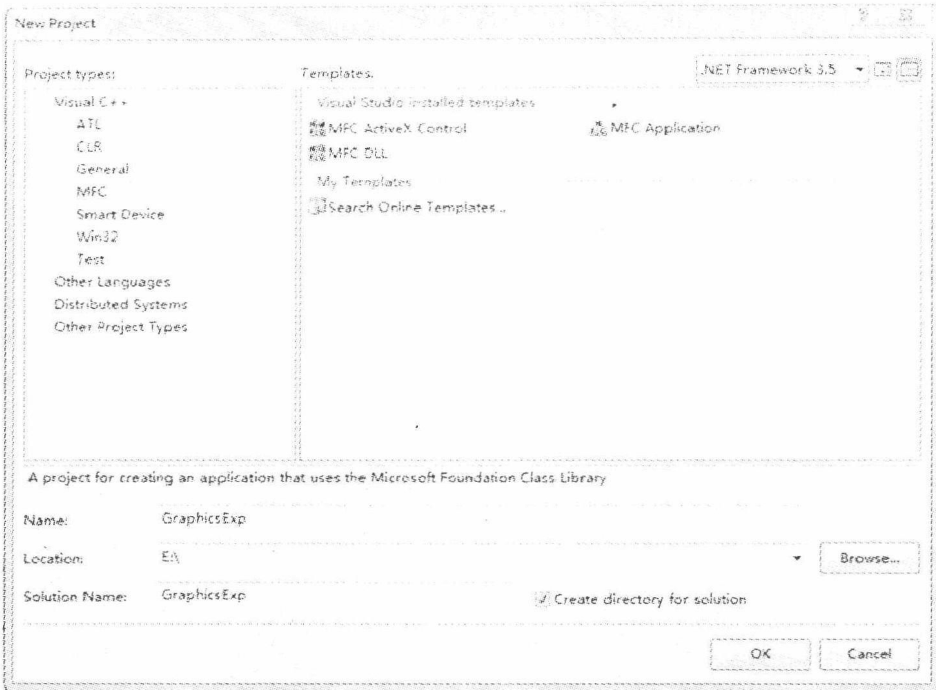


图 1.1　New Project 对话框

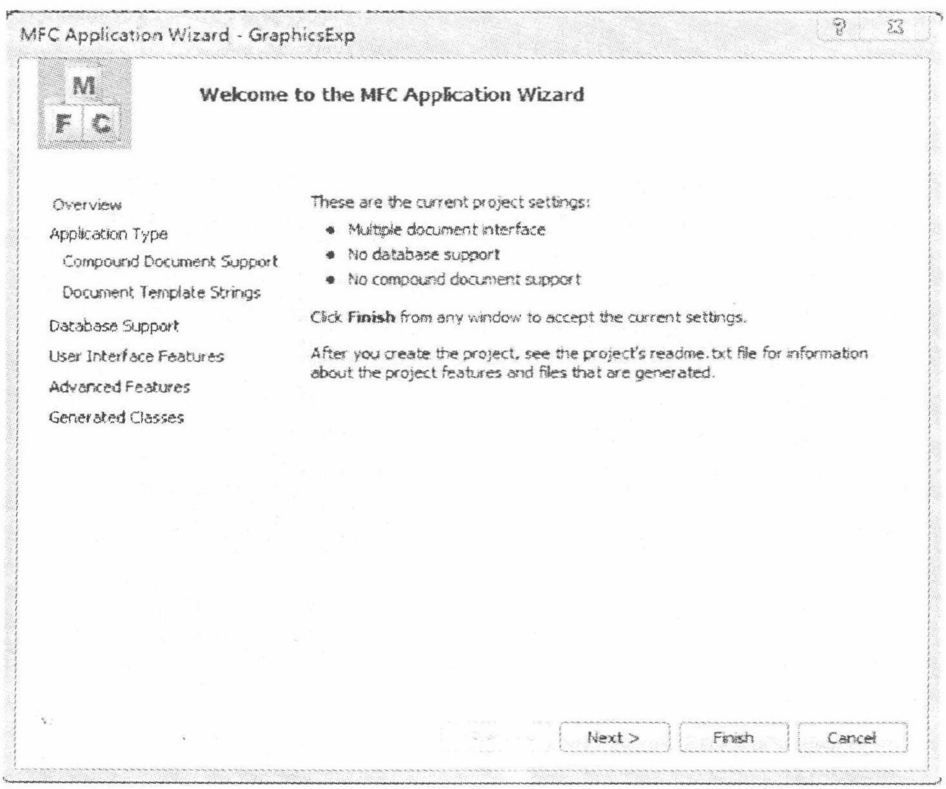


图 1.2　MFC Application Wizard 对话框

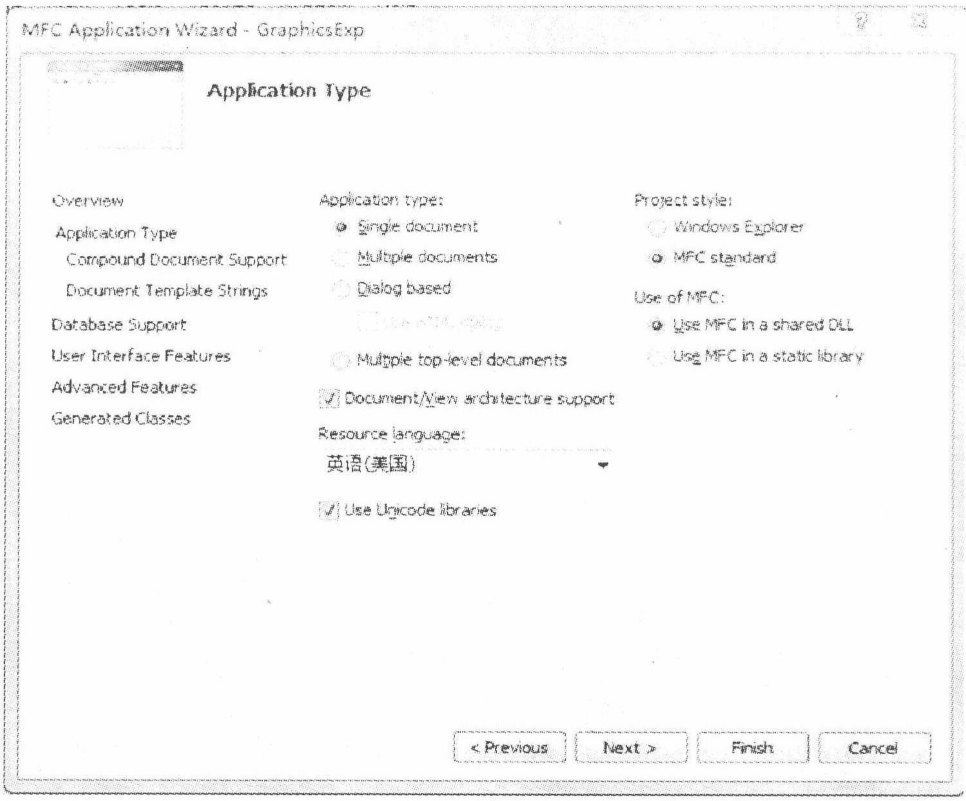


图 1.3　Application Type

本思路。由于在屏幕上(空白区)画图是与视图有关的,所以以下的具体步骤都是在视图类中进行的。

1. 映射鼠标消息

在 Visual Studio 2008 中没有 ClassWizard,则映射鼠标消息的方法为:

(1)单击工具栏中的"Properties Window"按钮,在屏幕上会出现"Properties"窗口,如图 1.4 右侧所示;

(2)在"Properties"窗口区单击"Messages"按钮,"Properties"窗口将如图 1.5 所示,单击鼠标左按钮找到 WM_LBUTTONDOWN,单击其右侧的按钮,然后单击"<Add>On Lbutton Down",按下鼠标左键,消息映射处理函数成功添加到了 CGraphicsExpView 类中,如图 1.6 所示;

(3)同样地,在图 1.5 中移动鼠标到新的位置添加 WM_MOUSEMOVE 和释放鼠标左按钮 WM_LBUTTONUP。

2. 在视图类中添加数据成员

接着要向视图类中添加数据成员以便存放鼠标的状态、位置和光标形状。为此,打开文件 GraphicsExpView. h,将以下语句添加到 CGraphicsExpView 类定义中:

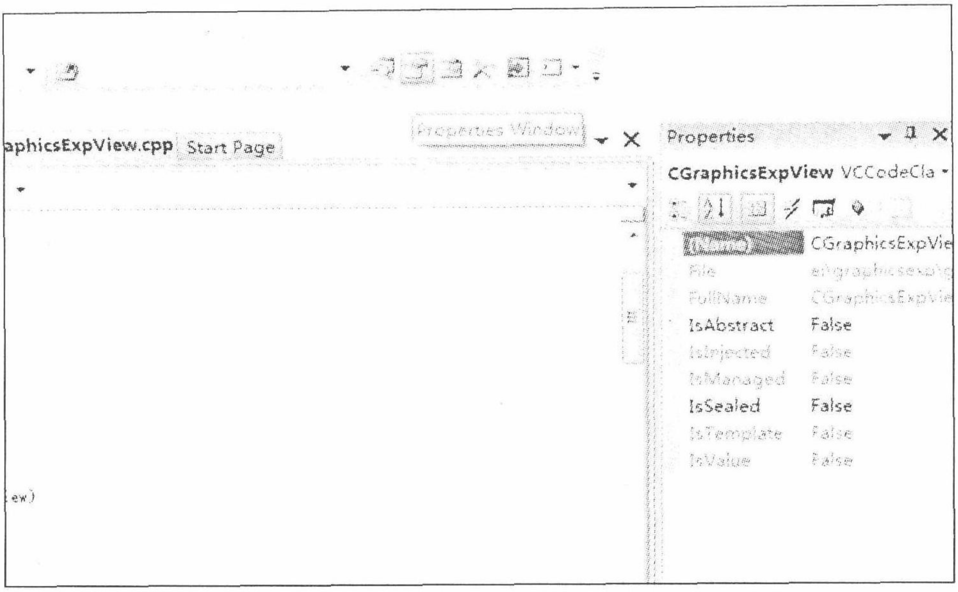


图 1.4　Properties 窗口

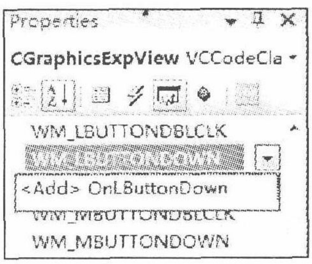


图 1.5　Properties 窗口－Messages

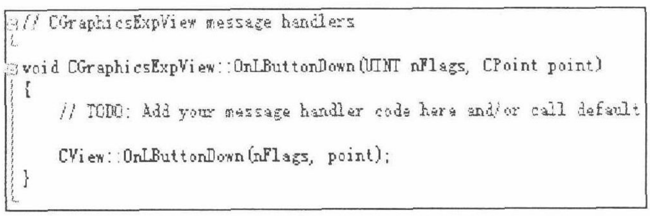


图 1.6　消息映射函数

```
protected:
    int m_Drag;
    CPoint m_pntPrev;
    CPoint m_pntOrigin;
```

接着，打开文件 GraphicsExpView. cpp，将以下代码添加到 CGraphicsExpView 类的构造函数中，以便初始化数据成员 m_Drag：

```
    m_Drag=0;
```

3. 添加消息处理函数代码

映射鼠标消息后，接着添加自己的消息处理代码，以便能在视图窗口中画直线。

(1)在消息处理函数 CGraphicsExpView::OnLButtonDown 的基本定义中添加以下代码：

```
m_pntPrev=point;          //保存鼠标光标的当前位置
m_pntOrigin=point;        //保存画线的起始位置。
SetCapture();             //使随后的鼠标消息都被送往视图窗口
```

```
m_Drag=1;                    //标志鼠标为拖动状态
RECT rect;
GetClientRect(&rect);        //获取窗口客户区的坐标
ClientToScreen(&rect);       //将窗口客户区坐标转换为屏幕坐标
ClipCursor(&rect);           //将光标限定在窗口客户区内
```

(2)在消息处理函数 CGraphicsExpView::OnMouseMove 中添加以下代码：

```
HCURSOR m_hCursor=LoadCursor(NULL,IDC_CROSS);
SetCursor(m_hCursor);        //设置十字形光标
CClientDC dc(this);          //创建设备上下文环境
if (m_Drag)
{  dc.SetROP2(R2_NOT);  //设置绘图模式为反转模式，以实现橡皮筋技术设置
   dc.MoveTo(m_pntOrigin);
   dc.LineTo(m_pntPrev);
   dc.MoveTo(m_pntOrigin);
   dc.LineTo(point);
   m_pntPrev=point;
}
```

(3)在消息处理函数 CGraphicsExpView::OnLButtonUp 中添加以下代码：

```
if (m_Drag)
{  m_Drag =0;                //标志鼠标为非拖动状态
   ReleaseCapture();         //释放对鼠标的使用并恢复正常输入
   ClipCursor(NULL);         //让鼠标可以在屏幕上任意移动
  }
```

4. 定制视图窗口

在缺省情况下，创建视图窗口时视图窗口的背景颜色为当前窗口的背景颜色，光标形状为箭头光标。如果在视图窗口内移动鼠标，则十字形光标和箭头光标将交替闪烁显示；如果当前窗口背景为黑色，则所画的直线几乎看不见。因此，有必要修改窗口特性，方法为在 CGraphicsExpView 的成员函数 PreCreateWindow 中添加以下代码：

```
    cs.lpszClass=AfxRegisterWndClass(CS_HREDRAW| CS_VREDRAW,0,
(HBRUSH)::GetStockObject(WHITE_BRUSH),0);
```

5. 建立并运行程序，用鼠标在其中画各种直线，如图 1.7 所示。

三、定义直线类并在文档类中保存直线

上述程序运行后可以在用户区绘制直线了，但当移动窗口或改变窗口大小时，屏幕上的图形立即不见了。之所以会出现这种问题，主要是由于当窗口改变时，应用程序首先清除屏幕，然后调用视图类的 OnDraw 成员函数重新绘制窗口。但前面的步骤中，我们还没有保存所画直线的代码。为此，必须在文档类中添加能存放直线的文档数据。对于用户在视图窗口中所画的直线，可以在文档中添加数据成员来存放直线的坐标，以便在重画窗口时可以恢复直线。

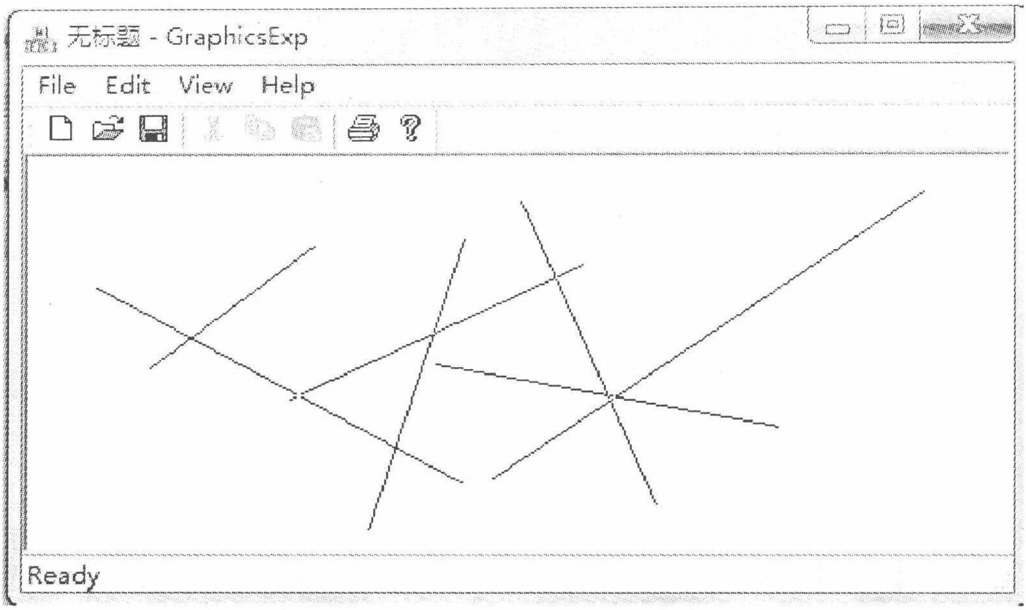


图 1.7 运行结果图

1. 定义直线类

(1)添加新类 Line 来存放直线的坐标。

如图 1.8 所示，在类视图“Class View”中根节点 GraphicsExp 上单击鼠标右键，在弹出的快捷菜单 Add 中选择“Class...”，弹出对话框如图 1.9 所示。

在图 1.9 所示对话框中，在左侧“Categories”中选择“C++”，单击“Add”按钮，弹出如图 1.10所示对话框。在“Class name”类名处输入 Line，单击“Finish”按钮。

在 Line. h 头文件中添加如下成员变量和成员函数：

```
CPoint m_pntStart;        //存放直线的起点坐标
CPoint m_pntEnd;          //存放直线的终点坐标
Line(CPoint pntStart, CPoint pntEnd);
void DrawLine(CDC * pDC);//画直线
```

在 Line. cpp 实现文件中添加如下代码：

```
Line::Line(CPoint pntStart, CPoint pntEnd)
{   m_pntStart=pntStart;
    m_pntEnd=pntEnd;
}
voidLine::DrawLine(CDC * pDC)
{   pDC->MoveTo(m_pntStart);
    pDC->LineTo(m_pntEnd);
}
```

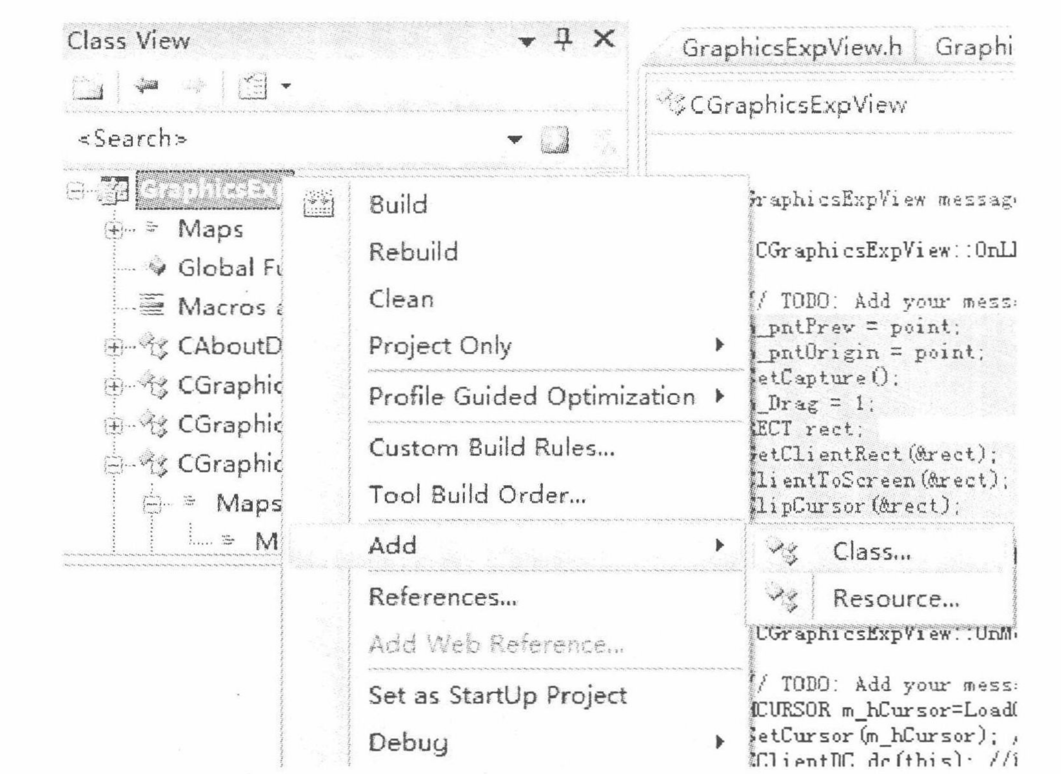


图 1.8　添加新类

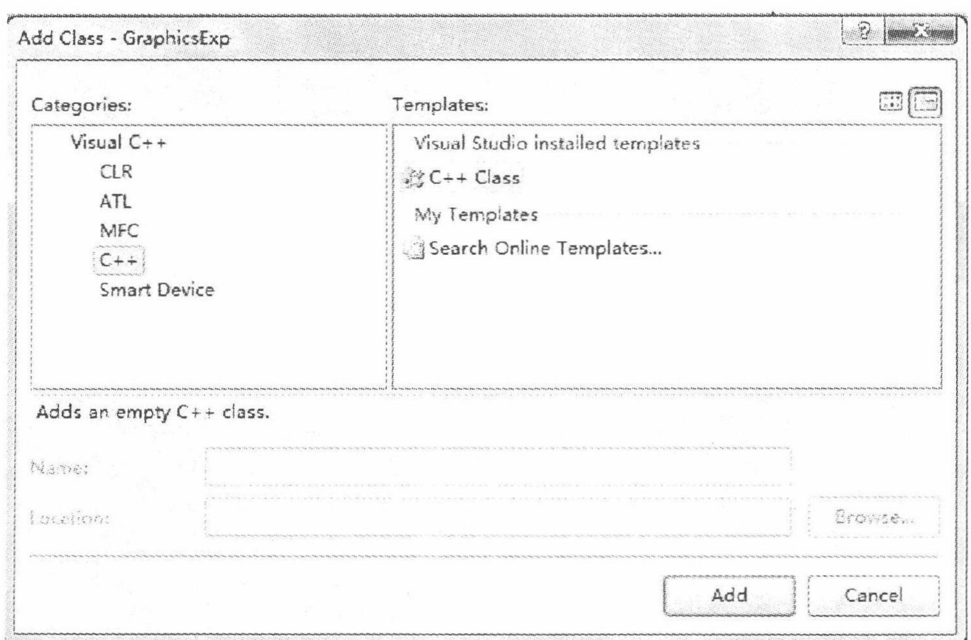


图 1.9　Add Class 对话框

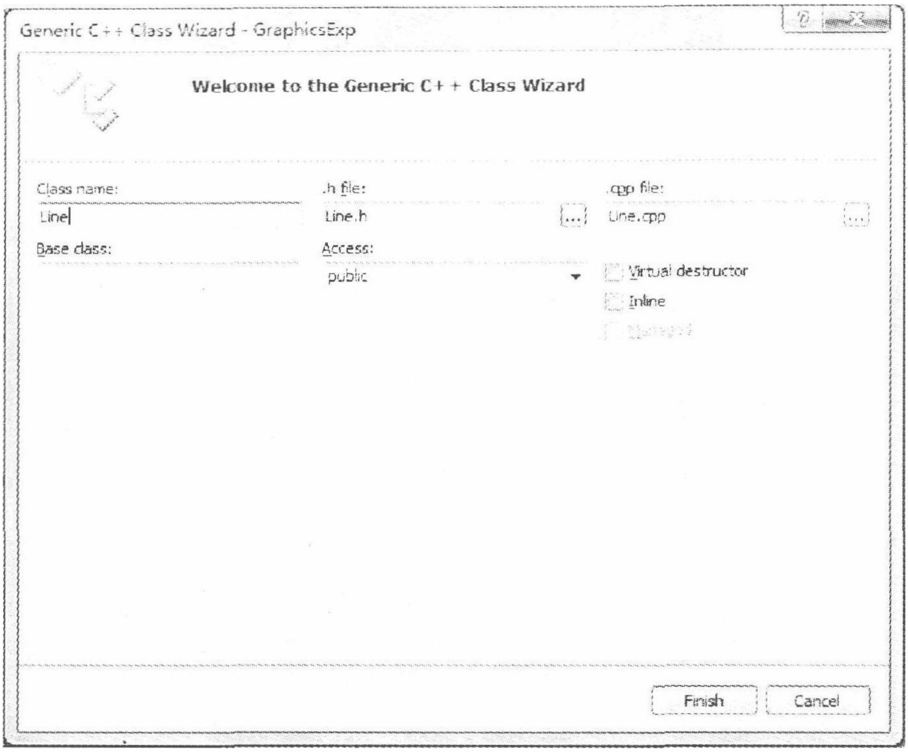


图 1.10　设置新类属性

(2)在 GraphicsExpDoc. h 文件中包含头文件,代码如下:

```
# include "Line.h"
# include <vector>
using namespace std;
```

并在 CGraphicsExpDoc 类中添加如下代码:

```
protected:
    vector <Line> m_vLines;//存放直线
public:
    void AddLine(CPoint pntStart, CPoint pntEnd);
    Line GetLine(int index);
    int GetLineNumber();
```

(3)在 GraphicsExpDoc. cpp 文件中添加如下代码:

```
void CGraphicsExpDoc::AddLine(CPoint pntStart, CPoint pntEnd)
{   m_vLines.push_back(Line(pntStart, pntEnd));   }

Line CGraphicsExpDoc::GetLine(int index)
{      return m_vLines.at(index);      }
```