



国家新闻出版改革发展项目库入库项目
高等院校计算机类规划教材
全国高等院校计算机基础教育研究会立项项目成果



计算机图形学 可视化编程

杨坡 白刚 编著



北京邮电大学出版社
www.buptpress.com



国家新闻出版改革发展项目库入库项目
高等院校计算机类规划教材
全国高等院校计算机基础教育研究会立项项目成果

计算机图形学可视化编程

杨坡 白刚 编著



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书将计算机图形学的基本原理与编程实践相结合,绘制出具体图形,形象地呈现给读者。本书将 Microsoft Visual Studio 集成开发环境的 MFC 框架作为开发平台,可以更好地展示计算机图形学算法形成的各种图形以及实现对图形的交互式操作。

本书可以作为高等院校计算机相关专业的本科生上机实践教材,也可作为计算机图形学研究者的学习参考书。

图书在版编目(CIP)数据

计算机图形学可视化编程 / 杨坡, 白刚编著. -- 北京: 北京邮电大学出版社, 2021. 10
ISBN 978-7-5635-6518-4

I. ①计… II. ①杨… ②白… III. ①计算机图形学—高等学校—教材 IV. ①TP391.411
中国版本图书馆 CIP 数据核字(2021)第 186208 号

策划编辑: 马晓仟 责任编辑: 廖娟 封面设计: 七星博纳

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号

邮政编码: 100876

发行部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 保定市中画美凯印刷有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 8.25

字 数: 216 千字

版 次: 2021 年 10 月第 1 版

印 次: 2021 年 10 月第 1 次印刷

ISBN 978-7-5635-6518-4

定价: 25.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

计算机图形学是利用计算机研究图形生成和显示的一门重要计算机学科。本书作者积累了丰富的计算机图形学讲授经验,使用 VS 的 MFC 框架开发了涉及直线段的扫描转换、圆弧的扫描转换、多边形填充、裁剪、二维图形的几何变换和真实感图形等知识点内容的九个案例,具体安排如下。

第 1 章:介绍 VS 环境下的图形编程框架。

第 2 章:展示直线段扫描转换算法形成的直线。

第 3~4 章:使用中点画圆法和 Bresenham 算法绘制圆、圆弧。

第 5 章:实现多边形扫描线填充算法。

第 6~7 章:实现直线段和多边形的裁剪算法。

第 8 章:展示二维图形的几何变换。

第 9 章:综合使用图形学知识绘制真实感物体。

对于 VS 的 MFC 框架,本书从操作者的角度进行了详细讲述,给出了大部分 *.h 和 *.cpp 文件,逻辑清晰,读者可以很容易地按照本书提供的步骤一步步完成上机实践。涉及核心算法的部分,读者查阅相关图形学资料自行实现即可。

本书通俗易懂,注重实践,能帮助计算机图形学的入门者快速掌握各种绘图算法,并使其对该领域产生浓厚的兴趣,为后续的深入学习和熟练应用打下坚实的基础。

在本书编写过程中,作者参考了很多计算机图形学相关的网络资源和书籍,在此向这些提供帮助的学者致谢。限于作者的时间和水平,书中难免有疏漏之处,欢迎读者批评指正。

目 录

第 1 章 VS 环境下的图形编程	1
实验目的	1
实验内容	1
实验指导	1
第 2 章 直线段的扫描转换算法	17
实验目的	17
实验内容	17
实验指导	18
第 3 章 圆周画法	30
实验目的	30
实验内容	30
实验指导	30
第 4 章 圆弧画法	41
实验目的	41
实验内容	41
实验指导	41
第 5 章 多边形扫描线填充算法	53
实验目的	53
实验内容	53
实验指导	53
第 6 章 直线段裁剪算法	72
实验目的	72

实验内容	72
实验指导	72
第 7 章 多边形裁剪算法	82
实验目的	82
实验内容	82
实验指导	82
第 8 章 二维图形的几何变换	91
实验目的	91
实验内容	91
实验指导	91
第 9 章 绘制真实感房屋简图	104
实验目的	104
实验内容	104
实验指导	105
参考文献	123

第 1 章 VS 环境下的图形编程

实验目的

- 熟悉 VS 环境下图形编程的基本概念
- 掌握 VS 环境下图形编程的基本方法

实验内容

实现一个简单的图形界面程序。要求：制作“画线”菜单项，单击该菜单项后，程序弹出对话框，操作者在对话框中输入直线起点和终点的二维坐标值，单击“确定”按钮后，在绘图区域绘制出对应的直线段。

实验指导

1. 背景知识

(1) 设备上下文

在 C++ 程序设计中，与设备（如监视器、打印机等）打交道的是 Device Context（设备上下文或设备场景），其保存的是设备的相关数据。在 MFC 中，其基类为 CDC。在绘图程序中往往使用 CDC 的几个派生类 CClientDC（客户区设备上下文）、CPaintDC 和 CWindowDC。其中，CClientDC 负责客户区的输出，CPaintDC 主要用于 OnPaint 消息响应，CWindowDC 的输出范围为整个监视器。详细的使用信息可以通过 MSDN 或互联网进行查找。

(2) 图形坐标系

屏幕坐标

屏幕坐标是描述物理设备（显示器、打印机等）的坐标系。坐标系原点在屏幕的左上角，X 轴水平向右为正，Y 轴垂直向下为正，度量单位是像素。原点、坐标轴方向和度量单位都是不能改变的。

设备坐标

又称为物理坐标，是描述屏幕中和打印机显示或打印的窗体坐标系。默认坐标原点在客户区的左上角，X 轴水平向右为正，Y 轴垂直向下为正，度量单位为像素。原点和坐标轴方向可以改变，但是度量单位不可以改变。

逻辑坐标

逻辑坐标是在程序中控制显示、打印使用的坐标系。该坐标系与定义的映射模式有关，默认

的映射模式为 MM_TEXT。我们可以通过设置不同的映射模式来改变该坐标系的默认行为。

(3) 常用绘图函数

可以使用“CClientDC dc(this)”语句建立设备上下文对象。设备上下文对象是对相应显示设备的逻辑抽象,提供对显示设备的操作。

常用的绘画函数:

```
dc.MoveTo(x1,y1); // 将当前点移动到相应坐标位置(x1,y1)
dc.LineTo(x2,y2); // 由当前坐标位置到(x2,y2)画一条直线
dc.SetPixel(x,y,color); // 使用指定颜色 color 在指定位置(x,y)画点
dc.GetPixel(x,y); // 获得指定位置像素点的颜色
```

(4) RGB 颜色系统

系统常用 COLORREF 数据类型来指定颜色值, COLORREF 数据类型的格式为 0x00BBGGRR,其中 BB 指定蓝色值分量,GG 指定绿色值分量,RR 指定红色值分量。每个颜色分量使用一个字节来表示,故取值范围为 0~255,可以使用 RGB 宏来将单色分量转换成 COLORREF 变量。

2. 建立单文档应用程序

第一步:在 VS 开发环境菜单中,选择“文件”→“新建”→“项目”菜单项,如图 1.1 所示。

第二步:在“新建项目”窗口,选择“MFC 应用程序”,填写项目名称和项目路径后,单击“确定”按钮,如图 1.2 所示。



动态画线

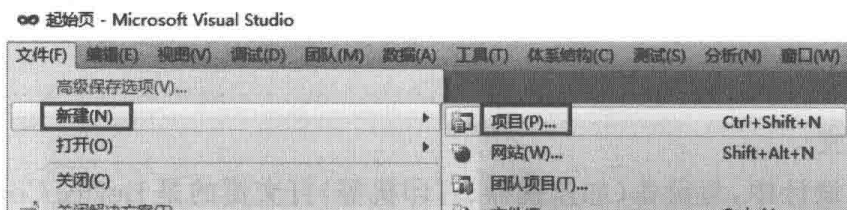


图 1.1 新建项目



图 1.2 选择 MFC 应用程序

第三步:在弹出的窗口中,单击“下一步”按钮,如图 1.3 所示。

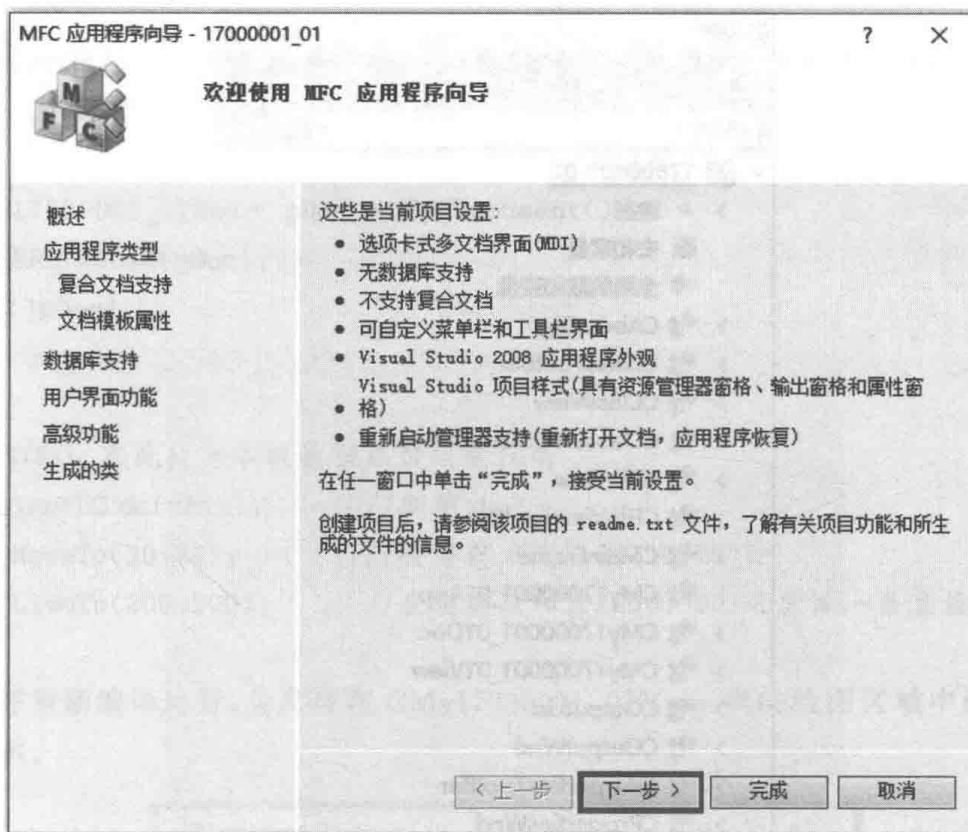


图 1.3 MFC 应用程序向导

第四步:“应用程序类型”选择“单个文档”。最后,单击“完成”按钮,如图 1.4 所示。

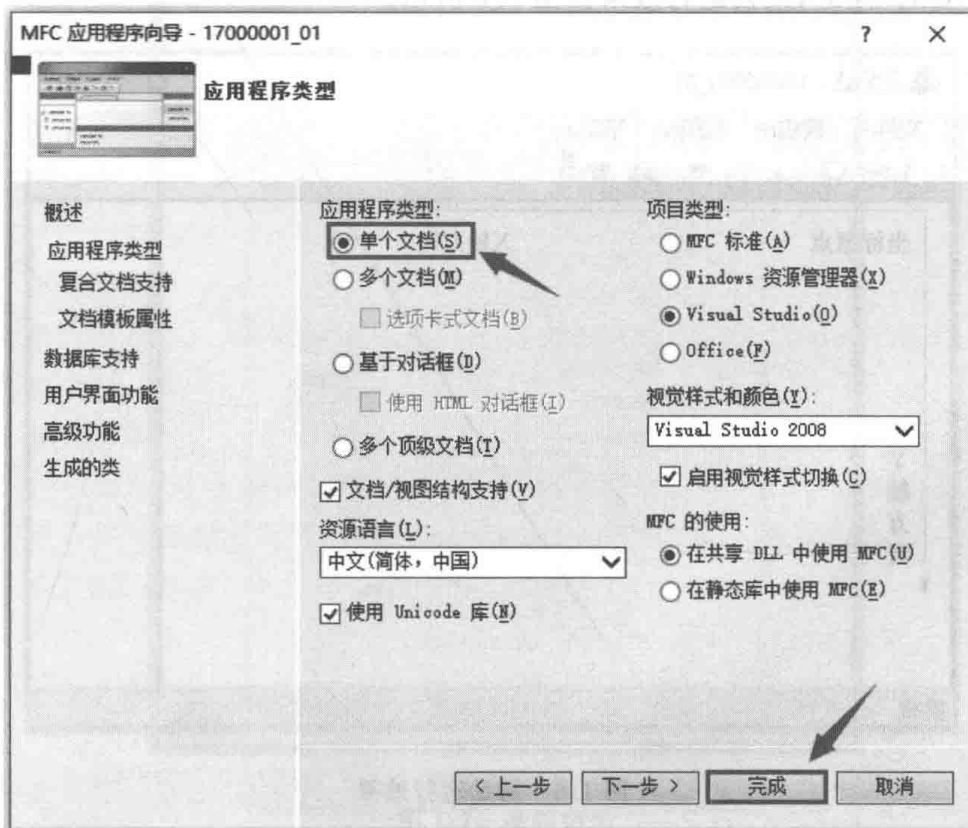


图 1.4 选择“单文档”应用程序

至此,新建项目完毕。新建项目的类视图如图 1.5 所示。

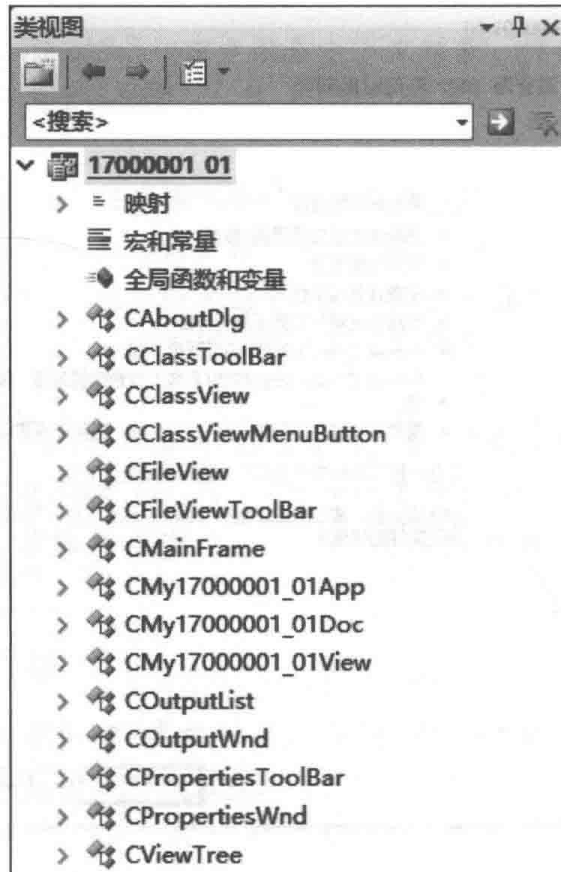


图 1.5 项目类视图

按快捷键“Ctrl+F5”,最初运行效果如图 1.6 所示。

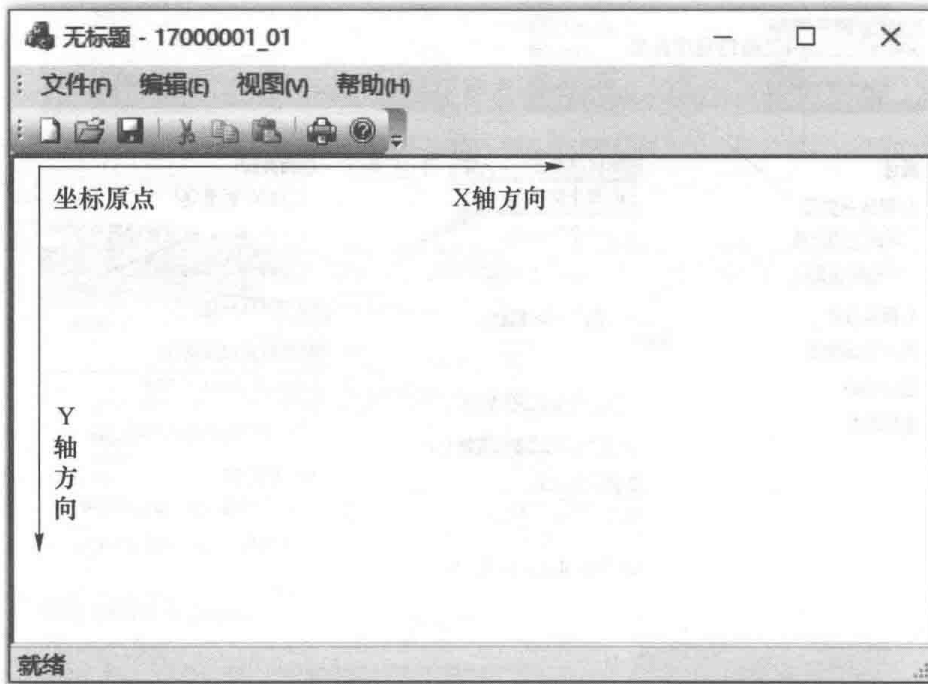


图 1.6 初始运行效果

图中白色区域就是“CMy17000001_01View”类对象的绘图区域,坐标原点为左上角。

3. 添加绘图功能代码

找到 CMy1700001_01View 类的成员函数 OnDraw,在提示位置添加代码,如下所示:

```
void CMy1700001_01View::OnDraw(CDC * /* pDC */)  
{  
    CMy1700001_01Doc * pDoc = GetDocument();  
    ASSERT_VALID(pDoc);  
    if (!pDoc)  
        return;  
  
    // TODO: 在此处为本机数据添加绘制代码  
    CClientDC dc(this);    //创建 dc  
    dc.MoveTo(30,20);    //将当前点定位到(30,20)处  
    dc.LineTo(200,300);    //在当前点和点(200,300)之间画一条直线  
}
```

保存程序重新编译运行,会发现在 CMy1700001_01View 类的绘图区域中画出一条直线,如图 1.7 所示。

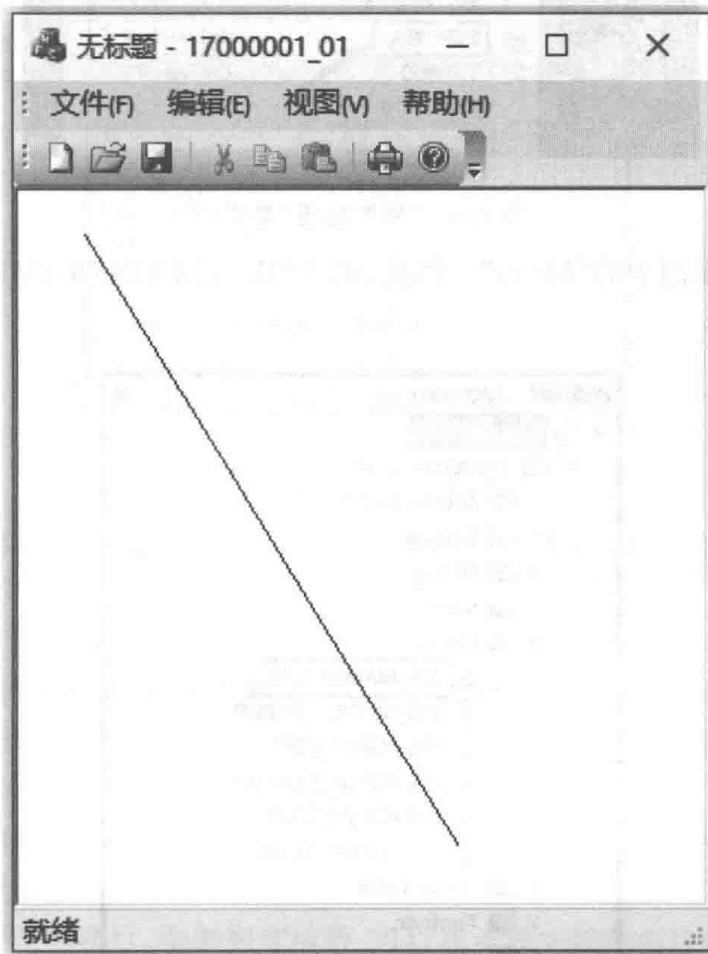


图 1.7 画线结果

又如,输入如下代码:

```
dc.SetPixel(50, 50, RGB(0,0,255));
```

将在坐标点(50,50)处画出一个蓝色的像素点。

4. 菜单项的消息响应

(1) 添加菜单项

第一步:选择菜单栏中的“视图”→“资源视图”,打开项目的资源视图窗口,如图 1.8 所示。

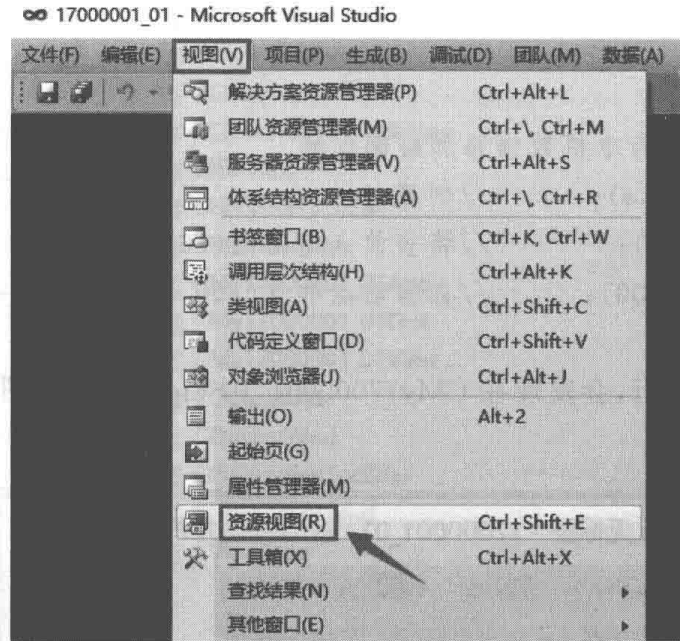


图 1.8 “资源视图”菜单项

第二步:打开资源视图中的“Menu”文件夹,双击“IDR_MAINFRAME”,打开项目的菜单资源,如图 1.9 所示。

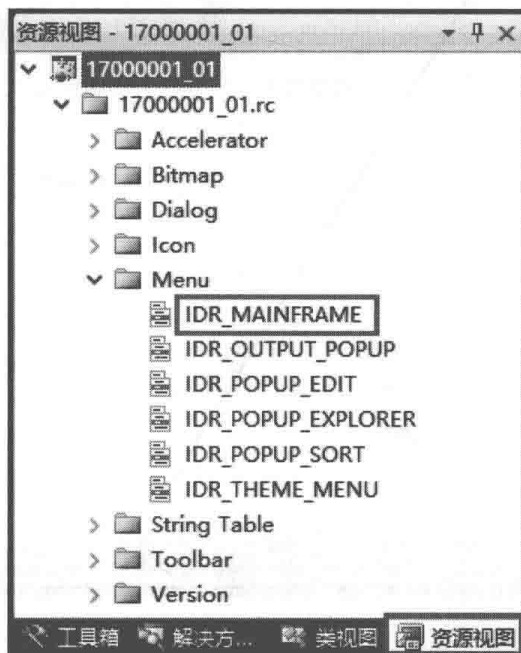


图 1.9 “资源视图”窗口

第三步:在菜单栏中,添加“作图”→“画线”,如图 1.10 所示。

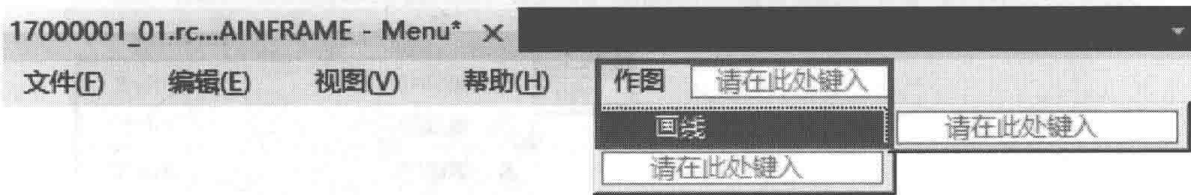


图 1.10 添加菜单项

第四步:单击“画线”菜单项,在属性窗口内,为其设置合适的 ID 值,如图 1.11 所示。

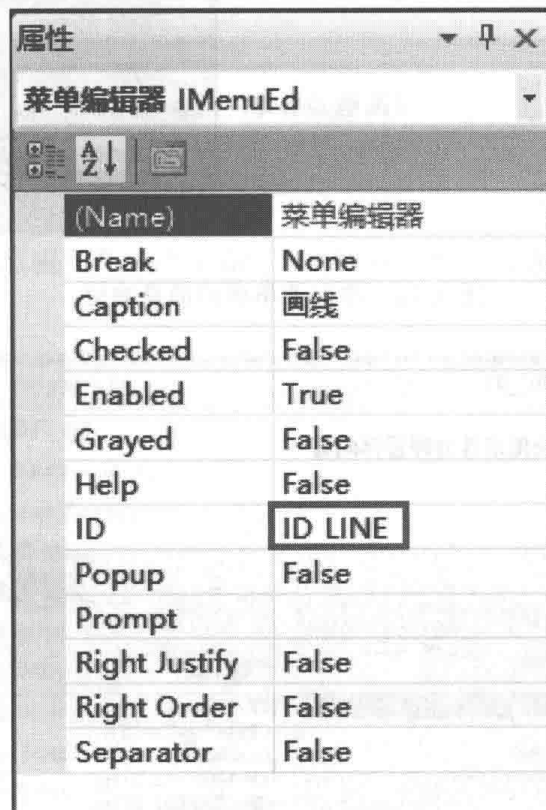


图 1.11 设置菜单项的 ID 值

当操作者单击该菜单项时,系统会生成一个携带相应 ID 号的 WM_COMMAND 消息,并放置在消息队列中。

(2) 添加消息响应

第一步:选中“画线”菜单项,单击右键选择“添加事件处理程序”,如图 1.12 所示。

第二步:在“事件处理程序向导”界面的类列表中,选择“CMy17000001_01View”类,消息类型选择“COMMAND”,并填写合适的函数名称,如图 1.13 所示。

此时,点击“添加编辑”按钮,发现在“17000001_01View.cpp”文件内生成了“作图”菜单项的响应函数,如图 1.14 所示。

操作者单击“作图”菜单项后,会调用此函数,可以在函数中添加相应的响应代码。

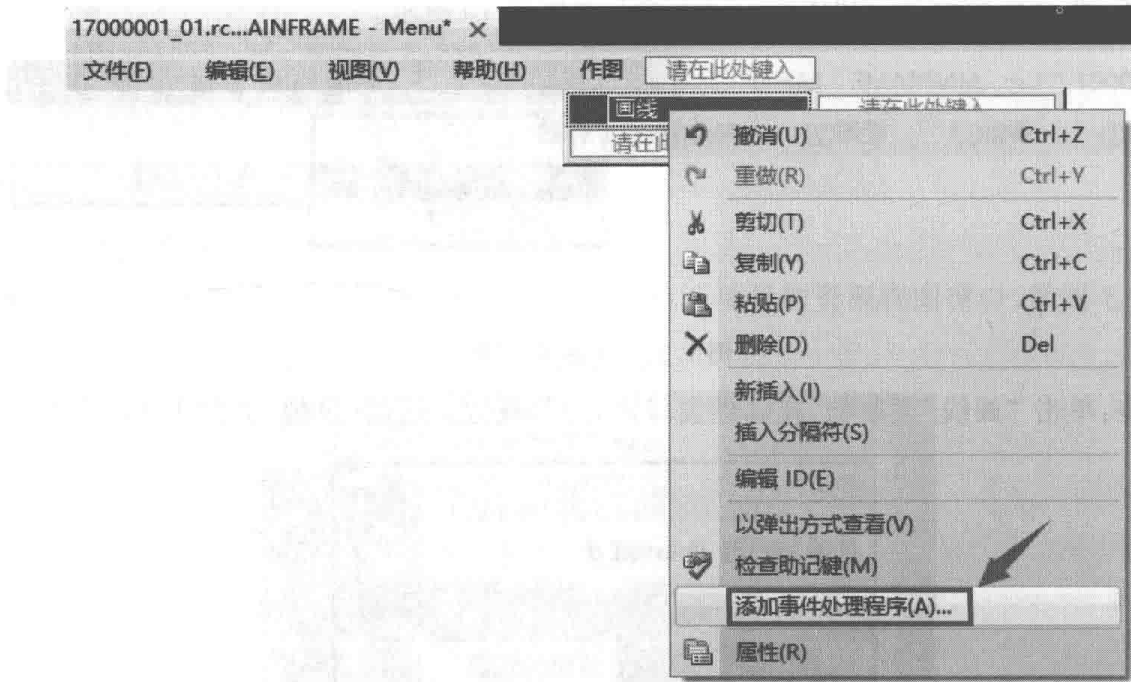


图 1.12 添加菜单项的消息响应

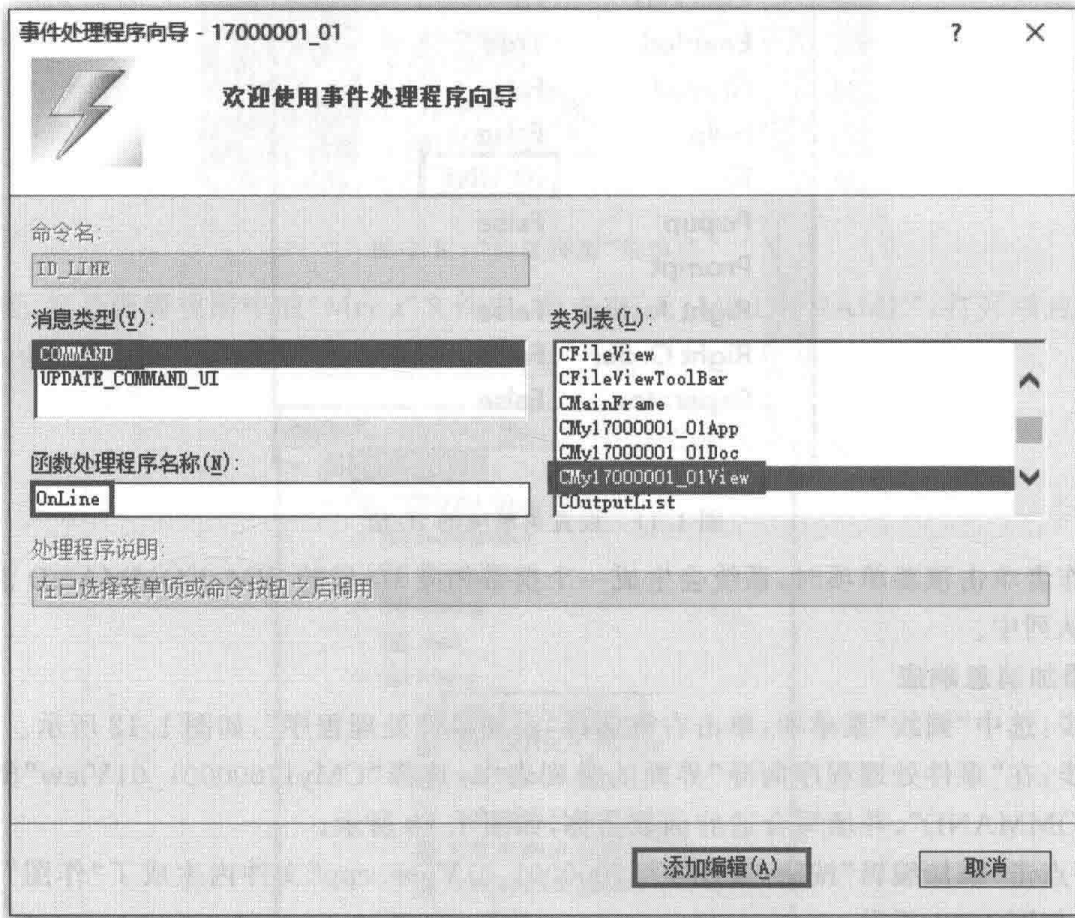


图 1.13 事件处理程序向导

```
17000001_01View.cpp x
(全局范围)
129
130
131 // CMy17000001_01View 消息处理程序
132
133
134 void CMy17000001_01View::OnLine()
135 {
136     // TODO: 在此添加命令处理程序代码
137 }
138
```

图 1.14 事件处理函数

5. 添加对话框

第一步:打开项目的“资源视图”,在“Dialog”文件夹上右击,选择“插入 Dialog”,如图 1.15 所示。

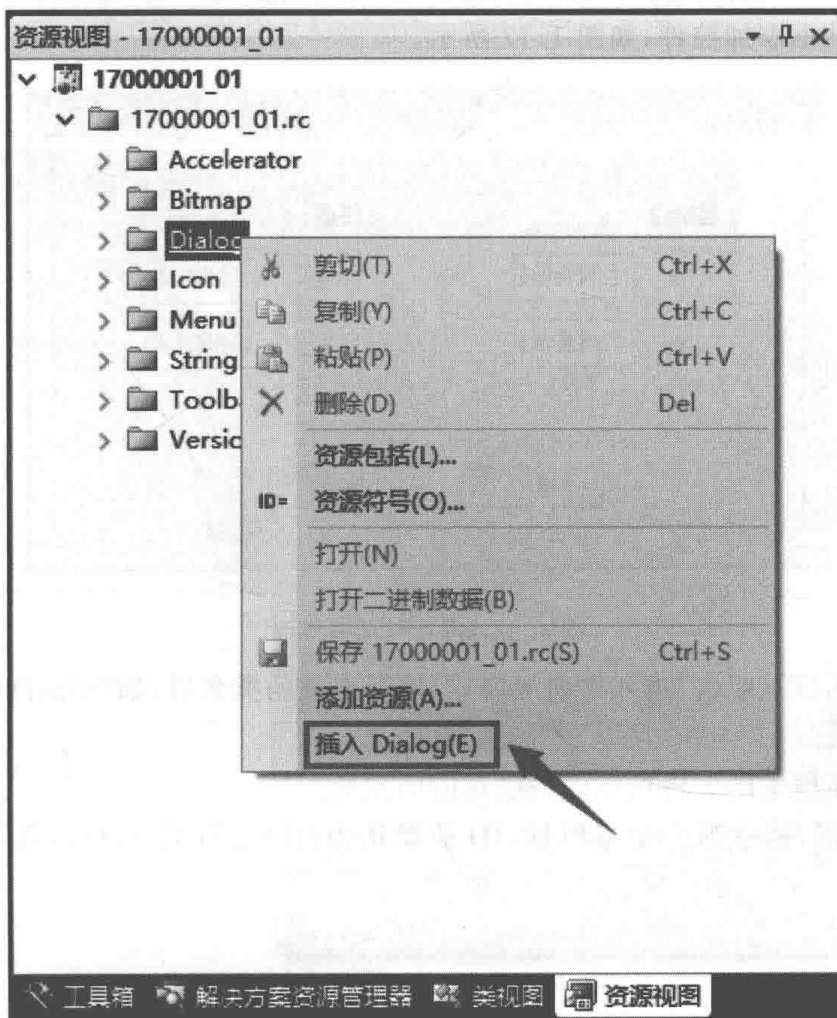


图 1.15 添加对话框

新添加的对话框初始界面如图 1.16 所示。

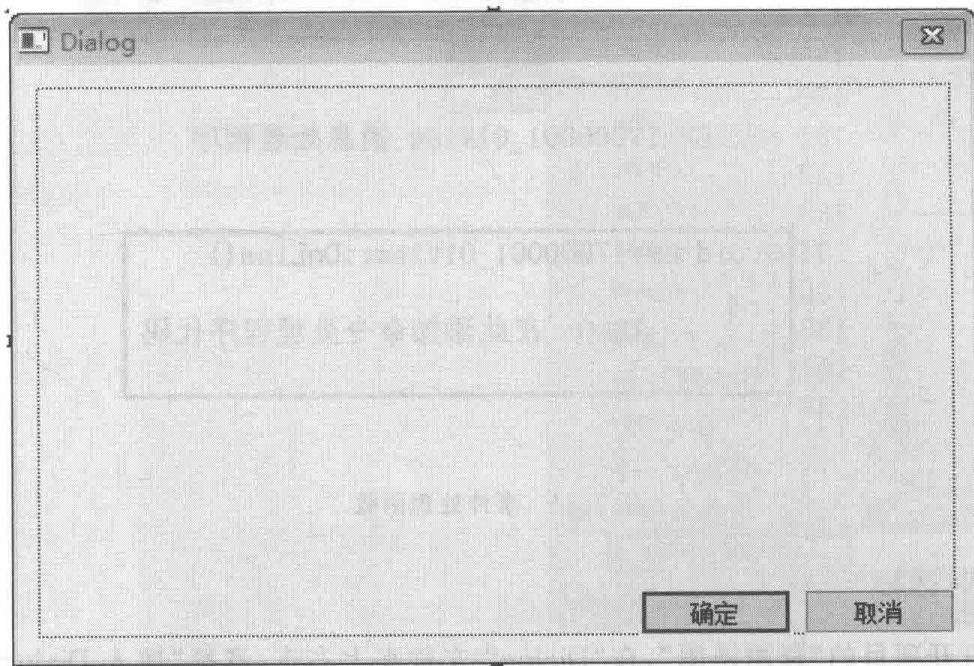


图 1.16 对话框初始界面

第二步:为对话框添加控件,如图 1.17 所示。

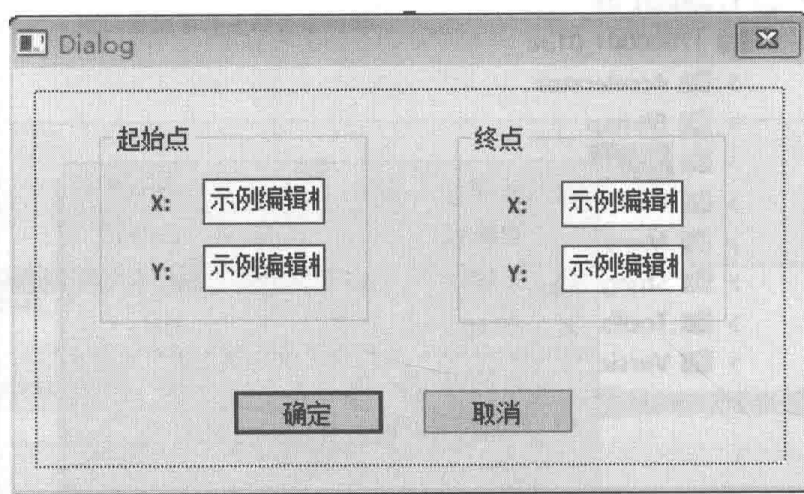


图 1.17 设计对话框界面

第三步:在对话框上双击,进入类向导窗口,填写合适的类名后,如“LineDlg”,单击“完成”按钮,如图 1.18 所示。

第四步:为对话框中的编辑框控件关联对应的变量。

(1) 打开对话框,选中第一个编辑框,ID 值默认为:IDC_EDIT1,右击选择“添加变量”,如图 1.19 所示。

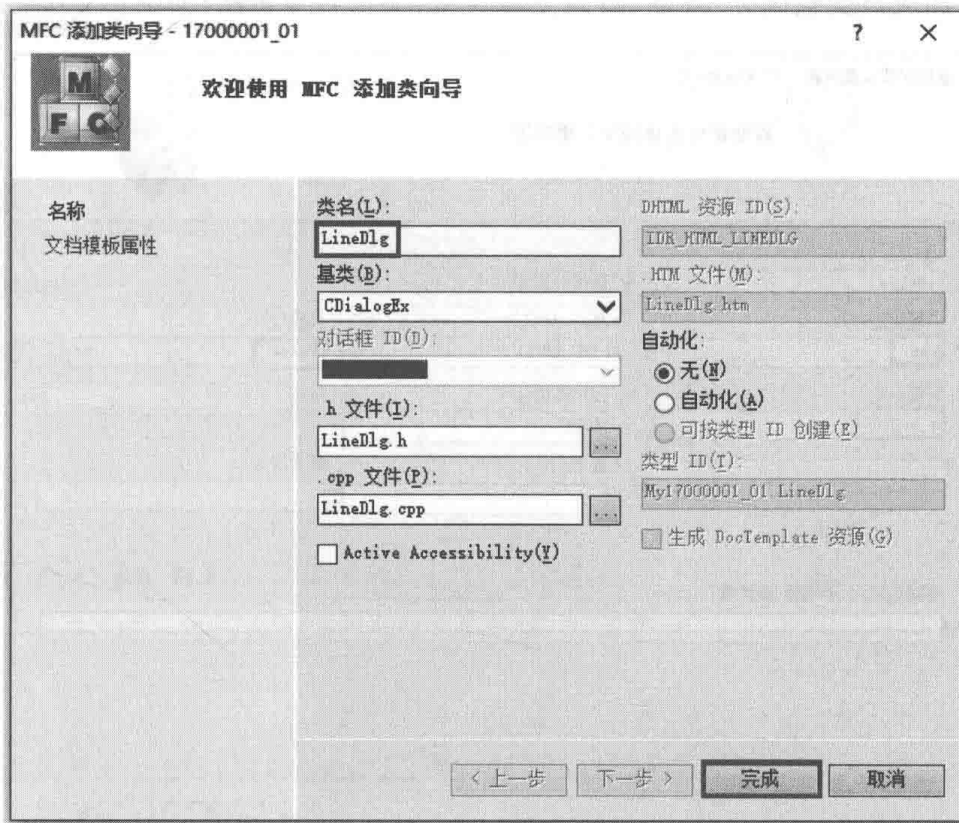


图 1.18 对话框类

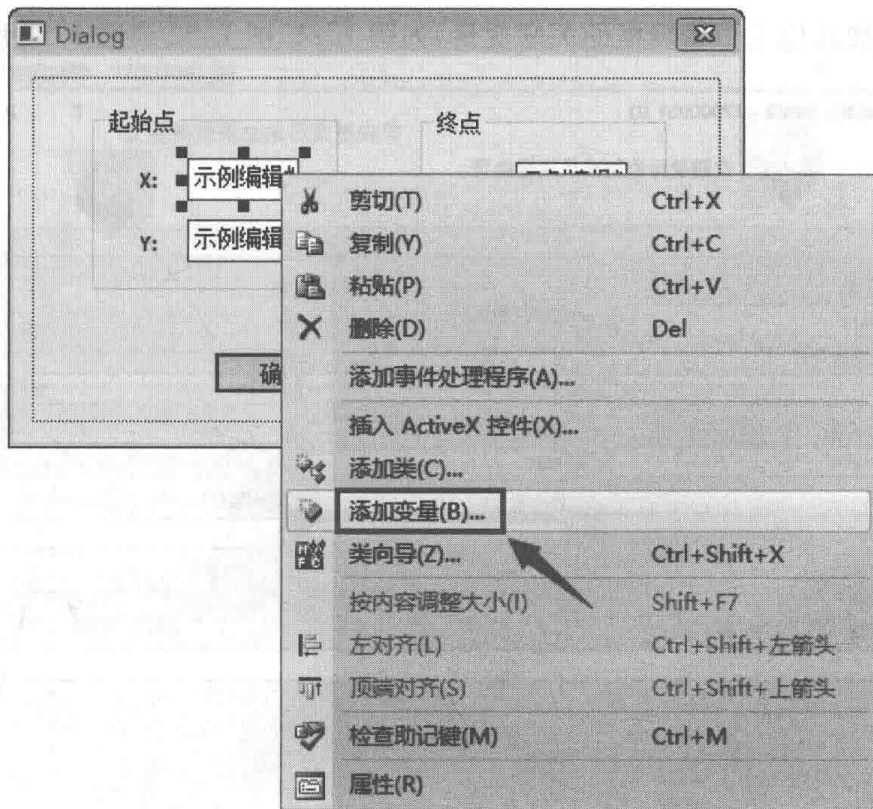


图 1.19 为编辑框关联变量