



21 世纪全国高校应用人才培养信息技术类规划教材
山西省省级精品课程配套教材



计算机图形学课程设计教程

(Visual C++版)

孔令德 编 著



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国高校应用人才培养信息技术类规划教材

计算机图形学课程设计教程 (Visual C++ 版)

孔令德 编著

ISBN 7-301-16300-3

187 毫米 × 109 毫米 16 开本 30 2 印张 324 千字
2010 年 6 月第 1 版 2010 年 6 月第 1 次印刷
定价：38.00 元



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书综合运用计算机图形学基本原理编写了“颜色渐变正八面体、三维五角星、光照线框球、光照正方体、金属球、圆环、钻石、太阳和月亮”等18个案例。所有案例采用面向对象语言 Visual C++ 6.0 的 MFC 平台开发,并未借助 OpenGL 或 DirectX 等图形库函数,但绘制的图形质量完全可以和后者相媲美。

本书以生成三维动态真实感图形为主线,涉及的算法主要有“中点 Bresenham 算法”、“基于背景色的反走样算法”、“有效边表填充算法”、“正交投影和透视投影算法”、“背面剔除算法”、“Z-Buffer 消隐算法”、“Gouraud 双线性光强插值算法”和“Phong 双线性法矢插值算法”等。以类的形式对算法进行封装,创建了三维光照场景,可以完成三维物体的透视变换、动态消隐、真实感光照和纹理映射等课程设计任务。

本书案例进阶合理、算法讲解详尽、类结构清晰,适合作为研究生教材和本科生课程设计教材,也可供计算机图形学爱好者自学使用。

图书在版编目(CIP)数据

计算机图形学课程设计教程(Visual C++ 版)/孔令德编著. —北京:北京大学出版社, 2010.6

(21 世纪全国高校应用人才培养信息技术类规划教材)

ISBN 978-7-301-16500-3/TP·0067

I. 计… II. 孔… III. ①计算机图形学—高等学校—教材②C 语言—程序设计—高等学校—教材 IV. TP391.41

中国版本图书馆 CIP 数据核字(2009)第 230949 号

书 名: 计算机图形学课程设计教程(Visual C++ 版)

著作责任者: 孔令德 编著

策 划 编 辑: 温丹丹

责 任 编 辑: 温丹丹

标 准 书 号: ISBN 978-7-301-16500-3/TP·0067

出 版 发 行: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

电 话: 邮购部 62752015 发行部 62750672 编辑部 62765126 出版部 62754962

网 址: <http://www.pup.cn>

电 子 信 箱: zyjy@pup.cn

印 刷 者: 河北滦县鑫华书刊印刷厂

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 20.5 印张 2 插页 524 千字

2010 年 6 月第 1 版 2010 年 6 月第 1 次印刷

定 价: 38.00 元

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究

举报电话: 010-62752024 电子信箱: fd@pup.pku.edu.cn

前言

亲爱的读者,当您欣赏到本书插图中的美丽图形时,您能相信这些图形是单纯使用面向对象语言 Visual C++ 6.0 的 MFC 编程平台开发的吗?这些图形的制作质量完全可以和使用 OpenGL 或 DirectX 等高级图形开发包所绘制的图形相媲美。

作为山西省省级计算机图形学教学名师,笔者一直在思考一个问题,如何才能有效提高学生的计算机图形学上机实践能力呢?经过多年的探索,笔者认为只有通过案例教学方式,让读者沉浸在一个个真实的案例中,才能真正掌握计算机图形学的基本原理和实现细节。

只要您认真阅读本书,然后运行本书案例的源程序,就可以轻松掌握使用 Visual C++ 的 MFC 框架绘制真实感图形的主要技术。本书包含了“金刚石图案”、“颜色渐变直线”、“反走样卫星图”、“RGB 颜色模型演示”、“填充任意多边形”、“颜色渐变三角形”、“旋转正方体”、“颜色渐变正八面体”、“贴图正方体”、“交叉面”、“三维五角星”、“光照正方体”、“光照线框球”、“台球”、“金属球”、“圆环”、“钻石”、“太阳和月亮”,共 18 个案例。这些案例由浅入深、从二维到三维、从局部光照模型到真实感图形,涵盖了计算机图形学的所有内容。

笔者通过本书的案例想告诉读者,计算机图形学并不是一门深不可测的课程,只要具备线性代数、数据结构知识,使用过 Visual C++ 程序设计语言,就可以按照本书的讲解,开发出和 3DS Max 一样精美的图形。当然读者也许会说,我没有学习过 Visual C++ 语言。为此,笔者假设读者使用过 Visual Basic 等基于对象的程序设计语言。在案例 1 金刚石图案中,以非常简单的平面图形算法,详细讲解了 MFC 的开发过程,形成了 Test 程序模板,后续的开发工作全部基于该模板实现。事实上,本书使用的只是 MFC 的框架,编程难度和 Visual Basic 基本一样。有读者会问,既然 Visual Basic 语言也可以绘制图形,为什么要选用比较难学的 Visual C++ 呢?原因是这样的,Visual C++ 是仅比汇编语言高一个层次的中级语言,该语言提供的面向对象功能十分完善。本书封装了计算机图形学中的大量图形类,如二维点类、三维点类、面类、矢量类、颜色类、Gouraud 光强插值填充类和 Phong 法矢插值填充等形成三维光照场景。读者只要会写出三维物体的数学模型,构建点表和面表就可以生成带有光照或纹理的三维真实感图形。又有读者会问,基于 Visual C++ 的 OpenGL 和 DirectX 高级图形库也可以进行图形的开发,而且图像质量和内存处理等方面都十分完美,为什么要使用单纯的 MFC 平台进行图形开发呢?这是因为,计算机图形学讲授的是基本图形的生成原理和算法,使用 MFC 开发可以掌握每一个算法原理和程序实现细节,而使用 OpenGL 或 DirectX 图形库时,这些原理和算法都被包装成函数库,读者只要知道相应函数的功能和接口参数就可以直接使用,不需要深入了解计算机图形学的基本原理。笔者认为,使用单纯的 MFC 进行图形开发属于研究范畴,而使用高级图形库开发属于应用范畴,两种学习方式大相径庭。作为山西省省级精品课程“C++ 程序设计”和“计算机图形学”的第一负责人,笔者一直致力于使用 MFC 进行计算机图形学原理和算法的实现。本书的 18 个案例给出了如下知识点。

“案例 1——金刚石图案”设计目的是建立 Test 工程模板,为后续的案例开发提供一

个通用的程序模板。

“案例2——颜色渐变直线”设计目的是使用中点 Bresenham 算法绘制任意斜率的直线，直线的颜色从起点到终点渐变，这种任务如果使用 MFC 本身的画线函数是无法实现的。

“案例3——反走样卫星图”设计目的是实现直线、圆、椭圆等二维基本图元的反走样算法。

“案例4——RGB 颜色模型演示”设计目的是让读者深入理解计算机图形显示器的物理颜色模型。

“案例5——填充任意多边形”设计目的是实现交互式绘图的基本技术，如回显、橡皮筋、约束、引力域等。

“案例6——颜色渐变三角形”设计目的是使用有效边表算法实现任意三角形的面颜色渐变填充，为真实感光照模型的绘制做知识储备。

“案例7——旋转正方体”设计目的是对三维物体的线框模型实现旋转、消隐、透视和动画等操作，构建三维动画场景。

“案例8——颜色渐变正八面体”设计目的是在面颜色渐变的基础上，实现三维物体的颜色渐变。

“案例9——贴图正方体”设计目的是对三维物体进行纹理映射，本案例将照片作为纹理绑定到正方体的各个表面上。

“案例10——交叉面”设计目的是实现 Z-Buffer 消隐算法，Z-Buffer 消隐算法是绘制真实感图形的基础算法。

“案例11——三维五角星”设计目的是绘制立体五角星，通过恰当设计顶点的颜色生成逼真的立体图形。

“案例12——光照正方体”设计的目的是构建三维光照场景。通过改变正方体的材质和光源的颜色，可以产生不同的光照效果。

“案例13——光照线框球”设计目的是使用直线反走样技术，绘制双点光源光照照射的球体。

“案例14——台球”设计目的是通过绘制 Gouraud 双线性光强插值模型，演示马赫带效应。

“案例15——金属球”设计目的是通过选择金属材质、双点光源参数来绘制 Phong 双线性法矢插值真实感图形。

“案例16——圆环”设计目的是通过选择材质、光源参数来绘制单点光源照射的 Gouraud 双线性光强插值真实感图形。

“案例17——钻石”设计目的是采用多点光源光照模型来模拟真实钻石。

“案例18——太阳和月亮”设计目的是将材质模型、光源模型和纹理映射结合起来模拟大自然中的太阳和月亮。

本书的每个案例虽然独立，设计目的也不尽相同，但这些案例以一条主线贯穿：直线（任意斜率的直线及其反走样技术）→面填充（有效边表填充、边缘填充）→立体投影（正交投影和透视投影）→立体消隐（背面剔除、Z-Buffer 消隐）→简单光照模型（材质、纹理、光源）→真实感图形（正方体、线框球、台球、金属球、环、金刚石、太阳和月亮）。通读本书可以掌握三维立体的真实感图形生成原理和算法。本书不仅详细讲解了相关的基本图形学原理，而且给出了完整的实现代码，二者一一对应。作者愿意和读者手挽手，一步步进入计算机图形学殿堂。

本书每个案例的源程序可以从笔者的个人网站：<http://www.klingde.com> 下载。所有案例建议在 1024 × 768 分辨率下运行。

在本书编写过程中，计算机工程研究所的潘晓、左亮亮和王鹏参与了调试工作，并提出许多宝贵的建议，在此深表感谢。

本书不仅可以作为大学计算机专业本科生教材，也可作为计算机图形学研究生教材，还可供计算机图形学爱好者自学使用。希望本书的出版对我国 3D 计算机图形学事业有所帮助，同时也请计算机图形图像处理方面的专家留下宝贵的指导意见。来信请寄：klingde@163.com。

孔令德

2010 年 5 月 22 日

目 录

案例 1	金刚石图案	1
案例 2	颜色渐变直线	23
案例 3	反走样卫星图	54
案例 4	RGB 颜色模型演示	78
案例 5	填充任意多边形	92
案例 6	颜色渐变三角形	106
案例 7	旋转正方体	135
案例 8	颜色渐变正八面体	159
案例 9	贴图正方体	165
案例 10	交叉面	171
案例 11	三维五角星	189
案例 12	光照正方体	197
案例 13	光照线框球	224
案例 14	台球	240
案例 15	金属球	246
案例 16	圆环	261
案例 17	钻石	271
案例 18	太阳和月亮	291
参考文献		319

案例 1 金刚石图案



本案例知识要点

- 创建 Test 程序模板的 MFC 框架
- 设计个性化的菜单项和工具条
- CTestView 类的修改
- 绘制金刚石图案的算法
- 一维堆内存的分配与释放

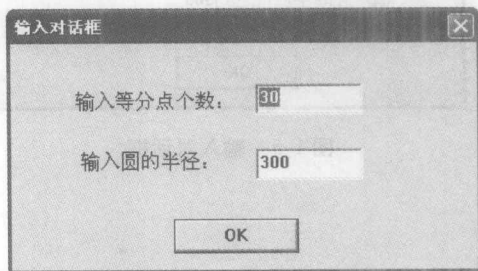
一、案例需求

1. 案例描述

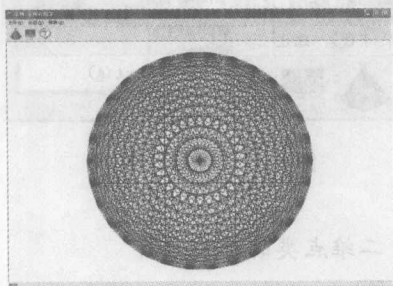
将半径为 R 的圆周 N 等份，然后用直线将各等分点隔点相连，形成的图案称为“金刚石”图案，试编程实现。

2. 案例效果图

金刚石图案的输入对话框和效果图如图 1-1 所示。



(a) 输入对话框



(b) 效果图

图 1-1 金刚石图案

3. 功能说明

- (1) 程序运行界面提供“文件”、“绘图”和“帮助”3个菜单项。
- (2) 工具条提供和菜单项对应的图标。
- (3) 单击菜单的“绘图”项或工具条的“绘图”按钮，弹出输入对话框，分别读入圆的等分点个数和圆的半径。
- (4) 以屏幕客户区中心为圆心绘制金刚石图案。

二、案例分析

本案例设计的目的是学习使用 Visual C++ 的 MFC 开发平台建立 Test 工程，为后续的

案例设计提供一个通用的程序模板。Test 工程包含了菜单设计、工具条图标按钮设计、输入对话框设计和关于对话框设计, 在建立 Test 程序模板的基础上, 本案例以绘制金刚石图案为例, 讲解了二维点类 CP2 的设计方法和 CTestView 类的修改方法。

1. 菜单分析

根据案例需求, 需要在 MFC 环境中建立一个由“文件”、“绘图”和“帮助”3 个菜单项组成的弹出菜单, 其中“文件”的子菜单为“退出”, 完成退出应用程序的工作, 如图 1-2 所示; “绘图”的子菜单为“金刚石”, 用于绘制金刚石图案, 如图 1-3 所示; “帮助”的子菜单为“关于”, 用于显示开发人员信息, 如图 1-4 所示。

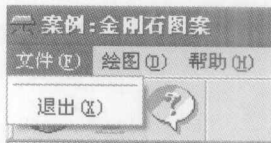


图 1-2 “文件”菜单

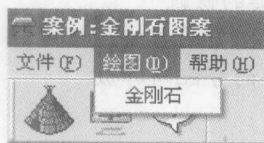


图 1-3 “绘图”菜单

定义一个输入对话框类, 提供“等分点个数”和“圆的半径”两个参数的输入界面, 如图 1-5 所示。

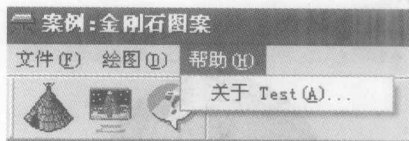


图 1-4 “帮助”菜单

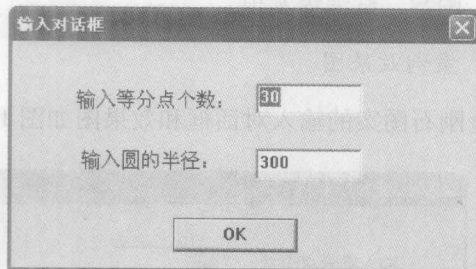


图 1-5 输入对话框

2. 二维点类的结构分析

本案例定义了一个二维坐标点类 CP2, 用于对各个点的 double 型坐标 (x, y) 进行整体处理, 如图 1-6 所示。

CP2
+x: double
+y: double

图 1-6 二维坐标点类图

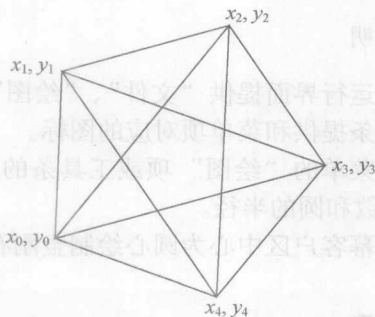


图 1-7 $n=5$ 时的线段连接

3. 金刚石图案分析

本案例设计的难点是设计内层循环时不进行重复线段的连接。例如当圆的等分点个数 $n=5$ 时, 线段的连接情况如图 1-7 所示, 线段连接点见表 1-1。

表 1-1 线段连接方式

起 点	终 点
(x_0, y_0)	$(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$
(x_1, y_1)	$(x_2, y_2), (x_3, y_3), (x_4, y_4)$
(x_2, y_2)	$(x_3, y_3), (x_4, y_4)$
(x_3, y_3)	(x_4, y_4)

为此, 设计一个二重循环, 代表起点的外层整型变量 i 从 $i=0$ 循环到 $i=n-2$, 代表终点的内层整型变量 j 从 $j=i+1$ 循环到 $j=n-1$ 。以 $(p[i].x, p[i].y)$ 为起点, 以 $(p[j].x, p[j].y)$ 为终点连接各线段。

说明: 类图 1-6 中, “+”代表公有成员, “-”代表私有成员, “#”代表保护成员。

4. 一维堆内存的分配与释放

金刚石图案的圆的等分点个数是输入值, 需要定义动态对象数组。本例定义了 CP2 类的一维对象数组指针 $CP2 * P$ 。使用动态对象数组, 可以避免静态数组的“大开小用”。动态数组的分配都在堆区中进行, 动态数组的大小只有在程序运行时才能确定, 这样编译器在编译时就无法为它们预定存储空间, 只能在程序运行时系统根据输入值进行内存分配, 这种方法称为动态存储分配。

Visual C++ 中一维动态数组分配的格式为:

指针变量名 = new 类型名 [下标表达式];

new 运算符返回的是一个指向所分配类型数组的指针, 动态创建的数组本身没有名字。

Visual C++ 中动态数组释放的格式为:

delete[] 指向该数组的指针变量名;

数组分配格式和数组释放格式中的方括号是非常重要的, 两者必须配对使用, 如果 delete 语句中少了方括号, 编译器认为该指针是指向数组第一个元素的指针, 就会产生回收不彻底的问题 (只回收了第一个元素所占空间), 加了方括号后就转化为指向数组的指针, 回收了整个数组。delete[] 的方括号中不需要填数组元素数, 系统自己确定。即便写了, 编译器也会忽略。

三、案例设计

1. Test 程序模板的设计

Visual C++ 提供了软件代码自动生成和可视化资源编辑功能。微软基类库 (Microsoft Foundation Class Library, MFC) 是以 C++ 形式封装的 Windows API (Application Program Interface), 包含了一百多个已经定义好的常用类。在程序设计过程中, 一般是通过调用 MFC 的已有类来完成设计任务的。作为上机操作的基础, 首先讲解基于 MFC 的 Test 程序

模板的开发过程。

(1) 从 Windows 的程序菜单中启动 Microsoft Visual C++ 6.0, 如图 1-8 所示。

(2) 在图 1-9 所示的 Visual C++ 集成开发环境中, 选择 File | New 命令, 弹出 New 对话框, 切换到 Projects 选项卡。在左边窗口中选择 MFC AppWizard (exe), 在右边的 Projects name 文本框中输入应用程序名, 这里输入 Test, 在 Location 文本框中设置应用程序存放目录, 这里设置为 D:\Test, 其余保持默认值, 如图 1-10 所示。单击 OK 按钮。



图 1-8 Microsoft Visual C++ 6.0 启动菜单

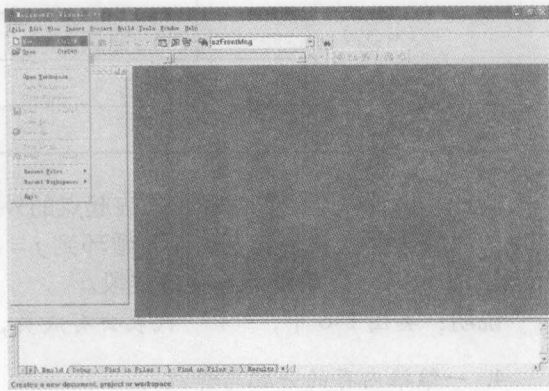


图 1-9 Visual C++ 集成开发环境

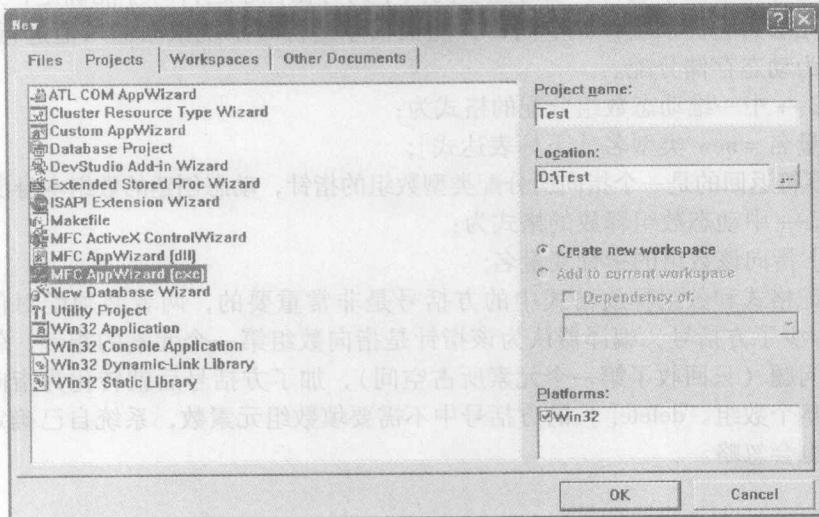


图 1-10 New 对话框

(3) 在 MFC AppWizard-Step1 对话框中, 选中 Single Document 单选按钮, 其余保持默认值, 如图 1-11 所示。单击 Finish 按钮结束。

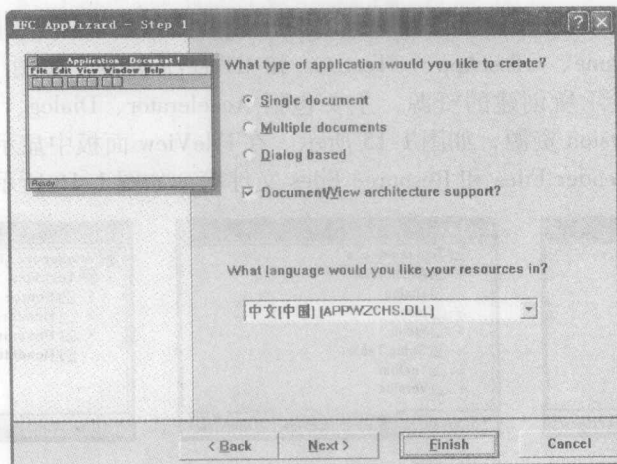


图 1-11 MFC AppWizard-Step1 对话框

(4) 弹出 New Project Information 对话框，如图 1-12 所示。单击 OK 按钮。

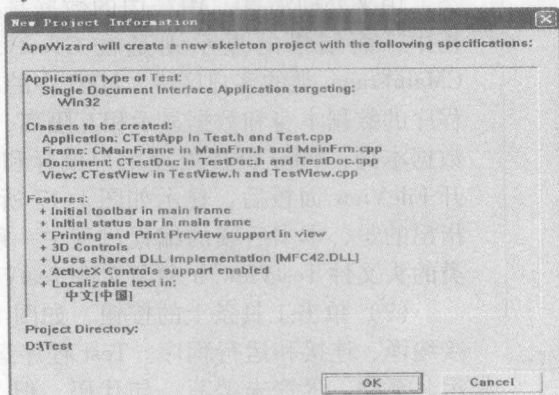


图 1-12 New Project Information 对话框

(5) 完成上述步骤后，应用程序 Test 的 MFC 框架即被生成，出现程序工作区。如图 1-13 所示。

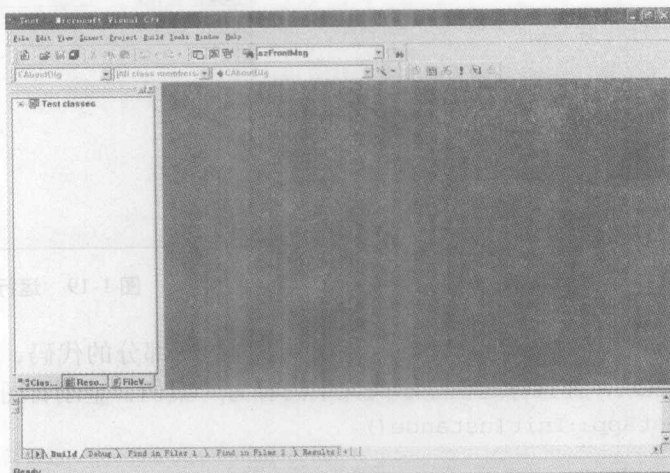


图 1-13 应用程序的 MFC 框架

(6) 在工作区的 ClassView 面板中显示 MFC AppWizard(exe) 所创建的类, 主要包括 CAboutDlg、CMainFrame、CTestApp、CTestDoc 和 CTestView 类, 如图 1-14 所示。在 Resource View 面板中显示所创建的资源, 主要包括 Accelerator、Dialog、Icon、Menu、String Table、Toolbar 和 Version 资源, 如图 1-15 所示。在 FileView 面板中显示源程序文件, 主要包括 Source Files、Header Files 和 Resource Files 文件等, 如图 1-16 所示。

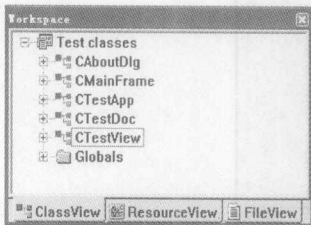


图 1-14 ClassView 面板

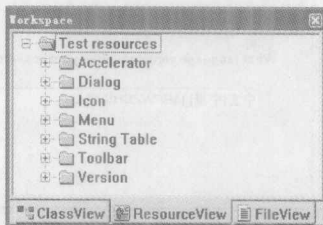


图 1-15 ResourceView 面板

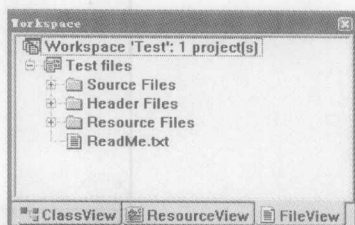


图 1-16 FileView 面板

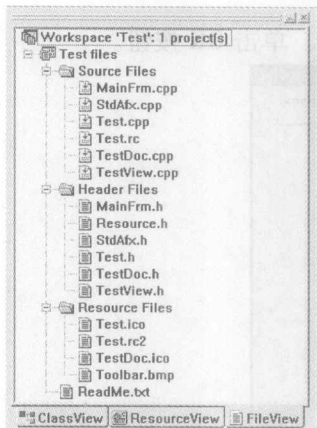


图 1-17 FileView 面板展开显示

从 ClassView 面板可以看出, CTestApp 是应用的主函数类, 用来处理消息。MFC 中的数据是存储在 CTestDoc 类文档中, 而结果的显示则是在 CTestView 类中, 即显示在 CMainFrame 类的客户区中。MFC 中的 Doc/View 结构用来将程序的数据本身和数据显示相互隔离, Doc 负责管理和维护数据本身, View 负责处理用户鼠标和键盘的操作。全部展开 FileView 面板后, 显示如图 1-17 所示的内容。特别需要指出的是, 本书后续的编程过程中一般使用的是 CTestView 类的头文件 TestView.h 和源文件 TestView.cpp。

(7) 单击工具条上的按钮, 如图 1-18 所示, 就可以直接编译、连接和运行程序。Test 程序运行结果如图 1-19 所示。至此, 尽管未编写一句代码, 但 Test 程序已经生成了一个可执行的 MFC 框架。后续的工作就是针对具体的设计任务, 为该框架添加程序代码和修改资源文件。

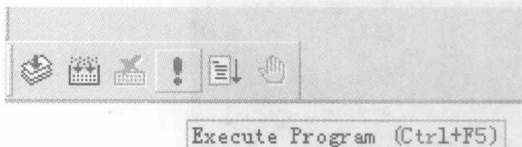


图 1-18 执行按钮

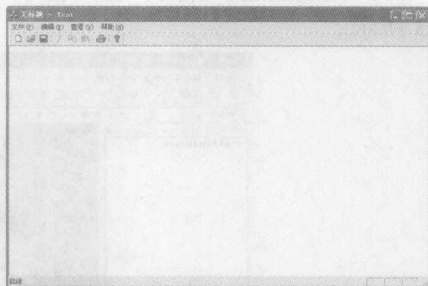


图 1-19 运行结果

(8) 在 Test.cpp 文件的 InitInstance() 函数中添加阴影部分的代码, 将程序运行界面设置为窗口最大化模式, 并且标题栏显示的文字内容为“案例: 金刚石图案”。

```
#01 BOOL CTestApp::InitInstance()
#02 {
#03     AfxEnableControlContainer();
```

```

#04      //Standard initialization
#05      //If you are not using these features and wish to reduce the size
#06      //of your final executable, you should remove from the following
#07      //the specific initialization routines you do not need.
#08      #ifdef _AFXDLL
#09          Enable3dControls();           //Call this when using MFC in a shared
                                           DLL
#10      #else
#11          Enable3dControlsStatic(); //Call this when linking to MFC
                                           statically
#12      #endif
#13      //Change the registry key under which our settings are stored.
#14      //TODO: You should modify this string to be something appropriate
           //such as the name of your company or organization.
#15      SetRegistryKey(_T("Local AppWizard-Generated Applications"));
#16      LoadStdProfileSettings(); //Load standard INI file options
                                           (including MRU)
#17
#18      //Register the application's document templates. Document tem-
           plates
#19      // serve as the connection between documents, frame windows and
           views.
#20      CSingleDocTemplate* pDocTemplate;
#21      pDocTemplate = new CSingleDocTemplate(
#22          IDR_MAINFRAME,
#23          RUNTIME_CLASS(CTestDoc),
#24          RUNTIME_CLASS(CMainFrame), //main SDI frame window
#25          RUNTIME_CLASS(CTestView));
#26      AddDocTemplate(pDocTemplate);
#27      //Parse command line for standard shell commands, DDE, file open
#28      CCommandLineInfo cmdInfo;
#29      ParseCommandLine(cmdInfo);
#30      //Dispatch commands specified on the command line
#31      if (!ProcessShellCommand(cmdInfo))
#32          return FALSE;
#33      //The one and only window has been initialized, so show and update
           it.
#34      m_pMainWnd->ShowWindow(SW_MAXIMIZE); //窗口极大化
#35      m_pMainWnd->SetWindowText("案例:金刚石图案"); //标题设置
#36      m_pMainWnd->UpdateWindow();
#37      return TRUE;
#38  }

```

程序解释:

- ◇ 第34行语句设置窗口显示状态为最大化。ShowWindow(int nCmdShow) 函数的参数 nCmdShow 常设为以下几个值: SW_SHOW, 窗口以当前的位置和尺寸显示; SW_MINIMIZE, 窗

口极小化显示; SW_MAXIMIZE, 窗口极大化显示。

◇ 第 35 行语句设置窗口标题文字为“案例: 金刚石图案”文本。

说明: 本书在以后的程序中约定, 阴影部分表示用户自己添加的代码, 其余部分是系统自动生成的代码。

2. 二维点类的设计

CP2
+x:double +y:double
+CP2(double, double)

图 1-20 二维点类类图

为了提高运算精度, 整体处理一个点的 x 和 y 值, 本案例设计了 CP2 二维点类, 其数据成员为 double 型的 (x, y) , 如图 1-20 所示。在类视图面板 ClassView 里选中 Test classes, 单击鼠标右键选择 New Class, 如图 1-21 所示。打开 New Class 对话框, 在 Class type 中选择 Generic Class 一般类, 在 Name 编辑框中输入类名 CP2, 如图 1-22 所示。单击 OK 按钮, 在类视图 ClassView 中添加了新类 CP2, 如图 1-23 所示。在文件视图面板的 Source Files 下 Visual C++ 向导自动添加了“P2.cpp”和“P2.h”文件, 如图 1-24 所示。双击打开“P2.h”文件, 二维点类的初始定义如图 1-25 所示。

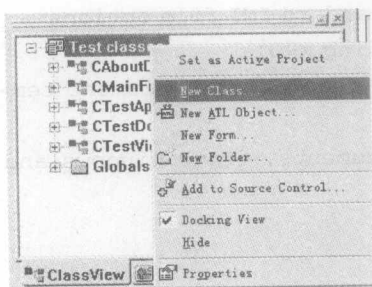


图 1-21 添加新类

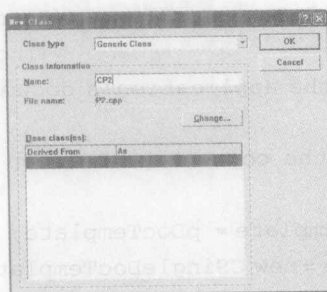


图 1-22 定义点类

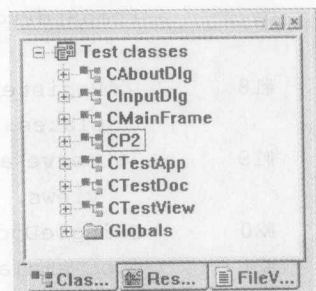


图 1-23 新添加的点类

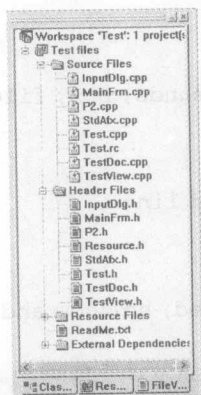


图 1-24 “P2.h”和“P2.cpp”文件

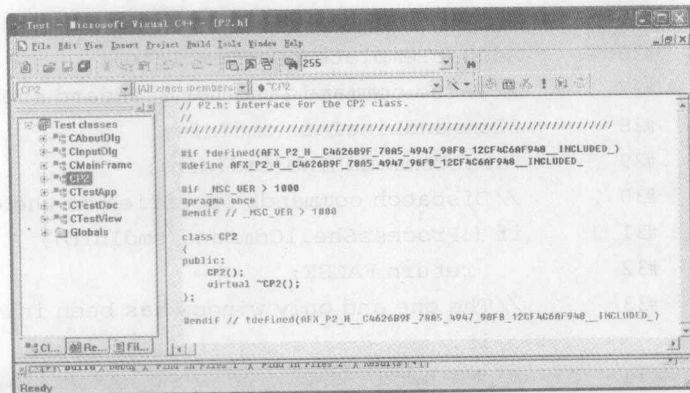


图 1-25 点类的初始定义

修改二维点类的初始定义, 添加 double 型变量成员 x 和 y 。

P2.h 的头文件为:

// P2.h: interface for the CP2 class.

```
#01 ///////////////////////////////////////////////////////////////////
#02 #if !defined(AFX_P2_H__C4626B9F_78A5_4947_98F8_12CF4C6AF948__
```

```

INCLUDED_)
#03 #define AFX_P2_H__C4626B9F_78A5_4947_98F8_12CF4C6AF948__INCLUDED_
#04 #if _MSC_VER > 1000
#05 #pragma once
#06 #endif // _MSC_VER > 1000
#07 class CP2
#08 {
#09 public:
#10     CP2 ();
#11     virtual ~CP2 ();
#12     CP2 (double, double);
#13     double x;
#14     double y;
#15 };
#16 #endif // !defined (AFX_P2_H__C4626B9F_78A5_4947_98F8_12CF4C6AF948__
        INCLUDED_)

```

程序解释:

◇ 第12行语句重载构造函数。

◇ 第13~14行语句设置 double 型二维点坐标。访问权限控制符为 public 的设置目的是为了在类外部能直接赋值。

P2.cpp 的源文件为:

```

#01 //P2.cpp: implementation of the CP2 class.
#02 //////////////////////////////////////
#03 #include "stdafx.h"
#04 #include "Test.h"
#05 #include "P2.h"
#06 #ifdef _DEBUG
#07 #undef THIS_FILE
#08 static char THIS_FILE[] = __FILE__;
#09 #define new DEBUG_NEW
#10 #endif
#11 //////////////////////////////////////
#12 //Construction/Destruction
#13 //////////////////////////////////////
#14 CP2::CP2 ()
#15 {
#16     this->x=0.0;
#17     this->y=0.0;
#18 }
#19 CP2::~~CP2 ()
#20 {
#21 }
#22 CP2::CP2 (double x0, double y0)
#23 {
#24     this->x=x0;
#25     this->y=y0;
#26 }

```

程序解释:

- ◇ 第 16~17 行语句在默认构造函数中初始化。
- ◇ 第 22~26 行语句定义重载构造函数。

3. 输入对话框类的设计

(1) 设计输入对话框界面

在资源视图面板 Resource View 中选择 Dialog, 单击鼠标右键, 选择 Insert Dialog, 如图 1-26 所示, 出现图 1-27 所示的初始对话框样式。删除 Cancel 按钮, 并利用图 1-28 所示的控件箱内的控件, 分别添加两个静态文本控件 Static Text 和两个编辑框控件 Edit Box, 将控件拖到适合的位置。在输入对话框的设计过程中使用了图 1-28 所示的 Visual C++ 的控件箱, 这可以在工具条空白处右击鼠标选择 Controls 使之显示, 如图 1-29 所示。

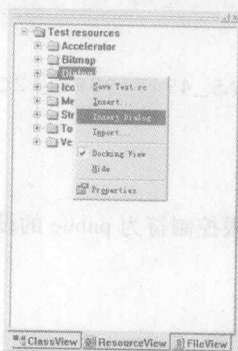


图 1-26 添加对话框图

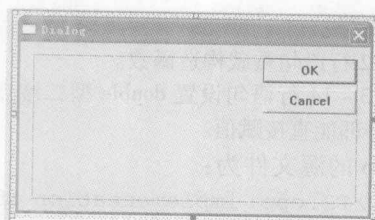


图 1-27 对话框的初始样式



图 1-28 控件箱图

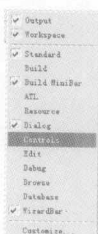


图 1-29 控件箱
选择快捷菜单

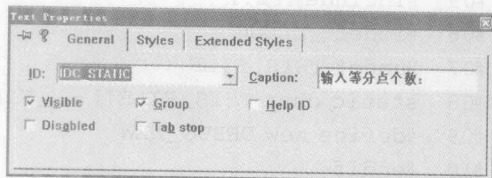


图 1-30 第 1 行的 Static 控件属性设计

添加控件后的输入对话框如图 1-5 所示, 下面分别设计各控件的属性。图 1-5 第 1 行的静态文本控件 Static 的 Caption 属性设置为: “输入等分点个数:”, 如图 1-30 所示。第 1 行的编辑框控件 Edit1 Box 的 ID 属性保持为 IDC_EDIT1, 如图 1-31 所示。第 2 行的静态文本控件 Static 的 Caption 属性设置为: “输入圆的半径:”, 如图 1-32 所示。编辑框控件 Edit2 Box 的 ID 属性保持为 IDC_EDIT2, 如图 1-33 所示。设置输入对话框的 Caption 属性为: “输入对话框”, 如图 1-34 所示。

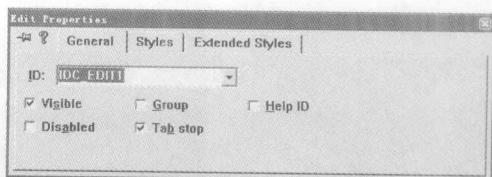


图 1-31 第 1 行的 Edit 控件属性设计

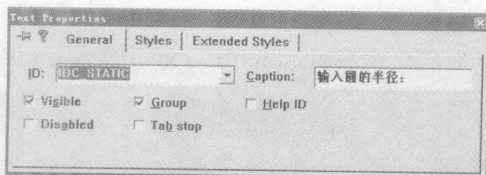


图 1-32 第 2 行的 Static 控件属性设计