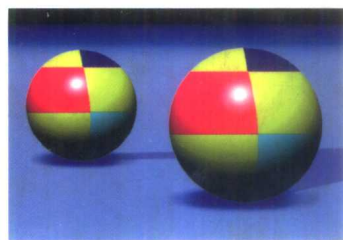


最新 VC++ 绘图程序设计技巧 与实例教程



- ▶ VC++ 绘图基础
- ▶ 三维工程图
- ▶ 二三维图形变换
- ▶ 绘画程序设计方法与实例

刘静华 王永生 著

- ▶ 程序动画设计
- ▶ 分数维图形及其应用
- ▶ 真实感图形程序设计



科学出版社

最新VC++绘图程序设计 技巧与实例教程

刘静华 王永生 著

科学出版社

2001

内 容 简 介

本书全面系统地介绍了计算机 Visual C++ 绘图原理、方法、技巧和实例,包括二维、三维与分维图形变换、三维消隐、绘图程序设计、工程图形(零件图、三视图、展开图、轴测图、透视图)绘制、分维图形绘制、解析曲面绘制、动画图形绘制、真实感图形绘制等内容,并配有成套程序与图形,反映了微机绘图技术的最新概貌。内容编排由浅入深,层次清晰;文字通俗易懂,便于自学;精选绘图源程序近百个,开发难度大,应用范围广。初学者可通过仿效、练习,来掌握绘图技术。有经验者可直接引用或稍加修改,得到满足自己需要的程序和图形,而事半功倍。

本书可作为大专院校师生计算机绘图教材或教学参考用书,亦可供工程技术人员从事计算机绘图工作使用,也适合各类自学人员。全部源程序汇于光盘,供读者选用。

图书在版编目(CIP)数据

最新 VC++ 绘图程序设计技巧与实例教程/刘静华,王永生著. —北京:科学出版社,2001

ISBN 7-03-009022-5

I. 最... II. ①刘... ②王... III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 05674 号

新 蕾 印 刷 厂 印 刷

科学出版社发行 各地新华书店经销

*

2001 年 3 月第一版 开本: 787×1092 1/16

2001 年 3 月第一次印刷 印张: 41 1/4

印数: 1—5 000 字数: 975 000

定价: 68.00 元(含盘)

(如有印装质量问题,我社负责调换 环伟)

前 言

计算机绘图技术发展越来越快,应用范围越来越广。为满足和促进计算机绘图教学、科研与应用工作的开展,在《最新C语言实用绘图程序与图形》(科学出版社,1996)的基础上,我们经过精心策划,进一步编著《最新VC++绘图程序设计技巧与实例教程》一书,其中包括作者多年的研究和工作成果。微机绘图技术许许多多,不可能也没必要全部取之。我们选择其中最精彩部分、应用范围最广的常用技术,通过由浅入深、层次清晰、通俗易懂、便于自学地编排,组成本书,呈现在您的面前。这是将计算机绘图技术大世界集于小世界之中的一本书,它领你走进计算机图形的美妙世界,任你描绘,任你创造,任你遨游。这多彩的世界给你快乐,给你智慧,给你财富。该书不可不读,当你学完以后,你会拥有一个神奇美好的计算机绘图世界。这时,微机图形世界就在你手中。

本书程序思路清晰,逻辑性强,模块结构,书写规范,可读性好。微机所绘图形能够产生惊人效果,令你拍案叫绝。如果读者使用的计算机设备支持Visual C++程序设计语言,我们极力鼓励你输入并运行书中的每一个程序,并把屏幕的输出结果与书中给出的图形相比较。这样做,你会通过仿效、练习,从而学会用Visual C++语言进行计算机绘图,最后达到开发绘图程序(或图形软件)的较高技术水平。

全书共分九章。第一章概述微机绘图技术基础,主要包括学习AppWizard、设备环境和GDI对象、Windows光栅操作和Exercises示例程序。第二章讲解绘图变换,即用户坐标到屏幕坐标变换、二维几何变换、二维裁剪、三维几何变换和投影变换。第三章讨论绘图程序设计步骤、绘图程序结构和绘图程序设计方法。第四章介绍二维工程图形,有花瓣图、螺旋图、圆形图、常用三种坐标曲线图、简单零件图和零件展开图,以及机械制图交切子程序。第五章,研究了近些年来出现的分数维图形,即龙图、C图、H图、龟图、曼德勃曼特图、朱莉娅图等。这些图形新颖、有趣,吸引很多人对其进行研究并应用之。现在它们被越来越多地用在造型环境中,用来设计壁纸、地板革图案和纺织品花样,也可将其映射到器皿上进行图案设计,效果极佳。第六章阐述三维工程图形,包括图形数据结构、消隐算法和立体图形绘制方法,同时配套实际例子。第七章论述一般空间曲线、组合体交线、浮动水平线算法和解析曲面绘制。第八章动画绘制,介绍了常用的几种动画技术,即异或动画、奇妙的调色板动画、块图形动画、帧动画和实时动画。第九章对绘制真实感图形技术进行了详细的讨论,包括光照与明暗技术概述、简单光照模型、整体光照模型和微机绘制真实感图形实例,拥有十分广泛的应用前景。

全书由刘静华、王永生主编。参加编写工作的有周林明、阎中标、陈超、郝国舜、王琦琪、刘佐英等。书中错漏之处难免,恳请读者指正。

作 者

2000年8月

目 录

前言

第一章 绘图基础	(1)
第一节 学习 App Wizard	(1)
一、利用 AppWizard 生成应用程序的框架	(1)
二、添加自己的代码并编译、运行	(5)
第二节 设备环境和 GDI 对象	(5)
一、Windows 设备环境	(6)
二、GDI 对象	(6)
第三节 Windows 光栅操作	(7)
第四节 Exercises 示例程序	(8)
一、常用绘图函数说明	(8)
二、完整的程序代码	(9)
第二章 绘图变换	(25)
第一节 用户坐标到屏幕坐标变换	(25)
一、窗口到视口的变换	(25)
二、实型值到整型值的变换	(26)
三、Y 坐标轴方向变换	(26)
第二节 二维几何变换	(27)
一、基本变换	(27)
二、复合变换	(31)
第三节 二维裁剪	(33)
一、线段裁剪	(34)
二、多边形裁剪	(36)
第四节 三维几何变换	(37)
一、比例变换	(38)
二、错切变换	(40)
三、对称变换	(41)
四、平移变换	(43)
五、旋转变换	(44)
六、逆变换	(46)
第五节 投影变换	(48)
一、平行投影	(49)
二、透视投影	(59)
三、变换实例	(73)
第三章 图形程序设计	(101)
第一节 绘图程序设计步骤	(101)

一、明确绘图程序功能	(101)
二、分析图形几何关系	(102)
三、写出绘图算法	(102)
四、编写绘图程序	(102)
五、上机调试运行、绘图	(103)
第二节 绘图程序结构	(136)
一、应用程序框架	(136)
二、程序设计的结构	(139)
第三节 绘图程序设计方法	(143)
一、图形层次结构和程序模块结构	(143)
二、面向对象程序设计	(144)
三、绘图子程序和主程序	(144)
四、绘图方法	(145)
五、绘图程序设计成功要点	(146)
第四章 二维工程图形	(149)
第一节 花瓣图、螺旋图和圆形图	(149)
一、花瓣图	(149)
二、螺旋图	(157)
三、圆形图	(173)
第二节 常用三种坐标曲线图	(180)
一、直角坐标曲线	(180)
二、极坐标曲线	(181)
三、参数坐标曲线	(182)
四、摆线图	(183)
第三节 机械制图交切子程序	(191)
一、求两已知线段交点子程序	(192)
二、求已知直线的平行线子程序	(194)
三、求两已知点内外分点子程序	(196)
四、求两已知圆交点子程序	(197)
五、求直线与圆交点子程序	(198)
六、求切点、切线子程序	(199)
七、求垂足、垂线子程序	(206)
第四节 简单零件图	(208)
一、轴类零件	(208)
二、螺纹	(218)
三、齿轮	(227)
四、弹簧	(230)
五、剖面线	(234)
六、参数化绘图	(247)
第五节 零件展开图	(253)
一、斜口圆管	(253)
二、正圆锥管	(261)
三、正交异径管	(262)

四、球罐	(272)
第五章 分数维图形	(276)
第一节 Fractal 图	(276)
一、分数维概念	(276)
二、Fractal 图	(277)
第二节 曼德勃罗特图形	(326)
一、曼德勃罗特图形	(326)
二、曼德勃罗特图形放大图	(332)
三、龟图	(334)
第三节 朱莉娅图形	(339)
一、朱莉娅图形	(339)
二、奇异引力线图	(345)
第四节 分形图应用	(357)
一、分形图的映射图	(357)
二、器皿图案设计	(379)
第六章 三维工程图形	(398)
第一节 图形数据结构	(398)
一、概述	(398)
二、图形几何信息与拓扑信息	(398)
三、二种基本图形数据结构	(399)
四、抽象数据结构与具体存储结构	(403)
第二节 消隐算法	(404)
一、概述	(404)
二、画家算法	(405)
第三节 立体图形	(413)
一、三视图	(413)
二、轴测图	(414)
三、透视图	(416)
四、单个凸多面体	(416)
五、多个多面体	(431)
第七章 解析曲面	(505)
第一节 一般空间曲线	(505)
一、球面三叶玫瑰线	(505)
二、圆柱螺线	(508)
三、圆锥螺线	(509)
第二节 组合体交线	(510)
一、截交线	(510)
二、相贯线	(517)
三、平面交线	(519)
第三节 浮动水平线算法	(522)
一、平面截线与浮动曲线	(523)
二、浮动曲线交点	(524)
三、浮动曲线边界与尖顶	(526)

四、浮动水平线法绘图实现过程	(528)
第四节 解析曲面	(529)
一、参数式曲面	(530)
二、函数式曲面	(532)
第八章 动画图形绘制	(576)
第一节 异或动画	(576)
一、异或动画原理	(576)
二、程序举例	(577)
第二节 奇妙的调色板动画	(593)
一、调色板动画的原理	(593)
二、生成调色板动画的步骤	(593)
三、程序举例及说明	(593)
第三节 帧动画	(605)
一、帧动画原理	(605)
二、制作帧动画	(605)
三、程序举例与说明	(605)
第四节 块动画	(621)
一、块动画的原理	(621)
二、块动画的制作过程	(621)
三、程序举例及说明	(621)
第五节 实时动画	(639)
一、实时动画原理	(639)
二、实时动画的优点	(639)
三、提高实时动画的速度	(639)
第九章 真实感图形绘制	(640)
第一节 简单光照模型	(640)
一、Phong 光照模型	(640)
二、表面法线的计算	(641)
三、Gouraud 明暗法	(641)
四、透明、阴影、纹理	(642)
第二节 整体光照模型	(643)
一、采用光线跟踪的整体光照模型	(643)
二、光线跟踪基础	(643)
三、光线与对象相交计算	(644)
四、光线跟踪算法	(645)
五、阴影的处理	(645)
六、图形反走样技术	(646)
七、彩色处理	(646)
第三节 微机绘制真实感图形	(646)
主要参考文献	(650)

第一章 绘图基础

Visual C++ 自诞生以来,一直是 Windows 环境下最主要的应用开发系统。Visual C++ 不仅仅是 C++ 语言的集成开发环境,而且与 Win32 紧密相连。所以,利用 Visual C++ 开发系统可以完成各种各样应用程序的开发,从底层软件到上层直接面向用户的软件都可以用 Visual C++ 来完成开发。由于 Windows 是基于 GUI 的操作系统,而 Visual C++ 提供了丰富的图形设备接口 (GDI) 函数,使得用 Visual C++ 开发 Windows 系统下的图形应用程序特别方便、简单。

本章主要是介绍利用 Visual C++ 开发图形应用程序的一些基础知识,以便为后面的学习作好准备。本章的结构分为三个部分:

- 利用 AppWizard 创建一个最简单的应用程序,了解在 Visual C++ 开发环境下生成应用程序所需步骤。
- 学习绘图的预备知识。
- 利用本章所学知识创建一个简单的图形程序。

第一节 学习 AppWizard

AppWizard 是 Visual C++ 开发系统中的一个工具,利用它可以快速地生成一个应用程序的基本框架;然后,你的任务就是在基本框架内添加相应的代码,以完成你所需的工作。

AppWizard 可以生成三种不同类型的程序:单文档界面 (SDI) 的应用程序、多文档界面 (MDI) 的应用程序和基于对话框的应用程序。从用户的角度看,SDI 应用程序只有一个窗口。如果某 SDI 应用程序依赖于磁盘文件形式的“文档”,则每次只能读进一个文档,如 Windows 系统中的记事本。MDI 应用程序则可以拥有和不同文档相对应的多个子窗口,如 Microsoft Word 就是一个很好的例子。基于对话框的应用程序的实质就是一个对话框。

一、利用 AppWizard 生成应用程序的框架

下面是利用 AppWizard 创建“Hello .World!”应用程序的步骤:

(1) 从 Developer Studio 的 File 菜单项中选择 New,然后在 New 对话框中单击 Project 标签,如图 1-1 所示。

确保选择 MFC AppWizard(exe).然后在 Location 编辑框中输入存放工程文件的路径 f:\LearningVC\,再在 Project Name 编辑框中输入工程名字 Hello,而后单击 OK 按钮。

下面将出现一系列的对话框,AppWizard 将按照你在这些对话框中所作的选择生成相应的代码。

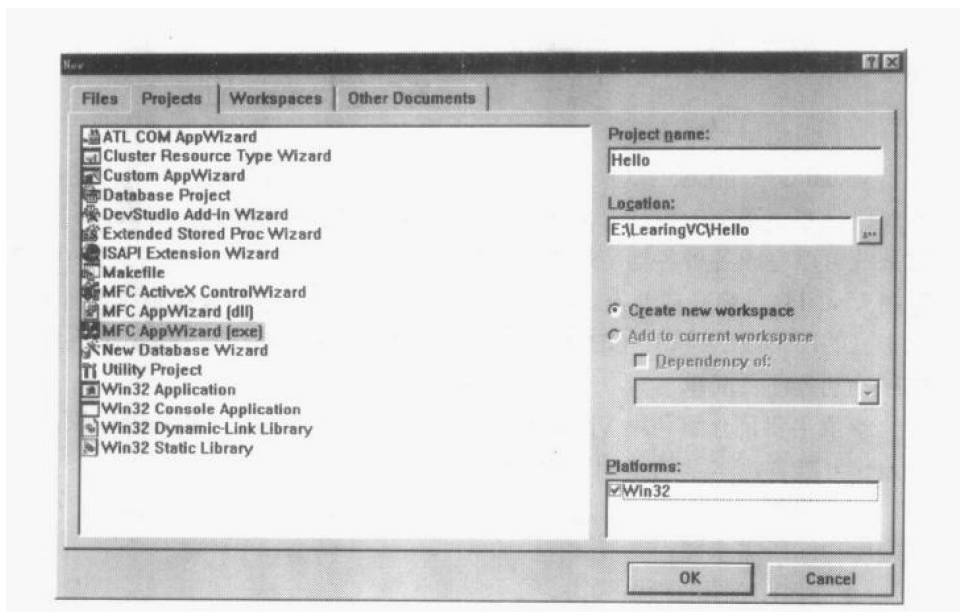


图 1-1 New 对话框

(2) 在 MFC AppWizard-step1 (第一步) 对话框中选择“Single Document”选项, 语言选择“中文”, 如图 1-2 所示。

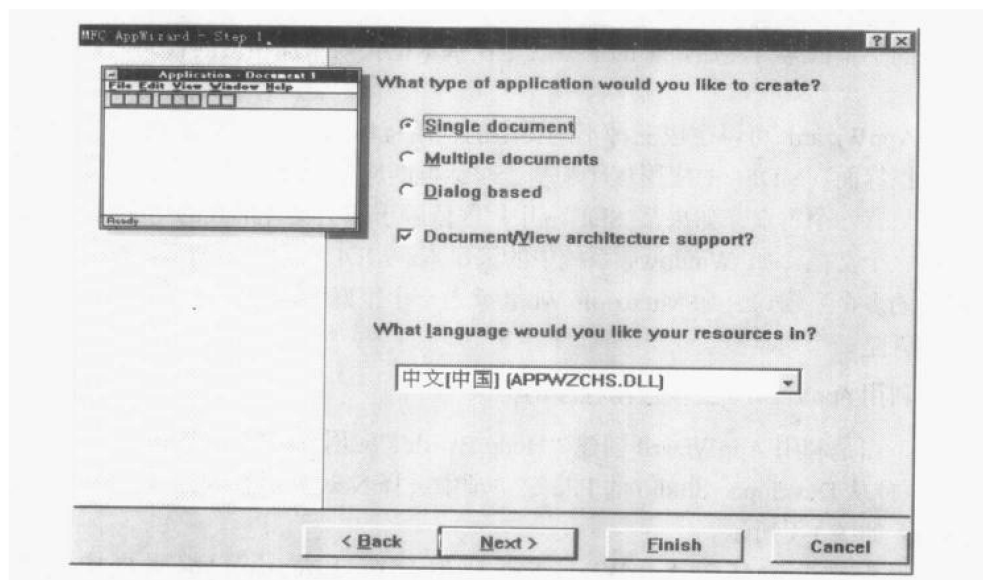


图 1-2 MFC AppWizard 第一步

(3) 在 MFC AppWizard-step 2 of 6 (第 2 步)对话框和 MFC AppWizard-step 3 of 6 (第 3 步)对话框中接受缺省的选项, 见图 1-3, 图 1-4 所示。

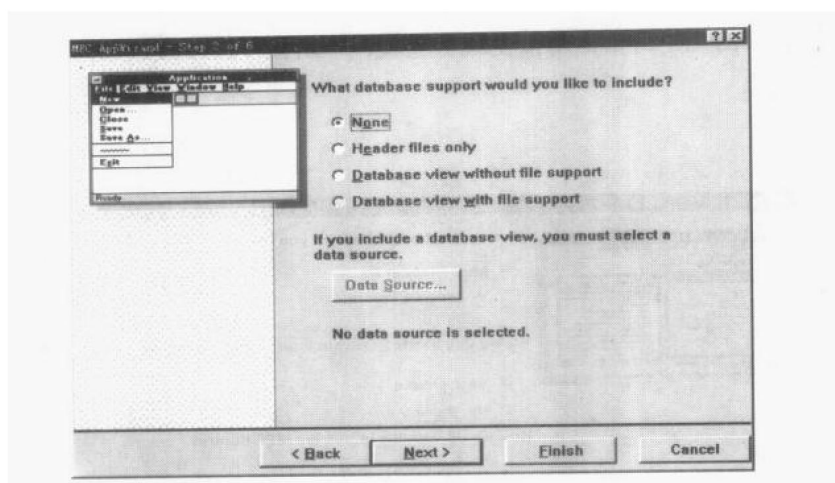


图 1-3 MFC AppWizard -Step 2 of 6

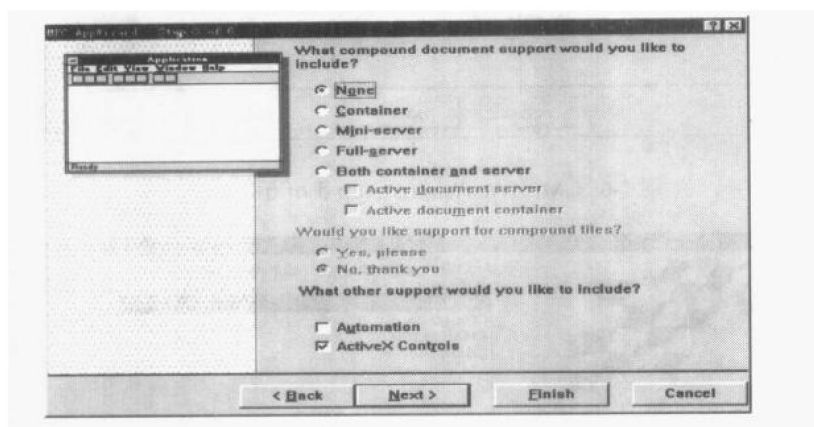


图 1-4 MFC AppWizard -step 3 of 6

(4) 因为我们这个例子程序不需要打印功能，所以在第四步对话框中将“Printing and print preview”选项去掉，如下图 1-5 所示。

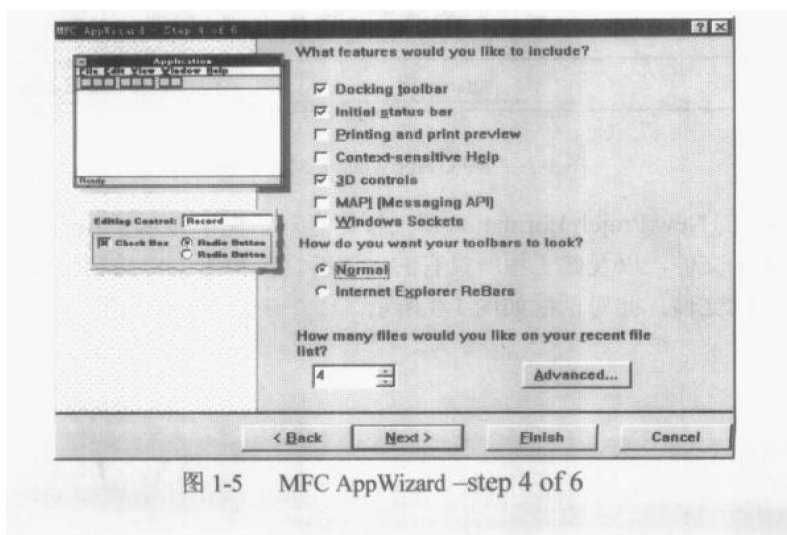


图 1-5 MFC AppWizard -step 4 of 6

(5) 在接下来的几个对话框中接受缺省选项，不用作任何修改，如图 1-6，图 1-7 所示。

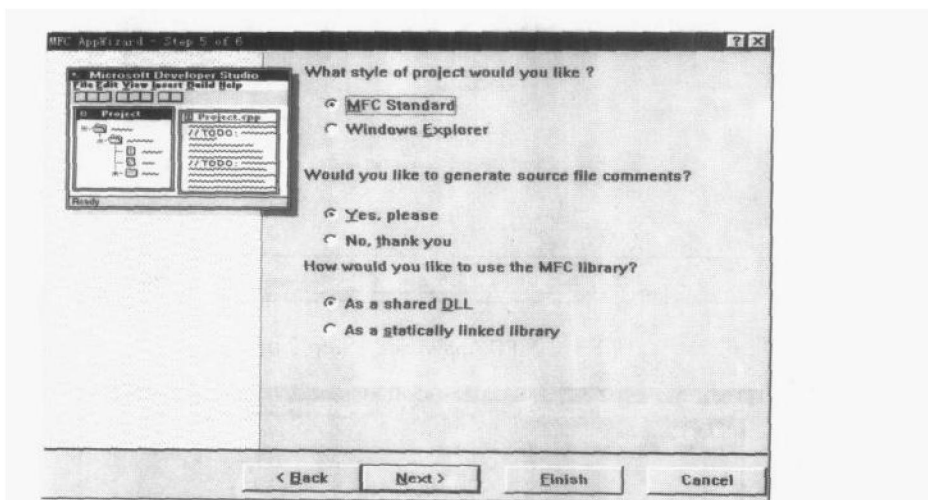


图 1-6 MFC AppWizard-step 5 of 6

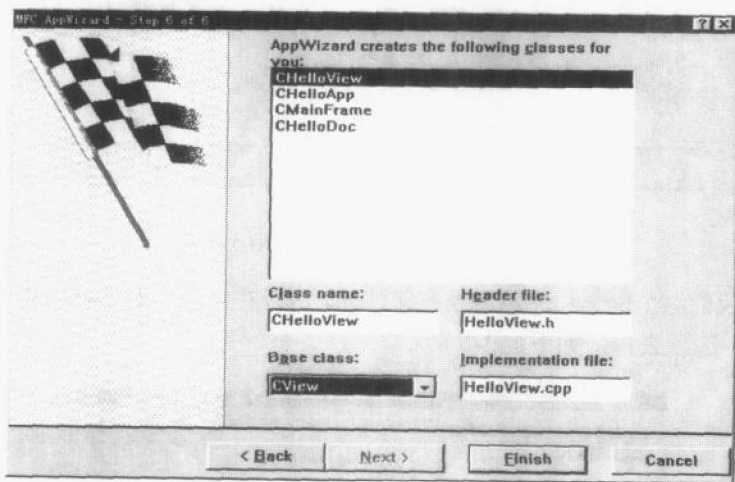


图 1-7 MFC AppWizard-step 6 of 6

(6) 在最后的 New Project Information 新工程信息对话框中显示了 AppWizard 将要给你生成的一些类和文件，以及此工程所具有的一些特性，单击 OK 按钮，AppWizard 就会为你生成相应的文件。此对话框如图 1-8 所示。

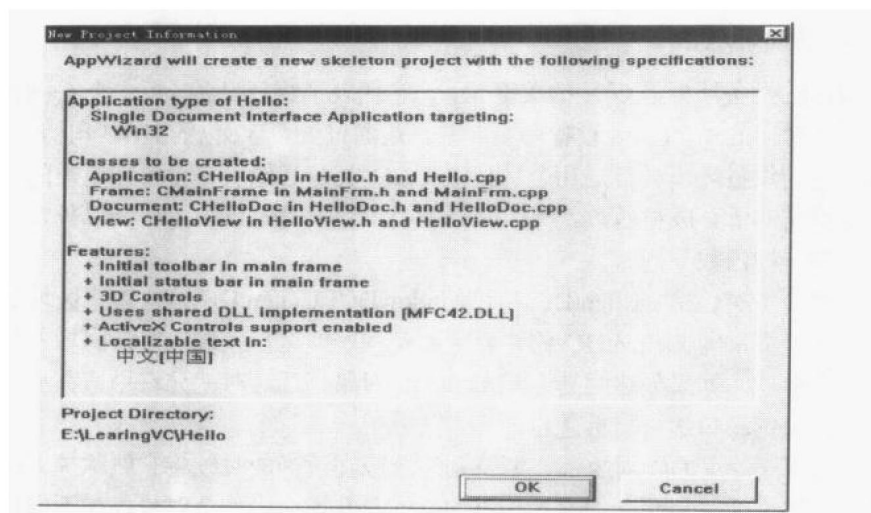


图 1-8 New Project Information

二、添加自己的代码并编译、运行

在生成的 `HelloView.cpp` 文件中找到 `OnDraw` 函数，在此函数中加入以下的一行代码，它将在屏幕上显示一行文字“Hello,World!”。

```
Void CHelloView:: OnDraw(CDC *pDC)
{
    CHelloDoc *pDoc=GetDocument();
    ASSERT_VALID(pDoc);
    pDC->TextOut(100,100,"Hello,World!");
}
```

然后，选择 **Build** 菜单中的 **Build Hello.exe** 菜单项来编译、连接所生成的代码，如果此过程没有出错，则在 `E:\LearningVC\Hello` 目录下将会建立一个新的 **Debug** 目录，在该目录下将生成一个可执行文件 `Hello.exe`。

接着选择 **Build** 菜单中的 **Execute Hello.exe** 菜单项，试着运行一下这个程序。从上面的这个例子可以看出，借助于 AppWizard，创建一个标准的 Windows 应用程序是件非常容易的事。但 AppWizard 仅仅是为你生成了一个框架，具体的功能实现还得你自己去做。到目前为止，我们的这个示例程序功能十分有限，它只能在屏幕上显示一行文字。下面部分将介绍用 Visual C++ 绘图所需的预备知识。

第二节 设备环境和 GDI 对象

在 Visual C++ 环境中绘图需要做两件事情：首先创建一个 GDI 绘图对象，并将它放入设备环境中；然后，调用设备环境类中的各种绘图函数开始绘图。

一、Windows 设备环境

Windows 的设备环境是 GDI 的关键元素,它代表了物理设备。每一个 C++ 设备环境对象都有与之对应的 Windows 设备环境,并且是通过一个 32 位的 HDC 句柄来标识。

Windows 的设备环境类有好几个,其中 CDC 类是其他所有设备环境类的基类,它包含了绘图所需的所有成员函数。其他的派生类,除了 CmetaFileDC 类以外,都仅仅是构造函数,和析构函数不同。

显示设备环境类包括 CClientDC 和 CWindowDC。CclientDC 是在客户区域绘图时对应的设备环境类,即 CclientDC 的映射区域是程序的客户区域。客户区域不包括标题栏、边框和菜单栏。如果创建的是 CWindowDC 对象,那这时就允许用户在显示器的任何地方绘图,包括窗口边框、标题栏等等。

通常创建了一个 CDC 对象后,要特别注意应在适当的时候将它删除掉。要保证设备环境对象能够被适时地删除掉,最简单的办法是在堆栈中构造对象,如下面的代码所示:

```
void CtestView::OnMyDraw()
{
    CDC *pDC=GetDC();
    ..... //用户自己的绘图代码
    ReleaseDC(pDC);
}
```

设备环境类中有很多绘制函数,可用来绘制各种各样的图形,在本章的第四节将举例说明。

二、GDI 对象

Windows 的 GDI 对象是通过 GDI 类来表示的,所有的 GDI 对象类的基类是 CgdiObject 类。下面给出了 Visual C++ 中的 GDI 类的列表:

- CBitmap——位图,是一种位矩阵,每一个显示像素都对应于其中的一个或多个位。我们可以利用位图来表示图像,也可以用它来创建刷子。
- CBrush——刷子,定义了一种位图形式的像素,利用它可对区域内部填充颜色。
- CFont ——字体,是一种具有某种风格和尺寸的所有字符的完整集合,它常常被当作资源保存在磁盘中,其中有一些还依赖于某种设备。
- CPalette——调色板,是一种颜色映射接口,它允许应用程序在不干扰其他应用程序的前提下,可以充分利用输出设备的颜色描绘能力。
- CPen——笔,是一种用来画线及绘制有形边框的工具,我们可以指定它的颜色及厚度,并且可以指定它的类型。
- CRgn——区域,是由多边形、椭圆或二者组合形成的一种范围,可以利用它来进行填充、裁剪以及鼠标击中测试。

我们一般不创建 CgdiObject 类的对象,而是去创建它的派生类的对象。有些 GDI 派生类(如 Cpen 和 Cbrush)的构造函数允许我们提供足够的信息,从而一步即可完成对象的创建工作;而另外一些类(如 Cfont 和 CRgn)的对象的创建则需要两步才能完成。

我们需要首先调用它的默认构造函数来构造对象，然后再进一步调用相应的创建函数，如 `CreatFont` 或 `CreatePolygonRgn` 等。

GDI 对象创建成功后，就可以调用 CDC 类的 `SelectObject` 函数将对象放入设备环境中，然后就能使用这个对象进行绘图。当不再需要这个对象或程序要关闭时，应将所创建的对象删除。为了删除一个 GDI 对象，必须先将它从设备环境中分离出来。那么如何将它从设备环境中分离出来呢？事实上，CDC 类的 `SelectObject` 函数在将 GDI 对象放入设备环境的同时，它还返回了一个指向上一次被选对象的指针。因此，一个比较简单的方法是在将自己创建的 GDI 对象放入设备环境中时，将原来的 GDI 对象也保存起来，当任务完成后，再将原来的对象恢复，这样也就将自己的对象删除了。

下面给出一个具体的例子：

```
void CtestView::OnDraw(CDC *pDC)
{
    Cpen newpen(PS_SOLID,1,RGB(0,0,255));
    Cpen *old=pDC->SelectObject(&newpen);
    pDC->MoveTo(10,10);
    pDC->LineTo(40,60);
    pDC->SelectObject(old); //此时 newpen 就被分离出来了
} //当这段程序退出时 newpen 就被自动地删除了。
```

Windows 的 GDI 提供了一种“硬件独立”的颜色接口，应用程序只须提供“绝对的”颜色代码，GDI 就会将该代码以合适的颜色或颜色组合映射到计算机的视频显示器上。每一种 Windows 颜色都是通过“红”、“绿”、“蓝”三原色的组合而成，如 `RGB(255, 0, 0)` 是红色，`RGB(255, 255, 0)` 是黄色，而 `RGB(255, 255, 255)` 是白色。其中 `RGB` 是一个宏，它用来将三种颜色的分量组合成一个 Windows 颜色。

第三节 Windows 光栅操作

在屏幕上无论绘制何种图形，其实质就是为单个像素设置颜色。下面我们从一个黑白背景讲起。

在显示系统中，最常用的背景颜色白色是将 24 位都置为 1（或 `TRUE`）来表示，它是真正地表示颜色出现的最大值。和白色相对照的颜色当然就是黑色，它是将 24 位都置为 0 来表示，就是完全的没有颜色。至于其他任何一种颜色，它的 24 位都不可能全是 1 或 0，而是有些位是 1，有些位是 0。显示图形时，就是用绘图设备上下文中的现存颜色的每一位去替换每一像素的每一位，这样屏幕上就会出现你所需的颜色。

上面的这个操作过程是一个简单的位替换过程，如果我们不是这样执行替换，而是将颜色的每一位和像素的每一位进行或、与等操作，会出现什么情况呢？很明显，像素最终的颜色既不会保持不变，也不会是绘图设备上下文的现存颜色，而是另外一种颜色。这种操作也被称为光栅操作（或 `ROPs`）。下面我们将详细介绍 Windows 中的光栅操作代码。

在 Windows 中有两组光栅操作代码：ROP 代码和 ROP2 代码。CDC::SetROP2() 函数中的光栅操作是较为简单的二进制光栅操作，或称 ROP2 代码。之所以这样称呼它们是因为它们仅仅定义两个像素，即一个画笔像素、一个目的像素，最终结果是二者相结合得到的。表 1-1 列出了所有的 ROP2 代码，缺省的绘图方式是 R2_COPYPEN。

表 1-1 16 个 ROP2 的光栅操作代码

R2_BLACK	像素总是黑色
R2_WHITE	像素总是白色
R2_NOP	像素保持不变
R2_NOT	像素是屏幕颜色的反
R2_COPYPEN	像素是画笔颜色
R2_NOTCOPYPEN	像素是画笔颜色的反
R2_MERGEENNOT	像素是画笔颜色和屏幕颜色的反的结合
R2_MASKENNOT	像素是画笔和屏幕颜色“与”的结合
R2_MERGEENNOTPEN	像素是屏幕颜色和画笔颜色反的结合
R2_MASKNOTOPEN	像素是屏幕颜色和画笔颜色反的与
R2_MERGEEN	像素是屏幕颜色和画笔颜色的结合
R2_NOTMERGEEN	像素是 R2_MERGEEN 颜色的反
R2_MASKPEN	像素是画笔颜色和屏幕颜色与的结合
R2_NOTMASKPEN	像素是 R2_MASKPEN 颜色的反
R2_XORPEN	像素是画笔颜色和屏幕颜色的异或的结合
R2_NOTXORPEN	像素是 R2_XORPEN 颜色的反

第四节 Exercises 示例程序

一、常用绘图函数说明

常用的绘图函数有如下这些：

```
CPoint CDC::GetCurrentPosition();
```

返回当前的光标位置。

```
CPoint CDC::MoveTo(int x,int y);
```

移动光标到指定点处，并返回上一次的光标位置。

```
BOOL CDC::LineTo(int x,int y);
```

从当前光标处开始画线到指定点处。如成功则返回真，否则返回假。

```
BOOL CDC::Arc(LPCRECT lpRect,POINT ptStart,POINT ptEnd);
```

画一个椭圆弧。LpRect 是一个矩形方框，确定了这个椭圆；ptStart 和 ptEnd 则确定了椭圆弧的起点和终点。给定的起点和终点到椭圆心的连线与椭圆的交点是实际上的椭圆弧的起点和终点。


```
BOOL CDC::PolyLine(LPPOINT lpPoints,int nCount);
```

将给定的点序列用折线连起来。LpPoints 就是存放这些点的数组。

```
BOOL CDC::Ellipse(int x1,int y1,int x2,int y2);
```

以当前的笔画一个椭圆，并用当前的刷子填充椭圆的内部。x1 和 y1 确定了矩形的左上角，x2 和 y2 确定了矩形的右下角。椭圆的圆心是矩形的中心，当矩形是正方形时画出的就是一个圆。

```
BOOL CDC::Rectangle(int x1,int y1,int x2,int y2);
```

以当前的笔画一个矩形，并用当前的刷子填充矩形的内部。x1 和 y1 确定了矩形的左上角，x2 和 y2 确定了矩形的右下角。

下面用上述的函数绘制一些简单图形，最后的运行结果如图 1-9 所示。

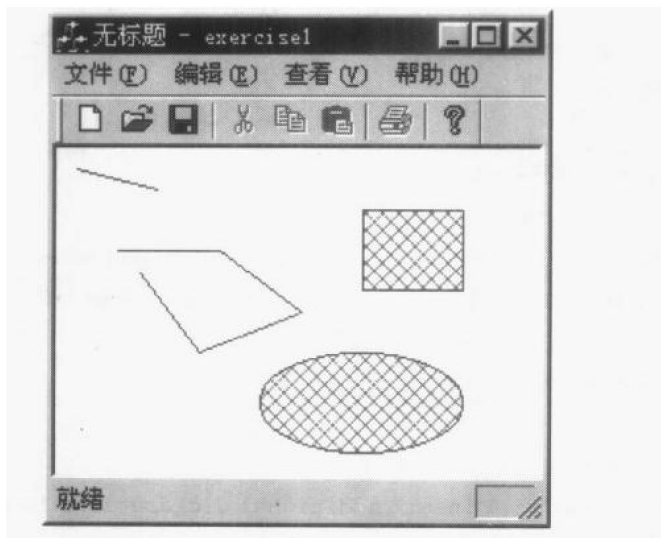


图 1-9 程序运行结果

二、完整的程序代码

```
// exercisel.h : main header file for the EXERCISE1 application
#ifdef !defined(AFX_EXERCISE1_H__3B895427_4340_11D3_9F49_00400565CEC3__INCLU
DED_)
#define AFX_EXERCISE1_H__3B895427_4340_11D3_9F49_00400565CEC3__INCLUDED_

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
#ifndef __AFXWIN_H__
    #error include 'stdafx.h' before including this file for PCH
#endif
#include "resource.h"          // main symbols

// CExerciselApp:
```