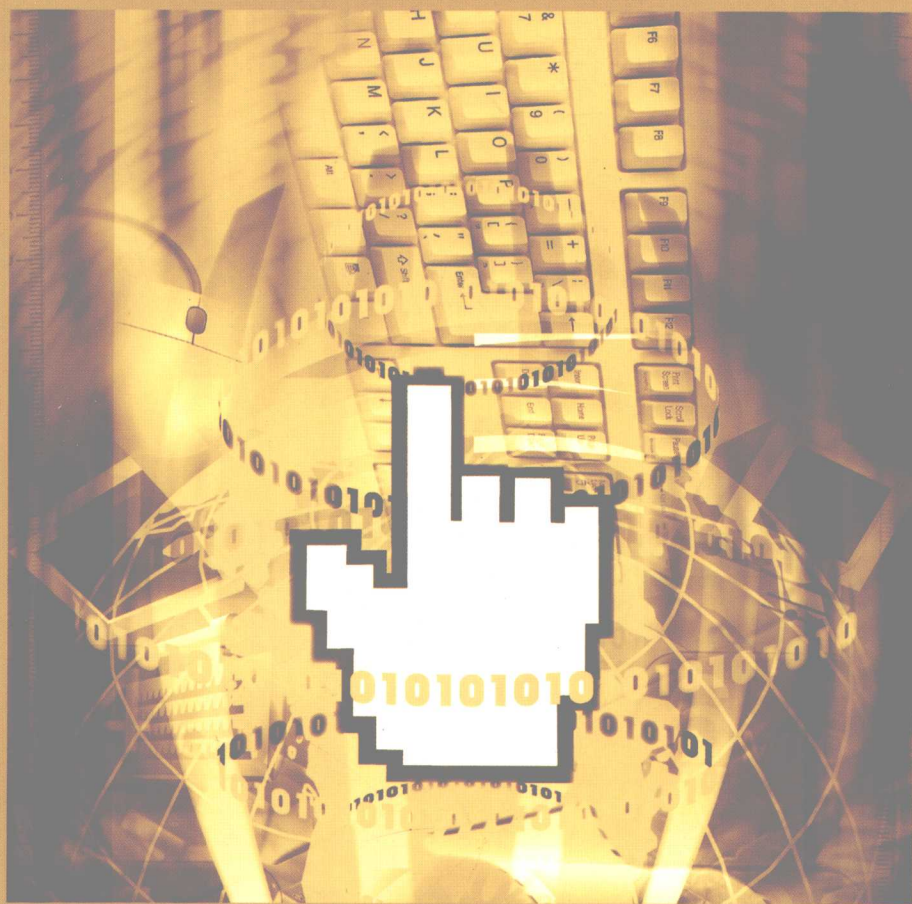




全国高校素质教育教材研究编审委员会审定

计算机图形与图像学 应用基础

赵子玉 宋焕生 王国强 编著



兵器工业出版社

TP391.41

Z348.1

7

内容简介

全国高校素质教育教材研究编审委员会审定

计算机图形与图像学 应用基础

赵子玉 宋焕生 王国强 编著

TP391.41
Z348.1

兵器工业出版社

内容简介

本书理论联系实际,通过 Visual C++ 语言编程实例,讲解图形图像学的基本原理、算法及实现。本书的特点是便于自学,以循序渐进的方式使读者逐步提高计算机软件编程能力。本书可以作为高等院校师生图形图像学基础应用的教学教材,也可以作为广大图形图像、游戏编程爱好者的自学资料,同时也可供相关技术人员阅读。通过本教材的学习,可为读者从事软件开发和课题研究工作奠定基础。

图书在版编目 (CIP) 数据

计算机图形与图像学应用基础/赵子玉,宋焕生,
王国强编著. —北京:兵器工业出版社,2007.12
ISBN 978-7-80248-002-5

I. 计… II. ①赵…②宋…③王… III. 计算机图形学
IV. TP391.41

中国版本图书馆 CIP 数据核字 (2007) 第 184145 号

出版发行:兵器工业出版社
发行电话:010-68962596 68962591
邮 编:100089
社 址:北京市海淀区车道沟 10 号
经 销:各地新华书店
印 刷:北京银祥福利印刷厂
版 次:2007 年 12 月第 1 版第 1 次印刷

责任编辑:林利红
封面设计:张骥年
责任校对:郭 芳
责任印制:赵春云
开 本:787×1092 1/16
印 张:19.25
字 数:560 千字
定 价:39.80 元

(版权所有 翻印必究 印装有误 负责调换)

前 言

在现代社会中, 计算机软件编程已成为高等院校的毕业生在各行各业求生存、谋发展的一项基本技能。在高年级大学生、研究生的课程开设中开设计算机软件编程类必修课和选修课(如开设计算机图形图像学应用基础课程), 训练学生的计算机软件编程能力是非常必要的。

当前, 对于高年级大学生、研究生的计算机图形图像学的教学, 特别需要一本便于自学、具有详尽程序实例的图形图像学基础应用方面的教材。本书通过 Visual C++ 6.0 程序实例, 由简到繁, 以循序渐进的方式讲授图形图像学方面的基础知识, 使读者掌握二维、三维图形变换, 小平面实体建模, 真实感图形生成, 图像变换软件的编程方法。与现有的国内外同类教材相比, 本新编教材的特点为:

(1) 目前, 在 Visual C++ 环境中已经集成了生成点、线、面、体和真实感图形的 Windows 库函数, 如 API, MFC, OpenGL, Direct3D 等, 介绍这些库函数使用方法的教材也是层出不穷, 但是, 以直接写硬件的画点子程序和图形生成算法为基础, 进行画线、二维和三维几何变换动画演示、小平面实体建模、真实感图形生成、图像变换, 并给出全部 Visual C++ 源程序的教材现在还很少见到。

(2) 本教材为了加快图形显示速度, 在 Windows 系统 Visual C++ 6.0 环境下, 采用 Microsoft 的 DirectDraw (直接写硬件显存) 技术编制了画点子程序, 其余的输出汉字点阵、画线、输出西文字符、画实体小平面、生成真实感图形、图像变换等都是根据图形生成算法或图像格式, 调用画点子程序完成的。本教材附录中还给出了 DOS 系统下使用 VESA VBE 标准枚举所有的显示模式的 Borland C 源程序并进行了详尽注释。

参加本书编写工作的有上海交通大学赵子玉、王建伯, 长安大学宋焕生、王国强, 由赵子玉汇总和整理。书中内容虽为作者多年教学和科研工作的总结与体会, 但由于作者水平有限, 书中难免存在缺点和不足, 殷切希望广大读者批评指正。

书中原程序可通过网络下载, 网址: <http://www.bancuo.com/download/jsjtt/index.htm>, 也可与编者联系索取, 编者 E-mail: zhaoziyu@sju.edu.cn。

编 者

2007 年 11 月

目 录

第 1 章 概论	1
第 2 章 Visual C++ Windows 编程基础	5
2.1 Visual C++6.0 开发环境	5
2.2 Wizard 创建 Win32 应用程序	7
2.3 Windows 编程基础	9
第 3 章 画点与画线	16
3.1 显示分辨率与颜色	16
3.2 DirectDraw 显示模式及画点、画线	18
3.3 画点的子程序	27
3.4 汉字点阵显示	32
3.5 画线	35
第 4 章 二维、三维图形变换	44
4.1 几何变换的数学基础	44
4.2 二维图形几何变换	45
4.3 三维图形几何变换	63
4.4 基本变换子程序设计	66
4.5 投影变换	75
第 5 章 实体建模理论和数据结构	78
5.1 三维建模程序运行结果示例	78
5.2 数学模块和图形界面模块	84
5.3 顶点和小平面对数	96
5.4 物体数据文件(.dat)格式	96
5.5 装入和存储顶点和小平面对数——将 .dat 文件从磁盘读入内存中,或反之	97
5.6 画一个线框小平面对	101
5.7 在屏幕上画一个实体小平面对	101
第 6 章 给画面增加物体	103
6.1 初始化物体列表	109
6.2 对物体列表 ObjList 赋值	109
6.3 给画面增加边缘反射镜面	110
6.4 从磁盘文件 *.SCN 给画面增加物体	110
6.5 从一个函数给画面增加物体	111



第 7 章 在屏幕上排序和显示物体	112
7.1 对放置到屏幕上的物体进行排序	120
7.2 在屏幕上放置物体 (PlaceObjectOnScreen)	121
7.3 显示物体和反射 (DisplayObjectsInScene)	122
第 8 章 三维建模程序	123
8.1 观测者和光源向量	131
8.2 装入描述文件 Model. des	132
8.3 仿射变换	132
8.4 求出小平面的法向向量	132
8.5 光照模型	133
8.6 检查小平面是否可见	135
8.7 反射物体的屏幕缓冲区 Refl	135
8.8 获得小平面屏幕坐标	135
8.9 在屏幕上画一个实体小平面	136
8.10 显示画面文件 .SCN	136
8.11 HelloModel. cpp 程序	137
第 9 章 生成物体的数据库	152
9.1 增加顶点	154
9.2 生成顶点前的初始化	154
9.3 生成圆锥体和金字塔的数据文件	155
9.4 为一个圆柱体生成数据文件	157
9.5 为一个半球生成数据文件	159
9.6 为一个圆球生成数据文件	162
9.7 为方程式生成数据文件	165
9.8 为一个环生成数据文件	168
9.9 为旋转实体生成数据文件	170
第 10 章 画面文件	175
10.1 CubePlan. SCN	175
10.2 SphrWall. SCN	175
10.3 FourCol. SCN	177
10.4 Well. SCN	178
10.5 Shapes. SCN	181
10.6 SolOfRev. SCN	183
10.7 StakTors. SCN	184
10.8 PlotEqn1. SCN~PlotEqn4. SCN	185
第 11 章 光线跟踪基础	187
11.1 光线跟踪 (Ray-trace) 原理概述	187
11.2 定义画面的格式文件 .RT	188



第 12 章 光线跟踪程序	217
12.1 光线跟踪程序概述	229
12.2 装入一个 .RT 文件	233
12.3 扫描画面	245
12.4 跟踪光线	246
12.5 生成纹理的表面、确定颜色	259
12.6 结束光线跟踪	266
第 13 章 光线跟踪生成的 .CPR 文件的图像转换	268
13.1 光线跟踪 .CPR 文件转 256 色粗格式图像数据文件	268
13.2 粗格式图像数据转 256 色或 16M 色 .BMP 文件	275
附录 A VESA VBE 显示模式设置	285
A.1 显示模式枚举程序 VbeTest.c	285
A.2 DOS 视频中断	294
A.3 在 VESA 显示模式下画点	296
参考文献	299

第1章 概 论

计算机图形学 (Computer Graphics) 是研究怎样用计算机生成、处理和显示图形的一门学科。它是计算机科学领域中的一个重要而又年轻的学科, 它随着计算机硬件特别是图形显示设备的发展而逐渐产生、发展起来。平面上指定范围内可见点阵的集合统称为图像, 如人们常见报纸上印刷出的照片就是一种典型的例子。图形是图像的一种特例, 它仅仅是一种能用各种几何参数 (如各种形式的线段) 与属性参数进行有效描述的图像。计算机图形学的研究对象可以概括成: 点、线、面、体、场的数学构造方法与其图形显示, 以及它们随时间的变化情况 (如动画)。目前, 计算机图形学的主要应用领域有:

- (1) 计算机辅助设计与制造 (CAD/CAM)。用来进行建筑工程、机械结构和部件的人机交互设计和布局。
- (2) 绘制图表。用来绘制三维地貌图、矿藏分布图、气象图、人口分布图, 以及各类等值线、面图等。
- (3) 仿真与动画。用来制作动画片、游戏、商业广告及造型设计。
- (4) 计算机辅助教学。用动画或仿真方法进行学习、研究, 可以把抽象的概念形象化, 使学生易于理解和接受。
- (5) 过程控制, 办公自动化, 分析计算, 医学诊断, 军事训练等。

1962 年, 麻省理工学院的一位博士研究生 Ivan E. Sutherland 发表了一篇题为“画板 (Sketchpad): 人一机图形通信系统”的博士论文, 提出了用键盘、光笔作为人一机交互通信手段, 利用计算机完成画点、线及简单图形的方法, 引入了许多直到今天仍在使用的计算机图形学概念, 并实现了图形的旋转、平移、放大、缩小等功能。标志着计算机图形学的正式诞生。

20 世纪 70 年代, 由于随机扫描图形显示器件的广泛应用, 掀起了第一次计算机图形应用的高潮。但在这一时期, 所用的随机扫描图形显示器只能用于显示线条图形, 无法显示有灰度变化的图形。所以, 这期间计算机图形的应用与研究主要是针对线框图形的 CAD/CAM。

20 世纪 80 年代初期以来, 由于廉价的大规模集成电路的出现和技术成熟的光栅扫描显示器的广泛应用, 出现了计算机图形学的第二次高潮。这种显示器采用电视光栅扫描方式, 将图形按光栅序列扫描到屏幕上, 图形由像素点 (Pixel) 组成。采用这种显示器, 即使显示器的图形很复杂也不会出现闪烁现象。特别是这种显示器的灰度层次多, 色彩丰富, 适用于显示逼真图形。

随着计算机和集成电路技术日新月异的发展, 显示适配器也得到了改进。第一个出现的个人计算机显示适配器是 1982 年由 IBM 公司推出的, 它是一种单色的文本方式适配器 (MDA), 不能显示图形和颜色。随后不久, Herclues (大力神) 计算机公司推出了 Her-



clues 单色图形显示器,能以黑白两色显示图形。MDA 之后,出现了彩色适配器 CGA,但它提供的彩色图形方式比较原始,1985 年 IBM 公司又推出了增强性图形适配器 EGA,EGA 提供的彩色图形功能大大超过了 CGA。1987 年,IBM 公司在 EGA 的基础上推出了视频阵列图形显示器 VGA。VGA 作为未来的显示标准赢得了广泛的市场。许多芯片厂家很快就推出了 IBM VGA 的仿制产品,同时,他们又对产品的性能加以改进,使得其产品具有比 IBM VGA 更高的分辨率及更丰富的色彩,人们通常把这些产品统称为 Super VGA。目前,Super VGA 已成为最普及的图形显示器。

不同的 Super VGA 制造商之间存在着许多差别,由于缺乏统一的标准,用户在使用 Super VGA 卡时都要详尽地研究具体的资料,编写驱动程序。一个新的工业标准化组织——视频电子标准协会 (Video Electronics Standards Association) 提出了一个针对 Super VGA 的标准,即 VBE (Vesa Biso Extension) 标准。第一代 VBE 标准是 1991 年 10 月发布的 V1.2 (包括 V1.0 和 V1.1)。第二代 VBE 标准是 1994 年 11 月发布的 V2.0, V2.0 增加了显示器休眠管理等一些扩展的功能,现在的许多 3D 独立显卡支持 V2.0。第三代 VBE 标准是 1998 年 9 月发布的 V3.0,增加了显示器扫描频率控制等一些扩展的功能,现在的许多主板显卡支持 V3.0。

现在,图形显示器一般是用阴极射线管 (CRT) 或液晶作为显示器,加上控制电路组成的。显示器上图形的清晰程度是由显示器的分辨率决定的。分辨率用屏幕横向的像素点 (列数) 和竖向的像素点 (行数) 表示,如 320×200 , 640×480 , 800×600 , 1024×786 。点数越密、分辨率越高,显示的图线也就越细、越清晰。

在实际应用中,广泛地应用彩色图形显示器,如 VGA, SVGA。一般微机所用的显示器,可以使用 Turbo C、Borland C 等编程语言置为文本方式和图形方式。对于图形方式来说, Turbo C、Borland C 图形方式可以设置的最高分辨率为 VGA 模式 640×480 , 每个像素 16 色,按 VESA 标准,使用 Turbo C 的 DOS 中断编程设置图形方式时,可将显示器分辨率设置成 640×480 , 600×800 , 1024×786 , 1024×786 , 1280×1024 等,每个像素 256 色、64K 色、16.8M 色的 SVGA 模式。

由于上述采用 VESA 标准进行图形编程时使用了 DOS 方式的中断调用,编程难度较高。在 Windows 方式下进行图形编程时,可以使用 Windows 提供的 API 图形函数。虽然 API 函数对显示硬件的兼容性强,但 API 函数不直接操作硬件,故图形显示速度慢。为此,微软推出了 Direct X,它是一些直接操作硬件设备的函数集合,特别适合游戏编程。

下面我们将计算机图形学的一些基本理论作一简述。

1. 图形变换

图形变换分二维图形变换和三维图形变换,这是图形学中的最基本的内容。通过变换矩阵,可以实现图形的平移、旋转和按比例放大、缩小,以及透视变换。例如:对于三维图形,空间的一点用齐次坐标表示为 $[x, y, z, 1]$,经过变换矩阵变换后为:

$$[X, Y, Z, H] = [x, y, z, 1] \cdot T$$

$$T = \begin{bmatrix} a & b & c & p \\ d & e & f & q \\ g & h & i & r \\ l & m & n & s \end{bmatrix} = \begin{bmatrix} T_1 & T_3 \\ T_2 & T_4 \end{bmatrix}$$

上式中,



变换矩阵为:

$$T_1 = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad T_2 = \begin{bmatrix} 1 & m & n \end{bmatrix} \quad T_3 = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad T_4 = \begin{bmatrix} s \end{bmatrix}$$

上述各矩阵的意义如下:

T_1 : 比例、旋转、错切等变换。

T_2 : 平移变换。

T_3 : 透视变换。

T_4 : 总体变比变换。

2. 模型建立

计算机产生和显示逼真三维图形的首要任务是建立待显示景物的模型,生成原始图形数据,然后经过坐标变换、隐面消除和明暗处理,最后得到所要显示的图形。模型可以使用物体的点、线和面来进行描述,即本书的顶点预览(Preview of Vertices)、线框模型(Wireframe Model)和实体模型(Solid Model)。顶点预览和线框模型是用物体的顶点和轮廓线描述物体,其优点是处理速度快,易于做到实时处理。但是,这种方法产生的图形可能存在多义性,并且逼真性差,只适合应用在对图形逼真性要求不太高的领域。实体模型是用物体的边界小平面描述物体的,这种描述尽管比顶点预览和线框模型复杂,但它可以表示出物体的表面特性。

把一个三维物体显示到二维屏幕上,不可避免地会遇到面与面、线与线、物体与物体之间的相互遮挡问题。如何在显示时消去景物中被遮挡的部分以得到逼真的三维图形呢?这便是隐面消除所要解决的问题。在本书的实体建模部分,我们使用凸多面体外法线与视线的夹角 β 来判断小平面的可见性,即:

(1) $0^\circ \leq \beta < 90^\circ$, 则 $\cos\beta > 0$, 此时,外法线代表的小平面可见。

(2) $\beta = 90^\circ$, 则 $\cos\beta = 0$, 此时,外法线代表的小平面有积聚性。

(3) $90^\circ < \beta \leq 180^\circ$, 则 $\cos\beta < 0$, 此时,外法线代表的小平面不可见。

根据小平面的可见性绘制物体,可解决同一物体不同部位之间的遮挡问题。对于由一些基本物体,如圆柱、球、立方体和圆锥等构成的复杂景物模型,需要对其不同基本物体的前后位置进行排序,先画后面的物体,再画前面的物体,这样就可解决不同基本物体之间的遮挡问题。

三维物体显示的逼真性,很大程度上是由明暗处理决定的。在用隐面消除解决了物体的遮挡问题以后,就要用明暗处理模型来决定物体表面的亮度,进行逼真显示。在明暗处理中,应根据视点、物体表面特性和光照模型计算亮度(即视点所获得的光强度)。最常用的照明模型是 Phong 模型和 Whitted 模型。前者比较简单,仅考虑了环境光、漫反射光、镜面反射光和透射光对视点的贡献,例如本书实体建模部分使用的光照模型;后者除了考虑以上因素外,还考虑了物体之间的多次反射和折射的效果,例如本书光线跟踪部分使用的光照模型。经过隐面消除和明暗处理后,就可得到具有较高逼真性的物体。

3. 光线追踪

光线跟踪是两维屏幕上生成三维逼真画面的技术。使用该技术,计算机追踪从观测者的眼睛到屏幕上的每一个像素,再到任何一个相交的物体,然后再到光源的路径。所有的这些信息用来确定显示屏幕上每一个像素的颜色。用光线跟踪方法显示真实感图形有如下



优点:

(1) 有反射效果。它不仅考虑到光源的光照,而且考虑到场景中各物体之间彼此反射的影响,因此显示效果十分逼真。

(2) 有消隐功能。采用光线跟踪方法,在显示的同时,自然完成消隐功能。而且,事先消隐的做法也不适用光线跟踪,因为那些背面和被遮挡的面,虽然看不见,但仍能通过反射或透射影响着看得见的面上的光强。

(3) 有影子效果。光线跟踪能完成影子的显示,方法是从相交处向光源发射一根阴影探测光线。如果该光线在到达光源之前与场景中任一不透明的面相交,则相交处于阴影之中,否则,相交处于阴影之外。

(4) 可生成纹理。为了生成逼真性很强的图形,就要在物体表面显示纹理。通过调用不同随机纹理生成函数,来修正由光线跟踪确定的像素点的颜色,就可获得不同效果的表面纹理。

在计算机图形学的基础上,我们进一步地学些计算机图像学的基础知识也是顺理成章的事情。故本书最后对 BMP 图像格式、真彩色图像到 256 色图像的转换做了介绍。

本教材的正文可分为三个部分:第一部分,1~4 章为 Visual C++ 和图形图像学的基础知识;第二部分,5~10 章为小平面实体建模;第三部分,11~13 章为光线跟踪与图像转换。本教材附录 A 中给出了 DOS 系统下使用 VESA VBE 标准枚举所有的显示模式并在屏幕画点和输出汉字的 Borland C 实例程序。

第2章 Visual C++ Windows 编程基础

C 语言表达能力强, 控制语句丰富, 构造复杂数据类型的能力强, 书写简洁, 代码质量高。正是这些特点, 使得 C 语言越来越受到软件设计师的喜爱。Visual C++ 从 C 扩充而来并保持兼容, 其面向对象的特性使其成为 Windows 的主要编程语言。在 Visual C++ 中除了可以编译 MFC 程序外, 还可以建立 Win32 Application 和 Win32 Console Application。前者以 WinMain () 为主函数, 可以使用所有的 Win32 API。后者以 main () 为主函数, 不能使用 GDI 函数, 工作方式类似 DOS 程序。

首先, 本章在 2.1 节介绍 Visual C++ 6.0 开发环境。然后, 本章在 2.2 节使用 Visual C++ 6.0 编程平台的 App Wizard 自动生成一个 Win32 Application 程序 Hello.cpp, 该程序的运行结果是在 Windows 窗口中显示典型的 “Hello World!”, 创建 Hello.cpp 的目的是获取一个 Win32 应用程序框架, 以便在接下来的章节中, 在该 Win32 程序框架中加入一些自己的程序。最后, 本章在 2.3 节结合 Hello.cpp 介绍了 Windows 编程的基础知识。

2.1 Visual C++ 6.0 开发环境

通过实际动手编程来进行学习是最好的方法。在学习本书的同时, 可获得使用 Visual C++ 编程的实际经验。在本章, 着重介绍 Visual C++ 6.0 的开发环境及其为开发应用程序所提供的一些工具。Visual C++ 6.0 提供的工具很多, 在任何一个应用程序中都不可能全部使用到这些工具 (在短短的一章中也不可能学会使用所有这些工具)。在此着重介绍一些主要的工具, 即在学习本书过程中和应用程序中经常要使用的工具。

在开始学习 Visual C++ 6.0 开发环境之前, 应启动 Visual C++ 6.0, 在计算机上实际看看 Visual C++ 的各区域是如何布置的, 以及如何更改这些布置。

当 Microsoft Visual C++ 6.0 启动以后, 可看到一个如图 1-1 所示的开发工作室窗口。开发工作室中每个区域都有其特殊的用途。您可以重新布置各个区域, 来定制开发室环境, 使其适合自己的特殊开发需要。

2.1.1 工作区

当首次启动 Visual C++ 6.0 时, 在开发工作室的左侧有一块区域, 占去一大块面积, 却没有显示内容。这就是工作区 (Workspace)。工作区允许以下三种不同方法查看应用程序的各个部分:

- Class View (类视图)。使您能够在 Visual C++ 类的层次上找到并操作源代码。



- Resource View (资源视图)。使您能够在应用程序中找到并编辑各种资源,包括对话框的设计、图标和菜单等。
- File View (文件视图)。使您能够查看构成应用程序的文件并确定它们的位置。

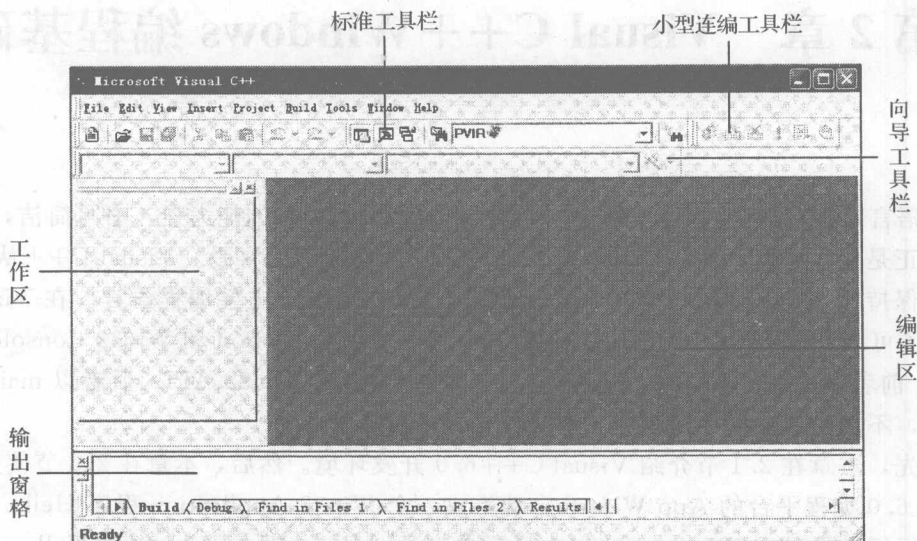


图 1-1 Visual C++6.0 的开始屏幕

2.1.2 输出窗格

第一次启动 Visual C++6.0 时,也许看不到输出窗格 (Output Pane)。等到完成对第一个应用程序的编译后,输出窗格才出现在开发工作室的底部。这时除非关闭它,否则输出窗格将一直开着。输出窗格是开发工作室给您提供信息的地方;在那里可看到编译程序的进展说明、警告及出错信息;在逐步查看代码时,输出窗格是 Visual C++ 调试程序显示所有变量当前值的地方。当关闭输出窗格后,它会在 Visual C++ 需要显示有关信息时自动打开。

2.1.3 编辑区

编辑区 (Editor Area) 位于开发工作室的右部。它是使用 Visual C++6.0 时进行一切编辑的区域;当编辑 C++ 的源代码时将在这里显示代码编辑窗;在设计对话框时,窗口绘制器也在此显示。另外,当设计应用程序中使用图标时,编辑区将显示图标绘制器。编辑区基本上就是开发工作室的全部区域,不能为窗格、菜单或工具栏所占据。

2.1.4 菜单栏

第一次运行 Visual C++6.0 时,在菜单栏 (Menu Bars) 下有三个工具栏。Visual C++6.0 还有许多其他的工具栏。可以根据最适应自己工作的方式去定制和创建自己的工具栏。初次打开的工具栏如下:

- 标准工具栏 (Standard Bar)。该工具栏包括绝大多数标准工具:打开和保存文件、



剪切、复制、粘贴，以及可能有用的其他各种命令。

- 向导工具栏 (Wizard Bar)。该工具栏使您能执行许多类向导动作，而不用打开类向导。
- 小型连编工具栏 (Build Bar)。给您提供开发和测试应用程序时最可能用到的连编和运行命令。整个连编工具栏还允许您在多种连编配置之间切换，如在调试和释放连编配置之间进行切换。

2.1.5 重新布置开发工作室环境

开发工作室为重新布置开发环境提供两种简单的方法。一种是在工具栏上单击鼠标右键，这将打开如图 1-2 所示的弹出式菜单，以便打开或关闭各种工具栏和窗格。

另一种重新布置开发环境的简单方法是用鼠标抓住工具栏或窗格左侧底部的双重栏，可以把工具栏从其当前的位置拖开，形成如图 1-3 所示的浮动工具栏。可以把这些工具栏 (和窗格) 拖到开发工作室的任何一个边缘，并停在那儿。而且，即使这些工具栏已停在某一位置，还可以通过双重栏把这个工具栏向左或向右拖到需要的位置。



图 1-2 工具栏开关菜单



图 1-3 浮动小型工具栏示例

2.2 Wizard 创建 Win32 应用程序

2.2.1 开始第一个项目

作为第一个 Visual C++ 6.0 应用程序，您将创建一个简单的应用程序，如图 1-4 所示。为建立这个应用程序，需要创建项目工作区和使用应用程序向导创建应用程序外壳。

2.2.2 创建项目工作区

在 Visual C++ 中，每一个应用程序开发项目都需要有自己的项目工作区。工作区包括应用程序源代码所存放的目录，

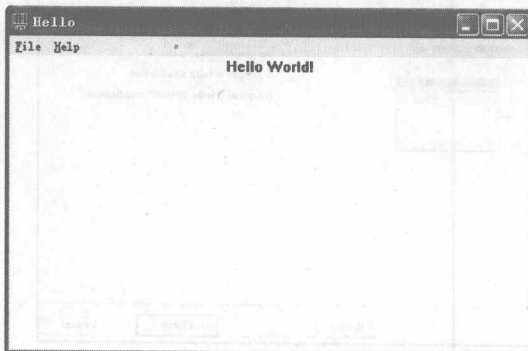


图 1-4 第一个 Visual C++ 6.0 应用程序



以及各个连编配置文件所存放的目录。可以通过如下步骤创建一个新的项目工作区：

(1) 选择 File | New, 打开 New Wizard, 如图 1-5 所示。

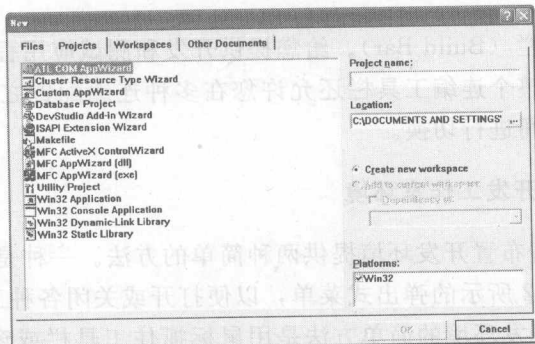


图 1-5 New Wizard

(2) 在 Project 选项卡上, 选择 Win32 Application。

(3) 在 Project Name (项目名) 域中, 给项目输入一个名称, 如 Hello。

(4) 单击 OK 按钮。这使 New Wizard 做两件事情: 创建一个项目目录 (在 Location 域中指定), 然后启动 AppWizard。

2.2.3 使用应用程序向导创建应用程序外壳

AppWizard 将问您一系列问题, 包括您打算建立什么类型的应用程序, 以及需要什么样的特性和功能。使用这些信息, AppWizard 将创建一个可以立即编译和运行的应用程序外壳。这个外壳提供建立应用程序所必需的基础结构。执行以下步骤可看到它是如何工作的:

(1) AppWizard 的第一步, 制定您将创建的一个典型的 Hello World! 应用程序, 如图 1-6 所示。先选择 A typical "Hello World!" application, 再单击 Finish 按钮, AppWizard 将生成应用程序外壳。

(2) AppWizard 在生成应用程序外壳之前会显示一个清单, 列示此向导将在应用程序外壳中放置的内容, 如图 1-7 所示。清单内容取决于您所执行 AppWizard 过程中所做的选择。单击 OK 按钮, AppWizard 就可以生成应用程序。

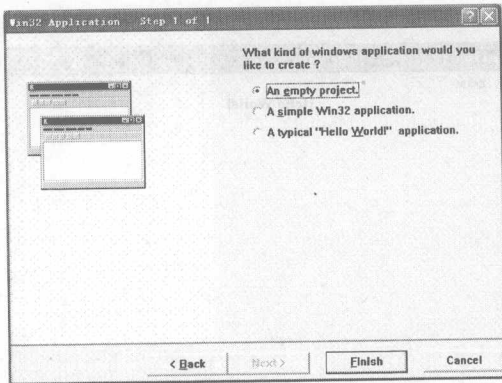


图 1-6 AppWizard 的第一步

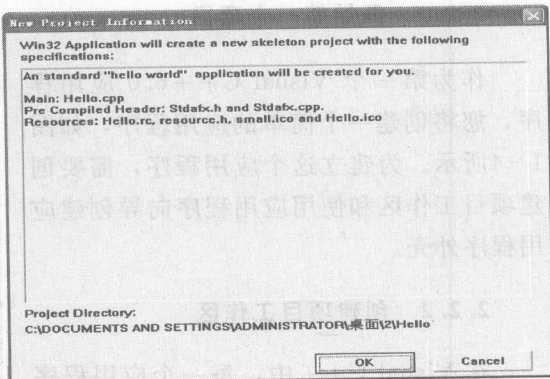


图 1-7 New Project Information 屏幕



(3) AppWizard 生成应用程序外壳之后, 将返回开发工作室环境。在工作区窗格中, 可以看到应用程序外壳中的类的树形视图, 如图 1-8 所示。在开发工作室的编辑区中编辑 C++ 源代码。

(4) 选择 Build | Build Hello.exe 以连编应用程序。

(5) 在 Visual C++ 编译器连编应用程序时, 输出窗格中将滚动显示进度和编译器的其他信息。当应用程序连编结束后, 输出窗格将显示信息, 告诉您是否出错或有警告, 如图 1-9 所示。

(6) 选择 Build | Execute Hello.exe, 运行应用程序。

(7) 应用程序显示一个处于活动状态的窗口, 上面有菜单栏 File, Help。如前面的图 1-4 所示。可选择 File | Exit, 关闭该窗口。

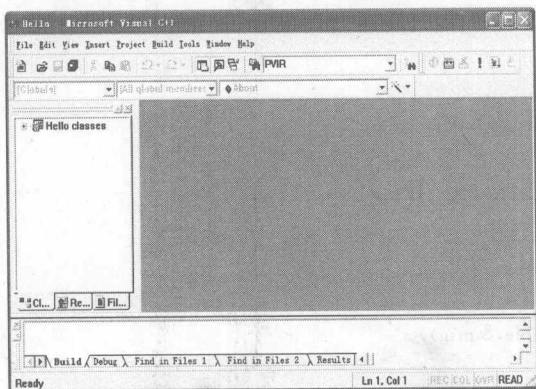


图 1-8 工作区中项目类的树形视图

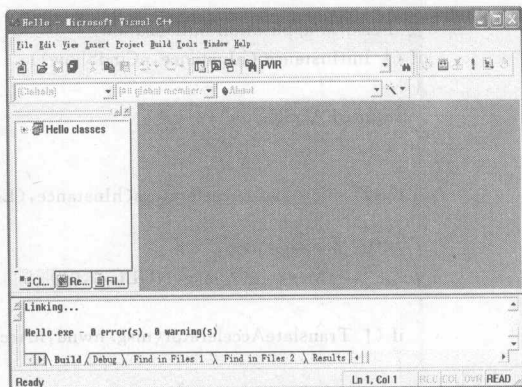


图 1-9 输出窗格显示编译器错误

2.3 Windows 编程基础

下面是由 Visual C++ 6.0 AppWizard 自动生成的 Hello.cpp:

```
// Hello.cpp : Defines the entry point for the application.
//

#include "stdafx.h"
#include "resource.h"

#define MAX_LOADSTRING 100

// Global Variables:
HINSTANCE hInst;                                // current instance
TCHAR szTitle[MAX_LOADSTRING];                 // The title bar text
TCHAR szWindowClass[MAX_LOADSTRING];           // The title bar text

// Forward declarations of functions included in this code module:
ATOM MyRegisterClass(HINSTANCE hInstance);
BOOL InitInstance(HINSTANCE, int);
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK About(HWND, UINT, WPARAM, LPARAM);
```



```
int APIENTRY WinMain(HINSTANCE hInstance,
                    HINSTANCE hPrevInstance,
                    LPSTR lpCmdLine,
                    int nCmdShow)
{
    // TODO: Place code here.
    MSG msg;
    HACCEL hAccelTable;

    // Initialize global strings
    LoadString(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadString(hInstance, IDC_HELLO, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);
```

```
    // Perform application initialization:
    if (! InitInstance (hInstance, nCmdShow))
    {
        return FALSE;
    }

    hAccelTable= LoadAccelerators(hInstance, (LPCTSTR)IDC_HELLO);

    // Main message loop:
    while (GetMessage(&msg, NULL, 0, 0))
    {
        if (! TranslateAccelerator(msg, hwnd, hAccelTable, &msg))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }
}
```

```
return msg.wParam;
}
```

```
//
// FUNCTION: MyRegisterClass()
//
// PURPOSE: Registers the window class.
```

```
//
// COMMENTS:
```

```
// This function and its usage is only necessary if you want this code
// to be compatible with Win32 systems prior to the 'RegisterClassEx'
// function that was added to Windows 95. It is important to call this function
// so that the application will get 'well formed' small icons associated
// with it.
```

```
ATOM MyRegisterClass(HINSTANCE hInstance)
```

```
{
```

```
    WNDCLASSEX wcex;
```

```
    wcex.cbSize= sizeof(WNDCLASSEX);
```

```
    wcex.style = CS_HREDRAW | CS_VREDRAW;
```

```
    wcex.lpfnWndProc = (WNDPROC) WndProc;
```

```
    wcex.cbClsExtra = 0;
```