开发人员专业技术丛书

Windows 图形编程

(Windows Graphics Programming Win32 GDI and DirectDraw)

(美) Feng Yuan 著

英宇工作室 译

机械工业出版社
China Machine Press

PH PTR
Pearson Education
培生教育出版集团

译 者 序

当我第一次接过这本书的英文版样书时，我就惊诧于这本书是一个中国人写的。

本书作者 Feng Yuan 是一位留美学者，软件工程学博士，毕业于南京大学。在过去的四年中，他在 HP 温哥华研究与开发实验室从事 DeskJet 打印机驱动程序的开发。本书正是在此基础上写作的，它凝结了作者多年实际开发的汗水。

本书的内容覆盖了 Windows 的图形系统，从 Win32 API 的表层到最底层的显示和打印机的驱动程序，特别集中在 Win32 GDI 和 DirectX 的 DirectDraw 部分。它主要集中讨论了 Windows NT 4 和 Windows 2000 平台，还覆盖了如下一些内容：Windows 图形系统的体系结构、GDI 和 DirectDraw 的数据结构、绘图设备、GDI 对象、像素、直线和曲线、填充区域和位图、逻辑字体和文本格式化、元文件、打印和打印池、DirectDraw 接口、GDI 调试工具、EMF 反编译器、图像绘制和 3D 效果。

本书的最大特点是其实用性。它不仅讲述了这些内容，重要的是，它还给出了实际编程中用到的分析问题和解决问题的方法。本书是从实际工作中收集而来的，而不是利用其他参考文献中编辑而成的，并且它提供了许多可以直接重用的函数、C++类、示例程序源代码，这对于从事 Windows 图形编程工作的人员来说，如图形软件工程师，排版系统、中文办公软件的工程师，以及对 Windows 内幕感兴趣的程序员，大多数是可以直接使用的。本书是一本名副其实的 Windows 图形编程宝典。

本书由英宇工作室的邓浩组织翻译，参加翻译的人员还有裘强、薛赛男、林碧森、欧阳宇、易重英、陆绍飞、牛韬、王安鹏、祁力、徐继伟、谢君英、刘舫、李超、阳爱军、朱翠莲、郭增元、杨青青、李娜。还有部分人员参与了录入和排版工作，他们是温春美，李明生、林江勇、武堂、王明举、郭茜茜。

我们一直在尽力将本书翻译好，但由于时间仓促，加之译者水平有限，书中肯定会有不少疏漏，恳请读者指正。我们接收广大读者意见的邮箱为：yingyu@263.net。

2001 年 12 月

前　言

一本关于 Windows 编程的新书，要想真正有价值，它应该是深入的、完整的、最新的、正确的、基于实践的和对读者有帮助的。

一本深入的书应该从 API 的底层谈起，一直讲到 API 的设计原理、内部数据结构以及实现方法。它应该为你提供检查和探测工具。

一本完整、最新的书，应该把注意力集中在迄今为止对 Win32 API 的最好实现——Windows 2000 上（它将是微软将来的操作系统的基础），并覆盖它的所有新特性。

一本正确的书，应该是对 Win32 API 功能进行实际探索，并验证所有细节后书写的。只阅读微软的开发文档显然是不够的，因为这些文档只是 Win32 API 的概要，并且有些不完全的、过时的甚至是模棱两可的信息。

一本实用的、有用的书，对于程序员和软件工程师来说，不应该只描述 API 和提供一些简单例子，而应该解决现实世界的问题，提供可重用的生成模块，提供有用的工具，并鼓励读者写出很专业的程序。

特别是，这种书一般都需要阐述 Win32 GDI，或者 Windows 图形编程，它是所有 Windows 编程的基础生成部分。到目前为止，讲解最深的可能是 Matt Pietrek 的书，它讨论了 Windows 3.1 GDI 的内部工作原理。关于 GDI 最完整和最新的描述由微软的 MSDN 库提供。Charles Petzold 最有名的一本书《Programming Windows》，第五版，其中有一半的篇幅讲解 Windows 98 GDI 编程。

但是因为所有的 Windows 编程中经常需要处理 Windows GDI，所以，需要一本更深入、更完整、更新、更正确和更有用的书来讲解 Windows GDI。这就是本书的写作目标。

本书内容

本书讲解如何用 Win32 GDI API 进行图形编程，同时介绍了 DirectDraw 和 Direct3D 立即模式。它覆盖了在所有 Win32 平台上实现的共同特性，仅在 Windows NT/2000 下实现的纯 32 位特性，及在 Windows 2000 和 Windows 98 中增加的最新特性。例如，对 alpha 混合、透明位传输、渐变填充、从右到左的阅读顺序、分层窗口、向打印机发送 JPEG/PNG 图像等都进行了详细讲解。

本书是关于 Windows 编程的，并更深一步介绍这些东西是如何实现的，因此，使读者可以更高效、自信地使用 Win32 API。

本书教你全面地、分析地阅读 Win32 文档，尽力理解 Win32 API 设计和实现者的思路，并用逻辑推理和实验来完整理解 Win32 API，甚至找出文档中遗漏的信息或错误的信息。

本书教你如何使用编程工具来理解 Win32 API。更重要的是，本书示范了如何建立你自己的工具，经常使用核心硬件级的系统编程技术，并设计一些有趣的实验对 Win32 API 的没有文

档记载的世界进行探索。开始的几章可以作为系统级 Windows 编程书，也可用于 Windows 编程的其他领域。

本书教你创建可重用的生成块以方便实际使用。除了一些简单的测试和示例程序外，本书包含了大量的可重用函数、C++类、驱动程序、工具和重要程序，可用于产品级的 Windows 程序。它开发了一个面向对象的 Windows 编程 C++类库，支持简单的窗口、SDI 窗口、MDI 窗口、对话框、工具栏、状态窗口、属性表、子类、通用对话框等等。它提供一些类，处理 DIB/DDB/DIB 段、EMF 绘制、图像处理、颜色量化、错误扩散半色调，JPEG 图像解码/编码、字体文件解码、字体嵌入、PANOSE 字体匹配、图元绘制、3D 文本、设备无关的多重页面布局、DirectDraw 封装、Direct3D IM 封装等。

本书的代码不依赖于 Microsoft Foundation Class，或者任何的第三方类库，可用于任何C++程序。所有这些类命名为以 K 打头，因此你可以很容易地将他们与 MFC、ATL、OWL 或者你自己的类库混合使用。

本书组织结构

本书覆盖了 Windows 图形编程的三个层次：实现层、API 层和应用层。

实现层覆盖了 Win32 GDI API 和 DirectX COM 接口背后的内容，它是 Windows 图形引擎和 Win32 子系统客户 DLL 没有文档记载的领域。第 2、3、4 章覆盖了实现层的所有内容，为理解 API 层建立一个坚固的基础。

API 层提供了 Win32 GDI API、DirectDraw 以及一些 Direct3D IM 的精辟的、准确的、一步一步的描述。应用层建立在 API 层之上，用于解决现实领域的问题、实现可重用函数、C++类和重要程序。API 层和应用层根据一些主题一起讨论。一般说来，一章的开始部分覆盖 API 层，接着转移至与现实应用相关的程序。对于讲述更复杂的主题（如位图），用一章覆盖基本的主题，两章覆盖更高级的应用。

第 1 章“基本技术和知识”，回顾了基本的 Windows 编程技术，这些在本书的后面将是有用的。它包括基本的 Windows 编程、Intel 汇编语言、程序开发环境、Win32 可执行文件格式以及 Windows 操作系统的体系结构。我最喜欢的部分是 API“钩子”，它通过重写 Win32 模块的导入/导出目录来实现。

第 2 章“Windows 图形系统体系结构”从整体上浏览了 Windows 图形系统，从 Win32 子系统 DLL 一直到图形设备驱动程序。它讲述了 Windows 图形系统组件、GDI 体系结构、DirectX 体系结构、打印子系统体系结构、图形引擎、显示驱动程序和打印机驱动程序。我最喜欢的部分是对系统服务调用的描述，它在用户模式 GDI 实现代码和核心模式图形引擎之间架起了一座桥梁，列出没有文档记载的系统服务调用（从 GDI32.DLL、USER32.DLL、NTDLL.DLL 和 WIN32K.SYS）的工具和一个简单的打印机驱动程序，它能生成带嵌入位图的 HTML 页面。

第 3 章“GDI/DirectDraw 内部数据结构”可以作为侦探故事或者寻宝故事来阅读。它首先解释 Win32 的基于句柄的面向对象编程范例，接着将讲解 GDI 对象句柄的实质，然后定位 GDI 句柄表，对它解码，最后将 Windows 图形系统内部保存的复杂数据结构展现给读者。并用虚拟内存查询。Microsoft 的调试符号文件、自己开发的工具和 Microsoft 的 Visual C++调试器来定位

GDI 句柄表。并开发了一个内核模式驱动程序来读内核模式地址空间的数据。第 3 章开发的 Fosterer 程序用 Microsoft 的 GDI 调试器扩展功能来解码 GDI 句柄表和图形引擎/DirectX 内部数据结构。在 Windows NT 和 Windows 2000 上，你最好不要错过 Fosterer 程序。但首先，你需要安装调试符号文件并获得 Microsoft 的 WinDbg。

对内部数据结构的描述应该看成是一种参考资料，通过它可以对 DDI 级别的调试取得更深的了解，因为其中的细节信息在不同版本之间会不断变化，甚至不同服务包间也有变化。当你需要理解某些东西时，你可以任意跳过任何你不感兴趣的章节然后再回来，如想了解 GDI 对象资源是如何使用的及其性能问题。

第 4 章“Windows 图形系统窥视”，介绍了多种技术和工具窥视 Windows 图形系统和整个 Windows 系统。在这一章学习了将 DLL 嵌入外部过程，钩入 API 函数调用链，信息收集，对信息解码和报告，对 API 进行说明以用于窥视程序，窥视 Win32 API 函数调用，通过二进制重定位钩入 API，系统服务调用挂钩，COM 接口钩入，及内核模式的 DDI 接口钩入。我最喜爱的部分是使用汇编语言编写 proxy 程序，钩入模块内的函数调用、系统服务调用和 DDI 调用来观察系统是怎样工作的。第 4 章是面向是核心硬件编程人员，如果你现在不需要用到它，跳过它。

第 5 章“图形设备抽象”，首先描述了整个 Windows 图形编程 API 和实际用法，第 5 章的内容包括视频显示卡、帧缓冲区、GDI 设备上下文对象、通用框架窗口类及在窗口中绘图。我最喜欢的部分是 WinPaint 程序，它形象化地显示了窗口的绘图消息。

第 6 章“坐标空间和变换”，讨论了 GDI 支持的四种坐标空间，窗口到视口映射，世界坐标空间变换（affine 变换），最后所讨论了滚动和缩放的用法。我很怀念以前下围棋（一种东方的棋盘游戏）的时光，在写这本书的时候我编写了一个简单的围棋棋盘显示程序作为第 6 章的示例程序。

第 7 章“像素”对 GDI 对象，句柄和 GDI API 级别上的句柄表做了一般性的描述，然后描述了一个监控系统范围内 GDI 句柄使用情况的程序，接着讨论了简单区域，并对 GDI 裁剪、颜色空间、像素绘图进行全面描述，最后以一个 Mandelbrot 集绘图程序结束。本章最好的部分也许是对系统区域、元区域、裁剪区域和 Rao 区域的描述，GDI 使用它们控制裁剪，ClipRegion 程序将它们可视化。

第 8 章“直线和曲线”包括二元光栅操作、背景模式、背景颜色、逻辑画笔对象、直线、Bezier 曲线、弧线、路径和用自己定义的 GDI 不支持的风格直线做图。我喜欢的部分也许是与将椭圆曲线转换成 Bezier 曲线相关的数学知识。

第 9 章“区域”包括画刷、长方形、椭圆、Chords、饼状图、圆角矩形、多边形、闭合路径、区域、渐变填充，并对图形应用程序使用的各种区域填充技术进行了总结。我最满意的部分是使用渐变填充绘制三维按钮，及对与区域相关的数据结构的描述。

第 10 章“位图基础”集中讨论了 GDI 支持的三种位图格式，即设备无关位图、设备相关位图和 DIB 段，还讨论了用于 DIB、DDB 和 DIB 段的封装类，及内存设备环境和这些位图的普通用法。我最喜爱的部分是这些封装类，尤其是利用内存映射的 DIB 段实现设备无关的、高分辨率增强型元文件的绘制。

第 11 章“高级位图图形学”包括三元光栅操作、透明位图、无掩码的透明处理、alpha 混

合以及 Windows 2000 的新特性——分层窗口。我最喜欢的部分就是光栅操作的全部内容，尤其是光栅操作图及用多个三元光栅操作模拟四元光栅操作。

第 12 章“用 Windows 位图进行图像处理”，讨论了对位图中像素的直接存取、位图 offine 变换、位图颜色变换、位图像素变换和位图的空间滤波器。我最满意的部分是面向对象的、基于模板的、通用图像处理框架的设计，它的扩展性很好。

第 13 章“调色板”介绍了系统调色板、逻辑调色板、调色板消息、用于位图的调色板，颜色量化和用错误扩散来降低位图颜色深度。这里实现的八叉树颜色量化算法似乎可以产生比起商业软件更好的调色板。

第 14 章“字体”介绍了字符集、代码页、图元、字面、字体族、位图字体、矢量字体、TrueType 字体、字体安装和字体嵌入。我特别喜欢对 TrueType 字体文件的解码。

第 15 章“文本”讨论了逻辑字体、字体映射、PANOSE 字面匹配、文本测量、简单文本绘制、高级文本绘制、文本格式和特效。最后的文本特效这部分是最好的，它包括文本着色、阴影、阴文、阳文、空心、旋转、竖排文本、将文本放置在曲线上、将文本作为图像、将文本作为轮廓，甚至是简单的三维文本。

第 16 章“元文件”讲述了创建和显示元文件的基本知识、内部元文件组织详细内幕、在元文件对 GDI 功能编码、元文件解码、枚举元文件、EMF 反编译器以及从打印池程序中捕获 EMF。其中最好的部分是 EMF 反编译器和 EmfScope 程序，它能捕获 Windows 95/98 EMF 打印池程序文件。

第 17 章“打印”讨论了打印池、利用 GDI 实现基本打印功能、设计你的打印程序、JPEG 图象打印（包括直接发送 JPEG 到打印机驱动程序）和 C++ 语法亮显源代码的打印。最好的部分是用于多页、多列页面显示的通用框架结构，它与分辨率和显示比率无关。JPEG 和源代码打印程序都使用了这个通用框架结构。

第 18 章“DirectDraw 和 Direct3D 立即模式”为有经验的.GDI 程序员介绍了 DirectX 2D/3D 绘图部分。它介绍了 COM 基础知识、DirectDraw 和 DirectDraw 表面的封装类、在 DirectDraw 中绘图的三种方法、DirectDraw 裁剪器、后台表面、DirectDraw 中的字体和文本、DirectDraw 立即模式的封装类、窗口模式的 DirectX、双缓冲和纹理。我最满意的部分是使用 GDI 创建 DirectDraw 字体表面，它能够在 DirectX 表面上高效地显示文本。

如何阅读本书

本书主要适合于使用 Win32 API 或基于 Win32 API 的类库的中级的、有经验的或高级程序员阅读。

初级 Windows 程序员应该从阅读其他的书开始，或通过在线帮助熟悉 Windows 编程的框架和概念内容，并跟踪一个程序来理解它是怎样工作的。

如果你只对 Windows 图形编程感兴趣，或对幕后系统级执行细节不感兴趣，则可以阅读第 1 章和第 2 章，跳过第 3 章和第 4 章，然后阅读第 5 章以后的内容。你甚至可以跳过第 1、2 章的某些小节。从第 5 章开始的内容系统地增加了难度。

如果你是有经验的或高级程序员，你清楚什么是你最感兴趣的。也许你会先看看一些东

西，然后跳到第 3 章。

如果你只对系统级编程感兴趣，如 API 窥视，则需在阅读第 1、2 章后，接着阅读第 3、4 章。

如果你不是一个程序员——比如说，你是测试工程师——你可以通过阅读第 2 章获得对 Windows 图形系统的整体印象，要了解更多有关 GDI 资源泄漏问题，可以阅读第 3 章的第一部分，从中获得一些工具来查找资源泄漏。

本书光盘内容

本书列举了很多例子程序，收集了可重复使用的函数和类。更准确地说，它有超过 1300KB 的 C++ 源文件，400KB 的 C++ 头文件，加上 JPEG 库源文件的修改版本，它是基于独立 JPEG 组的自由源代码的（www.ijg.org）。全部源代码被编译成 49 个可执行文件，三个内核模式驱动程序和一个用户模式动态库。

一般说来，书中不会将所有的源代码列举出来，光盘中包含了全部源代码、Microsoft Visual C++ 6.0 workspace 文件、预编译好的 debug 和 release 版本的二进制文件和在本书中做位图/图像处理时用到的 JPEG 图像。光盘带有一个自动安装程序，它会自动安装光盘中的内容、程序组、链接和一些重要的 Web 地址以便下载 Microsoft 工具及取得支持。

这些程序在 Windows 2000 的最终版本（build 2195）下开发和测试，使用支持 DirectX 7.0 2D/3D 硬件加速的显卡，但大多数程序可以工作在 Windows 95/98/NT 4.0 下，且不需要 DirectX 支持。

要编译所有的程序，你的系统必须安装以下工具：

- Visual C++ 6.0，编译器。
- Visual Studio 6.0 Service pack 3，编译器升级包，可在 msdn.microsoft.com /vstudio/sp/vs6sp3 中下载。
- MSDN library，在线帮助文档。
- Platform SDK，请访问 www.microsoft.com/downloads/sdks/platform/platform.asp 得到最新的头文件、库文件及工具。确认你的 VC 6.0 的 include 和 library 目录用 Platform SDK 更新过。
- Windows 2000 debug symbol files，有几个工具要使用它，对调试有帮助。从 www.microsoft.com/windows200/downloads/otherdownloads/symbols 中下载。
- Windows 2000 DDK，几个核心模式驱动程序需要用到它，位于 www.microsoft.com/ddk，将 DDK 的 inc 目录加入到 VC 的 include 目录，将 DDK 的 libfre \ i386 目录加入到 VC 库文件目录。
- WinDebug，第 3 章的工具要使用它，位于 www.microsoft.com/ddk/debugging。

即使本书的所有示例代码都是用不带 MFC 的 C++ 写的，MFC/ATL/OWL 程序员也可以很容易地使用这里开发的代码。甚至 Visual Basic 或 Delphi 程序员也能从本书的内容和代码中受益，因为这些开发环境允许直接调用 Win32 API 函数。

本书特点

写作对作者来说，是一个很好的机会，迫使他构思、研究，并以一种有序的方式介绍事物。当一本书完成时，作者的受益是最多的。我希望，其他的程序员也能从中学到很多东西，因为它是我自己学习过程的详细记录。

更重要的是，这个学习的教室现在从我的办公室和总公司突然扩展到整个世界，现在读者都变成我的老师和同学了。如果你发现任何错误和缺点，或者你有任何意见和建议，请通过我的个人网站联系我：http：//www.fengyuan.com。

网站也提供常见问答、更新以及在使用更复杂的示例程序时的指导等等。

目　录

第 1 章　基本技术和知识

我们将深入研究 Windows 图形系统，从 Win32 API 的表面深入到显示/打印驱动程序的内部。Windows 图形系统中有很多重要的地方，我们将着重研究其中最重要的部分：Win32 GDI（Graphics Device Interface，图形设备接口）和 DirectX 的 DirectDraw 部分。

Win32 GDI API 实现于多平台，如 Win32、Win95/98、Win NT 3.5/4.0、Windows 2000 及 WinCE，但是在各个平台上有很大的不同。例如，Win32 和 Win95/98 基于老式的 16 位 GDI 实现，带有很多限制；而在真正的 32 位的 Win NT 3.5/4.0 和 Windows 2000 系统上执行时，可以实现很多强大的功能。DirectDraw 接口可以在 Win95/98、Win NT 4.0 和 Windows 2000 平台上实现。本书将着重讲述 Windows NT 4.0 和 Windows 2000 平台，即这些 API 最强大的实现平台，在适当时候会提到其他平台。

在深入研究 Windows 图形系统之前，我们先来复习一下对我们的研究很重要的一些基本技术。在本章中，我们将涉及到 C/C++ 进行简单的 Windows 编程、汇编语言初步编程、程序开发和调试工具、Win32 可执行文件的格式以及 Windows 操作系统的体系结构。

注意　本书假设读者是中高级 Windows 程序员，因此对于这些技术只做了简要的概述。想要更进一步了解这些内容的读者可以参阅其他书籍。

1.1　用 C/C++ 进行基本的 Windows 编程

我们现在使用的编程语言已从晦涩难懂的机器语言奇迹般地发展到了当今的高级程序设计语言，像 C、VB、Pascal、C++、Delphi 和 Java 等。不管程序员使用的是何种语言，他一天所写的代码几乎是一样多的。自然，随着程序语言抽象性越来越高，编写的程序也就越来越高级。

用于 Windows 编程最常用的语言就是 C，例如，微软的平台软件开发工具（Platform Softwore Development Kit，Platform SDK）和设备驱动程序工具（Device Driver Kit，DDK）的示例程序就是用 C 编写的。面向对象的语言，如 C++、Delphi 和 Java 发展得很快，将取代 C 和 Pascal 成为新一代的 Windows 编程语言。

面向对象的语言比以前的语言有了明显的改进。例如 C++，即使没有它的纯面向对象的特性，如类、继承和虚拟函数，只有现在的一些普通特性，也比 C 有很大的改进，如在函数原型、内部函数、重载、操作符和模板等方面。

但是编写面向对象的 Windows 程序并不是一件易事，主要因为 Windows API 没有设计为支持面向对象的语言。例如，Windows 消息处理和对话框处理过程回调函数必须是全局函数。C++ 编译器不允许传入一个普通类型的成员函数作为回调函数。Microsoft Foundation Class（微软基础类库，MFC）设计为封装 Windows API 在一个类层次中，MFC 已经成了编写面向对象的 Windows 程序的事实上的标准。它要逐渐变成连接面向对象的 C++ 和面向 Win32 API 的 C 语言的桥梁。MFC 传递一个单一的全局函数作为一般的窗口消息过程，该函数使用一个从 HWND 到 CWnd 对象指针的映射，把 Win32 窗口句柄转换为一个指向 C++ 窗口对象的指针。当 Microsoft 还在为 MFC 的大小和复杂度发愁时，OLE、COM 和 ActiveX 已经流行起来，因此编写不太重要的 COM 服务程序和 ActiveX 控制程序，建议还是使用另一种 Microsoft 的类库，即活动模板库（Active Template Library，ATL）。

随着向面向对象的编程发展趋势，本书中列出的示例代码大部分是 C++，少部分是 C 语言的。为了增强代码对 C 程序员、C++ 程序员、MFC 程序员、ATL 程序员、C++ 生成器程序员、甚至是对 Delphi 和 VB 程序