

计算机图形与图像丛书

C 的 Windows 动画程序设计

C FOR WINDOWS[®] ANIMATION PROGRAMMING



学苑出版社

- ▶ Over 700k of royalty free source code on the enclosed disk
- ▶ Tips for drawing characters, motion, and perspective
- ▶ Supports QuickC for Windows, Turbo C++ for Windows, Borland C++, Microsoft C, SDK, Watcom C, and more

Lee Adams

DISK
●
INCLUDED

计算机图形与图像丛书

C FOR WINDOWS
ANIMATION PROGRAMMING

C 的 Windows 动画程序设计

原著	Lee Adams(美)
翻译	施小龙 葛玉宝 邓明辉
审校	万 博

学苑出版社
1994 年. 北京

(京)新登字 151 号

内 容 简 介

本书主要是针对利用各种 C 进行 Windows 动画程序设计的人编写的,介绍了进行 Windows 动画程序设计的高级技术。本书共分四部分,第一部分介绍图形编程,介绍 Windows 编程的基本概念,说明如何用 Windows 的图形函数绘制图形,讨论各种图形输出的方法,介绍三维模型,说明如何在动画中使用各种字体;第二部分介绍 Windows 中的动画,说明实时动画和帧动画的设计;第三部分介绍 CEL 动画的原理、特色、格式、透视效果、动画移动、背景处理等;第四部分介绍动画仿真。附录给出一些重要的信息,包括如何设置各种 C。

本书是从事动画创作人员不可缺少的参考书。

欲购本书的用户可直接与北京 8721 信箱联系,电话 2562329,邮编 100080。

版 权 声 明

本书英文版由 McGraw-Hill 公司出版,版权归 McGraw-Hill 公司所有。本书简体字中文版由 McGraw-Hill 出版公司授权北京希望电脑公司和学苑出版社独家出版、发行。未经出版者书面许可,本书的任何部分均不得以任何形式或任何手段复制或传播。

计算机图形与图像丛书

C FOR WINDOWS

ANIMATION PROGRAMMING

C 的 Windows 动画程序设计

原 著:Lee Adams

翻 译:施小龙 葛玉宝 邓明辉

审 校:万 博

责任编辑:甄国宪

排 版:万博图书创作社

出版发行:学苑出版社 邮政编码:100036

社 址:北京市海淀区万寿路西街 11 号

印 刷:北京朝阳展望印刷厂印刷

开 本:787×1092 1/16

印 张:43.75 字数:1064 千字

印 数:1~5000 册

版 次:1994 年 9 月第 1 版第 1 次

本册定价:89.00 元

ISBN7-5077-0884-5/TP·26

前 言

计算机动画是由人类发明和创新精神的遗产之一——传统电影动画机的古老技术——发展而来的。电影动画机的历史可追溯至 1824 年。当时,法国人 Pater Roget 创立了视觉暂留理论。

七年之后的 1831 年,Joseph Plateau 发明了一种叫 Phenakistoscope 动画设备。它实际上是一个旋转的圆盘。人们可以通过许多片格窗观看到镶嵌在其内部滚筒上的图画。通常认为 Plateau 的 Pheakistoscope 的出现标志着动画技术的开始。

Plateau 的发明是对当时所谓的幻灯机的一种改进。当时的幻灯机是将手工绘制的图片放到灯罩中,当依次点亮这些图片就产生了运动的错觉。

1834 年,英国人 Horner 发明一种自动化程序更高的叫“活动画片玩具”的设备。它是一个将手工绘制的图片镶嵌在内部表面的旋转圆筒。当圆筒旋转时,观众可以从一个固定的槽中看到依次转过的所有的图片。它产生的效果有点像手指快速翻动书页的情景。

在 18 世纪中叶,上述两种动画设备很流行。1860 年左右,当时在法国工作的 Emile Reynaud 用一套翻转的镜片取代了活动画片玩具的观察槽。这使得动画的效果更为生动。这种设备在欧洲和北美得到了大量生产和销售。

随着技术的进一步成熟,放映画片小型流动剧院产生了。聪明的美国发明家们也一直在对动画设备进行改进。1870 年,Heyl 完善了一套相当于快门的翻转镜片,使其能够用汽灯产生投影效果。他的这个所谓的幻觉镜在费城音乐厅演示时吸引了 1500 多名观众。

但是,直到 Thomas Edison 电影放映机的出现,才真正产生了传统意义上的动画电影。它称得上是世界上第一个电影摄影机和电影放映系统;电影放映机迅速推广,在许多国家得到使用和改进。法国人 Lumiere 兄弟设计的电影放映机也在欧洲的大部分地区广为流行。看电影的观众蜂拥而至。

1892 年,Emile Reynaud 在巴黎开办了第一家固定的商业性电影院——LaTheatre Optique。1896 年,北美第一家电影院也在纽约开张了。

Edison 的电影放映机体积庞大,不适合于逐帧曝光,而 Lumiere 兄弟的电影放映机却是一种很容易进行逐帧曝光的便携式摇柄式的摄像机。同时,动画电影的观念渐渐为一些头脑灵活的人所接受。其中包括北美新闻界的卡通画家们。

Winsor McCay 每天为 The New York Herald 报纸画一种称为 Little Nemo 的滑稽连环画。在 1912 年,他绘制了一部短动画片。McCay 创立了动画循环技术,即重复地演示一系列的图片来模仿一些重复性动作。例如跑动、走动、咀嚼等等。1914 年,他制作了一部几乎和查理·卓别林的电影同样流行的卡通电影 Geftie the Dinosaur。

很快,其他北美报纸的卡通画家们也开始尝试制作动画电影。1913 年一位叫 Raoul Barre 的加拿大人发明了可以批量生产动画电影的定位系统。通过在画纸上打定位孔,每一帧就可放置于画板相应的标志之上,从而保证了胶片位置的准确性。不久,美国漫画家 John Bray 建立了第一个商业性的动画制作厂,他使用了 Barre 发明的定位方法,并在 1915 年把曾制订出透明画片基本规则的卡通画家 Kansas 城的 Earl Hurd 雇至门下。

动画透明片,在本书中称为图元或图案(cel),这个词来源于绘制单幅动画图片的透明赛璐璐板。将两张或三张赛璐璐板叠加在一起,众多的前景人物和背景就可以集成为一幅动画图片。如果背景绘制在放大的赛璐璐板上,它在放映过程中就可以移动,从而产生振动背景效果赛璐璐板。

动画电影逐渐成为一种产业。动画片中的主角,像 Betty Boop 和 Felix the cat 也逐渐全球闻名。Betty Boop 的创造者 Max Fleischer 和 Pave 完善了声音效果。一位名叫 Walt Disney (沃特·迪斯尼)的年轻人则创造出至今经久不衰的米老鼠和唐老鸭。1938,迪斯尼公司生产出第一部整幅动画电影——白雪公主和七个小矮人。Disney 公司的艺术家们使用了十万多张底片制作的影片一推出就获得了巨大的票房成功。1940 年和 1941 年,Disney 公司分别推出了 Fantasia 和 Pinocechio。

大型的制片厂,如华纳兄弟公司,哥伦比亚公司,米高梅公司等也很快开始生产动画影片。华纳公司的 Looney Tunes 和 Merrie Melodies 系列获得了巨大的商业成功。现代传奇人物像 Tex Avery ,Chuck Tones 和 Friz Freleng 也开始创作 BugsBuny 和 Daffy Dack 等动画形象来和迪斯尼公司的米老鼠、唐老鸭等形象竞争。动画电影也逐渐开始用于一些正式严肃的场合如培训、模拟、教育、协调公共系统、官方宣传和科学研究等。

二十世纪五十年代到六十年代,传统的电影动画技术开始完善。虽然已经去除了很多不利因素,动画制作仍是一个需要大量时间和人力投入的行业。首先,需要设置“故事板”,用一系列的图片勾勒出故事的整体结构和主要情节;其次,要把对话、音响效果和配乐的声音录制下来,接着,要进行更细致的设计以确保图画和声音的匹配。最后,主要设计师创建出关键帧或关键图片。

根据主设计师写在关键帧上的指令,称作 inbetweener 的二级设计师创建一系列的中间帧图片。接着,将中间帧合在一起制成试验片。修正了定时错误和定位错误之后,试验片的图帧手工绘制于赛璐璐板上形成所谓的动画透明片。随着静电复印的发明和廉价的复印机的出现,制片厂开始把经过测试的帧直接影印至动画透明片上。二级设计师再手工将颜色加到每幅动画透明片上。从放置动画透明片的架子下面来照亮它们。然后就可以用能够进行逐帧曝光的 16 毫米电影摄影机进行拍摄。

二十世纪六十年代中期,电视的发展使得对动画片的需求飞速增长。华纳公司创造了 Bugs Bunny, Daff Duck, Porky Pig, Tweety and Syluester, the Tasmanian Devil, the Road Runner and Wile E. Coyote, Speedy Gonzales Foghorn Leghorn, Yosemite Sam, Pepo Le Pew, Elmer Fudd 等一系列动画形象。同样地, Hanna - Barbera 公司也成功推出诸如 Tom and Terray, Huckleberry Hound, Quick Draw McGraw and Babalooley, Yogi Bear and Boo Boo Ben, the Flintstones, Scooby - Doo, the Tetsons, Magilla Gorilla, Snaggletooth, Top Cat, Augie Doggie and Daddie Doggie, the smafs 等动画形象。

七十年代初,一些大学和研究中心开始试验计算机辅助动画设计。这之后 15 年间,软件和硬件系统的开发研究使中间处理实现了自动化,大大缩短了从关键帧到中间影像的生产时间。很快,摄像机本身改为程控。八十年代末,全部动画片的生产过程都可以由专用的计算机系统完成。随着计算机硬件价格的降低,个人机的性能变得日益强大,在桌面系统上进行动画设计也已成为可能。

动画的演化目前告一段落。十九世纪的发明家和革新家们主要关注的是放映设备。二

十世纪初漫画家和企业家用手工绘制和电影动画。这些动画界的先驱者们为众多的九十年代的个人机程序员指明了方向。后者已拥有了进行数字化动画试验所需的具有足够速度,能源供给和存储空间的桌上计算机系统。这些程序员将帮助我们挖掘知识更新的新型思考工具——计算机辅助设计的二维,三维动画,仿真和可视化的巨大潜力。

简 介

阅读本书的读者可以是专业程序员,也可以是业余的编程爱好者,本书都将致力于帮助读者掌握在个人机上进行专业动画设计的技巧。

0.1 怎样充分利用本书

本书提供了在当前个人机上进行动画设计的切实可行的规则和大量的程序实例。读者可以通过阅读本书和进行动画软件设计,充分利用运行于 Windows 协同多任务操作环境下的个人机的强大功能。

0.1.1 三合一

《C 的 Windows 动画程序设计》实际上将三本书溶为一体。首先,它介绍了个人机上 Windows 图形应用程序的设计。其次,它为高水平的动画设计提供专业指导。最后,它拥有大量的注释程序清单。

0.1.2 源代码

本书提供了 19000 多行程序清单,这大约 800K 的源代码也在本书附带的参考盘中提供。读者可以将其粘剪贴到自己的应用程序中去。

0.1.3 解答

本书也是一本解决方法的总汇,它帮助读者获得商业化的编程,并巩固读者已掌握的技巧。这将使读者在竞争日益激烈的世界中获得成功。本书保持了以前 Windcnest/McGraur Hill 图形编程系列丛书的实用和方便的风格。这将帮助读者掌握 Windows 下的编程方法。

0.2 谁该使用动画

任何想改进软件的交互式能力的编程者或开发者都应该使用动画。即使简单的光点动画(指瞬时,非全屏的动画序列,系前人偶然命名)都可以使程序软件变得生动,使其演示效果达到世界级水准。

0.2.1 动画的实际应用

动画(及其孪生兄弟仿真和可视化)在许多领域得到广泛应用。PC 机动画的市场包括教育,仿真培训、娱乐、游戏软件展示、商业代理点购显示,基于磁盘的动画等等。

计算机动画还是建筑设计,工业设计和工程设计的有力工具。典型的动画序列包括穿行(walk-throughs)、掠行(walk-bus)、空中分列式、旋转器件和装配线序列。个人机上的动画还是仿真技术和可视化技术的重要组成部分。仿真和可视化的对象包括生物医学、工程学、

化学和数学。甚至也日益依赖于计算机为法庭动画来帮助分析和判断重大的案件。

计算机动画在电视节目制作中也日益发挥重要作用。这在电视和商业作为大众传播媒体的电视节目中都一样。动画参与节目的制作包括以下方面:节目时间表、片头字幕、商业宣传、特技效果、社团联络和产品促销。计算机动画它的适用领域日益拓广正迅速成为当今和未来软件库中不可分割的一部分。

0.3 谁应该使用本书

想要利用在全世界范围内发行量已达 1 千万份的 Windows 进行动画设计的人都应该阅读本书。目前,市场上出售 Windows 应用程序软件包已超过 5000 种。它们当中的大部分都利用了 Windows 软件包提供的强大动画能力。

如果读者对多媒体技术、磁盘广告、计算机辅助培训和交互式视频技术感兴趣,那么你就必须掌握本书提供的 Windows 动画技术。

如果读者对教学软件设计、辅导性软件的设计,或者软件示范程序的设计感兴趣,那么就需要了解本书所展示的 Windows 动画技巧。

如果读者想编写娱乐软件、游戏软件或是模拟软件,那么他应在自己的程序设计工具箱中加入本书所提供的 Windows 动画方法。

只要读者属于下列成员,就会通过阅读本书而获益非浅。

- 开发 Windows 图形应用程序和 Windows 动画应用程序的新手。
- 有经验的 Windows 图形程序的程序员和开发人员。
- 想要提高生产效率的工程管理人员。
- 正在为 Windows 应用程序环境选择软件工具的程序员,开发员(developer)和技术管理人员。
- 想在自己的 Windows 应用程序中使用动画的软件开发人员。
- 想要跟踪 Windows 的图形、动画、仿真和可视化发展趋势的研究人员,管理人员。
- 想要通过掌握 Windows 图形技术和动画技术而在市场中更好地为老客户服务,并吸引更多新客户的签约程序员(contract programmer)。
- 想要为其家庭客户和办公室客户扩展 Windows 图形特征的协同程序员(corporate programmer)。
- 想要在 Windows 图形领域和动画领域保持技术领先的开发者。
- 想要扩展自身的 Windows 动画设计技巧,对编程感兴趣的业余编程爱好者。

0.4 特殊性能

本书的精心编写使得它很容易移植到用户的编辑环境中。实用性和实效性是本书的宗旨。拥有它,读者就掌握了达到编程目标的方向。

0.4.1 学习工具

本书的主要目的是帮助读者编写 Windows 动画图形应用程序。它被设计成一个学习工

具,它是程序员的工具箱。然而,本书并不试图包罗万象,因为没有任何一本书能够对 Windows 动画编程的每一领域都做出详细的描述。本书着重实用性,因此列出了丰富的编程实例。众多的编程实例和免费程序源代码清晰地展示了所讨论的技术,大量的插图使得阅读更为方便。除此之外,本书还强调了图形的设备独立性问题。

完整性

本书完全独立,自我包容,没有作任何删节。读者可以完全依照本书而建立起每一个示例程序。为进一步方便读者,随书所附的参考上还提供了所有示例程序的源代码。

图标

本书中使用的图标位于页的左半部,用来标识文章中的重要部分。当读者正在查找所感兴趣的论题时,图标可以引起读者的注意。本书使用了两种图标:论题指示图标和程序指示图标。

论题指示图标

论题指示图标表示对文本的强调。它帮助读者注意论题及其它特别的信息,如图 1 所示。图标 tip 意味着提高编程效率的建议。图标 Fact 表示重要的背景知识。图标 Here's how 表示实用的编程规则。图标 Expert 为有经验的程序员及开发者提供适当的建立。图标 Sample App 置于程序清单的一侧,表示可以建立该示例程序并使其在下运行。图标 User's Guide 表示对运行和使用示例程序的指令的说明。图标 Demo 意味着对示例程序的性能特性和图形输出的讨论。图标 Code 代表对源逻辑代码,算法及过程的讨论。图标 Toolkit 表示参考盘的有关情况。图标 Disk 表示示例程序的磁盘操作情况,通常是关于动画演示期间,图像文件的存取。

代码指示图标

代码指示图标用于确认产品文件的程序清单,如图 2 所示。def 图标代表模块定义文件。h 图标代表含有定义常数函数原型的包含文件。rc 图标代表用于说明由示例应用程序所用的菜单和其他资源的资源文件。c 图标代表 C 源文件,该源文件中包含了示范程序的入口和核心函数。

屏幕图像

本书的每一个示例程序都伴有许多屏幕输出。这些屏幕输出是用激光打印机打印的窗口显示图像。通过这些图像可以使读者确信在自己的计算机系统上进行的演示和本书中对演示的性能说明是一致的。这同样意味着,读者可以不用计算机的帮助而独立地学习本书。

读者可以在旅行,上下班间隔或其它任何的闲暇时光利用这些打印图像方便地使用此书。但是,请记住,贯穿全书的这些屏幕打印图像并不能完全体现出示例程序所表现出的细致丰富的色彩、阴影和色阶。

插图

本书包含一套完整的手绘描图。它们是本书的重要组成部分,并和本书内容密切相关。它们提供了在绘画、设计和有关制作卡通的技巧的具体方面。实际的例子。这些插图帮助阐明了在其他情况下要用大量篇幅才能说清楚的概念和技术细节。

0.4.2 图形的约定

充分考虑到图形在传递信息方面的巨大力量,本书由始至终使用了大量的插图、图标以及屏幕打印图像。出版商和作者尽了最大的努力用图形来支持书中的要点。

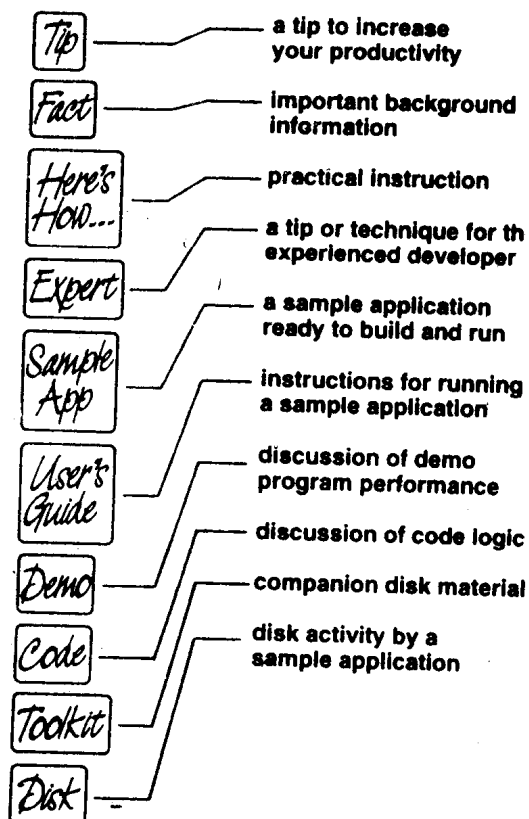


图1 书中通用图标,帮助读者获取相关材料

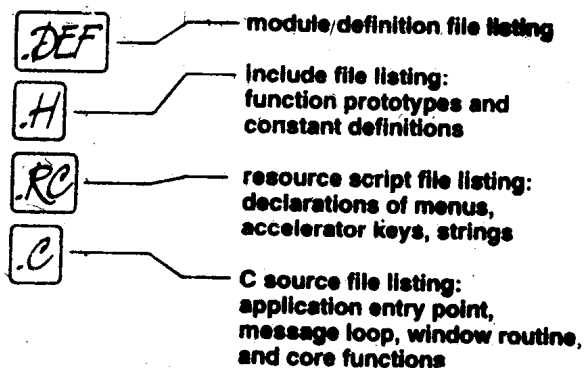


图2 书中列出的各个程序在空白处标以图标

总之,本书实践了它所宣称的宗旨,使用了大量的图形。也许在接触本书以前,你曾愤怒地发现,使用一本图形拙劣的书学习掌握 Windows 图形编程技术就像一只蚊子试图征服捕蝇纸一样。

0.5 本书的组织

本书经过了精心组织,为读者提供了 Windows 协同多任务环境下动画编程的实用、方

便的技术指导。本书的组织、布局并没有把任何教导式的学习方法强加于读者。读者可以选择最适合自己的学习方式:浏览各个主题挑出自己感兴趣的内容。或者系统地从头至尾地学习本书。不论读者采取什么样的学习方式,本书都能很好地为读者服务。前面的章节叙述了以后章节中高水平动画设计的基础知识,但是由于每一章都是按一个独立的模块进行组织,所以读者可以独立地学习任一章节。

本书分为四部,从总论出发,由顶向下进入各个专题。

0.5.1 图形编程技巧

书的第一部分介绍了基本的图形学编程技巧,其中包括:管理窗口显示区、图形独立显示、永久性图像、使用 bitblt(位块传递)。透明 PUT、三维造型和阴影,以及字体标题。图 3 的图像是第一部分样本应用程序产生的图像。

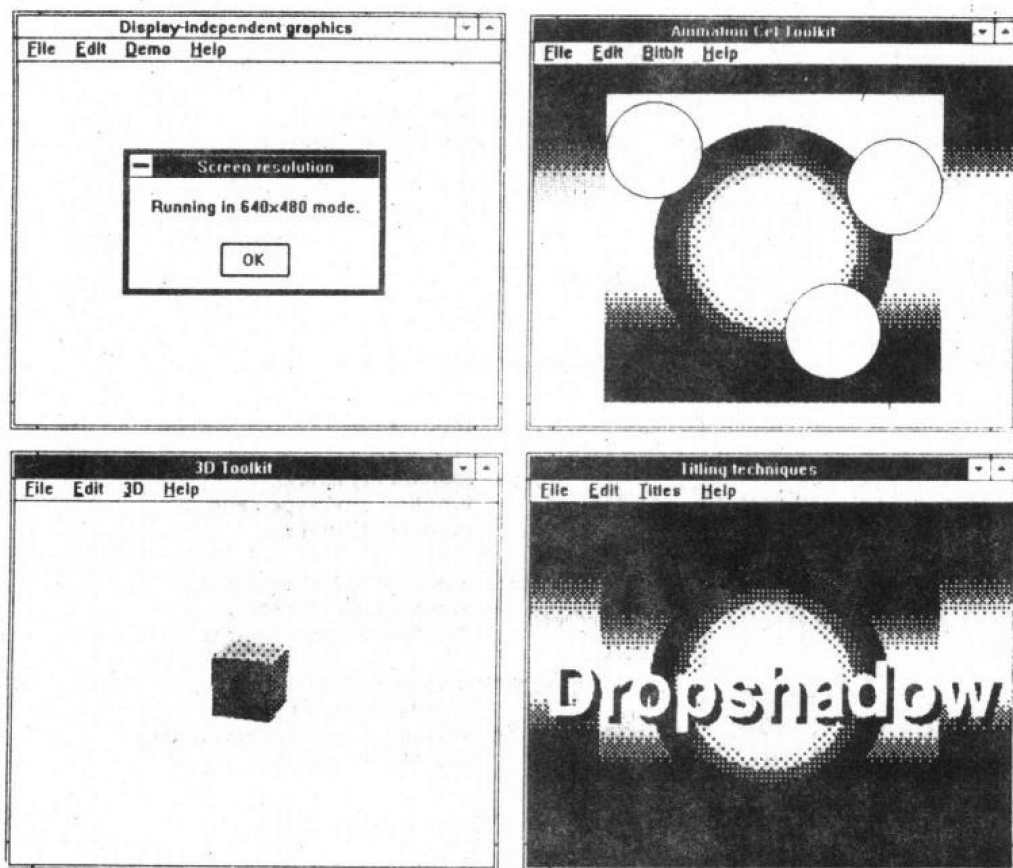


图 3 本书第一部分中的范例程序中使用的的基本编程技巧,如独立显示图形,bitblt,位图,三维造型,标题及文本输出

0.5.2 计算机动画

书的第二部分讨论了独立于硬件的动画实现方法,包括只要稍做修改就可以剪贴至任

意 Windows 动画应用程序中去的实时动画技术和帧动画机,如图 4 所示。

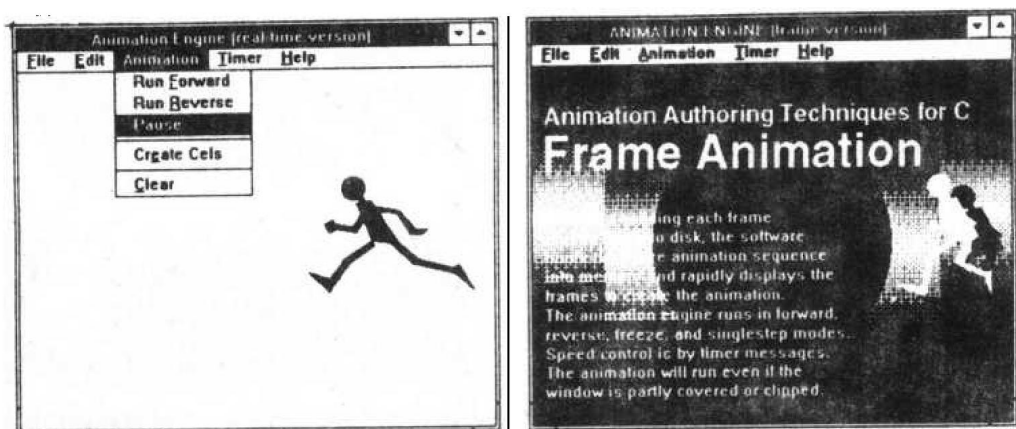


图 4 书中第二部分中的范例程序阐述了如何在 Windows 中硬件上重新配置动画机以创建实时动画与帧动画

0.5.3 图案动画

书的第三部分提供了编写 Windows 图案动画程序的实用规则,并用示例程序演示了图案动画。所提供的规则包括:人物动画图样设计透视、运动技术、背景移动、中介等等。图 5 显示了由这部分的示例程序所产生的图像。

本书的第四部分描述并展示了基于物理学原理的动画的一些重要领域。基于物理学原理的动画或多或少地遵循物理学的法则。图 6 所示的就是这部分中的示例程序的演示。

0.5.4 附录材料

附录材料为读者提供了在自己的计算机系统上构造示范程序的帮助信息。如果读者使用的是 Microsoft C,可参阅附录 A。如果工作环境是 QuickC for Windows,则参附录 B。相应地,使用 Borland C++ 参阅附录 C,使用 Turbo C++ for Windows 参阅附录 E,若用 WATCOM C 进行编译,则参阅附录 F。

附录 G 介绍了如何检测 Windows 源程序在不同的图形模式与不同的内存模式下的运行情况。附录 H 是有关声卡的内容。附录 I 中列出了一些动画厂商名。附录 G 列出了一些杂志、编目及目录名。根据它们可以查阅动画制造的软硬件工具的来源。

0.5.5 术语和索引

本书列出了重要单词和短语的术语汇编。读者可以通过它们更好地理解本书,并能对计算机图形学的编程尤其是动画编程方面有更深的领悟。索引可帮助读者立即找到感兴趣的文中内容。

0.6 关于程序清单

本书的重要组成部分之一就是源程序清单。所有列出的程序都在不同的图形显示模式



图 5 书中第三部分的范例程序阐述了某些创造性技术,如人物动画,图元动画,运动模糊,投影外形,背景移动,跑动循环,中介,关键帧及各种动画实现方法

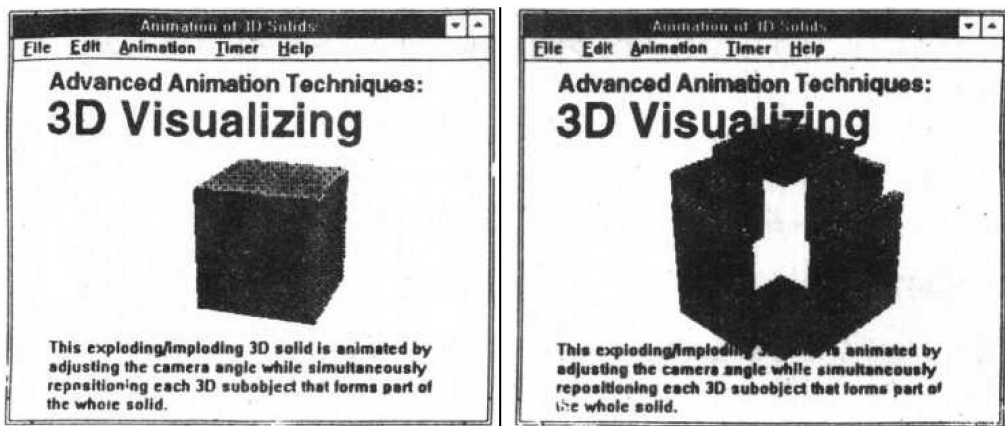


图 6 书中第四部分的范例程序阐述了各种仿真及模拟技术,包括行为动画,视点动画,前向动态及动态动画等

和内存模式下通过了严格的测试。

0.6.1 没有任何问题的程序清单

本书所列的程序清单是对激光打印机打出的程序源代码的直接再复制。读者可以确信程序清单是完整的,没有任何问题存在。书中所列的源程序代码和附带的参考盘上的源程序一模一样。

源代码格式

所有的程序清单遵循一致的组织格式,如图7所示。这种统一是为了使读者将注意力集中于程序的逻辑组织而不是外在的组织方式。

```

0001 case WM_PAINT: /* if image needs to be refreshed... */
0002 hDCPaint= BeginPaint(hWnd,&ps); /* load structure */
0003 EndPaint(hWnd, &ps); /* validate client area */
0004 if (PaintImage==zBLANK) break; /* if client area is blank */
0005 zCopyToDisplay(hDC); /* else copy hidden frame to display */
0006 break;
0007
0008
0009 /* ----- hidden frame operations ----- */
0010 hDC hFrameDC; /* memory display-context for hidden-frame */
0011 HBITMAP hFrame; /* handle to hidden-frame bitmap */
0012 HBITMAP hPrevFrame; /* default bitmap */
0013 BOOL FrameReady= FALSE; /* hidden-frame created? */
0014
0015
0016 /* ----- calculate screen position to center the window ----- */
0017 WindowW= (DisplayWidth - zWINDOW_WIDTH) / 2; /* horizontal */
0018 WindowH= (DisplayHeight - zWINDOW_HEIGHT) / 2; /* vertical */
0019 if (WindowW < 0) WindowW= 0;
0020 if (WindowH < 0) WindowH= 0;
0021
0022
0023 GlobalCompact((DWORD)-1); /* maximize contiguous memory */
0024 hDisplayDC= SetDC(hWnd); /* set the display-context */
0025 hFrameDC= CreateCompatibleDC(hDisplayDC); /* create frame... */
0026 hFrame= CreateCompatibleBitmap(hDisplayDC, zFRAMEWIDE, zFRAMEHIGH);
0027 if (hFrame==NULL)
0028 {
0029 LoadString(hInst, IDS_NotReady, lpCaption, MAX);
0030 LoadString(hInst, IDS_NoMem, lpMessage, MAXTEXT);
0031 MessageBox(GetFocus(), lpMessage, lpCaption, MB_OK);
0032 DeleteDC(hFrameDC);
0033 FrameReady= FALSE;
0034 return;
0035 }
0036 hPrevFrame= SelectObject(hFrameDC, hFrame); /* select the bitmap */
0037 zClearHiddenFrame(); /* clear the hidden frame */

```

Annotations on the right side of the code block:

- line numbers
- blocks are indented
- general comments
- specific comments
- display-independent graphics
- careful memory management
- careful error management
- status reports to user
- careful resource management
- persistent graphics

图7 书中的范例程序清单为 Windows 动画编程提供了操作指导,源代码具有较强的可读性,可直接用于读者自己的软件工程中

行号

程序清单的每一行都有行号。文章中在涉及到表达式说明和程序段时都用到行号。清单中的行号是在打印时加上去的。若读者想把源程序键入到自己的计算机中去,请不要把行号也加进去。参考盘的源文件中不带有行号。

0.6.2 程序的稳定性

书中所有的示例程序都一致地遵循一套惯例。这套惯例不仅使学习本书变得容易,同时也减小了与运行在读者计算机系统上的其他 Windows 应用发生冲突的可能。

一致的用户界面

如图8所示,每一示例程序使用几乎完全相同的用户界面、标题条、菜单条以及菜单结构。在运行时,示例程序通过消息框提供提示信息。

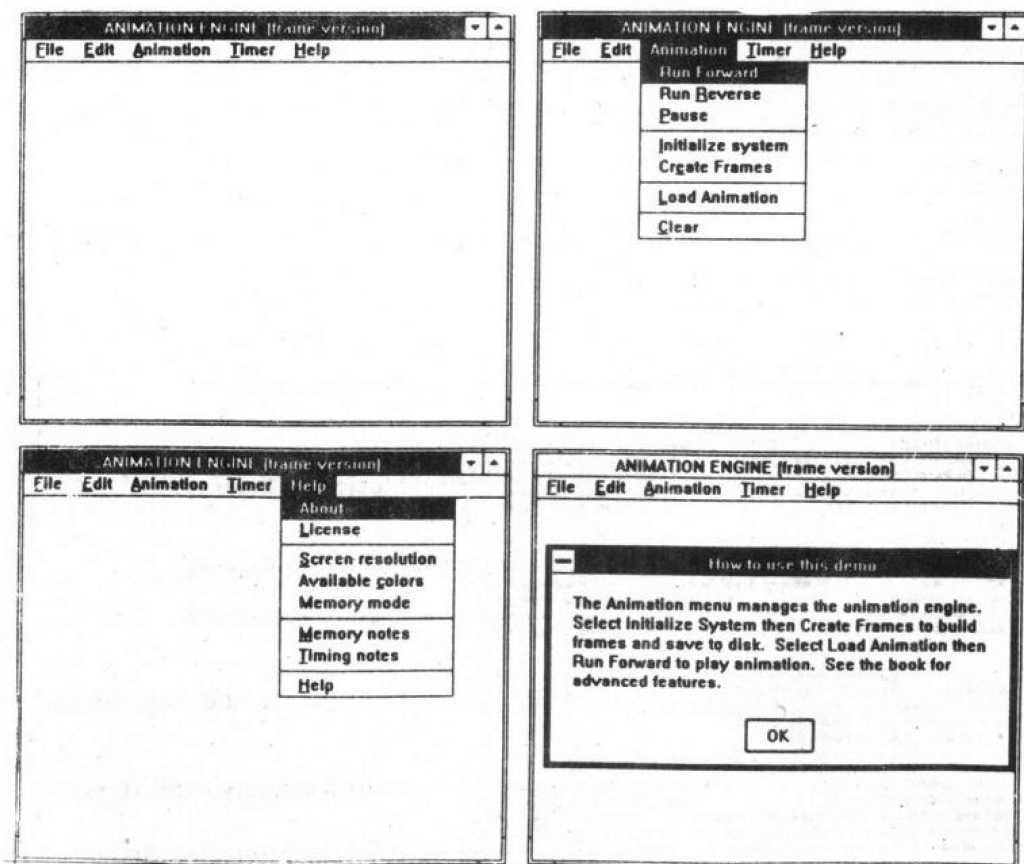


图 8 书中的范例程序具有相同的用户接口,使读者可以将精力集中于图形编程上

一致的演示帧

如图 9 所示,每一示例程序使用同样的演示帧。程序的逻辑使得窗口显示区总是出现在屏幕的中央。以屏幕像素点来衡量,每一示例程序所用的是一个标准大小的显示帧。这种一致性可以确保图像文件是以预知的尺寸大小存入磁盘的。此外,大小为 400×300 像素的二维图像区域可以使动画图很容易适合于使用 4:3 比率的电视和录像。

持续图形

每一示例程序都提供了持续图形。这意味着读者可以在动画演示时移动演示窗口。当另一 Windows 应用程序窗口部分覆盖了窗口显示区没有被覆盖部分时动画演示仍会继续。如果拖动窗口以至于部分窗口被屏幕的边界所剪取,没有被剪取的部分的动画演示仍会继续。

实例

一个应用程序正处于运行状态,如果再次双击其图标试图二次激活该应用程序,系统将对此进行检测并将控制权返回早先的运行程序。更广阔的意义上说,演示程序具有排它性。例如,如果在运行 BLINK.EXE 时试图加载 BOUNCING.EXE,程序逻辑将确保 Windows 把控制权返交给 BLINK.EXE。这种管理方式在 Window 内存管理器收到大量系统的有关

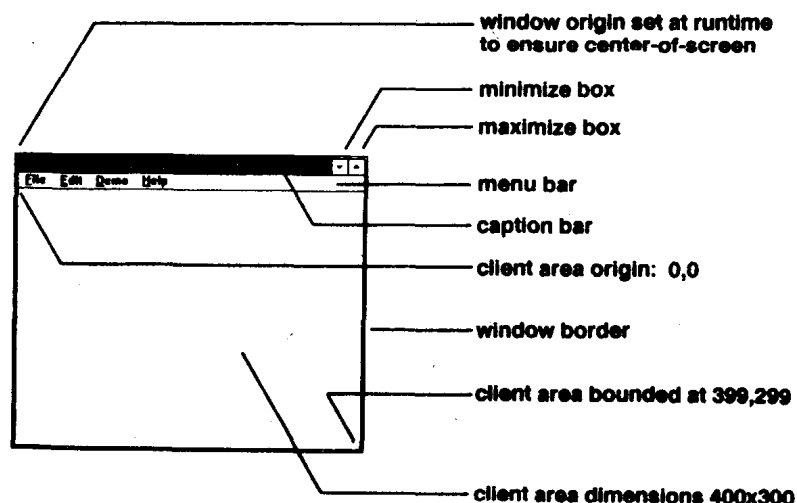


Image Storage Requirements

640x480x16-color VGA
or any 4-bit 16-color mode

$400 \times 300 = 120,000 \text{ pixels}$
 $120,000 \times 4 \text{ bpp} / 8 = 60,000 \text{ bytes}$

图9 书中的每个应用程序窗口大小均为横纵比为 4:3,可方便读者开发 Windows,多媒体 PC,录像带输出等各种应用程序

数据段切换的内存指令时,可以使计算机系统免于崩溃。

实际的惯例

Windows 应用程序的常量和变量的命名惯例并没有在本书的示例程序中严格遵守。此外,错误跟踪以及对 GDI 函数返回值的检查也没有严格地执行。示例程序没有试图截取 Windows 3.0 环境下不可恢复的应用错误(VAE)或是 Windows 3.1 环境下的一般保护性失败(GPF)。本书所支持的是应用程序原型化的设计环境,在 Windows 惯例方面有所放松。这种设计环境对教授本书和学习本书都很方便。

0.6.3 硬件兼容性和软件兼容性

如果用户使用的是 Windows 3.0 版、3.1 版或更高版本的规范平台,那么用户的硬件和操作系统是和本书所列的示例程序兼容的。

正确配置

本书的所有示例程序都在装备有 33MHZ Intel 80386 的微处理器的具备 4MB 内存, VGA 显示系统的计算机上创建并通过测试的。当 Windows 运行在 VGA 显示模式下,它使用缺省的 640×480 , 16 色模式。此外,示例程序还在下列 Windows 显示模式下通过测试。

- $640 \times 480 \times 16$ 色 VGA
- $640 \times 480 \times 2$ 色单显 VGA 和 MCGA
- $800 \times 600 \times 16$ 色 SVGA
- $1024 \times 768 \times 16$ 色 SVGA XGA 和 8514/4
- $720 \times 348 \times 2$ 色 Hercules

- 640×350×16 色 EGA
- 640×350×2 色单显 EGA
- 640×480×256 色 SVGA、XGA 和 8514/A

本节的所有示例程序还在下列的 Windows 运行内存模式下通过了测试。

- 实模式
- 标准模式
- 增强模式

为了实现标准模式下的示例程序演示,使用了装备有 16MHZ Intel 80386SX 微处理器、2MB 内存的计算机。计算机速度、可用内存的大小、计算机的图形能力都是影响动画演示性能的因素。不能期望只有有限可用内存的 16MHZ 计算机和具有 4MB 或 8MB 内存, 25MHZ 或 33MHZ 的计算机具有同样的演示性能。读者可参考第六章 Windows 下的动画中更详细的讨论。

0.6.4 如何使用代码为己所用

读者可以在自己的 Windows 应用程序中使用本书提供的源代码,但是要接受图 7 中所列的授权协议和限制保证。源代码是免费提供。但是,请记住,本书中的示例程序是经过改进的原型,他们并不是用于商业销售的产品代码。而且,因为数学的需要对程序进行的优化并不是针对速度的提高和如何有效利用内存。

读者可以通过将本书提供的程序代码作为编程的起点来提高编程效率。将程序粘贴到自己的应用程序中可以很快地建立起程序并将其投入运行。这种照菜单点菜的方式使用户在竞争中获益非浅。这有助于用户在上级或客户规定的期限之前完成任务。

0.7 使用本书需要什么

如果读者使用的是通用 C 或 C++ 编译器来产生 Windows 应用程序,那么使用本书应有的软件条件已经具备。

0.7.1 软件

使用下述软件包之一即可构造所有的演示程序:

- Microsoft C 及 SDK
- Quick C for Windows
- Borland C++
- Turbo C++ for Windows
- Symantec Zortech C++
- WATCOM C