

游戏编程大师系列

Delphi

经典 游戏程序设计40例



本书内附
范例光盘

日高 徹 著 博硕文化 译

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Delphi

经典游艺程序

设计40例

日高 徹 著
博硕文化 译

中国铁道出版社
2004年·北京



北京市版权局著作合同登记号: 01-2003-5304 号

版 权 声 明

本书中文简体字版经 Japan UNI Agency, Inc., 由 Softbank Publishing, Inc. 授权中国铁道出版社出版 (2003), 任何单位或个人未经出版者书面允许不得以任何手段复制或抄袭本书内容。

Delphi ゲームプログラミングのエッセンス

Copyright©2002 by 日高 徹

Simplified Chinese translation rights arranged with Softbank Publishing, Inc. through Japan UNI Agency, Inc., Tokyo.

图书在版编目 (CIP) 数据

Delphi 经典游戏程序设计 40 例 / 日高 徹著; 博硕文化译. - 北京: 中国铁道出版社, 2004.6

(游戏编程大师系列)

ISBN 7-113-05984-8

I. D... II. ①日... ②博... III. 软件工具 - 程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2004) 第 051512 号

书 名: Delphi 经典游戏程序设计 40 例

作 者: 日高 徹著 博硕文化译

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 郭毅鹏

责任编辑: 苏 茜 黄园园 翟玉峰

封面设计: 清风书坊 江 涛

印 刷: 北京鑫正大印刷有限公司

开 本: 787 × 960 1/18 印张: 35.25 字数: 710 千

版 本: 2004 年 11 月第 1 版 2004 年 11 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-05984-8/TP · 1232

定 价: 60.00 元

版权所有 侵权必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

出版说明

本书通过专题实例的方式来介绍各种游戏的程序设计技巧，全书共分为 3 篇，分别为“绘图篇”、“小型人工智能篇”、“简易工具篇”，用 40 个不同的专题实例进行讨论，并以 Delphi 为最终实现。相信读者可以通过本书从中学到不同类型游戏的设计思维与技巧。

随书所附光盘中 Delp_Ex40 文件夹的内容为书中所涉及的简体中文版的相关范例，J_Delp_Ex40 文件夹的内容为日文原书所附的相关范例。

本书由 Softbank Publishing, Inc. 通过 Japan UNI Agency, Inc. 提供版权，并由博硕文化翻译，由中国铁道出版社计算机图书项目中心审选。宋美娜、张益、程亚桥、房雅丁、郭炜、许可等同志完成本书的整稿工作。

编者

2004 年 5 月

目 录

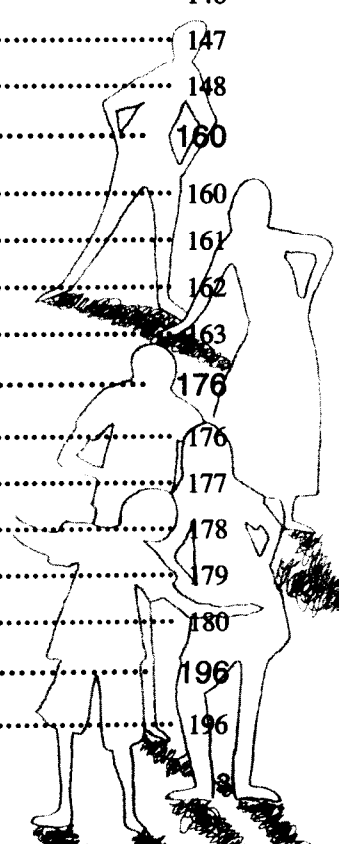
绘图篇	1
应用范例 1 可在背景底下移动的 ICON	2
● 将图形图像放大	2
● ICON 的设置	3
● 使用 Timer 组件	4
应用范例 2 由小而大的图画世界	8
● 准备图样	8
● 图样的基本规格	9
● 数组的初始设置	9
● 点阵图的区域的功能	11
● 点阵图的复制	11
● 画面的切换	13
● 最后修饰的重点	14
应用范例 3 图样的重叠与多重地图 (Multi Map)	21
● 图样重叠的基础知识	21
● 逻辑运算	22
● 制作去除用模板点阵图	23
● 何谓多重图像	25
● 执行范例程序以确认结果	25
应用范例 4 大型图像的 4 个方向滚动	35
● 各种滚动	35
● 滚动的世界	36
● 甜甜圈型世界地图的滚动原理	37
● 按钮与计时器的功用	38
应用范例 5 复合图案的时间差移动	50
● 以眼睛确认时间差移动	50
● 设置独有的 record 类型	51
● 程序的程序化	52

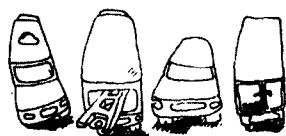


CONTENTS 目录

● 各自移动的技巧	53
应用范例 6 向真正的简易零件贴图挑战	62
● 何谓简单的零件贴图功能	62
● 画面显示的基础系统	63
● 点阵图与坐标	63
● 新记录类型的内容	64
● 在背后活跃的程序	65
● 力求简单	66
应用范例 7 星星的诞生与陨落	77
● 处理大量的角色	77
● 寻找诞生的星星	78
● 活跃的随机数	79
● 以计数器提升加速度	79
● 调整平衡性的重要性	80
应用范例 8 多重滚动	91
● 何谓真正的多重滚动	91
● 虚拟多重滚动	92
● 循环滚动的实质内容	93
● 将调色板统一起来	93
● 简易零件贴图功能与去除操作	94
应用范例 9 可大可小可随处移动的聚光灯处理	101
● 看不到的内容	101
● 聚光灯处理演算法	102
● 聚光灯的坐标与半径	103
● 程序的重点	103
● 各种用途与形状	105
应用范例 10 自动滚动功能与简易零件贴图	116
● 单向滚动的原理	116

● 简易零件贴图功能与去除操作	117
● 跳动的秘密	118
● 角色的黑影	119
● 停止滚动	119
应用范例 11 零件贴图中的优先顺序	129
● 制定显示顺序的原则	129
● 类似零件贴图管理法	130
● 显示顺序的实际状态	131
● 为了便于分析程序	132
应用范例 12 各种化妆显示画面特效	145
● 化妆显示①	145
● 化妆显示②	146
● 化妆显示③	147
● 化妆显示④	148
应用范例 13 各种卸妆效果的处理	160
● 卸妆①	160
● 卸妆②	161
● 卸妆③	162
● 卸妆④	163
应用范例 14 各种画面改变的化妆法	176
● 画面改变与画面显示的不同	176
● 画面改变的化妆法①	177
● 画面改变的化妆法②	178
● 画面改变的化妆法③	179
● 画面改变的化妆法④	180
应用范例 15 热气飘浮的画面	196
● 画面摇动的效果	196

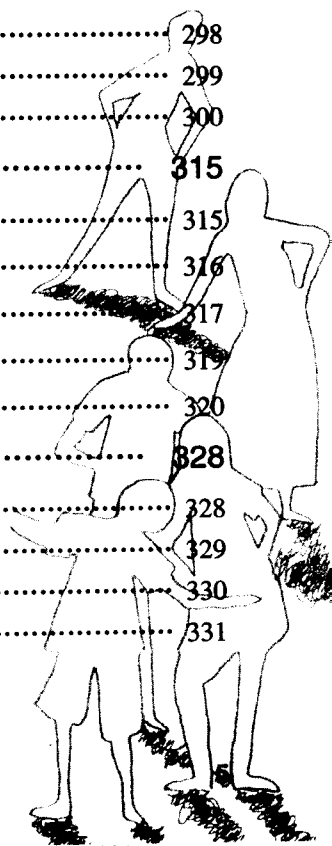


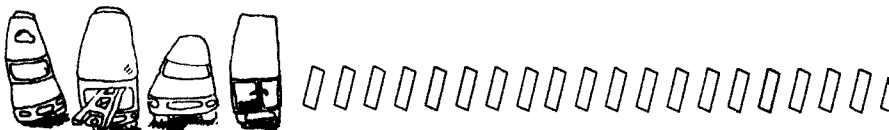


CONTENTS 目录

● 基本原理为画面滚动	197
● 本应用范例独有的特征	198
● 变量的功用	198
应用范例 16 马赛克处理	206
● 确认马赛克的实际状态	206
● 马赛克处理的机制	207
● 检查鼠标的 ON-OFF	207
● 改变马赛克的大小	208
应用范例 17 碰撞检查与事后处理	217
● 碰撞的认定	217
● 碰撞判定基准	218
● 碰撞检查的时机	219
● 动画处理	219
● 各角色记录类型的功能	220
应用范例 18 线条光线的威力	231
● 光线的原貌	231
● 使线条看起来像是光线的技巧	232
● 程序设计上的问题	233
● 各变量的内容	234
应用范例 19 闪电	244
● 闪电的原理	244
● 闪电光线的寿命	245
● 靠近闪电程序	247
● 闪电控制的秘诀	248
应用范例 20 对应于大画面的大闪电	259
● 为何要制作大闪电	259
● 追求自然的闪电	260
● 随机数的控制	261

● 闪电专用的程序	262
应用范例 21 立体迷宫	269
● 不迷路的方法	269
● 立体化的规则	270
● 4 个方向的处理	272
● 立体化程序的未来	273
应用范例 22 残留图像效果	282
● 用途视创意而定	282
● 寻找相同的轨迹	283
● 残留图像显示的实际状态	284
● 移动方向与移动效果	285
应用范例 23 行动的记录与重现	298
● 体验过去与未来	298
● 示范与随机数	299
● 负责记录与重玩操作的配角	300
应用范例 24 XOR 的攻击效果	315
● XOR 的特征与颜色变化	315
● 攻击效果①	316
● 攻击效果②	317
● 攻击效果③	319
● 攻击效果④	320
应用范例 25 数字钟	328
● 近似蓝色液晶的数字钟	328
● 读取时钟	329
● 数字显示的机制	330
● 液晶数字图案与显示位置	331





CONTENTS 目录

应用范例 26 Image List 的实力与用法 338

- 何谓 Image List 338
- 比较两者的实力 339
- Image List 的用法 340
- 如何活用 Image List 341

小型人工智能篇 353

应用范例 27 残留的轨迹是抛物线 354

- 算术式的抛物线 354
- 抛物线是相对坐标 356
- 确认动作内容 357
- 整数与实数 358

应用范例 28 残留的轨迹是正弦曲线 367

- 三角函数 367
- 正弦曲线的指定法 368
- 操作正弦曲线 369
- 各变量的功用 370

应用范例 29 残留的轨迹是钟摆线 379

- 钟摆的动作 379
- 钟摆与三角函数 380
- 抑制钟摆的摆动 380
- 钟摆程序的实际状况 381

应用范例 30 残留的轨迹是圆形运动 392

- 圆的轨迹也是三角函数 392
- 调整圆形运动 393
- 从程序的观点来看移动方向 394

应用范例 31 白点是目标 404

- 为了瞄准目标 404

● 单位为弧度	405
● 亲眼确认瞄准结果	406
● 没有误差的程序	407

应用范例 32 人畜无害的猎点 414

● 猎点的意义	414
● 目标点逃不掉	415
● 如何求出追捕方向	416
● 追捕能力的调整	417

应用范例 33 点的生存竞争 425

● 数个点的紧急出动	425
● 目标的选择基准	426
● 靠近用的程序	427

应用范例 34 RPG 中的敌人的出现时机 436

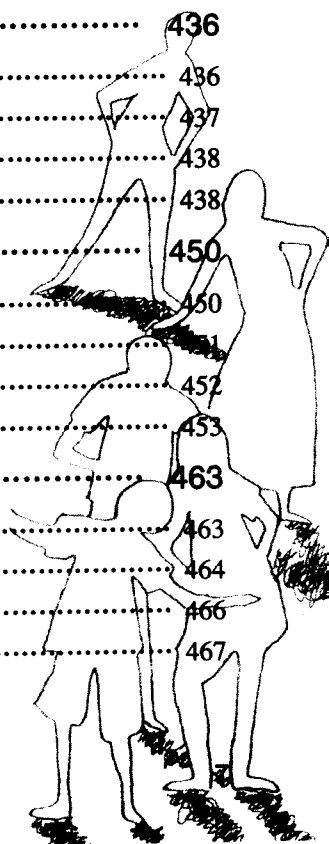
● 区域与出现的时机	436
● 敌军出现的计算基准	437
● 确认个别区域的出现几率	438
● 均衡是程序的生命线	438

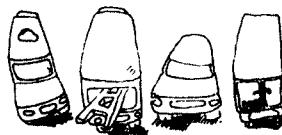
应用范例 35 半自动制作迷宫 450

● 迷宫的规格	450
● 迷宫制作的秘密	451
● 迷宫分割的规则	452
● 体验半自动制作迷宫	453

应用范例 36 不可思议的房间与迷宫 463

● 体验不可思议的迷宫	463
● 房间与通道的位置	464
● 通道的基本方针	466
● 通道工程的现场操作	467





CONTENTS 目录

应用范例 37 在迷宫中追赶对手 481

- 适当的行走 481
- 一路追赶对手 482
- 红白迷宫运动会 483
- 追赶程序 484

应用范例 38 与商店老板拼智慧 495

- 杀价高手就是购物高手 495
- 智慧的真面目 496
- 关于单选按钮组 498

简易工具篇 513

应用范例 39 点阵图编辑器的世界 514

- 可以玩的点阵图编辑器 514
- 操作 2 个 Form 画面 516
- 追加菜单功能表 516
- 版本的升级 517

应用范例 40 动画检查器 535

- 居中的剪贴板 535
- 动画检查器的用法 536
- 工具的扩充性 537

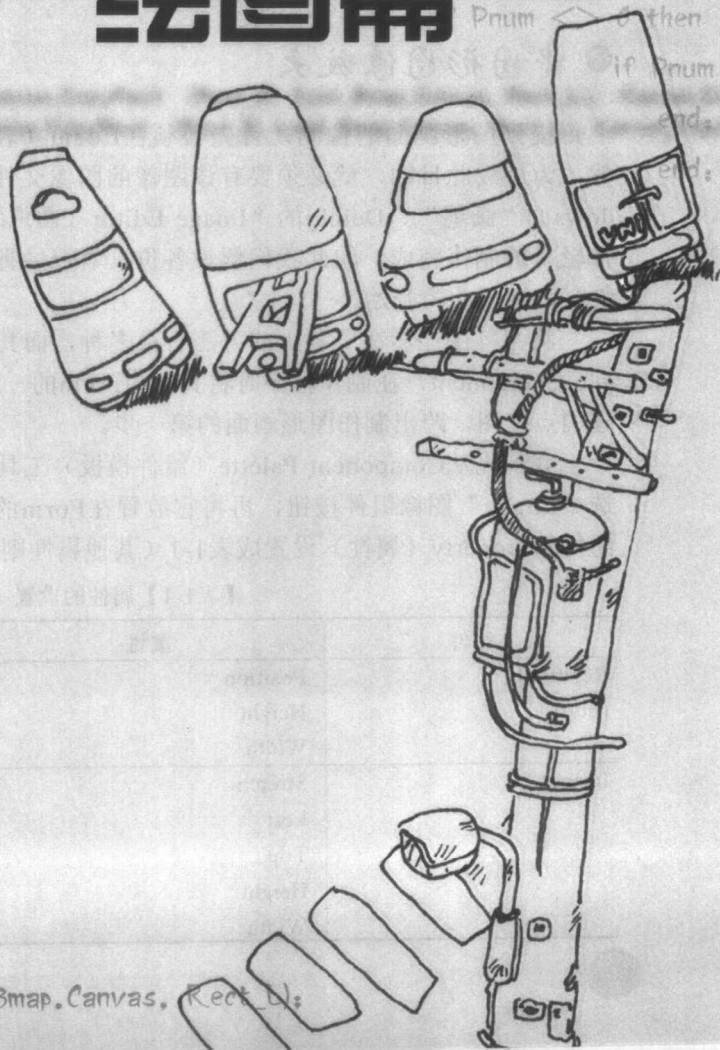
Delphi

经典

游戏程序

设计

绘图篇



vas.CopyMode := cmSrcCopy;

rap.Canvas.CopyRect(Rect_D, Load_Bmap.Canvas, Rect_U);

应用范例 1

可在背景底下移动的ICON

在Windows操作系统环境下，画面模式并没有“文字画面”与“图形画面”的分别。由于包括文字的所有内容皆以图形画面显示，因此我们不需要只能处理文字的画面。

不过如要编写程序或检查程序的执行有无错误时，则多以文字进行确认，当然这些文字内容也是图形画面的一种，但文字终究也只是文字而已。我们试图从正面来控制图形画面。

● 将图形图像放大

提到图形画面的控制，首先必须在Form（表单）上显示适当的图像。当然，为达成此目的，就必须要有该图像的图像文件，而这种图像文件可用Windows的“画笔”、Delphi的“Image Editor（图像编辑器）”及其他各种绘图软件轻易的制作完成。因此我们假设各位读者已经拥有点阵格式的文件（BMP文件），并以此为开始。

将点阵图显示在屏幕上的方法有很多种，而其中最简单的方法就是使用Image Component，在制作程序时将其作为Form的一环。接下来我们使用Delphi内建的点阵图，跨出制作图形画面的第一步。

我们先从Component Palette（组件模板）工具条的“Additional”标签中点选“Image”图像组件按钮，再将它放置在Form的左上角。此时的Form与图像组件的Property（属性）设置成表1-1（其他属性则用默认值）。

【表1-1】属性的设置

组件	属性	值
Rei40_01 (Form1)	Position	poScreenCenter
	Height	480
	Width	640
Image1	Stretch	True
	Left	0
	Top	0
	Height	432
	Width	592



应用范例1 可在背景底下移动的ICON

然后再点选这个Image1图像组件，并利用“Object Inspector”上的“Picture”项，按下“...”按钮后，屏幕会显示“Picture Editor”对话框。这时我们就Load（载入）Delphi所内附的文件（下面为安装Delphi后的默认路径）。

\Program Files\Common Files\Borland Shared\Images\Splash\256Color\Factory.bmp

在“Picture Editor”对话框Load进来后，排烟的工厂图像显示在Form中时，会比之前设置还大。这个图像将成为接下来要执行的程序的背景。

此外，文件位置（磁盘或文件夹等）会因各操作系统环境的不同而不同，存取文件时，请配合操作系统环境，适当的处理。

Form的相关属性方面，大部分都在程序上设置好了。这样做是为了防止因不同的显示模式产生不良的执行结果。但为了能够确认执行程序时的环境，在设计Form时也必须指定相关属性。本书所列的程序原则上皆配合执行Form来设置其属性。

Object Inspector（对象检视器）设置的Form大小会依据画面模式自动调整，因此画面模式如与本书的开发环境（1024×768点）不同时，就必须重新设置Form属性。同时也要检查放在Form上的组件是否隐藏起来、Form大小（640×480点）与所用的组件的值是否与指定的内容相同。

● ICON的设置

背景图形完成之后，再准备另一个组件，再以同样的步骤将其放置在Form上。放在哪个位置并无任何限制，但需先设置好表1-2所示的属性。如仅设置AutoSize=True，则显示大小（Display Size）如有变化时，便无法适时响应。

【表1-2】属性的设置

组件	属性	属性值
Image2	Height	32
	Width	32

然后打开“Picture Editor”对话框，再读取ICON用的文件。ICON的图案采用：

\Program Files\Common Files\Borland Shared\Images\Icons\Chemical.ico

不同于点阵图形，ICON图像文件由于可指定透明色，因此可自动的进行叠色处理，并显示在屏幕上。这简直是游戏机专用的子画面（sprite），用以显示

图像的硬件功能，但其实力究竟如何呢？我们决定定期的改变显示位置并实际执行，以观察其结果。

● 使用 Timer 组件

最后再点选Component Palette的“System”的Timer组件，将其放置于Form的适当位置。由于这个组件在执行时是看不见的，因此放在Form上的任何位置皆可。

Timer组件能以1/1000秒为单位进行设置，但设置的时间低于处理所需时间是没有任何意义的，因此这里设置成50/1000秒（Interval=50）。由于硬件型号（CPU的执行速度）与处理内容会影响设置的界限，因此请按不同环境作适当的设置。

程序的大部分皆编写于双击Timer组件时才会显示的begin~end;之间。由于我们仅将Image2的显示位置（Left,Top）改变成斜向移动，因此程序既短小且简单。如以行数来比喻的话，这个程序的行数相当于变量或背景图形（Image1）的起始设置的行数。这个程序的内容为装有药品的烧瓶（Icon 图）在背景图形的范围内移动（图1-1）。



会动的ICON图

【图1-1】执行画面

不过，若直接将此画面放在游戏内显示出来，恐怕还不太理想。我们只要执行这个程序就可了解，ICON只是ICON而已，只可用于对闪烁的图案不甚敏感的场所。

如欲结束程序的执行，可在Component Palette的“Standard”中选取Main



Menu Component（主菜单组件），并将其放在适当的位置，以便鼠标在窗口的标题栏（Title Bar）按下右键时，能够选择“关闭”命令。当然选择标题栏右侧的(X)按钮也可将程序结束，这是Windows的基本形式之一。

Column

Windows95问世以后，单击鼠标右键已成为基本操作的主要动作了。单击鼠标右键的作用多是用来终止某项操作的，因此Windows 3.1时代的用户对这种操作方式有一点轻视的心态。不过很多的电脑入门书及电脑杂志都已指出单击鼠标右键是基本操作的主要动作。

举例来说，当我们在文本文件上双击鼠标左键，就能打开编辑器软件编辑该文本文件，有时还能直接执行该文本文件。这时如单击鼠标右键，就可“打开”，或在“打开方式”选项中选择。而若在Autoexec.bat文件双击鼠标左键后，就算要看该文件的内容，也看不到该文件，因为电脑会自动的执行该文件。

换句话说，如要灵活运用Windows，按右键的动作是不可或缺的。本书的主菜单设置也是基于这种Windows的基本形式。

project Prei_01

◆ 属性的设置

对象	属性	属性值	内容
Rei40_01 (Form1)	BorderStyle	bsDialog	固定Form大小
	Caption	Moving Icon	标题或说明
	Position	poScreenCenter	执行时的显示位置
	Height	480	Form的高度
	Width	640	Form的宽度
Image1	Stretch	True	放大
	Left	0	Image1的X坐标
	Top	0	Image1的Y坐标
	Height	432	Image1的高度
	Width	592	Image1的宽度
Image2	Height	32	Image2的高度
	Width	32	Image2的宽度
Timer1	Interval	50	时间间隔
MainMenu1			主菜单