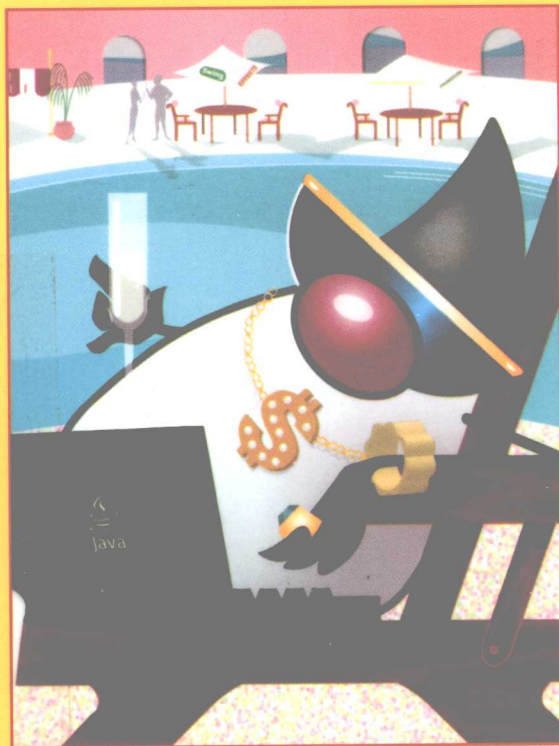


Java 动画、图形 和极富客户端效果开发

Filthy Rich Clients Developing Animated and Graphical Effects
for Desktop Java Applications



(美) Chet Haase 著
Romain Guy

蔡黄辉 等译

- 介绍如何实现更吸引人的图形和动画效果。
- 阐述与示例并举，用语规范标准，通俗易懂。
- 作者是来自Sun公司的Java SE组的客户端架构师，内容权威、准确可靠。

 Sun
microsystems



机械工业出版社
China Machine Press

Sun 公司核心技术丛书

TP312/2884

2008

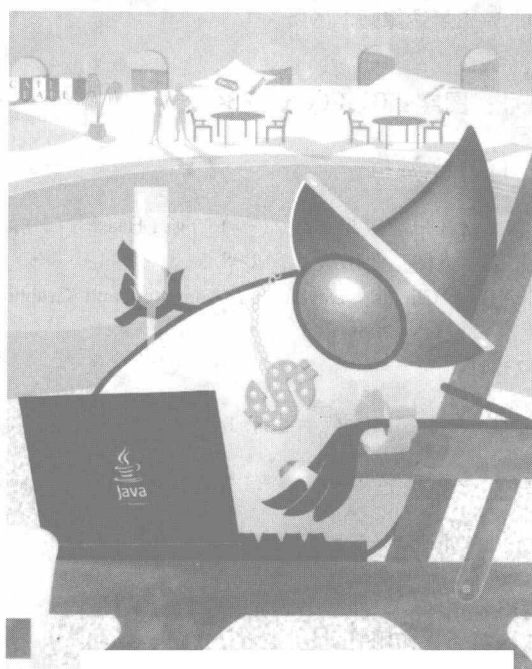
Java 动画、图形 和极富客户端效果开发

Filthy Rich Clients

Developing Animated and Graphical Effects
for Desktop Java Applications

(美) Chet Haase 著
Romain Guy

蔡黄辉 等译



机械工业出版社
China Machine Press

本书是一本关于利用图形和动画效果创建更好、更有效和更酷的桌面应用程序的书。本书讨论 Java 基本原理、Swing、Java 2D、图形、图形用户界面 (GUI)、动画、性能和基于这些基本原理的用于创建漂亮应用程序的高级效果。本书第一部分讲述 Java 图形和用户界面编程的基本概念。第二部分讲述 Java 2D 和 Swing 中创建极富界面时有用的更多高级主题。第三部分讲述动画的基本原理。第四部分介绍极富客户端效果开发及示例。

本书适合 Java 图形和动画效果开发人员使用,也可作为相关专业培训教材。

Simplified Chinese edition copyright © 2008 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Filthy Rich Clients: Developing Animated and Graphical Effects for Desktop Java Applications* (ISBN 978-0-13-241393-0) by Chet Haase, Romain Guy. Copyright © 2008.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Sun Microsystems, Inc..

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2008-1784

图书在版编目(CIP)数据

Java 动画、图形和极富客户端效果开发/(美)哈斯(Haase, C.), (美)盖伊(Cuy, R.) 著;蔡黄辉等译. —北京:机械工业出版社, 2008. 5

书名原文: *Filthy Rich Clients: Developing Animated and Graphical Effects for Desktop JAVA Applications*

ISBN 978-7-111-23841-6

I. J… II. ①哈… ②盖… ③蔡… III. JAVA 语言—程序—设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 048278 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:李东震

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2008 年 5 月第 1 版第 1 次印刷

186mm × 240mm · 24 印张

标准书号: ISBN 978-7-111-23841-6

定价: 49.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线 (010) 68326294

译者序

我们都知道用常规的 Swing 和 AWT 组件无须设置也可以开发 Applet 和 Java 应用程序，但是界面的美观程度却总是差强人意。

本书一开始介绍了图形和图像的基本原理及性能问题，然后介绍了如何让我们只需要对一些组件进行简单定制就可以取得令人惊讶的效果。

书中还介绍了如何通过现有的框架轻而易举地开发出一些传统的界面设计工具设计出来的效果（例如渐变、滤色镜、玻璃窗格等）和动画（例如平滑移动、淡入淡出、动态转换等）。最后还介绍了几个可以用来容易地开发出界面极其丰富的应用程序的工具和框架，例如 Timing Framework、Animated Transitions 等。

这本书由浅入深，从图形和 GUI 的基本原理到高级的图形渲染，再到动画及效果，最后还举了一个实际的开发项目来阐述如何开发图形极端丰富的应用程序。书中引用了很多的代码示例及插图，在讲解原理的同时结合实际的效果，使人很容易理解。

本书既适合于对开发 Java 应用程序感兴趣的新手，也适合有多年应用程序开发经验的程序员。

由于时间仓促，加上译者水平有限，书中难免有翻译疏漏或不妥之处，还请广大读者和同行不吝指正！

这里感谢我的妻子沈晓霞，感谢她在我翻译这本书的时候对我提供的支持，她还为这本书的校对提供了很多帮助。

参加翻译的人员还有：陈品珍、蔡德平、沈斌、张红、王海鹏、蔡红健、张建时、沈晓霞、沈安祥、陈品菊、杨文英、廖倩、陈凯金、陈品琴、王建平、施月平、施磊。

蔡黄辉

2008 年 1 月

序

这是一本关于创建漂亮应用程序的书。不仅是许多没有特色的标准外观下的文本框和按钮，而且是真的非常漂亮的应用程序。

很多年前，图形用户界面是由标准的外观规范规定的。采用这种方式是为了尽量集中应用程序中的所有 GUI 代码，这样可以容易地编写应用程序的文档（每个人都知道滑块是什么，所以不需要描述它）和围绕台式计算机相对较少的图形性能工作。

但是计算机行业与消费者之间最近十几年的碰撞导致在强调用户界面美观方面得到很大的提高：从广为人知的增加复杂系统的易理解性到吸引消费者的引人注目的酷，哪怕仅仅是简单的“哇！”，美学正在进入设计的所有方面。

把这一点与摩尔定律带给我们的显著增强的计算机能力结合起来，尤其当它表现在商品图形渲染硬件上时，您已经获得广泛的、有趣的设计可能性。

其中有许多的细微之处，从“怎么创建一个漂亮的界面”和“如何使得像素漂亮”到“如何才能快速地创建它”，本书讲述了所有这些主题及更多。就我而言，这是一种非常快乐的编程任务，我确信它也会给您带来快乐。

——James Gosling[⊖]

⊖ James Gosling 是 Sun 公司副总裁，Sun 公司研究院院士，是 Java 技术的创始人，以“Java 技术之父”而闻名于世。——编辑注

前 言

欢迎阅读本书。这是一本关于利用图形和动画效果创建更好、更有效和更酷的桌面应用程序的书。在 2006 年 JavaOne 会议上成功举行关于这个主题的会议后，我们开始编写这本书。这个会议研究了使用动画、高级 2D 效果和一些 3D 效果创建更丰富的应用程序。但是看起来关于这个主题我们能够讨论几天。理解为什么应该开发这样的应用程序、这些技术如何使它们工作和如何能够适当地开发支持极富客户端的效果，这些真是非常丰富的话题。

这本书因此诞生。我们在本书中开始讨论 Java 基本原理、Swing、Java 2D、图形、图形用户界面（GUI）、动画、性能和基于这些基本原理的用于创建漂亮应用程序的高级效果。

欢迎加入我们。它应该很有趣。

本书组织

本书自始至终有一个连续的流程，所以读者需要按照这个顺序阅读，至少需要理解这些内容是如何安排的。本书中有许多适于随机地阅读的代码片断和讨论，但是这些内容后台的技术可能与书中早先的讨论有关。这种关系出现时通常会注明，以便可以更容易地查阅早先的背景资料。

本书的最初意图是解释我们在本书最后演示的非常酷的效果。但是，据我们所知，还没有解释 Swing、图形、Java 2D 渲染和动画的背景的图书，而这些内容是理解这些效果如何工作所必须的知识。所以我们从头开始。我们先研究这些领域的基本原理，你逐渐掌握它们，以便阅读到这本书的最后时可以理解所有内容。^①

本书不但提供了许多可以用于创建极富客户端的时髦的示例效果，而且提供了它如何工作的所有知识，以便你可以独自继续研究。

第一部分：图形和图形用户界面基础

第一部分讲述本书其他部分始终使用的 Java 图形和用户界面编程的基本概念。图形和用户界面开发的全面描述超出了本书的范围，但是我们在本书中涵盖了足够使每个人熟练掌握 API、技术和理解稍后章节中创建这些元素所必须的详细资料。

如果读者已经完全理解 AWT、Java 2D 和 Swing，那么这一部分开头的内容可能有点老套。然而，当你继续阅读时，将依赖这些基本的概念。而且，本书自始至终有许多有趣的、深奥的技巧，它们应该对所有的桌面 Java 开发人员都有用。

第二部分：高级图形渲染

第二部分讲述 Java 2D 和 Swing 中创建极富界面时有用的更多高级主题。第二部分的前半部分讲述特定图形的合成、渐变和图像处理的技术。第二部分的后半部分讲述更多 Swing 相关

① 把这种方法看作启蒙的一个常规途径。

的技术：玻璃面板、分层面板和重绘制管理器。

第三部分：动画

极富客户端不是静止不动的，它是活的。它需要移动，它需要变化。它需要灵魂以便用户知道它的存在。好看只是成功的一半，成功还需要看起来是活动的。

第三部分讲述动画的基本原理，动画可以用来为应用程序带来生命。我们讲述动画图形和 GUI 的一些基础，讨论 Java SE 核心库中有助于开发动画的现有工具，还讨论 Timing Framework 库，它使得在 Java 中开发动画更容易。

第四部分：效果

第四部分依赖本书前面讲述的所有内容。效果是极富客户端的核心，它使得极富客户端有别于纯粹的富客户端。效果分为两个类型。第一个类型是静态效果，它为了使应用程序有更丰富的外观而使用了图形技术。第二个类型是使 GUI 移动的动态的或活动的效果。我们还讨论 Animated Transitions，这是另外一个动画效果，可以通过本书的 Web 站点上可下载的一个工具库激活它。这一部分的最后一章演示了如何开发一个简单的极富客户端，从最初的设计图表一直到各种效果的实现。

本书写作风格

我们为本书采用了一个非正式的编写风格，因为我们真地感觉正在和您（读者）交谈。我们两人中的一个经常在本书的段落中用到“我”这个词。这就是区分谁在编写这部分内容的一个技巧。当然，实际上没有关系，而且您可能也不在意。但是万一在意，下面是一个提示：Romain 编写的章节中图像和截屏图通常更吸引人，而在 Chet 编写的章节中有较多的脚注和原始文本。这些区别正好反映了两个人的个性：Romain 有很强的审美能力，他可以取得漂亮的图像，而 Chet 谈的非常多。

读者需要具备的条件

有 Java 语言和 Swing 的经验将对阅读本书很有帮助。本书不是这些主题的一本初级读物，而是假定读者熟悉 Java 和 Swing。然而，即使对于高级的 Swing 程序员来说，Swing 的一些渲染基础（对于要理解创建极富客户端很重要）可能也不是很容易明白。所以本书的前面几章解释 Swing 和 Java 2D 如何相互作用，以便创建我们在本书的其他部分研究的各种定制效果。

与本书配套的 Web 资源

我们已经在 Web 站点 <http://filthyrichclients.org> 上汇集了本书相关的信息。从本书中使用的工具库的演示到关于本书的其他信息和适当的相关技术，这个站点全都有。我们已经确定在本书中肯定没有错误，但如果发生了怪事（书中出现错误），Web 站点上勘误表会揭示这些错误。

Web 站点代码

本书有许多演示代码。在页面中有许多代码片断。大多数情况下，这些代码都是从发布在

Web 站点上的演示中复制过来的。那个地方还有在本书的 Web 站点上查询的项目名称。每个演示项目都包含可生成和运行的源代码，通过它们可以看到实际运行的应用程序，也可以把这些代码装配到项目中来使用它们。这些演示不是微不足道的可以忽略的内容，我们希望您到本书的 Web 站点检查这些演示。我们在编写本书时特别开发了这些相关的演示，Web 站点上软件中的内容与书中每一章的内容结合得很好。

Web 站点库

本书中也使用和讲述了工具库。对于我们开发的一些演示这些库是有用的，但是更重要的是它们可以在项目中用作单独的库。

可以在其他 Web 站点上正在开发的项目中获得这些库，如下所示，但是本书的 Web 站点 <http://filthyrichclients.org> 提供的版本与本书中使用的版本相匹配。书中包括下列库：

- **Timing Framework**：这本书中有两章详细地讲述了这个库（提示：寻找名字用“Timing Framework”开头的章）。这个项目正在由 <http://timingframework.dev.java.net> 开发，但是，在本书的 Web 站点上可以获得与本书中代码和描述相匹配的这个库的特定版本。
- **Animated Transitions**：本书的结尾（第 18 章）才讲述这个库，它的名称就是“Animated Transitions”。同样，在 java.net 上也可以获得这个项目，虽然在编写本书时它还没有发布。但是无论如何，在本书的 Web 站点上可以获得与书中描述相匹配的版本。

其他项目

有许多外面的项目适宜研究寻求极富客户端，但是本书和我们的演示只分别提到和使用一些特殊的项目：

SwingLabs：本书的演示和摘录上下文中提到的许多工具在 SwingLabs 的 Web 站点上都可以获得。一定要在 <http://swinglabs.dev.java.net> 上检查这些技术及其他技术。

JOGL：Java 对 OpenGL 的绑定库为利用 OpenGL API 及跨 Java 运行的大多数平台的硬件加速编写 3D 应用程序和效果提供了一个途径。可以在 <http://jogl.dev.java.net> 上找到 JOGL。

其他 Web 资源

我们两人都定期但经常更新我们的博客。通常，当需要知道更多关于图形、性能、Java 2D 和 Java 桌面客户端技术，请访问 Chet 的技术博客 <http://weblogs.java.net/blog/chet/>。要查看更多令人激动的图像，请在 Romain 的 www.curious-creature.org 英文博客上检查最新的 Swing 演示和讨论。

在这两个 Web 站点上将发现非常重要的信息，这些信息很好地补充了这本书。甚至可能无意地阅读到本书结局的预览。实际上，发布这些条目时，甚至我们自己也不知道会这样。

如果喜欢阅读本书的一些脚注，请检查 Chet 的非正式的风趣的博客 <http://chetchat.blogspot.com>。最后，如果您可以读懂法文，请访问 Romain 的法文博客 www.progx.org，它不可思议地把有趣的故事和编程建议结合在一起。

致 谢

作者想感谢曾经认识的每个人，但这将花费太多的时间和空间。于是，我们在此只感谢那些直接帮助影响这本书的技术、内容、风格和整理打印稿的人。

首先，我们要感谢这本书的评审人员。他们花了无数时间阅读原稿，尽量帮助我们使这本书尽可能优秀和正确。我希望有一张 Scott Violet 正在体育馆的仰卧举重设备之间阅读第 3 章的照片。一些 Sun 公司的专家也提供了帮助，他们是 Bob Eckstein、Scott Violet、Chris Campbell、Dmitri Trembovetski、Amy Fowler、Jim Graham、Phil Race 和 Hans Muller。我们也要感谢 Ken Russell，他帮助我们了解最初的 JavaOne 介绍中的 3D 内容，本书就是从这个地方诞生的。当我们理解错误时，技术的创建人和实现人纠正我们，这有点奇怪（有时甚至痛苦）。Jim Graham 的许多技术评审特别全面和有帮助。Bob Eckstein 在写作风格方面的评审使整本书流畅了许多。我们也从 Sun 公司外面获得许多帮助，其中包括 Jan Haderka、Jeff Kurtz、Guillaume Laforge、Ido Green、Natasha Lloyd、Daniel Klein、Dimitri Baeli、James Lemieux、Bill Snyder、Chris Brown、Jean-Laurent de Morlhon、Jan Bösenberg 和 Jean-Baptiste Freymann 的帮助。

我们还感谢那些帮助我们进行自由讨论、测试、设计评审和以许多痛苦的方式热诚批评我们的技术工作的人。没有他们不停的抱怨，像 Timing Framework 和 Animated Transitions 这样的工具将完全不一样。其中包括许多前面作为评审人提到的人员，还有来自 Sun 公司的 Vincent Hardy、Richard Bair 和 Joshua Marinacci 以及来自 Sun 公司之外的人员：Dieter Krachtus、Stuart Scott 和 Stephen Lum。另外有许多人对我们的工作提了很好的建议和反馈，但是我们要特别提到这些在各种 Timing Framework 特性方面提供建议和评审设计更改的人。我们也应该特别提到 Chris Campbell，因为他在 Timing Framework 中 Interpolator 和 Evaluator 上的工作有助于彻底简化非线性插值在这个库中工作的方式。

也要感谢 Craig Aspinall，他爽快地捐赠了他的 filthyrichclients.com 域名，这样对于要查找和本书相关的 Web 站点的人来说少了很多迷惑。filthyrichclients.com 和 filthyrichclients.org 将指向相同的 Web 站点并为那些要把他们的 Swing 应用程序变得极富的人提供许多资源。

我们感谢所有阅读我们的博客并给我们有用反馈的人。尽管这些技术非常有趣，在最后我们要提高开发人员的感受，所以听取外部世界关于开发人员的需要非常重要。没有我们敏锐的读者，本书的很大一部分将不可能完成。

也要感谢 java.sun.com 和 java.net Web 站点，本书的一部分内容基于他们发布的原创文章。我们要特别感谢 java.net 的编辑 Chris Adamson 和 Daniel Steinberg，他们帮助我们为这个站点写文章，我们还要感谢 Sun Microsystems 公司的 Laureen Hudson，他是 java.sun.com 的编辑，还是一个非常有趣的人。如果为这些站点写文章没有这么愉快，本书可能也不会现在存在。

哦，还要感谢您购买和阅读这本书。使用这些材料和编写与它相关的书很有趣，但当人们可以一起关注和使用时将更有趣。现在让我们帮助您使应用程序变得更极富。

目 录

译者序
序
前言
致谢

引言 1

第一部分 图形和图形用户界面基础

第 1 章 桌面 Java 图形 API: Swing、
AWT 和 Java 2D 7

1.1 抽象窗口工具包 7

1.2 Java 2D 8

1.3 Swing 8

第 2 章 Swing 渲染基本原理 9

2.1 事件 9

2.2 Swing 绘制 10

2.2.1 异步重绘请求 10

2.2.2 同步的绘制请求 11

2.3 Swing 渲染 12

2.3.1 paintComponent() 13

2.3.2 paint() 15

2.3.3 setOpaque() 17

2.4 双缓冲 17

2.5 线程 19

2.5.1 线程模式 20

2.5.2 定时器和事件派发线程 24

2.5.3 通过 SwingWorker 轻松使用
线程 24

2.5.4 线程小结 27

第 3 章 Graphics 基本原理 28

3.1 Java 2D 28

3.2 渲染 29

3.2.1 获取 Graphics 对象 29

3.2.2 图形状态 31

3.2.3 图形基本元素 48

第 4 章 图像 60

4.1 图像类型 60

4.2 BufferedImage 62

4.3 图像缩放 64

4.3.1 质量与性能的比较 67

4.3.2 getFasterScaledInstance(): 更快、
更好地缩放图像的方法 73

第 5 章 性能 76

5.1 使用剪辑 76

5.2 兼容图像 80

5.2.1 为什么您应该关心 80

5.2.2 什么是受管图像 81

5.2.3 生成兼容的资源 82

5.3 受管图像 84

5.3.1 夺取 DataBuffer 86

5.3.2 频繁地渲染到图像 88

5.4 中间图像 89

5.4.1 好主意 90

5.4.2 它是如何做的 90

5.4.3 注意事项 94

5.4.4 小结 95

5.5 最理想的基本渲染 95

5.6 基准 96

5.7 命令行标记 96

5.7.1 渲染 97

5.7.2 调试性能 99

第二部分 高级图形渲染

第 6 章 合成 101

6.1 AlphaComposite 101

6.2 AlphaComposite 的 12 个规则 102

6.2.1 Clear 103

6.2.2 Dst 103

6.2.3	DstAtop	104
6.2.4	DstIn	104
6.2.5	DstOut	104
6.2.6	DstOver	104
6.2.7	Src	105
6.2.8	SrcAtop	105
6.2.9	SrcIn	105
6.2.10	SrcOut	106
6.2.11	SrcOver	106
6.2.12	Xor	106
6.3	创建并设置一个 AlphaComposite	107
6.4	AlphaComposite 的常规用法	108
6.4.1	使用 Clear	108
6.4.2	使用 SrcOver	109
6.4.3	使用 SrcIn	110
6.5	AlphaComposite 的问题	111
6.6	创建自己的合成	112
6.6.1	Add 合成	113
6.6.2	实现 CompositeContext	115
6.6.3	组合这些像素	116
6.7	小结	117
第7章	渐变	118
7.1	两停顿线性渐变	118
7.2	规则渐变的特效	119
7.3	多停顿线性渐变	123
7.4	放射状渐变	125
7.5	优化渐变	127
7.5.1	缓存这个渐变	128
7.5.2	更巧妙的缓存	129
7.5.3	使用循环渐变的优化	129
第8章	图像处理	131
8.1	图像滤色镜	131
8.2	用 BufferedImageOp 处理一个图像	132
8.3	AffineTransformOp	134
8.4	ColorConvertOp	134
8.5	ConvolveOp	136
8.5.1	构造一个内核	137
8.5.2	影响边界	138
8.6	LookupOp	139
8.7	RescaleOp	140
8.8	定制 BufferedImageOp	142

8.8.1	基本的滤色镜类	142
8.8.2	着色滤色镜	143
8.9	滤色镜性能的一个注意点	147
8.10	小结	147
第9章	玻璃窗格	148
9.1	在玻璃窗格上绘制	149
9.2	屏蔽输入事件	152
第10章	分层窗格	157
10.1	使用分层窗格层	157
10.2	在单个层中排列组件	160
10.3	分层窗格和布局	161
10.4	可选的使用布局的 JLayeredPane	162
第11章	重绘管理器	165
11.1	当 Swing 变得太聪明时	165
11.2	接触 RepaintManager	166
11.3	基于 RepaintManager 的反射	167
11.3.1	为反射腾出空位	168
11.3.2	绘制反射	170
11.3.3	一个不智能、但反而更合适的 RepaintManager	172
11.4	小结	174

第三部分 动画

第12章	动画基本原理	175
12.1	与时间有关	175
12.2	基本概念	175
12.2.1	基于帧的动画	175
12.2.2	帧频	177
12.2.3	基于时间的运动	177
12.3	定时	182
12.3.1	“现在是什么时候?”	182
12.3.2	“到时候可以唤醒我吗?”	184
12.3.3	“再次唤醒我, 再一次, 再一次”	185
12.4	分辨率	190
12.4.1	System.currentTimeMillis() 和 System.nanoTime() 的分辨率	192
12.4.2	睡眠分辨率	194
12.4.3	定时器分辨率	197

12.4.4 关于分辨率的决定	199
12.5 活化 Swing 应用程序	200
12.5.1 活化图形	200
12.5.2 活化 GUI	202
12.6 小结	209
第 13 章 平滑移动	210
13.1 背景:为什么我的动画看起来很差	210
13.2 什么因素导致动画停顿及如何使它们平滑	210
13.2.1 定时(几乎)是一切	211
13.2.2 颜色:差异是什么	213
13.2.3 垂直回描:同步的感觉	219
13.3 演示:SmoothMoves	222
13.3.1 创建图形对象	222
13.3.2 运行定时器	223
13.3.3 渲染	224
13.3.4 渲染选项	224
13.4 小结	227
第 14 章 Timing Framework:	
基本原理	228
14.1 概述	228
14.2 核心概念	229
14.2.1 Animator	229
14.2.2 回调	231
14.2.3 周期	232
14.2.4 重复	233
14.2.5 分辨率	234
14.2.6 启动行为	234
14.3 内插	239
14.3.1 加速和减速	240
14.3.2 Interpolator	243
14.4 小结	252
第 15 章 Timing Framework:	
高级特性	253
15.1 触发器	253
15.1.1 概念和用法	253
15.1.2 触发器的超类	254
15.1.3 内建的触发器	255
15.2 属性设置器	261

15.2.1 PropertySetter	263
15.2.2 Evaluator	266
15.2.3 KeyFrames	268
15.3 小结	280

第四部分 效 果

第 16 章 静态效果	281
16.1 模糊	281
16.1.1 动机	281
16.1.2 简单的模糊	283
16.1.3 高斯模糊	285
16.1.4 性能诀窍	288
16.2 反射	289
16.2.1 动机	289
16.2.2 绘制反射	290
16.2.3 模糊的反射	290
16.3 阴影	291
16.3.1 动机	291
16.3.2 简单的阴影	292
16.3.3 逼真的阴影	293
16.4 高亮	294
16.4.1 动机	295
16.4.2 增亮	295
16.4.3 聚光	297
16.4.4 为了更容易阅读而文本高亮	298
16.5 锐化	300
16.5.1 动机	300
16.5.2 简单的锐化	301
16.5.3 非锐化屏蔽	303
16.5.4 锐化一个缩小的图像	303
16.6 小结	305
第 17 章 动态效果	306
17.1 运动	306
17.1.1 动机	306
17.1.2 前进、前进、走了	308
17.2 淡入淡出	310
17.2.1 动机	310
17.2.2 淡入淡出的策略	312
17.2.3 AlphaComposite 淡入淡出	312
17.2.4 颜色淡入淡出	313

17.2.5 交叉淡入淡出	314	18.6 Animated Transitions 的底层原理 ...	352
17.2.6 容易地生成淡入淡出	315	18.6.1 设置下一个界面——静静地 ...	353
17.3 脉动	315	18.6.2 使 Layout 不起作用——动态地改变布局	353
17.3.1 动机	315	18.6.3 使 Swing 高兴：性能	354
17.3.2 感觉我的脉动	316	18.7 小结	354
17.3.3 自动发光	318	第 19 章 一个极富客户端诞生了	355
17.3.4 跳动的脉动	321	19.1 Aerith	355
17.4 弹簧	323	19.1.1 运行 Aerith	355
17.4.1 动机	323	19.1.2 大概的位置	356
17.4.2 弹簧热	324	19.2 在纸上设计工作流程	356
17.5 变形	327	19.3 想像图	358
17.5.1 动机	327	19.4 在纸上设计界面	359
17.5.2 变形按钮	328	19.5 实体模型	360
17.6 小结	331	19.6 从实体模型到代码	361
第 18 章 Animated Transitions	332	19.6.1 使用层	361
18.1 使应用程序的状态动态延续	332	19.6.2 混合模式	363
18.2 Animated Transitions 库	334	19.6.3 使用导向线	364
18.2.1 动态的应用程序状态	334	19.7 我不是画家	364
18.2.2 GUI 的状态	334	19.8 选择合适的颜色	366
18.2.3 API	335	19.9 阅读与设计相关的书籍	367
18.3 例子：SearchTransition	336	19.10 小结	367
18.4 例子：再看一下 SearchTransition：定制	344	第 20 章 结论	368
18.5 例子：ImageBrowser	346	20.1 项目	368
18.5.1 GUI 的结构	349	20.2 Java 站点	369
18.5.2 图像和 ImageHolder	350	20.3 作者的站点	369
18.5.3 ScreenTransition	351	20.4 互相交流	370

引言

这不是一本关于数据绑定的书。^①

富客户端概述

我们应该描述极富客户端是指什么。但首先，我们需要描述什么是富客户端。富客户端是一个通常与桌面应用程序相关的短语。富客户端应用程序通常与瘦客户端或 Web 客户端应用程序对比，瘦客户端或 Web 客户端应用程序本质上是运行在服务器上的、有一个在用户桌面上浏览器中运行的简单前台的软件应用程序。

富客户端应用程序有更多位于用户台式电脑的编程逻辑和功能。极端地说，应用程序都可以是本地的，比如字处理应用程序或图片编辑软件。或者应用程序可以运行在客户/服务器，就像瘦客户端应用程序一样。数据仍然由服务器提供，重要的功能仍然来自服务器、数据库、网络或其他地方。但是本地应用程序比典型的 Web 客户端负责更多的逻辑、用户界面和交互。

Web 客户端和富客户端之间的区别是一个重要的区别，因为应用程序开发人员应该知道在决定他们的产品采用哪种路线时可以交替使用这两种方法。在不同的应用程序域、系统和技术之间交替方式各不相同，但主要特点归结如下：

Web 客户端

这些应用程序对用户来说看起来像简单的 Web 页面。它们最大的优点在于简单。它们比富客户端运行得更快，只花费服务器处理信息和通过网络把信息发送到用户计算机的时间。这些应用程序也趋向于有一个简单的、面向浏览器的图形用户界面（GUI）。然而，这种简单也需要付出代价。这种应用程序模式趋向于变得非常标准：每个页面都有内容、让用户填写的字段和把信息提交到服务器的按钮。这种交互模式倾向于面向成批处理：用户发送信息、服务器处理信息和结果页面发回给用户。在这种完成 - 发送 - 处理 - 返回 - 显示的应用程序模式下有可能出现很严重的交互延迟。

富客户端

这些应用程序有非常“丰富”的用户体验，它们利用了用户台式计算机的本地设备，例如图形加速卡，比 Web 客户端提供更有活力和特性更丰富的应用程序体验。有时候富客户端应用程序比简单的 Web 页面启动花更长时间，因为它们更接近于应用程序，而且它们的 GUI 比 Web GUI

① 我们认为应该明确承认这点。当我们在一个博客条目中描述这本书和它的大纲时，我们收到一个评论说实际上这本书应该是关于数据绑定的。虽然 Java 桌面应用程序与数据源交互是需要讨论的一个有趣和重要的领域，但这本书实际上根本不是关于数据绑定的。如果您认为本书是讲述数据绑定而打开它，那就应该合上这本书并查看附近架子上其他的书。或者现在改变观点并认识到阅读这本书将很有趣。

更棘手，因为这种应用程序比简单的面向 Web 的应用程序有更多的事件。它们的交互模型完全不同，因为更多的应用程序逻辑是在本地的，即使这个应用程序正在与后台的服务器通信。

最近，Web 客户端出现了一个新的名为异步 JavaScript 和 XML（Asynchronous JavaScript and XML, AJAX）的模式，在这种模式下，许多客户端 - 服务器的交互可以与用户与每个 Web 页面的交互并行处理。这种透明的客户端 - 服务器交互允许动态更新 Web 页面，而不是比较单调的完成 - 发送 - 处理 - 返回 - 显示的传统 Web 客户端应用程序模式。然而，这种模式仍然受限于应用程序所在的浏览器容器和许多浏览器模式为应用程序设置的约束，包括支持 JavaScript 特性的范围、浏览器的安全模式和应用程序所在的浏览器容器的物理 GUI。

AJAX 应用程序开始研究一些本书描述的极富特性，包括在它们 GUI 中的一些图形效果。这显然很好。我们相信这些特性可以创造更多可用的应用程序。但是，设定 AJAX 的浏览器约束仍然是一个 Web 客户端技术，而且目前我们只集中讨论富客户端模式。

极富客户端

极富客户端是我们杜撰的一个词，它指那些图形非常丰富出众的应用程序。它们一开始就吸引用户并一直使他们很兴奋。它们使得用户把这个应用程序告诉他们的朋友。简而言之，它们使用户真正享受他们的应用程序体验。上次享受使用一个软件应用程序是什么时候？您大概需要在生活中有更多的极富客户端。

极富客户端的关键是图形和动画效果。通过更吸引人的 GUI、给应用程序一个脉动的动态效果、使用户保持和应用程序合理流程连接的动画式转换，这些效果可提高用户应用程序体验的方式。

这里我不只讨论媒体播放机。我们讨论提高各种软件，从典型的基于表单的企业应用程序到免费的专业消费者应用程序。所有的应用程序都可以通过考虑用户体验和如何应用极富效果提高这些体验受益。

为了直观描述极富客户端的效果和稍后将在这本书中介绍的内容，让我们看一些截屏图（图 1 到图 7）。

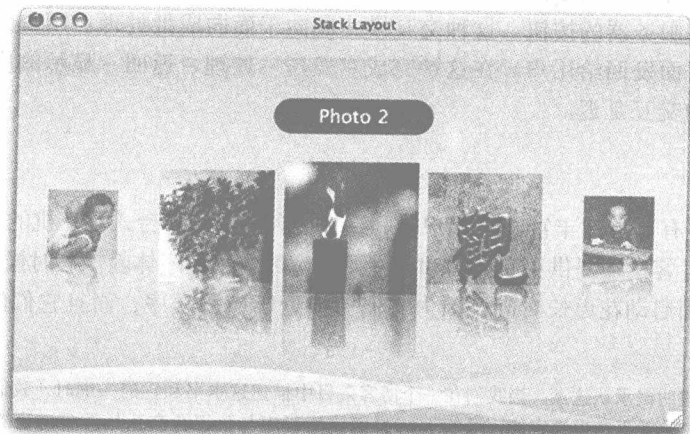


图 1 第 10 章演示了如何在用户界面中支持多层信息

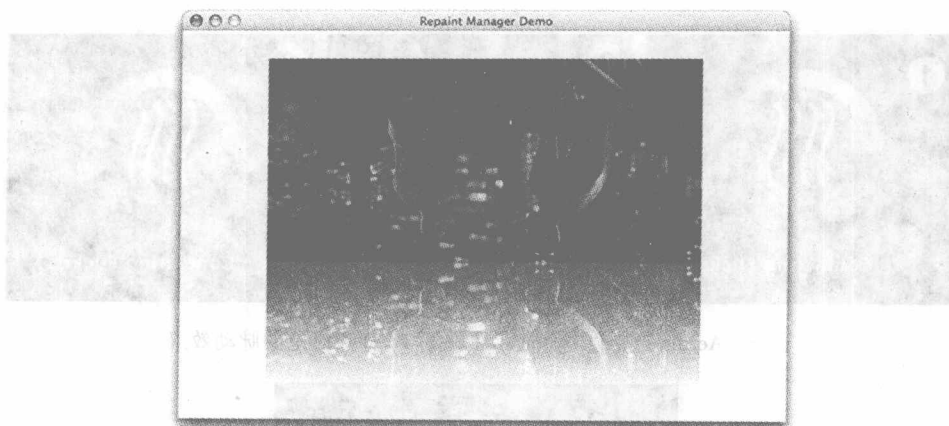


图2 第11章讨论的“反射”给应用程序带来现实主义和华美

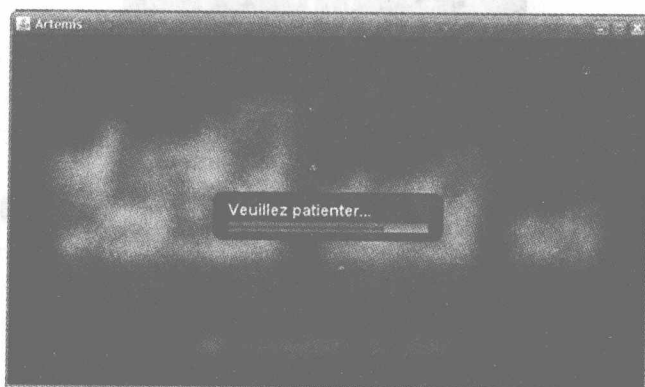


图3 第16章讨论的模糊效果可以把用户的注意力集中在没有模糊的内容上

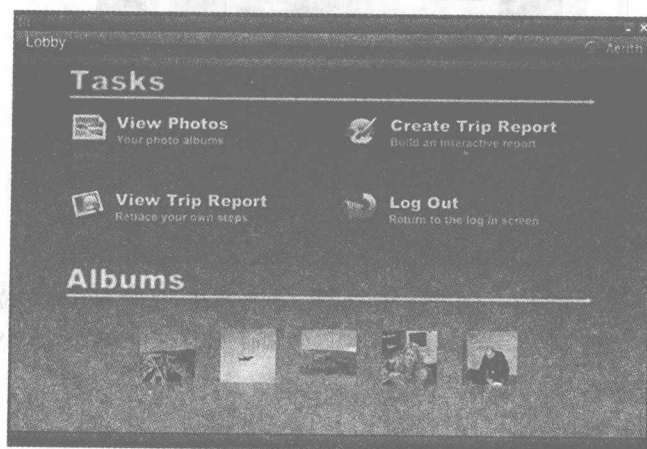


图4 在 <http://aerith.dev.java.net> 上可以获得源码和字节格式的
Aerith 应用程序演示了本书中讨论的许多效果和技术

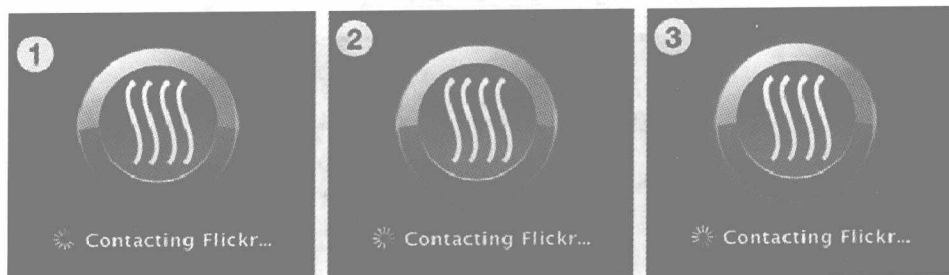


图5 Aerith 的装载屏幕演示了在第 17 章中讨论的脉动效果



图6 如第 17 章所述，自动发光效果应用到 Aerith 的装载屏幕

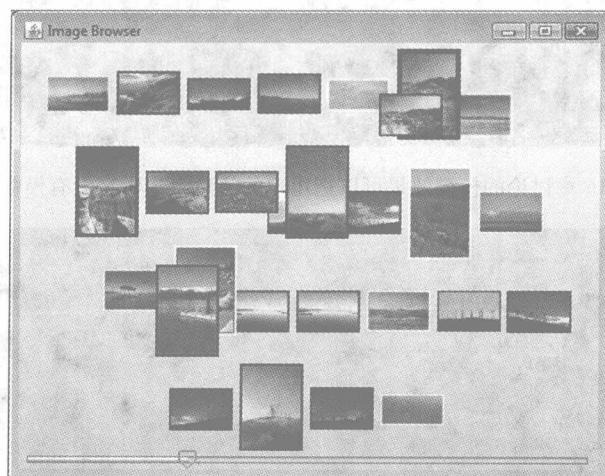


图7 第 18 章讨论了在 GUI 不同状态之间的动画。在这里，略图位置的改变将导致图片自动和平滑地改变它们在窗口中的位置和大小

效果使应用程序更多产

图形效果，尤其是动画效果，可以做得很夸张。用户最终需要的是使应用程序中的所有内容都很闪亮、出其不意、有脉动和不断地淡入淡出。我们在本书中讲述的技术可以做到这一