

HTML5 Canvas领域的标杆之作，公认的权威经典，Amazon五星级超级畅销书，资深技术专家David Geary最新力作

全面讲解Canvas元素的各种功能特性，以及如何利用Canvas进行图形绘制、动画制作、游戏开发、自定义控件制作和移动应用开发，包含大量实例，可操作性极强

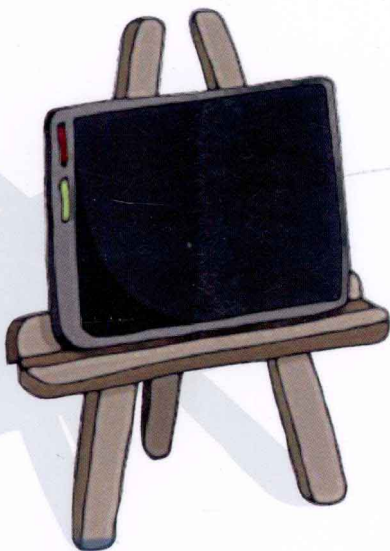
Core HTML5 Canvas
Graphics, Animation, and Game Development

HTML5 Canvas核心技术

图形、动画与游戏开发

(美) David Geary 著

爱飞翔 译



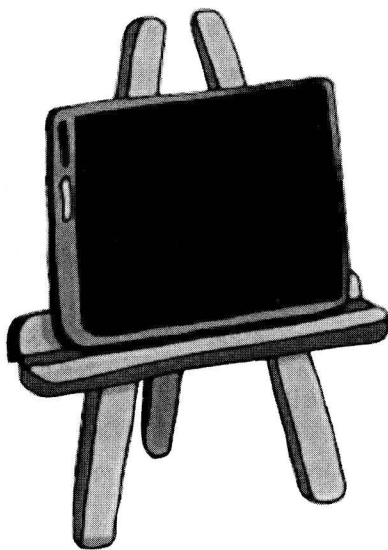
机械工业出版社
China Machine Press

Core HTML5 Canvas
Graphics, Animation, and Game Development

HTML5 Canvas核心技术

图形、动画与游戏开发

(美) David Geary 著
爱飞翔·译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

HTML5 Canvas核心技术: 图形、动画与游戏开发 / (美) 基瑞 (Geary, D.) 著; 爱飞翔译. —北京: 机械工业出版社, 2013.2

(华章程序员书库)

书名原文: Core HTML5 Canvas: Graphics, Animation, and Game Development

ISBN 978-7-111-41634-0

I. H… II. ① 基… ② 爱… III. 超文本标记语言—游戏程序—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2013) 第037106号

版权所有·侵权必究

封底无防伪标均为盗版

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: 01-2013-0208

本书是 HTML5 Canvas 领域的标杆之作, 也是迄今为止该领域内容最为全面和深入的著作之一, 是公认的权威经典、Amazon 五星级超级畅销书、资深技术专家 David Geary 最新力作。它不仅全面讲解了 canvas 元素的 API, 以及如何利用 Canvas 进行图形绘制、动画制作、物理效果模拟、碰撞检测、游戏开发、移动应用开发, 还包含大量实例, 可操作性极强。

全书共分 11 章。第 1 章介绍了 canvas 元素及如何在网络应用程序中使用它; 第 2 章深入研究了如何使用 Canvas 的 API 进行绘制; 第 3 章告诉读者如何绘制并操作 Canvas 中的文本; 第 4 章专门讲解图像、图像的操作及视频处理; 第 5 章介绍如何实现平滑的动画效果; 第 6 章讲解如何用 JavaScript 语言来实现精灵; 第 7 章展示了如何在动画中模拟物理效果; 第 8 章介绍了进行碰撞检测所用的技术; 第 9 章以一个简单但是高效的游戏引擎开始, 提供了游戏制作所需的全部支持功能; 第 10 章讨论了实现自定义控件的通用方法; 第 11 章专门讲述如何实现基于 Canvas 的手机应用程序。

Authorized translation from the English language edition, entitled Core HTML5 Canvas: Graphics, Animation, and Game Development, 9780132761611 by David Geary, published by Pearson Education, Inc., Copyright © 2012 David Geary.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese Simplified language edition published by Pearson Education Asia LTD., and China Machine Press Copyright © 2013.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括中国台湾地区和香港、澳门特别行政区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

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

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: 关 敏

藁城市京瑞印刷有限公司印刷

2013年5月第1版第1次印刷

186mm×240mm·31.75印张 (含0.25印张彩插)

标准书号: ISBN 978-7-111-41634-0

定 价: 99.00元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

读者信箱: hzsj@hzbook.com

译者序

HTML5 是一个富有活力的前沿领域，虽说早在 2008 年年初就发布了第 1 份草案，然而该标准的流行则是 2010 年之后的事情了。

HTML5 技术的兴起有多方面的原因。其中比较重要的一点就是，1999 年制定的 HTML4.01 标准在十几年后已经无法满足急速增长的网络开发需求了。与传统的“服务器－客户端”架构相比，越来越多的开发者开始选择以网页的形式来制作应用软件与游戏。这样做能够降低维护成本，将原来更新客户端所花的精力投入到网页程序的完善之上，以便更加及时地满足新出现的客户需求。如此一来，怎样弥补网页程序在图形绘制、设备底层功能调用、文件访问、影音播放等方面的劣势，就成为制定新标准时必须考虑的问题了。HTML5 标准新增的各类 API 能够很好地应对这些状况。

此外，近年来日益兴起的移动开发也引发了人们对 HTML5 技术的关注。在传统的开发方式中，我们必须移植出版本繁多的客户端，以应对那些操作系统、屏幕大小、硬件配置各不相同的手机及平板电脑。如果我们将这种开发流程以 HTML5 应用的形式统一起来，那么就可以省去在各种设备之间进行移植所带来的问题。

不论是在传统的桌面操作系统之上，还是在新兴的移动设备之中，各大浏览器厂商都在努力适应新的 HTML5 标准，力求提供一套功能丰富而且外观统一的 API。尽管 HTML5 标准仍在不停地更新之中，但是其基本开发思路已经受到众多开发者及用户的肯定。所以说，在 HTML5 标准最终定型之前，提前学习新标准，及早推出开发成本适中而且内容丰富的各类 HTML5 网络应用软件与游戏，不仅可以提高自身的技术能力，还可以把握软件市场的走向，总体来看，是一项明智的抉择。

网络应用开发是一项外延很广的领域，在企业级开发中，我们要学习各种服务器通信技术、软件架构及开发框架，而在另外一些实用软件及游戏的开发中，则又需要投入大量时间进行网页前端的美化。从开发者学习知识的角度看，如果将 HTML、JavaScript、CSS 等制作网络应用程序所需的技术分开研究，则难以把握它们之间的联系，本书的出现正好解决了这个问题。选择 Canvas 为切入点是恰到好处的，因为在各类网络应用软件与游戏中，都要或多或少地用到与绘制相关的功能。如果能够将网页开发常用的 HTML、JavaScript、CSS 等技术有机地结合起来，那么就可以充分地发挥 Canvas 在绘图方面的优势，制作出效果可与原生应用程序相媲美的产品来。

研究 Canvas 的另一个好处是，我们在学习它的同时，还能掌握与之相关的各项实用技术，诸如图形与曲线的编辑、文本的绘制与输入、图像滤镜，等等。在学习这些技术的过程中，读者将会逐步掌握如何通过 JavaScript 代码来动态地运用 Canvas 所提供的各项功能。本书后半部分集中讲解了 Canvas 的几项重要用途，包括动画与精灵、游戏物理学、碰撞检测、游戏开发，等等。借助一个精美的弹珠台游戏，作者将这些重要技术详细而深入地讲解了一番，学习这部分内容将有助于提高图形开发及游戏制作的水平。

全书最后不仅演示了如何实现圆角矩形、滚动条、滑动条及图像查看器等自定义控件，而且还教给大家实现这些自定义控件所遵循的流程，使我们能够用相似的办法实现一大批符合自己需求的新控件来。此外，作者还详细介绍了如何让开发出来的应用程序能够更好地运行于各类移动设备之上。

本书作者清晰而又透彻的行文风格，不仅把图形绘制、动画制作、游戏开发、自定义控件、移动开发等几项重要知识讲解得十分精彩，而且还启迪了我们的开发思路。读完全书之后，大家更需要思考如何利用 JavaScript 程序，将类似 Canvas 这样的新兴技术与传统的 HTML、CSS 结合起来，开发出更加丰富灵活的网络应用程序来。

在本书的翻译过程中，我得到了机械工业出版社华章公司各位编辑及工作人员的帮助，在此表示由衷的感谢。

本书由爱飞翔翻译，王鹏、舒亚林及张军也参与了部分翻译工作。译者非常愿意与诸位朋友通过微博（weibo.com/eastarlee）或电子邮件（eastarstormlee@gmail.com）探讨各类 HTML5 技术问题。由于水平有限，错误和不当之处在所难免，敬请广大读者批评指正。

前言

2001年夏天，笔者阅读了一本有关网络应用程序开发的畅销书。此时，我已经从事了15年图形用户界面（Graphics User Interface，GUI）与“图形密集型应用程序”（graphic-intensive application）的开发。当时我并不知道，那本书的作者 Jason Hunter，会在 No Fluff Just Stuff（NFJS）^①巡回演讲上成为我的好朋友。

看完了 Jason 所著的那本有关 Servlet 的书^②之后，我将它放在膝上，凝望着窗外。做了这么多年的 Smalltalk、C++ 与 Java 语言的开发工作，并富有激情地写完了 1622 页的《Graphic Java 2: Swing》^③一书后，我对自己说：是不是真的该像编写 HTML 网页那样，用类似打印语句那样的代码来实现用户界面了？是的，我的确这样做了。

从那时起，我就在自认为是“软件开发的黑暗时代^④”之中艰难前行。我是 Apache Struts 项目^⑤的第二个代码提交者，并且创造了 Struts 模板库（Struts Template Library），该库最终演进成了流行的 Tiles 项目^⑥。我在“JavaServer Faces（JSF）^⑦专家组（expert group）”中待过 6 年多，还在超过 120 场 NFJS 研讨会及许多其他会议中做过关于服务器端 Java 开发（server-side Java）的演讲，并且与人合著了一本关于 JSF 的书^⑧。曾经有一段时间，Google Web Toolkit^⑨与 Ruby on Rails^⑩技术令我感到兴奋，不过到了这段苦日子的最后，整个工作基本都是枯燥的业务逻辑：在客户端向用户展示数据表单，并在服务器端对其进行处理，再也找不回当初做图形开发及图形用户界面时的那种激情了。

① NFJS是一个有关Java与敏捷技术的软件研讨会，其网站是：<http://www.nofluffjuststuff.com/>。——译者注

② 由Jason Hunter与William Crawford编写的《Java Servlet Programming》，O'Reilly出版社2001年出版。

③ 《Graphic Java 2: Mastering the JFC, Volume 2: Swing, 3rd Edition》，David Geary著，由Prentice Hall出版社于1999年出版。（本书中文版为《Java 2图形设计 卷2: SWING》，由机械工业出版社于2000年出版——译者注）

④ 欧洲黑暗时代（Dark Ages或Dark Age），指的是约476年至1000年的中世纪前期（Early Middle Ages）。详情参阅：<http://zh.wikipedia.org/zh-cn/欧洲黑暗时代>。——译者注

⑤ Struts是Apache软件基金会（ASF）赞助的一个开源项目。它通过采用Java Servlet / JSP技术，实现了基于Java EE Web应用的Model-View-Controller（MVC）设计模式的应用框架（Web Framework），是MVC经典设计模式中的一个经典产品。详情参见：<http://zh.wikipedia.org/zh-cn/Struts>。——译者注

⑥ Apache Tiles，是以Java语言编写的HTML模板框架。其官网是：<http://tiles.apache.org/>。——译者注

⑦ JavaServer Faces（JSF）是一种用于构建Java Web应用程序的标准框架（是Java Community Process规定的JSR-127标准）。它提供了一种以组件为中心的用户界面（UI）构建方法，从而简化了Java服务器端应用程序的开发。由于由Java Community Process（JCP）推动，属于Java EE 5中的技术规范，而受到了厂商的广泛支持。详情参阅：http://zh.wikipedia.org/zh-cn/JavaServer_Faces。该项目专家组的网址为：<http://www.java-serverfaces.org/specification/expert-group>。——译者注

⑧ 《Core JavaServer™ Faces, Third Edition》，由David Geary与Cay Horstmann合著，Prentice Hall出版社于2010年出版。

⑨ Google Web Toolkit（GWT），是一个前端使用JavaScript，后端使用Java的AJAX框架，它通过编译器将Java程序代码编译成JavaScript，可让开发人员使用Java语言，快速构建与维护复杂但高效能的JavaScript前端应用程序，借此减轻开发人员负担。其项目网址是：<https://developers.google.com/web-toolkit>。——译者注

⑩ Ruby on Rails，简称RoR或Rails，是使用Ruby语言编写的开源Web应用框架。它严格按照MVC结构开发，可减少应用开发时的编码及配置工作。详情参阅：http://zh.wikipedia.org/zh-cn/Ruby_on_Rails。——译者注

2010年夏天，在HTML5技术以不可阻挡之势流行起来的时候，我偶然看到了一篇讲述Canvas元素的文章，那时我就知道，救赎的时刻即将来临。我立刻放下了职场中的一切事务，将所有时间都投入到写作之中，力求写出一本最好的Canvas书籍。从那时起，直到2012年3月本书完成，我完全沉浸在Canvas技术和这本书的写作之中。这是迄今为止带给我最多写作乐趣的一本书。

从文字处理到电子游戏，它们所需的全部图形功能，Canvas元素都会提供给你。尽管它在各个平台中的性能有所差异，不过总的来说，Canvas的运行速度还是很快的，尤其是在iOS5平台上，Mobile Safari浏览器可以利用硬件加速来支持Canvas的渲染。浏览器厂商们在遵循HTML5规范方面做得都相当好，所以，编码良好的Canvas应用程序无需修改即可在任何兼容HTML5的浏览器中运行，偶有轻微的不兼容现象。

HTML5就像是“软件开发黑暗时代”之后的“文艺复兴”，可以说，Canvas元素是HTML5中最激动人心的技术。本书将深入讲解Canvas元素及相关的HTML5技术（例如动画时间控制规范），利用这些技术可以开发出能够跨桌面浏览器及移动平台运行的应用程序。

如何阅读本书

本书的写作方式，用禅意一些的话来说，就是可以使你“无需阅读即可领会其内容”（read it without reading）。

在撰写每一章（通常耗时数月）的过程中，我有时一个字都不写，只是一直反复地阅读写作材料。在那段时间中，我致力于准备写作提纲、程序清单、屏幕截图、表格、示意图、条目列表、注解、小窍门及注意事项。这些被我叫做“支架”（scaffolding）^①的东西，才是本书中最为重要的部分。书中的正文，是我在搭好上述支架后才写的，它们是为了给周边的支架材料提供上下文说明，并阐明其中的重要之处。写好后，我又会反复阅读几遍，尽可能地删掉多余的文字。

因为本书专注于提供支架式的学习材料，且惜墨如金，所以这样一本书很容易让读者“无需阅读即可领会其内容”。你可以略读其中的文本，而专注于屏幕截图、程序清单、示意图、表格，以及其他支架材料，无论你想了解什么话题，这么做都能让你学到很多知识。你可以随意将书中的正文视为二等公民，而且，如果乐意的话，等需要查阅正文的时候再来读也不迟。

全书概览

本书由两部分组成。第一部分包括前4章，几乎占了本书篇幅的一半。这部分讲解了Canvas元素的API，告诉读者如何在Canvas元素上面绘制图形与文本，如何绘制并操作图像。本书剩下的7章将告诉读者，如何实现动画效果及动画精灵，如何创建物理模拟效果，如何进行碰撞检测，以及如何开发电子游戏。在本书最后两章之中，有一章会讲如何实现一些自定义控件，例如进度条、滑动条（slider）、图像查看器（image panner），另外一章将会告诉你如何创建基于Canvas的手机应用程序。

① 作者在这里使用了“支架式教学”（又名“鹰架理论”，Scaffolding Instruction或Instructional Scaffolding）的一些术语。该教学法在教授学生一项新的概念或技能时，通过提供足够的支援来提升学生的学习能力，当学生具备了自主的学习策略，提升了认知、情感、知识与学习技能时，这些支援会逐渐取走。详情参阅：<http://www.hudong.com/wiki/支架式教学>。——译者注

第1章“基础知识”，介绍了 canvas 元素，并且演示了如何在网络应用程序中使用它。本章有一个小节，介绍了进行 HTML5 开发所需的一般性入门知识，包括浏览器、控制台、调试器、性能分析器、时间轴等。接下来的内容为读者讲解如何实现 canvas 元素的一些基本功能：如何在 canvas 之上绘制、canvas 参数以及绘图表面自身的保存与恢复、打印 canvas，还有一段关于离屏 canvas (offscreen canvas) 的简介。本章最后简单地介绍了一些基础的数学知识，包括基本的代数运算、三角函数、向量数学，以及根据计量单位推导等式。

第2章“绘制”，这是全书篇幅最长的一章，深入研究了如何使用 Canvas 的 API 进行绘制，告诉读者如何在 canvas 上绘制线条、弧线、曲线、圆、矩形以及任意多边形，如何以纯色、渐变色及图案对其进行填充。本章不仅讲述了绘图机制，还向读者展示了如何使用 Canvas API 来制作一些有实际用途的范例程序，例如：通过绘制临时的辅助矩形来创建橡皮带线条 (rubber band)；拖动 canvas 之中的图形；实现一个简单的保留模式图形子系统，用以跟踪 canvas 之中的多边形，以便让用户对其进行编辑；使用剪辑区域，在不影响 Canvas 后方背景的情况下擦除图形。

第3章“文本”，告诉读者如何绘制并操作 canvas 之中的文本。你将会学到如何对文本进行描述与填充，如何设置字型属性，以及如何调整文本在 canvas 之中的位置。本章也讲了如何在 Canvas 中实现自制的文本控件。本章最后讲述了如何制作闪烁的文本编辑光标以及可编辑的文本段。

第4章“图像与视频”，专门讲解图像、图像的操作以及视频处理。你将会看到如何在 canvas 中绘制与缩放图像，你还将学到如何通过获取每个像素的颜色分量来编辑图像。在这一章中还可以学到更多有关剪辑区域的使用方法，以及如何将图像做成动画的知识。接下来的内容强调了安全与性能的重要性，最后以视频处理来结束本章内容。

第5章“动画”，告诉读者如何运用名为 requestAnimationFrame() 的方法来实现平滑的动画效果。该方法定义在 W3C 一个题为“基于脚本动画的定时控制”(Timing control for script-based animations) 的规范中。读者还将看到计算动画帧速率的方法，以及如何进行其他活动的排期，例如以另外一种帧速率来刷新动画的用户界面。本章将会告诉读者在播放动画时用来恢复背景图像的三种策略，并且比较了每种方式所存在的性能隐忧。本章还演示了如何实现基于时间的运动效果，动画的背景滚动，使用视差来制作仿 3D 效果，以及在播放动画的过程中检测并响应用户手势。本章收尾部分讲了定时动画，并实现了一个简单的动画定时器，其后讨论了动画制作的最佳实践。

第6章“精灵”，向读者讲解了如何以 JavaScript 语言来实现精灵（即带有动画效果的显示对象）。精灵对象具有可视化的表现形式，这通常指的是一幅图像。你可以在 canvas 内移动它们，并且循环地播放某个图像集中的图片，来以此产生动画效果。精灵是用以制作游戏的基础构建单元。

第7章“物理效果”，向读者展示了如何在动画中模拟物理效果，对其进行建模，包括下坠的物体、抛射体的弹道，以及摇晃的钟摆。本章还向读者展示了如何将时间与运动这两个要素封装入动画之中，以模拟真实世界的移动效果，诸如刚起跑的短跑运动员（加速运动）与正在刹闸的汽车（减速运动）。

在许多游戏中，碰撞检测是另一个重要的方面。所以，本书第8章“碰撞检测”专门讲解进行精灵之间碰撞检测所用的技术。本章开头讲述了使用外接矩形或外接圆形所进行的简单碰撞检测。这种办法实现起来很简单，然而并不是十分可靠。鉴于在很多情况下，简单的碰撞检测结果都不可靠，所以，本章的大部分内容都用于讲解分割轴定理 (Separating Axis Theorem)。该定理

是在任意 2D 或 3D 多边形之间，进行碰撞检测所用的最佳手段之一。然而，它并不是说给那些有数学恐惧症的人（mathematically faint of heart）听的，所以，这一章要花大量的篇幅，并以外行人听得懂的词语来表述这个定理。

第 9 章“游戏开发”，以一个简单但是高效的游戏引擎开始。它提供了游戏制作所需的全部支持功能，包括绘制精灵、维护高分榜、实现基于时间的运动效果，以及播放多声道的音乐。接下来，本章详述了两个游戏。第一个是简单的“Hello World”类型的游戏，它演示了游戏引擎的用法，并且为后续的游戏制作提供了一个便利的出发点。它还向读者演示了如何实现大部分游戏所普遍具有的功能，例如资源管理、平视游戏状态显示^①，以及用于显示高分榜的用户界面。第二个是优秀的弹珠台（pinball）游戏，它利用了本书前面讲到的许多素材，并且演示了如何在实际游戏中进行复杂的碰撞检测。

很多基于 Canvas 的应用程序都需要一些自定义控件，所以，第 10 章“自定义控件”将会教你如何来实现它们。本章讨论了实现自定义控件的通用方法，然后通过 4 个自定义控件来演示这些技术，这 4 个控件是：圆角矩形、进度条、滑动条，以及图像查看器。

本书的最后一章“移动平台开发”，专门讲述如何实现基于 Canvas 的手机应用程序。你将会学到如何控制视窗（viewport）的大小，使得应用程序能够在手机上正常地显示出来。同时还会看到如何利用 CSS3 媒体特征查询（media query）技术来使应用程序适应不同的屏幕大小及显示方向。

你还将了解到如何让基于 Canvas 的应用程序以全屏模式运行，并为其提供桌面图标与启动画面，让它看起来与 iOS5 平台的原生应用程序几乎一模一样。本章的末尾将实现一个虚拟键盘，它使得用户在 iOS5 应用程序中，不需要将输入焦点放在文本域（text field）控件之中即可输入文本。

预备知识

为了有效阅读本书，你必须能够比较熟练地使用 JavaScript、HTML 及 CSS 技术。举例来说，笔者假设你已经知道如何使用 JavaScript 语言的原型继承（prototypal inheritance）来实现一个对象，并且已经熟知一般网络应用程序的开发。

本书也用到了你也许很久以前学过，但是早就忘掉的数学知识，诸如基本的代数运算、三角函数、向量数学，以及根据计量单位推导等式。在第 1 章的末尾，你将会看到一个涵盖上述话题的基础知识简介。

本书源代码

本书所有代码的版权均归笔者所有，对这些代码的使用必须遵守随源码发行的许可协议。此许可协议是在 MIT 许可协议的基础上修改而成的，它允许你使用本书代码来做任何事情，包括用其制作可以贩售的软件产品。然而，你不可以使用本书代码来创建教材，包括图书、培训视频、

① Heads-up Display，简称 HUD。是目前普遍运用在航空器上的飞行辅助仪器。平视的意思是指飞行员不需要低头就能够看到所需的重要资讯。电子游戏领域借用该词来表示叠加在游戏主画面之上的游戏状态信息，例如主角的生命值、道具、分数、当前关卡等。详情参阅：<http://zh.wikipedia.org/zh-cn/平视显示器>与[http://en.wikipedia.org/wiki/HUD_\(video_gaming\)](http://en.wikipedia.org/wiki/HUD_(video_gaming))。——译者注

简报 (presentation), 等等。详细情况请参考源码附带的许可协议内容。

在实现这些范例代码时, 笔者明智地选择将程序清单中所含的注释量降到最小。为此, 笔者尽量将代码本身写得易读一些, 每个方法中平均包含大约 5 行代码, 非常易于理解。

笔者还严格遵循了 Douglas Crockford 在其经典著作《JavaScript, The Good Parts》^①中所提出的建议。比方说: 所有函数作用域内部的变量都声明在函数头部; 每个变量的声明都单独占一行代码; 总是使用 `===` 操作符及其同类操作来进行相等性测试 (equality testing)。

Canvas 规范与本书将来的发展情况

HTML5 的 API 在持续地进化, 每一次进化, 大都伴随着新功能的加入。Canvas 标准当然也不例外, 实际上, 就在这本书将要送去印刷的前几天, WHATWG^② 的 Canvas 标准就进行了一次更新。此次更新又向其中加入了如下新功能:

- 用于创建椭圆形路径的 `ellipse()` 方法。
- 两个名为 `getLineDash()`、`setLineDash()` 的新方法, 以及用于绘制虚线的 `lineDashOffset` 属性。
- 进行了功能扩充之后的 `TextMetrics` 对象, 它可以让开发者精确地指定文本的外接矩形框。
- 名为 `Path` 的路径对象。
- 名为 `CanvasDrawingStyles` 的绘制风格对象。
- 对点击区域 (hit region) 的广泛支持。

在那时, 尚未有任何浏览器支持这些新加入的功能, 所以当时不可能编写代码来测试它们。

在 2012 年 3 月 26 日的这次更新之前, 你已经可以使用 Canvas 来绘制弧线及圆形了, 但是规范之中尚未有条款规定 Canvas 必须支持椭圆的绘制。这次更新之后, 除了可以绘制弧线与圆形之外, 你也可以使用新加入的 `ellipse()` 方法在 2D 环境中的 Canvas 上面来绘制椭圆了。与此类似, 该绘制环境现在也明确支持虚线的绘制了。

原来的 `TextMetrics` 对象只能返回一种度量结果, 那就是字符串的宽度。然而, 经过了 2012 年 3 月 26 日的规范更新之后, 你可以设定 canvas 中某个字符串所占据矩形区域的宽度与高度了。对 `TextMetrics` 对象的这次功能扩充, 使得我们可以更加简单、高效地实现基于 Canvas 的图形控件。

除了椭圆绘制功能的加入与 `TextMetrics` 对象的改进之外, 更新之后的规范还加入了一些归纳在 `Path` 与 `CanvasDrawingStyles` 名下的新方法。在规范更新之前, 并没有明确规定用于存储路径与绘制风格的机制。现在, 不仅有了专门用于表示这些抽象机制的对象, 而且很多 2D 环境下的 Canvas 方法也多出来了一个可以接受 `Path` 对象参数的版本。例如, 你可以调用 `context.stroke()` 方法在当前绘制环境中进行路径描边, 这时所描绘的路径是当前绘制环境中所设定的路径。然而, 现在绘制环境对象也可以使用 `stroke(Path)` 方法来描绘路径了, 该方法所描绘的路径是经由参数所传入的那个路径对象, 而非当前环境中的路径。当你在 `Path` 对象上调用 `addText()` 这样的方法来修改路径时, 你可以指定一个 `CanvasDrawingStyle` 对象, 这样的话, 路径对象在添加文本时也会使用这个对象中的属性。

① 《JavaScript, The Good Parts》, Douglas Crockford 著, O'Reilly 出版社 2008 年出版。

② Web Hypertext Application Technology Working Group, 简称 WHATWG, 是一个以推动网络 HTML 5 标准为目的而成立的组织。在 2004 年, 由 Opera、Mozilla 基金会和苹果这些浏览器厂商组成。其官方网站是: <http://www.whatwg.org/>。——译者注

更新之后的规范包含了对点击区域的广泛支持。点击区域是由路径所定义的。你可以在其上关联一个可选的鼠标光标，以及一些无障碍访问（accessibility）参数，诸如一个“无障碍丰富互联网程序”[Ⓐ]，一个搭配有点击区域的标签，等等。一个 canvas 可以有多个点击区域。除了这些好处之外，使用点击区域还可以实现碰撞检测并提升无障碍访问性，使它们变得更为容易且更加高效。

最后，WHATWG 与 W3C[Ⓑ] 规范都包含了两个用于无障碍访问的 Canvas 环境方法，这样的话，应用程序就可以在当前路径周围绘制聚焦环（focus ring），以便让用户可以通过键盘在 Canvas 之中进行焦点切换。这项功能并不是 2012 年 3 月 26 日更新规范时添加的，实际上，它被加入规范当中已经有一段时间了。然而，在本书成稿时，尚未有浏览器厂商支持这项功能，所以本书中不会讲到这部分内容。

因为 Canvas 规范在不断地演进，同时浏览器厂商也在不断地实现新的功能，所以本书也会经常地更新。在此期间，可以访问 <http://corehtml5canvas.com/> 来查阅这些 Canvas 新功能，并预览本书在下一个版本中对这些新功能的覆盖程度。

配套网站

本书的配套网站是：<http://corehtml5canvas.com/>。在这里可以下载书中的源代码，运行精选的本书范例程序，并查找其他与 HTML5 及 Canvas 技术有关的资源。

致谢

书写是一项团队运动，我很荣幸能够与诸位优秀的同伴一起完成本书的写作。

首先，我要感谢长期共事的编辑 Greg Doench 先生。他也是我的好朋友。从笔者提出本书写作计划的那一刻起，他就由衷地相信此书一定会是本佳作。同时，他也给了我相当的自由度，让我可以按照自己的想法来写作本书。从概念酝酿到付诸刊印，再到印刷成书，Greg 一直在监督着本书的制作。他做了这么多，我已经很满足啦！

还让笔者感到幸运的是，在制作本书的过程中，Greg 还带来了他自己的优秀团队。Julie Nahil 非常出色地完成了与生产管理有关的工作，她让一切事情都走上了正轨。Alina Kirsanova 将 DocBook 格式的原始 XML 文件转换成读者手中印刷精美的图书。Alina 女士也出色地完成了本书的校对工作，删除了书中的一些小错误及矛盾之处。

笔者又一次高兴地看到 Mary Lou Nohr 女士对本书进行了审稿工作。在 15 年的写书生涯中，Mary Lou 女士一直是审稿编辑，她不仅将每一本书都修改得比我的原文更好，而且还教会了笔者

Ⓐ Accessible Rich Internet Applications，全名为 Web Accessibility Initiative - Accessible Rich Internet Applications，简称 WAI-ARIA，是一份由 W3C 所提出的有关网络可及性的倡议，它针对无障碍网页服务提出了一些规范建议。详情参阅：<http://en.wikipedia.org/wiki/WAI-ARIA>。该规范的官方网址是：<http://www.w3.org/TR/wai-aria/>。——译者注

Ⓑ 万维网联盟（World Wide Web Consortium），又称 W3C 理事会。1994 年 10 月在麻省理工学院计算机科学实验室成立。建立者是互联网的发明者蒂姆·伯纳斯-李。W3C 制定了一系列标准并督促 Web 应用开发者和内容提供者遵循这些标准。标准内容包括使用语言的规范，开发中使用的导则和解释引擎的行为等。W3C 也制定了包括 XML 和 CSS 等在内的众多影响深远的标准规范。官方网址是：<http://www.w3.org/>。——译者注

写作的技巧。

对于任何一本技术类书籍而言，要想获得成功，技术评审都是至关重要的。所以我会主动邀请一些我认为拥有合适的水准、能为本书做出重大贡献的人来当技术评审。我运气真好，找到了一些优秀的技术评审人员，他们帮助我对本书的内容打磨抛光，使之更加精致。首先，要感谢 Philip Taylor 先生，他是我所见过的最为博学且一丝不苟的技术评审之一。Philip 实现了将近 800 个与 Canvas 有关的测试用例，大家可以看看这里：<http://philip.html5.org/tests/canvas/suite/tests>。在评审本书的每一章时，他都发给我数页极富见解的评论，这些评论只有在对 Canvas 元素了解得细致入微的前提下才能写得出来。他对本书的贡献远远超出了技术评审的范围，仅他一个人的努力，就得以让本书的质量大为提升。

接下来，我要感谢 thirstyhead.com 公司的 Scott Davis 先生，他是位一流的 HTML5 及手机网络应用程序开发专家。Scott 做了很多场有关 HTML5 与移动开发技术的会议演讲，与他人联合创立了 HTML5 Denver User Group，并曾向 Yahoo! 公司的开发者教授过手机开发技术。与 Philip 一样，Scott 对书中许多不同的主题都提出了优秀的建议，这也远远超出了技术评审的要求。由于他严格的评审意见，笔者将本书近四分之一的内容全部改写了一遍，这也让本书的发行时间推迟了整整三个月。重写之后，本书的品质又提升了一个档次，为此，我要由衷地感谢 Scott。

知名游戏 Runfield (<http://fhtr.org/runfield/runfield>) 的作者之一 Ilmari Heikkinen 先生，对本书中与动画、精灵、物理学和检测碰撞有关的章节，提出了一些独到的建议。Ted Neward、Dion Almaer、Ben Galbraith、Pratik Patel、Doris Chen、Nate Schutta 与 Brian Sam-Bodden 都给出了一些很好的评审意见。

我还要感谢的是 jsperf.com 网站的创立者 Mathias Bynens 先生，他允许笔者使用从那个网站上截取的图片。

在“物理效果”一章中所用到的那份精灵表 (sprite sheet)，是由 MJKRZAK 绘制的，笔者感谢这位网友。该精灵表是我从 People's Sprites 网站[⊖]的公共领域区下载的。笔者还要感谢 Ilmari Heikkinen 先生，他允许我在“动画”这一章中使用他所绘制的天空图案作为讲解视差范例时的素材。“精灵”这一章的某些图像，是取材于知名的开源游戏 Replica Island[⊗]。

最后，感谢 Hiroko、Gaspé 与 Tonka 在这段时间中对我的理解，我将这一年半的生活全部都用在了本书的写作之上。

作者简介

David Geary 先生是一位杰出的作家、演讲者与顾问，他从 20 世纪 80 年代就开始用 C 语言及 Smalltalk 语言来实现基于图形的应用程序与用户接口了。David 在波音公司从事了 8 年 C++ 语言与面向对象软件开发的培训工作，在 1994 年至 1997 年间，曾在 Sun Microsystems 公司担任软件工程师。他写了 8 本有关 Java 的书籍，其中有两本讲 Java 组件框架、Swing 与 JavaServer Faces (JSF) 技术的书非常畅销。他所著的《Graphic Java 2: Swing》一直都是畅销的 Swing 教程，与 Cay Horstmann 合写的《Core JavaServer™ Faces》，是 JSF 领域的畅销书。

David 是一位富有激情的演讲者，他在全世界数百场会议中都发表过演讲。他参加了 6 年的

⊖ 该网站网址是：<http://panelmonkey.org/>。——译者注

⊗ 该游戏的网站是：<http://replicaisland.net/>。——译者注

No Fluff Just Stuff 巡回演讲，在超过 120 场讨论会中做了发言，并且三度荣获 “JavaOne Rock Star” 头衔[⊖]。

2011 年，David 与 Scott Davis 联合创立了 HTML5 Denver Meetup group（网址是 <http://www.meetup.com/HTML5-Denver-Users-Group>），截至 2012 年本书出版时，该讨论组已经有超过 500 名成员了。

读者可以通过 Twitter（用户名是 [davidgeary](#)）及本书的配套网站（网址是 <http://corehtml5-canvas.com/>）联系到 David。

⊖ JavaOne是一个主要由Java开发人员所参与的周年研讨会。JavaOne Rock Star是向那些发表了精彩演讲并且展示了演说能力的开发者授予的荣誉称号。详情参阅：<http://www.oracle.com/javaone/quick-links/rock-star/index.html>。——译者注

目 录

译者序

前言

第 1 章 基础知识	1
1.1 canvas 元素	1
1.1.1 canvas 元素的大小与绘图表面的大小	4
1.1.2 canvas 元素的 API	5
1.2 Canvas 的绘图环境	6
1.2.1 2d 绘图环境	6
1.2.2 Canvas 状态的保存与恢复	8
1.3 本书程序清单的规范格式	9
1.4 开始学习 HTML5	10
1.4.1 规范	10
1.4.2 浏览器	11
1.4.3 控制台与调试器	11
1.4.4 性能	13
1.5 基本的绘制操作	15
1.6 事件处理	18
1.6.1 鼠标事件	18
1.6.2 键盘事件	22
1.6.3 触摸事件	23
1.7 绘制表面的保存与恢复	23
1.8 在 Canvas 中使用 HTML 元素	25
1.9 打印 Canvas 的内容	32
1.10 离屏 canvas	35
1.11 基础数学知识简介	37
1.11.1 求解代数方程	37
1.11.2 三角函数	38
1.11.3 向量运算	39
1.11.4 根据计量单位来推导等式	42
1.12 总结	44

第 2 章 绘制	45
2.1 坐标系统	46
2.2 Canvas 的绘制模型	47
2.3 矩形的绘制	48
2.4 颜色与透明度	50
2.5 渐变色与图案	52
2.5.1 渐变色	52
2.5.2 图案	54
2.6 阴影	57
2.7 路径、描边与填充	60
2.7.1 路径与子路径	63
2.7.2 剪纸效果	64
2.8 线段	69
2.8.1 线段与像素边界	70
2.8.2 网格的绘制	71
2.8.3 坐标轴的绘制	72
2.8.4 橡皮筋式的线条绘制	74
2.8.5 虚线的绘制	79
2.8.6 通过扩展 CanvasRenderingContext2D 来绘制虚线	80
2.8.7 线段端点与连接点的绘制	81
2.9 圆弧与圆形的绘制	83
2.9.1 arc() 方法的用法	83
2.9.2 以橡皮筋式辅助线来协助用户画圆	85
2.9.3 arcTo() 方法的用法	86
2.9.4 刻度仪表盘的绘制	88
2.10 贝塞尔曲线	93
2.10.1 二次方贝塞尔曲线	93
2.10.2 三次方贝塞尔曲线	95
2.11 多边形的绘制	97
2.12 高级路径操作	102
2.12.1 拖动多边形对象	102
2.12.2 编辑贝塞尔曲线	107
2.12.3 自动滚动网页, 使某段路径所对应的元素显示在视窗中	115
2.13 坐标变换	116
2.13.1 坐标系的平移、缩放与旋转	116
2.13.2 自定义的坐标变换	119
2.14 图像合成	123
2.15 剪辑区域	128
2.15.1 通过剪辑区域来擦除图像	128

2.15.2 利用剪辑区域来制作伸缩式动画	133
2.16 总结	135
第 3 章 文本	137
3.1 文本的描边与填充	137
3.2 设置字型属性	141
3.3 文本的定位	144
3.3.1 水平与垂直定位	144
3.3.2 将文本居中	146
3.3.3 文本的度量	147
3.3.4 绘制坐标轴旁边的文本标签	148
3.3.5 绘制数值仪表盘周围的文本标签	151
3.3.6 在圆弧周围绘制文本	152
3.4 实现文本编辑控件	154
3.4.1 指示文本输入位置的光标	154
3.4.2 在 Canvas 中编辑文本	159
3.4.3 文本段的编辑	163
3.5 总结	174
第 4 章 图像与视频	175
4.1 图像的绘制	176
4.1.1 在 Canvas 之中绘制图像	176
4.1.2 drawImage() 方法的用法	177
4.2 图像的缩放	179
4.3 将一个 Canvas 绘制到另一个 Canvas 之中	183
4.4 离屏 canvas	186
4.5 操作图像的像素	189
4.5.1 获取图像数据	189
4.5.2 修改图像数据	195
4.6 结合剪辑区域来绘制图像	208
4.7 以图像制作动画	211
4.8 图像绘制的安全问题	216
4.9 性能	216
4.9.1 对比 drawImage(HTMLImage)、drawImage(HTMLCanvas) 与 putImageData() 的绘图效率	217
4.9.2 在 Canvas 中绘制另一个 Canvas 与绘制普通图像之间的对比; 在绘制时缩放图像与保持原样之间的对比	217
4.9.3 遍历图像数据	218
4.10 放大镜	222

4.10.1 使用离屏 canvas	224
4.10.2 接受用户从文件系统中拖放进来的图像	225
4.11 视频处理	227
4.11.1 视频格式	227
4.11.2 在 Canvas 中播放视频	229
4.11.3 视频处理	230
4.12 总结	234
第 5 章 动画	235
5.1 动画循环	235
5.1.1 通过 requestAnimationFrame() 方法让浏览器来自行决定帧速率	237
5.1.2 Internet Explorer 浏览器对 requestAnimationFrame() 功能的实现	241
5.1.3 可移植于各浏览器平台的动画循环逻辑	241
5.2 帧速率的计算	248
5.3 以不同的帧速率来执行各种任务	249
5.4 恢复动画背景	250
5.4.1 利用剪辑区域来处理动画背景	250
5.4.2 利用图块复制技术来处理动画背景	252
5.5 利用双缓冲技术绘制动画	253
5.6 基于时间的运动	254
5.7 背景的滚动	257
5.8 视差动画	261
5.9 用户手势	264
5.10 定时动画	266
5.10.1 秒表	266
5.10.2 动画计时器	269
5.11 动画制作的最佳指导原则	270
5.12 总结	271
第 6 章 精灵	272
6.1 精灵概述	273
6.2 精灵绘制器	275
6.2.1 描边与填充绘制器	275
6.2.2 图像绘制器	279
6.2.3 精灵表绘制器	281
6.3 精灵对象的行为	284
6.3.1 将多个行为组合起来	285
6.3.2 限时触发的行为	287
6.4 精灵动画制作器	289