

TURING

图灵程序设计丛书

[美] Daniel Shiffman 著
周晗彬 译

代码 本色

用编程模拟自然系统

The Nature of Code

Simulating Natural Systems with Processing



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

[美] Daniel Shiffman 著
周晗彬 译

代码 本色

用编程模拟自然系统

The Nature of Code

Simulating Natural Systems with Processing

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

代码本色 : 用编程模拟自然系统 / (美) 希夫曼
(Shiffman, D.) 著 ; 周晗彬译. — 北京 : 人民邮电
出版社, 2015. 1

(图灵程序设计丛书)

ISBN 978-7-115-36947-5

I. ①代… II. ①希… ②周… III. ①程序语言—
程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第200505号

内 容 提 要

本书由纽约大学 Nature of Code 课程老师 Daniel Shiffman 写就, 是一本借助开源语言 Processing 全面介绍如何用代码模拟自然世界的学习指南。作者从模拟无生命物体、活物、智能系统三个层面, 从手工编写 Processing 代码到使用现有的物理函数库模拟高级而复杂的行为, 利用有趣的事例渐进式介绍了算法和模拟方面的高级编程策略和技术。主要内容涉及向量、力、粒子系统、三角函数、自治智能体、细胞自动机、分形、遗传算法和人工神经网络。

本书适合游戏设计师、好学的程序员、物理学爱好者及所有对计算机模拟和互动编程感兴趣的人学习参考。

◆ 著 [美] Daniel Shiffman

译 周晗彬

责任编辑 李松峰 毛倩倩

执行编辑 张 庆

责任印制 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京鑫正大印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 26.5

字数: 653千字

2015年1月第1版

印数: 1—3 000册

2015年1月北京第1次印刷

著作权合同登记号 图字: 01-2013-8804号

定价: 99.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

献 辞

献给我的祖母——Bella Manel Greenfield (1915 年 10 月 13 日—2010 年 4 月 3 日)。



Bella Manel 出生在纽约市，是一位数学领域的女性先驱，她于 1939 年在纽约大学获得博士学位，师从理查德·柯朗^①。她曾与理查德·贝尔曼^②一起在拉莫·伍尔德里奇公司（现在的 TRW）和兰德公司工作。之后，她在贝尔蒙特的那慕尔圣母大学和加州大学洛杉矶分校任教。1995 年，纽约大学的柯朗研究所设立了 Bella Manel 奖，用于表彰女性和少数民族人士的杰出研究工作。

① 1888 年 1 月 8 日—1972 年 1 月 27 日，德国数学家，曾在纽约大学担任数学教授。他创建了数学研究组织并且做得非常成功，柯朗数学研究中心后来成为了一个应用数学研究中心的标杆。——编者注

② 1920 年 8 月 26 日—1984 年 3 月 19 日，美国数学家、动态规划的创始人，曾在兰德公司工作多年（并在此期间提出了动态规划）。——编者注

推 荐 序

我们在这个行星上已经生活了数十年，虽然不一定对这个世界的自然规律有深入的理解，但肯定已经习以为常。例如在足球比赛中，我们能欣赏到美妙的曲线任意球。如果足球在空中突然直角转向，我们一定会觉得很“不自然”。又例如，我们有时候能看到上千只鸟集体飞翔，它们并不是乱飞一通，而是按照某种规律组成不断变形的群体。如果它们互相碰撞而掉下来，我们很可能会怀疑它们是否生病，做出这些“不自然”的行为。

充满好奇心的人们，可以通过学习物理学、化学、生物学等学科去了解各种自然现象。但对于一些程序员、艺术家，他们除了希望对这些原理有所了解，还希望能在作品中模仿这些自然现象。

模仿、模拟等词汇意味着我们并不是要完整地复制自然世界，而是通过抽象、近似化等方式，获取当中我们认为重要的特性。例如，我们知道水是由水分子所组成，但肉眼看不到这么小的水分子，更常见的是水滴、容器中的水、海洋等。要模仿淋浴花洒的水流动态，我们可考虑以水滴为单位，逐一模拟它们以某初始速度射出，然后受地心引力影响而产生抛物线的移动路径。但要模仿海洋时，我们可能更关注它海面的波浪，而不是海面下巨量容积的海水。在此情况下，我们可能会模拟海面上一些分布点的垂直运动，做出波浪起伏的效果。

或许读者（及正在考虑阅读本书的人）会问，为什么要用软件模拟这些自然现象呢？抛开职业、学业上的需要，我认为最简单的答案是，用程序编写这些现象本身就是很有趣的事情。编程不单能处理网页请求、计算账目、储存数据，原来还可以创造出富含自然现象的虚拟世界！

若以职业来考虑，游戏、动画、电影特效、视觉艺术等行业都会需要这方面的知识。例如在游戏方面，由于许多游戏都含有一个虚拟世界，这些自然现象的模拟技术可以应用于程序式建模（如地形、植物）、程序式动画（如粒子特效、云层变化）、游戏逻辑（如刚体物理）、人工智能（如非玩家角色的移动）等。在动画方面，虽然不需要能互动的虚拟世界，但为了视觉上的真实性也需要使用计算机实现各种自然现象，例如为了制作《冰雪奇缘》，迪士尼与加州大学洛杉矶分校就研究出一种模拟雪运动的新技术。

虽然本书书名含“代码”二字，却并不是只有程序员才能阅读。在国内游戏行业里有一句俗语：“不会美术的程序员不是好策划。”我们不必为自己的知识技能设限。刚刚在 2014 游戏开发者大会（中国）上，前同事 Ken Wong 就道出自己如何从一位概念美术师（参与作品《爱丽丝：

疯狂回归》)，退隐一年学习游戏编程及思考游戏设计，然后建立团队创作出获得苹果年度设计大奖的《纪念碑谷》。

这本书作为这个领域的入门书籍，使用了简易的 Processing 编程语言作为例子，非专业程序员也会很容易理解。但如果读者对编程真的完全没有概念，可以先读一些 Processing 入门书籍。由于本书涉猎甚广，若读本书后感到意犹未尽，除了可再读本书的参考文献，*Texturing and Modeling, Third Edition: A Procedural Approach* 也是一个不错的选择。

叶劲峰

游戏程序员

2014 年 10 月

致 谢

“我们周围的世界以一种复杂而精彩的方式运作着。在生命的早期，我们通过感知和互动了解所处的环境。我们期望周围物理世界的行为方式和我们自身的感知记忆一致。比如，石头在重力的作用下坠落，风能把更轻的物体吹得更远。本课程的主要目的就是理解和模拟物理世界的运动元素，并在我们的数字模拟世界中加入这些元素。我们的目标是根据用户对物理世界的感知记忆创建直观、丰富且令人满意的体验。”

——James Tu，动态物体课程描述，2003年春，ITP

关于本书的故事

2003年，作为纽约大学Tisch学院ITP课程（交互通信计划）的毕业生，我学习了一门名为“动态物体”的课程，课程讲师是ITP的兼职教授、交互设计师James Tu。这门课的内容是一系列软件实验，目的是产生实时的“非真实”图像。课程期间，我们需要根据各种规则获取有生命物体的图像并填充颜色，而且让它们在屏幕上移动。课程涵盖了向量、力、振荡、粒子系统、递归、转向和弹簧这些知识，它们和我的工作密切相关。

我经常在项目中用到上述概念，却从来没有仔细研究过这些算法背后的科学原理，也没有用面向对象的方法规范地实现它们。就在那个学期，我还选修了“衍生艺术系统基础”（Foundations of Generative Art Systems），这门课的讲师是Philip Galanter，他的研究方向主要集中在衍生艺术的理论和实践上，涵盖混沌、细胞自动机、遗传算法、神经网络和分形等话题。Tu和Galanter的课程使我在模拟算法和技术上有了很大提高，为我之后的教学工作带来了很大帮助。同时，这两门课程也是本书的基础。

但本故事背后还隐藏着另一个谜题。

Galanter的课程几乎全是理论，而Tu的课程使用的是Macromedia Director和Lingo编程语言。在那个学期，我试着将很多算法用C++语言实现（那时候还没有openFrameworks和Cinder等创意编程环境，因此我在用笨拙的方式编写C++代码）。在学期末，我发现了Processing语言（<http://www.processing.org>）。Processing在那时候还只是alpha版本（版本0055）。由于我有一定的Java经验，因此一直在思考：能否用这门对艺术家友好的开源语言开发一套编程和模拟的教学示

例？在ITP和Processing社区的帮助下，我完成了这件事。这8年来，我一直在用Processing进行算法和应用教学。

首先，我要感谢ITP的创立者Red Burns，在这10年里，他一直支持并鼓励着我；感谢ITP的主席Dan O’Sullivan，他一直是我的教学导师，也是第一个建议我开始Processing课程教学的人，促使我整理这些教学资料；感谢*Pro Android Media*的作者、杰出的开发者Shawn Van Every，他为我提供了很多帮助，也在ITP给我很多灵感。在编写本书的过程中，ITP的教员Clay Shirky、Danny Rozin、Katherine Dillon、Marianne Petit、Marina Zurkow和Tom Igoe给了我很多建议和反馈。其他ITP同僚也提供了很多帮助，他们是：Brian Kim、Edward Gordon、George Agudow、John Duane、Marlon Evans、Matt Berger、Megan Demares、Midori Yasuda和Rob Ryan。

ITP课程教学中，还有举不胜举的学生提供了很多反馈。本书的很多材料都源自我的同名课程，这些课程我已经教了5年。在本书的编写及修订过程中，我拥有了一摞写满注释的打印草稿，也积累了许多来自学生的电子邮件，感谢他们的点评、纠正以及鼓励。

我还要感谢Processing社区中的程序员和艺术家。如果没有Processing的开发者Casey Reas和Ben Fry，我不会编写这本书。通过阅读Processing的源代码，我积累了许多知识。Processing语言的简洁用法，以及它的网站和IDE为编程提供了极大的便利，同时也为学生带来很多乐趣。除此之外，我还从许多Processing程序员那里得到数不胜数的建议与灵感，他们是：Andrés Colubri、Jer Thorp、Marius Watz、Karsten Schmidt、Robert Hodgins、Seb-Lee Delisle以及Ira Greenberg。Heather Dewey-Hagborg为本书的第10章（神经网络）提供了大量优秀反馈；Scott Murray在电子邮件中提供了一些关于内联SVG的实用建议；Golan Levin提供了很多文章来源（可用于阅读扩展）。

我还要感谢本书的编辑Shannon Fry，她为我的写作过程提供了细心周到的反馈，让我能把每一章完成得更出色。

特别值得一提的是Zannah Marsh，她不知疲倦地为本书提供了很多插图。尤其要感谢她的耐心工作，因为在本书的编写过程中，我经常改变插图需求。我也想感谢David Wilson，他为本书设计了布局和封面。特别感谢Steve Klise，他设计并开发了本书的网站，帮我制定了PDF版“随意支付”的付费模式。

我在前言中会提到，本书是由开源出版系统Magic Book生成的。Magic Book是由ITP的开发者、设计师和艺术家组成的团队经过一年多时间开发完成的。本书的各种格式（HTML、PDF等）都是由它生成的，而Magic Book的输入文件仅仅是一个简单的ASCII文档和CSS布局文件。Magic Book项目是由Rune Madsen发起的，他开发了初始的Ruby/Sinatra框架。如果没有Rune的贡献，直至2013年我一定还在为本书的最终成形苦苦挣扎。Steve Klise为Magic Book修复了很多错误，并使我能在代码块中加入注释内容。Miguel Bermudez、Evan Emolo和Luisa Pereira Hors也在其他方面做出许多贡献，他们研究了ASCIIDOC和CSS分页媒体的来龙去脉。ITP的研究人员Greg Borenstein在本书的Web出版和打印方面提供了大量建议和支持。Magic Book使用Prince引擎（princexml.com）从HTML文档中产生PDF文档，因此我要感谢PrinceXML的CEO Michael Day，

他（以风驰电掣般的速度）回答了我的很多问题。

最后，我要感谢我的家人：我的妻子Alikí Caloyeras，她一直在支持我的写作，同时也在写自己的大部头图书；还有我的孩子Elias和Olympia，为了腾出更多时间陪伴他们，我有更多动力尽快完成本书。我还要感谢我的父亲Bernard Shiffman，他慷慨地教会了我很多数学知识，也提供了很多关于本书的反馈。除此之外，还要感谢我的母亲Doris Yaffe Shiffman和我的兄弟Jonathan Shiffman，他们经常会问“你的书进展如何”。

Kickstarter

还有一个组织使本书的出版变为可能：Kickstarter。

2008年，我完成了我的第一本书*Learning Processing*（Morgan Kaufmann/Elsevier出版）。我花费整整3年时间编写*Learning Processing*，却没有仔细思考选择哪一家出版社，只是想：“你们真的要出版我写的书？好，成交！”遗憾的是，这次出版经历并不算太好。整个过程中，出版社为我安排了5位编辑，我也没有收到太多的内容反馈。出版社以外包的形式完成排版工作，这导致本书有很多错误和不一致的地方。除此之外，我发现这本书的价格也不符合自己的期望。我希望这是一本价格平易近人的平装本*Processing*介绍性图书，但最后这本书的定价是50美元，接近“教科书”的价格。

我想特别指出，出版社的本意是好的。他们非常希望出版优秀的书籍，这对读者、出版社和作者来说都是一件好事。出版社也在努力地出好书，但是他们的预算很紧张，因此能投入的精力也比较少。除此之外，我觉得他们对*Processing*这种开源“创意”编程环境并不太熟悉，他们擅长编辑计算机科学领域的教科书。

因此，对于本书，我觉得非常有必要尝试自出版。既然我无法从出版社的编辑那里获得太多编辑支持，为什么不直接雇用一位编辑？既然出版社的定价不符合我的意愿，为什么不自己制定价格（对于PDF版本，还可以让读者定价）？剩下的还有市场问题——出版社有没有为你带来附加价值并增加更多读者？从某些角度上看，确实有。比如O'Reilly的“Make”系列，O'Reilly会专门为书籍等产品建立社区。对于*Processing*的编程学习，只需要一个简单的URL就能增加更多读者，那就是processing.org。

遗憾的是，很快我就发现有一种东西出版社能提供，而我却没法通过自出版途径得到。那就是截止时间。若独自做这件事，我可以挣扎两年，宣称自己在编写这本书，事实上却只懒懒散散地写出寥寥数页内容。在我的待办事项里，写书这件事永远被排在最后面。于是，我就找到了Kickstarter，Kickstarter中有许多对本书内容感兴趣的读者（他们还会为此付钱），于是我就有了截止日期的压力。现在你能读到这本书应该归功于这个网站。

最重要的是，自出版允许我以很灵活的方式发布内容和制定价格。在Elsevier的网站上，你可以用53.95美元购买*Learning Processing*电子书，在这53.95美元中，我能获得5%的分成，也就是

2.70美元。如果用自出版的方式出版这本书，我可以制定更便宜的售价，比如10美元的价格，这能为读者节省80%的开支，同时也让自己得到3倍的收益。我甚至还可以让读者自己对PDF版定价。

由于我拥有本书的全部内容，因此还可以用其他数字方式出版这本书。本书的内容和全部源代码皆采用知识共享署名-非商业性使用的许可证，你可以在Github上获取全部内容。除此之外，你也可以在Github上提交问题及要求，进行纠误及评论。最后，由于本书使用了灵活的按需印刷服务，因此我可以时常发布新版本，让内容保持最新。（对本书的一次购买就包含终身的免费升级。）

因此，我要感谢Kickstarter公司（尤其是Fred Benenson，他说服我决定冒险，并指导我选择合适的许可证）和本书的全体支持者，在这里，我还要特别感谢下面这些热心的支持者：

- Alexandre B.
- Robert Hodgin
- JooYoun Paek
- Angela McNamee (Boyhan)
- Bob Ippolito

所有支持者都对本书的出版做出了直接贡献。他们对初稿和定稿的捐款让我有更多动力完成写作，而且让我得以支付设计和编辑费用（包括在周六上午写作之时雇人照看小孩）。

除了捐助资金，Kickstarter的用户还审读了本书的预发布章节，期间提供了无数反馈，指出了本书的错误和混乱部分。我要特别感谢Frederik Vanhoutte和Hans de Wolf，他们在牛顿物理学上有深厚的功底，为本书的第2章和第3章提出了很多建议。

前言

这是一本什么书

我在 ITP (<http://itp.nyu.edu>) 教授一门名为“计算媒体导论”的课。在这门课中，学生主要学习一些编程基础知识（变量、条件语句、循环、对象和数组等）。除此之外，他们还学习如何使用基本元素（图像、像素、计算机视觉、组网、数据和 3D 等）开发交互式应用。课程内容以我之前写的入门书 *Learning Processing* 为主，而本书是 *Learning Processing* 的续篇。一旦你掌握了编程基础并且接触了形形色色的应用场景，接下来很可能就是深入研究某个特定的方向。举个例子，你可以专注于计算机视觉（比如阅读 Greg Borenstein 写的 *Making Things See* 等书）。当然，本书的内容只是众多发展方向之一，它只是延续了 *Learning Processing*，展示了 Processing 语言在算法和模拟方面的更高级编程技术。

本书的目标非常简单：我们想看看真实世界中发生的各种自然现象，以及如何通过编程对它们进行模拟。

那这到底是一本什么样的书？这是不是一本有关科学的书？我可以很肯定地回答：不是。事实上，我们确实会涉及物理学和生物学的个别话题，但不会从严谨的学术层面进行研究，因为这不在本书讲述范围之内。相反，我们会简单探讨某些科学原理，只攫取我们需要的那一部分内容，并根据它们构建相关的示例程序。

那这是不是一本有关艺术或设计的书呢？我还会回答：不是。尽管我们的工作结果都是视觉上可见的事物（用 Processing 开发的演示动画），但也仅仅是用简单的图形和色彩做出的演示，我们真正专注的是它们背后的算法和相关编程技术。然而，我还是希望艺术工作者和设计师们能将本书中的知识融入工作实践，创造一些真正新颖有趣的作品。

如果非要给这本书归类，我觉得它只是一本普普通通的编程书。尽管书中的一些章节取材自科学原理（比如牛顿物理学、细胞生长、进化等），而且一些编程结果会激发艺术创作的灵感，但归根结底本书重心是代码的实现，尤其是其中的面向对象编程技术。

关于Processing语言

本书使用 Processing 语言，原因有很多。第一，它是我用着最舒服的编程语言和开发环境，

我很喜欢用它来工作；第二，它是免费开源的，并且非常适合初学者，它的开发者社区很活跃。对很多人来说，Processing 或许是他们学习的第一门编程语言。因此，我希望这本书能拥有广泛的受众，并希望通过 Processing 用一种友好的方式阐述其中的原理。

本书中所写的例子并不严格限定于 Processing 语言，我们还可以用 ActionScript、JavaScript、Java（脱离 Processing 开发环境），或是其他开源的“创意编程”开发环境，比如 openFrameworks、Cinder，以及最近发布的 pocode。我希望自己完成这本书之后，能将本书中的例子移植到其他开发环境中，并发布其他语言的示例程序。如果你对移植本书的示例程序感兴趣，请随时联系我（daniel@shiffman.net）。

本书中的所有示例都已在 Processing 2.0b6 版本上测试通过，大部分例子也兼容早期版本。我会时常更新这些示例，使它们兼容最新版本。你可以从 GitHub 获取最新代码（<http://github.com/shiffman/The-Nature-of-Code-Examples>）。

阅读需知

读懂本书的前提条件是：你上过一学期的 Processing 编程课（并且熟悉面向对象编程）。这并不是说如果你学的是其他语言和开发环境就读不懂本书，关键是你必须学过编程。

如果你之前没有写过代码，阅读本书会有一定难度，因为本书假定你有一定的编程基础。对此，我建议你去读读 Processing 的介绍性书籍。你可以在 Processing 官方网站找到很多相关图书（<http://processing.org/learning/books/>）。

如果你是有一定经验的开发者，但之前从未接触过 Processing，那么可以直接下载 Processing 开发工具（<http://processing.org/download/>），然后从它的快速入门页面（<http://processing.org/learning/gettingstarted/>）开始，跑一下示例程序。

我还想强调一下，面向对象编程的经验才是关键。在本书的引言中，我们还会回顾 Processing 的基础知识，不过我还是建议你先读读关于对象的 Processing 教程：<http://processing.org/learning/objects>。

你用什么阅读本书

你是在 Kindle 上阅读本书，还是看纸质版？你是在笔记本电脑上看 PDF 版，还是用平板看 HTML5 动画版？你是不是舒舒服服地躺在椅子上，通过上面的各种媒介学习本书的内容？

你现在阅读的这本书是用 Magic Book（<http://www.magicbookproject.com>）生成的。Magic Book 是 ITP（<http://itp.nyu.edu>）开发的用于自出版的开源框架，它允许使用易读易写的纯文本格式编写文档。准备好文档内容后，你只需要点击一个神奇的按钮，Magic Book 就会为你生成各种图书格式——PDF、HTML5、印刷版、Kindle 电子书等^①。所有的外观样式都是通过 CSS 控制的。

^① 中译本电子版可在 iTuring.cn 点击购买。——编者注

本书的第 1 版只有 PDF 版、纸质版以及 HTML5 版（包括用 Processing.js 开发的动画演示）。希望明年开授课程时，这本书可以有其他格式。如果你想助我一臂之力，请随时联系我（daniel@shiffman.net）。

本书的“故事”

如果你浏览了本书目录，会发现本书的 10 章内容分别讲述不同话题。从某种意义上说，本书只是 10 个不相关概念和例子的集合。然而，我一直在找一个能娓娓道来的故事，以便将这些材料串联在一起。在你阅读各章内容之前，我想给你讲讲这个故事。

第一部分：无生命的物体

想象以下场景：草地上有一个足球，球员一脚将它踢到空中。足球在重力的作用下迅速下降，而空气阻力又让它能在空中飘移一段时间，直到落在高高跃起的运动员的头上。在这个过程中，足球是无生命的物体，它对自己的运动没有自主权，只能等待外界环境在它身上施加外力。

我们该如何用 Processing 对足球的运动建模？如果你曾经写过在窗口中移动一个圆圈的程序，可能编写过下面这行代码：

```
x = x + 1;
```

你在 x 位置画一个图形，在每一帧动画中，将 x 的值递增，并重画这个图形，最后就产生了图形在运动的假象。你可以进一步完善这个程序，给图形位置加一个 y 坐标，以及在 x 轴和 y 轴上的速度：

```
x = x + xspeed;  
y = y + yspeed;
```

故事的第一部分会进一步研究这个问题：我们将继续研究 $xspeed$ 和 $yspeed$ 这两个变量，了解它们如何形成一个向量（第 1 章），而向量正是物体运动的基石。尽管我们不会在向量这个概念上搞出什么新东西，但它是本书其余部分的基础。

了解向量以后，我们很快会意识到：一切的外力（第 2 章）都是向量。踢足球就相当于在足球上施加外力。外力会让物体做什么样的运动？根据牛顿运动定理，外力等于质量乘以加速度（ $F = ma$ ）。外力能让物体加速，而对外力进行建模可以让我们根据各种运动定理模拟物体的运动状态。

被运动员施加作用力的足球还可能会发生旋转。物体的运动受加速度控制，旋转受角加速度（第 3 章）控制。了解角度和三角函数的基本知识就能模拟物体的旋转运动，并掌握钟摆运动原理，比如钟摆的摆动和弹簧的弹跳运动。

一旦解决了单个无生命物体的基本运动和力学问题，我们将把这些原理运用到成千上万的物体上，并用一个系统管理它们，这个系统称作粒子系统（第 4 章）。在粒子系统中，我们将学习面向对象编程的某些高级特性，比如继承和多态。

从第 1 章到第 4 章,所有例子都是从零编写的,也就是说,我们直接用 Processing 编写模拟物体运动的算法代码。我们肯定不是第一批尝试用 Processing 模拟物体运动的开发者,所以下一步将学习如何使用现有的物理函数库(第 5 章)对更高级和更复杂的行为进行建模。我们会接触 Box2D (<http://www.box2d.org>) 和 toxiclibs' Verlet Physics package (<http://toxiclibs.org/>) 这两个函数库。

第二部分:活物

如何对有生命的事物进行建模?这不是一个简单的问题,但我们可以从对外界环境有感知能力的对象开始建模。试想,在外力作用下从桌子上落下物体的运动以及海豚在水里的游动,这两种运动都是由外力引起的,但它们有本质区别:物体不能决定自己何时从桌面上落下,而海豚可以决定自己何时跳出水面。海豚可以有自己的意愿,可以感到饥饿或恐惧,这些情绪会影响它的运动。通过自治智能体模拟技术(第 6 章),我们将生命注入之前无生命的物体,让它们能根据对外界环境的理解决定如何运动。

有了自治智能体的概念,再结合在第 4 章学习的建模系统,我们将深入研究群体行为模型,这种模型能表现出复杂系统的特性。我们是这么描述复杂系统的:“一个复杂系统的整体不等同于局部的简单组合。”复杂系统的局部可能是很简单且容易理解的个体,但它们组成的整体会表现得非常复杂、智能且难以预测。对复杂系统的建模会让我们超越简单的运动建模,进入基于规则的系统领域。在这里,我们会提出诸多疑问。我们能用细胞自动机(在某个网格区域内繁殖的细胞系统,第 7 章)建立什么样的模型?通过分形(描述大自然的几何学,第 8 章),我们又能建立怎样的模型?

第三部分:智能

在第一部分,我们让物体产生运动。在第二部分,我们让物体有自己的意愿,并把自身意愿和生存规则结合在一起。在本书的最后部分,我们会使这些物体变得更智能。在这里,我们会提出疑问。为了让模型进化,能否将生物的进化过程应用到计算系统中(第 9 章)?受人类大脑的启发,我们能否开发一个神经网络(第 10 章),使其能够从自身错误中自我学习以适应环境?

本书的教学大纲

尽管要在一个学期中讲述本书内容过于紧张,我还是将其设计成了一个 14 周的课程。值得一提的是,我发现有些章适合跨周讲解。我的教学大纲一般是这样的:

- 第 1 周 引言和向量(第 1 章)
- 第 2 周 力(第 2 章)
- 第 3 周 振荡(第 3 章)
- 第 4 周 粒子系统(第 4 章)
- 第 5 周 物理函数库 I(第 5 章)

- 第 6 周 物理函数库 II 和操纵（第 5 章和第 6 章）
- 第 7 周 期中项目：演示运动建模项目
- 第 8 周 复杂系统：群集和一维细胞自动机（第 6 章和第 7 章）
- 第 9 周 复杂系统：二维细胞自动机和分形（第 7 章和第 8 章）
- 第 10 周 遗传算法（第 9 章）
- 第 11 周 神经网络（第 10 章）
- 第 12 周~第 13 周 期末项目研讨
- 第 14 周 期末项目演示

如果你打算在自己的课程中采用上述安排，请随时联系我。我希望最终能够提供一套视频和幻灯片材料作为辅助教程。

生态系统模拟项目

关于编程学习，我倒是非常希望你舒舒服服地躺在椅子上，读几篇编程教程即可一切。但我不得不承认，只读教程是完全不够的，你还需要写一些代码，针对每一章的内容做一两个实践项目对编程学习会有很大帮助。在 ITP 讲授这门课程时，我发现学生们也非常乐意针对每一章的知识点一步步地开发完整的实践项目。

每章最后都有一系列具有针对性的练习题，这些练习题组成了一个完整的项目。这个项目的场景是，你要为某科技馆开发一套展览软件——电子生态系统，其中要用程序模拟大自然的生物，并将它们投影到整个屏幕中供游客观看。当然，我们的创意不该局限于这一个项目上，而只是用这个示例项目描述本书的内容，把所有知识点都串联起来。我鼓励你开动脑筋，自己想一个更有创意的想法。

如何获取在线代码和提交反馈

若想获取本书相关内容，请访问本书官方网站（<http://www.natureofcode.com>）。你还可以在 GitHub（<http://github.com/shiffman/The-Nature-of-Code>）上找到本书的源代码和插图。如果你有任何意见或想纠正书中的错误，请在 GitHub 的 issues 上提交这些问题。

本书所有示例项目和练习题的源代码^①都可以在 GitHub（<http://github.com/shiffman/The-Nature-of-Code-Examples>）上找到，每章内容多多少少都会包含一些代码片段，为了更好地阐述其中关键点，我对这些代码片段做了删减和简化。如果你想看到完整的代码和注释，请上 GitHub 获取。

如果碰到任何代码上的疑问，我建议你求助 Processing 论坛（<http://forum.processiong.com>）。

① 亦可免费注册 iTuring.com 至本书页面下载。——编者注

目 录

第 0 章 引言 1

- 0.1 随机游走 1
- 0.2 随机游走类 2
- 0.3 概率和非均匀分布 6
- 0.4 随机数的正态分布 9
- 0.5 自定义分布的随机数 12
- 0.6 Perlin 噪声（一种更平滑的算法） 14
 - 0.6.1 映射噪声 16
 - 0.6.2 二维噪声 18
- 0.7 前进 21

第 1 章 向量 23

- 1.1 向量 24
- 1.2 Processing 中的向量 26
- 1.3 向量的加法 28
- 1.4 更多的向量运算 31
 - 1.4.1 向量的减法 32
 - 1.4.2 向量加减法的运算律 33
 - 1.4.3 向量的乘法 34
 - 1.4.4 更多的向量运算律 36
- 1.5 向量的长度 36
- 1.6 单位化向量 38
- 1.7 向量的运动：速度 39
- 1.8 向量的运动：加速度 43
- 1.9 静态函数和非静态函数 47
- 1.10 加速度的交互 49

第 2 章 力 54

- 2.1 力和牛顿运动定律 54
 - 2.1.1 牛顿第一运动定律 54
 - 2.1.2 牛顿第三运动定律 55

- 2.1.3 牛顿第三运动定律（从 Processing 的角度表述） 56

- 2.2 力和 Processing 的结合：将牛顿第二运动定律作为一个函数 57
- 2.3 力的累加 58
- 2.4 处理质量 59
- 2.5 制造外力 61
- 2.6 地球引力和力的建模 65
- 2.7 摩擦力 67
- 2.8 空气和流体阻力 70
- 2.9 引力 75
- 2.10 万有引（斥）力 83

第 3 章 振荡 86

- 3.1 角度 86
- 3.2 角运动 88
- 3.3 三角函数 92
- 3.4 指向运动的方向 93
- 3.5 极坐标系和笛卡儿坐标系 96
- 3.6 振荡振幅和周期 98
- 3.7 带有角速度的振荡 101
- 3.8 波 104
- 3.9 三角函数和力：钟摆 107
- 3.10 弹力 114

第 4 章 粒子系统 122

- 4.1 为什么需要粒子系统 122
- 4.2 单个粒子 123
- 4.3 使用 ArrayList 127
- 4.4 粒子系统类 132
- 4.5 由系统组成的系统 134

4.6 继承和多态的简介	136	5.9.4 第4步: 使用夹具将形状连接 到物体上	178
4.7 继承基础	138	5.10 复杂的形状	180
4.8 用继承实现粒子类	142	5.11 Box2D 关节	186
4.9 多态基础	144	5.11.1 步骤1: 确保有两个物体	189
4.10 用多态实现粒子系统	145	5.11.2 步骤2: 定义关节	189
4.11 受力作用的粒子系统	147	5.11.3 步骤3: 配置关节的属性	190
4.12 带排斥对象的粒子系统	151	5.11.4 步骤4: 创建关节	190
4.13 图像纹理和加法混合	156	5.12 回到力的话题	195
第5章 物理函数库	160	5.13 碰撞事件	196
5.1 Box2D 及其适用性	161	5.13.1 步骤1: Contact 对象, 你 能否告诉我哪两个物体发 生了碰撞	198
5.2 获取 Processing 中的 Box2D	162	5.13.2 步骤2: 夹具对象, 你能否告 诉我你连接在哪个物体上	198
5.3 Box2D 基础	163	5.13.3 步骤3: 物体, 你能否告诉我 你连接在哪个粒子对象上	198
5.3.1 SETUP	164	5.14 小插曲: 积分法	200
5.3.2 DRAW	164	5.15 toxiclibs 的 Verlet Physics 物理库	202
5.3.3 Box2D 世界的核心元素	164	5.15.1 获取 toxiclibs	203
5.4 生活在 Box2D 的世界	165	5.15.2 VerletPhysics 的核心元素	203
5.5 创建一个 Box2D 物体	167	5.15.3 toxiclibs 中的向量	203
5.5.1 第1步: 定义一个物体	167	5.15.4 构建 toxiclibs 的物理世界	204
5.5.2 第2步: 设置物体的定义	168	5.16 toxiclibs 中的粒子和弹簧	205
5.5.3 第3步: 创建物体	168	5.17 整合代码: 一个简单的交互式弹 簧	207
5.5.4 第4步: 为物体的初始状态 设置其他属性	169	5.18 相连的系统 I: 绳子	209
5.6 三要素: 物体、形状和夹具	169	5.19 相连的系统 II: 力导向图	211
5.6.1 第1步: 定义形状	169	5.20 吸引和排斥行为	214
5.6.2 第2步: 创建夹具	169	第6章 自治智能体	218
5.6.3 第3步: 用夹具将形状连接 到物体上	170	6.1 内部的力	218
5.7 Box2D 和 Processing 的结合	171	6.2 车辆和转向	219
5.7.1 第1步: 在主程序 (即 setup() 和 draw() 函数) 中添加 Box2D	173	6.3 转向力	220
5.7.2 第2步: 建立 Processing 盒子 对象和 Box2D 物体对象之间 的联系	173	6.4 到达行为	226
5.8 固定的 Box2D 对象	176	6.5 你的意图: 所需速度	229
5.9 弯曲的边界	177	6.6 流场	231
5.9.1 第1步: 定义一个物体	177	6.7 点乘	236
5.9.2 第2步: 定义形状	177	6.8 路径跟随	239
5.9.3 第3步: 配置形状	177	6.9 多段路径跟随	246