异步图书
www.epubit.com.cn

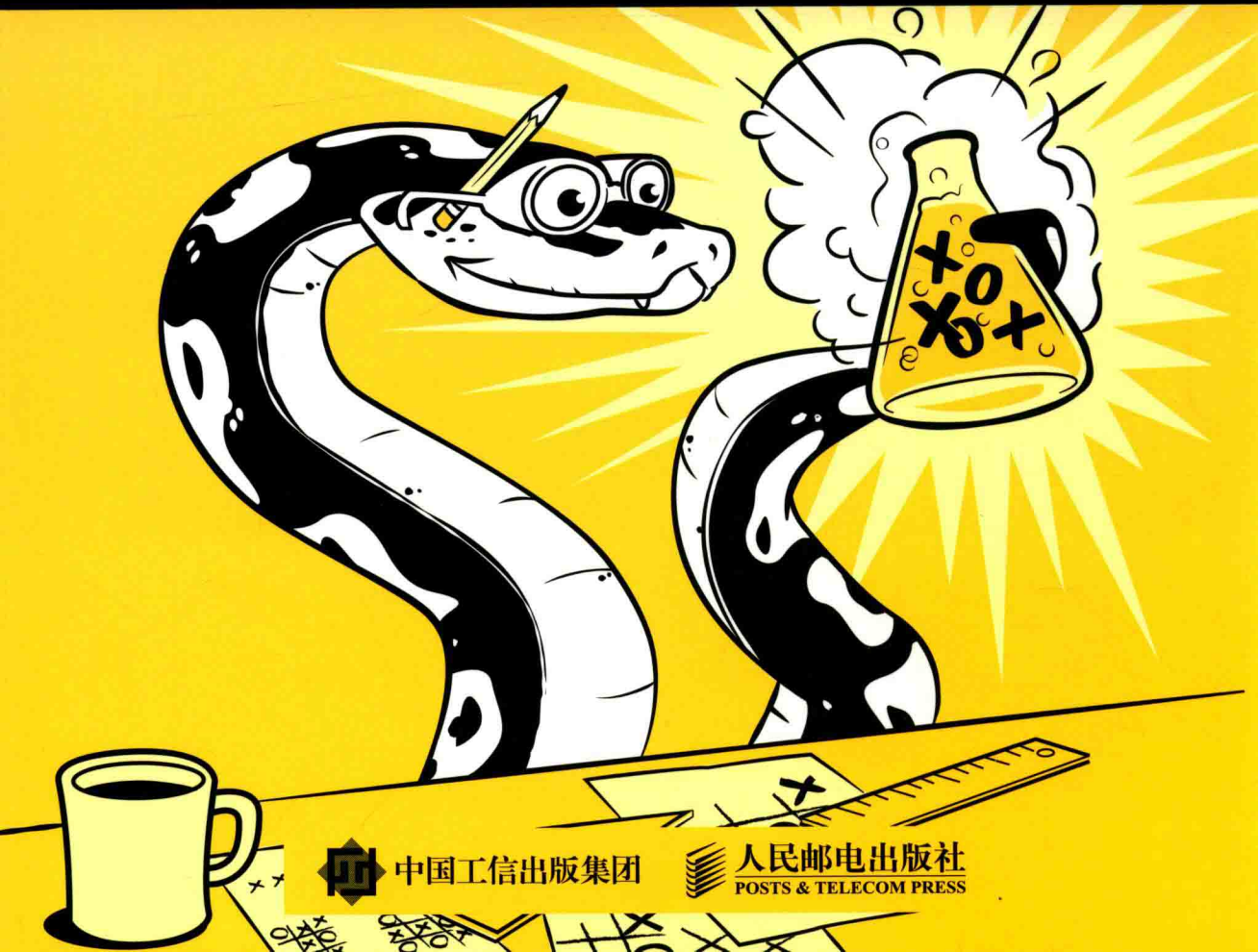
Python 编程畅销书作者新版力作
初学者学习游戏开发实战指南

no starch press

Python（第4版）游戏编程快速上手

Invent Your Own Computer Games with Python, 4th Edition

【美】Al Sweigart 著　　李强 译



中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

异步图书
www.epubit.com.cn

no starch press

Python（第4版）游戏编程快速上手

Invent Your Own Computer Games with Python, 4th Edition

【美】Al Sweigart 著　　李强 译

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

Python游戏编程快速上手 : 第4版 / （美） 斯维加特（Al Sweigart）著 ; 李强译. -- 北京 : 人民邮电出版社, 2017.10
ISBN 978-7-115-46641-9

Ⅰ. ①P… Ⅱ. ①斯… ②李… Ⅲ. ①软件工具－游戏程序－程序设计 Ⅳ. ①TP311.561

中国版本图书馆CIP数据核字(2017)第199295号

版权声明

◆ 著 [美] Al Sweigart
译 李 强
责任编辑 陈冀康
责任印制 焦志炜

◆ 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
北京市艺辉印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：21
字数：468 千字 2017 年 10 月第 1 版
印数：1－2 400 册 2017 年 10 月北京第 1 次印刷

著作权合同登记号 图字：01-2017-5413 号

定价：69.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

内容提要

Python 是一种高级程序设计语言，因其简洁、易读及可扩展性日渐成为程序设计领域备受推崇的语言。

本书通过编写一个个小巧、有趣的游戏来教授 Python 编程，并且采用直接展示游戏的源代码并通过实例来解释编程的原理的方式。全书共 21 章，12 个游戏程序和示例贯穿其中，介绍了 Python 基础知识、数据类型、函数、流程控制、程序调试、流程图设计、字符串操作、列表和字典、笛卡尔坐标系、密码学基础、游戏 AI 模拟、动画图形、碰撞检测、声音和图像等方方面面的程序设计知识。本书可以帮助读者在轻松有趣的过程中，掌握 Python 游戏编程的基本技能。

本书适合不同年龄和层次的 Python 编程初学者阅读。

前　言

当我在孩童时代第一次玩视频游戏的时候，我就被深深吸引住了。但是，我并不只是想玩视频游戏，我想要开发游戏。我找到像这本书一样的一本书，它教我如何编写第一个程序和游戏。它很有趣也很容易。我所开发的第一款游戏就像是本书中的这些游戏一样。它们就像我父母给我买的任天堂游戏一样有趣，但它们是我自己所开发的游戏。

现在，作为一名成年人，我仍然能够享受到编程的乐趣，并且能从中获到回报。但是，即便当你长大成人后并没有成为一名程序员，编程也还是一种有用而且有趣的技能。它训练你的大脑去思考逻辑、做出规划，并且当你从代码中发现错误的时候，会重新考虑你的思路。

目前的编程书籍大多分为两种类型。第一种，与其说是教编程的书，倒不如说是在教“游戏制作软件”，或教授使用一种呆板的语言，使得编程“简单”到不再是编程。而第二种，它们就像是教数学课一样教编程：所有的原理和概念都以小的应用程序呈现给读者。本书采用了不同的方式，教你通过开发视频游戏来学会编程。我直接展示了游戏的源代码，并且通过实例来解释编程的原理。在我学习编程的时候，这种方法起到了关键作用。对于其他人的程序是如何工作的，我学习的越多，对自己的程序的思路也越多。你所需要的只是计算机、一种叫做 Python 解释器的免费软件以及这本书。一旦你学会了如何开发本书中的游戏，你就能够自己开发游戏了。

计算机是不可思议的机器，并且学习编写计算机程序并不像人们想象的那样难。计算机程序就是计算机所能够理解的一堆指令，这就像一本故事书就是读者可以读懂的一堆句子一样。

要对计算机发号施令，就使用计算机能够理解的语言来编写一个程序。本书介绍的是一种叫做 Python 的编程语言。有很多种不同的编程语言，如 BASIC、Java、

JavaScript、PHP 和 C++等。

当我还是一个孩子的时候，BASIC 是作为第一门编程语言来学习的。然而，此后出现了像 Python 这样的新的编程语言。Python 学起来甚至比 BASIC 还要简单！但是 Python 也是供专业程序员使用的一种正规语言。此外，安装和使用 Python 完全是免费的，你只需要连接到因特网并下载它就可以了。

视频游戏无外乎计算机程序，它们也是由指令构成的。在本书中，我们将要创建的游戏看上去比 Xbox、PlayStation 或者 Nintendo 的游戏简单。这些游戏没有绚丽的图案，因为我们要用它们来教授基本的编程知识。我们有意选择这些简单的游戏，以便你可以专注于学习编程。游戏并非复杂才有趣。

本书的目标读者

编程并不难。但是，却很难找到教你通过编程来做有趣事情的学习材料。有些计算机书籍会介绍很多大部分新手程序员都不需要了解的话题。本书将介绍如何编写自己的计算机游戏。你将学习到有用的技巧和可以展示的有趣游戏。本书的目标读者是：

- 想要自学计算机编程的完全初学者，他们甚至之前没有任何编程经验；
- 想要通过创建游戏来学习编程的青少年；
- 想要教其他人编程的成年人和教师；
- 任何想要通过学习专业编程语言来学习如何编程的人，无论是年轻人还是老年人。

本书主要内容

在本书的大多数章节中，我们都会介绍并讲解一个单独的新的游戏项目。有几章会介绍额外的有用的主题，例如调试。当游戏用到新的编程概念的时候，会讲解这些概念，并且这些章是有意让读者按照顺序来阅读的。以下是每章内容的一个简短说明。

- 第 1 章介绍了如何通过每次一行代码来体验一下如何使用 Python 的交互式 shell。
- 第 2 章介绍了如何在 Python 的文件编辑器中编写完整的程序。
- 在第 3 章中，我们将编写本书中的第 1 个程序——猜数字程序，它会要求玩家猜测一个神秘数字，然后针对玩家的猜测是太高了还是太低了给出提示。

- 在第 4 章中，我们将编写一个简单的程序来给用户讲几个笑话。
- 在第 5 章中，我们将编写一个猜测游戏，其中，玩家必须从两个山洞中做出选择，一个山洞中是友善的龙，另一个山洞中是饥饿的龙。
- 第 6 章介绍了如何使用调试器来修正代码中的问题。
- 第 7 章说明了如何使用流程图来规划像 Hangman 游戏这样较长的程序。
- 在第 8 章中，我们将根据第 7 章的流程图，来编写 Hangman 游戏。
- 第 9 章使用了 Python 的字典数据类型，给 Hangman 游戏扩展了新的功能。
- 在第 10 章中，我们将学习如何使用人工智能来编写人机对抗的 Tic-Tac-Toe 游戏。
- 在第 11 章中，我们将学习如何开发一款叫做 Bagels 的推理游戏，其中，玩家必须根据线索来猜测神秘数字。
- 第 12 章介绍了笛卡尔坐标系，我们将在后面的游戏中用到它。
- 在第 13 章中，我们将学习如何编写一款寻宝游戏，其中，玩家要在海洋中搜索丢失的藏宝箱。
- 在第 14 章中，我们将开发一个简单的加密程序，它允许我们加密和解密秘密消息。
- 在第 15 章中，我们将编写一款高级的人机对抗的 Reversi 类游戏，其中有一个几乎无法战胜的人工智能对手。
- 第 16 章基于第 15 章的 Reversi 游戏进行了扩展，编写了多个 AI 并让其在人机对抗游戏中竞争。
- 第 17 章介绍了 Python 的 pygame 模块，并且展示了如何使用它来绘制 2D 图形。
- 第 18 章介绍了如何使用 pygame 实现图形动画。
- 第 19 章介绍了在 2D 游戏中，如何检测物体何时彼此碰撞。
- 在第 20 章中，我们将通过添加声音和图像来改进简单的 Pygame 游戏。
- 第 21 章组合了第 17 章到第 20 章的概念，开发了一款叫做 Dodger 的动画游戏。

如何使用本书

本书中的大多数章，都是以该章的特色程序的一个运行示例开始的。这个运行

示例，展示了当你运行程序的时候所看到的样子。用户输入的部分用粗体显示。

我建议你自行将每个程序的代码输入到 IDLE 的文件编辑器中，而不是下载或复制粘贴它们。如果花一些时间来录入代码的话，你将会记住更多内容。

行号和缩进

当输入本书中的源代码的时候，不要输入每行开始的行号。例如，如果你看到如下的代码行，不需要输入左边的 9.以及其后面的一个空格：

```
9. number = random.randint(1, 20)
```

应该只是输入如下内容：

```
number = random.randint(1, 20)
```

这里的行号只是为了方便在图书中引用程序中特定的行。它们并不是真正的程序源代码的一部分。除了行号，其他的地方完全按照本书代码的样子录入。注意，有些代码行有 4 个或 8 个（或者更多的）空格缩进。代码行开始处的空格是根据 Python 如何解释指令而变化的，因此，包含这些空格是很重要的。我们来看一个示例。这里的缩进空格用黑色的圆点（•）标记出来，以便你可以清晰地看到它们。

```
while guesses < 10:
••••if number == 42:
••••••••print('Hello')
```

第 1 行代码没有缩进，第 2 行代码缩进了 4 个空格，并且第 3 行代码缩进了 8 个空格。尽管本书中的示例并没有使用黑色的圆点来标记空格，但 IDLE 中的每一个字符都具有相同的宽度，因此，可以通过统计每一行之上或之下的字符数，来计算出空格的数目。

较长的代码行

有些代码指令太长了，在本书中无法放到一行之中，并且会换行到下一行。但是，这些代码行在你的计算机屏幕上显示是没有问题的，因此，输入完整的一行而不要按下回车键。通过查看左边的行号，你就知道什么时候一条新的指令开始了。如下的示例只有两条指令：

```
1. print('This is the first instruction!xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
     xxxxxxxxxxxx')
2. print('This is the second instruction, not the third instruction.')
```

第 1 条指令换行到页面中的第 2 行，但是，第 2 行并没有一个行编号，因此，

你很清楚这仍然是第 1 行的代码。

下载和安装 Python

我们需要安装一个叫做 Python 解释器的软件。解释器程序理解我们用 Python 语言编写的指令。从现在开始，我把“Python 解释器软件”直接简称为“Python”。

下面，我将介绍如何针对 Windows、OS X 或 Ubuntu 下载和安装 Python 3，特别是 Python 3.4。还有比 Python 3.4 更新一些的版本，但是我们在第 17 章到第 21 章所使用的 pygame 模块，当前只支持 Python 3.4。

Python 2 和 Python 3 之间还有一些显著的区别。本书中的程序使用 Python 3，并且，如果你试图用 Python 2 运行它们的话，将会得到错误。这一点很重要，实际上，我在本书中加入了一些企鹅的卡通图片来提醒你这一点。

在 Windows 下，从 https://www.python.org/downloads/release/python-344/下载 Windows x86-64 MSI 安装程序，然后双击它。你必须输入计算机的管理员密码。

按照安装程序在屏幕上显示的说明来安装 Python，如下所示。

1．选择 **Install for All Users**，然后单击 **Next** 按钮。

2．单击 **Next** 按钮，将程序安装在 C:\Python34 文件夹下。

3．单击 **Next** 按钮，跳过 Customize Python 部分。

在 Mac OS X 操作系统中，从 https://www.python.org/downloads/release/python-344/ 下载 Mac OS X 64-bit/32-bit 安装程序，并且双击它。按照安装程序在屏幕上显示的说明来安装 Python，如下所示。

1．如果你得到一条“‘Python.mpkg’ can’t be opened because it is from an unidentified developer”的警告，按下 **Ctrl** 键的同时用鼠标右键点击 Python.mpkg 文件，然后，从所出现的菜单中选择 Open。可能需要输入计算机管理员的密码。

2．在 **Welcome** 部分，单击 **Continue** 按钮，并且单击 **Agree** 按钮以接受许可协议。

3．选择 Macintosh HD（或者任意硬盘驱动器名称），并且单击 **Install** 按钮。

如果你使用的是 Ubuntu 操作系统，可以通过 **Ubuntu Software Center**，按照如下步骤安装 Python。

1．打开 **Ubuntu Software Center**。

2．在窗口右上角的搜索框中输入 **Python**。

3．选择 **IDLE (Python 3.4 GUI 64 bit)**。

4．单击 **Install** 按钮。可能需要输入计算机管理员的密码来完成安装。

如果在以上的步骤中遇到问题，可以通过 https://www.nostarch.com/inventwithpython/ 找到替代的 Python 3.4 安装说明。

启动 IDLE

IDLE 表示交互式开发环境（Interactive DeveLopment Environment）。对于编写 Python 程序来说，这个开发环境就像是字处理软件一样。在不同的操作系统上，启动 IDLE 的方式有所不同。

在 Windows 操作系统中，单击左下角的启动按钮，输入“IDLE”并且选择 **IDLE (Python GUI)**。

在 Mac OS X 操作系统中，打开 **Finder** 窗口，点击 **Applications**。接下来单击 **Python 3.x**。然后单击 **IDLE** 图标。

在 Ubuntu 或者 Linux 操作系统中，打开一个终端窗口，然后输入“idle3”。也可以单击屏幕上端的 **Applications**。然后单击 **Programming** 和 **IDLE 3**。

当第一次运行 IDLE 时，出现的窗口是交互式的 shell，如图 1 所示。你可以在交互式 shell 的>>>提示符后输入 Python 指令，Python 就会执行这些指令。显示完执行指令的结果之后，会出现一个新的>>>提示符，并等待下一条指令。

图 1　IDLE 程序的交互式 shell

寻求在线帮助

可以从 https://www.nostarch.com/inventwithpython/找到许多与本书相关的代码文件和其他资源。如果你想要询问和本书相关的编程问题，请访问 http://reddit.com/r/inventwithpython，或者，可以将你的编程问题通过 E-mail 发送到 al@inventwithpython.com。

当询问编程问题时，注意如下几点。

- 如果录入了本书中的程序，但是发现一个错误，在提问之前，请先通过 https://www.nostarch.com/inventwithpython#diff 的在线 diff 工具检查录入错误。复制代码并将其粘贴到 diff 工具中，以查看你的程序和本书中的代码之间的任何差异。
- 从网上查找是否有人也问过（或者回答）和你同样的问题。

记住，你将自己的编程问题描述的越好，其他人就越能够给予你帮助。当询问编程问题的时候，请做好以下的事情。

- 当描述错误的时候，说明你想要做什么。这会让帮助你的人知道你是否完全走错了路。
- 复制并粘贴完整的错误消息和代码。
- 说明你的操作系统和版本。
- 说明你已经尝试了哪些方法去解决问题。这就告诉人们，你已经做了一些工作来试图自己解决问题。
- 要有礼貌。不要命令帮助你的人或者给他们以求让其快速做出回答。

既然你已经知道了如何寻求帮助，你将立刻开始学习如何编写自己的计算机游戏。

目　录