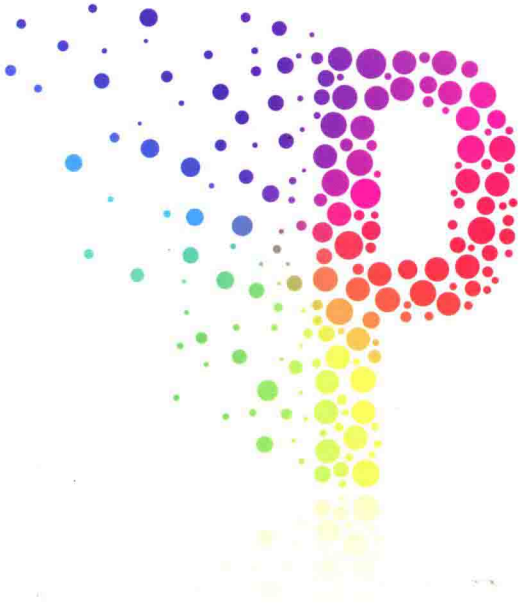
HZ BOOKS
华章科技

全面系统地讲解了Processing的功能与使用方法，配合Xtion和Microduino的Joypad平台，在Processing上实现各种有趣的案例和游戏开发例程，帮助读者快速掌握Processing开发方法。

电子与嵌入式系统
设计丛书



Processing 开发实战

黄文恺 吴 羽 伍冯洁 编著

机械工业出版社
China Machine Press

电子与嵌入式系统
设计丛书

Processing 开发实战

黄文恺 吴羽 伍冯洁 编著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Processing 开发实战 / 黄文恺，吴羽，伍冯洁编著 . —北京：机械工业出版社，2016.5
（电子与嵌入式系统设计丛书）

ISBN 978-7-111-53821-9

I. P… II. ①黄… ②吴… ③伍… III. 程序设计 IV. TP311.1

中国版本图书馆 CIP 数据核字（2016）第 104456 号

Processing 开发实战

出版发行：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：陈佳媛　　责任校对：殷　虹
印　　刷：北京市荣盛彩色印刷有限公司　　版　　次：2016 年 6 月第 1 版第 1 次印刷
开　　本：186mm × 240mm　1/16　　印　　张：13.75
书　　号：ISBN 978-7-111-53821-9　　定　　价：59.00 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）88379426　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzit@hzbook.com

前　言

机器人技术是跨学科的综合性技术，涉及的学科比较广泛，包含光学、机电一体化、电子信息、通信技术和计算机编程等专业。在机器人教学实践当中，很难把所有的学科知识都介绍给学生，全部精通更不现实。在搜寻手势控制机器人动作资料的过程中，笔者发现 Kinect 可以很方便地在 Processing 中使用，不像在其他开发平台上那么复杂。其他专业的软件开发平台，单单配置环境就要耗费大量时间。有相当一部分机器人爱好者是非计算机专业的人员，对于他们来说，专业软件开发平台的编程技术会成为其学习的障碍，从而导致他们放弃深入探究的计划。

Processing 是一门具有革命性和前瞻性的新兴计算机语言，它致力于在电子艺术的环境下介绍程序语言，并将电子艺术的概念介绍给程序员。Processing 简单易学的界面和编程风格，使很多机器人爱好者或电子制作爱好者完成机器人的控制，或实现可控的电子产品，例如控制智能家居等。笔者在学习的过程中，对 Processing 深深着迷，通过动手实践，并将 Arduino 与之结合，设计了很多有趣的产品。本书是入门书籍，重点引导读者学习 Processing 的基础知识。除了入门基础知识，本书也会介绍 Processing 如何与 Arduino 进行通信，以及如何使用 Kinect 或 Xtion 等进阶内容。更多与 Arduino 互动的例子，以及使用各种传感器开发的小游戏都收录在笔者的另一本书《Processing 与 Arduino 互动编程》中。

本书的主要内容及读者对象

本书适合零基础的人学习，没有学过 C 语言的读者可以从第一篇入门基础篇开始学习，该篇从基本的语法开始（为了能更好地向读者展示程序运行效果，该篇的部分实例会用到后面章节中的函数，读者可以暂不理会，先学习基础知识，等学习到后面章节时再深入理解），再到绘图的数学基础，循序渐进地进行介绍。第一篇的最后部分会介绍面向对象的知识，主要概述类和对象，这是比较抽象的内容，如果初学者感到难以理解可以跳

过，不影响其他部分的学习。但该部分有利于读者建立面向对象的思想，建议读者翻阅更多的资料，掌握类和对象的相关知识。第二篇是图像图形篇，有一定编程基础的读者可以直接阅读该篇。它是本书中最具魅力的篇章，学习这些章节有利于读者创造各种各样令人惊艳的图案，或定制自己的软件界面。该篇的结尾是综合实例，读者可以借助这些实例综合运用前述的知识，绘制各种动画或展现出独特的艺术视觉效果。第三篇是互动篇，该篇有鼠标、键盘的互动以及串口通信，通过实例展示 Processing 与 Arduino 的互动，包括传感器读取和摇杆的控制程序，让读者掌握两者的交互方式。第四篇是高级应用篇，主要展示如何用 Kinect 或 Xtion 进行互动编程，读者可以在此基础上自行扩展，如采用 Kinect 或华硕的 Xtion 控制机器人，甚至控制无人飞机等。

致谢

首先要感谢刘嘉杰、黄海锋、罗雯钰、肖昌伟、张雯雯、陈思强、潘强，他们牺牲了节假日时间，帮助我整理书稿，并对每一个程序进行验证。在此要感谢你们付出的努力。

其次要感谢“广州市教育局青少年科技教育计划”对本书的撰写、器材的购置提供的资助。

最后要感谢读者朋友们，感谢您花费时间和精力阅读本书。由于时间有限，书中难免存在疏漏与错误，敬请批评指正。希望有更多志同道合的朋友加入到机器人的制作与开发中来！

黄文恺

2015 年 8 月于广州大学

目　录

第一篇
入门基础篇

- 第 1 章　Processing 简介
- 第 2 章　语言基础
- 第 3 章　数学基础
- 第 4 章　类和对象

第 1 章
Processing 简介

Processing 是一种开源编程语言，专门为电子艺术和视觉交互设计而创建，其目的是通过可视化的方式辅助编程教学，并在此基础之上表达数字创意。2001 年，麻省理工学院（MIT）媒体实验室的 Casey Reas 和 Benjamin Fry 发起了此计划，其特定目标之一便是开发一个有效的工具，通过激励性的可视化反馈，帮助非程序员进行编程的入门学习。Processing 语言建立在 Java 语言的基础之上，但使用简化的语法和图形编程模型。

Processing 简单易用的特点，以及其丰富的拓展让学习者可以创造出很多富有想象力的作品。因为源自 Java，所以大量的 Java 库都可以添加进来调用，比如 Box2D、Unity 等引擎都可以在开发时调用。除了强大的 Java 库，还可以结合 Arduino，让图形化界面和硬件产生互动，添加 Kinect 或 Xtion 又可以识别人体的肢体动作并进行处理，这些丰富的拓展让 Processing 的作品充满了想象力。

1.1 初识环境

通过访问 Processing 官网 www.Processing.org/download，可以下载 Processing，请根据自己的操作系统，选择相应的 Mac、Windows、Linux 的版本。

相对于其他语言非常复杂的开发环境，Processing 开发环境非常简洁明了，如图 1-1 所示。

Processing 开发环境由代码编辑区、消息区域、控制台、管理文件的标签栏、常用指令按钮的工具栏和一排菜单构成。当程序运行时，会弹出一个新的窗口，称为显示窗口。

利用工具栏中的按钮可以实现以下功能。

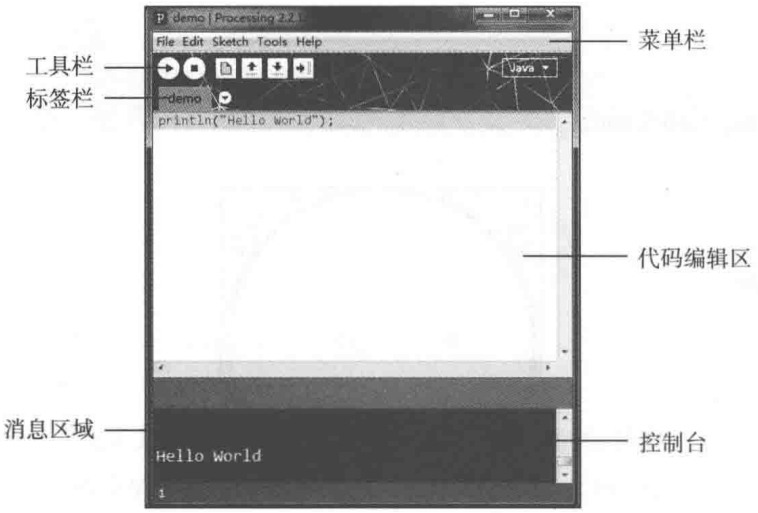


图 1-1　Processing 开发界面

：用于完成程序的检查和编译，并运行程序。

：停止运行中的程序，但不关闭显示窗口。

：新建一个 Processing 程序。

：打开文件。

：保存当前的文件。

：将当前程序以 Java 插件嵌入 HTML 文件内导出。

：建立新的标签页，可以用于创建新的类。

用 Processing 编写的程序被称作草图（sketch），这些草图是在文本编辑器里完成的。每个草图有独立的文件夹，文件扩展名 .pde 代表此文件是由 Processing 开发环境创建的。

1.2　绘制第一个图形

Processing 提供了很多默认的函数，用来绘制点、线、椭圆、矩形等。先来绘制一个简单的圆，ellipse(x,y,width,height) 是椭圆绘制函数，前两个参数设置椭圆圆心的位置，第三个参数设置长轴，第四个参数设置短轴。当参数 width 和 height 相等时得到的就是正圆。

例如，以画面中心为原点画一个直径为 100 像素的圆，如图 1-2 所示。

示例：

1-1

```
ellipse(50,50,100,100);
```

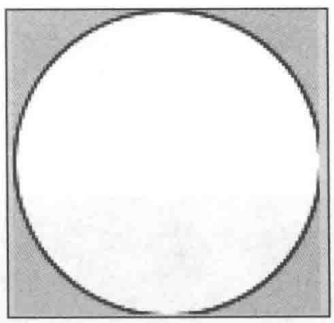


图 1-2 圆

1.3 绘制第一个动画

Processing 的文本编辑框中默认会有两个函数：setup() 和 draw()。前者之中的内容只被执行一次，后者内容会被循环执行。如下例中，设置圆的半径为连续增大的值，在画板中连续绘画呈现的结果就是一个圆逐渐变大的动画效果，如图 1-3 所示。

示例：

1-2

```
int r = 0;
void draw()
{
    ellipse(50,50,r++,r++);
}
```

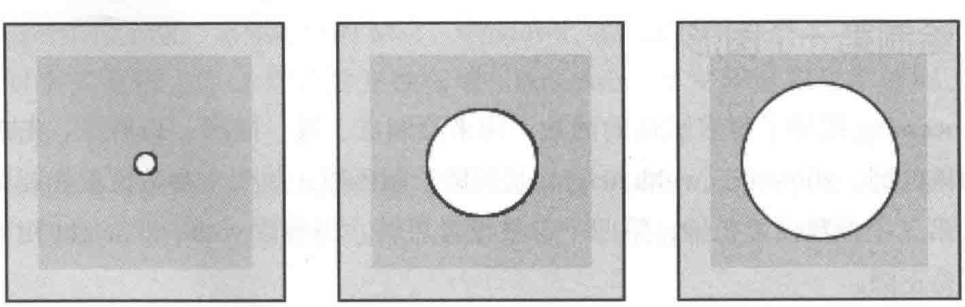


图 1-3 逐渐变大的圆

1.4　第一个交互

调用鼠标绘制一根直线，可以移动鼠标来进行交互，如图 1-4 所示。

示例：

1-3

```
void draw()
{
    line(mouseX,mouseY,0,0);      //绘制一条直线
}
```

mouseX 和 mouseY 是系统变量，也是鼠标所在的坐标位置，后面会详细说明。在绘画过程中，不消除原有的绘画留下的痕迹。此时如果想要消除原有的绘画痕迹，需要每次载入一个空白页面。只需加入一行代码即可，如图 1-5 所示。

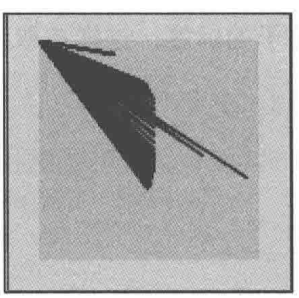


图 1-4　绘制直线

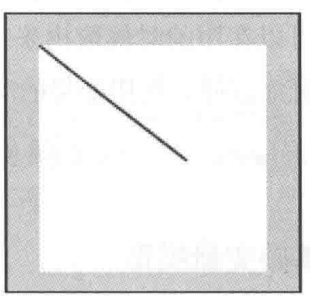


图 1-5　无痕迹绘图

示例：

1-4

```
void draw()
{
    background(255);
    line(mouseX,mouseY,0,0);      //直线
}
```

以上例子介绍了一些基础概念，并没有做详细解释，主要目的是让读者对 Processing 有初步的认识。接下来将引导读者从语言基础开始，结合简单图案绘制，循序渐进，再进阶到制作动画和交互。为了能得到最大的收获，需要读者尽可能多地上机编程操作，以快速掌握这门语言。

第 2 章
语言基础

2.1 变量

变量是储存指定数据类型的空间。在一个程序中，同一变量可以多次引用，变量储存的值也可以更改。你可以把变量理解为存储某种类型东西的箱子，可以把东西放进去，也可以在用的时候取出来。变量的名字就是箱子的标签，这样在取出里面东西的时候会更加方便。使用变量的时候要提前声明。

```
int variable;      //前面是数据类型，后面是变量的名字
```

2.1.1 基础变量类型

不同的数据类型就好像不同大小的箱子，可以放不同的数据。数据类型有基础数据类型和复杂数据类型，如表 2-1 所示。有些数据类型内存空间大小一样，但是它们的取值范围不一样，这说明类型决定了数据在内存空间中的表示方式。计算机中的数据都是以二进制位来储存的，一个位也就是 0 或 1。boolean 类型的数据只有一位，也就是占用一个二进制位来储存这种数据。

表 2-1 数据类型

数据类型	占用空间	取值范围
boolean	1 位	true 或 false
byte	8 位（1 字节）	-128 ~ 127
char	16 位（2 字节）	0 ~ 65535
color	32 位（4 字节）	0 ~ 4294967295
int	32 位（4 字节）	-2147483648 ~ 2147483647
float	32 位（4 字节）	3.4E+38 ~ 3.4E+38

2.1.2 变量名字

变量是储存数据的箱子，变量名字是箱子的标签。为了更方便地提取数据，变量的命名有一些规则。

- 变量名字在同一作用域下是独一无二的。
- 变量名字的第一个符号是字母或下划线。
- 变量名字区分大小写，不一样的大小写是不同的变量。
- 构成变量名字的只能是字母、数字、下划线。
- 变量名字最好有实际意义，能表达所储存的内容。

2.1.3 变量赋值

把数据存入变量这个箱子叫作变量赋值。通过运算符“=”来把数据存入变量。有以下两种方式来进行变量赋值。

2-1

```
int variable = 2;
```

2-2

```
int variable;
variable = 3;
```

Processing 内置了打印函数，可以用 print() 或 println() 函数把变量的数值在控制台输出。

示例：

2-3

```
int variable;
variable = 3;
println(variable);          //输出：3
```

执行上面的例子可以在文本框下的控制台上看到输出结果为 3。

2.1.4 系统变量

有一些变量是系统内置的，不需要声明。这些变量不允许赋值，这些变量的值可以让编程人员获得系统在运行时的一些参数。比如，mouseX、mouseY 的值是鼠标在

画板上停留的坐标。利用好系统变量可以更方便地设计交互效果。表 2-2 列出了部分 Processing 系统变量。

表 2-2 系统变量举例

width	sketch 窗口的像素宽度
height	sketch 窗口的像素高度
frameCount	运行的帧数
frameRate	每秒运行的帧数
screen.width	整个屏幕的像素宽度
screen.height	整个屏幕的像素高度
key	最近一次按下的键值，不包括特殊键
keyCode	按下的键对应的键盘编码，不包括小写字母，包括特殊键
keyPressed	键盘是否被按下
mousePressed	鼠标是否被按下
mouseButton	被按下的鼠标键值

示例：

下面例子打印系统变量 mouseX、mouseY，并显示在画板上。

2-4

```
void setup()
{
    size(900,600);
    textSize(26);
}
void draw()
{
    background(0);
    text("x:"+mouseX,100,200);
    text("y:"+mouseY,300,200);
}
```

还有一些关键字，比如 final，可以让变量变成常量，当后续代码再修改该变量的时候就会报错。通常用于设置不能修改的常量值。

示例：使用 final 关键字来声明变量。

2-5

```
final int a = 2;
a = 3;    // 出错
```

2.2 运算符

运算符可以对变量进行基本的运算。运算符包括基本运算符、自增自减运算符、关系运算符逻辑运算符和条件运算符。

2.2.1 基本运算符

表 2-3 所示是 Processing 的基本运算符。

表 2-3 运算符

运算符	名 称	使用方法
+	加	a+b
-	减	a-b
*	乘	a*b
/	除	a/b
%	求余	a%b

加减乘除是常用的基本运算，求余是两个数相除所得的余数。一般都是同类型的数据通过运算符进行运算得出结果。比如两个 int 类型的 1 相加得出 2。

示例：

2-6

```
int a = 1;
int b = 1;
int c = a+b;
println(c);          // 输出：2
```

同一数据类型的相加得到相同数据类型的结果。还有一些特例是不同数据类型的数相加，如 float 类型和 int 类型的数相加，得到的是 float 数据类型的结果。后面章节会介绍运算符的其他作用，例如：在字符串中可以连接组合字符串。

2.2.2 自增自减运算符

自增“++”、自减“--”运算是非常方便的操作，可以把变量的值自动增加 1 或者减少 1。

示例：