

Processing

创意编程与交互设计

赵 婷 李 莹 王志新 编著



清华大学出版社
北 京

内 容 简 介

Processing是以数字艺术为背景的程序设计语言,语法简洁易学,使用它可以很方便地创作震撼的视觉表现及互动媒体作品。本书重点引导读者学习Processing的入门基础知识,同时介绍Processing如何与Arduino进行通信,以及如何通过与Kinect和Leap Motion等体感系统互动创作作品。

本书包含两个部分:第一部分为图形生成篇,通过大量的编程示例,带领读者从入门的图形生成方法开始,从基本的语法再到程序绘图,直到创作出各种令人惊艳的图案,或者定制自己的UI界面;第二部分为交互设计篇,重点讲解鼠标、键盘的互动及串口通信,通过实例展示Processing和Arduino的互动,以及如何与Kinect和Leap Motion进行互动编程,帮助读者创作更多的体感互动效果。

为便教力学,书中附赠教学课件及全部程序代码源文件,读者可扫描前言中的二维码获取。

本书既可作为新媒体、艺术设计等相关专业学生的学习用书,也可供设计师、程序员和艺术工作者等阅读参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。举报:010-62782989, beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

Processing创意编程与交互设计 / 赵婷, 李莹, 王志新编著. —北京: 清华大学出版社, 2022.1
ISBN 978-7-302-59131-3

I. ①P… II. ①赵… ②李… ③王… III. ①程序设计 IV. ①TP311.1

中国版本图书馆CIP数据核字(2021)第179216号

责任编辑: 李 磊

封面设计: 杨 曦

版式设计: 思创景点

责任校对: 马遥遥

责任印制: 宋 林

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦A座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 大厂回族自治县彩虹印刷有限公司

经 销: 全国新华书店

开 本: 170mm×240mm 印 张: 15 字 数: 338千字

版 次: 2022年1月第1版 印 次: 2022年1月第1次印刷

定 价: 69.00元

产品编号: 090901-01

PREFACE

前言



随着计算机的普及和互联网数字化的发展，不同职业的人在各自的领域中开始通过编程来解决问题，探索各种可能性，甚至是发展出新的理论体系。编程也变得像是现实世界与数字世界沟通的通用语言，因此掌握编程语言成为信息时代的一种必要技能。

在艺术和设计领域，有很多艺术家、设计师、程序员和教育工作者，他们通过不断探索编程在艺术和设计中的应用，创造出了专门供艺术家、设计师及相关专业学生等使用的创意编程工具，如 Processing、openFrameworks、Cinder、p5.js 和 TouchDesigner 等。这些工具在生成艺术、视听艺术、数据可视化、交互艺术等各个领域得到了充分的应用和发展，其理论与技术也成为很多艺术院校的必修课程。

在这个数字媒介的时代，Processing 这样的开源软件对于艺术家和设计师来说有着突破性的意义，它超越了既定的运算处理规则，让用户可以更自由地使用计算机语言，利用计算机的性能去表现自己对数字媒介的理解和创意。

Processing 是一门具有革命性和前瞻性的新兴计算机语言，它使应用编程实现交互图形变得更加容易。该语言是以数字艺术为背景的程序设计语言，是 Java 语言的延伸，支持许多现有的 Java 语言架构，但语法更加简单。它具有跨平台的特点（支持 Windows、iOS 和 Android），对 OpenNI、OpenCV 和 Kinect 有良好的支持，除了可以很方便地创作震撼的视觉效果和互动媒体作品外，还可以实现诸如图形处理和人工智能等高级应用。

本书适合零基础的人学习，内容包含图形生成篇和交互设计篇两部分。第一部分是图形生成篇，从基本的语法开始讲解，再到绘图的数学基础知识，循序渐进。每一章的扩展练习，通过实例综合运用前述的知识，绘制各种动画或展现独特的艺术视觉效果。学习该部分知识，能够帮助读者创造出各种令人惊艳的图案，或者定制自己的软件界面。第二部分是交互设计篇，该篇内容包括鼠标、键盘的互动及串口通信，通过实例展示 Processing 和 Arduino 的互动，包括传感器读取等的控制程序，让读者掌握两者的交互方式，还展示了如何用 Kinect 和 Leap Motion 进行互动编程，读者可以在此基础上自行扩展，创建更多的体感互动效果。

为便于读者学习和训练，以及教师教学，本书附赠全部程序代码源文件、教学课件，扫描右侧二维码即可获取。由于本书为双色印刷，所以部分图片效果无法完全呈现，读者可扫描书中二维码，查看效果。

本书作者在生成艺术、数据可视化和声音可视化领域探索多年，在学习 Processing 的过程中，通过大量的动手实践，汇集了一些经验和编程实例，创作了很多有趣的作品，希望通过本书呈现给读者朋友。

资源

作者

2021.10

目录 CONTENTS

图形生成篇

第 1 章 初识图形交互设计

2

- 1.1 了解交互设计 2
- 1.2 Processing 的功能与特性 3
- 1.3 开启第一个绘图程序 5
- 1.4 扩展练习 7

第 2 章 绘制图形

8

- 2.1 画布 8
- 2.2 基本图形 9
- 2.3 自定义形状 15
- 2.4 创建文字 17
- 2.5 绘图顺序 20
- 2.6 颜色填充 21
- 2.7 描边属性 23
- 2.8 扩展练习 25

第 3 章 变量与语法

27

3.1 了解变量	27
3.2 创建变量	29
3.3 系统变量	30
3.4 简单运算	32
3.5 条件语句	34
3.6 for 循环	37
3.7 注释	41
3.8 映射	42
3.9 扩展练习	43

第 4 章 动画与几何变换

45

4.1 移动	45
4.2 函数	48
4.3 随机	50
4.4 平移	56
4.5 旋转	57
4.6 缩放	59
4.7 隔离	60
4.8 缓动	62
4.9 扩展练习	64

第 5 章 函数和对象

67

5.1 函数	67
5.2 面向对象编程	73
5.3 提前下雪	80
5.4 扩展练习	82

第 6 章 数组

84

6.1 数组的概念	84
-----------------	----

6.2 创建数组	86
6.3 对象数组	89
6.4 数组函数	91
6.5 扩展练习	93

第 7 章 3D 图形

95

7.1 3D 坐标系	95
7.2 三维灯光	100
7.3 材质	103
7.4 摄像机	105
7.5 扩展练习	107

第 8 章 粒子系统

108

8.1 粒子基础	108
8.2 粒子运动	111
8.3 互动粒子	114
8.4 连续粒子流	116
8.5 扩展练习	119

第 9 章 媒体处理

122

9.1 加载图片	122
9.2 加载滤镜与蒙版	129
9.3 加载视频	131
9.4 像素化效果	135
9.5 加载音频	138
9.6 扩展练习	144

第 10 章 使用库创作

146

10.1 扩展库概述	146
10.2 库的安装和调用	147

10.3 库的应用范例 149

10.4 扩展练习 155

交互设计篇

第 11 章 互动响应 160

11.1 鼠标响应 160

11.2 键盘响应 165

11.3 声音交互 169

11.4 时间触发 171

11.5 扩展练习 174

第 12 章 Arduino 互动基础 175

12.1 Arduino 入门 175

12.2 互动装置解析 177

12.3 Processing 与 Arduino 通信 181

12.4 扩展练习 186

第 13 章 Arduino 互动编程实例 188

13.1 Arduino 程序架构 188

13.2 Arduino 编程语法 190

13.3 光敏控制粒子 197

13.4 距离控制视频 200

13.5 红外感应控制电机 204

13.6 扩展练习 206

第 14 章 Kinect 与体感互动

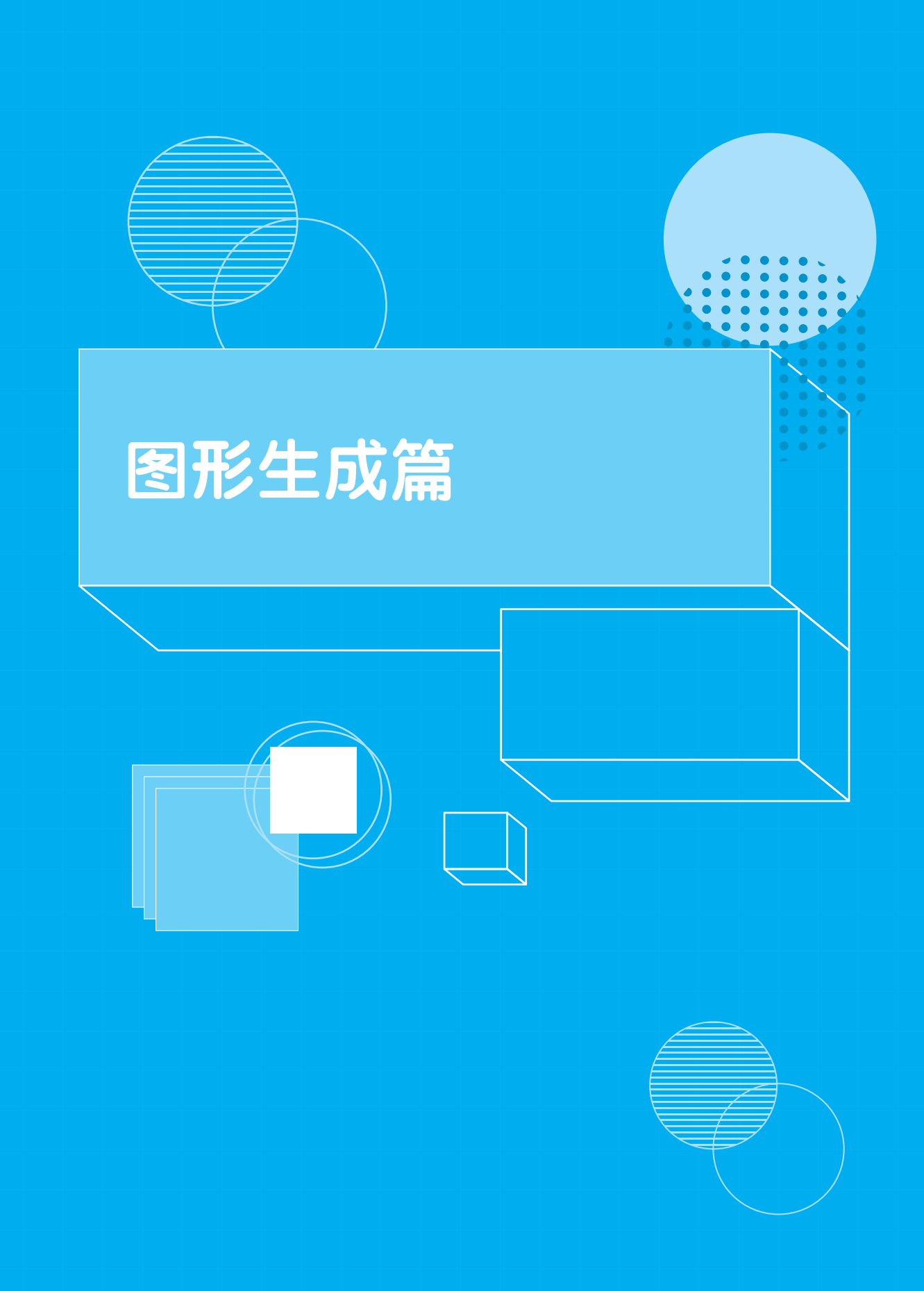
208

14.1 Kinect 简介	209
14.2 Kinect 相关驱动	210
14.3 多维图像信息	211
14.4 实时图像处理	214
14.5 身体追踪	216
14.6 扩展练习	218

第 15 章 Leap Motion 手势互动

220

15.1 Leap Motion 简介	220
15.2 安装与调试 Leap Motion	221
15.3 手势捕捉基础操作	224
15.4 手势互动	226
15.5 扩展练习	228



图形生成篇

第1章

初识图形交互设计

Processing 是一款比较容易入手的面向视觉设计的软件原型工具，其完全开源、高分辨率渲染和完美的硬件接口能力，使得设计师能够轻松掌握编程的技巧，只需要用很少的代码就可以完成很复杂的计算机交互作品。在创意视觉领域，Processing 拥有举足轻重的地位。

1.1 了解交互设计

设计是一门尽力满足传达要求的学问，设计师通过作品向用户阐释事物的功能、功效、使用方法、特性、技术水平和文化背景等含义。随着数字科技和人工智能的发展，信息产品的电子化使得传达的载体不再只是一个固化模式，而是一个需要用户参与的、多态的响应式系统。用户对一个设计作品的理解，需要通过与作品的交互，即相互传达才得以完美实现。

交互是人类生存和发展的需要，如今人们每天都接收和创造着大量的信息和数据，比如每日新闻、广告信息、数据重现、可视化分析、朋友圈人际关系拓扑图、科学实验模拟等。当系统将信息传送给用户，或用户将信息传送给系统时，便发生了交互，这些信息可以是文本、语音、色彩、图形、机械和物理的输入或反馈。正是因为有了交互，才使人类感知、认识和控制外部世界的能力不断扩展。

随着数字科技的发展和人们审美水平的提高，国内互动艺术逐渐开始发展，各大院

校也纷纷设立相应的课程。在早期交互中,媒体设计师偏爱用 Flash 的 ActionScript 语言实现互动网络和界面,但是 Flash 并非一个完全开源的系统,并且很多浏览器不支持。对比来看,Processing 是一个设计师比较容易入手的互动编程工具,由于 Processing 能够和多种硬件接口完美结合,再加上其高分辨率的渲染效果,使其在创意视觉方面发挥了重要的作用。

1.2 Processing 的功能与特性

Processing 是面向视觉设计的软件原型工具,它提供了一个基于 Java 的结构性的程序框架,并将视觉设计师常用的一些图形图像操作封装为 API(application programming interface,应用程序接口),这样设计师和艺术家就可以用很少的代码完成一个复杂的计算机互动艺术品。由于 Processing 是在 Java 基础上开发的,因此具有良好的跨平台特性,用 Processing 编写的软件可以在 iOS、Windows 和 Android 等操作系统上运行。

在一些硬件相关的项目中,Arduino 和 Processing 常常通过串口连接在一起,协同完成一个交互作品。Processing 逐渐成为计算机端构建互动界面原型的首选工具,一旦规划好传达需求的功能,交互设计师就可以利用大量现有的函数库或开源项目代码快速完成原型验证。

Processing 开发环境 (processing development environment, PDE),可以从它的官方网站上下载安装,如图 1-1 所示。

在官方网站上还有详细的教程,使用者在开始编写自己的程序前,可以先跟随教程尝试编写其中的案例,如图 1-2 所示。

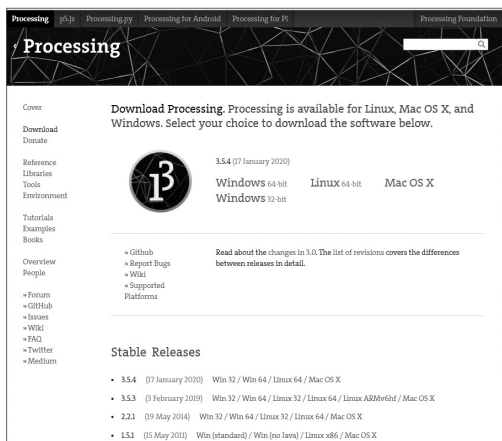


图 1-1

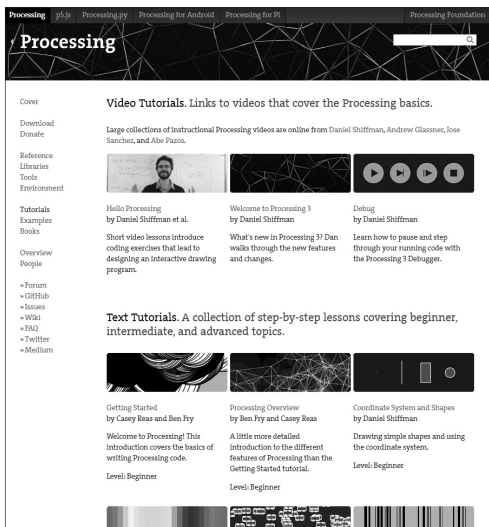


图 1-2

PDE 主要有两个窗口——运行窗口和编辑窗口。运行窗口用来运行编辑好的程序效果；编辑窗口主要用来编辑代码和发布程序，包括工具栏、标签栏、代码编辑区、消息区和控制台，如图 1-3 所示。



图 1-3

1. 工具栏

工具栏中包含了运行、停止、新建、打开、保存、发布等按钮，主要功能如下。

- 运行：单击【运行】按钮可以运行程序。运行时会弹出运行窗口，用户可以通过运行窗口来观察程序的视觉效果和交互效果。
- 停止运行：单击【停止运行】按钮，可以关闭运行窗口。
- 开发模式：单击右上角的【Java】按钮，可以切换开发模式。Processing 支持多种开发模式，可以通过单击添加模式 (JavaScript、Android、Python 等)，添加后需要重新运行 Processing 才能显示。

2. 标签栏

单击标签栏上的▼下三角按钮，在弹出的菜单中选择【新建标签】命令，如果代码有很多行，就可以用它来扩展程序。不过一般我们会用它来定义一个单独的类，如图 1-4 所示。

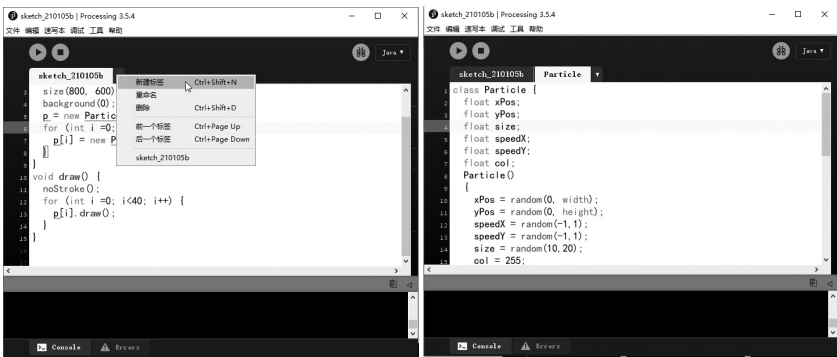


图 1-4

3. 代码编辑区

所有的程序代码都在代码编辑区编辑。程序会用不同的颜色来区分数据类型、系统

变量、系统常量、系统函数、语句等。

4. 消息区

消息区主要显示程序编译时的错误（语法错误），还有一些提示消息，如图 1-5 所示。

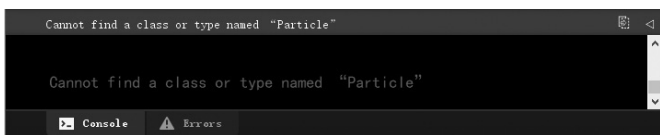


图 1-5

5. 控制台

控制台用于显示程序运行时的错误，还可以用 `print()` 或 `println()` 函数在这里输出信息，如图 1-6 所示。



图 1-6

1.3 开启第一个绘图程序

下面我们开始写第一行代码，让 Processing 向世界问声好。打开 Processing，在代码编辑区输入下面的代码：

```
Println("Hello,world");
```

单击【运行】按钮后，在控制台输出“Hello,world”（不包括双引号），如图 1-7 所示。

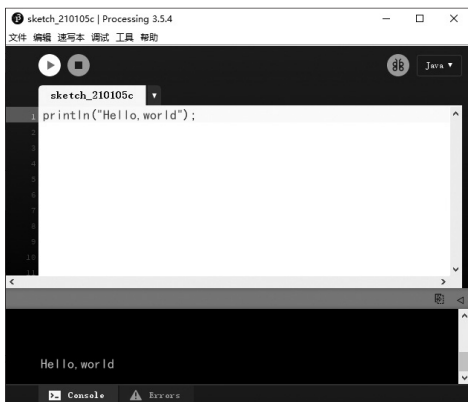


图 1-7

下面我们绘制一个直径为 200 像素的圆，设置其原点在画布的中心，如图 1-8 所示。

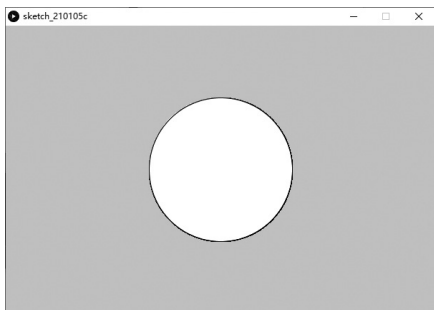


图 1-8

在编辑区输入代码如下：

```
void setup() {
  size(600,400);
}
void draw() {
  ellipse(300,200,200,200);
}
```

让这个圆动起来也很容易，比如创建这个圆由小到大的动画，如图 1-9 所示。

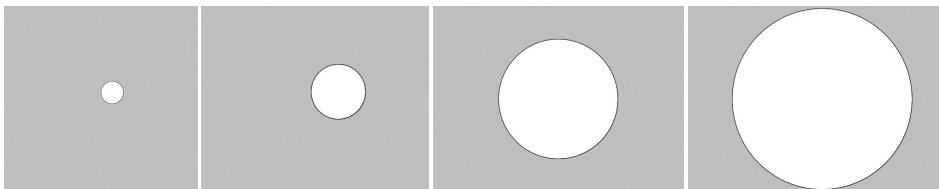


图 1-9

在编辑区输入代码如下：

```
float diameter=0;
void setup() {
  size(600,400);
}
void draw() {
  ellipse(300,200,diameter,diameter);
  diameter++;
}
```

接下来我们进一步尝试交互效果，在前面绘制的基础上添加鼠标交互，圆跟随着鼠标移动并逐渐放大，如图 1-10 所示。

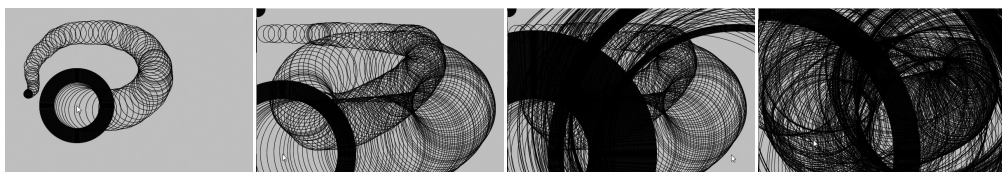


图 1-10

在编辑区输入代码如下：

```
float diameter=0;
void setup() {
  size(600,400);
}
void draw() {
  noFill();
  ellipse(mouseX,mouseY,diameter,diameter);
  diameter++;
}
```

上面的圆形跟随鼠标的移动而改变原点位置，是因为把它的特征属性“位置”设置成了随着时间变化的量，并且变化的量由鼠标位置来决定，这就是在程序动画的基础上添加交互因素的设计方案。

我们已经学习了第一个图形、第一个动画、第一个交互。本章不要求读者完全理解上面例子中每一行代码的原理，主要目的是对交互动画有个初步的了解，并且能够带着交互动画的概念来完成本书的学习。

1.4 扩展练习

通过前面的讲解，读者应该对交互设计和图形编程有了一些了解。本节课堂练习的任务就是登录 Processing 官网，下载与自己电脑系统对应的版本并安装 Processing，通过浏览官网上的范例和教程快速了解 Processing 的功能和特性，然后输入前面的代码，运行并查看图形效果。

打开安装好的 Processing，新建草图，输入如下代码，初步体验鼠标的交互：

```
void setup() {
    size(720, 720);           // 设置画布尺寸
    noCursor();               // 不显示光标
}

void draw() {
    colorMode(HSB, 360, 100, 100); // 定义颜色模式
    rectMode(CENTER);          // 确定矩形的轴心模式
    noStroke();                // 没有描边
    background(mouseY/2, 100, 100); // 背景颜色随鼠标 Y 轴坐标变化
    fill(360-mouseY/2, 100, 100); // 矩形填充颜色随鼠标 Y 轴坐标变化
    rect(360, 360, mouseX+1, mouseY+1); // 矩形大小跟随鼠标 X 轴坐标变化
}
```

提示： // 后面的文字为注释，方便理解语句的含义，程序运行时并不参与执行。

运行该程序，查看效果如图 1-11 所示。

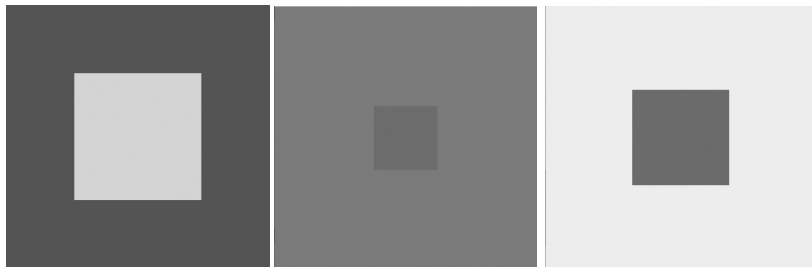


图 1-11



扫码看效果