

· 开发宝典丛书 ·

以全新的Xcode 6为开发环境，详细讲解全新的iOS 8应用开发
通过117个实例全面展现iOS开发中最常见的14类界面模块

iOS开发 范例实战宝典

(基础篇)

杨佩璐 魏彩娟 编著



(附赠51CTO学院学习卡)

- ✓ **实例丰富**：详解117个iOS经典实例的开发过程，提高实战开发水平
- ✓ **涵盖广泛**：涵盖按钮、滑块、开关、进度条、指示器、选择器、视图和分段控件等iOS开发中最常见的14类界面模块
- ✓ **由浅入深**：实例按照“实现原理、实现过程、重点代码”的编排顺序讲解，很容易掌握
- ✓ **代码精简**：精简结构性代码，只保留关键代码和核心代码，以节省篇幅，让本书更超值
- ✓ **配流程图**：为复杂的实例配有详细的程序流程图，以帮助读者轻松理解程序的执行过程
- ✓ **重点讲解**：对每个实例的核心功能都给予了专门讲解，以便于读者更好地掌握
- ✓ **最新技术**：书中的实例完全适用于全新的iOS 8开发平台，也兼容iOS 7开发平台
- ✓ **答疑解惑**：提供了QQ群、技术论坛和E-mail等完善的学习交流和沟通方式（见前言中的说明）



清华大学出版社

· 开发宝典丛书 ·

iOS 开发 范例实战宝典

(基础篇)

杨佩璐 魏彩娟 编著

清华大学出版社

内 容 简 介

《iOS 开发范例实战宝典》分为基础篇和进阶篇两个分册，其内容包含了 iOS 开发必知必会的 240 个经典实例和几百个开发模块。书中的实例紧跟技术趋势，以最新的 iOS 8 为版本编写，内容覆盖了 iOS 开发的方方面面，几乎涉及 iOS 开发的所有重要知识。书中给出了每个实例的具体实现过程，并对程序代码做了详细注释，对其中的重点和难点进行了专门分析，而且精讲每个实例的重点代码，读者可以在这些实例的基础上做出更多更新的功能。

本书为《iOS 开发范例实战宝典（基础篇）》，共 13 章，包含了 117 个开发实例。其中包括 7 个按钮类实例、9 个滑块类实例、2 个开关类实例、7 个进度条类实例、8 个指示器类实例、6 个选择器类实例、4 个视图类实例、3 个分段控件类实例、11 个导航栏类实例、6 个标签栏类实例、11 个菜单类实例、6 个提醒对话框类实例、18 个文本处理类实例和 19 个表实例。

本书涉及面广，涉及 iOS 软件开发的各种常用应用。适合所有想全面学习 iOS 开发技术的人员阅读，也适合 iOS 专业开发人员作为案头必备的参考书。

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。侵权举报电话：010-62782989 13701121933

图书在版编目（CIP）数据

iOS 开发范例实战宝典. 基础篇 / 杨佩璐, 魏彩娟编著. —北京：清华大学出版社，2015
（开发宝典丛书）

ISBN 978-7-302-39576-8

I. ①i… II. ①杨… ②魏… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字（2015）第 046516 号

责任编辑：杨如林

封面设计：欧振旭

责任校对：胡伟民

责任印制：沈 露

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 刷 者：清华大学印刷厂

装 订 者：三河市新茂装订有限公司

经 销：全国新华书店

开 本：185mm×260mm 印 张：43.25 字 数：1080 千字

版 次：2015 年 5 月第 1 版 印 次：2015 年 5 月第 1 次印刷

印 数：1~3500

定 价：99.80 元

前言

移动应用开发是当前 IT 开发的热点。由于苹果提供了完备的开发工具和成熟的软件盈利方式，苹果的 iOS 开发成为热点中的热点。苹果开发技术较为封闭，尤其是相对于开源技术的 Android 开发而言更是如此。同时，由于移动开发的发展时间较短，这使得开发资料相对匮乏，尤其是实用性比较强的开发资料更是为数不多，开发者往往缺乏应用指导资料。

笔者结合自身多年的 iOS 开发经验和心得体会，花费了一年多的时间分析了 iOS 开发中常见的几百个应用场景，并进行了精心整理，挑选了最为典型的 240 个 iOS 开发实例，编写成了《iOS 开发范例实战宝典》（分为基础篇和进阶篇两个分册）。

本书为《iOS 开发范例实战宝典（基础篇）》，包含了 117 个经典实例，涉及 iOS 开发中较为基础的 14 个界面开发专题。希望各位读者能在本书的引领下跨入 iOS 开发的大门，并成为一名开发高手。

本书特色

1. 实例丰富，代码精讲

本书详细讲解了 117 个 iOS 开发经典实例，并对重点代码做了大量注释和讲解，以便于读者更加轻松地学习。通过对这些实例的演练，可以快速提高读者的开发水平。

2. 内容全面，涵盖广泛

本书全面介绍了 iOS 开发中最为常见的 14 类界面模块，包括按钮、滑块、开关、进度条、指示器、选择器、视图、分段控件、导航栏、标签栏、菜单、提醒对话框、文本处理和表。这些内容是 iOS 开发必知必会的内容，需要读者重点掌握。

3. 讲解详细，循序渐进

本书中的每个实例都给出了详细的分析过程和实现步骤，书中的每个实例都按照“实现原理→实现过程→重点代码”三个步骤进行分析。对于复杂的实例，还给出了完备的流程图来帮助读者理解实例的工作机制，掌握起来更加容易。

4. 专注核心，举一反三

为了在有限的篇幅内讲解更多的开发实例，本书只给出了每个实例的核心代码及分析。完整的实例代码读者可以自己下载阅读，并进行测试和练习，而且还可以对这些代码进行改造，以用于实际的开发之中，从而起到举一反三的作用。

本书内容及体系结构

第 1 章 按钮类效果

本章 7 个实例，主要内容包括发光的按钮、弹出式按钮、超文本链接和抛光效果的按钮等内容。通过本章的学习，读者可以实现各种常见按钮的开发。

第 2 章 滑块类效果

本章 9 个实例，主要内容包括具有多个颜色的滑块控件、环形滑块控件、具有范围的滑块控件以及具有文字的滑块控件等内容。通过本章的学习，读者可以掌握滑块类控件的技术与应用。

第 3 章 开关类

本章 2 个实例，主要内容包括自定义开关的外观、实现滑块窗口滑动切换的效果。通过本章的学习，读者可以掌握开关的一些常见技术及应用。

第 4 章 进度条类和指示器类效果

本章 15 个实例，主要内容包括扁平带梯度效果的进度条、扇形进度条、环形进度条、具有范围的进度条、文本上传进度条、倒计时进度条、带进度条的工具栏、变色的指示器，以及仿 Facebook web 上正在加载中的效果等内容。通过本章的学习，读者可以掌握进度条类和指示器类的各种常见技术及应用。

第 5 章 选择器类效果

本章 6 个实例，主要内容包括时间设置器、闹铃、城市经纬度查询、定制多选功能选择器、转盘选择器和老虎机。通过本章的学习，读者可以掌握选择器一些常见技术及应用。

第 6 章 视图

本章 4 个实例，主要内容包括更改空白视图的背景颜色、关闭应用程序、手电筒及旋转大挑战。通过本章的学习，读者可以掌握视图的一些常见技术及应用。

第 7 章 分段控件

本章 3 个实例，主要内容包括滑块式分段控件、开关式分段控件和自定义分段控件。通过本章的学习，读者可以掌握关于分段控件的一些常见技术及应用。

第 8 章 导航栏

本章 11 个实例，主要内容包括具有阴影的导航栏、具有图片的导航栏、具有分段控件的导航栏、具有子标题的导航栏、上下滑动的导航栏和具有下拉菜单的导航栏等内容。通过本章的学习，读者可以掌握关于导航栏的一些常见技术及应用。

第9章 标签栏

本章6个实例，主要内容包括右上角带有数字的标签栏、具有渐变效果的标签栏、中间凸起的标签栏、标签栏控制器实现的视图切换效果、具有动画效果的标签栏以及滚动的标签栏。通过本章的学习，读者可以掌握关于标签栏的一些常见技术及应用。

第10章 菜单

本章11个实例，主要内容包括立方体菜单、仿 Windows 8 的 Metro 风格、下拉菜单、浮动的菜单、具有按钮的菜单、仿 Tumblr iOS App 菜单、边栏菜单和九宫格菜单等内容。通过本章的学习，读者可以掌握关于菜单的一些常见技术及应用。

第11章 提醒对话框

本章6个实例，主要内容包括具有文本框的警告视图、全屏的警告视图、具有进度条的警告视图、具有列表的警告视图、坠落的警告视图、自定义的动作表单、弹出视图，以及模糊界面背景。通过本章的学习，读者可以掌握关于提醒对话框的一些常见技术及应用。

第12章 文本处理

本章18个实例，主要内容包括具有多个颜色的标签、发光的标签、循环渐变的标签、滚动的标签、具有光晕效果的标签和标签云等内容。通过本章的学习，读者可以掌握关于文本处理的一些常见技术及应用。

第13章 表

本章19个实例，主要内容包括邮编查询、水平列表、表的自动调整、排排看、归类、自定义索引的表、自制的列表单选控件、下拉刷新列表、背景随动和卡片插入式列表等内容。通过本章的学习，读者可以掌握关于表的一些常见技术及应用。

本书读者对象

- ☐ 想全面学习 iOS 开发技术的人员；
- ☐ iOS 专业开发人员；
- ☐ iOS 开发爱好者；
- ☐ 大中专院校的学生；
- ☐ 社会培训班学员；
- ☐ 需要一本案头必备手册的程序员。

本书配套资源获取方式

本书提供以下的配套资源：

- ☐ 本书开发环境；
- ☐ 本书实例源代码；

为了节省读者的购书开支，本书放弃以配书光盘的方式提供这些资源，而是改为采用提供下载的方式。读者可以登录清华大学出版社网站（www.tup.com.cn），搜索到本书页面，然后按照提示下载，也可以在本书服务网站（www.wanjuanchina.net）的相关版块上下载这些配套资源。

本书售后服务方式

编程学习的最佳方式是共同学习。但是由于实际环境所限，大部分读者都是独自前行。为了便于读者更好地学习 iOS 语言，我们构建了多样的学习环境，力图打造立体化的学习方式，除了对内容精雕细琢之外，还提供了完善的学习交流和沟通方式。主要有以下几种方式：

- ❑ 提供技术论坛 <http://www.wanjuanchina.net>，读者可以将学习过程中遇到的问题发布到论坛上以获得帮助。
- ❑ 提供 QQ 交流群 336212690，读者申请加入该群后便可以和作者及广大读者交流学习心得，解决学习中遇到的各种问题。
- ❑ 提供 book@wanjuanchina.net 和 bookservice2008@163.com 服务邮箱，读者可以将自己的疑问发电子邮件以获取帮助。

本书作者

本书主要由山东中医药大学的杨佩璐和河南牧业经济学院的魏彩娟编写。其中，杨佩璐编写了第 1~7 章，魏彩娟编写了第 8~13 章。其他参与编写的人员有陈晓建、陈振东、程凯、池建、崔久、崔莎、邓凤霞、邓伟杰、董建中、耿璐、韩红轲、胡超、黄格力、黄缙华、姜晓丽、李学军、刘娣、刘刚、刘宁、刘艳梅、刘志刚、司其军、滕川、王连心、沃怀凯、闫玉宝。另外，刘媛媛负责了各款 iOS 硬件环境下的代码验证和调试。

虽然笔者对本书中所述内容都尽量核实，并多次进行文字校对，但因时间所限，可能还存在疏漏和不足之处，恳请读者批评指正。

编者

目 录

第 1 章 按钮类效果	1
实例 1 发光的按钮	1
实例 2 弹出式按钮	3
实例 3 超文本链接	11
实例 4 抛光效果的按钮	14
实例 5 具有进度条的按钮	17
实例 6 唱碟机按钮	25
实例 7 环形按钮	31
第 2 章 滑块类效果	40
实例 8 具有多个颜色的滑块控件	40
实例 9 环形滑块控件	43
实例 10 具有范围的滑块控件	49
实例 11 具有文字的滑块控件	56
实例 12 自定义的滑块控件	62
实例 13 自定义的声音调节滑块控件	71
实例 14 具有步长的滑块控件	75
实例 15 模拟现实音量控制条	80
实例 16 iOS 视频修剪控件	87
第 3 章 开关类	101
实例 17 自定义开关的外观	101
实例 18 实现滑块窗口滑动切换的效果	109
第 4 章 进度条类和指示器类效果	116
实例 19 扁平带梯度效果的进度条	116
实例 20 扇形进度条	125
实例 21 环形进度条	129
实例 22 具有范围的进度条	135
实例 23 文本上传进度条	143
实例 24 倒计时进度条	151
实例 25 带进度条的工具栏	158
实例 26 变色的指示器	164
实例 27 仿 Facebook web 上正在加载中的效果	167

实例 28	Windows Phone 风格的指示器	174
实例 29	三个方块组成的指示器	179
实例 30	三个点的指示器	182
实例 31	多个方块组成的指示器	187
实例 32	仿 Yahoo 天气应用的加载效果	190
实例 33	消息提示指示器	197
第 5 章	选择器类效果	203
实例 34	时间设置器	203
实例 35	闹铃	206
实例 36	城市经纬度查询	211
实例 37	定制多选功能选择器	216
实例 38	转盘选择器	228
实例 39	老虎机	235
第 6 章	视图	245
实例 40	更改空白视图的背景颜色	245
实例 41	关闭应用程序	247
实例 42	手电筒	252
实例 43	旋转大挑战	254
第 7 章	分段控件	263
实例 44	滑块式分段控件	263
实例 45	开关式分段控件	273
实例 46	自定义分段控件	289
第 8 章	导航栏	295
实例 47	具有阴影的导航栏	295
实例 48	具有图片的导航栏	296
实例 49	具有分段控件的导航栏	300
实例 50	具有子标题的导航栏	302
实例 51	上下滑动的导航栏	310
实例 52	具有下拉菜单的导航栏	313
实例 53	具有页面控件的导航栏	327
实例 54	包含多个按钮的导航栏	332
实例 55	导航栏的颜色调节	338
实例 56	滚动的导航栏	343
实例 57	具有导航记录的导航栏	351
第 9 章	标签栏	368
实例 58	右上角带有数字的标签栏	368
实例 59	具有渐变效果的标签栏	370

实例 60	中间凸起的标签栏	373
实例 61	标签栏控制器实现的视图切换效果	379
实例 62	具有动画效果的标签栏	384
实例 63	滚动的标签栏	388
第 10 章	菜单	396
实例 64	立方体菜单	396
实例 65	仿 Windows 8 的 Metro 风格	400
实例 66	下拉菜单	403
实例 67	浮动的菜单	409
实例 68	具有按钮的菜单	417
实例 69	仿 Tumblr iOS App 菜单	425
实例 70	边栏菜单	435
实例 71	九宫格菜单	440
实例 72	侧面弹出式菜单	450
实例 73	分享菜单	456
实例 74	扇形菜单	463
第 11 章	提醒对话框	471
实例 75	具有文本框的警告视图	471
实例 76	全屏的警告视图	472
实例 77	具有进度条的警告视图	477
实例 78	具有列表的警告视图	482
实例 79	坠落的警告视图	488
实例 80	弹出视图, 模糊界面背景	493
第 12 章	文本处理	506
实例 81	具有多个颜色的标签	506
实例 82	发光的标签	510
实例 83	循环渐变的标签	514
实例 84	滚动的标签	517
实例 85	具有光晕效果的标签	519
实例 86	标签云	524
实例 87	自动计算文本长度	530
实例 88	仿 QQ 登录	533
实例 89	阅读浏览器	540
实例 90	艺术字	544
实例 91	网址管理器	551
实例 92	拨号器	555
实例 93	我的邮箱管理器	557
实例 94	数字天才	563

实例 95	九宫格	568
实例 96	单位换算器	574
实例 97	计算器	577
实例 98	表情键盘	580
第 13 章	表	587
实例 99	邮编查询	587
实例 100	水平列表	590
实例 101	表的自动调整	592
实例 102	排排看	595
实例 103	归归类	599
实例 104	自定义索引的表	604
实例 105	自制的列表单选控件	611
实例 106	下拉刷新列表	617
实例 107	背景随动	620
实例 108	卡片插入式列表	624
实例 109	嵌套的表	626
实例 110	仿 QQ 聊天	630
实例 111	树形展开列表	640
实例 112	圆角表视图	646
实例 113	表单元格的自定义折叠	651
实例 114	具有搜索功能的表视图	655
实例 115	自定义表单元格的动画效果	660
实例 116	两个列表的显示	666
实例 117	表单元格内容的复制	672

第 1 章 按钮类效果

按钮是用户交互的最基础控件。即使是在 iPhone 或者 iPad 中，用户使用最多的操作也是单击。而单击操作最多的控件往往是按钮控件。本章将主要讲解按钮控件相关的实例。

实例 1 发光的按钮

【实例描述】

本实例实现的功能是当用户单击按钮时，按钮会发光。以这样的方式，提示用户已经单击按钮了。运行效果如图 1.1 所示。

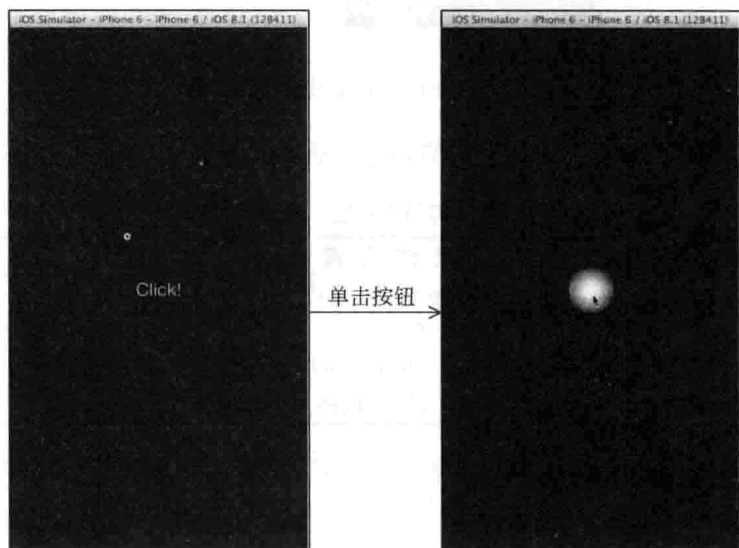


图 1.1 实例 1 运行效果

【实现过程】

当用户单击界面上的按钮时，按钮发光，当按钮没有被单击时，不会发光。具体的实现步骤如下。

- (1) 创建一个项目，命名为“发光按钮”。
- (2) 打开 ViewController.h 文件，编写代码，实现插座变量和动作的声明。程序代码如下：

```
#import <UIKit/UIKit.h>
@interface ViewController : UIViewController{
    IBOutlet UIButton *button; //按钮的插座变量
```

```
}
- (IBAction)click:(id)sender;           //动作 click:的声明
@end
```

(3) 打开 Main.storyboard 文件，对 View Controller 视图控制器的设计界面进行设计，效果如图 1.2 所示。

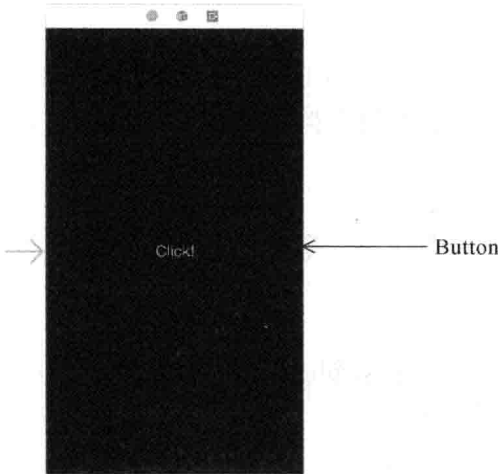


图 1.2 实例 1 View Controller 视图控制器的设计界面效果

需要添加的视图、控件以及对它们的设置如表 1-1 所示。

表 1-1 实例 1 视图、控件设置

视图、控件	属 性 设 置	其 他
设计界面	Background: 黑色	
Button	Title: Click! Font: System 20.0 Text Color: 浅橘黄色	与动作 click:方法关联

(4) 打开 ViewController.m 文件，编写代码，实现单击按钮时，按钮发光。程序代码如下：

```
- (IBAction)click:(id)sender {
    [button setShowsTouchWhenHighlighted:YES];           //发光
}
```

【代码解析】

本实例关键功能是单击按钮时按钮发光。下面就是这个知识点的详细讲解。

要单击按钮，使按钮发光，可以使用 UIButton 的 setShowsTouchWhenHighlighted 属性。其语法形式如下：

```
@property(nonatomic) BOOL showaTouchWhenHighlighted;
```

在代码中，使用了 setShowsTouchWhenHighlighted 属性对按钮在单击时是否发光进行了设置，代码如下：

```
[button setShowsTouchWhenHighlighted:YES];
```

此代码实现的功能，就是在单击按钮时，使按钮发光。如果不想设置在单击按钮时，使按钮发光，开发者可以不写发光的代码，因为它默认就是不发光的，或者将 YES 改为 NO。程序代码如下：

```
[button setShowsTouchWhenHighlighted:NO];
```

实例2 弹出式按钮

【实例描述】

在 iPhone 或者 iPad 上，屏幕的空间大小非常有限，为了腾出更多的空间，开发者一般都会采用一种弹出式按钮。本实例即实现了此功能。当单击屏幕上的按钮后，就会以此按钮为中心弹出三个按钮。单击弹出的三个按钮中的任意一个按钮，就会弹出相应的警告视图；再次单击中心按钮之后，就会将这三个按钮隐藏。运行效果如图 1.3 所示。

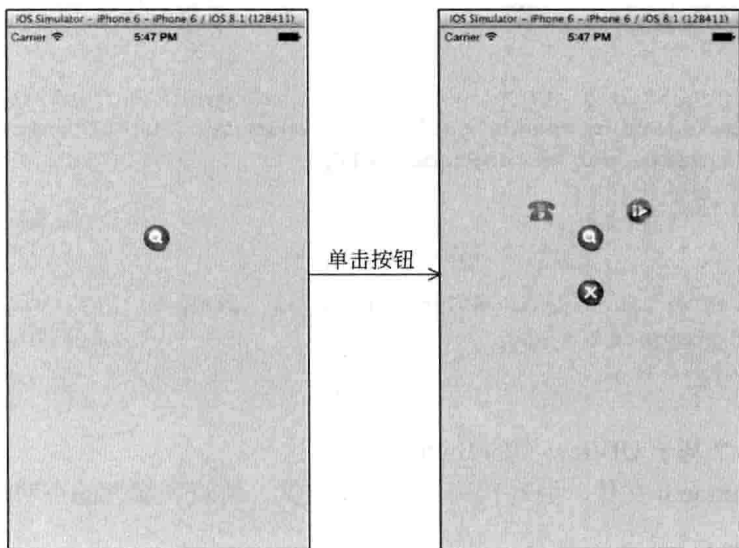


图 1.3 实例2 运行效果

【实现过程】

当用户单击界面的按钮时，会在此按钮的周围弹出三个按钮。具体的实现步骤如下。

- (1) 创建一个项目，命名为“弹出式按钮”。
- (2) 添加图片 button1.png、button2.png、button3.png 和 button4.png 到创建项目的 Supporting Files 文件夹中。
- (3) 创建一个基于 UIButton 类的 but 类。
- (4) 打开 but.h 文件，编写代码，实现协议和属性的声明。程序代码如下：

```
#import <UIKit/UIKit.h>
@class but;
//协议
@protocol butDelegate <NSObject>
- (void)subButtonPress:(but*)button; //单击按钮
```

```

@end
@interface but : UIButton
@property (nonatomic, weak) id<butDelegate> delegate;    //属性的声明
@end

```

(5) 打开 but.m 文件，编写代码，通过触摸实现单击的功能。程序代码如下：

```

#import "but.h"
@implementation but
//初始化
- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self) {
        // Initialization code
    }
    return self;
}
//触摸开始
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event{
    self.highlighted = YES;    //以高亮状态显示
}
//触摸结束
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event{
    if ([_delegate respondsToSelector:@selector(subButtonPress:)]) {
        [_delegate subButtonPress:self];    //调用 subButtonPress:方法
    }
    self.highlighted = NO;    //不以高亮状态显示
}
//触摸取消
- (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event{
    self.highlighted = NO;    //不以高亮状态显示
}
@end

```

(6) 创建一个基于 UIView 类的 Button 类。

(7) 打开 Button.h 文件，编写代码，实现宏定义、协议和属性的声明等功能。程序代码如下：

```

#import <UIKit/UIKit.h>
#import "but.h"    //头文件
//宏定义
#define kDCPathButtonParentView self.parentView
#define kDCPathButtonCurrentFrameWidth kDCPathButtonParentView.frame.size.width
#define kDCPathButtonCurrentFrameHeight kDCPathButtonParentView.frame.size.height
#define kDCCovertAngelToRadian(x) ((x)*M_PI)/180
//枚举类型
typedef enum{
    kDCPathButtonRotationNormal = 0,
    kDCPathButtonRotationReverse,
}DCPathButtonRotationOrientation;
//协议的声明
@protocol ButtonDelegate <NSObject>
@optional
//单击按钮的动作
- (void)button_0_action;

```



```

- (void)button_1_action;
- (void)button_2_action;
- (void)button_3_action;
- (void)button_4_action;
- (void)button_5_action;
@end
@interface Button : UIView<butDelegate>{
    CGPoint kDCPathButtonSubButtonBirthLocation;
    CGPoint kDCPathButtonSubButtonTag_0_AppearLocation;
    CGPoint kDCPathButtonSubButtonTag_1_AppearLocation;
    CGPoint kDCPathButtonSubButtonTag_2_AppearLocation;
    CGPoint kDCPathButtonSubButtonTag_3_AppearLocation;
    CGPoint kDCPathButtonSubButtonTag_4_AppearLocation;
    CGPoint kDCPathButtonSubButtonTag_5_AppearLocation;
    CGPoint kDCPathButtonSubButtonFinalLocation;
}
//方法的声明
- (id)initDCPathButtonWithSubButtons:(NSInteger)buttonCount
    totalRadius:(CGFloat)totalRadius
    centerRadius:(NSInteger)centerRadius
    subRadius:(CGFloat)subRadius
    centerImage:(NSString *)centerImageName
    centerBackground:(NSString *)centerBackgroundName
    subImages:(void (^)(Button *))imageBlock
    subImageBackground:(NSString *)subImageBackgroundName
    inLocationX:(CGFloat)xAxis
    locationY:(CGFloat)yAxis
    toParentView:(UIView *)parentView;

//属性的声明
@property (nonatomic, weak) id<ButtonDelegate> delegate;
@property (nonatomic, getter = isExpanded) BOOL expanded;
.....
- (void)subButtonImage:(NSString *)imageName withTag:(NSInteger)tag;
@end

```

(8) 打开 Button.m 文件, 编写代码。实现弹出式按钮的动作和位置等功能。使用的方法如表 1-2 所示。

表 1-2 Button.m 文件方法总结

方 法	功 能
initDCPathButtonWithSubButtons:totalRadius:centerRadius:subRadius:centerImage:centerBackground:subImages: subImageBackground: inLocationX: locationY: toParentView:	按钮的初始化
configureCenterButton:image:backgroundImage:	对中心按钮进行配置
centerButtonPress	按下中心按钮的动作
configureTheButtons:	对其他按钮的配置
subButtonImage:withTag:	设置按钮的图像
isExpanded	获取按钮是否展开
button:appearAt:withDalay:duration:	按钮出现在指定的位置上的动画
button:shrinkAt:offsetAxisXoffsetAxisY:withDelay:rotateDirection: animationDuration:	按钮消失的动画
matchRotationOrientation:	匹配旋转的方向
offsetAxisY: withAngel:	获取 Y 的偏移轴
subButtonPress:	实现按下其他的按钮

这里需要讲解几个重要的方法（其他方法请读者参考源代码）。initDCPathButtonWithSubButtons:totalRadius:centerRadius:subRadius:centerImage:centerBackground:subImages:subImageBackground:inLocationX: locationY:toParentView:方法，实现按钮的初始化。程序代码如下：

```
- (id)initDCPathButtonWithSubButtons:(NSInteger)buttonCount totalRadius:
(CGFloat)totalRadius centerRadius:(NSInteger)centerRadius subRadius:
(CGFloat)subRadius centerImage:(NSString *)centerImageName centerBack
ground:(NSString *)centerBackgroundName subImages:(void (^)(Button
*))imageBlock subImageBackground:(NSString *)subImageBackgroundName
inLocationX:(CGFloat)xAxis locationY:(CGFloat)yAxis toParentView:(UIView
*)parentView{
    //判断 parentView 是否为空
    parentView == nil? (self.parentView = parentView):(self.parentView
    = parentView);
    xAxis == 0? (self.centerLocationAxisX = kDCPathButtonCurrentFrameWidth/2) :
    (self.centerLocationAxisX = xAxis); //判断 xAxis 是否为 0
    yAxis == 0? (self.centerLocationAxisY = kDCPathButtonCurrentFrameHeight/2) :
    (self.centerLocationAxisY = yAxis); //判断 yAxis 是否为 0
    self.buttonCount = buttonCount; //按钮的个数
    self.totalRadius = totalRadius;
    self.subRadius = subRadius;
    _expanded = NO; //展开设置为 NO
    kDCPathButtonSubButtonBirthLocation = CGPointMake(-kDCPathButton
    Current FrameWidth/2,
    -kDCPathButtonCurrentFrameHeight/2);
    kDCPathButtonSubButtonFinalLocation = CGPointMake(self.centerLocationAxisX,
    self.centerLocationAxisY);
    if (self = [super initWithFrame:self.parentView.bounds]) {
        [self configureCenterButton:centerRadius image:centerImageName backgroundImage:
        centerBackgroundName]; //配置中心按钮
        [self configureTheButtons:buttonCount];
        imageBlock(self);
        [self.parentView addSubview:self]; //添加视图
    }
    return self;
}
```

要实现单击中心按钮后，展开其他按钮或者关闭其他按钮，需要添加：configureCenterButton:image:backgroundImage:、centerButtonPress、subButtonImage:withTag:、isExpanded、button:appearAt:withDelay:duration:、button:shrinkAt:offsetAxisXoffsetAxisY:withDelay:rotateDirection:animationDuration: 和 matchRotationOrientation: 方法。其中 configureCenterButton: image: backgroundImage:方法，对中心按钮进行配置。程序代码如下：

```
- (void)configureCenterButton:(CGFloat)centerRadius image:(NSString *)
imageName backgroundImage:
(NSString *)backgroundImageName{
    self.centerButton = [[UIButton alloc] init];
    self.centerButton.frame = CGRectMake(0, 0, centerRadius * 2, centerRadius * 2);
    //中心按钮框架

    //中心按钮的中心点
    self.centerButton.center = CGPointMake(self.centerLocationAxisX,
    self.centerLocationAxisY);
    //判断 imageName 是否为空
    if (imageName == nil) {
        //如果为空
    }
```