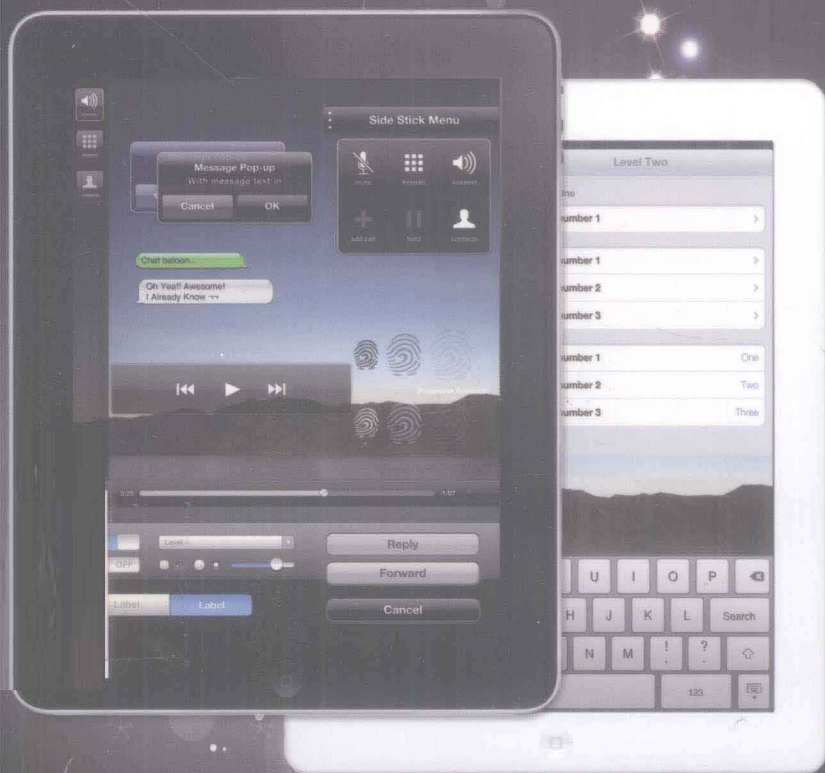


iPad 应用开发指南



- 涵盖iPad/iPad 2应用开发
- 提供基于Objective-C 2.0的示例源代码下载
- 教你从新手到高手，从入门到精通

杨正洪 刘剑雄 夏皇 郭晨 编著



iPad/iPad 2
应用程序开发
高手体验

清华大学出版社

iPad

应用开发指南

杨正洪 刘剑雄 夏 皇 郭 晨 编著

清华大学出版社
北 京

内 容 简 介

iPad/iPad 2 与 iPhone 虽然都使用同一个 iOS 操作系统, 开发工具 (Xcode) 也相同, 但是编程不尽相同。比如针对 iPhone 的开发, 一般都使用工具栏控制器和导航控制器; 而 iPad/Pad 2 开发, 则较多使用拆分视图控制器。所以, 本书是一本专门为 iPad/iPad 2 开发编写的技术书籍。

全书共 9 章, 主要介绍了如何使用拆分视图控制器、多触摸技术、UIBezierPath、模态视图、Popover (浮动菜单)、CoreText 等设计和开发 iPad 应用程序。本书还探讨了如何针对 iPad 开发视频应用程序, 如何完成文件操作, 如何定制键盘, 如何同其他应用交换文件等。此外, 该书还讨论了如何将已有的 iPhone 应用程序升级为 iPad 应用的技巧。

本书适合所有对 iPhone 和 iPad/iPad 2 应用开发感兴趣的软件开发人员、测试人员和系统架构师阅读。

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

iPad 应用开发指南 / 杨正洪等编著. —北京: 清华大学出版社, 2011.7

ISBN 978-302-25755-4

I. ①i… II. ①杨… III. ①便携式计算机—基本知识 IV. ①TP368.32

中国版本图书馆 CIP 数据核字 (2011) 第 098512 号

责任编辑: 夏非彼 夏毓彦

责任校对: 闫秀华

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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: 北京密云胶印厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 190×260 印 张: 19.25 字 数: 493 千字

版 次: 2011 年 7 月第 1 版 印 次: 2011 年 7 月第 1 次印刷

印 数: 1~4000

定 价: 45.00 元

产品编号: 042796-01

前言

2010 年的 IT 界注定是不平凡的，当整个电脑市场尤其是笔记本电脑市场出现疲软的情况之下，iPad 这个在人们眼中的“神物”横空出世，让这个似乎已经对数码新品失去新鲜感的用户重新复活。这个苹果公司 2010 年最吸引眼球的创新产品一经推出便开始对当前电脑业界进行着大刀阔斧的重新洗牌，随着大半年来市场份额的稳步攀升，越来越多针对 iPad 的专属应用程序走进我们的视线。

在不断猜测和期待中，iPad 2 和 iOS 4.3 在 2011 年 3 月终于浮出水面。总体来说，iOS 4.3 其实没有太多创新式的改变，仅仅对于之前一些功能做了加强处理，其实大部分是针对 iPad 2 新增的特性，例如：Facetime 视频通话和 Personal Hotspot（个人无线热点）功能。iOS 4.3 支持 iPhone 3GS、iPhone 4、iPod Touch 3、iPod Touch 4、iPad 和 iPad 2。本书就是专门针对 iPad 和 iPad 2 的应用开发书籍。

据苹果公司称，App Store 应用商店中针对 iPad 的应用数量已经超过了 6.5 万款，这一点对于苹果平板电脑的成功也起到了极其重要的作用。第三方应用有助于拓展此设备的功能。考虑到苹果产品的应用已经占据了此市场的大量份额，而且与 Android 3.0 等设备相比也保持着巨大差距，据称，基于 Android 3.0 系统设备的应用，目前只有 100 款能够让消费者使用，而 iPad 的应用数量远远超过这一数目，因此，竞争对手的产品要想在此方面赶超 iPad，或者想推翻 iPad 的市场主导地位，那将会异常艰难。

iPhone 和 iPad 虽然都使用同一个 iOS 操作系统，开发工具（Xcode）也相同，但是，编程且不尽相同。比如：针对 iPhone 的开发，一般都使用工具栏控制器和导航控制器；而 iPad 的开发，则较多使用拆分视图控制器。本书共分 9 章，主要介绍了如何使用拆分视图控制器、多触摸技术、UIBezierPath、模态视图、Popover（浮动菜单）、CoreText 等设计和开发 iPad 应用程序，本书还探讨了如何针对 iPad 开发视频应用程序，如何完成文档操作，如何定制键盘，如何同其他应用交换文件等。此外，该书还讨论了如何将已有的 iPhone 应用程序升级为 iPad 应用。本书适合所有 iPad 和 iPad 2 的开发人员阅读。

参加本书编写的还有：谢素婷、李建国、吴寒、郑齐心、郭萍、王晓蓉、杜理渊、薛文、李越、孙延辉、张欢涛、胡钦等。我们要特别感谢中网在线控股有限公司、西安八方公司、优胜教育集团和上海创云科技有限公司的大力帮助，这些公司的员工认真仔细地阅读了本书的初稿并测试了本书的实例程序，给我们提供了很多有益的建议和帮助。苹果美国公司 Wayne Lee、上海阿尔卡特公司的何进勇也阅读了本书的初稿并提出了中肯的建议。北京图格新知技术有限公司和夏非彼老师为本书的出版和编辑做了大量的工作，在此深表谢意。

本书实例代码的下载地址：http://www.yunwenjian.com/iphone4/ipad_sample.zip。

由于编者学识浅陋，见闻不广，必有许多不足之处。编者的电子邮箱是：yangzhenghong@yahoo.com。欢迎读者来信指正或探讨 iPad/iPad 2 的开发问题。谢谢。

杨正洪

2011 年 4 月于 San Jose

目录

C O N T E N T S

第 1 章 从 iPhone 到 iPad

1.1	iPhone 应用程序	2
1.2	开发模式	3
1.2.1	MVC 模式	4
1.2.2	Target-Action (目标-操作) 模式	5
1.2.3	Delegate (委托) 模式	6
1.2.4	导航控制器和工具栏控制器	7
1.3	表视图	7
1.4	西安旅游应用实例	10
1.4.1	西安旅游项目中的 M	12
1.4.2	西安旅游项目中的 V 和 C	16
1.4.3	西安旅游项目中的表视图和委托方法	20
1.4.4	西安旅游项目中的 Modal 视图控制器	26
1.4.5	西安旅游项目中的网页视图 (内嵌浏览器)	30
1.5	iPhone 应用的兼容模式	33
1.6	通用 (Universal) 应用	34
1.7	应用图标和启动界面 (splash screen)	37

第 2 章 第一个 iPad 应用和拆分视图 (Split View)

2.1	第一个 iPad 应用实例	41
2.2	UISplitViewController (拆分视图控制器)	47
2.2.1	拆分视图例子	47
2.2.2	理解拆分视图代码	51

2.3 团购应用实例	60
2.4 从导航控制器迁移到拆分视图.....	72
2.4.1 基于导航控制器的 iPhone 应用	72
2.4.2 转化为 iPad 应用.....	79

第3章 触摸屏操作的编程

3.1 手势 (gesture)	86
3.2 轻击 (tap) 手势编程.....	87
3.3 多次触摸和响应者链	91
3.4 轻扫 (swipe) 手势编程.....	95
3.5 离散和连续手势	99
3.6 创建自定义手势	101
3.7 添加声音	107
3.8 手势识别的优先顺序	110
3.9 长按手势	112

第4章 在 iPad 上画图 and 文字编辑器

4.1 创建画板和画图工具	120
4.2 画图工具协议和委托类.....	127
4.3 编写画图工具	130
4.4 Singleton (单例)	134
4.5 视图控制器代码	135
4.6 把图保存为 PDF 文件和发送邮件.....	139
4.7 CoreText 和文本框.....	144
4.8 两种画图 API 的比较	159
4.8.1 基于 C 的画图.....	160
4.8.2 基于 UIBezierPath 画图	164

第5章 Popover 和 Modal (模态) 视图

5.1 基于表视图的 Popover 实例	168
5.1.1 Flexible Space 的妙用	168
5.1.2 关闭 Popover	170
5.1.3 选择不同字体的 Popover 类	172
5.1.4 使用 Popover 类	176
5.2 基于样本数据和滑动条的 Popover 实例	180
5.3 手势和 Popover 的结合编程	189
5.4 拆分视图和浮动菜单	193
5.5 Modal 视图	193
5.6 Modal 还是 Popover	200

第6章 iPad 上的视频播放和管理

6.1 基本视频编程	202
6.1.1 MPMoviePlayerController	202
6.1.2 播放视频实例	203
6.1.3 视频通知	208
6.1.4 播放列表和多视频播放	208
6.2 流播放	215
6.2.1 两种网络播放模式	215
6.2.2 HLS 原理	215
6.2.3 流播放实例	217
6.3 在视频上叠加一些内容	219

第7章 文件操作

7.1 NSKeyedArchiver 和 NSKeyedUnarchiver NSCoder	229
7.2 读和保存文件操作	231
7.2.1 文件管理类	231
7.2.2 拆分视图	235

7.3 新建和删除文件	246
7.4 重命名文件和 Modal 视图	255
7.5 回滚修改	261

第 8 章 定制自己的键盘

8.1 在键盘上增加一个“取消”按钮	264
8.2 定制快捷键	265
8.3 在标准键盘上添加自己的键	268
8.4 键盘通知	270

第 9 章 同外部应用交换文件

9.1 注册支持的文件类型	275
9.2 传递文件给另一个应用	277
9.3 UIDocumentInteractionController	281
9.4 从另一个应用上接收文件	284
9.5 与 iTunes 同步文件	287

附 录

附录 A 安装 iPad 开发环境	289
附录 B 更多学习材料	294
附录 C iOS 4.3 特性	294
附录 D iPad 2 特性	295
附录 E iOS 的多任务功能	296

第 1 章

从 iPhone 到 iPad

从本章节你可以学习到：

- ❖ iPhone 应用程序
- ❖ 开发模式
- ❖ 表视图
- ❖ 西安旅游应用实例
- ❖ iPhone 应用的兼容模式
- ❖ 通用（Universal）应用
- ❖ 应用图标和启动界面（splash screen）

有人说, iPad 就是一台大个的 iTouch 4 或 iPhone 4, 因为他们使用相同的操作系统 iOS, 很多 iPhone 4 和 iPad 的 API 都是相同的。从最终用户的角度而言, 其实不然。iPad 更像一个 10 寸的笔记本电脑(它使用触摸屏, 而不是物理键盘), 整个屏幕是 1024×768 像素点。你可以在 iPad 上轻松地查看电子邮件和网页, 你看到的是完整的网页。在 iPhone 上, 整个屏幕是 320×480 像素点, 你只能看到部分网页, 要看到整个网页, 需要用手指移动网页。iPhone 和 iPad 屏幕大小的不同, 也使得 iPad 应用程序非常的不同。我们将通过一个实际的例子来比较。

要提醒读者的是, 本书假定读者熟悉 iPhone 应用程序的开发(包括 Xcode 和界面创建器的使用)。如果读者需要这部分的知识, 可以参考清华大学出版社出版的《iPhone 4 应用开发入门与实战》一书。iPhone 和 iPad 的编程语言是 Objective-C。本书也假定读者熟悉 Objective-C 语言。如果读者需要 Objective-C 这部分知识, 可以参考清华大学出版社出版的《Objective-C 程序设计》一书。在开发 iPhone/iPad 应用时, 你需要安装开发工具。在本书的附录 A 中, 描述了怎么安装最新的 Xcode 和 iOS 4.2 SDK。

1.1 iPhone 应用程序

我们以一个 iPhone 例子来回顾 iPhone 应用程序的开发步骤和体系结构。我们的 iPhone 应用例子是一个西安旅游的应用程序, 如图 1-1 所示。在启动西安旅游应用后, 系统先显示一个图, 如图 1-2 所示。在背后, 系统正在装载一些关于西安的图和文字数据。然后, 显示西安旅游应用的各个子系统, 比如: 景点图片、地方小吃等, 如图 1-3 所示。另外, 因为这个应用是免费的, 所以, 应用下方显示了一些动态的广告信息。当用户点击“景点图片”后, 将显示一个景点列表(有景点小图和文字, 如图 1-4 所示)。当用户选择一个景点, 应用就会显示该景点的详细信息(包括图片、介绍和路线等, 如图 1-5 和图 1-6 所示)。当用户点击界面下方的广告时, 应用将会打开一个广告网页。

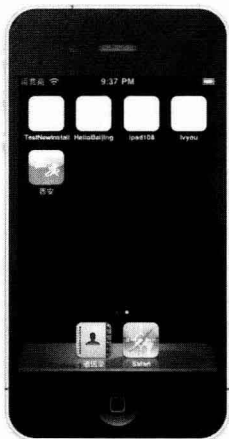


图 1-1 西安旅游应用

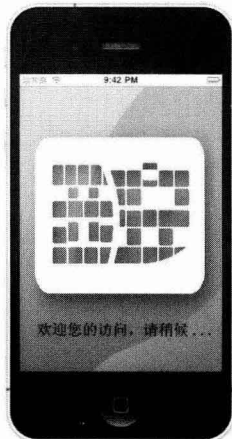


图 1-2 启动界面



图 1-3 各个功能



图 1-4 景点列表

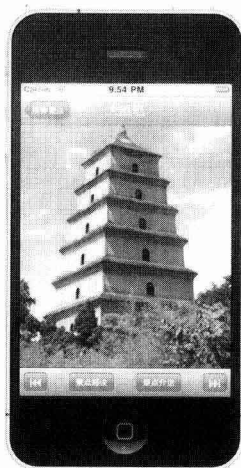


图 1-5 景点图



图 1-6 最后一个景点图

下面我们就以这个西安旅游应用为例，来阐述 iPhone 应用开发的一些要点。

1.2 开发模式

iPhone/iPad 开发架构是基于面向对象的技术。开发人员应该采用 MVC (Model、View 和 Controller) 模式。该模式将一个应用程序中的对象按照其扮演的角色分成 M、V 和 C 三种。在开发 iPhone/iPad 应用程序时，开发人员也会同时采用另外三种模式：Target-Action (目标-操作) 模式、Delegation (委托) 模式和 Notification (通知) 模式。对于通知模式，我们在第 5 章通过实例来阐述。

1.2.1 MVC 模式

我们先解释 MVC 模式。上一节中的西安旅游手机应用就包含了 MVC 三部分。

- M: 模型 (Model)。在 iPhone 程序中, 模型就是一组用 Objective-C 定义的类, 用来表示应用程序数据。比如: 西安的旅游图片、景点文字信息、路线、地方小吃等都是 M 提供的数据。我们用一个类来表示旅游图片这一项, 用类的属性来记录它包含的信息, 比如, 哪个风景区的图片。
- V: 显示界面, 又称视图 (View)。在 iPhone 程序里, 一般使用 Interface Builder 构建视图, 也可以用代码创建和修改。视图通常就是在手机上看到并与之交互的界面。比如: 图 1-4 显示了各个风景区的信息, 包括图和简短文字。从而用户可以选择某个风景区 (比如: 大雁塔)。
- C: 控制器 (Controller), 即连接 M 和 V, 接受用户的输入并调用模型和视图去完成用户的需求。当用户在图 1-4 上选择“大雁塔”时, 控制器就以“大雁塔”为输入参数来调用模型, 在获得模型的返回结果 (即关于大雁塔的照片和介绍信息) 后, 送给另一个 V 来显示 (如图 1-5 所示)。概括来说, 当 M 发生更改时, 更新 V; 当用户操作 V 时, 查询或更新 M。

图 1-7 显示了 MVC 之间的关系。我们看到, V 和 M 不会直接发生关系, 完全通过 C 来完成。

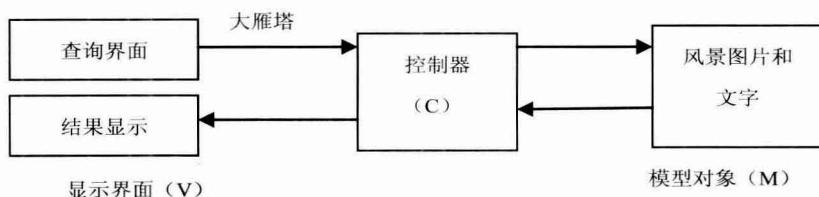


图 1-7 MVC 之间的关系

我们建议开发人员首先使用 MVC 模型来设计 iPhone 程序。刚开始使用这一模型时, 开发人员可能感觉到工作量的较大 (比如, 需要单独的类来存放照片数据), 但是, 这便于后续代码的扩展和维护。一个使用 MVC 编写的 iPhone 应用程序可以轻松地转换为 iPad 应用程序。在转换过程中, 主要的修改在于 V, 对于 M, 可以完全重用; 对于 C, 也可以部分重用。另外, MVC 结构也使应用程序结构清晰, 便于维护。

举一个实际的例子来阐述 MVC 架构的重要性。在西安旅游应用的前一个版本里, 一个控制器代码中包含了所有风景区数据的获取和操作。然后, 在同一个代码中, 创建了一个 UINavigationController 对象, 并以导航的方式来展现风景区数据。当把这个应用转到 iPad 上去时, 发现 iPad 使用 UISplitViewController 来显示数据, 而不是 UINavigationController。这时, 工程师需要从控制器代码中把所有 UINavigationController 的相关代码替换为 UISplitViewController 代码, 同时, 也会发现把数据单独放在一个或多个 M 类上的方便性。本节中的西安旅游应用, 就是按照 MVC 架构改写后的应用, 如图 1-8 所示, Scene 类表示一个

风景区的数据模型类，Scenes 类表示所有风景区的数据模型，SceneInfo.xib 等 xib 文件是视图层，firstViewController、SecondController 和 DetailController 是控制器。



图 1-8 西安旅游的 MVC 结构

1.2.2 Target-Action（目标-操作）模式

大多数 UI 对象都有事件（Event），图 1-9 所示的景点图片按钮就有很多事件，如：Touch Up Inside 事件（点击按钮事件）。你可以关联事件到某一个视图控制器类上的某一个方法，从而，当这个事件触发时，该方法就被调用。在 iPhone/iPad 开发上，把这个模式叫做设置 Target-Action（目标-操作）。这是从最终用户的角度来描述的，如图 1-9 所示，当用户对 iPhone 窗口上的景点图片按钮对象做了某些动作时（如：点击，其实是触发一个事件），系统就调用该 UI 对象的 Target（目标）所指定的控制类（File's Owner，即 FirstViewController）和 Action 所指定的操作（即 buttonPressed 方法）。所有上述控件对象同控制器类的关联（即设置目标-操作），都可以在 Interface Builder 上面完成，而无需编写代码。我们会在 1.4 节中有更多说明。

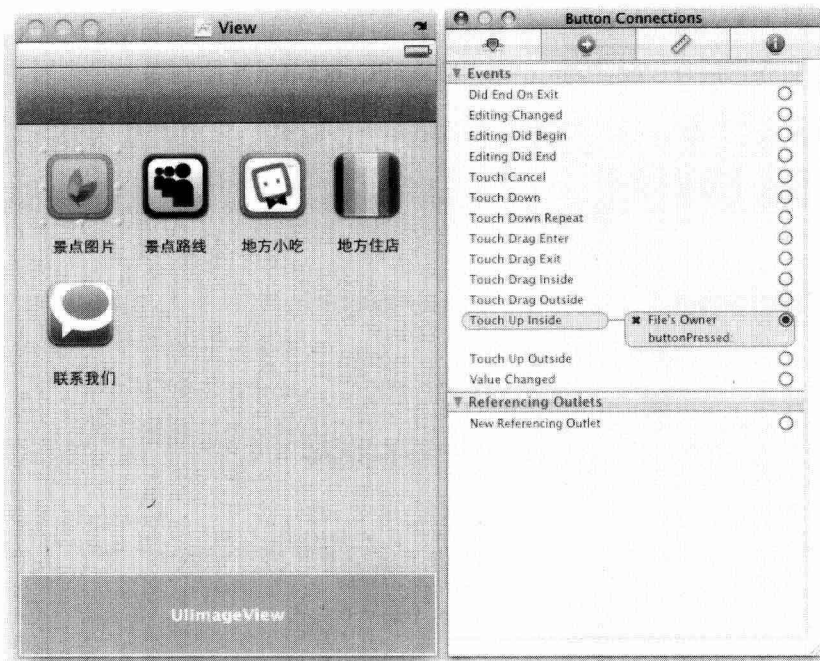


图 1-9 目标-操作

1.2.3 Delegate（委托）模式

委托模式允许一个对象去修改另一个对象的行为，或者接收另一个对象的消息。比如，在一个视图上，你放置了一个 UIWebView 控件，用来显示网页内容。装载网页的内容本身可能需要一些时间，所以，你希望在装载结束后有进一步的处理。这时，你可以设置一个符合 UIWebView 委托协议的视图控制器（比如：FirstIpviewController）：

```
@interface FirstIpviewController : UIViewController <UIWebViewDelegate>
{
```

myWebView 是 FirstIpviewController 上的 IBOutlet 属性，连接到视图上的 UIWebView 控件。下面的语句设置了 UIWebView 的委托就是 self，即 FirstIpviewController。

```
- (void)viewDidLoad {
    self.myWebView.delegate = self;
    [super viewDidLoad];
}
```

既然 FirstIpviewController 是 UIWebView 的委托，那么，FirstIpviewController 就可以处理一些 UIWebView 的相关事件了。webViewDidFinishLoad 就是其中一个事件，表示装载网页内容结束。

```

- (void)webViewDidFinishLoad:(UIWebView *)webView {
    //进一步处理，比如：弹出一个欢迎窗口，或者广告
}

```

在上面的例子中，通过委托，完成了一些异步调用的功能：当 `UIWebView` 装载结束后，调用视图控制器上的 `webViewDidFinishLoad`。在表视图上，也大量使用委托。我们在下节中讲解。

1.2.4 导航控制器和工具栏控制器

对于 iPhone 程序，一般是由一个视图控制器控制着正在显示的一个视图，换句话说，一个视图控制器总是与一个视图关联。导航控制器和工具栏控制器本身是一个例外，它们自己不显示任何内容，而是组织不同的视图控制器。

1.3 表视图

顾名思义，表视图是以表的形式展现数据。表视图并不显示表中的所有数据，而只是显示当前可见行的数据。图 1-10 使用了表视图显示了西安的部分旅游景点。用户通过滚动屏幕，就可以看到其他的旅游景点。



图 1-10 表视图的应用

西安旅游应用使用基于视图的应用程序模板，用表视图（`UITableView`）一行一行显示景区数据。表视图是显示表数据的视图对象，是 `UITableView` 的一个实例。表中的每个可见行都是一个表单元 `UITableViewCell`。

那么，表视图从哪里知道总共有多少行呢？如图 1-11 所示，`UITableView` 向视图控制器

发一个 `tableView: numberOfRowsInSection:` 消息，询问行数。反过来，控制器向模型对象（`Scenes`）发送消息 `count` 来获得行数。接下来，模型对象响应视图控制器。最后反映到 `UITableView` 上，我们就知道有多少行被显示了。对于一行中的具体信息（如风景区图片、标题和描述信息），也是通过视图控制器从数据模型那里获得的。一行就是一个 `UITableViewCell`。

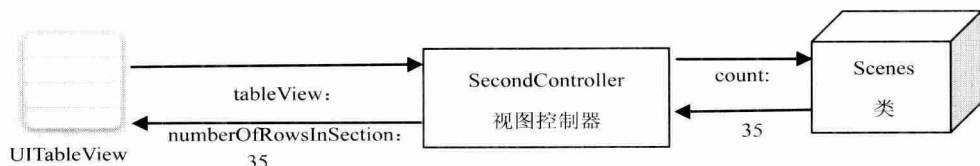


图 1-11 表视图和视图控制器

当用户点击中某一行时（比如：大雁塔，如图 1-12 所示），视图控制器（`secondController`）通过使用导航控制器，在一个新的视图上显示具体照片（这个新视图的控制器是 `DetailController`）。



图 1-12 表视图操作

`UITableViewDataSource` 协议是规定对象从哪里获得数据，而 `UITableViewDelegate` 协议规定在表视图上操作后的具体动作（比如，选择“大雁塔”后）。上述的 `numberOfRowsInSection` 方法就是帮助对象获取行数，是 `UITableViewDataSource` 协议的内容。在我们的例子中，`SecondController` 也实现了 `UITableViewDelegate` 协议的 `tableView: didSelectRowAtIndexPath:` 委托方法（也就是说，`SecondController` 是这个表视图的委托类），如图 1-13 所示，这个方法创建了一个新的视图控制器 `DetailController`，并把要显示的具体风景区信息传递给下一个新视图。