

图书基本信息

书名：<<IOS 7:iPhone/iPad应用开发技术详解>>

13位ISBN编号：9787111440512

10位ISBN编号：711144051X

出版时间：2013-11

出版时间：机械工业出版社

作者：刘一道

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

内容概要

本书由资深Mac/iOS开发工程师基于iOS 7撰写。

内容全面，从Objective-C语法知识、iOS功能特性，到高级开发方法和技巧，几乎涵盖了中初级iOS开发工程师需要掌握的所有技术和知识；实战性强，每个知识点都有辅助理解的小案例，最后还提供了两个综合性应用开发案例。

内容循序渐进，是系统学习iOS应用开发的经典著作。

全书共20章，分4部分。

准备篇（第1~2章）介绍了iOS、Objective-C和Xcode的入门知识，详细描述了iOS 7新特性、Objective-C应用开发环境的搭建，以及iPhone应用开发的完整过程；语法篇（第3~8章）详细讲解Objective-C的语法知识，其中包括类、对象、消息和协议，以及内存管理，重点讲解了Foundation框架常用类的使用方法；基础篇（第9~18章）讲解了iOS应用开发的框架，包含多状态和多任务、视图和视图控制器、事件和通知、音频和视频等，这些内容是本书的重点；实战篇（第19~20章）详细讲解了两个实战案例的完整开发过程，分别是基于iPad的应用程序“精灵小书柜”和基于iPhone的“弹球游戏”，目标是通过两个案例把前面学到知识应用到实践中，帮助读者真正掌握iOS应用开发的过程。

作者简介

刘一道，资深Mac/iOS开发工程师，有近10年软件开发经验，精通iOS、Objective-C、Xcode等与苹果开发相关的各项技术，同时对Android和Windows Phone等移动开发技术也有深入的研究，并积累了丰富的实战经验。

出版了《Objective-C程序设计入门与实践》等著作。

书籍目录

前言

第一部分 准备篇

第1章 初识iOS、Objective-C和Xcode

1.1 认识iOS

1.1.1 iOS的发展历程

1.1.2 iOS的设计和特性

1.1.3 iOS 7的新特性

1.1.4 iOS架构

1.1.5 iOS框架

1.1.6 iOS系统框架的变迁

1.1.7 Mac OS X和iOS平台不同框架的差异性

1.1.8 初步了解iOS开发者工具

1.2 认识Objective-C

1.2.1 发展历程及版本变化

1.2.2 语言性能与分析

1.2.3 框架和代码的关系

1.3 基于非苹果机平台搭建开发环境

1.3.1 前期准备

1.3.2 创建用于安装Mac OS X的VMWare虚拟机

1.3.3 安装Mac OS X

1.3.4 安装Xcode SDK

1.4 小结

第2章 创建你的

第一个iOS应用程序

2.1 应用程序的实现目标

2.2 入门的开始

2.2.1 新建一个Xcode项目

2.2.2 在模拟器中查看应用程序的效果

2.3 启动一个应用程序

2.3.1 探究main.m源文件

2.3.2 分析属性列表文件

2.3.3 查看串联图

2.4 检查视图控制器及其视图

2.4.1 如何使用检查器

2.4.2 更改视图的背景颜色

2.5 对视图进行配置和管理

2.5.1 新增用户界面元素

2.5.2 为按钮增添一个动作

2.5.3 为文本框和标签创建插座

2.5.4 打开Connections检查器验证连接

2.5.5 对文本框进行委托处理

2.5.6 让应用程序具有辅助功能

2.6 使用视图控制器完成应用程序

2.6.1 给用户的名称添加属性

2.6.2 实现changeGreeting : 方法

- 2.6.3 把视图控制器作为输入文本框的委托
- 2.7 测试应用程序
 - 2.7.1 排查和检测代码
 - 2.7.2 程序代码清单
- 2.8 小结
- 第二部分 语法篇
- 第3章 Objective-C--构建iOS应用程序的基石
 - 3.1 探窥Objective-C语言
 - 3.1.1 面向对象语言Objective-C是C语言的超集
 - 3.1.2 类和对象
 - 3.1.3 方法和发消息
 - 3.1.4 属性和存取方法
 - 3.1.5 块
 - 3.1.6 协议和类别
 - 3.1.7 类型和编码策略
 - 3.1.8 import语句
 - 3.2 Objective-C 2.0新增特性
 - 3.2.1 关联引用
 - 3.2.2 快速枚举
 - 3.2.3 选择器
 - 3.2.4 静态类型的使用
 - 3.3 进一步认识块
 - 3.3.1 块可以带参数和返回值
 - 3.3.2 块可以捕获封闭范围内的值
 - 3.3.3 在块内捕获变量值的变化
 - 3.3.4 块可以作为函数或者方法的参数
 - 3.3.5 用类型定义可以简化块语法
 - 3.3.6 使用属性可跟踪块
 - 3.4 小结
- 第4章 类--构建应用程序的类型对象原型
 - 4.1 认识根类
 - 4.1.1 NSObject简介
 - 4.1.2 根类和协议
 - 4.1.3 根类方法
 - 4.1.4 根类接口规范
 - 4.1.5 根类实例方法和类方法
 - 4.2 如何设计类
 - 4.2.1 设计接口
 - 4.2.2 设计实现
 - 4.2.3 如何使用类名
 - 4.2.4 如何比较类
 - 4.3 类的类型
 - 4.3.1 如何指定静态类型
 - 4.3.2 类型的自查处理机制
 - 4.4 变量
 - 4.4.1 局部变量
 - 4.4.2 全局变量

- 4.4.3 实例变量
 - 4.4.4 静态变量
 - 4.4.5 变量的存储类别
 - 4.5 属性
 - 4.5.1 属性的声明
 - 4.5.2 属性的实现
 - 4.5.3 属性类型和相关函数
 - 4.5.4 属性的类型编码
 - 4.5.5 属性重声明
 - 4.5.6 修改父类的属性
 - 4.5.7 新旧版本属性运行时的区别
 - 4.6 方法
 - 4.6.1 方法通用格式
 - 4.6.2 方法的调用
 - 4.7 继承
 - 4.7.1 类的继承关系
 - 4.7.2 继承父类的实例变量
 - 4.7.3 继承父类的方法
 - 4.7.4 哪些类需要被继承
 - 4.7.5 对象的合成
 - 4.8 方法重写
 - 4.8.1 方法重写的规则
 - 4.8.2 方法重写的使用
 - 4.9 方法重载
 - 4.9.1 方法重载的规则
 - 4.9.2 方法重载的使用
 - 4.9.3 调用还是重载
 - 4.9.4 重载的类型
 - 4.10 类的扩展
 - 4.10.1 类别的用法
 - 4.10.2 延伸的用法
 - 4.11 异常处理
 - 4.11.1 如何启用异常处理
 - 4.11.2 如何捕捉不同类型的异常
 - 4.11.3 如何抛出异常
 - 4.12 小结
- 第5章 对象--构建应用程序的重要“活体”
- 5.1 理解对象
 - 5.1.1 对象和根类的关系
 - 5.1.2 对象如何构建程序
 - 5.1.3 对象动态类型
 - 5.1.4 对象的生命周期
 - 5.2 创建对象
 - 5.2.1 对象分配处理的机制
 - 5.2.2 对象初始化方法的原型init
 - 5.2.3 初始化方法的返回值
 - 5.2.4 init方法的实现

- 5.2.5 多个初始化方法和指定初始化方法
- 5.2.6 使用dealloc方法
- 5.2.7 类工厂方法
- 5.3 对象的所有权
 - 5.3.1 对象所有权策略有哪些
 - 5.3.2 保留计数的处理机制
 - 5.3.3 自动释放所有权
 - 5.3.4 共享对象的有效性
 - 5.3.5 如何获取所有权
- 5.4 回收对象
 - 5.4.1 dealloc方法的实现
 - 5.4.2 通过引用返回的对象
 - 5.4.3 保留循环的处理机制
 - 5.4.4 对象的弱引用
 - 5.4.5 资源的有效管理
- 5.5 应用对象
 - 5.5.1 验证对象的功能
 - 5.5.2 比较对象
 - 5.5.3 复制对象
- 5.6 小结
- 第6章 消息和协议--对象之间的通信方式
 - 6.1 认识消息
 - 6.1.1 消息的基本语法
 - 6.1.2 消息处理机制
 - 6.1.3 获得方法地址
 - 6.1.4 使用隐藏的参数
 - 6.2 消息发送
 - 6.2.1 向nil发送消息
 - 6.2.2 向自己发送消息
 - 6.2.3 通过发送消息调用父类方法
 - 6.3 消息转发
 - 6.3.1 消息转发处理机制
 - 6.3.2 与多重继承关系
 - 6.3.3 与类继承关系
 - 6.3.4 与消息代理对象关系
 - 6.4 认识协议
 - 6.4.1 协议的分类
 - 6.4.2 协议对象
 - 6.4.3 预定义接口
 - 6.4.4 如何使用预定义方法
 - 6.4.5 为匿名对象声明方法
 - 6.5 应用协议
 - 6.5.1 如何采用一个协议
 - 6.5.2 如何服从一个协议
 - 6.5.3 协议类型校验处理机制
 - 6.5.4 如何实现协议嵌套协议
 - 6.5.5 如何引用其他协议

6.6 小结

第7章 Foundation框架--提供基本的系统服务

7.1 认识Foundation框架

7.1.1 Foundation类层次结构

7.1.2 与Core Foundation框架的区别

7.1.3 如何引用及查询Foundation框架信息

7.1.4 对象的可变性和不变性

7.2 创建和使用值对象

7.2.1 创建值对象

7.2.2 字符串和NSString字面常量

7.2.3 NSNumber字面常量

7.2.4 日期和时间

7.3 创建和使用集

7.3.1 将对象按某种顺序储存在数组中

7.3.2 将键值对储存在字典中

7.3.3 将无序对象储存在集合中

7.4 Foundation框架常用类的使用方法详解

7.4.1 数字类型

7.4.2 字符串类型

7.4.3 数组类型

7.4.4 字典类型

7.4.5 日期类型

7.5 小结

第8章 内存管理--应用程序高效运行的基础

8.1 内存管理基础知识

8.1.1 Cocoa的引用计数机制

8.1.2 内存管理规则

8.1.3 内存管理应用实例

8.1.4 内存管理混乱的原因及解决方法

8.2 存取方法

8.2.1 声明存取方法

8.2.2 实现存取方法

8.2.3 存取方法的使用

8.2.4 实现重置方法

8.2.5 应用存取方法常见错误

8.3 自动释放池

8.3.1 自动释放池工作原理

8.3.2 非Application Kit程序中的自动释放池

8.3.3 自动释放池和线程

8.3.4 作用域和嵌套自动释放池的关系

8.3.5 所有权策略

8.3.6 如何实现垃圾回收

8.4 内存中复制的处理机制

8.4.1 深复制和浅复制

8.4.2 独立副本

8.4.3 使用alloc和init方式复制

8.4.4 使用NSCopyObject函数复制

- 8.4.5 可变对象和不可变对象的复制
- 8.4.6 值对象和复制
- 8.5 Nib对象的内存管理
 - 8.5.1 插座对象实现的内存管理机制
 - 8.5.2 Nib文件实现的内存管理机制
- 8.6 小结
- 第三部分 基础篇
- 第9章 探究iOS应用程序的核心
 - 9.1 iOS应用程序核心架构
 - 9.1.1 iOS应用程序的生命周期
 - 9.1.2 iOS应用程序的主函数
 - 9.1.3 应用程序的委托
 - 9.1.4 主Nib文件
 - 9.1.5 事件处理周期
 - 9.1.6 应用程序的基本设置模式
 - 9.2 iOS应用程序的核心对象
 - 9.2.1 iOS应用程序的常见对象
 - 9.2.2 数据模型的定义
 - 9.2.3 构建用户界面
 - 9.3 iOS应用程序包
 - 9.3.1 典型的iOS应用程序捆绑包
 - 9.3.2 信息属性列表
 - 9.3.3 程序图标和启动图像
 - 9.3.4 Nib文件
 - 9.4 iOS应用程序关键任务的处理机制
 - 9.4.1 初始化和终止
 - 9.4.2 响应中断
 - 9.4.3 低内存警告
 - 9.5 iOS应用程序的行为定制
 - 9.5.1 以景观模式启动
 - 9.5.2 和其他应用程序的通信
 - 9.5.3 定制的URL模式
 - 9.5.4 应用程序的偏好设置
 - 9.5.5 关闭屏幕锁定
 - 9.6 小结
- 第10章 多状态和多任务--iOS功能日趋增强的表现
 - 10.1 应用程序的状态
 - 10.1.1 状态切换遵循的原则
 - 10.1.2 应用程序的状态及切换路径
 - 10.2 应用程序启动周期
 - 10.2.1 加载进入前台
 - 10.2.2 加载进入后台
 - 10.2.3 主函数
 - 10.2.4 启动处理的机制
 - 10.3 响应中断
 - 10.3.1 基于警告的中断处理机制
 - 10.3.2 中断应激处理

10.3.3 通话时用户界面的调整处理

10.4 前台与后台之间的切换机制

10.4.1 从前台切换到后台的流程

10.4.2 转换后台时应激处理

10.4.3 后台运行时的内存情况

10.4.4 从后台回转到前台的流程

10.4.5 在唤醒时处理排队的通知

10.4.6 应用程序的终止条件

10.5 主运行循环

10.5.1 主运行循环的运行机制

10.5.2 iOS应用程序中的事件

10.6 后台执行和多任务处理

10.6.1 判断设备系统多任务是否可用

10.6.2 在后台执行有限长度的任务

10.6.3 调度本地通知的递送

10.6.4 允许在后台运行的任务

10.6.5 选择退出后台的执行

10.7 小结

第11章 视图--iOS应用程序交互的基础

11.1 窗口、视图、视图控制器

11.1.1 三者之间的关系

11.1.2 窗口

11.1.3 视图

11.1.4 视图控制器

11.2 视图的几何属性特征

11.2.1 视图坐标系统

11.2.2 边框、边界和中心的关系

11.2.3 坐标系统变换

11.3 视图架构处理

11.3.1 视图交互模型

11.3.2 视图渲染架构

11.3.3 改变视图的层

11.3.4 内容模式与比例缩放

11.3.5 自动尺寸调整行为

11.4 视图的创建和管理

11.4.1 创建视图对象

11.4.2 视图的标识和命名

11.4.3 子视图的添加和移除

11.4.4 视图层次中的坐标转换

11.4.5 视图的查询

11.5 在运行时修改视图

11.5.1 实现视图动画

11.5.2 配置动画的参数

11.5.3 配置动画的委托

11.5.4 响应布局的变化

11.5.5 重画视图的内容

11.5.6 隐藏视图

- 11.6 定制视图对象
 - 11.6.1 初始化定制视图
 - 11.6.2 绘制视图内容
 - 11.6.3 响应事件
 - 11.6.4 视图对象的清理
- 11.7 小结
- 第12章 UIKit框架--创建基于触摸的用户界面
 - 12.1 UIKit标准视图的分类
 - 12.2 显示视图
 - 12.2.1 标签
 - 12.2.2 图片视图
 - 12.2.3 进度条视图
 - 12.2.4 等待视图
 - 12.3 控件
 - 12.3.1 按钮
 - 12.3.2 文本框
 - 12.3.3 滑块
 - 12.3.4 切换开关
 - 12.4 导航视图
 - 12.4.1 导航栏
 - 12.4.2 标签栏
 - 12.5 警告视图和动作表单
 - 12.5.1 警告视图
 - 12.5.2 动作表单
 - 12.6 文本和Web视图
 - 12.6.1 文本视图
 - 12.6.2 Web视图
 - 12.7 容器视图
 - 12.7.1 表视图
 - 12.7.2 滚动视图
 - 12.7.3 工具栏
 - 12.8 其他常用类型
 - 12.8.1 分页控件
 - 12.8.2 搜索栏
 - 12.9 小结
- 第13章 视图控制器--视图的幕后操纵者
 - 13.1 视图控制器基础知识
 - 13.1.1 视图控制器的功能
 - 13.1.2 视图控制器的管理机制
 - 13.1.3 视图控制器的分类
 - 13.1.4 视图控制器的内容多种展示方式
 - 13.1.5 多种视图控制器混合应用
 - 13.2 视图控制器的生命周期
 - 13.2.1 视图控制器的初始化
 - 13.2.2 视图的加载和卸载
 - 13.3 标准视图控制器
 - 13.3.1 标准视图控制器的功能及使用方法

- 13.3.2 标准视图控制器的应用
- 13.4 分割视图控制器
 - 13.4.1 分割视图控制器构建原理及功能
 - 13.4.2 分割视图控制器的应用
- 13.5 导航视图控制器
 - 13.5.1 导航视图控制器构建原理及功能
 - 13.5.2 导航视图控制器的应用
- 13.6 选项卡视图控制器
 - 13.6.1 选项卡视图控制器构建原理及功能
 - 13.6.2 选项卡视图控制器的应用
- 13.7 翻页视图控制器
 - 13.7.1 翻页视图控制器构建原理及功能
 - 13.7.2 翻页视图控制器的应用
- 13.8 小结
- 第14章 事件--应用程序的驱动动力
 - 14.1 iOS系统中的事件
 - 14.1.1 事件是如何驱动应用程序的
 - 14.1.2 什么是触摸事件
 - 14.1.3 什么是运动事件
 - 14.1.4 事件和触摸
 - 14.1.5 事件的传递
 - 14.1.6 响应者对象和响应者链
 - 14.1.7 调整事件的传递
 - 14.2 常见手势的处理实例
 - 14.2.1 触摸事件处理方法
 - 14.2.2 单个和多个触碰手势处理方法
 - 14.2.3 检测碰擦手势
 - 14.2.4 处理复杂的多点触摸序列
 - 14.2.5 触摸事件处理技巧
 - 14.3 键盘管理
 - 14.3.1 接收键盘通告
 - 14.3.2 显示键盘
 - 14.3.3 取消键盘
 - 14.3.4 移动键盘下面的内容
 - 14.4 小结
- 第15章 通知--消息的多样化展示
 - 15.1 通知的实现原理
 - 15.1.1 本地通知
 - 15.1.2 推送通知
 - 15.1.3 通知的应用场景
 - 15.2 通知的相关事务
 - 15.2.1 自定义警告声音
 - 15.2.2 创建和调度本地通知
 - 15.2.3 接收远程通知
 - 15.2.4 如何处理通知
 - 15.3 推送通知服务
 - 15.3.1 推送通知和路径

- 15.3.2 服务的反馈和质量
- 15.3.3 推送通知的安全处理机制
- 15.3.4 通知的负载处理
- 15.4 推送通知的配置和开发
 - 15.4.1 沙箱环境和产品环境
 - 15.4.2 配置处理流程
- 15.5 实现推送通知服务的通信功能
 - 15.5.1 普通提供的通信功能
 - 15.5.2 二进制接口和通知的格式
 - 15.5.3 服务的反馈
- 15.6 小结
- 第16章 Core Data框架--管理应用程序的数据模型
 - 16.1 认识Core Data
 - 16.1.1 初窥Core Data特性
 - 16.1.2 Core Data数据管理机制
 - 16.1.3 探究Core Data的本质
 - 16.2 Core Data堆栈配置
 - 16.2.1 托管对象和上下文
 - 16.2.2 读取请求
 - 16.2.3 持久化存储协调者
 - 16.2.4 持久化存储
 - 16.2.5 持久化文档
 - 16.2.6 托管对象模型
 - 16.3 Core Data模块的运作机制及基础类
 - 16.3.1 NSManagedObjectContext类
 - 16.3.2 NSManagedObject类
 - 16.3.3 NSManagedObjectContext类
 - 16.3.4 NSPersistentStoreCoordinator类
 - 16.3.5 NSPersistentDocument类
 - 16.3.6 NSFetchedResultsController类
 - 16.4 Core Data堆栈访问技术
 - 16.4.1 创建一个新的托管对象上下文
 - 16.4.2 读取托管对象模型和实体
 - 16.4.3 添加持久化存储
 - 16.5 托管对象的管理
 - 16.5.1 读取托管对象
 - 16.5.2 读取特定的属性值
 - 16.5.3 创建托管对象
 - 16.5.4 保存托管对象
 - 16.5.5 删除托管对象
 - 16.6 小结
- 第17章 音频和视频--强大的多媒体功能支持
 - 17.1 初识iOS多媒体框架
 - 17.1.1 iOS声音处理工具
 - 17.1.2 Core Audio框架
 - 17.1.3 音频硬件编解码
 - 17.1.4 iOS支持的音频回放和录制格式

- 17.1.5 Core Audio音频会话接口
- 17.1.6 iOS系统支持的音频单元
- 17.2 录制音频
 - 17.2.1 使用AVAudioRecorder类进行录制
 - 17.2.2 用音频队列服务进行录制
 - 17.2.3 音频中断处理
- 17.3 播放音频
 - 17.3.1 通过iPod媒体库访问接口播放媒体项
 - 17.3.2 使用系统声音服务播放短声音及触发振动
 - 17.3.3 通过AVAudioPlayer类轻松播放声音
 - 17.3.4 用音频队列服务播放和控制声音
 - 17.3.5 使用OpenAL播放和定位声音
 - 17.3.6 解析音频流
- 17.4 iPhone/iPad音频的最佳实践
 - 17.4.1 操作音频的贴士
 - 17.4.2 iOS中偏好的音频格式
- 17.5 iOS中的视频
 - 17.5.1 录制视频
 - 17.5.2 播放视频
- 17.6 小结
- 第18章 设备特性--支持多种功能的应用
 - 18.1 识别可用的硬件特性
 - 18.2 External Accessory框架的管道机制
 - 18.2.1 声明应用程序支持的协议
 - 18.2.2 在运行时连接配件
 - 18.2.3 处理和流相关的数据
 - 18.2.4 监控与配件有关的事件
 - 18.3 访问加速计事件
 - 18.3.1 配置加速计
 - 18.3.2 选择恰当的更新频率
 - 18.3.3 从加速计数据中分离重力成分
 - 18.3.4 从加速计数据中分离实时运动成分
 - 18.3.5 取得当前设备的方向
 - 18.4 Core Location框架提供定位服务
 - 18.4.1 获取用户的当前位置
 - 18.4.2 获取方向相关的事件
 - 18.5 Map Kit框架提供地图服务
 - 18.5.1 在用户界面中加入地图视图
 - 18.5.2 缩放和移动地图内容
 - 18.5.3 显示用户的当前位置
 - 18.5.4 坐标和像素之间的转换
 - 18.5.5 通过反向地理编码器获取地标信息
 - 18.6 地图上注解的实现
 - 18.6.1 在地图视图上显示注解
 - 18.6.2 添加和移除注解对象
 - 18.6.3 定义注解视图
 - 18.6.4 创建注解视图

- 18.6.5 处理注解视图中的事件
- 18.7 UIKit框架的应用
 - 18.7.1 显示照相机界面
 - 18.7.2 关闭照相机界面
 - 18.7.3 显示照片选取器界面
- 18.8 使用邮件编辑界面
- 18.9 小结
- 第四部分 实战篇
- 第19章 iPad应用开发实战--精灵小书柜
 - 19.1 系统的总体设计方案
 - 19.1.1 系统模块组成
 - 19.1.2 启动项目
 - 19.2 组装书柜
 - 19.2.1 书柜的实现
 - 19.2.2 书柜的框架及组装
 - 19.2.3 设置书柜的显示效果
 - 19.3 在小书柜上展示书的封面
 - 19.3.1 构建书的封面
 - 19.3.2 将书的封面展示在书柜中
 - 19.3.3 构建多面书柜
 - 19.3.4 通过手势展示多面书柜
 - 19.4 阅读书的内容
 - 19.4.1 设计内容显示的模板
 - 19.4.2 书的封面与内容关联
 - 19.4.3 动画效果翻页
 - 19.4.4 返回书柜主界面
 - 19.5 小结
- 第20章 iPhone应用开发实战--弹球游戏
 - 20.1 认识游戏引擎
 - 20.1.1 Cocos2D游戏引擎
 - 20.1.2 Box2D物理引擎
 - 20.1.3 Chipmunk物理引擎
 - 20.1.4 游戏引擎组成结构
 - 20.1.5 物理引擎中的刚体
 - 20.2 Cocos2D框架中的常用类
 - 20.2.1 节点类CCNode
 - 20.2.2 场景类CCScene
 - 20.2.3 层类CCLayer
 - 20.2.4 标签类CCLabel
 - 20.2.5 动作类CCAction
 - 20.2.6 导演类CCDirector
 - 20.2.7 精灵类CCSprite
 - 20.3 弹球游戏的准备工作
 - 20.3.1 弹球游戏实现目标
 - 20.3.2 安装Cocos2D-iPhone
 - 20.3.3 启动项目的创建
 - 20.3.4 程序控制权的转交

- 20.4 实现场景及其互动对象
 - 20.4.1 添加自定义的游戏场景
 - 20.4.2 创建场景类GameScene
 - 20.4.3 创建World对象和边界框
- 20.5 制作弹球
 - 20.5.1 添加弹球精灵
 - 20.5.2 创建弹球body并添加冲力
 - 20.5.3 使用tick方法刷新对象
- 20.6 设计球拍
 - 20.6.1 增加球拍
 - 20.6.2 移动球拍
 - 20.6.3 限制球拍
 - 20.6.4 球反弹的优化处理
- 20.7 方块的实现
 - 20.7.1 增加方块
 - 20.7.2 销毁方块
- 20.8 游戏逻辑处理
 - 20.8.1 碰撞检测机制
 - 20.8.2 球碰到屏幕底部
 - 20.8.3 游戏结束的条件
 - 20.8.4 添加游戏结束场景
- 20.9 添加游戏音乐
- 20.10 小结

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>