南 阳 理 工 学 院

本科生毕业设计(论文)

学院(系)： 软件学院

专 业： 软件工程

学 生： 王成丹

指导教师： 许豪

完成日期 2013 年 11 月

南阳理工学院本科生毕业设计（论文）

基于iOS7 Sprite Kit的  
连连看的设计与实现

Design and Implementation of the  
LinkGame Base On iOS7 Sprite Kit

总 计：毕业设计(论文) 1页

表 格： 1个

插 图： 1幅

南 阳 理 工 学 院 本 科 毕 业 设 计(论文)

基于iOS7 Sprite Kit的  
连连看的设计与实现

Design and Implementation of the  
LinkGame Base On iOS7 Sprite Kit

学 院(系)： 软件学院

专 业： 软件工程

学 生 姓 名： 王成丹

学 号： 105151840028

指导教师(职称)： 许豪 讲师

评 阅 教 师： 王五

完 成 日 期： 2013年11月31日

南阳理工学院

Nanyang Institute of Technology

基于iOS7 Sprite Kit的  
连连看的设计与实现

软件工程 王成丹

1. 当今随着社会的发展，我们现代人进入的快节奏的生活，在紧张的工作学习之余，可通过小游戏，看电影等娱乐方式来放松心情，缓解疲惫的身心。连连看这款小游戏正是下载率和好评率最高的小游戏之一，深受大众的喜爱。2013年9月份iOS7正式推出, 凭借全新设计和全新功能，iOS 7 的形态再次进化，成为更加了不起的先进移动操作系统，在发布不到一周的时间内，更新就超过了50%。与iOS7一起发布的还有Sprite Kit，苹果全新Sprite Kit开发框架的目的是让开发者更容易为iOS 7移动设备和Mac桌面电脑创建2D游戏。但由于iOS 7界面变化比较大，很多应用都需要进行一些界面元素的改变才能更好的适配iOS 7。现在，苹果App Store新上线了“为 iOS 7 设计”版块，用户可以在新版块中找到已经为iOS 7优化的应用。但该版块应用还相对匮乏，仍有很大的空间，在如此情况下基于iOS7 Sprite Kit的连连看就应运而生。本文提出了一个基于iOS7 Sprite Kit的连连看的设计与实现，文章从IOS7开发特点、设计模式、Sprite Kit框架应用、连连看算法、系统设计与实现等几个方面出发，分别对该系统进行了详细的阐述。
2. iOS7，Sprite Kit，连连看

Design and Implementation of the  
LinkGame Base On iOS7 Sprite Kit

Software Engineering Major Wang Chengdan

1. Today，with the development of society, our modern fast-paced life entering , remaining intense work in learning , through games, watch movies and other entertainment to relax and relieve physical and mental fatigue . The LinkGame is one of the highest rates and the favorable rate of small game download ,it be loved by the public . September 2013 iOS7 officially launched , with a new design and new features , iOS morphological evolution 7 again become greater advanced mobile operating system , released in less than a week's time , updated more than 50%. Sprite Kit was Released together with iOS7, Apple's new Sprite Kit purpose development framework is to allow developers to more easily create 2D games for the iOS 7 mobile devices and Mac desktop computers. However, due to changes in iOS 7 interface is relatively large , many applications require some interface elements changed to better fit iOS 7. Now, Apple's new App Store on the line "for the iOS 7 design " section, users can find the new forum has been optimized for the iOS 7 applications . However, the forum application is relatively scarce, there is still a lot of space , in such a case based iOS7 Sprite Kit Snake came into being . This paper presents the design and implementation of a lianliankan iOS7 Sprite Kit , based on articles from the characteristic aspects IOS7 development , design patterns , Sprite Kit Framework applications , Linkgame algorithms, system design and implementation , etc., were carried out of the system detail.
2. iOS7，Sprite Kit，Linkgame

目 录

[1 概述 1](#_Toc374309800)

[1.1 研究背景 1](#_Toc374309801)

[1.2 研究内容 1](#_Toc374309802)

[1.3 论文的内容和结构 1](#_Toc374309803)

[2 iOS7 Sprite Kit开发技术 2](#_Toc374309804)

[2.1 IOS开发框架 2](#_Toc374309805)

[2.2 开发语言Objective-C 4](#_Toc374309806)

[2.3 内存管理 4](#_Toc374309807)

[2.4 关于 Sprite Kit 4](#_Toc374309808)

[2.5 开发工具 9](#_Toc374309809)

[3 系统可行性研究与需求分析 10](#_Toc374309810)

[3.1 可行性分析 10](#_Toc374309811)

[3.1.1 操作可行性 10](#_Toc374309812)

[3.1.2 技术可行性 10](#_Toc374309813)

[3.1.3 经济可行性 10](#_Toc374309814)

[3.1.4 社会可行性 10](#_Toc374309815)

[3.2 功能需求分析 10](#_Toc374309816)

[3.2.1 系统流程图 12](#_Toc374309817)

[3.2.2 系统模块设计 13](#_Toc374309818)

[3.2.3 系统风格定位 15](#_Toc374309819)

[3.3 本章小结 15](#_Toc374309820)

[4 系统设计 15](#_Toc374309821)

[4.1 用例设计 15](#_Toc374309822)

[4.1.1 连连看用例描述 15](#_Toc374309823)

[4.2 模块设计 17](#_Toc374309824)

[4.3 连连看的算法设计 18](#_Toc374309825)

[4.3.1 连连看的游戏规则 18](#_Toc374309826)

[4.3.2 地图生成算法设计与分析 19](#_Toc374309827)

[4.3.3 图形连通算法设计 20](#_Toc374309828)

[4.4 本章小结 21](#_Toc374309829)

[5 编码与实现 21](#_Toc374309830)

[6 系统测试与运行 21](#_Toc374309831)

[6.1 软件测试简介 21](#_Toc374309832)

[6.2 测试步骤 21](#_Toc374309833)

[6.3 游戏系统测试 22](#_Toc374309834)

[6.4 游戏性能分析 23](#_Toc374309835)

[6.5 本章小结 23](#_Toc374309836)

[结束语 24](#_Toc374309837)

[参考文献 25](#_Toc374309838)

[附录 26](#_Toc374309839)

[致谢 27](#_Toc374309840)

# 概述

## 研究背景

iOS是由苹果公司为iPhone开发的操作系统，苹果公司最早于2007年1月9日的Macworld大会上公布这个系统，苹果首版iOS系统中的众多设计元素和用户体验，拟物化的风格多年来被保留至iOS 6。2013年9月份iOS7正式推出,iOS7在用户界面上有着与之前版本完全不同的视觉设计，开始扁平化风格路线，以往的拟物风格在iOS 7中不再出现。凭借全新设计和全新功能，iOS 7 的形态再次进化，成为更加了不起的先进移动操作系统，在发布不到一周的时间内，更新就超过了50%。与iOS7一起发布的还有Sprite Kit，苹果全新Sprite Kit开发框架的目的是让开发者更容易为iOS 7移动设备和Mac桌面电脑创建2D游戏，Sprite Kit提供了一个图形渲染（rendering）和动画的基础，你可以使用它让任意纹理（textured）图像或精灵动起来，Sprite Kit还提供了其他对游戏非常有用的功能，包括基本的声音播放支持和物理模拟。用它可以将音乐，声效，动画以及富有新意的界面融合在一起，以制作出界面友好、高效稳定和富有娱乐性的游戏。

当今随着社会的发展，我们现代人进入的快节奏的生活，在紧张的工作学习之余，可通过小游戏，看电影等娱乐方式来放松心情，缓解疲惫的身心。连连看这款小游戏正是下载率和好评率最高的小游戏之一，深受大众的喜爱。

由于iOS 7界面变化比较大，很多应用都需要进行一些界面元素的改变才能更好的适配iOS 7。现在，苹果App Store新上线了“为 iOS 7 设计”版块，用户可以在新版块中找到已经为iOS 7优化的应用。相信不久之后，相信不久之后，会有越来越多的应用选择使用iOS 7风格设计。但该版块应用还相对匮乏，仍有很大的空间，在如此情况下基于iOS7 Sprite Kit的连连看就应运而生。

## 研究内容

为了实现基于iOS7 Sprite Kit的连连看，在开发过程中存在着iOS7新特性、 Sprite Kit框架的使用、视图管理、内存管理、连连看算法实现等几个难点需要克服。

## 论文的内容和结构

本文共分为6章，各章主要内容介绍如下：

第1章 概述。介绍论文的选题背景，游戏的基本功能的实现以及论文结构的安排。包括课题的背景资料简介、国内外研究现状和发展趋势、课题研究的价值以及影响。为本游戏的编写调试提供理论依据。

第2章 iOS7 Sprite Kit开发技术。这部分主要是介绍实现系统的技术支撑体系和系统的开发方法。主要包括：开发基本知识，iOS7 Sprite Kit技术等，并且详细介绍了iOS的基本知识、各种特点、Sprite Ki的优势和手机游戏开发的市场价值以及iOS开发的详细步骤和游戏使用的主要技术及问题。

第3章 系统可行性研究与需求分析。本章概要地阐述了本系统的体系结构，描述了各个功能模块，并分析了系统功能与系统角色和流程分析，展示了本系统的流程图，为下一章的系统的设计与实现打下基础。

第4章 系统的设计。本章通过对基于IOS的连连看游戏的各个功能的用例描述，通过用例设计、模块设计，描述了系统的实现思路和游戏核心算法，展示了各个模块的模型和界面图，全面深刻的解析了该系统所具有的全部功能极其实现的方法。对每个界面的设计以及思路进行了详细的解释。

第5章 编码与实现。这部分主要介绍了展示了游戏的界面设计和功能的编码实现，全面深刻的解析了该系统所具有的全部功能及其实现的方法，并对每个功能的核心代码进行了详细的解释。

第6章 系统测试与性能分析。描述了软件测试的基本方法、测试的总类和测试的步骤，对游戏进行了的系统测试和性能上的分析，达到了需求分析中的功能要求，实现了连连看游戏的设计。以上标志着本次基于IOS的连连看游戏的设计与实现工作成功的完成。

# iOS7 Sprite Kit开发技术

## IOS开发框架

 框架是一个目录，这个目录包含了共享库，访问共享库里代码的头文件，和其它的图片和声音的资源文件。一个共享库定义的方法或函数可以被应用程序调用。IOS提供了很多可供在应用程序里调用的框架。要使用一个框架，需要将它添加到项目中之后才可以使用它。许多应用程序都使用了如 Foundation、UIKit、和Core Graphics这些框架。

几乎所有的iOS应用都会用到一些由UIKit框架所定义的UI控件。对这些基本控件的名称、角色和作用加以了解，你将能更好的在打造界面的过程中做出正确合理的设计决策。

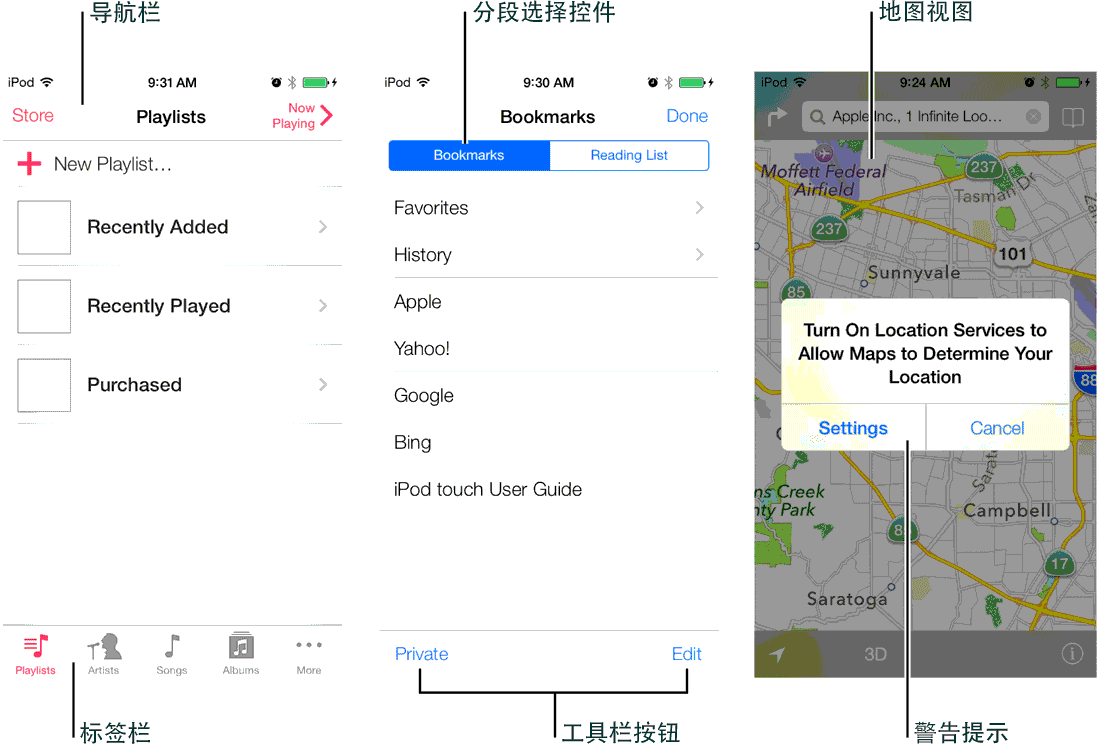


图2–1iOS7主要视图元素

UIKit框架提供的UI元素可以分为4大类：

栏(Bars)：包含与上下文内容情境相关的信息，告诉用户他们在应用当中所处的位置；同时还包含相关的导航功能，或是可以由用户发起的操作。

内容视图(Content views)：包含当前应用所特有的内容，并支持相关的交互行为，例如滚屏、插入、删除、对列表条目进行重新排序，等等。

控制元素(Controls)：产生行为或显示信息。

临时视图(Temporary views)：临时性的向用户呈现重要信息，或提供额外的功能及选项。

除了定义UI元素以外，UIKit框架还定义了那些可以完成诸如手势识别、绘制、辅助支持、打印支持等功能的对象。

从编程的角度来讲，UI元素可以被看做是不同类型的视图(views)，因为它们都继承自UIView类。一个视图知道怎样将它自己输出到屏幕上，知道怎样捕获用户在它的范围内执行的点击行为。

要在应用中管理一组或一系列具有继承关系的视图，你通常需要使用视图控制器(view controller)。视图控制器可以协调不同视图的显示方式，在用户发起行为后执行相应的功能，并且对界面切换时所需的动画过渡效果进行管理。虽然开发者们习惯于从视图和控制器的角度来思考问题，但用户只会将应用看做是由一个个界面所组成的整体。从这个角度讲，每一“屏”都对应着应用中的一个明确的视觉状态或模式。

每个iOS应用都包含一个窗口([window](https://developer.apple.com/library/prerelease/ios/documentation/General/Conceptual/Devpedia-CocoaApp/Window.html))与桌面应用中的概念有所不同，iOS中的窗口没有真正的可视部分，而且不能从屏幕上的一个地方移动到另外一个地方。多数iOS应用只有一个窗口,那些支持外接显示设备的应用会有多个窗口。iOS界面设计规范当中会用到“屏(screen)”这个词，因为它能被多数用户所理解。作为开发者，你可能还会在其他不同的上下文当中读到“屏”的概念，例如被用来指代UIScreen对象(用于访问外界显示设备)[。](http://beforweb.com/)

  每个框架对应IOS系统里的一层，每层建立在它下面层的上面。应该尽量使用上层的框架来代替下面的框架。更高层次的框架是对底层框架基于对象的抽象。IOS框架是系统运行的基础，它是硬件与软件结合的纽带，是iPhone、iPad及iTouch等设备的核心，是其应得以运行的平台环境。

## 开发语言Objective-C

Objective-C 是一种通用、高级、面向对象的编程语言，它扩展了标准的 ANSI C 编程语言。它主要使用于Mac OS X和GNUstep这两个使用OpenStep标准的系统，而在NeXTSTEP和OpenStep中它更是基本语言, Objective-C可以在gcc运作的系统写和编译，因为gcc含Objective-C的编译器。它是苹果的 OS X 和 iOS 操作系统，及其相关 API、Cocoa 和 Cocoa Touch 的主要编程语言。Objective-C是面向对象编程语言，它具有封装、继承、多态等面向对象语言的特性，同时它与其他语言相比更为简单，它支持单一继承、类封装、多态、接口、动态类型等特性。Objective-C语言对动态机制的支持较为彻底，编译器为运行环境保留了很多对象自身的信息，一些需要在编译时需要确定的信息可以推迟到运行时再确定。由于IOS应用的快速发展，Objective-C语言使用的频率也大大增加，跃入语言排行榜上的前十。

虽然Objective-C是C的母集，但它也不视C的基本型别为第一级的对象。 和C++不同，Objective-C不支援运算子多载(它不支持ad-hoc多型)。亦与C++不同，但和Java相同，Objective-C只容许对象继承一个类别(不设多重继承)。

## 内存管理

IOS系统中的Objective-C的内存管理机制是比较灵活的，Objective-C 2.0在之前手动管理内存机制的基础上，提供了一个可选的垃圾收集器。在iOS 5后增加了Automatic Reference Counting（ARC 自动引用计数）特性。

## 关于 Sprite Kit

iOS 7新引入的Sprite Kit类库算是给iOS游戏开发者带来一些福音，由于是用底层的东西做开发，在效率方面应该会有很大的提高。也不单单是在游戏方面，任何需要动画效果的app也可以利用Sprite Kit来实现。SpritKit框架用来做图像的渲染，并且可以将一系列的纹理图像或者一些sptite（精灵）做成动画。它采用传统逐帧渲染的方式来进行渲染工作，并且期间会经历一个渲染过程（如图）。它通过图像硬件来渲染每一帧图像，SpritKit做了充分的优化，这样就可以使被渲染的对象能够在每一帧中反复的修改位置。除此之外，SptitKit框架为游戏开发提供了其他的基本功能，如背景音乐、模拟物理环境。Xcode5里集成了这个框架和相关工具，使用起来就更加方便。

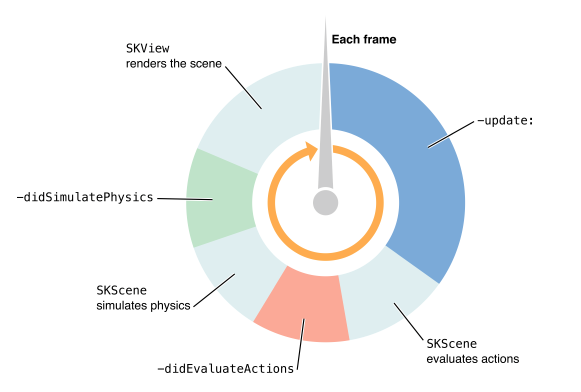


图2–2Sprite Kit图形渲染

动画和渲染工作是由SKView对象负责执行。SKView可以放置在window视图中，然 后做渲染工作。由于SKView也是一个视图，所以按照视图层级关系，你可以将其他视图对象放置于其中，作为其子视图。

Sprite Kit提供了一个图形渲染（rendering）和动画的基础，可以使用它让任意纹理（textured）图像或精灵动起来。Sprite Kit采用的是传统的渲染循环，允许在渲染前处理每一帧的内容。你的游戏确定场景的内容，以及这些内容如何在每帧中变化。Sprite Kit做的工作，就是有效地利用图形硬件来渲染动画的帧。Sprite Kit优化到本质上允许对动画每一帧的任意修改。

Sprite Kit还提供了其他对游戏非常有用的功能，包括基本的声音播放支持和物理模拟。此外，Xcode中提供了内置的Sprite Kit支持，可以很容易地创建并在你的应用程序中使用复杂的特效和纹理图册（atlases）。这种框架和工具的组合，使Sprite Kit对于游戏和其他需要类似动画的应用程序是一个很好的选择。对于其他类型的用户界面动画，使用Core Animation代替。

Sprite Kit在iOS和OS X可用。它使用主机设备提供的图形硬件，以高帧速率复合2D图像。Sprite Kit支持多种不同类型的内容，包括：

•    无纹理或有纹理的矩形（精灵）

•    文本

•    任意基于CGPath的形状

•    视频

Sprite Kit还提供了裁剪和其它特效的支持，允许你对全部或者部分内容应用这些效果。你可以在每一帧中活动（animate）或者改变这些元素。你也可以附加物理主体到将这些元素，使他们正确地支持武装和碰撞。

通过支持丰富的渲染基础和处理所有低级别的工作来提交OpenGL的绘图命令，Sprite Kit允许你全神贯注解决更高层次的设计问题，并创造伟大的游戏。

在Sprite视图内由呈现场景绘制Sprite内容

动画和渲染由SKView对象执行。你在一个窗口在放置这个视图，然后渲染它的内容。因为它是一个视图，所以它的内容可以结合在视图层次里的其它视图。

 游戏中的内容会被组织成场景（scenes），用SKScene对象代表它们。场景包含精灵和其他要渲染的内容。场景也实现了每帧的逻辑和内容处理。在任何给定的时间内，视图展示一个场景。只要一个场景被呈现出来，它的动画和每帧逻辑会自动执行。

 要使用Sprite Kit创建一个游戏，需要创建一个或多个的SKScene类的子类。例如，创建单独的场景类来分别显示主菜单、游戏画面和游戏结束后显示的内容。你可以很容易地在你的窗口中使用一个单一的SKView对象并在不同场景之间进行过渡。

节点树定义出现在一个场景中的内容

SKScene类实际上是SKNode类的后代。节点是所有内容的基本构建块，而场景对象作为一个节点对象树的根节点。场景及其后代决定哪个内容被绘制以及它渲染的方式。

 每个节点的位置在它的父节点定义的坐标系中指定。节点的内容的其他属性也适用于它后代的内容。例如，当一个节点是旋转，所有它的后代也跟着旋转。你可以使用节点树建立一个复杂的图像，然后通过调整最上层节点的属性旋转、缩放并融入整个图像。

 SKNode类绘制任何东西，但它对后代应用于它的属性。每一种可绘制内容 由Sprite Kit的不同子类表示。其他的节点子类不直接绘制内容，但修改它们后代的行为。例如，你可以在场景中使用一个SKEffectNode对象对整个子树应用一个核心图像过滤器（Core Image filter）。通过精确控制节点树的结构，你确定节点的渲染顺序，让你可以在一个场景中布局（layer）复杂的图形效果。

 所有节点对象都是响应者（responder）对象，派生（descending）自UIResponder或NSResponder，所以你可以继承任何节点类来创建接受用户输入的新类。视图类自动扩展响应链来包含场景的节点树。

纹理保存可复用的图形数据

纹理是用来渲染精灵的共享图像。当你需要对多个精灵应用相同的图像时，总是使用纹理。通常你通过加载存储在你的应用程序bundle的图像文件来创建纹理。然而，Sprite Kit也可以在运行时从包括核心图形图像在内的其他来源为你创建纹理，或者甚至渲染把节点树成纹理。

 Sprite Kit通过处理较低级别的代码需求来加载纹理和并让它们对图形硬件可用，来简化了纹理的管理。纹理管理由Sprite Kit自动管理。但是，如果你的游戏中使用了大量的图像，你可以通过控制部分的过程来提高性能。首先，你通过提示Sprite Kit纹理很快就需要来做这个。

 纹理图册是在你的游戏中一起使用的一组相关的纹理。例如，你可以使用一个纹理图册存储让一个角色活动需要的所有纹理或渲染游戏设置级别的背景需要的所有瓷砖。Sprite Kit用纹理图册来提高渲染性能。

 动作在场景中由节点执行

使用动作（actions）让场景的内容动起来。每一个动作都是一个对象，由SKAction类定义。你来告诉节点执行动作。然后，当场景处理动画帧，动作就被执行。有些动作在一帧动画内完成，而另一些在完成前应用变化于多帧动画。动作最常见的用途是改变节点的属性。例如，你可以创建动作来移动、缩放或旋转节点，或使其透明。然而，动作也可以更改节点树、播放声音、甚至是执行自定义代码。

 动作是非常有用的，但你也可以组合动作来创建更复杂的效果。你可以创建一组同时运行或顺序运行的动作。你可以让动作自动重复。

 场景中也能执行自定义的每帧处理。覆盖你的场景子类的方法来执行额外的游戏任务。例如，如果一个节点需要每帧移动，你可能会直接每帧地调整其属性而不是使用一个动作来这样做。

 添加物理主体和关节来在场景中模拟物理

虽然你可以控制场景中的每一个节点的确切位置，你经常想这些节点互相交流、碰撞并在这个过程中告知速度的变化。你可能还需要模拟重力和其他形式的加速度，这些都不在动作系统中处理的。要做到这一点，你可以创建物理主体（SKPhysicsBody），并将它们附加到你场景中的节点上。每个物理主体由形状、尺寸质量和其他物理特性定义。

 当场景中包含物理主体，场景就在这些主体上模拟物理。一些势力（forces），如重力和摩擦力，会自动应用。你也可以对物理主体调用方法来应用自己的势力。每个主体的加速度和速度会被计算，然后主体彼此碰撞。然后，模拟完成后，相应的节点的位置和旋转的被更新。

 你物理主体的交互拥有精确的控制。你确定哪些主体被允许相互碰撞 并单独决定哪些交互可以被你的应用程序调用。这些回调允许你勾（hook）到物理模拟中创建其他的游戏逻辑。例如，在一个物理主体被另一个物理主体击中时，你的游戏可能会销毁一个节点。

 场景在一个附加的SKPhysicsWorld对象上定义了物理模拟的全局特性。你可以使用物理世界定义整个模拟的重力，定义模拟的速度，并在场景中查找物理主体。你还可以使用物理世界通过一个关节（SKPhysicsJoint）把物理主体连接在一起。连接的主体根据关节的类型模拟在一起。

显示于场景中的节点树

SKScene类是SKNode类的子类。当使用SpritKit框架的时候，一个个节点为所有内容构成一个节点树，场景对象（SKScene类的实例或者其子类的实例）就是这些节点对象树的根节点。由它来决定哪些内容要被渲染出来、如何被渲染出来。

 每一个节点的坐标是根据其父类节点的系统坐标定义的。每一个节点为它其中子节点及其内容提供一些属性，例如，一个节点旋转，它里面包含的子节点及其内容也要跟着旋转。你可以使用节点树创建一个复杂的图像，整个图像会根据最外层节点属性的变化而发生变化。

 SKNode类不能绘制任何东西，但是它可以将自己的属性传递给它的子节点。绘制图像的工作由SpriteKit框架中的其他子类负责。其他的子节点也不能为自己绘制内容，但它们可以为其中的内容传递一些行为。例如，你可以使用SKEffectNode对象为场景中的整个子节点树提供一个图像滤镜。恰当地对节点树结构的控制，你可以决定哪些节点可以被渲染。

 所有的节点对象都是responder对象，都继承自UIResponder或NSResponder，所以你的任何节点的子类都可以响应用户输入。

## 开发工具

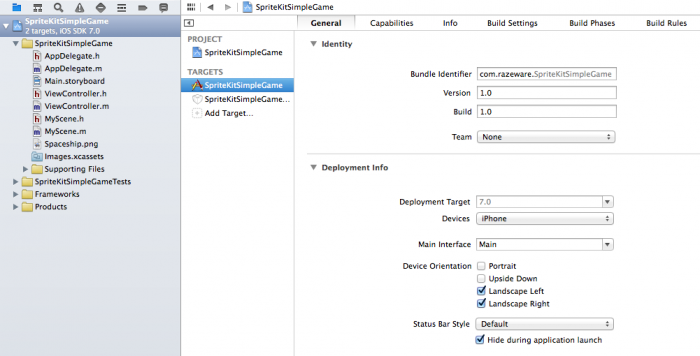


图2–3Xcode视图

要开发iOS的应用程序，需要一台安装有Xcode工具和Mac OS X的电脑。Xcode是苹果提供的开发工具集、提供项目管理、代码编辑、创建执行程序、代码级调试、代码库管理和性能调节等等功能。Xcode 是用于构建 Mac app 和 iOS app 的完整工具集。而 Xcode 5 中的这些工具比以往更强大，能帮助您创建快速、高效的高质量 app。Xcode 自动配置您的 app 以使用最新的 Apple 服务，在统一的资产目录下管理您的众多图像，并帮助您在 iOS 7 或 OS X Mavericks 上设计出外观精美的 app。

Xcode 5现在使用了ARC，取代了原来的垃圾回收（Garbage collection）机制，因此不论从启动速度和使用速度上来说都比之前快了不少。现在大部分的AppStore提交应用也都使用了ARC，新SDK中加入的系统框架也全都是ARC的了。另外，在Xcode5中新建工程也不再提供是否使用ARC的选项（虽然也还是可以在Build Setting中关掉）。

这个工具集的核心就是Xcode程序，提供了基本的源代码开发环境。Xcode是一个集成开发环境(IDE)，通过与 Cocoa 和 Cocoa Touch 框架紧密集成，Xcode 创建了一个具有高效产出且易于使用的开发环境。Xcode 工具箱包含出色的 Xcode IDE，并且与 Interface Builder 设计工具和 Apple LLVM 编译器进行了深度集成。Instruments 分析工具也包含于工具包之中，另外还有其他很多提供各种支持的开发者工具。构建代码编译成为可执行文件后，可在iPhone模拟器或者真实设备上运行和调试代码。

当你在Xcode中构建应用程序，你可以选择为iPhone模拟器创建，也可以选择为iPhone设备创建。模拟器提供了测试应用程序的本地环境，可以确保你的程序行为完全符合你的需求。当你对程序的基本行为已经满意后，你可以告诉Xcode构建它，然后在连接到你的计算机上的iPhone和iPod touch上运行它。在设备上面运行程序提供了终极测试环境，而且Xcode可以把内建调试器植入运行在设备上的代码中。

# 系统可行性研究与需求分析

## 可行性分析

### 操作可行性

游戏的操作可行性其实是一个游戏成功与否的关键，在很多著名游戏中游戏操作合理性经常成为玩家讨论的焦点，所以说游戏设计的合理性既是需要考虑的关键也是对玩家选择此游戏的主要吸引力。

游戏时间不能过长也不能过短，在经过反复思考和老师的指导后，决定将时间做成第一关3分钟左右，往后逐渐递减。这一决定保证了游戏的正常完成，同时也是考虑到了制作时间的限制。

在“连连看游戏”设计中，在游戏中玩家可以通过每消去两个图片，都可以得到增加游戏剩余时间的机会。而在进入第二关后，随着游戏复杂性和难度的提升，再加上游戏时间的逐渐减少，因而在这样的设计下，玩家可以感受到游戏的可玩性，同时也可以享受到完成游戏的成就感。

### 技术可行性

iOS以Objective-C作为开发语言，安装好Xcode5就完成了开发环境的搭建。而且在Xcode中，它自带了一些常用的组件。这为我们对这各组件开发提供了很大的帮助。因此在技术上是可行的。

### 经济可行性

开发该系统所需的相关资料可以通过已存在的相关系统进行调查采集，所需的其他应用软件、硬件系统也易于获得.因此,开发成本较低。而引进使用本系统后，与传统方式相比，具有高效率、低成本、高质量的特点，可以节省不少人力、物力及财力。所以，从经济的角度来看，该系统可行。

### 社会可行性

本游戏丰富的道具和公共模式的加入，增强游戏的竞争性。多样式的地图，使玩家在各个游戏水平都可以寻找到挑战的目标，长期地保持游戏的新鲜感。IOS游戏的大量玩家是年轻人，而现在年轻人的巨大的消费潜力使得本游戏开发具有社会可行性。

## 功能需求分析

业务逻辑模块主要是对数据进行逻辑处理。主要可分为三大模块即欢迎模块、游戏模块、图形模块。

（1）欢迎模块

打开软件后，首先展示给用户的是一个欢迎动画，随后就是主界面，包含了“开始游戏”、“游戏设置”、“关于游戏”和“退出游戏”四大模块。

（2）游戏模块

玩家点击“开始游戏”后，进入游戏界面。

用户点击开始游戏后，系统开始倒计时，屏幕最上行有倒计时的时间进度条，第一关设置时间为1000s，之后每关分别为[1000-125\*(关数-1)]s，直到时间小于等于0s时，就把此关的游戏时间规定为100s。

玩家每消掉任何一对图形会出现线条连路方式出现以及显示消图效果发出消图声音并且时间+5s。玩家每消掉一对图形后得分为关数\*10分。

如果玩家在规定的时间内将所有的图形消掉，系统提示玩家：恭喜过关，并进入下一关。在系统所给的时间内没有消掉所有图形则将退出游戏，并提示：游戏结束！如果排行榜为空或者玩家的得分已经超过排行榜的得分，则会提示输入姓名，将得分保存并更新到排行榜上。

如果玩家自行选择重排，游戏剩余时间减去5\*（4+关数/2）秒。

如果游戏出现了死锁的情况（也就是说没有图形可以相消的情况），系统会提示玩家：没有可以消的方块了，请使用游戏道具“重排”，然而“重排”每一次的游戏中最多只能用两次，如果重列用完了之后还出玩死锁情况，系统便自动再次重新排列，直到不出现死锁。

（3）图形模块

图形模块是用来管理图形的。包括图形的图片，图形的随机排列，图形的显示和消除等。

系统基本流程

本游戏是一款基于IOS的手机连连看游戏，玩家进入游戏后，进入主界面，显示开始游戏、游戏设置、游戏介绍、游戏帮助和退出游戏。

当玩家点击“开始游戏”后，玩家在规定时间内消去全部图片，即可进入下一关，如果没有在规定时间内消去全部图片，则失败。当玩家所得的分数高于游戏中排行榜的分数，则可以在排行榜上保存姓名和得分。玩家在游戏过程中，可以点击“菜单”按钮，选择“重新开始游戏”、“重新排列图片”和“退出游戏”；

当玩家在菜单里选择“退出游戏”后，将进入系统主界面，当玩家点击“游戏设置”后，可以对游戏的包括声音、震动、选关等各方面进行设置，当进行完设置之后，可以点击“保存退出”对所做操作进行保存；

当玩家点击“保存退出”后，将进入系统主界面，当玩家点击“关于游戏”后，可以看到游戏的排行榜，也可以清除排行榜的记录，还可以看到游戏的帮助文档，向玩家介绍本游戏的使用方法。

当玩家点击主界面的“退出游戏”，即退出连连看游戏程序。

### 系统流程图

系统流程图，如图3-1所示：

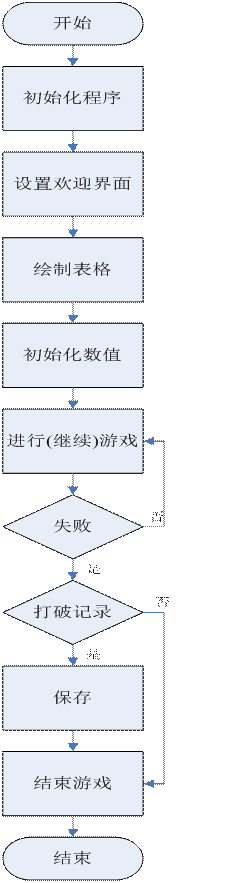


图3–1系统流程

游戏详细过程流程图，如图3-2所示：

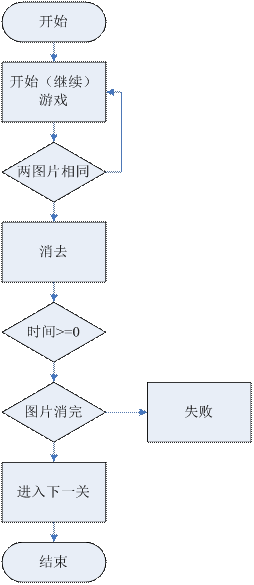


图3–2游戏详细过程流程图

### 系统模块设计

1.系统总体设计，总共分为五个模块，分别是：“菜单设计”，“界面设计”，“算法设计”，“消息响应设计”和“用户管理设计”，如图3-3所示：

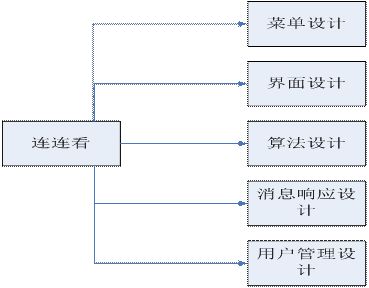


图3–3系统总体设计

2．系统详细设计：

（1）菜单设计总共分为四个模块，分别是：“开始游戏”，“游戏设置”，“规则说明”和“退出游戏”，图3-4所示：

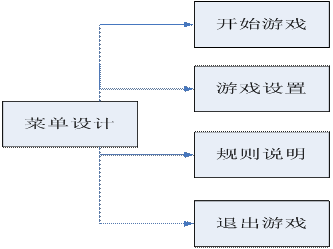


图3–4菜单设计

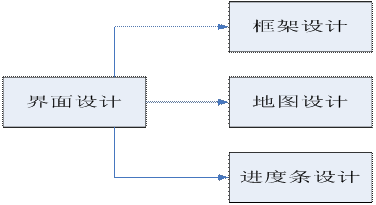
（2）界面设计，总共分为三个模块，分别是：“框架设计”，“地图设计”和“进度条设计”，如图3-5所示：

图3–5界面设

（3）算法设计，总共分为两个个模块，分别是：“画线设计”和“消除算法设计”，如图3-6所示：

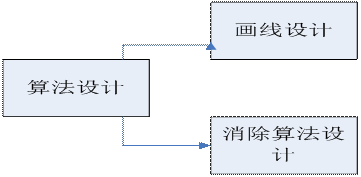


图3–6算法设计

（4）用户管理设计，只有排行榜设计一个模块，只有当排行榜为空或者玩家得分超过排行榜，才用得到，

### 系统风格定位

iOS游戏的大量玩家是年轻人，鉴于这点我的设计风格选择了可爱，图片基本上都是可爱的图片。这样设计首先是基于用户群，第二在参考了很多网上的连连看游戏，多数是纸牌造型，所以这样可以让本游戏变得更有个性。同时也尝试了很多其他的风格，但最后还是觉得可爱风格最适合本游戏。

## 本章小结

本章概要地阐述了本系统的体系结构，描述了各个功能模块，并分析了系统功能与系统角色和流程分析，为下一章的系统的设计与实现打下基础。

# 系统设计

## 用例设计

### 连连看用例描述

在“连连看系统”用例图中共有“玩游戏”、 “查看排行榜”、“游戏设置”、“查看帮助”、“背景设置”、 “音乐设置”、 “提示路径”、“重新排序”共8个用例。

玩游戏用例

|  |
| --- |
| 用例编号：1 |
| 用例名：玩游戏 |
| 执行者：玩家 |
| 目的：玩游戏 |
| 过程描述：  玩家启动系统;  在主界面中选择你想玩的模式，例如：鼠标点击“经典模式”按钮；  进入到游戏级别界面中，选择级别，例如：鼠标点击“初级”按钮；  开始玩游戏。 |
| 前置条件：一个想玩游戏的人 |
| 后置条件：排行榜内容更新 |

表4–1玩游戏用例

查看排行榜用例

|  |
| --- |
| 用例编号：2 |
| 用例名：查看排行榜 |
| 执行者：玩家 |
| 目的：查看自己游戏结果的排名、成绩及通关情况等详情。 |
| 过程描述：  在主界面中用鼠标点击“英雄榜”按钮，进入到排行榜界面；  开始查看详情  点击“确定”按钮，用例终止。 |
| 前置条件：已有玩家玩过游戏 |

表4–2查看排行榜用例

游戏设置用例

|  |
| --- |
| 用例编号：3 |
| 用例名：游戏设置 |
| 执行者：玩家 |
| 目的：设置游戏的属性及性能。 |
| 过程描述：  在主界面中点击“游戏设置”按钮，进入“游戏设置”界面；  在游戏设置界面中，根据需要设置。  点击“返回”按钮，用例终止。 |
| 前置条件：想要对游戏进行设置 |
| 后置条件：游戏属性被改变 |

表4–3游戏设置用例

查看帮助用例

|  |
| --- |
| 用例编号：4 |
| 用例名：查看帮助 |
| 执行者：玩家 |
| 目的：了解该系统的功能及使用方法 |
| 过程描述：  在主界面中点击“查看帮助”按钮，进入“游戏帮助”界面；  在“游戏帮助”界面中，可以根据需要查看与该系统相关功能和使用方法  点击“关闭”按钮，用例终止。 |
| 前置条件：初次使用该系统或对该系统不了解 |
| 后置条件：玩家知道如何使用此系统 |

表4–4查看帮助用例

背景设置用例

|  |
| --- |
| 用例编号：5 |
| 用例名：背景设置 |
| 执行者：玩家 |
| 目的：更换系统背景图片 |
| 过程描述：  在主界面中点击“游戏设置”按钮，进入“游戏设置”界面；  在“游戏设置”界面中，在“背景设置”模块下，点击“显示背景”；  选择你要使用的图片；  点击“返回”按钮，用例终止。 |
| 前置条件：想要更换背景图片 |
| 后置条件：背景图片已被更改 |

表4–5背景设置用例

音乐设置用例

|  |
| --- |
| 用例编号：6 |
| 用例名：音乐设置 |
| 执行者：玩家 |
| 目的：更换背景音乐 |
| 过程描述：  在主界面中点击“游戏设置”按钮，进入“游戏设置”界面；  在“游戏设置”界面中，在“音乐设置”模块下，点击“播放音效”，进行音效设置；  点击“播放音乐”，根据需要选择是否要进行无序播放；  调整音量大小；  点击“返回”按钮，用例终止。 |
| 前置条件：想要更换背景音乐或调整音量、音效大小。 |
| 后置条件：音乐被更改，或音量被更改 |

表4–6音乐设置用例

提示路径用例

|  |
| --- |
| 用例编号：7 |
| 用例名：提示路径 |
| 执行者：玩家 |
| 目的：玩游戏期间，无法继续游戏时给玩家下一步操作提示 |
| 过程描述：  在正在运行游戏的游戏界面，点击“提示”按钮或F5键  “提示”后的数字为“0”或游戏结束，此用例终止； |
| 前置条件：需要路径提示 |
| 后置条件：游戏界面有下一部路径显示 |

表4–7提示路径用例

重新排序用例

|  |
| --- |
| 用例编号：8 |
| 用例名：重新排序 |
| 执行者：玩家/系统 |
| 目的：游戏期间，为了降低游戏难度/当前图没有可以消除的 |
| 过程描述：  在正在运行游戏的游戏界面，点击“生命”按钮或F7键  “生命”后的标识为“x0”或游戏结束，此用例终止； |
| 前置条件：想要更换当前页面的头像排序 |
| 后置条件：当前游戏页面的头像排序被更改 |

表4–8重新排序用例

## 模块设计

业务逻辑模块主要是对数据进行逻辑处理。主要可分为三大模块即欢迎模块、游戏模块、图形模块。

（1）欢迎模块

打开软件后，首先展示给用户的是一个欢迎动画，随后就是主界面，包含了“开始游戏”、“游戏设置”、“关于游戏”和“退出游戏”四大模块。

（2）游戏模块

玩家点击“开始游戏”后，进入游戏界面。

用户点击开始游戏后，系统开始倒计时，屏幕最上行有倒计时的时间进度条，第一关设置时间为1000s，之后每关分别为[1000-125\*(关数-1)]s，直到时间小于等于0s时，就把此关的游戏时间规定为100s。

玩家每消掉任何一对图形会出现线条连路方式出现以及显示消图效果发出消图声音并且时间+5s。玩家每消掉一对图形后得分为关数\*10分。

如果玩家在规定的时间内将所有的图形消掉，系统提示玩家：恭喜过关，并进入下一关。在系统所给的时间内没有消掉所有图形则将退出游戏，并提示：游戏结束！如果排行榜为空或者玩家的得分已经超过排行榜的得分，则会提示输入姓名，将得分保存并更新到排行榜上。

如果玩家自行选择重排，游戏剩余时间减去5\*（4+关数/2）秒。

如果游戏出现了死锁的情况（也就是说没有图形可以相消的情况），系统会提示玩家：没有可以消的方块了，请使用游戏道具“重排”，然而“重排”每一次的游戏中最多只能用两次，如果重列用完了之后还出玩死锁情况，系统便自动再次重新排列，直到不出现死锁。

（3）图形模块

图形模块是用来管理图形的。包括图形的图片，图形的随机排列，图形的显示和消除等。

## 连连看的算法设计

连连看是一款老少皆宜、百玩不厌的休闲类小游戏。各种连连看也层出不穷、五花八门，例如“果蔬连连看”、“宠物连连看”、“麻将连连看”等。各种“连连看”大同小异，无非就是界面图片不同，其核心算法应该是类似的。

### 连连看的游戏规则

连连看游戏的消除规则玩过了应该挺容易理解，就是两点之间只能用水平或者竖直线段连接，最多不能超过2个拐点，规则简单容易上手，游戏速度节奏快，画面清晰可爱，适合细心的玩家。根据规则可以消除的两个图的连线分成了4种情况：

1、 直接相连

2、 转一次弯相连

3、 Z型相连

4、 U型相连

### 地图生成算法设计与分析

我们设地图是一个 m \* n 二维数组，生成算法要求生成一个不总有连续标记出现的二维数组，数组中每个元素都要出现偶数次 。

分析：这个数组必须要有偶数个元素：所以 m\*n%2==0；

其他就是考虑出现偶数次数据和乱序排列的问题了。现在我们设 m=6，n=6进行讨论。并设现在数组中共有6个元素可供选择，不要求所有元素都出现，但如果出现必须出现偶数次。

方法一：先生成一个顺序排列的二位数组 如：

map[m][n]={{1,2,3,4,5,6},

{1,2,3,4,5,6},

{1,2,3,4,5,6},

{1,2,3,4,5,6},

{1,2,3,4,5,6},

{1,2,3,4,5,6},}

然后循环遍历数组元素和数组中任意位置的数交换，得到打乱顺序的数组。

map[m][n]={{1,4,2,5,6,2,}

{2,1,6,5,1,2,}

{1,2,3,5,3,4,}

{1,4,3,6,6,6,}

{4,6,3,5,3,5,}

{4,2,5,1,4,3,}}

评价： 此方法生成初始数组的过程简单， 但是初识变量需要人为规定好如何布局， 以确保有偶数个元素出现， 并且每个元素出现的次数固定， 对元素个数也有一定要求。 随机打乱后数组元素的离散度不高。

方法二：

1、 根据数组可用元素个数 x,先生成二维数组的前 m\*n-x 个元素。 这些元素的值随机取任意有效值（可让每个元素与前方，上方元素不等） 。

2、另开辟一长度 x 的一维数组 a[x]记录每个有效值出现的次数。

3、 根据 a 将出现奇数次的元素排列于二维数组末尾， 其余位置用随机一对儿相投元素填充。

4、将二维数组的最后 x 个元素与任意数组元素交换位置。

方法三：

1、先随机生成二维数组半数的元素，将其复制到二维数组的另一半。

2、遍历数组，随机交换数组中的元素。

评价： 方法二和方法三生成数组的元素个数随机度和元素的离散程度差不多， 但都要优于方法一。

方法四： 模板法， 顾名思义， 就是使用现成的模板。 这个不属于生成图的方法， 但是我认为有必要提一下他，因为我认为这才是最好的获得连连看地图的方法。因为此方法系统资源开销小，有利于控制关卡难度，地图随机度也好，简单易得，但模板的选取也要考虑种种问题，有一定难度。

上述前三种方法都没有考虑全图是否有解，第四种方法要看模板是否有解，下面我们再来研究一下如何保证全图有解。全图有解其实在真实的应用中的意义本人认为是不大的， 因为很多情况下有解的图在解图过程中也可能转变成无解图， 所以即使生成的图有解也不能保证游戏能顺利完成，所以目前遇到图无解采用的方法就是随机重排。以下讨论只做算法研究。

判断图是否有解最直接的方法就是解该张图， 如果发现无解就重新生成一张再解 。另一方法：

1、解图，将消去的元素按其位置不变存放于另一个空二维数组中，直到原图无解；

2、将原图元素限定在非空元素位置重排；

3、然后继续解图，将消去元素对应位置放于另一个数组中。

4、重复2、3直到原图消完。此时另一个数组中存放的即是一个有解图。

此方法可行性较高。

另一方法：生成有解图。根据寻路算法规律将一对对元素按消去规律放置在地图中（需要利用栈的回溯） 。此方法冗繁复杂，并且效果不好，可行性不高，再次就不深入讨论了。

### 图形连通算法设计

每次用户选择两个图形，如果图形满足一定条件(两个图形一样，且这两个图形之间存在转弯少于3的路径)，则两个图形都能消掉。给定任意具有相同图形的两个格子，我们需要寻找这两个格子之间在转弯最少的情况下，经过格子数目最少的路径。如果这个最优路径的转弯数目少于3 ，则这两个格子可以消去。

在检验两个方块能否消掉的时候，我们要让两个方块同时满足两个条件才行，就是两者配对并且连线成功。

分3种情况：

从下面的这三种情况，我们可以知道，需要三个检测，这三个检测分别检测一条直路经。这样就会有三条路径。若这三条路径上都是空按钮，那么就刚好是三种直线(两个转弯点)把两个按钮连接起来了。

1.相邻。直接连接消除。

2.若不相邻的先在第一个按钮的同行找一个空按钮。

（1）找到后看第二个按钮横向到这个空按钮所在的列是否有按钮；

（2）没有的话再看第一个按钮到与它同行的那个空按钮之间是否有按钮；

（3）没有的话，再从与第一个按钮同行的那个空按钮竖向到与第二个按钮的同行看是否有按钮。没有的话路径就通了，可以消了。

3.若2失败后，再在第一个按钮的同列找一个空按钮。

（1）找到后看第二个按钮竖向到这个空按钮所在的行是否有按钮；

（2）没有的话，再看第一个按钮到与它同列的那个空按钮之间是否有按钮；

（3）没有的话，再从与第一个按钮同列的那个空按钮横向到与第二个按钮同列看是否有按钮。没有的话路径就通了，可以消了。

若以上三步都失败，说明这两个按钮不可以消去。

## 本章小结

本章首先进行了用例设计，并深入分析了连连看的游戏规则、地图生成算法，连通消除算法，综合权衡之下，本系统采用生成有解图，游戏过程中无解再重排的算法。为编码实现打下坚实基础。

# 编码与实现

## 游戏初始化

进入游戏界面，首先需要循环播放背景音乐，其实现代码如下：

首先导入AVFoundation框架

#import <AVFoundation/AVFoundation.h>；

- (void)initSound

{

NSString \*musicFilePath = [[NSBundle mainBundle] pathForResource:@"bgMusic" ofType:@"mp3"];

NSURL \*musicURL = [[NSURL alloc] initFileURLWithPath:musicFilePath];

AVAudioPlayer \*thePlayer = [[AVAudioPlayer alloc] initWithContentsOfURL:musicURL error:nil];

// 创建播放器

self.myBackMusic = thePlayer; //赋值给自己定义的类变量

[musicURL release];

[thePlayer release];

[myBackMusic prepareToPlay];

[myBackMusic setVolume:1]; //设置音量大小

myBackMusic.numberOfLoops = -1;//设置音乐播放次数 -1为一直循环

if (mainMusicStatus)

{

[myBackMusic play]; //播放

}

}

然后开始显示显示出来图片矩阵,如果从第一关开始，把地图设置成4行4列，之后第二关和第三关，就把地图设置成6行6列、8行8列，其中8行8列的地图生成核心代码如下：

static int imgMap[64] = {

1 , 1 , 2 , 2 , 3 , 3 , 4 , 4 ,

5 , 5 , 5 , 5 , 6 , 6 , 0 , 0 ,

7 , 7 , 7 , 7 , 8 , 8 , 0 , 0 ,

9 , 9 , 9 , 9 , 10, 10, 10, 10,

11, 11, 11, 11, 12, 12, 12, 12,

13, 13, 13, 13, 14, 14, 14, 14,

15, 15, 16, 16, 0 , 0 , 0 , 0 ,

0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 ,

};

- (void)initData

{

prePoint = CGPointMake(-1, -1);

countCleared = 0;

counter = 0;

NSMutableArray \*array = [[NSMutableArray alloc] init];

srandom((unsigned int)time(nil));

for (int i = 0; i < (TOTAL\_X - 2) \* (TOTAL\_Y - 2); ++i) {

MapNode \*mapnode = [[MapNode alloc] init];

mapnode.order = (int)(CCRANDOM\_0\_1() \* NSIntegerMax) % (int)(CCRANDOM\_0\_1() \* NSIntegerMax);

mapnode.imgid = imgMap[i];

[array addObject:mapnode];

}

NSArray \*sortarray = [array sortedArrayUsingSelector:@selector(myCompare:)];;

arrayMap = [[NSMutableArray alloc] init];

for (int x = 0; x < TOTAL\_X; ++x) {

for (int y = 0; y < TOTAL\_Y; ++y) {

if (x == 0 || x == (TOTAL\_X - 1) || y == 0 || (y == TOTAL\_Y - 1)) {

MapNode \*mapnode = [[MapNode alloc] init];

mapnode.order = 0;

mapnode.imgid = 0;

[arrayMap addObject:mapnode];

} else {

int i = (y - 1) \* (TOTAL\_Y - 2) + x - 1;

[arrayMap addObject:[sortarray objectAtIndex:i]];

}

}

}

}

## 判断消除图片

消除图片时要判断是否可达，即优化判断是否可以消除的算法是本程序的灵魂，连连看游戏分为三种消去方式，分别为“0折型”，“一折型”，“两折型”，由图4-5可以看出，如果两点间只需要一条直线能够连接起来，则A、B两点的横坐标或纵坐标必定相同，有了这个条件，我们判断 A、B两点是否只需要一条直接连接就简单了许多。

其分析如下：

1.“0折型”：

水平消去或者垂直消去，如果两个点击的两个图片相同，并且它们在同一水平或垂直方向上，即返回H\_LINE的画线类型，代码如下：

- (BOOL)match\_direct:(CGPoint)a other:(CGPoint)b

{

if (!(a.x == b.x || a.y == b.y)) {

return NO;

}

int i;

BOOL match\_x = NO;

if(a.x == b.x) {

match\_x = YES;

if(a.y > b.y) {

for(i = a.y - 1; i > b.y; --i) {

CGPoint point = CGPointMake(a.x, i);

if(![self isValiableNode:point] ||

![self isEmptyNode:point]) {

match\_x = NO;

}

}

}

if(b.y > a.y) {

for(i = b.y - 1; i > a.y; --i) {

CGPoint point = CGPointMake(a.x, i);

if(![self isValiableNode:point] ||

![self isEmptyNode:point]) {

match\_x = NO;

}

}

}

}

BOOL match\_y = NO;

if(a.y == b.y) {

match\_y = YES;

if(a.x > b.x) {

for(i = a.x - 1; i > b.x; --i) {

CGPoint point = CGPointMake(i, a.y);

if(![self isValiableNode:point] ||

![self isEmptyNode:point]) {

match\_y = NO;

}

}

}

if(b.x > a.x) {

for(i = b.x - 1; i > a.x; --i) {

CGPoint point = CGPointMake(i, a.y);

if(![self isValiableNode:point] ||

![self isEmptyNode:point]) {

match\_y = NO;

}

}

}

}

return match\_x || match\_y;

}

2.“一折型”，当点击的两个点相同，它们不在同一水平或垂直方向上，并且它们要用

两条直线相连，就返回ONE\_C\_LINE，代码如：

- (BOOL)match\_one\_corner:(CGPoint)a other:(CGPoint)b

{

CGPoint point = CGPointMake(b.x, a.y);

if([self isValiableNode:point] &&

[self isEmptyNode:point] &&

[self match\_direct:a other:point] &&

[self match\_direct:b other:point]) {

return YES;

}

point = CGPointMake(a.x, b.y);

if([self isValiableNode:point] &&

[self isEmptyNode:point] &&

[self match\_direct:a other:point] &&

[self match\_direct:b other:point]) {

return YES;

}

return NO;

}

3.“两折型”,当点击的两个点相同，它们不在同一水平或垂直方向上，并且它们要用三条直线相连，就返回TWO\_C\_LINE，代码如下：

- (BOOL)match\_two\_corner:(CGPoint)a other:(CGPoint)b

{

for(int i = a.x - 1; i >= 0; --i) {

CGPoint point = CGPointMake(i, a.y);

if(![self isValiableNode:point] || ![self isEmptyNode:point]) {

break;

} else {

if([self match\_one\_corner:point other:b]) {

return YES;

}

}

}

for(int i = a.x + 1; i < TOTAL\_X; ++i) {

CGPoint point = CGPointMake(i, a.y);

if(![self isValiableNode:point] || ![self isEmptyNode:point]) {

break;

} else {

if([self match\_one\_corner:point other:b]) {

return YES;

}

}

}

for(int i = a.y - 1; i >= 0; --i) {

CGPoint point = CGPointMake(a.x ,i);

if(![self isValiableNode:point] || ![self isEmptyNode:point]) {

break;

} else {

if([self match\_one\_corner:point other:b]) {

return YES;

}

}

}

for(int i = a.y + 1; i < TOTAL\_Y; ++i) {

CGPoint point = CGPointMake(a.x ,i);

if(![self isValiableNode:point] || ![self isEmptyNode:point]) {

break;

} else {

if([self match\_one\_corner:point other:b]) {

return YES;

}

}

}

return NO;

}

综合以上判断能否消除的条件，满足其中之一即为可以消除否则不可消除，具体代码如下：

- (BOOL)match:(CGPoint)a other:(CGPoint)b

{

if([self match\_direct:a other:b]) {

return YES;

}

if([self match\_one\_corner:a other:b]) {

return YES;

}

if([self match\_two\_corner:a other:b]) {

return YES;

}

return NO;

}

## 本章小结

本章通过对连连看游戏的各个功能的算法实现进行叙述，描述了系统的主要算法功能，展示了连连看算法的核心实现代码，全面深刻地解析了该系统所具有的核心功能及其实现的方法。对每个算法设计以及思路进行了详细的解释。

# 系统测试与运行

## 软件测试简介

测试是软件开发过程中的重要组成部分，是用来确认一个程序的品质或性能是否符合开发之前所提出的一下要求的过程。软件测试是发现软件中错误和缺陷的主要手段。在一般情况下，软件测试过程与整个软件开发过程基本上市平行进行的。当然，测试计划应该在需求分析阶段就已经开始制定了。随后的工作则会伴随着软件开发的过程逐步展开。缺陷在开发和维护的任何阶段都可能发生，并有此产生一个或多个bug，包括错误、误解和冗余，有时候甚至会误导开发者。所以软件测试是发现并指出软件系统缺陷的过程。测试包括寻找缺陷，但不包括跟踪漏洞及修复。

软件测试是软件工程过程的一个重要阶段，是在软件投入运行前，对软件需求分析、设计和编码个阶段产品的最终检验，是为了保证软件开发产品的正确性、完全性和一致性，从而检测软件错误、修正软件错误的过程。软件开发的目的是开放出实现用户需求的高质量、高性能的软件产品，软件测试已检查软件产品内容和功能特性为核心，是软件指令保证的关键不走，也是成功实现软件开发目标的重要保证。

## 测试步骤

系统测试过程按4个步骤进行，即：单元测试、组装测试、确认测试、系统测试。

1.首先，进行单元测试

单元测试集中对用源代码实现的每一个程序单元进行测试，检查各个程序模块是否正确地实现了规定的功能。

2.然后，进行集成测试

根据设计规定的软件体系结构，把已测试过的模块组装起来，在组装过程中，检查程序结构组装的正确性。

3.接着，进行确认测试

确认测试是检查已实现的软件是否满足了需求分析确定了的各种需求，以及软件配置是否完全、正确。

4.最后，是系统测试

系统测试把已经经过确认的软件纳入实际运行环境中，与其他系统成分组合在一起进行测试。严格地说，系统测试已经超出了软件工程的范围。而在实际工作中这几个阶段是相互交叉的。

模块检测是检验设计的最小单元。主要根据详细设计为指南，对重要的执行进行测试。对每个功能模块的测试过程中主要检查表单的设计、检查试图的设计等。

## 游戏系统测试

1.测试环境：iOS7模拟器

2.对程序单个模块进行测试：

（1）开始游戏模块：

测试方法：点击开始游戏，运行游戏。

测试结果：实现了游戏初始化，重新开始，重新排列和退出等功能。

（2）游戏设置模块：

测试方法：点击“游戏设置”，进行测试。

测试结果：实现了对游戏的声音控制、震动控制和选关控制。

（3）关于游戏模块：

测试方法：点击“关于游戏”，进行测试。

测试结果：实现了游戏的排行榜的显示、排行榜的清除和游戏的帮助。

（4）退出游戏模块：

测试方法：点击“退出”，进行测试。

测试结果：实现了游戏的退出功能。

3.对程序整个模块进行测试：

测试方法：用Xcode打开程序，选择iOS7模拟器，点击运行。打开游戏，运行程序进行测试。

测试结果：实现了界面优化，实现了开始游戏，游戏设置，关于游戏和退出游戏等功能；实现了连连看的基本功能。

## 游戏性能分析

本系统在测试和运行过程中。有以下几方面有点：

1.界面友好，操作方便

系统界面简单，功能较完善，易上手，可以激发玩家的乐趣。

2.通用性好，扩展性强

本系统可以适用于各种人群，具有优越的通用性能。

3.部分系统功能需完善

由于时间和技术因素，本系统缺少“提示”等功能，有待于进一步的解决。

## 本章小结

在系统测试中，发现了不少问题，并且对其进行了及时的修正。

总体上，本次系统的设计与开发实现了最初的任务要求，进行了游戏的系统测试和性能上的分析，达到了需求分析中的功能要求，实现了连连看游戏的设计。以上标志着本次基于iOS7的连连看游戏的设计与实现工作成功的完成。

结束语

本连连看游戏实现了连连看的基本游戏功能，本文主要内容分为课程设计分析、需求分析、程序设计和代码实现四个部分，给出了基于iOS7 Sprite Kit连连看游戏的实现过程。本研究成果可以在模拟器上运行，具有重新开始和重新排列等功能，并且通过在模拟器上测试，基本达到了预期目标。由于时间原因，和自己的能力等问题，不足之处是没有设置时间限制。该技术还需要进一步的完善，需要在日后的工作中不断地探索和研究，完善本程序。

通过本游戏的开发，达到了学习iOS技术和熟悉软件开发流程，熟悉iOS应用程序开发流程的目的。IOS系统手机在中国已经火热流行开来，设计本课题通过查阅相关资料可以促进自己了解更多IOS手机前沿信息，理清软件开发流程，进一步加固了对Java知识的运用。对以后致力于IOS应用程序开发奠定了基础。

通过本科阶段的学习研究，论文取得了一定成绩，但依然任重道远，尚有许多研究工作有待进一步深入学习和时间这里则其要者简要讨论如下：

1．随着科技进步，手机游戏也不断发展和更新，先进的游戏模式和技术不断支撑着手机游戏的发展和科技探新，因此要加强IOS理论学习，指导实践研究。

2．由于作者研究水平的不足和技术条件限制，系统某些功能还不够完善，有待于进一步研究与探索。

参考文献

1. 马克(David Mark)(美).Jack Nutting (美).Jeff LaMarche (美).Fredrik Olsson (美). 邓强 (译). 武海峰 (译). 精通iOS开发[M].北京：人民邮电出版社，2013：42-50.
2. 谷明雨.谢慧华.陈芷若.指尖的疯狂 不得不玩的100款游戏[J].移动信息，2012,6(6):64-67.
3. 代汝.智商大挑战.Android益智游戏TOP10[J].电脑爱好者，2013,7(15):34-36.
4. 克纳斯特 (Scott Knaster)(美).马利克 (Waqar Malik)(美). 达尔林普尔 (Mark Dalrymple)(美).庆成译.Objective-C基础教程[M].北京：人民邮电出版社，2013：21-30.
5. 刘乐廷.李敬兆.IOS内存开发管理机制的研究[J].计算机与现代化，2013,3(15):14-16.
6. 熊斌(中).IOS开发从入门到精通.[M].北京：北京希望电子出版社，2013：21-30.
7. 刘乐廷.李敬兆.IOS内存开发管理机制的研究[J].计算机与现代化，2013,3(15):14-16.
8. 刘杰逾.基于iOS平台的一款游戏设计与实现[D].成都：电子科技大学，2012：32-34.
9. 萨丹(Erica Sadun)(美).张彩霞 (译).高颖(译)等.iPhone开发秘籍[M].北京：人民邮电出版社，2013：52-55.
10. 王云.IOS平台客户端应用开发规范化的研究[D].北京：北京邮电大学，2013：12-13.
11. 苍梧.连连看算法及源代码［EB/OL］.http://www.cajcd.edu.cn/pub/wml.html， 2010-06-06 /2011-02-24.
12. Aaron Hillegass(美).李骏(译).夏伟频 (译)Objective-C编程[M].武汉：华中科技大学出版社，2012：52-55.
13. 蔡杰.基于iPhone OS的图形计算器的研究与应用[D].成都：电子科技大学，2013：32-34.
14. 万年水.连连看算法分析［EB/OL］.http://bsong.blog.163.com/blog/static/1870204512012111675426660/， 2012-12-16.
15. 诺伊堡 (作者), 陈建设 (译者), 刘洋 (译者), 薛晓东 (译者), 李明 (译).iOS程序设计(第2版)[M].北京：机械工业出版社，2013：23-25.
16. 谢蛟.智能手机游戏 UI设计的研究[C].武汉：中南民族大学，2012，46-47.

附录

致谢

大学的学习和生活以此论文画上了一个圆满的句号，大学四年中经历了快乐，坚信，分享，喜悦，成功，失败，感激……感谢在四年的时间里每一位对我谆谆教诲的老师，感谢每一位在我遇到困难时帮助我的朋友们。本次毕业设计与论文是在许豪老师的悉心指导下完成的，许老师从论文选题、论文设计到系统成型都细心指导，并给我提出了许多建设性的修改意见，使我的论文得以顺利完成。他严肃的科学态度，严谨的治学精神，深深地感染和激励着我，使我不仅增强了自学的能力和开发软件的能力，而且对一些软件开发模式以及软件工程思想有了更加深刻的认识，技术也有所提高，使我的社会竞争力变得更加强大。感谢他在本人学习期间给我的关心和鼓励，以及老师精益求精的工作作风，宽以待人的崇高风范，朴实无华、平易近人的人格魅力对我影响深远。

感谢同班的所有同学们，正是你们在学习和生活中的无私的关心和帮助，我才得以顺利地完成四年的学习，难得的学习生涯给我留下了无尽的回忆。祝他们事业有成！

我要用自己的不断努力回报社会，回报关心我、帮助过我的每一个人，我要积极努力工作，做一个全面发展的对社会有用的人才。

由于技术有限、经验匮乏，难免有许多考虑不周全、做得不完整的地方，望各位老师给予建议与指正。谢谢老师！