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： 贺弘博

指导教师： 曹蕾

完成日期 2015 年 04 月

南阳理工学院本科生毕业设计（论文）

基于IOS的“一个不能死”游戏的  
设计与实现

Design and Implementation of the Game named  
”No One Died”based on the IOS Platform

总 计：毕业设计(论文) 5页

表 格： 1个

图 片： 1个

南 阳 理 工 学 院 本 科 毕 业 设 计(论文)

基于IOS的“一个不能死”游戏的  
设计与实现

Design and Implementation of the Game named  
”No One Died”based on the IOS Platform

学 院(系)： 软件学院

专 业： 软件工程

学 生 姓 名： 贺弘博

学 号： 1115115664

指导教师(职称)： 曹蕾 副教授

评 阅 教 师： 曹蕾

完 成 日 期： 2015年04月02日

南阳理工学院

Nanyang Institute of Technology

基于IOS的“一个不能死”游戏的  
设计与实现

软件工程 贺弘博

1. 当今技术发展的今天，手机终端称为人们茶余饭后休闲的必需品。手机在人们心目中已经不再是那种单纯的通讯工具，它是集合了各种社交娱乐与一身的新时代终端。手机游戏在如今被人们当作一种“姿势”，用来放松精神，消遣时间。人们对手机游戏的关注度也日益提高。怎样让有游戏能够更加容易的被各年龄段玩家所接受，成为了游戏设计的主要课题。本次课程设计旨在制作出一款容易上手，开启率高，不易被删除的益智类游戏。该游戏软件基于iOS手机系统开发；主要语言为Objective-C；游戏代码设计采用KVO设计模式；工程中所采用的主要框架SpringKit是基于苹果iOS官方SDK的上层框架，具有良好的事件侦听能力与物理模拟能力。相对于其它同类的循环挑战类游戏相比，“一个也不能死”游戏风格明朗简洁，规则简单易懂，游戏过程中不需太复杂的操作，游戏过程不需要网络连接。相信这款游戏能够带给玩家愉快的体验，通过微博将自己的快乐的心情分享给周围的人们。
2. 一个不能死；手游；iOS；Objective-C；Xcode；SpringKit

Design and Implementation of the Game named  
”No One Died”based on the IOS Platform

Software Engineering Major He Hongbo

1. No One Died；hand travel；iOS；Objectiv-C；Xcode；SpringKit

目 录

[1 引言 1](#_Toc417857985)

[1.1 国内外研究现状 1](#_Toc417857986)

[1.2 选题背景 1](#_Toc417857987)

[1.3 选题意义 2](#_Toc417857988)

[1.4 论文的组织形式 2](#_Toc417857989)

[1.5 本章小结 3](#_Toc417857990)

[2 技术简介 3](#_Toc417857991)

[2.1 iOS简介 3](#_Toc417857992)

[2.1.1 Objective-C简介 3](#_Toc417857993)

[2.1.2 SpringKit简介 4](#_Toc417857994)

[2.1.3 Xcode简介 4](#_Toc417857995)

[2.2 KVC设计模式 5](#_Toc417857996)

[2.3 研究方法及手段 6](#_Toc417857997)

[2.4 本章小结 6](#_Toc417857998)

[3 需求分析 6](#_Toc417857999)

[3.1 可行性分析 7](#_Toc417858000)

[3.2 需求描述 7](#_Toc417858001)

[3.2.2 微博分享 8](#_Toc417858002)

[3.3 功能事件流分析 8](#_Toc417858003)

[3.3.1 主界面用例 8](#_Toc417858004)

[3.3.2 游戏界面用例 9](#_Toc417858005)

[3.3.3 结果界面用例 10](#_Toc417858006)

[3.3.4 分享界面用例 10](#_Toc417858007)

[3.4 功能分析 11](#_Toc417858008)

[3.5 本章小结 12](#_Toc417858009)

[4 系统设计 12](#_Toc417858010)

[4.2 概要设计 13](#_Toc417858011)

[4.3 详细设计 13](#_Toc417858012)

[4.3.1 系统总流程 13](#_Toc417858013)

[4.3.2 系统各个模块的处理流程 13](#_Toc417858014)

[4.4 本章小结 13](#_Toc417858015)

[5 系统实现 13](#_Toc417858016)

[5.1 节点特效的实现 13](#_Toc417858017)

[5.2 点击事件的实现 13](#_Toc417858018)

[5.3 主界面的设计 14](#_Toc417858019)

[5.4 游戏界面的设计 14](#_Toc417858020)

[5.5 成绩界面的设计 15](#_Toc417858021)

[5.6 声音系统的实现 15](#_Toc417858022)

[5.7 分享到新浪微博 16](#_Toc417858023)

[5.8 存储 16](#_Toc417858024)

[6 测试 16](#_Toc417858025)

[6.1 程序的测试 16](#_Toc417858026)

[6.1.1 测试的重要性及目的 16](#_Toc417858027)

[6.1.2 测试的步骤 17](#_Toc417858028)

[6.1.3 测试时出现的问题 18](#_Toc417858029)

[结束语 19](#_Toc417858030)

[参考文献 20](#_Toc417858031)

[附录 21](#_Toc417858032)

[致谢 22](#_Toc417858033)

# 引言

IT行业已经不仅是“逆水行舟”的状况，如果不能第一时间掌握前沿技术动态，就无法保证工程的优秀质量。在竞争的时期，使用最新的技术做出符合大众当前需求、贴近日常的作品显得尤为可贵。

游戏可以缓解高压工作人群的精神压力。对于这一类社会中流砥柱人群，拿到这款简单的循环挑战类游戏稍作闲暇，娱乐一下，放松心情。对于我自己，使用SpringKit这一最新的技术构建出自己的作品，既能满足个人的好奇心，又能够获得足够的经验。在未来的技术交流中，能够更好地与大家探讨最新的终端科技。

## 国内外研究现状

根据互联网市场数据供应商Netmarketshare曾发布数字报告，苹果的iPhone OS各项性能已经超越Linux操作系统。最新新数字证明出苹果系统不得不令人信服的强大，它优秀的体验与低消耗毫无疑问是行业内的经典。更重要的是，苹果的操作系统的载体是苹果旗下开发团队的iPhone、iPad和iPod Touch等产品设备。苹果在硬件上同样证明了自己的硬件同样拥有技术储备。

另一方面，linux以为其开发成本小，可靠性高，在服务器开发市场上占有不可动摇的份额，其安全性令人放心。但在掌上设备市场，即使计算谷歌的基于Linux的Android系统，总市场占有率仍低于苹果的iOS。

日常我们所能接触的终端游戏开发技术无疑集中在Cocos2d-iphone、Cocos2d-x、Unity 3D、Adobe Flash Air、 HTML 5当中。特别是Cocos2d所开发的2D游戏几乎占据了终端游戏收入排行榜的80%以上，几乎所有的安卓手游都会或多或少用到。Unity3d则是更多的出现在大型3D建模中。极少数资本开发商使用unreal引擎开发游戏。国外还有少数开发者用Corona, Game Salad等引擎或工具，但充其量也只是占据了少数的业余爱好者市场，更谈不上主流。

也许苹果公司在为自己的居家设备品牌化计划做最后的挑战，在iOS 7的变化中，全新的Sprite Kit Framework和Game Controller Framework可谓是提供给游戏开发者的屠龙宝刀。

## 选题背景

前些时期在游戏排行榜瞬间飙升榜首的益智类游戏中，绝大多数为简单的循环类游戏。他们之所以能够走红，是因为操作的简单，界面设计明朗愉快。

在这个眼球游戏横行的年代，通过画质和复杂的游戏构建来吸引用户，已经成为了过去式。我们所能做到的是在游戏创新上给人眼前一亮的感觉，如同《魂斗罗》、《超级玛丽奥》、《俄罗斯方块》等小巧的经典游戏反而被用户下载试玩。这足够说明并不是耗费大量人力物力做出的游戏才能被人们所欢迎，游戏在于创意与体验。

如今遍布大街小巷的人群在使用着iOS系统，应为它的画面效果优质，内核运行丝毫没有卡顿的现象。使得Objective-C现如今成为首屈一指的手机终端编程语言，使用OC的编程的开发者乘着强劲的上升趋势，逐年攀升。完美的渲染机制摒弃了图片的完整加载，使得程序内存有足够的空间让开发者发挥。相比java、javascrip等，程序员更加希望使用能够展现创造力的语言，而不是苦于内存管理这种“体力活”。

## 选题意义

对于拿出手机的动作已经见怪不怪的当今，人们在特定场合拿出手机无非是为了娱乐。相较于电脑操作，人们将更多的时间放在了手机操作上，甚至有人使用手机玩游戏到通宵。游戏能够令人调整心情，同样能够令人沉迷。人们在手游上面花费的时间远远高于其他软件，一款受欢迎的游戏可以带来各种收益，可以说制作游戏的热潮已经开始。例如前不久的《FlappyBird》、《一个都不能死》、《2048》。在AppStore游戏排行榜长时间独占前十位，这种单纯考验人们反应能力的益智类游戏成为人们关注的焦点。只是因为它们美观、简洁、易操作，并且所占用空间极其小。

在学习了Objective-C之后，一直想要做出作品来。在查阅了大量资料后，整理出有用的资源作为我的课题基础，这一过程我学到了许多之前未曾涉足的领域。对于本次课题设计，我就选择原型《一个都不能死》的软件开发。

希望通过本次设计进一步提高自我的编程水平，强化自己的优势，发现自身的不足。

## 论文的组织形式

本文分为六章：

第一章是引言，主要阐述国内为研究现状以及选题的背景及意义进行分析，从而得出本系统的开发的意义与价值。

第二章是对该游戏用到的技术的简介，主要是说明本次毕业设计用到什么样的技术，对本游戏有所认识。

第三章是对游戏的需求分析，在方便用户理解其功能的同时有助于开发者对系统的设计与实现；

第四章是游戏设计，本章详细的介绍的，本游戏制作的流程与实现的设计思想。

第五章是游戏实现过程，主要通过重要功能的技术详细讲解和运行代码还有界面运行效果图来详细介绍本游戏。

第六章是游戏的测试，主要阐述了，游戏测试的目的，和游戏测试方法。

## 本章小结

本章针对该课题的国内外研究现状、研究背景和意义、论文的主要等做了简单的介绍，为了便于阅读，介绍了以上国内外的技术实情。当我们想要进一强化自己的编程水准时，需要关注国外的技术动向，以及国内的各种论坛。不同的观点与大量的知识汇聚起来才是最接近完美的技术。只有这样才可以更好的发挥编程的价值。

# 技术简介

该课题基于iOS系统开发，使用的是Xcode编译工具，涉及框架SpringKit以及其他第三方封装的功能框架，游戏代码采用KVC设计模式。下面对技术进行介绍。

## iOS简介

2007年1月9日，美国苹果公司在Macworld大会上发布了苹果iPhone OS,苹果设计之初是为了iPhone需要搭载移动操作系统，后来逐渐拓展到其他的苹果设备上，例如iPad、appleTV等苹果产品。



图

iOS有许多地方与Mac OS X操作系统相似的地方，他们同样使用了Darwin原始码，相似与Unix的商业操作系统。随后，苹果公司在2008年3月6日正式发布iPhone OS，并在2010年6月7日召开的WWDC 2010定义iPhoneOS为“iOS”。第一代iOS操作系统名为“iOS4”，至目前已经连续更新发布之iOS 6.3，在安全性与用户体验上iOS一直走在世界的最前沿。

### Objective-C简介

OC（Objective-C）是一种程式语言，它具有通用、高阶、面向对象等特点。OC扩充了标准ANSIC语言，同时加入了Smalltalk 式的讯息传导机制。目前支持的编译器为GCC和LLVM，前端为Clang.

苹果公司将Objective-C注册为商标，并且在开发NeXTSTEP系统时被使用，之后被封装集成到OS X和iOS上，目前二者的API、Cocoa和Cocoa Touch使用OC为主要语言。

### SpringKit简介

在看过官方文档和视频之后了解到，Sprite Kit分为三部分， Scene（场景）、Actions（动作）和Physics（物理）。Scene指的是游戏场景及其中的节点，背景等；Actions则是对象化的事件特效，例如游戏角色的动作动画特效音效等；而Physics独立封装了物理世界的各种参量。

SKNode对于Sprite Kit的重要性如同CCNode对于cocos2d的意义一样。cocos2d的游戏场景可以用场景节点图的形式来展现，而节点同样是Sprite Kit游戏场景的根基。

Sprite Kit游戏中的所有视觉元素均使用SKNode的子类来绘制。

SKNode的几大子类包括:

SKSpriteNode(对应CCSprite，用于绘制精灵纹理);

SKVideoNode（在cocos2d中无对应类，用于播放视频）；

SKLabelNode(对应CCLabel，用于渲染文本）；

SKShapeNode(在cocos2d中无对应类，用于渲染基于Core Graphics路径的形状）；

SKEmitterNode（对应CCParticleSystem，用于创建和渲染粒子系统);

SKCropNode(无对应类，用于使用遮罩来裁剪子节点）；

SKEffectNode（无对应类\*，用于在子节点上使用Core Image滤镜）。

用节点树的形式来组织，这一点和iOS的视图层级也有相似之处。通常情况下，在一个游戏场景中，SKScene（场景节点）将作为根节点存在，而其它的内容节点都是其子节点。场景节点将负责运行一个动画循环，从而来处理其它子节点的动作，模拟物理世界，并将节点数中的内容渲染到屏幕上。

节点树中的每一个节点都为其子节点提供了一套坐标体系。当我们将某个子节点添加到节点树中时，可以通过设置其position和zPosition属性将其放置在父节点的坐标体系中。可以通过xScale,yScale和zRotation等属性来缩放和旋转节点。当某个节点的坐标体系被缩放或旋转时，其自身内容和所有子节点的内容也会发生同样的变化。

### Xcode简介

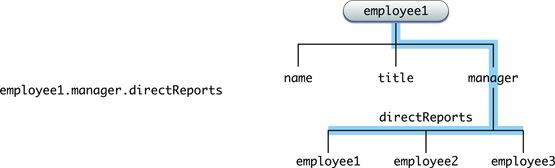
Xcode是苹果官方向开发人员提供的集成开发环境（非开源），是用来开发搭载于Mac OS X、iOS的App应用程序。 他的前身就是NeXT源码的Project Builder，开发于苹果公司的Mac操作系统下。目前Xcode 编译器采用LLVM编译器，前端为clang。它是苹果开发人员建立OS X 和App应用程序的最快捷方式，它包含了个中编译工具，支持多数语言同时编译。

## KVC设计模式

一提到设计模式，不得不说当前最热门的几种设计模式。在软件工程中，设计模式的重要性无疑是每一个程序员需要了解的。只有明确的设计模式，才不会产生逻辑“炸弹”。当我们要对一个完整的作品扩展修改时，设计模式显得尤为重要。

#### KVC(Key-Value-Coding)

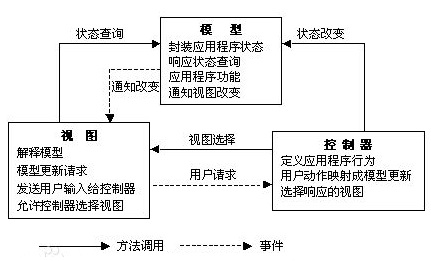
KVC设计模式指的是通过键值编码来访问你要存取的类的属性，其特点是通过指定表示要访问的属性名字的字符串标识符，可以进行类的属性读取和设置。 KVO是键值观察，特点是利用键值观察可以注册成为一个对象的观察者，在该对象的某个属性变化时收到通知 ，从而响应指定定的功能函数。



图

#### MVC（Model View Controller）

MVC由模型、视图、控制器3部分组成。模型：保存应用程序数据的类，处理业务逻辑的类；视图：即窗口，以及各种控件和其他用户能看到的并且能交互的元素；控制器：是最为重要的逻辑关口，也是表层控制，绑定了模型和视图，控制如何处理用户输入的类。



图

## 研究方法及手段

本课题使用Xcode为开发工具，在对OC十分了解的基础之上，深入发掘Xcode的编程方法。大量查找相关实例作品作为参考，制作出一款逻辑完整、程序运行流畅的游戏。制作过程以及游戏实现后的整体过程注意以下几项：

（1）掌握Xcode编译器，熟练掌握相关基本操作，通过官方文档熟悉每个功能模块与控件之间的关联。作为编程的工具，必须能够熟练操作它。

（2）明确需求分析，严格参照系统设计之初进行代码编撰。

（3）随时标注代码。严谨的代码风格固然能够书写上乘的代码，然而随着代码量堆积，没有注释的代码只能成为绊脚石。随时添加注释既能起到说明的作用，又能作为代码的备忘录使用，使得代码进度明确可读。

（4）断点（break point）测试。断点测试是程序员修改bug的必需功能，它让程序员精确分析函数之间的参量交互。程序报错起因往往是不经意写下的错误参数上。断点就是用来让程序员有时间检查代码的运行过程。

（5）根据最后实现的游戏进行论文的撰写。总结编程过程中的问题，在论文中写出技术问题，以及心得体会；备份核心代码，以及软件测试中遇到的问题以及解决方案的注释；撰写心得和致谢等内容。

## 本章小结

本章概述了完成基于iOS平台“一个不能死”游戏设计与实现所需的技术。游戏制作之初，我们要将游戏所用到的技术关联罗列出来，并对它们有一定程度上的掌握，只有这样才能够高效高质量地完成本次课题。

# 需求分析

需求(Requirement)是待开发系统的特征。需求分析(Requirements Analysis)就是开发人员着手分析即将编写的软件系统用来解决手头问题所需要的软件的特征。需求分析由一系列活动构成，其中包括各种需要解决方案的问题的讨论；确定系统解决方向必须具有的行为。

软件需求分析阶段的工作是在对问题进行调查了解的基础上的，用一定的方法和手段对问题进行分析建模。需求分析的结果应该映射出的必须实现什么，而不是怎么实现。它的主要用职能是明确需求，为用户和开发人员提供一起协商讨论的依据并作为设计和实现的蓝图。

本课题主要针对于iOS系统的苹果手机终端开发的游戏设计分析与实现，其分析主要包括对可行性分析、需求描述、面向对象的分析、主要功能的分析以及相关的数据的分析等，通过需求分析为下一阶段开发设计制作蓝图，使设计人员熟悉设计种需要实现的功能。

## 可行性分析

出于各方面的考虑，软件的制作设计多数个人或群体的利益。需要讨论此软件制作后被人们所接受的情形，本课题设计的可行性分析如下：

#### 经济可行性分析

该课题的系统成熟可靠，上手门槛不高，并且拥有成熟SDK文档，实用性可以保证，降低了学习成本。综上所述，该课题具有经济可行性。

#### 技术可行性分析

该课题在基于苹果平台的无限循环类游戏软件的设计与开发，所需苹果系统iMac开发平台，支持iOS7.0以及以上的，需要在苹果官方SDK的支持下开发，使用的IDE是xcode6.0或更高的版本，并在真机上机上进行设计开发与调试。objective－C是一门面向对象的语言，其UI界面也不例外，基于UIView基础之上的SpringKit框架，同样在功能上有明确的划分。该课题采用独立二进制文件归档，并且有iOS沙河机制保证数据的高效完整性，系统的整体内存管理由ARC机制直接保证。综上所述，该项目具有技术可行性

#### 操作可行性分析

游戏采用可视化界面，界面简洁明了，与玩家的交互触屏操作即可完成，对任何用户都能快速上手使用。综上所述，该项目具有操作可行性。

#### 法律可行性分析

游戏在开发过程中没有使用商业级代码，全部代码由本人独自完成。所使用到的框架未涉及第三方软件，不存在版权侵权问题以及与法律相抵触的情况。综上所述，该课题具有法律可行性

## 需求描述

通过对用户需求进行分析，本系统软件主要的功能分为四大方面。

#### 游戏界面功能

玩家一开始进入的主界面可以直接选择三种难度来挑战；主界面的小鸟被点击后会跳转到疯狂模式，反之亦然；主界面的三种难度（文章中称其为普通模式）与疯狂模式的三种难度操作方法相同，点击即刻开始游戏；游戏场景操作简单，有明确的计时和暂停按钮，暂停游戏后可以选择返回主界面或者继续游戏，返回主界面将丢失该挑战的游戏成绩；模式主界面都有微博分享按钮，点击后进入。

#### 游戏控制功能

游戏采用iOS SDK系统自带的计时器来控制图片的刷新，以及成绩的计算等；游戏的成绩是根据游戏图片的刷新频率计算得来的耗时的值；玩家的成绩在当前场景挑战结束时与本地纪录作比较，留存最高值作为当前场景难度的最高的成绩。

#### 游戏操作功能

在游戏中，玩家只需要点击小鸡所在的那一行，小鸡便会跳起，用来躲避随机飞过来的障碍物；障碍物只会出现在当前行的小鸡水平面上；随机出现的障碍物有不同的规格，玩家要根据其规格大小判断躲避的时机；疯狂模式中的所有难度都添加了一种特殊规格的障碍物，躲避难度较高。

#### 游戏分数更新

利用文件归档来存储最新的游戏成绩，每次存储取好的成绩作为了历史最优，并显示在结果界面上。

### 微博分享

采用新浪微博API利用OAuth2协议获得用户的授权后，允许我方软件发出微博。分享结果界面的截图，以及自定义图文等。

## 功能事件流分析

用例是需求分析中最重要的概念，需求表征了该程序的设计特性、特征和运作逻辑，通过对系统进需分析，可以看到这款游戏所包括的玩法和功能。

### 主界面用例

首先用户进入特性界面，由简单的LaunchScreen.xib文件构成初始化等待界面的排版。

##### 英文名称：<MyScene>

##### 中文名称：<主界面>

##### 简要说明：对选择界面场景初始化流程进行说明

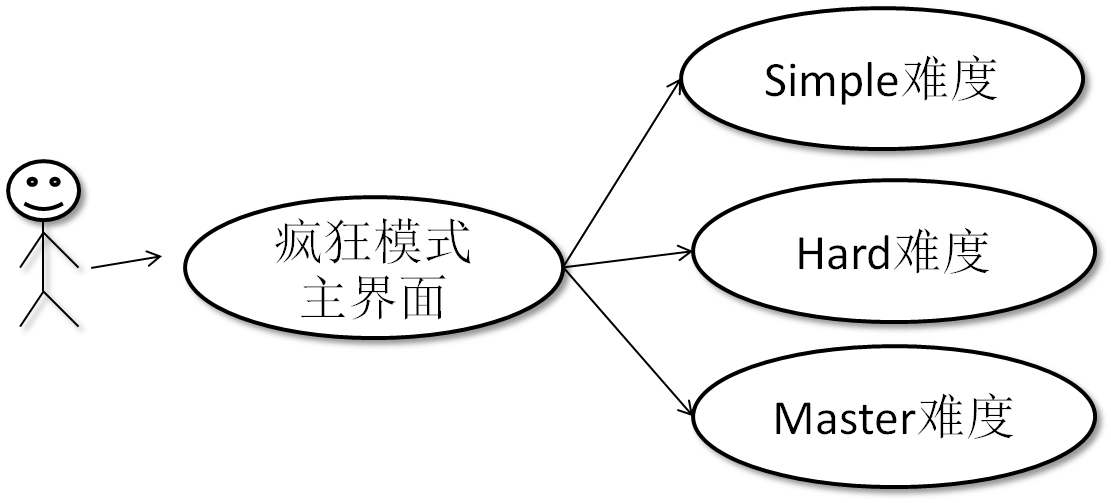
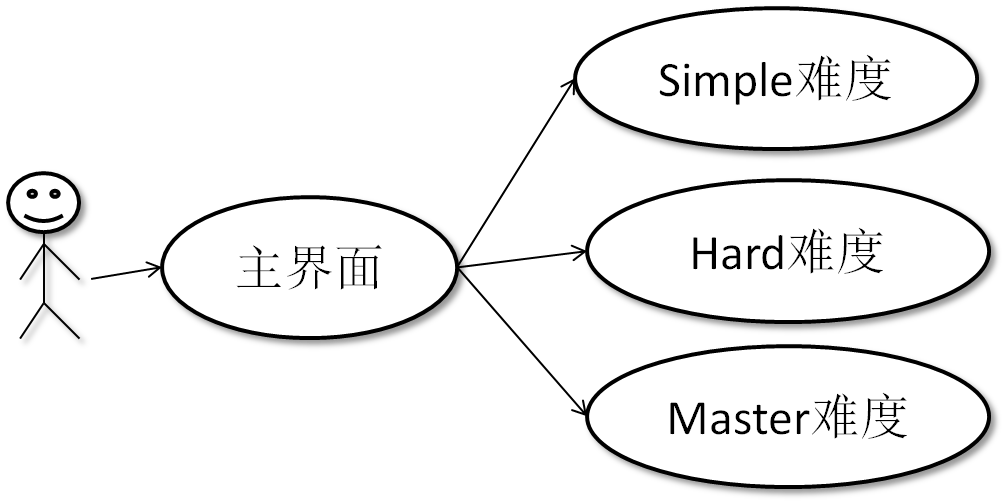
##### 基本事件流：

选择界面初始化；

选择界面背景\特效初始化。(包括特效粒子界面，逐帧动画，重力引擎以及碰撞检测)；

Update开始，标准刷新频率为60.0 fps；

进入待机状态。



图

##### 功能：

跳转疯狂模式<CreazyMode>

##### 前置条件：游戏激活状态。

### 游戏界面用例

##### 英文名称：<GameScene>

##### 中文名称：<游戏场景>

##### 参与者：<User>

##### 简要说明：对游戏场景初始化的流程进行说明。

基本事件流：

动画跳转界面；

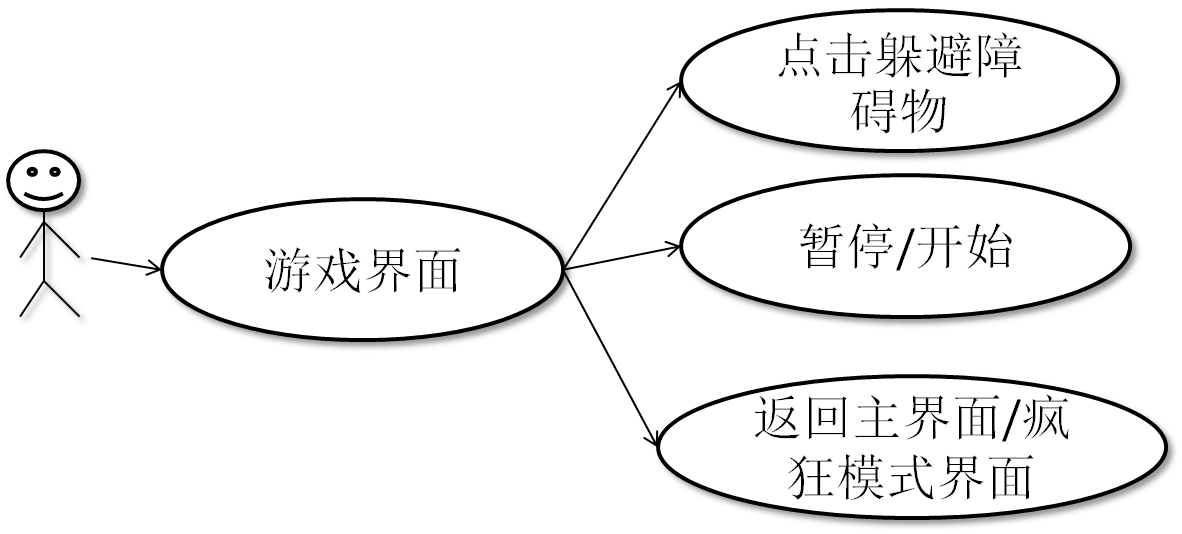
初始化游戏界面；

初始化游戏事件组；

Update开始，标准刷新频率为60.0 fps；

帧计时器开始计时；

刷新出现monster，sod。



图

##### 前置条件：游戏激活状态，进入游戏界面。

### 结果界面用例

##### 英文名称：<ResultScene>

##### 中文名称：<结果成绩场景>

##### 参与者：<User>

##### 简要说明：记录玩家得分，截图，分享上传

##### 基本事件流：

判断游戏成绩是否最高，并归档；

初始化场景文字与动画；

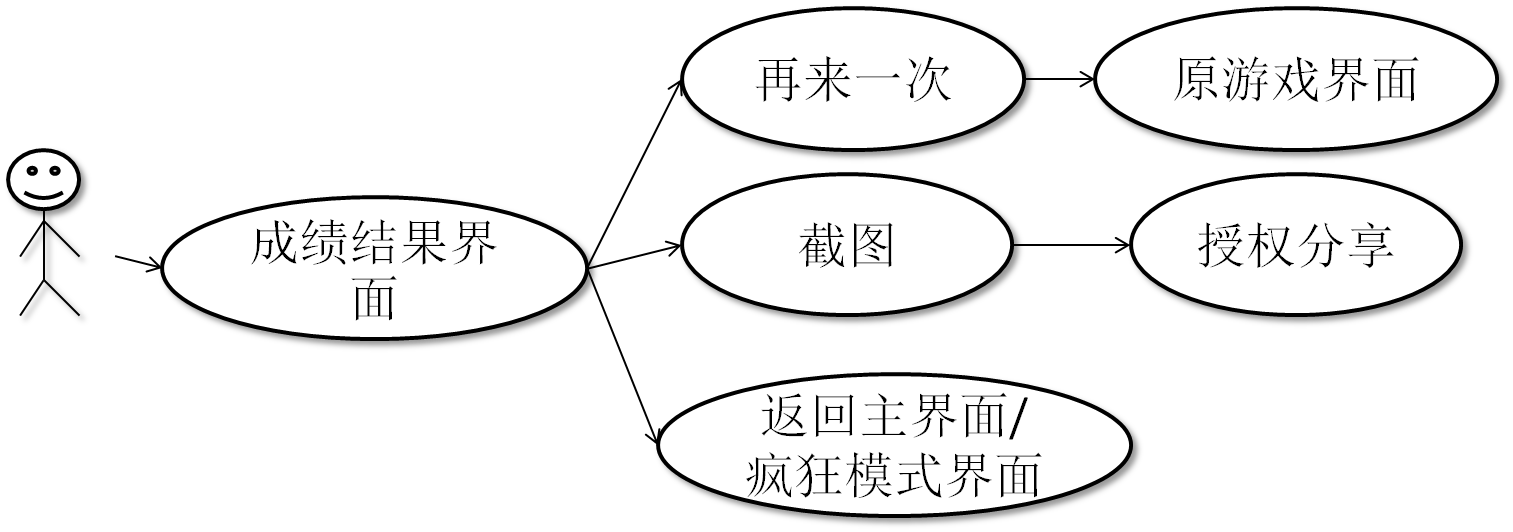
前置条件：游戏激活状态，进入结果界面；

##### 功能：

返回（Menu）：返回选择界面（MyScene）；

再次挑战（TryAgain）：再次进入游戏界面，难度为当前难度不变；

分享：（MicroBlog）：弹出对话框：1 截图并存储到相册；2 分享当前截图到微博（之后会有详细介绍）。



图

### 分享界面用例

##### 英文名称：<ComposeController> ；<OAuthViewController>

##### 中文名称：<分享界面><授权界面>

##### 参与者：<User>

##### 简要说明：发出微博的分享。

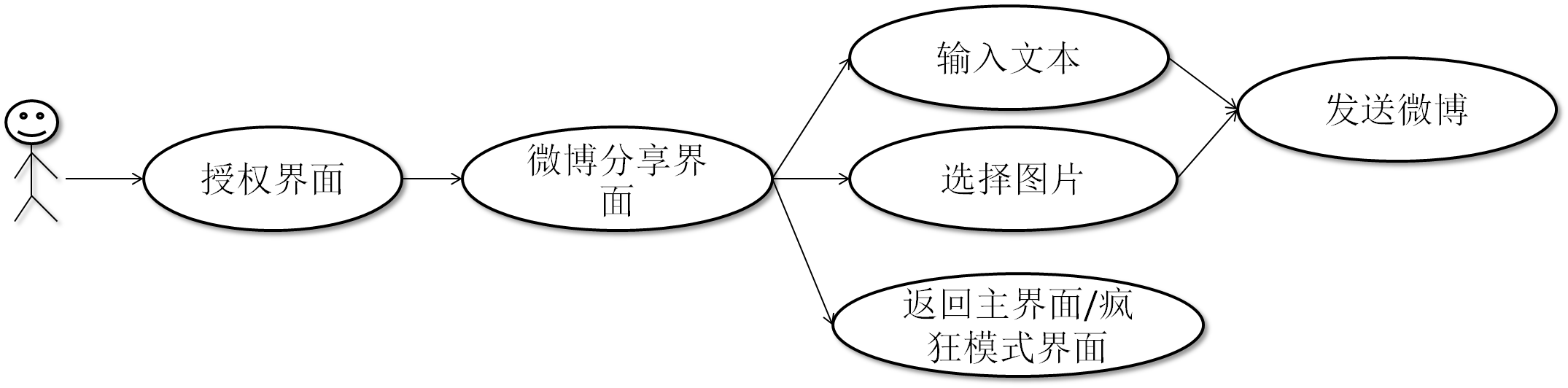
##### 基本事件流：

判断用户是否已授权（归档文件）

没有授权的话进入授权界面

已授权或者完成授权的话用户进入发微博（Compose）界面。

用户自定义文本或者从相册选择图片发微博。



图

##### 前置条件：游戏激活状态。

## 功能分析

在iOS设备上开发循环类游戏，需要关注的问题是

#### 通过触摸控制游戏的事件触发

点击动态背景（sod），触发所在行的小鸟跳跃动作，来躲避障碍物。小鸟被障碍物撞到时，判断为游戏结束，并记录当前时间为当前难度的成绩，也是游戏最为核心的循环逻辑

#### 事件的重合冲突

游戏有多层需要控制跳跃的小鸟。要求每一层的点击事件仅仅触发该层的小鸟的跳跃，不能够影响到其他层的小鸟。并且，当小鸟跳跃而起时，是不能够再次跳跃的，此时的该层跳跃的事件应该被加锁；反之，当小鸟落地时，跳跃时间将被解锁。

#### 小鸟的跳跃伴随着跳跃时的音效

允许用户同时点击不同层数，并同时让小鸟们跳起来，因为随即出现的障碍物有可能会同时出现（难度越高，同时出现的几率也就越大）。游戏主可以选择游戏的等级，游戏难度的增加，用湖所需要控制的游戏层数也会增加，同时需要用户的反应能力

#### 游戏的暂停

游戏的友好性在于能考虑用户的需要，随时可以暂停或者继续游戏，在不愿继续游戏时退出游戏。本程序可以在用户需要的时候点击返回（back）按键，等待选择，暂停和退出功能做成一体，在退出的提示下不做任何操作即可实现暂停的功能，选择Menu游戏主界面。

用户可以在主界面分享自己的成绩，并上传截图。如果用户第一次登陆微博，需要通过授权界面授权这款游戏发微博功能，之后用户别可以在新浪微博上发微博

## 本章小结

本章主要是对该系统进行了需求分析，主要包括了对该系统的可行性分析、需求描述、面向对象分析、主要功能分析等。针对对该系统实行需求分析，查阅需求分析作用，我学习到了，需求分析的作用是详细分析程序实现所要处理的对象，充分解析原系统工作构造，明确各种将要实现的需求然后在此基础上确定新系统的功能。确定对系统的整体实现要求，虽然功能需求是对软件系统的一项基本需求，但却并不是唯一的需求。

因为是游戏软件，视图的流畅性，用户体验无bug是贯穿头尾的原则问题。苹果设备的优良体验是众所周知的，完美的硬件适配为游戏的60fps保障。因为这款游戏的动画效果非常之多，通过iOS自带的ARC机制可以及时的释放不必要的内存占用，但是需要对引用计数有严格的逻辑监控。考虑到编译调试的过程，内存吃紧的情况任然有可能发生，所以项目的重构，以及再书写十分的重要。

# 系统设计

通过需求分析，指明了更加具体需求方案。再将其付诸到工程代码上，需要一步步遵从系统设计结构，做到时间逻辑的完整性。在系统设计中，考虑到程序发展方向，对于各种特殊情况的正确响应，需要将设计流程具体化。做到尽可能简明，松耦合，便于程序的扩展。

#### 整体性保障原则

系统开发遵循要“总分总”原则。先规划工程蓝图，进行模块划分。分模块进行苟能上的实现。再根据模板需求进行模块的之间的整理联系。

#### 实用性原则

衡量一款软件体系的最重要的指标就是其实用性，是否与需求结合、是否严格按照业务需求有针对性地实现，是系统产品成败的关键因素。

#### 先进性原则

要立关注并足于先进技术，让系统技术流淌先进的血液。保证系统可靠地运行同时，磨练业务的熟练度。充分考虑用户的需求，从语法开始分析，保证系统的可使用性、可维护性。

#### 安全性原则

系统的安全性不容忽视，必须充分考虑来自网络内部和外部潜在的风险，采取有效的逻辑安全。以安全为前提的需求是最根本的需求。

#### 模块化设计原则

根据系统的需要动态变化，应采用模块化的设计模式，使功能结构由面向对象的单继承结构，具有可伸缩性和可扩展性以实现系统以后发展的需要。

## 概要设计

本课题制作的游戏分场景描述，主要有：主界面场景、游戏场景、成绩结果场景、微博分享场景，在这其中主界面场景和游戏场景又有疯狂模式主界面场景和疯狂模式游戏场景的特殊场景分类；游戏场景由难度区别为Simple、Hard和Master三种难度。

详情如下图 - 所示

图

## 详细设计

详细[设计](http://baike.baidu.com/view/14417.htm" \t "_blank)是[软件工程](http://baike.baidu.com/view/1659.htm)中[软件开发](http://baike.baidu.com/view/190611.htm)的一个阶段性步骤，是对[概要设计](http://baike.baidu.com/view/551728.htm)的一个细化。主要在于实现设计每个模块，完善[局部](http://baike.baidu.com/view/470791.htm)模块功能，是真正意义上的细节设计。

### 系统总流程

图

### 系统各个模块的处理流程

图

## 本章小结

阿发冯

# 系统实现

Ss

## 节点特效的实现

节点的初始化

节点的动画

节点的音效

## 点击事件的实现

事件的初始化

音效的实现

跳转的实现

## 主界面的设计

Ss



图

## 游戏界面的设计

Ss



图

## 成绩界面的设计

Ss



图

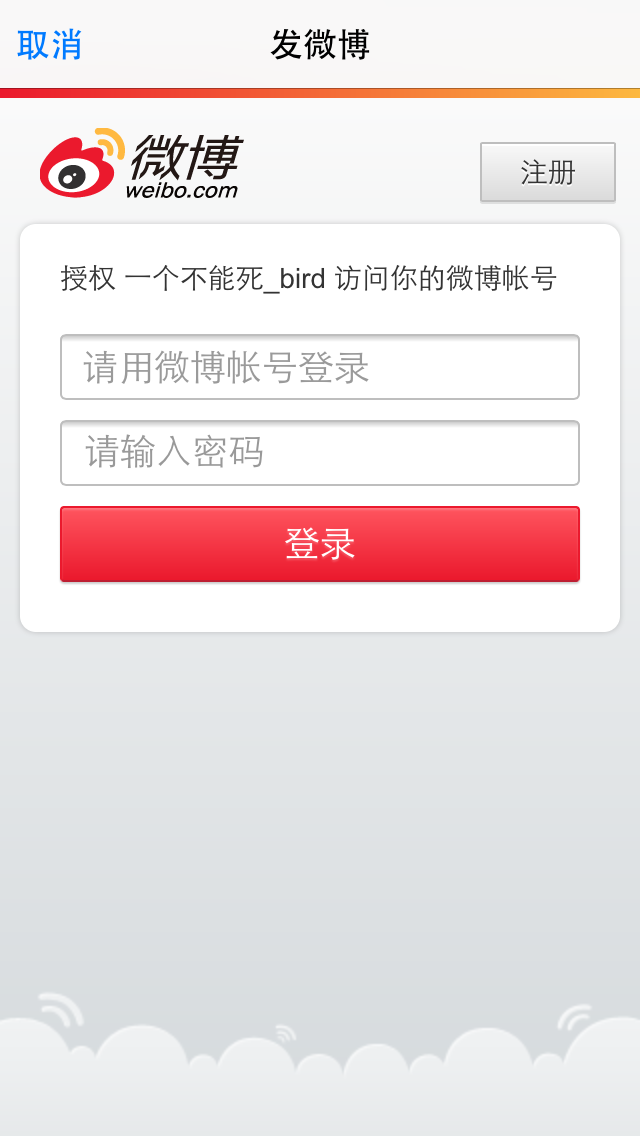
## 声音系统的实现

Ss

//

//

## 分享到新浪微博



图

## 存储

Ss

//

//

# 测试

在设计游戏的过程中，可能存在编译错误。语法的错误会被便以其明确报出，此类错误能够及时纠正。但另一类错误逻辑设计时的失误，例如下角标越界造成的内存泄露。这类错误隐蔽性强，因此，对这一类动态发生的错误的排查是耗时费力的。

游戏测试作为[软件](http://baike.baidu.com/view/37.htm" \t "_blank)测试的一部分，它具备了软件测试所有的一切共同的特性：测试的目的是发现软件中存在的缺陷。测试都是需要测试人员按照产品行为描述来实施。产品行为描述可以是书面的规格说明书，需求文档，产品文件，或是用户手册，[源代码](http://baike.baidu.com/view/60376.htm)，或是工作的可执行程序。

## 程序的测试

### 测试的重要性及目的

#### 测试的重要性

软件测试学仅处于运行维护阶段之前，是软件产品交付用户使用之前保证软件质量的重要手段。近来，软件工程界趋向于一种新的观点，即认为软件生命周期每一阶段中都应包含测试，从而检验本阶段的成果是否接近预期的目标，尽可能早的发现错误并加以修正，如果不在早期阶段进行测试，错误的延时扩散常常会导致最后成品测试的巨大困难。

在软件开发的总成本中，用在测试上的开销要占 30％到 50％。如果把维。护阶段也考虑在内，讨论整个软件生存期时，测试的成本比例也许会有所降低，但实际上维护工作相当于二次开发，乃至多次开发，其中必定还包含有许多测试工作[14]。

在实践中，软件测试的困难常常使人望而却步或敷衍了事，这是由于对测试仍然存在一些不正确的看法和错误的态度。

#### 测试的目的

软件测试的目的,第一是确认软件的质量，其一方面是确认软件做了你所期望做的事情（Do the right thing），另一方面是确认软件以正确的方式来做了这个事情（Do it right）。第二是提供信息，比如提供给开发人员或程序经理的回馈信息，为风险评估所准备的信息。第三软件测试不仅是在测试软件软件产品本身，而且还包括软件开发的过程。如果一个软件产品开发完成之后发现了很多问题，这说明此软件开发过程很可能是有缺陷的。因此，软件测试的第三个目的是保证整个软件开发过程是高质量的[15]。

如果测试的目的是为了尽可能多地找出错误，那么测试就应该直接针对软件比较复杂的部分或是以前出错比较多的位置。

(1)软件测试是为了发现错误而执行程序的过程;

(2)测试是为了证明程序有错，而不是证明程序无错误;

(3)一个好的测试用例是在于它能发现至今未发现的错误;

(4)一个成功的测试是发现了至今未发现的错误的测试

### 测试的步骤

游戏开发阶段，随着游戏文档说明、策划案例的编写和游戏初期版本的形成。就可以去编写最早期的测试规范、测试计划等等。其实游戏测试的用例相对软件测试的用例会庞大很多，游戏本身就是一个比软件功能多很多的软件，自然用例分的地方也会多很多，可以分：单人、多人、主流程、对局里等等。(产品设计文档、测试文档初期形成)

#### 单元测试

在这个测试步骤中所发现的往往是编码和详细设计的错误

#### 系统测试

在这个测试步骤中发现的往往是软件设计中的错误，也可能发现需求说明中的错误。

### 测试时出现的问题

结束语

apple推出sprite kit，让我想起了game center打败openfient的故事。相信apple还是希望开发者推出iOS平台独有的游戏，我相信apple会不断完善他，结合gamecenter，icloud，推出有别于别的平台独有的功能

参考文献

1. Carlo Chung（美）,刘威译 .Objective-C编程之道：iOS设计模式解析[M]北京：人民邮电出版社,2011年
2. Stephen G. Kochan（美），林冀，范俊，朱奕欣 译 .Objective-C 程序设计(第4版)[M]北京：电子工业出版社,2014年

[3]Tutorial Team，.iOS 5 By Tutorials

附录

致谢

第7页(第6行第1列)样式不对

第7页(第6行第1列)样式不对

第7页(第6行第1列)样式不对

第7页(第6行第1列)样式不对

第7页(第6行第1列)样式不对

第7页(第6行第1列)样式不对

第7页(第6行第1列)样式不对

第7页(第6行第1列)样式不对

第7页(第7行第1列)样式不对

第7页(第8行第1列)样式不对

第7页(第10行第1列)样式不对

第7页(第11行第1列)样式不对

第7页(第12行第1列)样式不对

第7页(第13行第1列)样式不对

第7页(第15行第1列)样式不对

第7页(第16行第1列)样式不对

第7页(第17行第1列)样式不对

第7页(第18行第1列)样式不对

第7页(第19行第1列)样式不对

第7页(第20行第1列)样式不对