

学校编码: 10384

分类号 \_\_\_\_\_ 密级 \_\_\_\_\_

学号: X2013230234

UDC \_\_\_\_\_

廈門大學

工程硕士学位论文

基于 IOS 的游戏绘本的故事流系统设计与  
开发

Design and Implementation of System for Story  
of Picture Book Game Based on IOS

洪宇晨

指导教师: 余莹莹 助理教授

专业名称: 软件工程

论文提交日期: 2015 年 6 月

论文答辩日期: 2015 年 7 月

学位授予日期: \_\_\_\_\_ 年 月

指导教师: \_\_\_\_\_

答辩委员会主席: \_\_\_\_\_

2015 年 7 月

## 厦门大学学位论文原创性声明

本人呈交的学位论文是本人在导师指导下，独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合法律规范和《厦门大学研究生学术活动规范（试行）》。

另外，该学位论文为（）课题（组）的研究成果，获得（）课题（组）经费或实验室的资助，在（）实验室完成。（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明。）

声明人（签名）：

年 月 日

## 厦门大学学位论文著作权使用声明

本人同意厦门大学根据《中华人民共和国学位条例暂行实施办法》等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论文（包括纸质版和电子版），允许学位论文进入厦门大学图书馆及其数据库被查阅、借阅。本人同意厦门大学将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

1. 经厦门大学保密委员会审查核定的保密学位论文，  
于 年 月 日解密，解密后适用上述授权。

2. 不保密，适用上述授权。

（请在以上相应括号内打“√”或填上相应内容。保密学位论文应是已经厦门大学保密委员会审定过的学位论文，未经厦门大学保密委员会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权。）

声明人（签名）：

年 月 日

## 摘要

随着现代科技不断地与飞跃式快速进步，手机、平板、手表等移动设备的迅猛发展，越来越多人的学习，游戏时间变得越来越碎片化，阅读如是。由于人们可以在公交车上，椅子上，甚至于走路时都可以使用移动设备，所以电子读物的使用场景也变得越来越丰富。传统有的以图文形式呈现的绘本已慢慢无法满足人们的阅读需求，固其正在一步步往多媒体的方向转变，这也是移动端绘本游戏开发的客观需求。

绘本用简单的文字和图画结合可以非常形象直观地向人们表达其所要表达的主题和内容。这一点对孩童是非常有帮助的，因为孩童的思维也较为简单，绘本这种形象直观的方式结合上一些寓教于乐的内容，既能引起孩童的共鸣，亦能让孩童喜欢上阅读。市面上看到的绘本游戏大多是将传统绘本搬到移动端，而未充分地运用其移动设备的优点与其结合。移动设备在满足人们便携性的同时，其交互体验亦可增加良好的用户体验，让人们参与创造故事，这样其可玩性会大大地增加。在一次偶然搜寻毕业生的设计中发现已有前辈开发出了二维交互式绘本游戏引擎，但是发现其还存在一些漏洞及功能缺失。本着避免重新造轮子的原则，想在该绘本游戏引擎的基础上开发故事线的功能，让绘本游戏的表现形式更有渲染力，流程更加丰富多样，同时丰富其可玩性。在增加功能的同时还需要对原有引擎进行架构上的优化以及修复闪退问题。

本文以原有的绘本引擎作为立足点，主要工作如下：

1. 优化架构，实现角色和交互分离。
2. 增加功能，让场景物体关联响应。
3. 优化内存，解决闪退问题。

论文的内容主要是先对设计的目的进行介绍。由于原有绘本是基于 IOS 的二维游戏引擎的架构，所以紧接着论文会介绍 IOS 的开发技术。由于需要优化 MVC 框架，论文会介绍 MVC 框架的理论基础。同时论文会对功能的需求进行分析，实现方式，以及测试方案。最后会介绍功能开发完成后的进一步改进计划，为后续的开发人员提供支持。

**关键词：** 绘本改进；MVC 框架；电子游戏绘本

## **Abstract**

With the development and the rapid growth of modern technology, there are more and more mobile phones, tablets, watches and other devices. The time to study and play becomes fragmented, so as reading time. Electronic books usage scenarios are more abundant, since people can use devices when commuting on the bus, when sitting on the chair, even when walking on the road. Traditional picture books that are presented in the form of picture books are unable to fulfill people's reading needs, so some of them shift towards the direction of new media, which is the objective need of developing mobile games of picture books from the market.

Simple combination of text and pictures enables picture books to visually express the themes and contents to viewers. Picture books are helpful for cultivating children's reading habit, since the simple form of them suits children's way of thinking. It is also in advantage of achieving emotional resonance for children. It is a waste that current picture books games move the rational form to mobile device without combining the advantage of mobile device. Mobile devices are easy to carry, it can also increase the interaction within its good user experience, so that people involved in creating the story, which would greatly increase its playability.

Although former graduates has developed a two-dimensional interactive picture books, there are still some loopholes and function loss. In line with the principle of avoiding re-create the wheel, adding the feature of story line on the game engine would make the game gain more expression power, optimize the process, while enrich its playability. The increased functionality also need to optimize existing engine architecture as well as repair crash problems.

In this paper, taking the original picture book engine as the standpoint, the main work is as follows:

1. Optimize architecture to achieve the separation of roles and interaction.
2. Increased functionality so that the associated object scene response.
3. optimize memory, problem solving: crash.

After introducing the purpose of design, IOS development technology will also

be introduced since the original picture book is a two-dimensional game engine architecture based on IOS. Due to the need to optimize MVC framework, this paper will introduce the theoretical basis of the MVC framework. Papers will be analyzed at the same time functional requirements, implementation, and testing program. Finally, to further improve the plan will be introduced after the function developed to provide support for subsequent developers.

**Keywords:** picture books improve; MVC framework; electronic game picture books

厦门大学博硕士论文摘要库

## 目 录

<b>第一章 绪论</b> .....	<b>1</b>
1.1 背景和目的.....	1
1.1.1 背景.....	1
1.1.2 目的和意义.....	1
1.2 论文组织结构.....	2
<b>第二章 相关技术概述</b> .....	<b>3</b>
2.1 MVC 框架 .....	3
2.2 IOS 框架 .....	3
2.2.1 IOS 开发环境.....	3
2.2.2 IOS 开发语言 .....	4
2.2.3 IOS UIKit 框架 .....	5
2.3 存储方式.....	6
2.4 Instruments.....	6
2.5 本章小结.....	7
<b>第三章 系统需求分析</b> .....	<b>8</b>
3.1 功能策划.....	8
3.1.1 用户画像.....	8
3.1.2 需求确认.....	11
3.2 需求分析及设计 .....	13
3.2.1 整体需求分析.....	13
3.2.2 具体功能模块设计.....	15
3.2.3 具体页面实现.....	20
3.3 小结 .....	22
<b>第四章 系统设计</b> .....	<b>23</b>
4.1 整体框架.....	23
4.2 功能模块设计.....	24
4.2.1 视图模块.....	25

4.2.2 控制模块.....	26
4.2.3 模型模块.....	26
<b>4.3 数据库设计.....</b>	<b>27</b>
4.3.1 输入设计.....	27
4.3.2 输出设计.....	27
4.3.3 数据流图.....	27
4.3.4 数据库 E-R 图.....	29
<b>4.4 类 .....</b>	<b>33</b>
4.4.1 功能相关视图类.....	33
4.4.2 数据存储中的类.....	34
4.4.3 其他模型中的类.....	34
<b>4.5 小结 .....</b>	<b>34</b>
<b>第五章 系统实现 .....</b>	<b>35</b>
5.1 Story 数据 .....	35
5.2 Story 编辑 .....	37
5.3 Story 的播放 .....	40
5.4 小结 .....	41
<b>第六章 系统测试 .....</b>	<b>43</b>
6.1 单元测试.....	43
6.2 功能测试.....	44
6.3 回归测试.....	45
6.4 兼容性测试.....	46
6.5 性能测试.....	50
6.6 稳定性测试.....	52
6.7 本章小结.....	54
<b>第七章 总结与展望 .....</b>	<b>55</b>
7.1 总结 .....	55
7.2 展望 .....	55



参考文献 .....	57
致 谢 .....	58

厦门大学博硕士论文摘要库

## Contents

<b>Chapter I Preface .....</b>	<b>1</b>
<b>1.1 Background and Objectives .....</b>	<b>1</b>
1.1.1 Background .....	1
1.1.2 The Purpose and Importance .....	1
<b>1.2 Structure .....</b>	<b>2</b>
<b>Chapter II Related Technical Overview .....</b>	<b>3</b>
<b>2.1 MVC Framework.....</b>	<b>3</b>
<b>2.2 IOS Framework .....</b>	<b>3</b>
2.2.1 IOS Development Environment.....	3
2.2.2 IOS for Programming Language.....	4
2.2.3 IOS UIKit Framework .....	5
<b>2.3 Storage.....</b>	<b>6</b>
<b>2.4 Instruments.....</b>	<b>6</b>
<b>2.5 Summary.....</b>	<b>7</b>
<b>Chapter III System requirement analysis and function design.....</b>	<b>8</b>
<b>3.1 Functional planning.....</b>	<b>8</b>
3.1.1 User Portrait .....	8
3.1.2 Demands Confirmation .....	11
<b>3.2 Demands Analysis .....</b>	<b>13</b>
3.2.1 Overall demand analysis .....	13
3.2.2 Specific features modular design .....	15
3.2.3 Achieve specific page .....	20
<b>3.3 Summary.....</b>	<b>22</b>
<b>Chapter IV functional System Architecture Design .....</b>	<b>23</b>
<b>4.1 Overall Framework .....</b>	<b>23</b>
<b>4.2 Functional Module Design .....</b>	<b>24</b>
4.2.1 View Module.....	25

4.2.2 Control Module .....	26
4.2.3 Model Module .....	26
<b>4.3 Data Library Design .....</b>	<b>27</b>
4.3.1 Input Design .....	27
4.3.2 Output Design .....	27
4.3.3 Data Flow Diagram .....	27
4.3.4 E-R Picture Of Databases .....	29
<b>4.4 Class .....</b>	<b>33</b>
4.4.1 Functionally Related View Class .....	33
4.4.2 Data Storage Class .....	34
4.4.3 Other Classes in the Model .....	34
<b>4.5 Summary .....</b>	<b>34</b>
<b>Chapter V System implementation .....</b>	<b>35</b>
5.1 Story data .....	35
5.2 Story Editor .....	37
5.3 Story Play .....	40
5.4 Summary .....	41
<b>Chapter VI System testing .....</b>	<b>43</b>
6.1 Unit Testing .....	43
6.2 Functional Testing .....	44
6.3 Regression Testing .....	45
6.4 Compatibility Testing .....	46
6.5 Performance Testing .....	50
6.6 Stability Testing .....	52
6.7 Summary .....	54
<b>Chapter VII Summary and Outlook .....</b>	<b>55</b>
7.1 Summary .....	55
7.2 Outlook .....	55

**References** ..... 57

**Acknowledgements** ..... 58

厦门大学博硕士学位论文摘要库

## 第一章 绪论

### 1.1 背景和目的

#### 1.1.1 背景

绘本从字面上的理解就是“画出来的书”，指的是以绘画为主，同时附上文字的书藉。它与其他的图画书的区别是，绘本在于从画里说故事，而图画书则是故事中配上画。绘本相较于其他的图画书更加直观，直接从图画中就能向读者展示图书的主题，内容以及细节。随着科技一步步的发展，绘本也变得更加多样化。市面上的绘本主要有电子绘本、有声绘本、绘本杂志，传统绘本。其中电子绘本借助移动设备的发展与普及而变得越来越有冲击力，因为人们可以充分地运用碎片化时间阅读电子绘本。同时电子绘本可以结合声音，动画，图片，游戏等等信息于一体，会比其他种类的绘本更具优势。

用“有得必有失”这句话形容现在的电子绘本形式一点也不为过。由于电子绘本的便捷性，娱乐性吸引了越来越多的用户。许多电子绘本产品为了抢占市场，在制作上大多草草了事，背景模糊生硬的同时，在核心的绘本剧情编排缺乏合理性。在交互上，要嘛是复杂地让人不懂用，要嘛是简单的乏味。这些电子绘本应用非但无法提供给用户愉悦，心灵抚慰，还会严重地伤害用户的心。用户的核心需求是什么？我用你的产品，我要有一些愉快的收获，同时如果能对我有所提高就更好了。所以本设计就是在原有绘本项目的基础上进行二次开发，在给用户提供良好体验的同时让用户可以自己去创造有意思的，符合自己的个性化内容，让用户能够玩起来，最终达到寓教于乐的目的。

#### 1.1.2 目的和意义

本项目的目的在于设计新的功能，让用户融入故事情节后可以自己创造一个属于自己个性化的故事线；能在阅读与游戏的过程中感受到有趣，好玩；同时在人机交互上感受到舒畅的体验；同时通过故事潜移默化地让用户获得知识的提升。

随着现在市场上的绘本产品不断更新，但是普遍还停留在供用户阅读，用户从应用中获得精神愉悦上。但是这里有两个问题，（1）一个产品提供内容的目标用户是有限的，一篇固定的绘本不一定能给用户产生愉悦感，说不定还会产生反感情绪，而这些不好情绪的矛头指向的便是产品本身；（2）即使用户喜欢其

中的一条内容，由于用户是被动接收到这则信息，加上没有互动，用户用了一段时间后也会觉得没有意思，用户的流失率是很大的。而一款能让用户在感觉愉悦的同时自发地编辑内容，剧情，然后转发给朋友分享快乐，将能更长久更长久地留住用户的同时，亦能丰富产品内容，达到寓教于乐的目的。

由于现在市场中的绘本产品大多都是根据现有模板进行内容填充，因为这样可以降低开发和运营的成本。所以本课题的意义在于二次开发和优化原有的绘本，让游戏架构更加容易上手，增加游戏架构的实用性，同时服务于后期的开发，让后期开发可以更加高效。此产品同时还有助于维护社区的活跃度，让社区的内容更加丰富。

## 1.2 论文组织结构

本论文共有七章，先从项目背景出发，介绍项目开发的目的和意义。接着介绍了项目开发所用的相关技术。介绍完项目开发所用的相关技术后，着重介绍了用产品思维进行功能设计的思路与方法，包括用户画像以及数据分析，用产品的方法论设计出了功能实现的具体逻辑。紧接着设计了完成功能的框架以及简要介绍了具体的代码实现。最后根据项目开发过程中产生的新的产品思路和遇到的问题进行总结与展望。

具体论文的安排如下：

1. 绪论。分析现状，引出为什么要进行该项目的开发的意义，介绍了本项目开发的背景。
2. 相关技术概述。简要地介绍了项目研究中所需要运用到技术的理论知识。
3. 功能需求分析与设计。主要描述用产品的思维对项目进行设计，包括描绘用户画像以及通过数据分析需求，具体介绍了用以用户为中心的思维设计产品。
4. 功能系统架构设计。详细描述原架构设计的同时详细介绍了项目开发的架构设计。
5. 具体实现。介绍了项目开发的具体实现，展示各个模块的代码。
6. 测试。介绍了项目开发所需要的各种黑盒测试方法，并贴出测试用例
7. 总结与展望。对本次项目开发进行了总结，描述项目开发中遇到的问题，同时展望了项目在未来的发展方向。

## 第二章 相关技术概述

### 2.1 MVC 框架

MVC 的全名是 Model View Controller，是一种软件设计的典范，

Model 就是模型层，表示应用的核心。主要负责将数据库读出的数据 Model 化，以及将 Model 解析成数据存入数据库中。

View 就是视图层，显示数据。主要负责展现视图，显示 Model。同时 View 层可将接收到的请求通过 Controller 将数据返回到 Model 或直接将数据返回 Model。也可请求 Controller 改变现有视图状态。

Controller 就是控制器，Controller 也可以理解成为逻辑层。主要负责将用户输入的信息或用户改变的视图状态返回给 Model 层或 View 层，

MVC 框架将逻辑，模型，视图分离的方法来组织代码，将所有的逻辑聚集到一个部件中，所有的视图表现放在一个层中，所有的模型数据读取又放在一个层中。通过这样的结构，当改进一个自定义界面与用户的交互的时候，不需要在每个页面重新去编写逻辑，而只需要改动 View 层的显示就能完成。MVC 分层对于管理复杂的应用是非常有帮助的，因为开发人员可以在一段时间内专门关注一个方面；同时 MVC 分层大大地简化了分组开发，让视图层，控制器，模型层都能同时地被不同的开发人员开发，并且不易产生影响。MVC 框架的可重用性也是非常高的，可以提高应用的灵活性以及可配置性。总而言之，MVC 框架实现地越彻底，三层的耦合度就越低，那么维护代码就会更加方便。在 IOS 开发中，需要解决控制器和视图层的松耦合互动的问题，面对该问题可以采取 Delegate（代理模式）。与此同时，通过 Action-Target 模式能让控制器坚挺视图层的时间。

### 2.2 IOS 框架

#### 2.2.1 IOS 开发环境

美国苹果公司在 2007 年 1 月 9 日的 Macworld 大会上公布了 IOS 这款移动操作系统，IOS 这款操作系统最早的设计仅仅是给 iPhone 使用的，之后陆陆续续地用到了 iPad、ITouch、Apple TV 以及 Apple Watch 产品上，不远的未来可能亦会用到车载系统中。IOS 属于类 Unix 的商业操作系统。

IOS 的系统架构分为四个层次：Core OS layer（核心操作系统层）、Core Services layer（核心服务层）、Media layer（媒体层）和 Cocoa Touch layer（可触摸层）。

(1) Core OS layer 是 IOS 系统架构中核心操作系统层，文件系统、内存管理、电源管理以及一些其他的操作系统任务都包括在该层中。这一层是直接和硬件进行交互的，所以应用开发者不会涉及到该层。

(2) Core Services layer 是核心服务层，IOS 的一些服务是通过它来访问的。

(3) Media layer 是媒体层，应用程序中音频和视频的录制，媒体文件的使用，基础动画效果的制作，图形的绘制等等都是通过媒体层使用的。

(4) Cocoa Touch layer 是可触摸层，顾名思义，用户在 IOS 设备上的触摸交互操作都是在这一层做的。

IOS 开发环境就是可以开发 IOS 软件的环境。苹果公司提供给开发者们一款官方的开发工具“Xcode”，这款开发工具集成了 IOS 开发环境，IOS SDK（IOS 软件开发工具包）。所以在 Xcode 上开发的应用可以直接在 IOS 环境中运行。

### 2.2.2 IOS 开发语言

苹果公司提供了两种开发语言用于开发 IOS 程序。

Objective-C 是苹果公司提供的一款扩充 C 的面向对象编程语言。用 Objective-C 写成的程序在通常情况下不会比其原始码大很多，因为它用 C 写成、很小的运行库。它完全兼容标准 C 语言。Objective-C 使用的是动态运行时类型，而且所有的方法都是函数调用，所以确定行为在运行应用时执行是被允许的，但也正是因为如此，所以很多常见的编译时性能优化方法都不能应用于 Objective-C，这也使得它性能劣于类似的对象抽象语言。一开始 Objective-C 是没有自动释放功能的，这也导致了用该语言开发经常会遇到内存溢出而导致闪退，但是在 Xcode4 中已经支持了自动释放的功能，所以苹果公司一直在不断地完善开发环境，让开发者们用 Objective-C 能够开发出更加稳定的应用。

Swift 是苹果公司在 2014 年发布的新开发的语言，它可与 Objective-C 共同运行在 IOS 平台。Swift 的语法内容混合了 Objective-C、JS、Python，语法简单并且使用方便。Swift 结合了 C 和 Objective-C 的优点并且不受 C 兼容性的测



Degree papers are in the “[Xiamen University Electronic Theses and Dissertations Database](#)”.

Fulltexts are available in the following ways:

1. If your library is a CALIS member libraries, please log on <http://etd.calis.edu.cn/> and submit requests online, or consult the interlibrary loan department in your library.
2. For users of non-CALIS member libraries, please mail to [etd@xmu.edu.cn](mailto:etd@xmu.edu.cn) for delivery details.