



本科毕业设计（论文）

手机游戏后期优化美术资源管理系统的设计与实现

**The design and implementation of the art resource
management system in mobile game development
project optimization processes**

学 院：_____

专 业：_____

学生姓名：_____

学 号：_____

指导教师：_____

北京交通大学

2016 年 5 月

学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：

指导教师签名：

签字日期： 年 月 日

签字日期： 年 月 日

中文摘要

自 2005 年，图形化手机网络游戏剧增，已经有超过 40 多款手机游戏面世。英特尔公司和盛大公司在 2005 年 6 月宣布联袂进军海内手机游戏产业市场，从此以后，手机游戏产业的阵营步入极速壮大阶段。手机游戏后期优化美术资源管理系统的设计开发，不仅可以实现在游戏优化过程中美术资源高效、合理的信息化管理、降低管理成本、提高工作效率，而且还能够辅助管理决策，优化配置资源，缩短游戏开发周期。从而达到提高核心竞争力的目的。

本文首先通过对手游开发行业的现状分析，再结合最近几年游戏开发行业的优缺点以及对本公司游戏开发程序员的调查走访，分析了当前对美术资源管理的具体需求，讨论了美术资源管理的系统目标。其次，通过绘制用户关系图确定系统用户，通过绘制系统的用例图，对系统各功能模块进行划分，从而确定了美术资源管理系统的功能模块。再次，对管理系统功能进行了详细的设计，提出了实现系统所采用的流程图、时序图以及类图，并对数据库进行了详细的设计。此外，本文还介绍了测试过程中所采用的测试方法和测试用例。最后对论文所阐述的内容进行了总结，并阐明了手机游戏后期优化美术资源管理系统的作用及意义。

本系统实现了手机游戏后期优化美术资源管理系统的大部分管理功能。具体包括建立用户信息管理、用来盛放资源的图集管理、美术资源管理、按硬件限制检索功按系统（Android 和 iOS）系统检索资源和检索资源信息等。笔者主要负责系统的设计和开发，完成了系统的需求分析、概要设计和详细设计等方面的工作。手机游戏后期优化美术资源管理系统提供具有高效性、最优性、准确性等特点的项目管理服务，以计算机管理资源代替人工管理资源，有效的减少了游戏开发过程中优化环节的时间，缩短了手游开发周期，提高了游戏运行的效率，提高了核心竞争力。

关键词：资源管理；优化；手机游戏；检索；

ABSTRACT

ABSTRACT: Since 2005, with the surge in graphical online mobile gaming, more than 40 mobile phone games has already come out. In June 2005, Shengda Corporation and Intel corporation announced the news that they would enter the field of domestic mobile game market together. Since then, the battle formation of mobile gaming industry began to unprecedented strange. The design and exploitation of the post-optimization for mobile gaming's art resource management system, can not only realize the effective and reasonable of informatization management, reduce management cost , improve work efficiency ,but also aid management decision in, optimize resource allocation , shorten the game development cycle during the process of game optimization. To attain the goal of advancing core competence.

This paper firstly analysed the current situation of mobile gaming, and through the analysis, combining the advantages and disadvantages of the game development industry in recent years and the interview and investigation of game development programmers, analyzed the current for the specific needs of the art resources management, then discusses the system objective of art resource management system. Secondly, through mapping the user diagram to determine the system user, through mapping system use case diagram, divide each function module division of the system, then determine the art resources management system function module. Once again, design the management system function in detail, put forward the implementation system adopted by the flow diagram, sequence diagram and class diagram, and design the database in detail .In addition, this article also describes the test method adopted in the process of testing and test cases. Finally, the paper makes a summing up and illuminates the effect and significance of the post-optimization for mobile gaming's art resource management system.

This system has realized the most management part of the post-optimization for mobile gaming's art resource management system, which includes setting up user information management, for resources's atlas management, art resources management, according to the hardware limitation retrieval and according to the system (Android and iOS) retrieval and retrieval resources information, etc. The author mainly responsible for system design and development, already completed the system demand analysis,

general design and detailed design, etc. The post-optimization for mobile gaming's art resource management system has many characteristics, such as efficiency, optimality, the accuracy of project management services. Use computer management to instead human resource management, effectively reduced the amount of time during optimization steps in the process of game development, shorten the games' development cycle, improve the efficiency of the game's running time, improved the core competitiveness.

KEYWORDS: Resource management; optimization; mobile game;search;

目 录

中文摘要.....	I
ABSTRACT.....	II
目 录.....	III
第一章 引言	1
1.1 课题背景与研究意义:	1
1.1.1 课题背景.....	1
1.1.2 研究内容.....	2
1.2 研究过程与交付结果	3
1.2.1 研究过程.....	3
1.2.2 交付结果.....	3
1.3 本人主要工作.....	4
1.4 论文组织结构	4
第二章 相关技术途径优选与确立	5
2.1 Unity3D 技术概述	5
2.2 基于 Unity3D 游戏优化方法概述.....	6
2.3 不同系统要求贴图格式的分析.....	6
2.3.1 Android 系统贴图格式分析	6
2.3.2 iOS 系统贴图格式分析	8
2.4 Unity3D 贴图压缩格式方法概述	10
2.4.1 图片压缩格式的比较	11
2.4.2 图片颜色格式的选择依据.....	12
2.4.3 图片压缩格式的确立.....	14
第三章 需求分析	16
3.1 系统功能性分析.....	16

3.1.1 用户行为分析.....	17
3.1.2 资源管理用户行为分析.....	18
3.1.3 图集管理用户行为分析.....	19
3.1.4 按硬件检索用户行为分析.....	19
3.1.5 按 Android 系统检索用户行为分析.....	21
3.1.6 按 iOS 系统检索用户行为分析.....	22
3.2 系统非功能性需求分析与定义	19
3.3 本章小节.....	24
第四章 系统概要设计.....	27
4.1 系统功能结构设计	28
4.2 系统体系架构设计	27
4.3 数据库设计.....	30
4.3.1 数据库概念结构设计.....	30
4.3.2 数据库逻辑结构设计.....	31
4.3.3 数据库物理结构设计.....	26
4.4 本章小结.....	34
第五章 详细设计和实现	35
5.1 注册登录子系统详细设计.....	35
5.1.1 功能说明.....	35
5.1.2 流程描述.....	35
5.1.3 业务逻辑和类设计.....	37
5.2 管理子系统详细设计.....	37
5.2.1 功能说明.....	37
5.2.2 流程描述.....	28
5.2.3 业务逻辑和类设计.....	41
5.3 按硬件检索子系统详细设计.....	41
5.3.1 功能说明.....	41
5.3.2 流程描述.....	42

5.3.3 业务逻辑和类设计.....	43
5.4 按系统检索子系统详细设计.....	44
5.4.1 功能说明.....	44
5.4.2 流程描述.....	44
5.4.3 业务逻辑和类设计.....	48
5.5 压缩资源子系统详细设计.....	47
5.5.1 功能说明.....	47
5.5.2 流程描述.....	47
5.5.3 业务逻辑和类设计.....	48
5.6 本章小结.....	49
第六章 系统测试	50
6.1 系统功能测试.....	50
6.2 系统性能测试.....	43
6.3 本章小结.....	44
第七章 总结与展望	55
7.1 工作总结.....	55
7.2 工作展望.....	55
参考文献	48
致 谢.....	50
附 录	51

第一章 引言

自 2005 年，图形化手机网络游戏剧增，已经有超过 40 多款手机游戏面世。英特尔公司和盛大公司在 2005 年 6 月宣布联袂进军海内手机游戏产业市场，从此以后，手机游戏产业的阵营步入极速壮大阶段，据不完全统计，目前国内手机游戏厂商已经将近 30 所。据专业人士分析，中国手机游戏产业当前处于市场导入期，在接下来的三年中，手游行业将脱离导入期，走入快速发展阶段，这样就促使了游戏开发管理理念和管理技术不得不快速发展，以便于快速、高效的促使新游戏的诞生。基于这种情况，我们亟需一套完整的管理系统和先进的管理技术理念以管理我们快速发展的手游开发行业。

1.1 课题背景与研究意义：

本课题源自于本人在北京掌趣科技股份有限公司的 8 个月的实习经历，深入到公司的项目中，在手机游戏研发的完整周期中发现问题，分析问题，解决问题。诞生了手机游戏后期优化美术资源管理系统。

1.1.1 课题背景

国内手机手机游戏发展现状：据调查显示国内的手机用户人数接近 3 亿，设想一下，若这些用户中的 10%，即使一个月下载一款手机游戏，着这也完全可以使手机游戏产业拥有非常庞大的规模。但就目前数据得知，仅有不足 1%的手机用户曾经下载过手机游戏，这个现象使得这块被大部分人寄予厚望的市场并没有达到和预期相吻合的水平。绝大部分手机用户却更容易接受短信和彩铃等类型的增值业务，对手机游戏这类增值业务的认可度却非常低。

据从业人员相关统计显示，在我国的手手机网络游戏的付费模式中，正在发生某些转变转变，越来越向现在的大部分 PC 网游一样趋近，同样仅在游戏过程中享受高级服务时才会向玩家收费。这样的模式对于平台推广商与游戏提供商来说是一个共赢与合作的新模式^[1]。但是目前来讲，这样的商业模式仍处在摸索阶段。在日本，几乎所有的游戏企业在准备新游戏开发前都会进行深入的市场调研，进行周密的策划，然后才会研究开发出一款完美的游戏，在精美包装和大力宣传后，进而投入运营，在运营过程中还会不断的开发游戏的价值、延长游戏的生命周期、从而带来长久不断的利益。通常惯用模式是在运营过程中，一开始玩家可以免费试玩，如果满意便会付费继续进行游戏，收费方式则一般有两种方式，分别是包月或一次性支付，收费说明公开透明，不会存在隐形

费用。

当然，我国手机游戏市场未来的增值空间非常大，其盈利模式也会丰富多样化，未来绝不会仅“游戏免费、高级服务收费”一条路可以走。另外，我国手机游戏市场还存在手机网上费用较高的问题，用户承受能力有限，很多用户表示不太愿意付费进行游戏，这个问题也牵绊着手机游戏产业的发展^[2]。

手机游戏存在问题：

（1）手机游戏市场发展空间大，潜在价值高，投入资金却很少，吸引了很多投资者相继进入市场，从而中小 SP 在激烈的市场竞争中该如何生存的问题是需要引起高度重视的^[3]。

（2）为了追求短期利益和低成本从而实现公司的利益最大化，开发手机游戏产品的投入和成本相对少，加上进入该市场的 SP 较多，但多数 SP 技术有限，产品设计中的游戏方案过程设计到任务设计均有不足与欠缺，在产品品质上下工夫又相对较少，诸多因素导致现在的游戏产品多质量粗糙^[4]。

（3）通常手机游戏产品的生命周期在 3 个月左右，所以如何了解用户需求，从而开发出受用户欢迎的产品至关重要，但同时更加迫在眉睫的问题是，如何保持游戏产品的粘性，持续保有用户^[5]。

（4）手机游戏的同质化也越来越严重，创新力不够。

1.1.2 研究内容

由上述手游产业现存问题的分析可以得出，在当今手游的市场中，开发手游的开发成本和投入方面均相对较少。手机游戏制作过程中得优化环节就显得至关重要，合理高效得优化手机游戏，可以给游戏项目带来质量上的大幅提升，使手机游戏适应性更强，在不同的设备上可以表现出更好的预期效果^[6, 7]。

问题的发现在使用同一图集（原数据）进行游戏开发过程中，不同的 UI 使用的美术资源格式大小不统一，为了提高游戏在不同的系统平台和硬件设施的适应性和性能，必须对 UI 资源进行管理，包括对美术资源的检索。

本论文研究的内容是如何才能使游戏后期优化过程中的美术资源被统一的管理，使计算机统一管理美术资源、检索美术资源的类型以及检索美术资源的大小，代替客户端开发人员手动管理检测美术资源，并研究何种压缩格式的美术资源是最优的，并以此格式为按系统检索美术资源算法的检索条件。

本论文研究目标在于，形成一套完整的基于 Unity3D 游戏引擎的美术资源管理系统，以分图集管理资源和自动审查方式提供具有高效性、最优性、准确性等特点的项目审查服务，以计算机管理资源代替人工管理资源，减少游戏开发过程中优化环节的时间，大

幅度增加资源的准确性，提高游戏运行的效率。

1.2 研究过程与交付结果

研究过程概括性地介绍了本课题的思路，交付结果则验证了本课题的可行性，本论文小结将从这两方面进行阐述。

1.2.1 研究过程

本课题的研究过程如下：

- （1）针对以 unity3D 为游戏引擎的手机游戏开发以及优化中，处理美术资源的方式。
- （2）研究当游戏运行调用美术资源的时候，CPU、GPU、帧缓冲区、显示器以及视频控制器等协作方式。
- （3）在针对以 unity3D 为游戏引擎的手游开发过程中，研究安卓系统平台和 iOS 系统平台美术资源的格式以及各自的优缺点。
- （4）根据本系统需求，研究设计对资源管理的最优方式。
- （5）针对研究结果，给出算法对按照格式要求和平台要求进行美术资源格式检索，并反馈检测结果。
- （6）针对研究结果，给出算法对给定标准的美术资源大小的检索，并反馈结果。
- （7）针对研究结果，给出算法对指定资源进行大小和格式方面的压缩，并反馈结果。
- （8）以软件工程思想为指导，分析，设计与实现手机游戏美术资源管理系统的技术原型，并验证该原型的可行性。
- （9）撰写论文

1.2.2 交付结果

针对缩短游戏优化时间的问题，预计形成一套完整的解决方案，在根据一定的硬件基础上，形成一套游戏资源管理系统，对用户提供的资源进行系统的管理并依照标准进行计算机检测美术资源代替客户端程序员手动检索测试美术资源，可以对美术资源进行手动压缩，包括大小压缩和格式压缩，针对美术资源的类型，大小，品质进行分类检索，并标记出不符合标准的美术资源，以列表的形式展现出来。代码成果：一套以 unity 引擎为基础的美术资源管理程序，编写语言为 JAVA。

1.3 本人主要工作

本人经历了手机游戏后期优化美术资源管理系统项目的整个开发周期，在该项目中，本人主要负责系统的设计和开发，最终目标是完成系统的图集信息管理、资源管理、按硬件标准检索、按 Android 系统检索、按 iOS 系统检索。我主要任务是从产品的角度详细描述了系统所具备的功能以及其中的特点，在系统概要设计方面，我设计并搭建了本系统客户端框架，并且对系统各个功能模块的数据库进行了详细设计。在对系统进行详细设计方面，本人根据需求分析以及概要设计，对每一个模块进行了详细的类的结构设计以及流程设计。最后本人分块实现该系统并进行测试。

1.4 论文组织结构

本文介绍了手机游戏后期优化美术资源管理系统的设计与实现过程，以软件工程的思想介绍了相关设计的细节。本论文的组织结构如下：

第一章 从课题背景与研究意义、研究过程与交付结果、本人主要工作和论文组织结构等方面介绍本系统。

第二章 从介绍 Unity3D 引擎、Unity3D 游戏项目优化分析、Unity3D 贴图格式压缩和本系统运行环境几个方面介绍了本系统的相关技术和原理。

第三章 从系统功能性分析和系统非功能性分析这两个方面坐出了系统的需求分析。

第四章 从系统的功能结构设计和系统体系架构设计两个方面介绍了系统结构，在数据库 ER 图的基础上介绍了数据库的概念结构、逻辑结构以及物理结构设计。

第五章 从注册登录子系统、管理子系统、按硬件检索子系统、按系统检索子系统和压缩资源子系统五个方面介绍了美术资源管理系统的详细设计与实现。

第六章 简要的介绍了系统测试的方法以及测试用例。

第七章 总结了系统建设中的收获与不足，并对未来工作做了部署

第二章 相关技术途径优选与确立

本章节介绍了本系统的关键技术途径的优选与确立。首先介绍了 Unity3D 的技术概述，之后介绍了基于 Unity3D 的游戏优化方法概述，举例列出啦 Unity3D 游戏优化的方法，最后通过比较同一张图片压缩成不同格式的美术资源大小变化，以及在 iPhone 平台下不同大小的资源压缩成 PVRTC 格式的数据大小变化，优选压缩率最高的格式作为按系统检索子系统的检索标准。

2.1 Unity3D 技术概述

Unity3D 是由丹麦 Unity 公司开发的游戏开发工具，作为一款跨平台的游戏开发工具，从一开始就被设计成易于使用的产品。支持包括 IOS, ANDROID, PC, WEB, PS3, XBOX 等多个平台的发布。同时作为一个完全集成的专业级应用，Unity 还包含了价值数百万美元的功能强大的游戏引擎^[8]。具体的特性包含整合的编辑器、跨平台发布、地形编辑、着色器，脚本，网络，物理，版本控制等特性^[9]。



图 2-1 unity3D 引擎介绍

Figure 2-1 unity3D engine introductions

Unity 是由 Unity Technologies 开发的一个让玩家轻松创建诸如三维视频游戏、建筑可视化、实时三维动画等类型互动内容的多平台的综合型游戏开发工具，是一个全面整合的专业游戏引擎^[10]。Unity 类似于 Director, Blender game engine, Virtools 或 Torque Game Builder 等利用交互的图型化开发环境为首要方式的软件。其编辑器运行在 Windows 和 Mac OS X 下，可发布游戏至 Windows、Mac、Wii、iPhone、Windows phone

8 和 Android 平台。开发网页端 Unity3D 引擎的游戏推荐使用 UnityWebPlayer 工具。它的网页播放器也被 Mac widgets 所支持。

2.2 基于 Unity3D 的游戏优化方法概述

对项目优化有很多，可以把 mesh 进行整合，对 DrawCall 函数的回调进行删减和对游戏物体的骨骼模型进行必要的减少，对项目的物理内容计算进行相应的缩减等等。

1. 市场上流通的 android 手机中，GPU 的种类多种多样，所有的 GPU 都支持 ETC 贴图格式，但是他们都有各自主要支持的格式，所以我们可以压缩不透明的贴图为 ETC 4bit 格式

2. 对于透明贴图，我们只能选择 RGBA 16bit 或者 RGBA 32bit。

3. 减少 FPS，在 ProjectSetting->Quality 中的 VSync Count 参数会影响你的 FPS，EveryVBlank 相当于 FPS=60，EverySecondVBlank = 30；

4. 若想提高游戏性能，从物理计算次数方面有很大的文章可做，在我们设置了 FPS 以后，对 fixed time step 数据进行修改，目的在于对物理计算的次数的缩减。

5. 为了避免游戏移植到手机上以后，低端手机系统崩溃的现象，在游戏开发过程中对图集选择不要超过 1024 大小。因为若 android 系统的版本不高于 2.2 版本的手机无法读取超过 1000 大小的图集。

2.3 不同系统要求贴图格式的分析

本小节通过对 Android 系统贴图格式和对 iOS 系统贴图格式的研究，分析介绍了每个系统平台上每种格式的优缺点以及最适合的压缩方法。

2.3.1 Android 系统贴图格式分析

对于 Android 系统上很重要的一点，就像压缩一幅 JPEG 图像，便可在磁盘中存储更多图像，对纹理进行压缩也能使更多的纹理装入图形硬件中。

在移动平台中广泛采用的 ETC 或 Ericsson 纹理压缩的开放标准是被支持的。使用 ETC v1 格式为目标压缩格式的纹理通常会有弊端，即所有的 Alpha 通道有关信息信息都会丢失，并且透明区域也会丢失。最有效的改变此现象的做法，就是通常会在纹理中使用 Alpha 通道，因此不少开发人员会采用其他纹理压缩算法，比如很多为硬件支持有限的专有格式。

为了在应用程序中仍然采用 ETC 压缩并存在透明区域，有许多技巧可供选用。本

页阐述这些技巧。

方法一： 纹理拼图

使总体纹理图形更高，可以将 Alpha 通道转换为可见的灰度图像并串联到原始纹理上。

优点：

- a.着色器代码几乎不需要太多改动
- b.文件数量只有一个，这样对纹理加载代码的修改程度降为最低

缺点：

- a.缩放会降低着色器执行速度
- b.纹理样本的包裹只能在一个方向上正确。

操作方法：在纹理压图片缩后，需要更改的只有在着色器中重新映射纹理坐标代码，所以对于实施来讲，这个方法最容易，例如将：

```
gl_FragColor = texture2D(u_s2dTexture, v_v2TexCoord);
```

修改代码为：

```
vec4 v4Colour = texture2D(u_s2dTexture,v_v2TexCoord);  
v4Colour.a = texture2D(u_s2dTexture,v_v2TexCoord *vec2(1.0,0.5)  
+v_v2TexCoord *vec2(0.0,0.5)).r;
```

为了使用图片的上半层，移动到下半层，以取得 Alpha 通道所处的第二样本，这就需要缩放图片纹理的坐标，可以向顶点着色器增加第二可变值，让顶点着色器代码变为如下所示：

```
attribute vec4 a_v4Position;  
attribute vec2 a_v2TexCoord;  
varying vec2 v_v2TexCoord;  
varying vec2 v_v2AlphaCoord;  
void main()  
{  
    v_v2TexCoord = a_v2TexCoord * vec2(1.0, 0.5);  
    v_v2AlphaCoord = v_v2TexCoord + vec2(0.0, 0.5);  
    gl_Position = a_v4Position;}
```

这会更好的利用管线，但这样使略多一点带宽用于额外的 vec2。在这些细小更改下，大部分应用程序都可使用纹理拼图文件正常运行。不过，为了保留将贴图纹理打包到较大区域的能力， 以下将介绍其他两种修改方法。

方法二： 单独包装 ALPHA

Alpha 通道作为第二包装的纹理交付，之后两纹理在着色代码中进行结合。

优点：

- a. 包裹纹理可以有两个方向
- b. 允许混合，匹配 Alpha/颜色通道，变的更灵活。

缺点：

- a. 需要第二个纹理采样器在着色器中。

操作方法：在加载第二个纹理时，请在 `glBindTexture` 和 `glCompressedTexImage2D` 之前调用 `glActiveTexture(GL_TEXTURE1)`，以确保在不同的硬件纹理插槽中分配 Alpha 通道。

```
glActiveTexture(GL_TEXTURE0);
loadCompressedMipmaps(TEXTURE_FILE, TEXTURE_FILE_SUFFIX, &iTexName);
glActiveTexture(GL_TEXTURE1);
loadCompressedMipmaps(ALPHA_FILE, TEXTURE_FILE_SUFFIX, &
iAlphaName);
```

在设置着色器统一变量环节，需第二个纹理采样器，然后捆绑到第二纹理单元上：

```
iLocSampler = glGetUniformLocation(iProgName,"u_s2dTexture");
glUniform1i(iLocSampler,0);
iLocSamplerAlpha = glGetUniformLocation(iProgName,"u_s2dAlpha");
glUniform1i(iLocSamplerAlpha,1);
```

两个样本再次在片段着色器中合并，但与第一次不同，是从不同的纹理执行。

```
vec4 colour = texture2D(u_s2dTexture, v_v2TexCoord);
colour.a = texture2D(u_s2dAlpha, v_v2TexCoord).r;
gl_FragColor = colour;
```

方法 3：单独的原始 ALPHA

作为原始的 8 位单通道图像提供，将 Alpha 通道在着色器中和纹理数据合并。

优点：

- a. 因为有损 ETC1 压缩中会导致失真，所以允许未经压缩的 Alpha。
- b. 允许混合，匹配 Alpha/颜色通道，变的更灵活。

缺点：

- a. 没有压缩的 Alpha 通道比压缩的将占用更多的内存带宽和空间。
- b. 需要第二个纹理采样器在着色器中。

操作方法：本例用新方法加载绑定此纹理，但由于纹理的未压缩本质，加载过程还是相对琐碎的：

```
FILE*pFile=NULL;
unsigned char *pTexData = NULL; unsigned int iWidth = 0;
```

与之前相同，纹理加载可以到独立的活动纹理中：

```
glActiveTexture(GL_TEXTURE0);
loadCompressedMipmaps(TEXTURE_FILE, TEXTURE_FILE_SUFFIX, &iTexName);
glActiveTexture(GL_TEXTURE1);
loadRawLuminance(ALPHA_FILE, &iAlphaName);
```

再次被单独加载到片段着色器中：

```
iLocSampler = glGetUniformLocation(iProgName,"u_s2dTexture");
glUniform1i(iLocSampler,0);
iLocSamplerAlpha = glGetUniformLocation(iProgName,"u_s2dAlpha");
glUniform1i(iLocSamplerAlpha,1);
```

两个样本在片段着色器中从两个不同的纹理进行合并：

```
vec4 colour = texture2D(u_s2dTexture,v_v2TexCoord);
colour.a = texture2D(u_s2dAlpha,v_v2TexCoord).r;
gl_FragColor = colour;
```

2.3.2 iOS 系统贴图格式分析

Zwoptex 是一款把若干资源图片拼接为一张大图的工具 PVR 格式的图片在 IOS 设备上是很常见的格式，通常情况下，因为 IOS 设备上使用的是 PowerVR 显示芯片 PVR 格式，在 iOS 的显示芯片可以直接读取，不用经过解析便可直接显示，所以其更节省内存，渲染速度更快。针对压缩成 PVR 格式实际数据情况如表格 2-1 所示：

表 2-1 针对压缩成 PVR 格式实际数据情况

Table 2-1 for compressed into PVR format of the actual data

测试对象	格式	大小	内存占用	压缩率
空的 cocos2D 模版工程	无	无	4MB	0%
带有一张 PNG 图片的工程	RGBA565	2048*1024	20MB	0%
带有一张 PNG 图片的工程	pvr	2048*1024	16MB	25%

对上述表格进行详细分析：经 cocos2D 官方文档得知：PVR 压缩图像下设 PVRTC2 和 PVRTC4 两种格式。这两种图像格式相比较其他图像格式来说，优点在于图像加载速

度快，图像内存占用低等特点。据计算机对处理图片内存占用的公式： $\text{numBytes} = \text{width} * \text{height} * \text{bitsPerPixel} / 8$ ，假设一张大小为 2048*2048 的 RGBA8888 格式图片，不妨计算一些：原本占用的内存大小为 16MB，经过压缩成 PVRTC4 格式，计算结果只占用 2MB 大小。

a. 一般 pvr 格式文件的图像格式有两种：

RGBA8888: 32-bit texture with alpha channel, best image quality

RGBA4444: 16-bit texture with alpha channel, good image quality

RGB565: 16-bit texture without alpha channel, good image quality but no alpha (transparency)

b. PVRTC 文件包含的图像格式有两种：

PVRTC4: Compressed format, 4 bits per pixel, ok image quality

PVRTC2: Compressed format, 2 bits per pixel, poor image quality

经查阅参考资料以及设计操作得知：

1、因为决定图片占用内存的因素是像素格式和大小，不是其扩展名。所以只要 png8 png32 jpg pvr 的像素格式都是 argb8888，那么最终图片占用的内存是相同的。

2、图片越小，占用内存越小，所以除非是 pvr4 的格式，就不要扩展成 2 的整次幂。

3、建议 rgb565 和 rgb5551 来替代 rgb888 的格式，因为仅去除透明通道不可以减少图片所消耗的内存，png 和 jpg 图片也无法减少图片体积。

4、可以考虑加入内存池，及时的开辟和释放图片加载时临时开辟的纹理数据造成的内存快速大幅度升高的缓冲区。

5、减少体积：jpg 和 png8 这两种图片格式均可以大幅度减少图片体积，但几乎不可以减少内存。

6、减少内存：1、没有透明色的图片：推荐 pvr.ccz 的 rgb565 格式，pvr.ccz 速度较快。2、如果透明色仅进行关键色标注而无渐变混合：推荐 rgb5551 的 pvr.ccz 格式

2.4 Unity3D 贴图压缩格式方法概述

对 Texture 贴图占用内存和空间进行详细分析。很多程序员在开发游戏项目的时候其实对游戏资源的大小、格式和内存占用的具体细节并不知情，这是因为 Unity3D 没有对游戏资源的处理方式进行很好的说明。在这里对 Unity3D 图片压缩方面进行实际的压缩，并对压缩的数据进行比较分析。在资源打包前和资源打包后内存和大小的问题在之前的项目中频频出现，所以在这对 Unity3D 对 Texture 处理的方式进行详细分析。

Nvidia Tegra 提供了 DXT 格式，android 系统都会支持 ETC 格式。ETC2 格式只有 OPENG3.0 才支持，PVRTC 是 Imagination PowerVR 提供的，Qualcomm Snapdragon 提供了 ATC 格式。IOS 系统一般只支持 PVRTC 的压缩格式。在游戏开发过程中如果贴图格式出现不兼容的情况，Unity3D 引擎会把贴图格式自动的转为 RGB (A)。据统计，最科学的兼容情况是根据 GPU 的种类进行资源打包操作，但在游戏开发的时候无法细化到针对 GPU 种类进行打包操作。所有的手机设备都很好支持 RGB 16BITS/ARGB 16BITS/RGB A16BITS/RGB 24BITS/ARGB 32BITS 等，但这些格式都属于非压缩格式，对内存和渲染方面消耗非常大^[11]。

2.4.1 图片压缩格式的比较

压缩成不同格式的 Texture 在储存上的大小变化比较，假设高清 (ARGB32) 大小所占储存为 1M，那么压缩后所占储存变化大概数据如表 2-1 所示：

表 2-1 压缩成不同格式的图片储存大小变化

Table 2-1 Compressed into different formats of image storage size change

压缩前格式	目标格式	大小变化
ARGB32	RGB PVRTC 4BITS	1 → 0.25
ARGB32	ARGB PVRTC 4BITS	1 → 0.25
ARGB32	RGB PVRTC 2BITS	1 → 0.13
ARGB32	ARGB PVRTC 2BITS	1 → 0.13
ARGB32	RGBA ETC2 4BITS	1 → 0.25
ARGB32	RGBA ETC2 8BITS	1 → 0.25
ARGB32	RGB + 1-bit ALPHA ETC2 8BITS	1 → 0.2
ARGB32	DXT1	1 → 0.3
ARGB32	DXT5	1 → 0.6
ARGB32	ARGB 16BITS	1 → 0.33
ARGB32	RGB 16BITS	1 → 0.5
ARGB32	RGB 24BITS	1 → 0.85
ARGB32	ARGB 32BITS	1 → 0.1

压缩成不同格式的 Texture 在内存上的大小变化比较，假设高清 (ARGB32) 大小所占内存为 1M，那么压缩后所占内存变化大概数据如表 2-2 所示：

表 2-1 压缩成不同格式的图片内存大小变化

Table 2-1 compressed into different formats of pictures memory size change

压缩前格式	目标格式	大小变化
ARGB32	RGB PVRTC 2BITS	1 → 0.0625
ARGB32	ARGB PVRTC 2BITS	1 → 0.0625
ARGB32	RGB PVRTC 4BITS	1 → 0.125
ARGB32	ARGB PVRTC 4BITS	1 → 0.125
ARGB32	RGBA ETC2 8BITS	1 → 0.25
ARGB32	RGBA ETC2 4BITS	1 → 0.125
ARGB32	RGB + 1-bit ALPHA ETC2 8BITS	1 → 0.125
ARGB32	DXT1	1 → 0.125
ARGB32	DXT5	1 → 0.25
ARGB32	ARGB 16BITS	1 → 0.5
ARGB32	RGB 16BITS	1 → 0.5
ARGB32	RGB 24BITS	1 → 0.8
ARGB32	ARGB 32BITS	1 → 0.1

另一种分析方法如下所示：在压缩 10 张 iOS 系统平台支持的 PVRTC 格式、空项目和无压缩的项目数据比较，项目空间占用大小如表 2-3 所示。

表 2-3 多种项目类型压缩大小数据

Table 2-3 Compression of multiple item types size data

打包类型	单张占用量	空间占用量	IPA 大小
空项目	0.0MB	42.3MB	10.0MB
10 张 1200*520 无压缩 Texture	2.8MB	70.2MB	22.9MB
10 张 1200*520 压缩成 1024*1024 PVRTC4	0.5MB	47.3MB	13.2MB
10 张 1024*1024 无压缩 Texture	4.0MB	82.3MB	14.6MB
10 张 1024*1024 压缩为 PVRTC4	0.5MB	47.3MB	11.6MB

2.4.2 图片压缩格式的选择依据

据 OpenGL 官方文档解释，在硬件设备上 CPU、GPU 和 DISPLAYER 三个硬件模块，使用图 2-3 所介绍的方式显示资源。首先 CPU 将经过计算完成的内容发送到 GPU，然后 GPU

负责渲染和处理等工作，完成后将把结果置入帧缓冲区内，最终被置入帧缓冲区的数据被视频控制器一行一行的按照 VSync 信号读取，如图 2-3，依据读取的数据进行数模转换，可行的数模转换后最终传交给显示器，在显示器上显示。

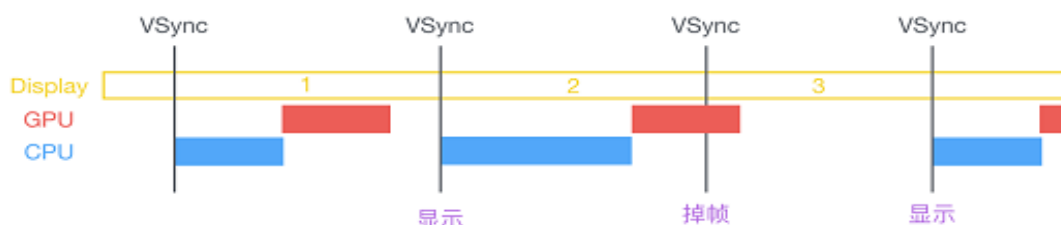


图 2-3 处理器、内存和显示器写作方式图

Figure 2-3 The writing way of CPU Memory and Displayer

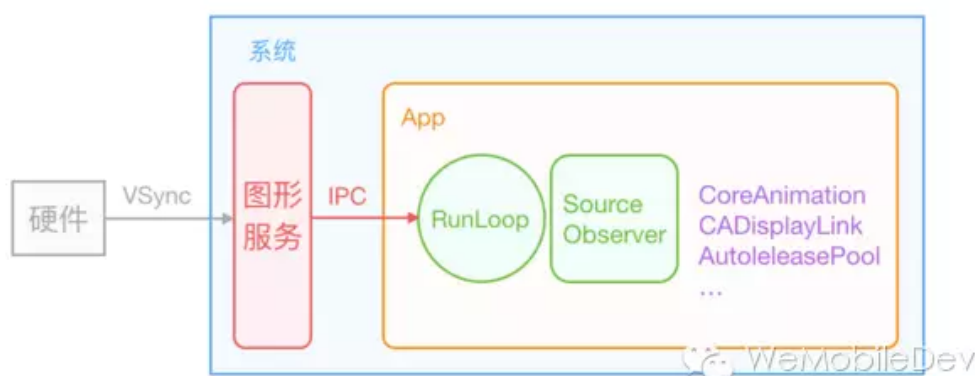


图 2-4 图像处理模块结构图

Figure 2-4 image processing module structure

在感知到 VSync 信号后,系统图形服务利用 CADisplayLink 方式告知 App,App 主线程得到通知后便开始在 CPU 中计算显示内容,比如布局计算、文本绘制、视图的创建、图片解码等。随后计算好的内容会被 CPU 会将传递给 GPU 来完成变换、合成、渲染过程,然后渲染结果被传递进入帧缓冲区,等待下一次 VSync 信号的到来,从而显示在屏幕上。但是因为垂直同步的机制的存在,在一个 VSync 时间内,如果 CPU 或 GPU 没有保证内容传交过程的完成,那么那一帧便会丢失,等下一次机会才能显示,而与此同时显示屏上显示的内容会与之前的内容一样。导致了界面卡顿。由上述分析得出,不管是 CPU 还是 GPU 的原因影响了显示过程,均会导致掉帧现象。所以,在游戏开发过程中,设置对 CPU 和 GPU 压力进行评估和优化环节至关重要。

GPU 资源消耗原因及解决方案: 全部所有包括图片、文本、栅格化的内容的 Bitmap, 由内存提交到显存, 绑定为 GPU Texture。不管是全部提交的过程, 还是 GPU 调整和渲染 Texture 的过程, 都要消耗很多 GPU 资源。若想使大量图片在较短时间显示, CPU 占用率低, GPU 占用高, 界面会出现掉帧现象, 为避免这种情况, 可以采用尽量减少图片大小以提高 GPU 的运行效率, 减少图片大小的方式最优的莫过于对图片进行压缩, 压

缩成格式最合适的图片资源是第一要考虑的事情。

结论：依据上述客观实例举证，若想使大量图片在较短时间显示，保证游戏产品在不同的系统平台，以及不同水平的硬件设备上流畅运行，必须从 GPU 的使用方面酌情考虑，如何减少 GPU 运行效率由图片所占内存大小所决定，那么以要压缩效率最高的目标格式的选择，至关重要。

2.4.3 图片压缩格式的确立

从一般的 3D 手机游戏来讲，在压缩游戏资源的时候，要把贴图种类按 UI 资源、模型资源和场景资源几个方面考虑，主要原因在于多重采样的机制存在。这种游戏采样大小会受场景摄像机的移动而改变，原理在于场景近大远小，这样的结果会产生处于远处的贴图会有大量白色噪点^[12]。

关于贴图的形状必须要加以区分。非正方贴图只有 16 位的压缩，所以最好游戏中都是正方的贴图。以下是贴图压缩格式遵循的一些规则：

1. 正方贴图：

IOS：

- a. 普通不透明：RGB PVRTC 4BITS
- b. 普通透明：RGBA PVRTC 4BITS

Android：

- a. 普通不透明：RGB ETC 4BITS
- b. 普通透明：不存在最兼容的通用贴图格式，大都采用 RGBA 16BIT 格式贴图

2. 非正方贴图：

- a. 不透明贴图：RGB 16BITS
- d. 透明贴图：RGBA 16BITS

3. 高清不压缩贴图：RGBA 32BIT

不要对滴模糊度和非重要贴图进行像素压缩的方法打包资源，最好的方法是将其之间进行压缩。

总结：

- 1. 手机引擎更喜欢 2 的 N 方大小的贴图，其中包含内存方面的消耗，压缩贴图比例，打包压缩资源的大小
- 2. 从以上数据可以看出，改变图片的大小，改变图片色彩位数还有压缩格式的选择都会直接或者间接有效减少图片对占用大小和内存，
- 3. Unity3D 会自动处理贴图格式，并不受程序开发人员提交的贴图格式而影响，
- 4. 由于 Unity3D 游戏引擎内部处理机制的存在，引擎会自动生成符合要求的图片类

型以替换游戏开发程序员所提供的格式图片，所以我们需要谨慎的选择图片压缩的方式。

5. 最为保真的无压缩的贴图格式为 RGBA32，同样最保真意味着最占用手机的空间和内存，Automatic Turecolor 和 RGBA32 拥有着同样的保真效果和大量的空间内存占用。

6. 通常意义上我们理解的无压缩的 16 色彩位贴图格式就是 RGBA16 格式，RGBA16 要比 RGBA32 节省大约一般的左右的存储空间以及内存空间，Automatic 16bits 和 RGBA16 拥有着同样的保真效果和空间内存占用。

2.5 本章小结

本章详细的描述了手机游戏后期优化美术资源管理系统的技术途径的优选以及原理分析，主要从对不同系统要求贴图格式的分析和 Unity3D 贴图压缩格式方法的概述两个方面重点阐述了本人对与此论文的原理的分析和路线的选择，为接下来的需求分析以及系统实现奠定了基础。

第三章 游戏美术资源管理系统的需求分析

本章首先从系统功能性分析入手进行论述，为后期的系统开发准备必要的资源并降低不必要的风险；然后从用户行为分析、资源管理用户分析，图集管理用户分析，按硬件检索用户行为分析、按 Android 系统检索用户行为分析、按 iOS 系统检索用户行为分析、压缩资源大小用户行为分析以及压缩资源格式用户资源分析八个方面重点介绍了系统功能性分析。从而为建立安全的、高效的、并且可以长期稳定运行的手机游戏后期美术资源管理系统打下坚实的基础，最后简要的从系统的可靠性、实用性、易用性和可扩展性对本系统做了非功能性系统需求分析。

3.1 系统功能性需求分析与定义

本小节重点介绍系统功能上的需求分析，把游戏项目开发程序员的游戏后期优化工作的目标划分为七个大类：安卓平台的资源管理、iOS 平台的资源管理、根据硬件限制资源管理、图集管理、资源管理、压缩资源大小以及压缩资源格式；然后按顺序对这七个大类主要目标行为进行分析，并分别为他们指定子功能模块，总之，系统中设计以下七类用户，如图 3-1 所示：

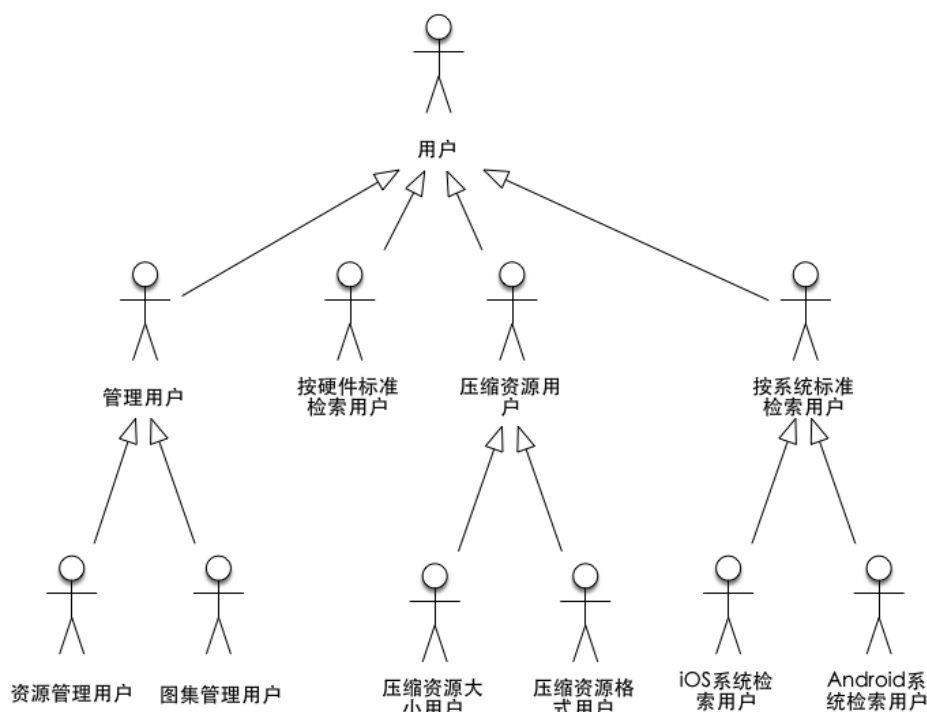


图 3-1 用户继承关系图

Figure 3-1 User inheritance relationship diagram

3.1.1 用户行为分析

用户行为分析是对我们单位的所有游戏开发程序员的优化工作进行调查，并详细记录他们的工作流程。然后，我们对所有程序员行为进行类别划分，把行为表现相似的程序员划分到同一个类别之下。

我们着重调查了我们单位的两位主要程序员。发现他们的工作比较简单，主要涉及对视频资源，模型资源，声音资源，图片资源，动画资源还有代码资源（减少 drawcall 的数量）进行优化，其中重要的部分优化就是对图片资源进行优化，图片资源的灵活性很强，是其他资源所不具备的，声音、动画和模型资源一般都是给定好的，修改性很小，然而代码资源的优化效率很低，投入很长时间却并不能很大程度上提高运行效率。

最后我们把工作任务放在了关键地方，就是对图片资源进行优化，设计出一套美术资源管理系统，对图片资源优化的程序员工作任务比较繁重，我们仔细总结，大致如下：审查图片格式是否是最优化的而且是否是符合标准的，unity3D 对图片资源的压缩格式是多种格式的，透明的图片和不透明的图片，android 平台的 ETC 格式和 iOS 平台的 PVRTC 和 PNG 格式不能相互兼容的，还有发布游戏的版本是对图片大小也是有很高的要求。综上所述，我们把手机游戏后期负责优化的程序员的行为划分为以下 8 类，如表 3-1 所示。

表 3-1 用户行为分析
Table 3-1 User behavior analysis

用户按行为分类	操作行为
登录系统	登陆系统
	注册账户
	修改个人信息
资源管理	添加资源
	更新资源
	查看指定资源信息
	删除资源
图集管理	新建图集
	修改图集信息
	查看指定图集信息
	删除图集
按硬件进行资源检索	选择硬件检索模块
	选择检索图集
	设定检索资源的大小限定值
	进行资源检索
	接受反馈信息

表 3-1 用户行为分析（续）

Table 3-1 User behavior analysis (continue)

用户按行为分类	操作行为
Android 平台支持格式检索	选择按系统要求检索模块
	选择 Android 平台检索模块
	选择检索图集
	设定检索资源的格式要求 ETC
	进行资源检索
	接受反馈信息
iOS 平台支持格式检索	选择按系统要求检索模块
	选择 iOS 平台检索模块
	选择检索图集
	设定检索资源的格式要求 pvr 和 png
	进行资源检索
	接受反馈信息
压缩资源大小用户	选择压缩资源模块
	选择压缩资源大小模块
	选择图集资源
	输入目标大小
	进行资源大小压缩
	接受反馈信息
压缩资源格式用户	选择压缩资源模块
	选择压缩资源格式模块
	选择图集资源
	选择压缩的目标格式
	进行资源格式压缩
	接受反馈信息

3.1.2 资源管理系统功能性需求定义

资源管理用户是对美术资源进行系统性的管理，主要包含一下几个功能点：添加美术资源，更新美术资源，查看指定美术资源信息，删除美术资源。用例图如图 3-2 所示：

（1）添加美术资源，用户登录后可手动向数据库中指定添加美术资源，添加成功后数据库会存储指定美术资源的基本信息，只有添加到数据库中的美术资源才可以进行美术资源的修改，查看，删除和按各种标准进行检索。

（2）更新资源，此功能可以更新存储在数据库中的美术资源信息，包括上传时间，名称，所在图集等基本信息。

（3）查看指定美术资源信息，此功能可以给用户提供美术资源的所有信息以使用

户接下来对美术资源的准确操作。

（4）删除美术资源，用户可以从数据库中删除指定美术信息，资源一旦删除便无法恢复。

3.1.3 图集管理用户行为分析

图集管理用户是对图集进行系统性的管理，主要包含一下几个功能点：新建图集，修改图集信息，查看指定图集信息，删除图集。用例图如图 3-3 所示：

（1）新建图集，用户登录后可手动向数据库中存储一个新的图集，图集用来盛放一个美术资源的集合，便于用户对大量美术资源进行管理，为检索美术资源奠定基础，新建成功后数据库会存储图集的基本信息。

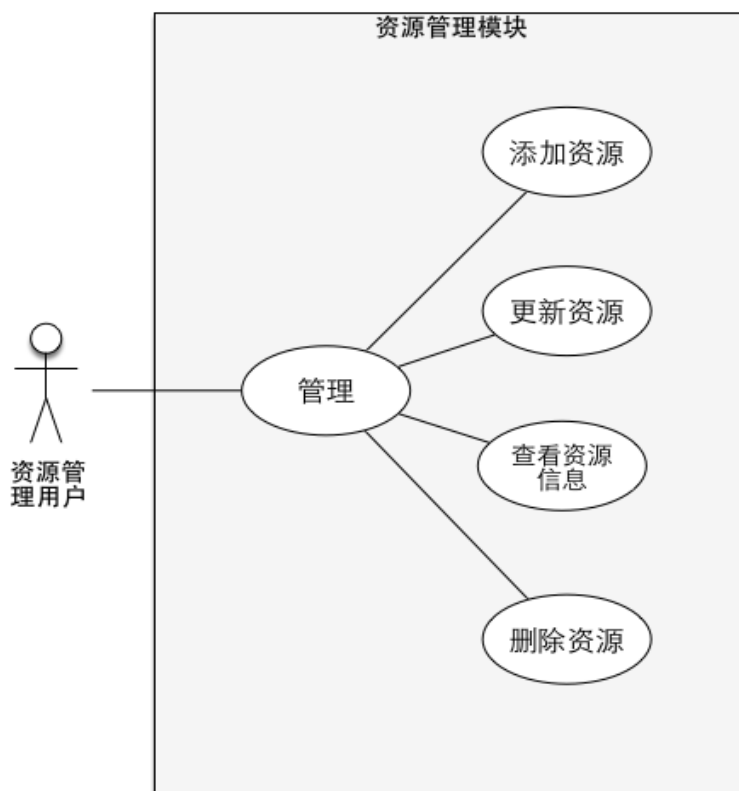


图 3-2 资源管理用户用例图

Figure 3-2 Resource management user use case diagram

（2）修改图集信息，此功能可以修改存用户储在数据库中的图集信息，包括图集名称和备注等基本信息。

（3）查看指定图集信息，此功能可以给用户提供浏览图集的所有信息的功能，以便用户对接下来美术资源的准确操作。

（4）删除图集，用户可以从数据库中删除指定图集，图集中的所有美术资源也一并被消除，资源一旦删除便无法恢复。

3.1.4 按硬件检索用户行为分析

按硬件需要对美术资源进行检索的用户主要是对美术资源的大小进行检索并反馈信息，主要包含以下几个功能点：选择硬件检索模块，选择检索图集，设定检索资源的大小限定值，进行资源检索。用例图如图 3-4 所示：

（1）选择硬件检索模块，用户进入本系统后会选择根据硬件检索还是根据系统平台进行检索功能，硬件检索会根据下一步用户给定的标准大小进行检索是否有超标资源，这一功能是本系统最重要的分支功能之一。

（2）选择检索图集，用户在使用此功能前需要向服务器中上传所有需要检索的美术资源，用户可以把上传的图片分图集保存，进行检索的时候需要选择图集进行检索。

（3）设定检索资源的大小限定值，据单位提供的数据：为了避免游戏移植到手机上以后，低端手机系统崩溃的现象，在游戏开发过程中对图集选择不要超过 1024 大小。因为若 android 系统的版本不高于 2.2 版本的手机无法读取超过 1000 大小的图集。以上两条信息介绍了单一图片的大小要求还有图集的大小要求，这两个要求是硬性要求，手机游戏开发程序员也也可以自己根据需求拟定图片大小限制以进行美术资源检索。

（4）进行资源检索，用户经过之前选择和设定检索标准后，系统会按照用户提供的图集信息和限定的检索标准对，本图集进行检索，把不符合检索标准的美术资源筛选出来。信息检索完成后，用户会接收到系统反馈给用户的信息，即被检索的美术资源中有哪些美术资源是不符合用户给定标准的，不合格的信息的名称会被展示到系统的文本框内，供用户参考。

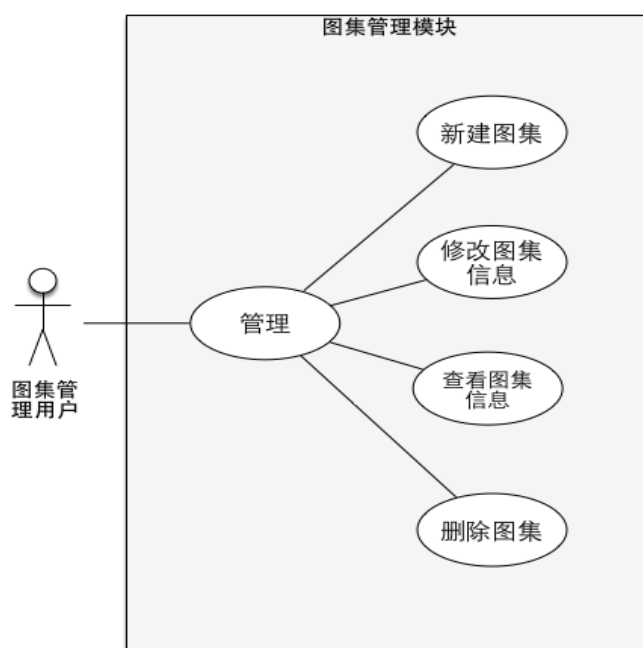


图 3-3 图集管理用户用例图

Figure 3-3 Atlas management user use case diagram

3.1.5 按 Android 系统检索用户行为分析

按 Android 系统格式要求对美术资源进行检索的用户是对美术资源的格式要求进行检索并反馈信息，主要包含以下几个功能点：选择按系统要求检索模块，选择 Android 系统检索模块，选择检索图集，设定检索资源的格式要求 ETC，进行资源检索。用例图如图 3-5 所示：

（1）选择按系统要求检索模块，进入系统后需要选择按硬件检索还是按系统进行检索，选择按系统进行检索将会使系统重点检查图片的格式而非大小，选择按系统进行检索下设两个模块分别为，按 Android 系统和 iOS 系统进行检索，两个系统平台对图片格式需求是不同的，每种平台要求的格式并不唯一。

（2）选择按 Android 系统检索模块，用户进入本系统后会选择根据硬件检索还是根据系统平台进行检索功能，选择按系统检索模块后，系统还会要求用户对检索 iOS 系统或者 Android 系统的美术资源做出选择。

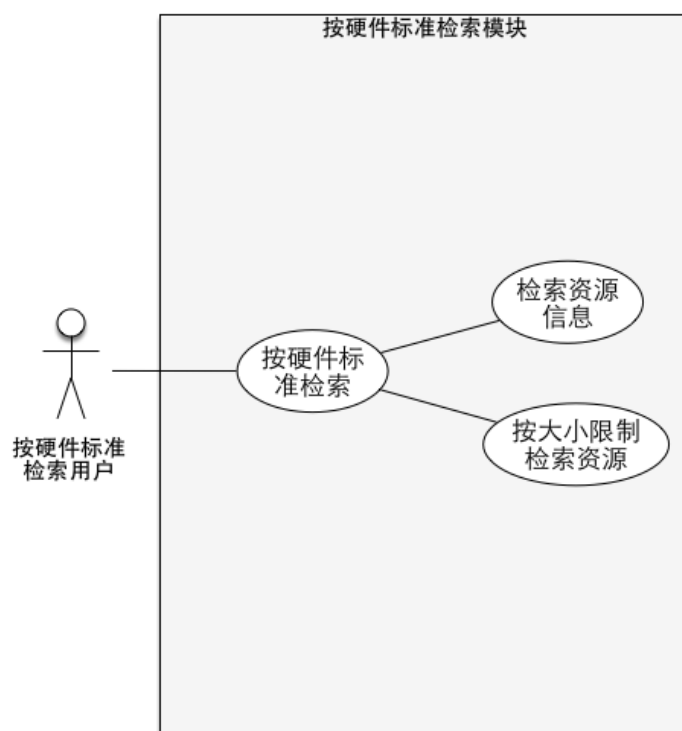


图 3-4 按硬件检索用户用例图

Figure 3-4 According to hardware retrieval user use case diagram

（3）选择检索图集，用户在使用此功能前需要向服务器中上传所有需要检索的美术资源，用户可以把上传的图片分图集保存，进行检索的时候需要选择图集进行检索。

（4）设定检索资源的格式限定，经过调研，我们发现安卓上硬件支持最广泛的格式是 ETC 格式，ETC 格式为压缩效率较高的格式，安卓手机绝大部分又是原声支持此压

缩格式的图片，所以按 Android 平台检索美术资源格式其实就是检测是否美术资源都是 ETC 格式。

（5）进行资源检索，用户经过之前选择和设定检索标准后，系统会按照用户提供的图集信息和限定的检索标准对，本图集进行检索，把不符合检索标准的美术资源筛选出来。信息检索完成后，用户会接收到系统反馈给用户的信息，即被检索的美术资源中有哪些美术资源是不符合用户给定标准的，不合格的信息的名称会被展示到系统的文本框内，供用户参考。

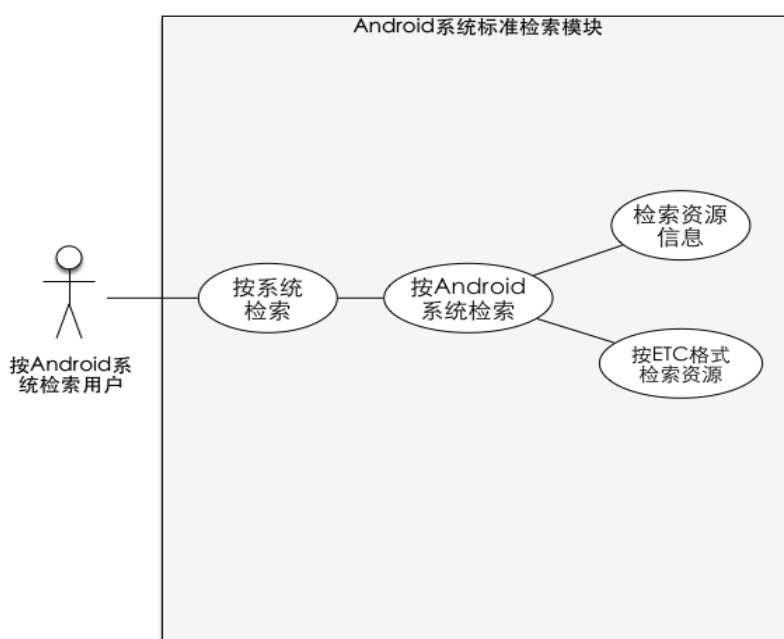


图 3-5 按 Android 系统检索用户用例图

Figure 3-5 According to Android system retrieval user use case diagram

3.1.6 按 iOS 系统检索用户行为分析

按 iOS 系统格式要求对美术资源进行检索的用户是对美术资源的格式要求进行检索并反馈信息，主要包含一下几个功能点：用户登陆，用户注册，选择按系统要求检索模块，选择 iOS 平台检索模块，选择检索图集，设定检索资源的格式要求 pvr tc，进行资源检索。用例图如图 3-6 所示：

（1）选择按系统要求检索模块，进入系统后需要选择按硬件检索还是按系统进行检索，选择按系统进行检索将会使系统重点检查图片的格式而非大小，选择按系统进行检索下设两个模块分别为，按 Android 系统和 iOS 系统进行检索，两个系统平台对图片格式需求是不同的，每种平台要求的格式并不唯一。

（2）选择按 iOS 系统检索模块，用户进入本系统后会选择根据硬件检索还是根据

系统平台进行检索功能，选择按系统检索模块后，系统还会要求用户对检索 iOS 系统或者 Android 系统的美术资源做出选择。

（3）选择检索图集，用户在使用此功能前需要向服务器中上传所有需要检索的美术资源，用户可以把上传的图片分图集保存，进行检索的时候需要选择图集进行检索。

（4）设定检索资源的格式限定，经过调研发现 Phone 的图形芯片（PowerVR MBX）对一种称为 PVRTC 的压缩技术提供硬件支持，当使用 pvrtec 格式的图片时与标准 JPEG 或 PNG 图像相比，有些图像的质量有可能下降，但使用 pvrtec 压缩图片可以节省大量的内存空间。所以按 iOS 平台检索美术资源格式其实就是检测是否美术资源都是 pvr 格式。

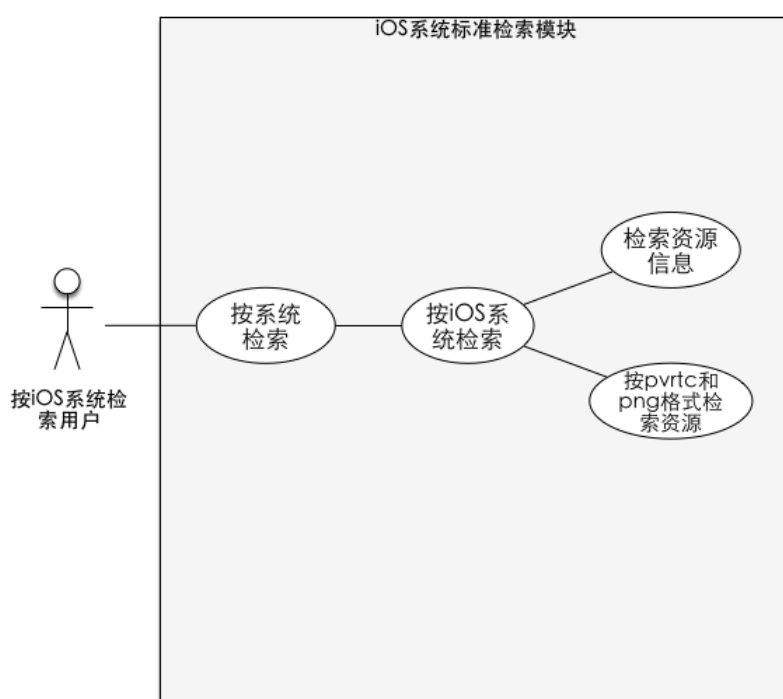


图 3-6 按 iOS 系统检索用户用例图

Figure 3-6 According to iOS system retrieval user use case diagram

（5）进行资源检索，用户经过之前选择和设定检索标准后，系统会按照用户提供的图集信息和限定的检索标准对，本图集进行检索，把不符合检索标准的美术资源筛选出来。信息检索完成后，用户会接收到系统反馈给用户的信息，即被检索的美术资源中有哪些美术资源是不符合用户给定标准的，不合格的信息的名称会被展示到系统的文本框内，供用户参考。

3.1.7 压缩资源大小用户行为分析

压缩资源大小模块主要是用户根据需求对资源大小进行修改压缩，主要包含以下几个功能点：选择压缩资源模块，选择压缩资源大小模块，选择图集集中的资源，输入目标大小，进行资源大小压缩。

（1）选择压缩资源模块，用户进入本系统后会选择压缩资源模块，硬件压缩目标大小会根据下一步用户输入数值来确定，这一功能是本系统最重要的分支功能之一。

（2）选择压缩资源大小模块，用户在使用此功能前需要向服务器中上传所有需要检索的美术资源，用户可以把上传的图片分图集保存，进行压缩的时候需要选择图集之中的资源进行压缩。

（3）选择图集集中的资源，用户需选择服务器上现有图集集中的资源，下一步操作将对此资源进行修改大小。

（4）输入目标大小，据单位提供的数据：为了避免游戏移植到手机上以后，低端手机系统崩溃的现象，在游戏开发过程中对图集选择不要超过 1024 大小。因为若 android 系统的版本不高于 2.2 版本的手机无法读取超过 1000 大小的图集。以上两条信息介绍了单一图片的大小要求还有图集的大小要求，这两个要求是硬性要求，手机游戏开发程序员也可以自己根据需求设定压缩图片的目标大小。

（5）进行资源大小压缩，用户经过之前选择和目标压缩大小后，用户进行图片大小的压缩，系统会到服务器中找到指定资源，并对其大小进行修改。资源压缩完成后，用户会接收到系统反馈给用户的信息，即压缩是否成功。

3.1.8 压缩资源格式用户行为分析

压缩资源格式模块主要是用户根据需求对资源格式进行修改压缩，主要包含以下几个功能点：选择压缩资源模块，选择压缩资源格式模块，选择图集集中的资源，输入目标大小，进行资源大小压缩。

（1）选择压缩资源模块，用户进入本系统后会选择压缩资源模块，硬件压缩目标大小会根据下一步用户输入数值来确定，这一功能是本系统最重要的分支功能之一。

（2）选择压缩资源格式模块，用户在使用此功能前需要向服务器中上传所有需要检索的美术资源，用户可以把上传的图片分图集保存，进行压缩的时候需要选择图集之中的资源进行压缩。

（3）选择图集集中的资源，用户需选择服务器上现有图集集中的资源，下一步操作将对此资源进行修改格式。

（4）选择目标格式，经过调研，ETC 格式为压缩效率较高的格式，Android 平台

检索美术资源格式其实就是检测是否美术资源都是 ETC 格式。使用 pvrtec 压缩图片可以节省大量的内存空间，所以按 iOS 平台检索美术资源格式其实就是检测是否美术资源都是 pvr 格式。用户可以选择这两个格式对资源进行压缩。

（5）进行资源格式压缩，用户经过之前选择资源和目标格式后，用户进行图片格式的压缩，系统会到服务器中找到指定资源，并对其进行压缩。资源压缩完成后，用户会接收到系统反馈给用户的信息，即压缩是否成功。

3.2 系统非功能性需求分析与定义

手机游戏后期优化美术资源管理系统遵循高内聚低耦合的思想，提高了系统的可维护性，便于软件工程化管理，本系统采用了 MVC 架构。为了更加方便的对应用程序的数据层和业务规则进行修改，我们在保证系统功能性需求完备的前提下，还必须考虑系统的非功能性需求。我们主要从以下几个方面考虑：

（1）系统实用性

手机游戏后期优化美术资源管理系统的开发首先将实用性放在首位，能够满足游戏开发程序员使用计算机技术对美术资源进行科学、规范、高效管理的要求。本系统采用传统的面向过程的开发方法与面向对象的软件开发相结合，在获取详细的需求分析后，利用 Java 开发语言和数据库管理系统软件，建立起一个相对功能完备的系统，再根据用户体验和系统测试的结果，经过修改以及回归测试后，形成较为完善的版本，构造一个操作方便、美观实用的管理信息系统。

（2）可靠性要求

手机游戏后期优化美术资源管理系统在运行过程中不应该发生系统故障，在操作产生问题造成系统崩溃的时候，必须具有数据保护性，强稳定性，不应该造成系统崩溃，并且在系统崩溃以后不应造成数据丢失等现象。

（3）系统可扩展性

可扩展性是一个资源管理系统完备性的必要特性，我们不可能开发一成不变的应用系统，我们开发的应用系统也不可能一直完全可以满足日常的工作需求。因此系统的可扩展性在评价一套应用系统时占据着重要的地位。我们在开发手机游戏后期美术资源管理系统时，同样要求该套系统具有良好的可扩展性，为以后公司发展带来的功能上的变化做好准备。

（4）系统易用性

对于使用本系统的程序员来讲，其对计算机及系统的理解能力及处理能力都不尽相同，由此，系统在设计过程中必须充分考虑到程序员之间的计算机操作水平的差距，所以，系统必须满足简单易用性原则。从而使得只要懂得计算机简单操作、能够上网即可

对系统进行操作从而完成美术资源的管理。

3.3 本章小节

本章内容主要从系统功能性分析以及系统非功能性分析这两个方面分析了系统的需求。首先从从用户行为分析入手重点介绍了系统功能性分析；最后简单的介绍一下系统非功能性分析，对本系统的需求进行了严谨细致的剖析，为以后本系统的概要设计和详细设计奠定了扎实的基础。

第四章 游戏美术资源管理系统的概要设计

本章重点介绍手机游戏后期优化美术资源管理系统的体系架构设计、功能结构设计及数据库设计的过程。其中，数据库设计的部分以图表和文字描述的方式，具体展示了了数据库的概念结构设计、数据库逻辑结构设计以及数据的物理结构设计等三部分。

4.1 系统功能结构设计

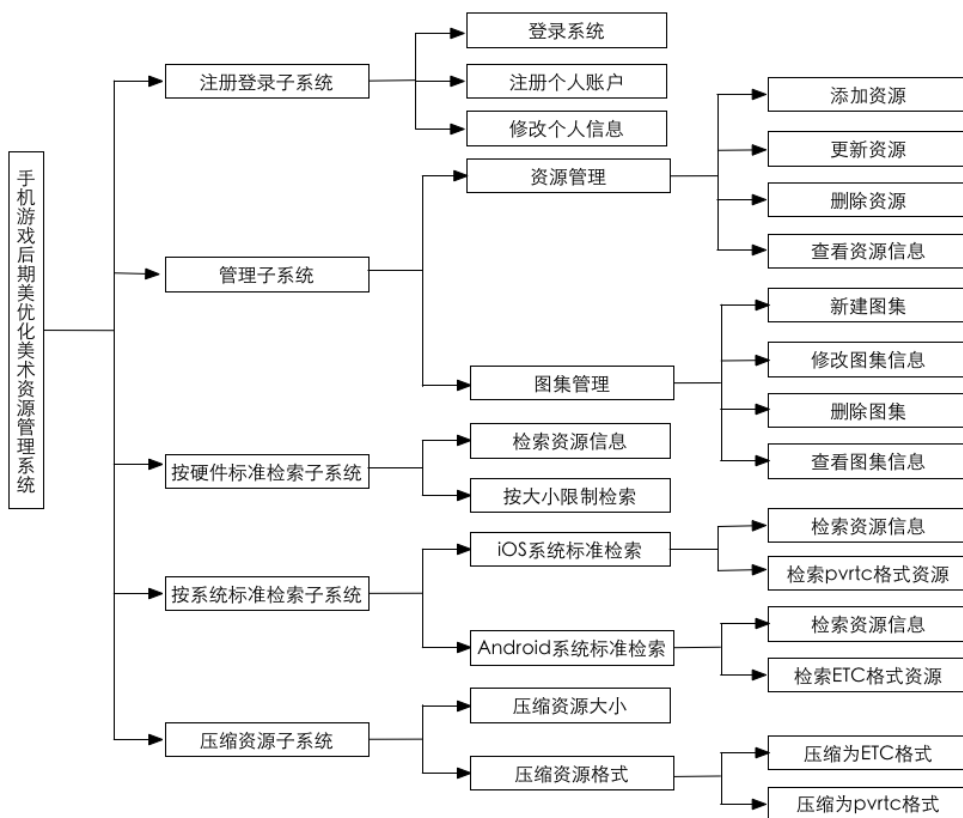


图 4-1 系统功能结构设计图

Figure 4-1 System function structure design

根据前期的功能性需求分析，我们可以把美术资源管理系统划分为五个子系统：注册登录子系统，主要功能是负责用户可以正常的使用系统，并对系统用户进行统计管理；管理子系统，本子系统可分为资源管理模块和图集管理模块，主要功能是使用户可以对美术资源和图集进行管理，实现对美术资源和图集的增删改查功能；按硬件标准检索子系统，主要功能是按照用户给定地址和用户给定资源大小的限制进行数据的检索，并反馈信息；按系统标准检索子系统，主要功能是分别根据用户需求区分 Android 系统标准或者 iOS 系统标准，然后分别进行 ETC 格式资源的检索、资源信息检索、反馈信息功能和 PVRTC 格式资源的检索、资源信息的检反馈检索信息功能。压缩资源子系统，本系

统的功能就是对指定美术资源进行压缩，用户可以分别进行美术资源大小的压缩和格式的压缩。

4.2 系统体系架构设计

手机游戏后期优化美术资源管理系统采用模型-视图-控制器（Model-View-Controller，MVC）体系结构进行设计。

MVC—Model View Controller，是模型(Model)、视图(View)和控制器(Controller)的缩写，它是我在学校软件工程第一个接触到的设计模式。MVC 系统架构的目标在于实现系统职能方面的分工，系统的 View 层用来与用户进行交互；其中 Model 层实现系统中的业务逻辑；Controller 层则是 Model 与 View 之间的桥梁。

模型（Model）用于表示业务逻辑和业务数据，是处理业务的实体，是这个程序的数据核心，它将与业务逻辑相关的数据还有对数据的处理方法进行封装，model 层的内存操作机制是封闭的，对其他两个层级处于不可见状态，并且 model 层模型并不依赖于 View 层和 Controller 层，模型和视图可以是一对多的关系，视图可以了解到在数据模型上发生的任何改变。

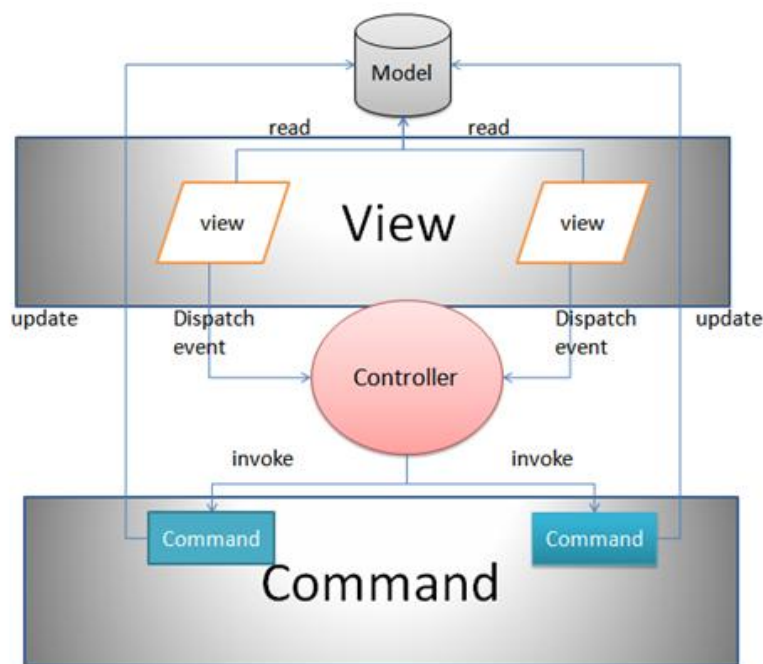


图 4-2 MVC 系统架构原理图

Figure 4-2 MVC system architecture schematic

视图（View）负责与用户进行交互的部分，是用户看到的系统并与之交互的界面，它本身不负责任何业务逻辑和对模型的控制，但是它能够让系统实现数据有目的显示。一个系统模块可能有很多不同的视图和不同的展现手段，但不管视图展现的形式如何，MVC 仅对 view 层上的数据进行采和处理，并不对视图上的业务和流程进行处理，用户通过对视图中控件的操作以向程序下层发出请求。视图一般没有程序上的逻辑，它只关注

于数据信息的显示而不关心数据的处理，这样就实现了数据处理与数据展示的分层。

控制器（Controller）是 Model 和 View 之间的沟通的桥梁，对视图和模型的交互进行协助，控制器被用来对应用程序的流程进行控制，接受系统用户输入的数据，调用 model 和 view 去操作系统以实现用户的指令。一方面它用来接收来自视图的指令，从而进行对模型的操作，另一方面，一旦模型发生改变时，它会将数据模型发生的变化通知给视图，以改正数据在界面上的显示。当用户通过视图传达一个指令，控制器负责将这个指令传递给适当的模型进行处理，从模型返回的数据经由控制层交给合适的视图展示给用户。概括来说是：控制器的功能决定了用户的指令由某个模型处理、数据由某个视图展示，它只处理事件并做出响应。图 4-2 对 MVC 框架做了简要描述。

本系统的视图层针对不同功能进行设计，硬件模块和系统模块相互独立，系统模块视图下设不同系统需求的模块视图，每一个视图对应系统的一个功能模块，不同行为的用户通过对不同视图中的控件进行操作，给控制器发出唯一指令，控制器对指令进行传输，以达到对模型进行修改更新的目的。

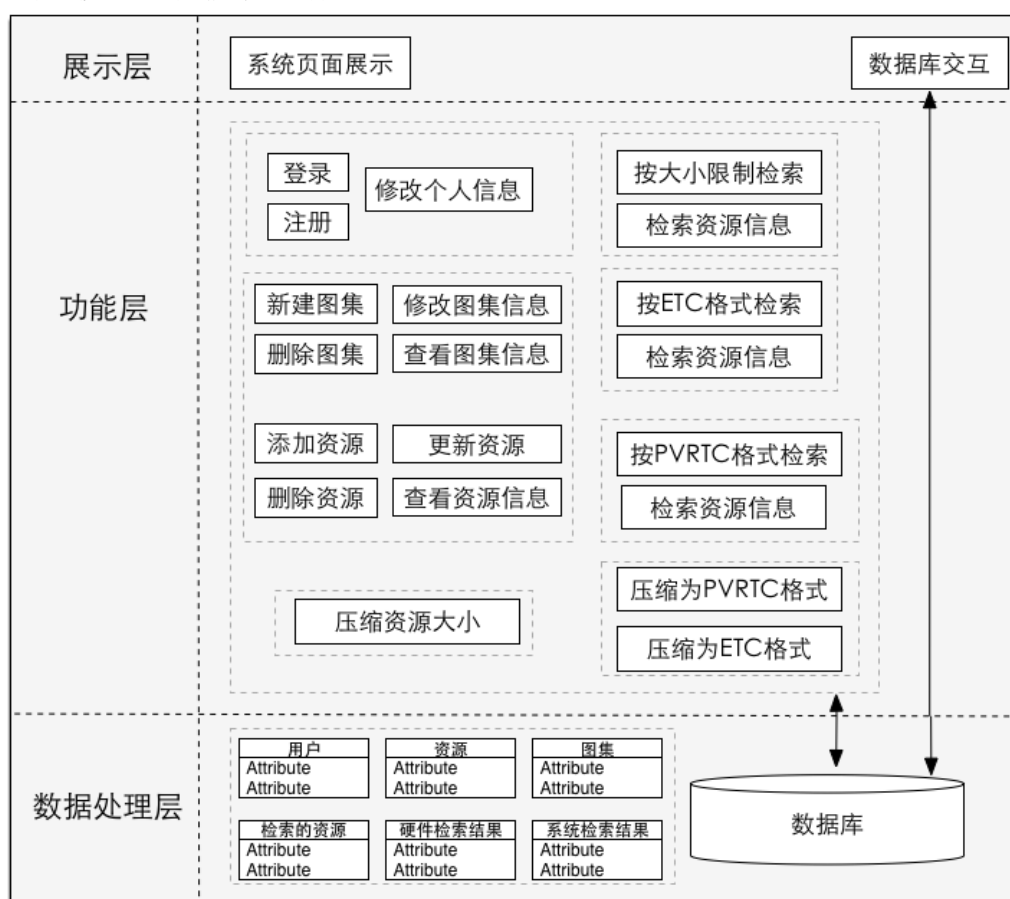


图 4-3 系统架构设计图

Figure 4-3 System architecture design

系统的模型分为本地数据模型和功能模型。本地数据模型是指对给定地址进行检索的美术资源信息，它接收来自控制器的指令，与功能控制器进行交互并或在对象自身上

做出相应变更。功能模型负责程序内部的业务逻辑，例如检索类封装了获取制定路径美术资源信息方法，用于对数据进行收集。

当用户对视图进行操作时，控制器是协调模型与视图层的中间纽带，美术资源管理系统通过控制器类来完成对数据模型的检索，合理的管理模型层的正确执行，也可以正确的把数据呈现到视图层。如图 4-3 所示。

4.3 数据库设计

数据库设计（Database Design）语意为在一个特定的系统环境中，构建最优化的数据库模式，构建数据库及其系统，使之能够对数据进行管理和存储。本节介绍美术资源管理系统的数据使用情况。

4.3.1 数据库概念结构设计

E-R 图也被称为实体-联系图(Entity Relationship Diagram)，该图可以用来表示实体的类型、实体中属性和实体之间联系等。根据系统需求以及功能设计，我们可以对数据库进行系统的设计，首先是针对需求对各相关实体以及实体之间的联系进行系统化的设计，概念模型系统数据库的 E-R 图如图 4-3 所示。

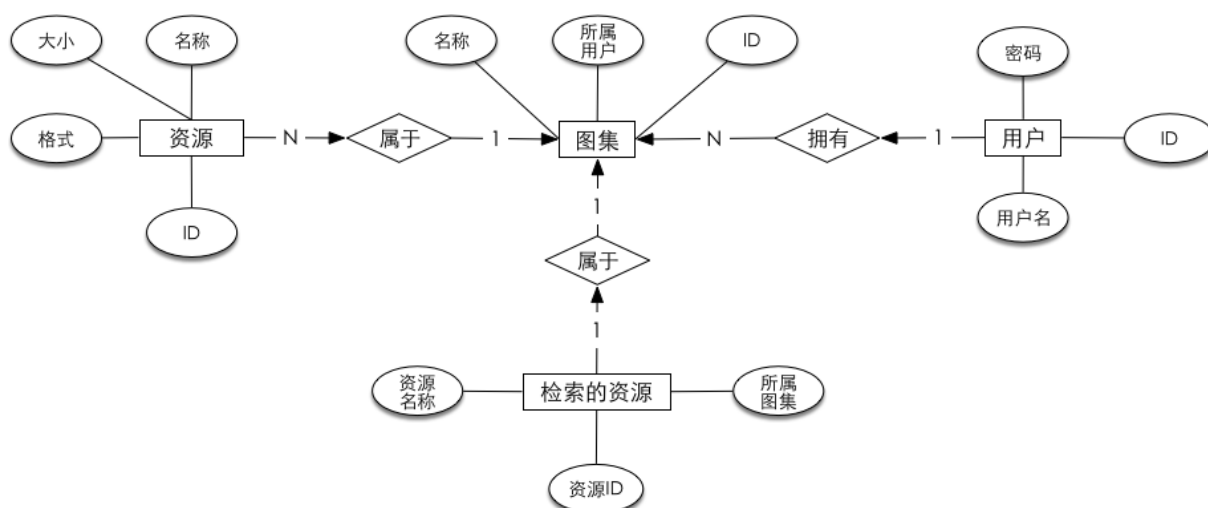


图 4-4 数据库设计 E-R 图

Figure 4-4 E-R diagrams for Database design

4.3.3 数据库逻辑结构设计

把概念结构设计阶段设计出来的 E-R 图转换为与选用的 DBMS 产品所支持的数据模型相符合的逻辑结构就是数据库的逻辑结构设计。美术资源管理系统的逻辑结构设计由六个表构成，包括：用户信息表、资源信息表、图集信息表、待检索资源表、按硬件检索结果表以及按系统检索结果表，美术资源管理系统的所有实体以及是实体间的逻辑关系图如图 4-4 所示：

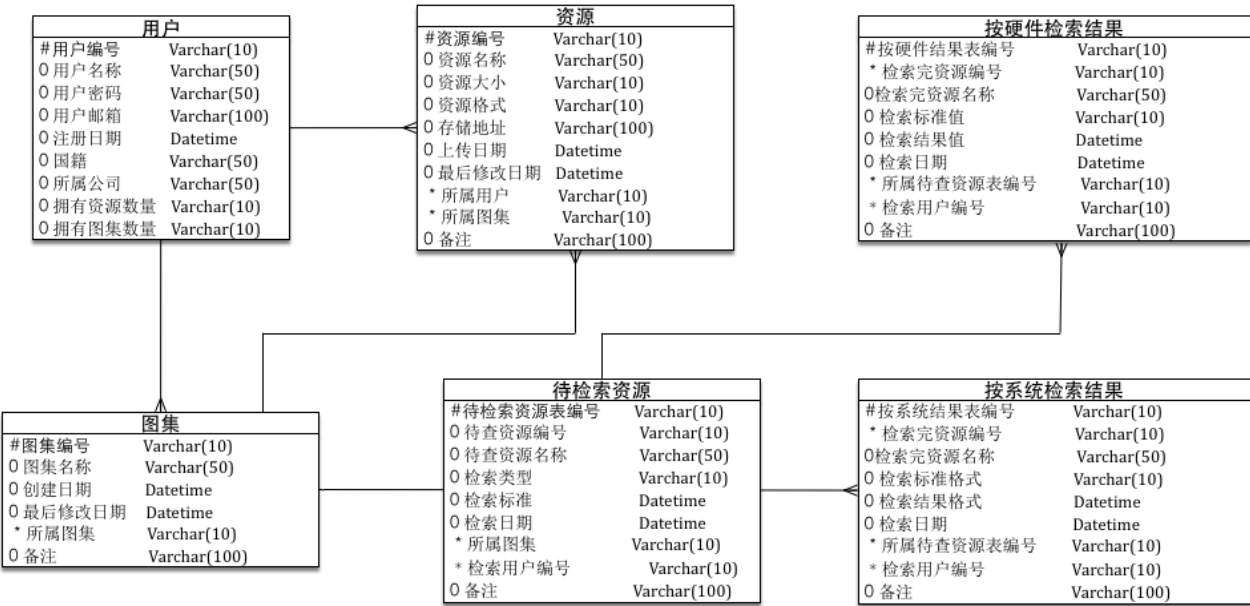


图 4-4 数据库逻辑关系图

Figure 4-4 Database logic relation diagram

4.3.3 数据库物理结构设计

根据分析，美术资源管理系统的数据库中，使用到 6 个实体，在逻辑结构的设计中，我们一共设计了 6 张数据表，数据表一一对应实体和角色，用户信息表、资源信息表、图集信息表、待检索资源信息表、按硬件检索结果表、按系统检索结果表。具体表结构信息如下。

(1) 用户信息表信息包括：用户编号、用户名称、用户密码、注册日期、用户邮箱、国籍、所属公司、拥有资源数量、拥有图集数量。表结构如表 4-1 所示。

表 4-1 用户信息表

Table 4-1 User information table

字段名称	字段代码	数据类型	是否主外键
用户编号	User_ID	Varchar(10)	主键
用户名称	User_name	Varchar(50)	

表 4-1 用户信息表（续）

Table 4-1 User information table (continue)

字段名称	字段代码	数据类型	是否主外键
用户密码	User_password	Varchar(50)	
用户邮箱	Email	Varchar(100)	
注册日期	RegisterDate	Datetime	
国籍	Country	Varchar(50)	
所属公司	Company	Varchar(50)	
拥有资源数量	ResourceNum	Varchar(10)	
拥有图集数量	AtlasNum	Varchar(10)	

（2）资源信息表信息包括：资源编号、资源名称、资源大小、资源格式、存储地址、上传日期、最后修改日期、所属用户、所属图集、备注。表结构如表 4-2 所示。

表 4-2 资源信息表

Table 4-2 Resource information table

字段名称	字段代码	数据类型	是否主外键
资源编号	Resource_ID	Varchar(10)	主键
资源名称	Resource_name	Varchar(50)	
资源大小	Resource_size	Varchar(10)	
资源格式	Resource_format	Varchar(10)	
存储地址	Resource_address	Varchar(100)	
上传日期	Resource_SubDate	Datetime	
最后修改日期	Resource_ModDate	Datetime	
所属用户	Resource_Dep_UID	Varchar(10)	外键
所属图集	Resource_Dep_AID	Varchar(10)	外键
备注	Resource_Memo	Varchar(100)	

（3）图集信息表信息包括：图集编号、图集名称、创建日期、最后修改日期、所属用户、备注。表结构如表 4-3 所示。

表 4-3 图集信息表

Table 4-3 Atlas information table

字段名称	字段代码	数据类型	是否主外键
图集编号	Atlas_ID	Varchar(10)	主键
图集名称	Atlas_name	Varchar(50)	
创建日期	Atlas_CreateDate	Datetime	
最后修改日期	Atlas_ModDate	Datetime	
所属用户	Atlas_Dep_UID	Varchar(10)	外键
备注	Atlas_Memo	Varchar(100)	

（4）待检索资源信息表信息包括：待查资源表编号、待查资源编号、待查资源名称、检索类型、检索标准、检索日期、所属图集、检索用户编号、备注。表结构如表 4-4

所示。

表 4-4 待检索资源信息表

Table 4-4 Retrieved resource information table

字段名称	字段代码	数据类型	是否主外键
待查资源表编号	P_Atlas_ID	Varchar(10)	主键
待查资源编号	P_Resource_ID	Varchar(10)	
待查资源名称	P_Resource_name	Varchar(50)	
检索类型	P_Atlas_type	Varchar(10)	
检索标准	P_Atlas_standard	Varchar(10)	
检索日期	P_Atlas_time	Datetime	
所属图集	P_Atlas_Dep_AID	Varchar(10)	外键
检索用户编号	P_Atlas_Dep_UID	Varchar(10)	外键
备注	P_Atlas_memo	Varchar(100)	

(5) 按硬件检索结果表信息包括：按硬件结果表编号、检索完资源编号、检索完资源名称、检索标准值、检索结果值、检索日期、所属待查资源表编号、检索用户编号、备注。表结构如表 4-5 所示。

表 4-5 按硬件检索结果表

Table 4-5 According to the hardware search results table

字段名称	字段代码	数据类型	是否主外键
按硬件结果表编号	F_HW_ID	Varchar(10)	主键
检索完资源编号	F_HWRes_ID	Varchar(10)	外键
检索完资源名称	F_HWRes_name	Varchar(50)	
检索标准值	F_HW_standard	int(50)	
检索结果值	F_HW_result	int(50)	
检索日期	F_HW_time	Datetime	
所属待查资源表编号	F_HW_Dep_AID	Varchar(10)	外键
检索用户编号	F_HW_Dep_UID	Varchar(10)	外键
备注	F_HW_memo	Varchar(100)	

(6) 按系统检索结果表信息包括：按系统结果表编号、检索完资源编号、检索完资源名称、检索标准格式、检索结果格式、检索日期、所属待查资源表编号、检索用户编号、备注。表结构如表 4-6 所示。

表 4-6 按系统检索结果表

Table 4-6 According to the system search results table

字段名称	字段代码	数据类型	是否主外键
按系统结果表编号	F_system_ID	Varchar(10)	主键
检索完资源编号	F_systemRes_ID	Varchar(10)	外键
检索完资源名称	F_systemRes_name	Varchar(50)	
检索标准格式	F_system_standard	int(50)	
检索结果格式	F_system_result	int(50)	

表 4-6 按系统检索结果表（续）

Table 4-6 According to the system search results table (continue)

字段名称	字段代码	数据类型	是否主外键
检索日期	F_system_time	Datetime	
所属待查资源表编号	F_system_Dep_AID	Varchar(10)	外键
检索用户编号	F_system_Dep_UID	Varchar(10)	外键
备注	F_system_memo	Varchar(100)	

4.4 本章小结

本章内容首先对美术资源管理系统整体做了系统架构设计，接着介绍了系统的功能结构设计，最后介绍了美术资源管理系统的数据库设计，包括数据库概念结构设计、逻辑结构设计。

第五章 游戏美术资源管理系统的详细设计和实现

本章主要从注册登录子系统、管理子系统、按硬件检索子系统和按系统检索子系统四个子系统的功能说明、流程描述和业务逻辑以及类设计几个方面，分别阐述了手机游戏后期美术资源管理系统的详细设计与实现。

5.1 注册登录子系统详细设计

注册登录子系统的详细设计从功能说明、流程描述、类设计以及业务逻辑四个方面进行详细阐述。

5.1.1 功能说明

设计本子系统的目的在于能够对所有的用户进行统一的管理。本子系统可以在新用户注册的时候储存用户的名称、密码、电子邮箱地址以及所在公司等详细信息；用户登录本系统后可以调取储存在服务的服务器信息，本系统能够根据登录用户的信息正确的调取存储在服务器中的用户私有资源；本系统还可以提供用户修改自己的个人信息，新信息将取代旧信息存储在服务器中。

5.1.2 流程描述

运行程序以后，程序将会呈现登录页面，登录页面需要用户输入信息，点击登录按钮后方可进入系统，登录页面还包含注册按钮，点击注册按钮后将会跳转注册页面，用户可以注册成为本系统用户。进入系统后，系统视图被分左右两块，左侧视图中包含一个树形菜单栏，右侧视图被用于内容显示。左侧的树形菜单中有一级菜单库个人中心选项，点击此按钮右侧内容显示界面将会呈现修改个人信息界面。其流程图如图 5-1 所示下：

（1）注册用户

运行美术资源管理系统程序后，用户第一次使用本系统需要注册用户，点击登录注册页面的注册按钮，页面会跳转到注册页面，用户需要输入一些基本信息，包括用户名、密码、邮箱、所在公司等。点击注册按钮，系统会检查用户输入信息的合法性，如果不合法，系统会提示用户输入信息不合法，若信息合法，系统会提示用户注册成功，并跳转到登录页面。

（2）登录系统

运行美术资源管理系统程序后，用户需要输入个人信，包含用户名和密码，信息输

入完成后，用户需要点击登录按钮，程序会将文本框内用户输入的信息与数据库中的用户信息表进行逐项核对，若表中不含有此账户，系统将会提示重新注册，若但密码输入错误，系统会提示用户密码输入错误，若检查无误，系统将会自动跳转到程序主界面。

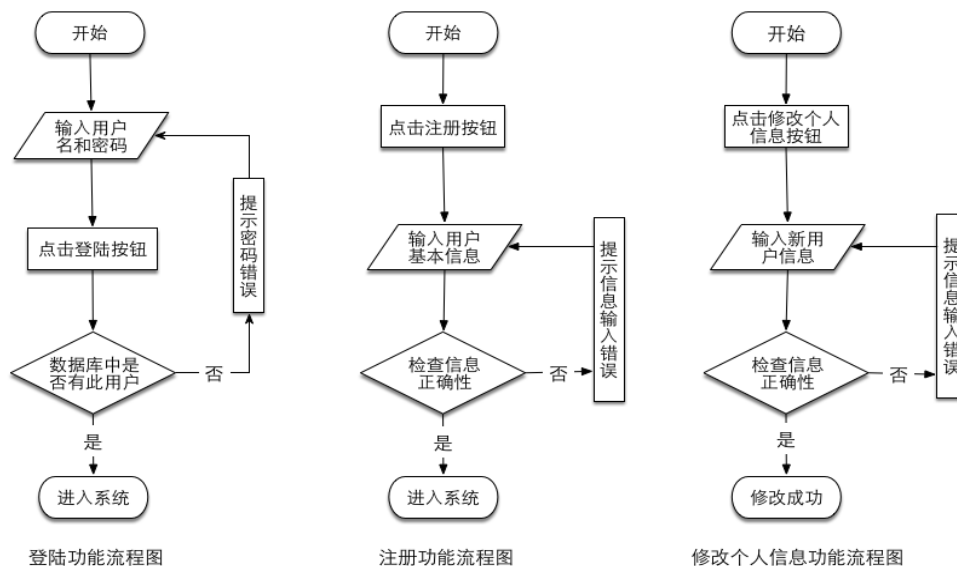


图 5-1 登录注册子系统流程图

Figure 5-1 Login subsystem flow chart

（3）修改个人信息

用户登录到本系统后，可以进行个人信息的修改，在系统主界面的左边栏，用户点击一级菜单修改个人信息后，界面会展开显示二级菜单，用户点击二级菜单中的修改个人信息按钮，系统主界面右侧内容界面将会跳转到修改个人信息界面，用户需要输入新的个人信息，点击修改，系统会检测用户输入信息的合法性，若不合法，系统将会提示用户输入信息不合法，若合法系统将会提示用户修改个人信息成功并跳转到系统主界面。

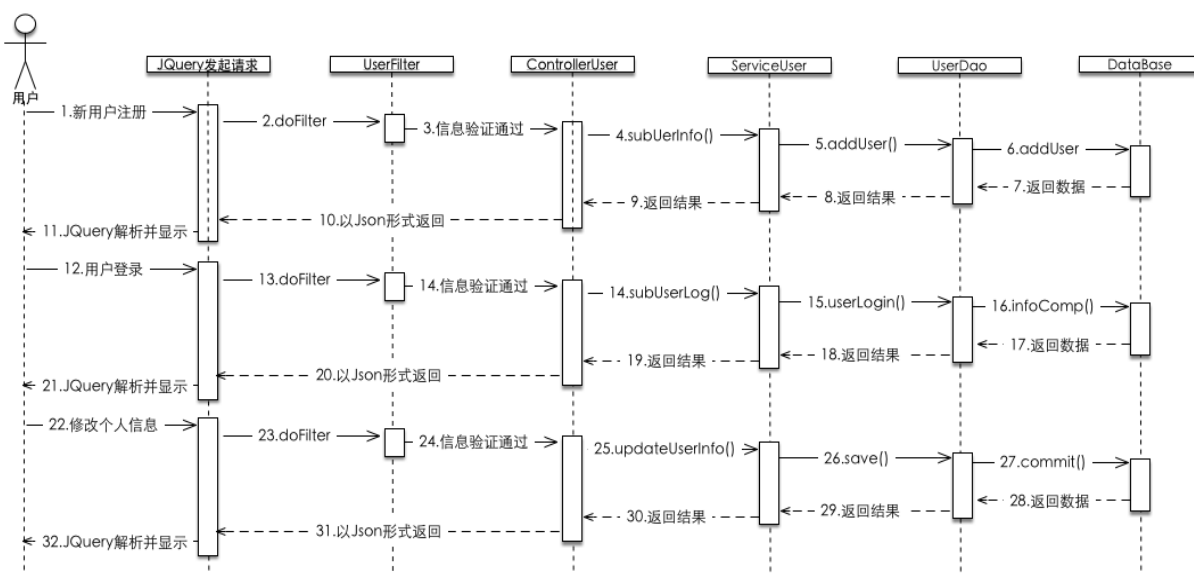


图 5-2 注册登录子系统时序图

Figure 5-2 Registration subsystem timing diagram

5.1.3 业务逻辑和类设计

注册登录子系统涉及系统中用户的登录、注册和信息修改功能。必须注册本系统账户才可以使用本系统，用户登录后可以修改个人信息，注册登录子系统时序图如图 5-2 所示。注册登录子系统使用到的类主要有 ControllerUser.java，ServiceUser.java，User.java，UserDAO.java。类图如图 5-3 所示。

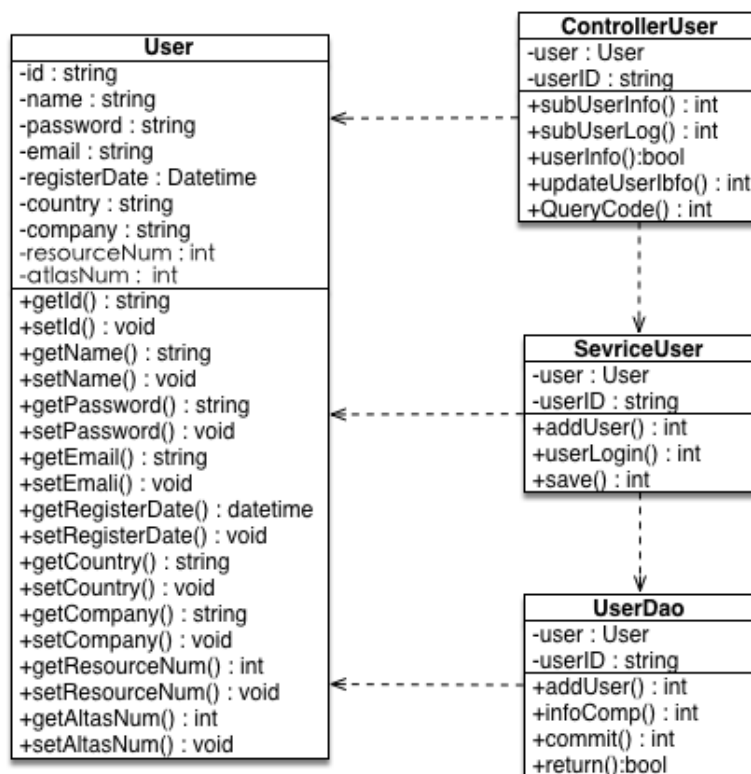


图 5-3 注册登录子系统类设计图

Figure 5-3 Registration subsystem class design

5.2 管理子系统详细设计

管理子系统的详细设计从功能说明、流程描述、类设计以及业务逻辑四个方面进行详细阐述。

5.2.1 功能说明

管理子系统分为两个模块，第一个是资源管理模块，另外一个即是图集管理模块。设计管理子系统的目的在于能够使用户可以对自己的图集和资源进行管理。用户在开始使用美术资源管理系统的时候必须要拥有图集，用户可以新建图集，并对图集执行增删改查功能。再用户拥有一个或者多个图集的时候用户可以把资源存贮在图集之中，并对

其中一个图集里面的资源进行管理，可以对资源进行增删改查功能；两个模块协助运行，以保证本系统的合理性，方便用户使用本系统进行管理。

5.2.2 流程描述

用户登入到系统以后，页面分左右两侧，其中左侧包含一个树形菜单，右侧为内容显示界面。左侧的树形菜单中有一级菜单管理系统，它其中还包括二级菜单图集管理、和资源管理。其流程图如图 5-4 和图 5-5 所示：

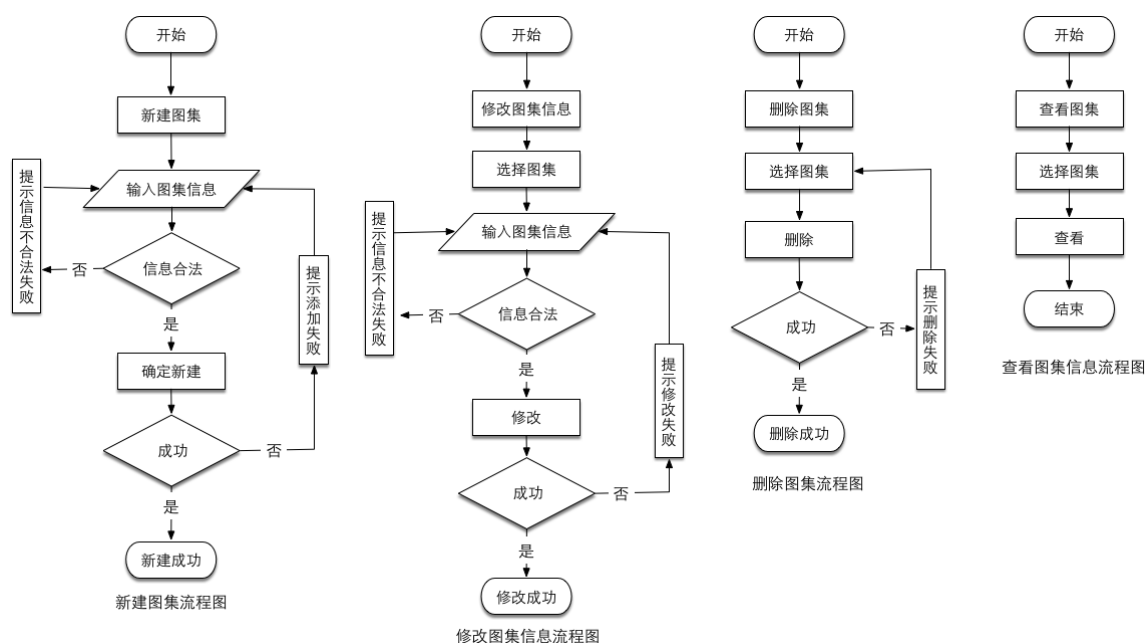


图 5-4 管理子系统图集管理模块流程图

Figure 5-4 Flow chart of the atlas management module of management subsystem

(1) 新建图集

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单管理系统，下设图集管理系统，点击图集管理系统按钮，右侧内容显示界面会跳转到图集管理界面，用户需点击新建图集按钮，系统要求用户输入新建图集所需要的信息，信息输入完成后点击新建图集，系统会检测信息的合法性，若不合法会提示用户信息不合法，并要求用户重新输入信息，若合法，系统将会执行新建图集操作，并将新建图集成功与否反馈给用户。

(2) 修改图集信息

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单管理系统，下设图集管理系统，点击图集管理系统按钮，右侧内容显示界面会跳转到图集管理界面，用户需点击修改图集信息按钮，系统会提示用户选择需要修改的图集，选择图集，

并输入修改图集所需要的信息，信息输入完成后点击修改按钮，系统会检测信息的合法性，若不合法会提示用户信息不合法，并要求用户重新输入信息，若合法，系统将会执行修改图集信息操作，并将修改成功与否反馈给用户。

（3）删除图集

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单管理系统，下设图集管理系统，点击图集管理系统按钮，右侧内容显示界面会跳转到图集管理界面，用户需点击删除图集按钮，系统会提示用户选择图集，然后用户需点击删除按钮，系统将会执行删除图集操作，并将删除成功与否反馈给用户。

（4）查看图集信息

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单管理系统，下设图集管理系统，点击图集管理系统按钮，右侧内容显示界面会跳转到图集管理界面，用户需点击查看图集按钮，系统会提示用户选择图集进行查看，然后用户需点击查看按钮，系统将会将所选图集的相关信息展示给用户。

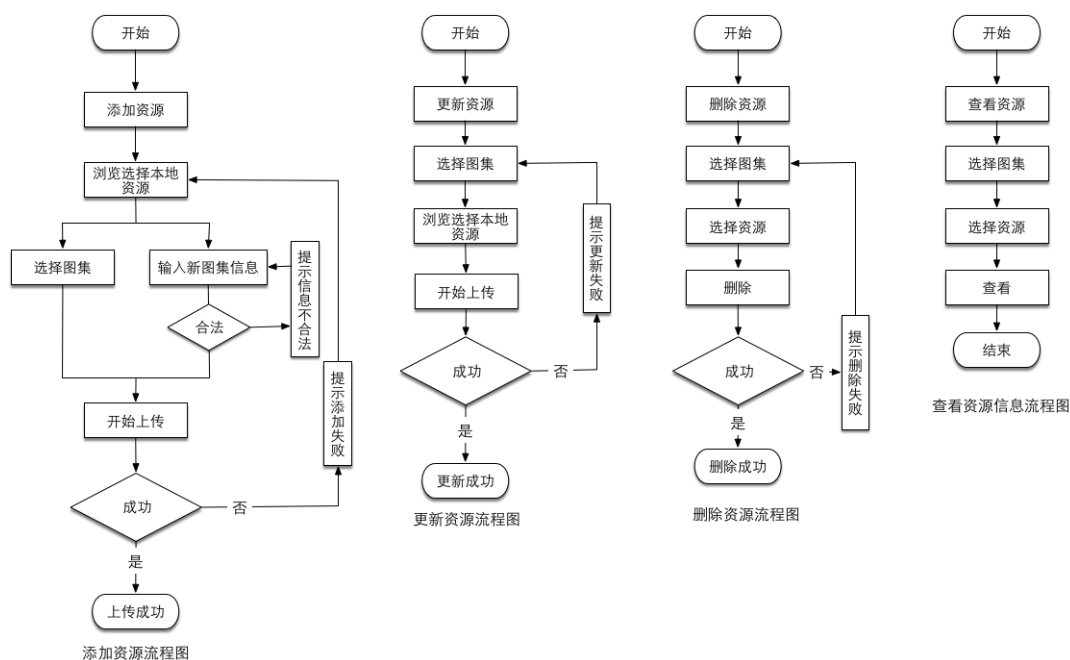


图 5-5 管理子系统资源管理模块流程图

Figure 5-5 Flow chart of the resource management module of management subsystem

（1）添加资源

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单管理系统，下设资源管理系统，点击资源管理系统按钮，右侧内容显示界面会跳转到资源管理界面，用户需点击添加资源按钮，系统将会提示用户进行本地资源浏览并进行选择资源操作，资源上传成功后，用户需要选择将资源上传到哪一个图集，用户可以选择已存在的图集或者新建一个图集。若用户选择新建图集，系统会要求用户输入信息并检测信息

的合法性并反馈。图集选择完毕后，用户点击上传按钮，系统将会把上传成功与否反馈到添加资源界面。

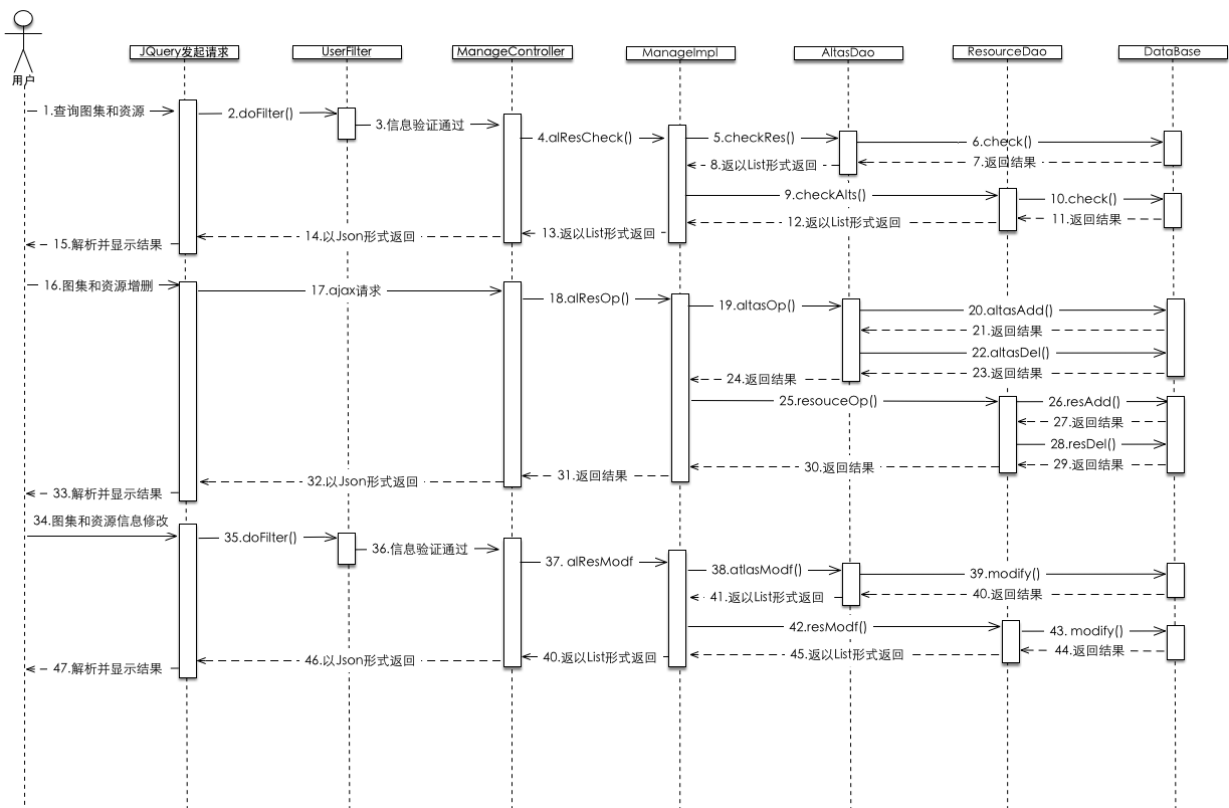


图 5-6 管理子系统时序图

Figure 5-6 Management subsystem timing diagram

(2) 更新资源信息

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单管理系统，下设资源管理系统，点击资源管理系统按钮，右侧内容显示界面会跳转到资源管理界面，用户需点更新资源按钮，用户接下来需要选择图集，浏览并选择本地资源，开始上传操作，系统会反馈更新结果。

(3) 删除资源

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单管理系统，下设资源管理系统，点击资源管理系统按钮，右侧内容显示界面会跳转到资源管理界面，用户需点删除资源按钮，用户需选择图集，再选择需要删除的资源，点击删除按钮，系统会提示用户删除成功与否。

(4) 查看资源信息

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单管理系统，下设资源管理系统，点击资源管理系统按钮，右侧内容显示界面会跳转到资源管理界面，用户需点击查看资源信息按钮，系统会提示用户选择图集，进入到图集中再选择

需查看的资源，然后用户需点击查看按钮，系统将会将所选资源的相关信息展示给用户。

5.2.3 业务逻辑和类设计

管理子系统涉及用户对图集和资源的增删改查操作。用户必须拥有图集才能存放资源进行检索操作，用户可拥有多个图集，图集中可以拥有多个资源。管理子系统时序图如图 5-6 所示。管理子系统使用到的类主要有 Resource.java, ResourceDao.java, Atlas.java, AtlasDao.java, ManageImpl.java 和 ManageController.java，其中 ManageImpl.java 实现了对图集和资源的增删改查功能。类图如图 5-7 所示。

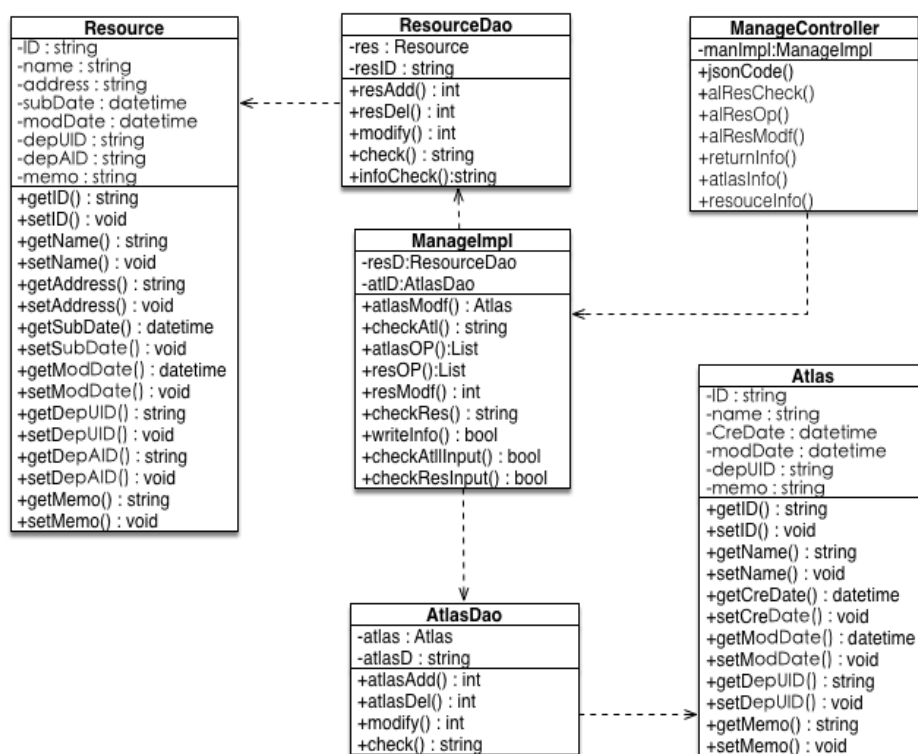


图 5-7 管理子系统类设计图

Figure 5-7 Management subsystem class design

5.3 按硬件检索子系统详细设计

按硬件检索子系统的详细设计从功能说明、流程描述、类设计以及业务逻辑四个方面进行详细阐述。

5.3.1 功能说明

按硬件检索子系统包含两个功能，按大小限制检索，另外一个即是检索资源信息。按硬件检索子系统是美术资源管理系统的核心子系统之一，设计其的目的在于能够使用

户可以根据游戏运行终端的硬件性能限制图片的大小进行资源的检索。用户在开始使用按硬件检索资源子系统的时候必须要拥有图集，图集中存有资源，否则不可以进行检索。用户可以按系统提示选择想要进行检索资源的图集，选中图集后用户需要手动输入检索资源的限制大小，点击检索按钮后，系统将会进行按硬件检索流程，检索后的结果将会呈现给用户，用户还可以选择图集进行资源信息检索，用户需输入资源的检索信息，例如：资源名称、资源大小、上传日期等已知信息，系统检索后将会把符合用户标准的信息以列表的形式将满足检索标准的资源的所有信息呈现出来。

5.3.2 流程描述

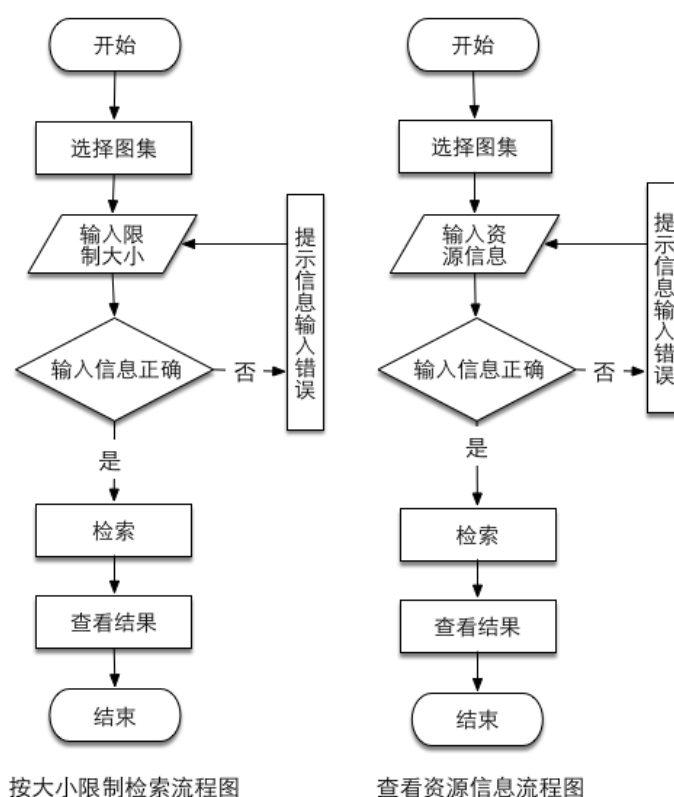


图 5-8 按硬件检索子系统流程图

Figure 5-8 According to the hardware retrieval subsystem flow chart

(1) 按大小限制检索

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单按硬件限制检索，下设按大小限制检索资源，点击按大小限制检索资源按钮，右侧内容显示界面会跳转到检索界面，用户需选择某个图集进行检索，图集选择完成后，用户需要手动输入检索限制，点击检索按钮，系统会审查信息合法与否，若不正确将提示用户输入信息不合法，若信息合法，系统将会执行检索功能，并将信息以列表形式反馈给用户。

(2) 查看信息资源

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单按硬件

限制检索，下设查看资源信息按钮，点击查看资源信息按钮，右侧内容显示界面会跳转到检索界面，用户需选择某个图集进行检索，图集选择完成后，用户需要选择资源名称、上传日期、资源大小，资源格式等信息并手动输入检索资源的信息，点击检索按钮，系统会审查信息合法与否，若不正确将提示用户输入信息不合法，若信息合法，系统将会执行检索功能，并将信息以列表形式反馈给用户。

5.3.3 业务逻辑和类设计

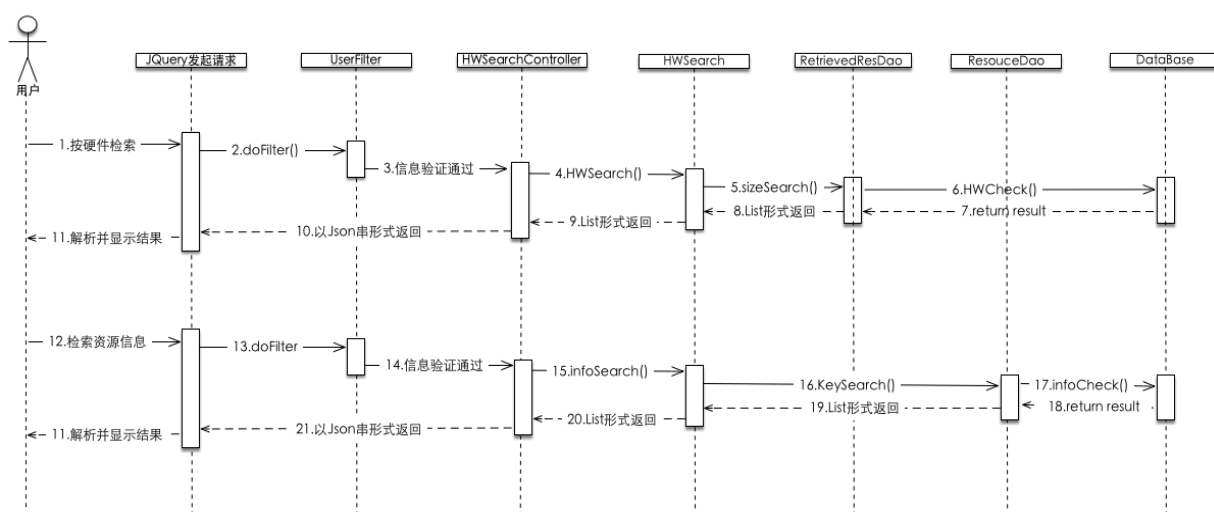


图 5-9 按硬件检索子系统时序图

Figure 5-9 According to the hardware retrieval subsystem-timing diagram

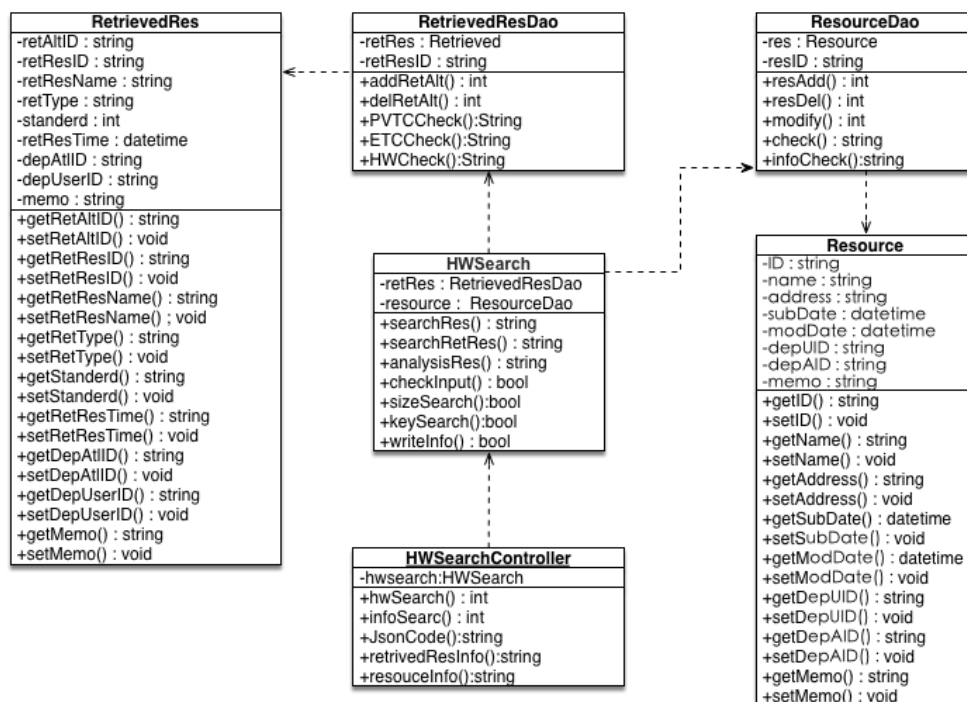


图 5-10 按硬件检索子系统类设计图

Figure 5-10 According to the hardware retrieval subsystem class design

按硬件检索子系统涉及用户对某个图集中的资源进行按大小限制检索和检索资源信息功能。用户必须在拥有图集和资源的前提下才能进行检索，按大小检索需要用户提供检索限制，检索资源信息需用户选择检索的信息种类，并输入信息。按硬件检索子系统时序图如图 5-9 所示。按硬件检索子系统使用到的类主要有 Resource.java, ResourceDao.java, RetrievedRes.java, RetrievedResDao.java 和 HWSearch.java, 其中 HWSearch.java 实现了检索资源、分析资源等算法。类图如图 5-10 所示。

5.4 按系统检索子系统详细设计

按硬件检索子系统的详细设计从功能说明、流程描述、类设计以及业务逻辑四个方面进行详细阐述。

5.4.1 功能说明

按系统检索子系统分为两个模块，按 Android 系统标准检索模块，另外一个即是按 iOS 系统标准检索模块。按系统检索子系统是美术资源管理系统的另一个核心子系统，设计其目的在于能够使用户可以根据游戏运行终端的系统对图片格式设置的标准，进行资源的检索。用户在开始使用按系统检索资源子系统的时候必须要拥有图集，图集中存有资源，否则不可以进行检索。用户可以按系统提示选择想要进行检索资源的图集，选中图集后用户需要选择检索资源的格式（Android 平台为 ETC，iOS 平台为 PVRTC），点击检索按钮后，系统将会进行按系统标准检索流程，检索后的结果将会呈现给用户，用户还可以选择图集进行资源信息检索，用户需输入资源的检索信息，例如：资源名称、资源大小、上传日期等已知信息，系统检索后将会把符合用户标准的信息以列表的形式将满足检索标准的资源的所有信息呈现出来。

5.4.2 流程描述

（1）按 ETC 格式检索

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单按系统要求检索，下设按 Android 系统检索，点击按 Android 系统检索资源按钮，右侧内容显示界面会跳转到检索界面，用户需选择某个图集进行检索，图集选择完成后，用户需要选择按 ETC 格式进行检索，点击检索按钮，系统将信息以列表形式反馈给用户。

（2）按 PVRTC 格式检索

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单按系统要求检索，下设按 iOS 系统检索，点击按 iOS 系统检索资源按钮，右侧内容显示界面会跳转到检索界面，用户需选择某个图集进行检索，图集选择完成后，用户需要选择按

PVRTC 格式进行检索，点击检索按钮，系统将信息以列表形式反馈给用户。

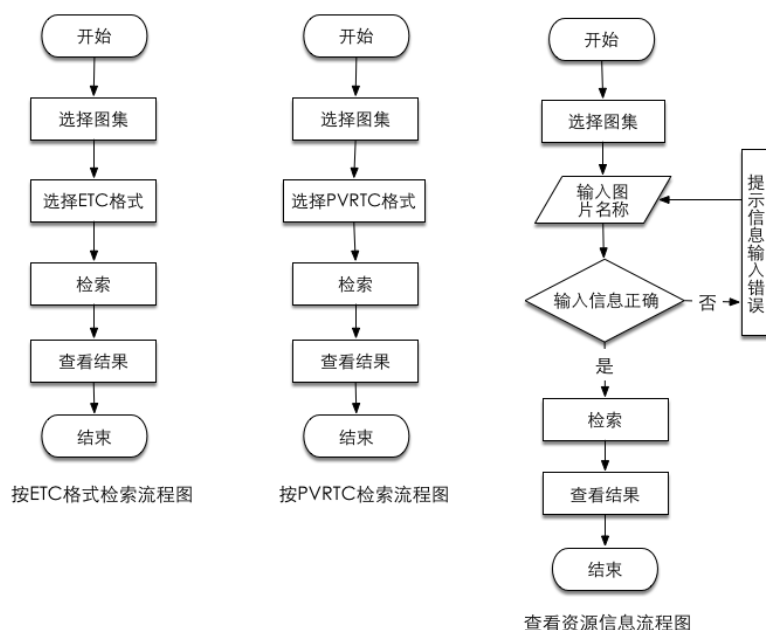


图 5-11 按系统检索子系统流程图

Figure 5-11 According to the system retrieval subsystem flow chart

(3) 查看信息资源

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单按大小限制检索，下设查看资源信息按钮，点击查看资源信息按钮，右侧内容显示界面会跳转到检索界面，用户需选择某个图集进行检索，图集选择完成后，用户需要选择资源名称、上传日期、资源大小，资源格式等信息并手动输入检索资源的信息，点击检索按钮，系统会审查信息合法与否，若不正确将提示用户输入信息不合法，若信息合法，系统将会执行检索功能，并将信息以列表形式反馈给用户。

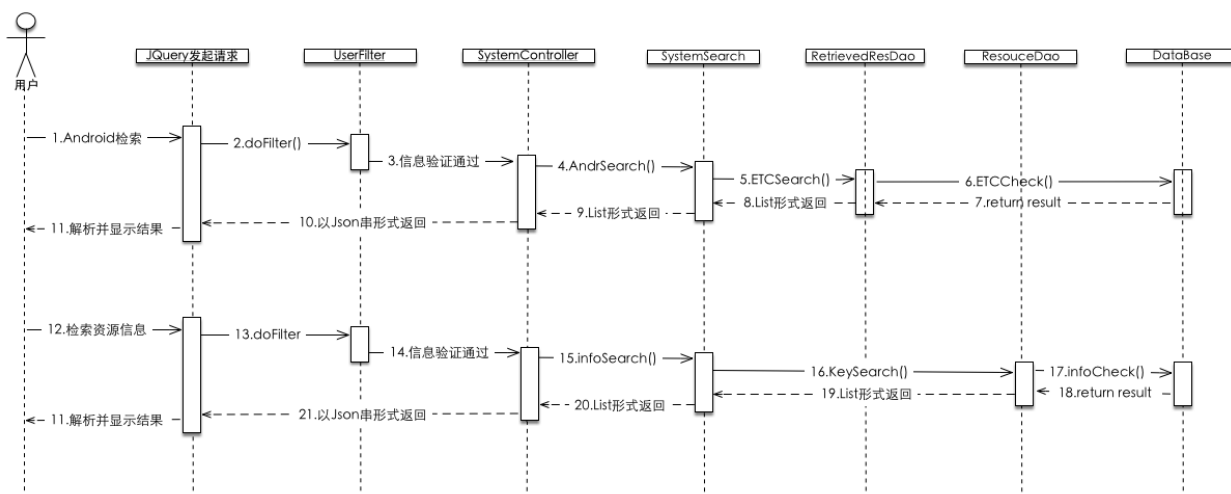


图 5-12 按 Android 系统检索时序图

Figure 5-12 Search by Android system timing diagram

5.4.3 业务逻辑和类设计

按系统检索子系统涉及用户对某个图集集中的资源进行按 iOS 系统限制或按 Android 系统限制检索和检索资源信息功能。用户必须在拥有图集和资源的前提下才能进行检索，按系统检索需要用户需选择按 iOS 系统或者 Android 系统检索。检索资源信息需用户选择检索的信息种类，并输入信息。如图 5-12 和图 5-13 所示。

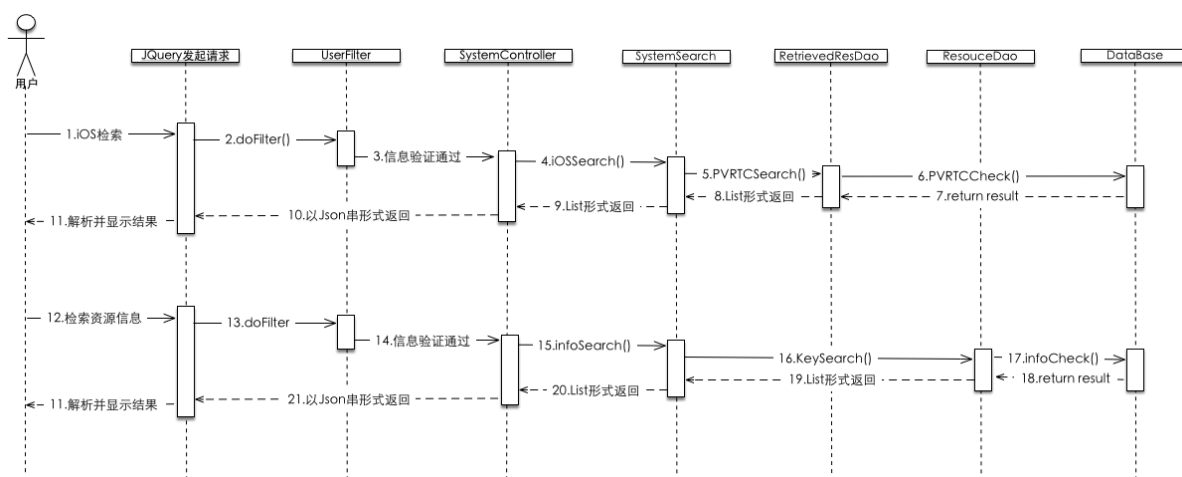


图 5-13 按 iOS 系统检索时序图

Figure 5-13 Search by iOS system timing diagram

按系统检索子系统使用到的类主要有 Resource.java, ResourceDao.java, RetrievedRes.java, RetrievedResDao.java 和 SystemSearch.java, 其中 SystemSearch.java 实现了检索资源、分析资源等算法。类图如图 5-14 所示。

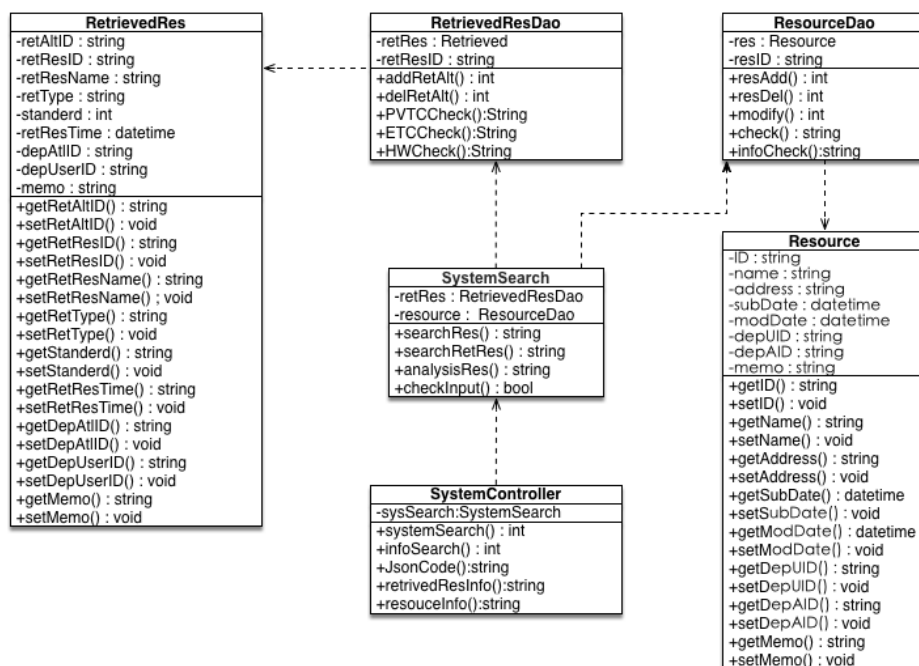


图 5-14 按系统检索子系统类设计图

Figure 5-14 According to the system retrieval subsystem Class design

5.5 压缩资源子系统详细设计

压缩资源子系统的详细设计从功能说明、流程描述、类设计以及业务逻辑四个方面进行详细阐述。

5.5.1 功能说明

压缩资源子系统分为两个模块，压缩资源大小模块，另外一个即是压缩资源格式模块。按压缩资源子系统是美术资源管理系统的另一个核心子系统，设计其目的在于能够使用户可以根据按硬件检索和按系统检索完成后的结果，对资源进行修改。用户开始使用压缩资源子系统的时候必须要拥有图集，图集中存有资源，否则不可以进行压缩资源。用户可以按系统提示选择想要进行压缩资源的图集，选中图集后用户需要选择图集中需要压缩的资源，在资源选择完成后，用户可以选择压缩目标格式（或者输入压缩目标大小），信息确认和系统压缩后将会把压缩资源是否成功的信息反馈给用户。

5.5.2 流程描述

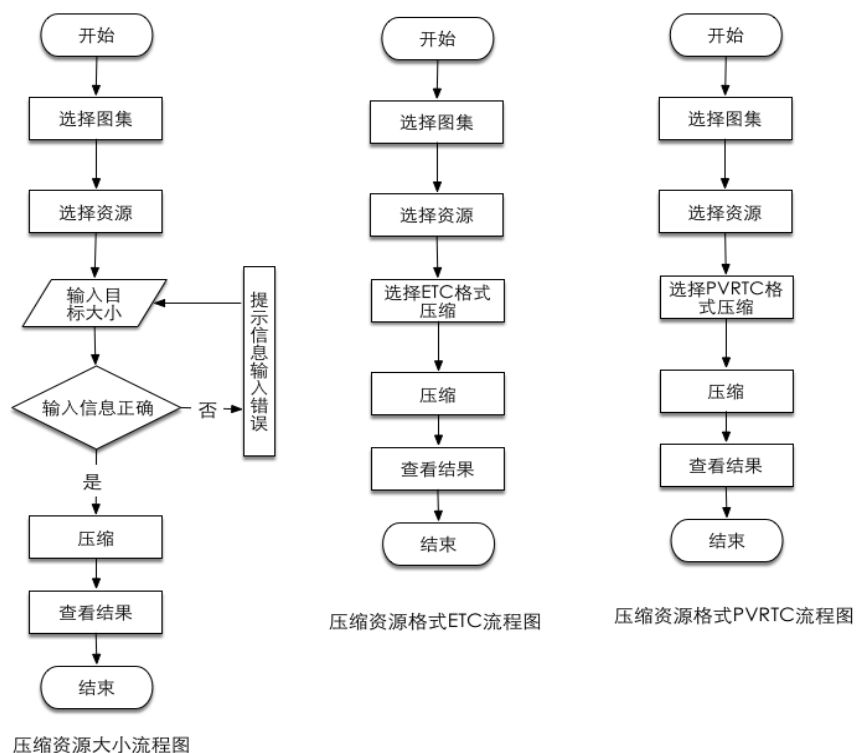


图 5-15 压缩资源子系统类设计图

Figure 5-15 Compressed resource subsystem class design

（1）压缩资源大小

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单压缩资源按钮，下设压缩资源大小按钮，点击压缩资源大小按钮，右侧内容显示界面会跳转到对资源大小压缩的界面，用户需选择某个图集，然后选择图集中的资源，资源选择完成后，用户需要输入目标压缩大小，点击压缩按钮，系统将把压缩成功与否反馈给用户。

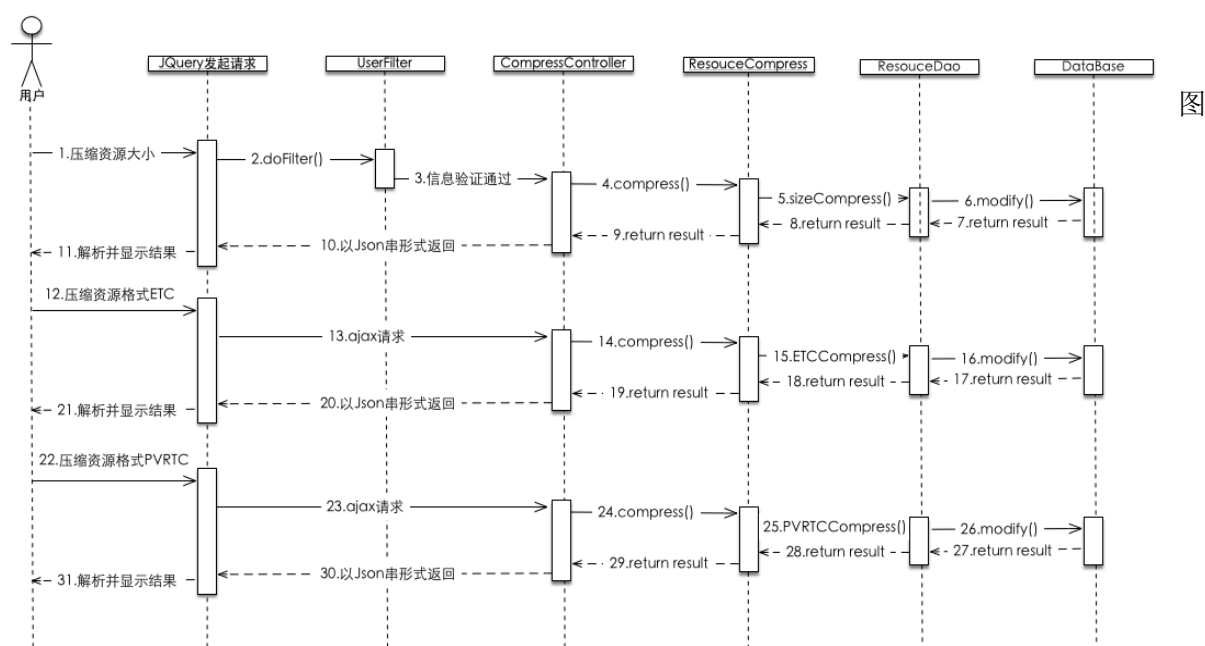
（2）压缩资源格式为 PVRTC

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单压缩资源按钮，下设压缩资源格式按钮，点击压缩资源格式按钮，右侧内容显示界面会跳转到对资源格式压缩的界面，用户需选择某个图集，然后选择图集中的资源，资源选择完成后，用户需要选择压缩目标格式为 PVRTC，点击压缩按钮，系统将把压缩成功与否反馈给用户。

（3）压缩资源格式为 ETC

用户进入系统后，在系统主界面的左侧设计了菜单栏，菜单栏包含一级菜单压缩资源按钮，下设压缩资源格式按钮，点击压缩资源格式按钮，右侧内容显示界面会跳转到对资源格式压缩的界面，用户需选择某个图集，然后选择图集中的资源，资源选择完成后，用户需要选择压缩目标格式为 ETC，点击压缩按钮，系统将把压缩成功与否反馈给用户。

5.5.3 业务逻辑和类设计



5-16 压缩资源子系统时序图

Figure 5-16 compressed resource subsystem sequence diagram

压缩资源子系统涉及用户对某个图集中的资源进行压缩资源大小操作、压缩资源为 ETC 格式操作、压缩资源格式为 PVRTC 格式操作。用户必须在拥有图集和资源的前提下才能进行压缩，压缩资源需要用户需选择压缩资源大小或者压缩资源格式两个模块。压缩资源大小用户需要输入压缩目标格式大小。压缩资源格式用户需要选择具体目标格式为 ETC 格式或者 PVRTC 格式。如图 5-16 所示。

按系统检索子系统使用到的类主要有 Resource.java，ResourceDao.java，ResourceCompress.java 和 CompressController.java 等 java 类，其中 ResourceCompress.java 实现了资源 ETC 格式压缩、分析资源、资源 PVRTC 格式压缩，资源大小压缩，资源文件读取，资源文件头类型读取等算法。类图如图 5-17 所示。

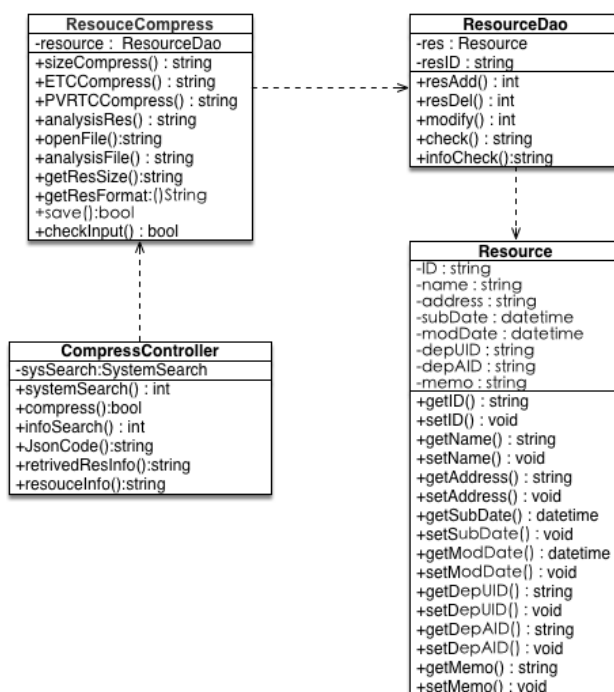


图 5-17 压缩资源子系统类设计图

Figure 5-17 compressed resource subsystem class design

5.6 本章小结

本章从注册登录子系统、管理子系统、按硬件检索子系统以及按系统检索子系统四个方面重点介绍了手机游戏后期优化美术资源管理系统的详细设计。在每个功能模块下，主要使用 java 类、类主要方法函数及系统界面设计描述了各个模块的详细设计。

第六章 游戏美术资源管理系统的测试

本章介绍了手机游戏后期优化美术资源管理系统的测试情况，为了保证本系统不仅能够在用户的操作下正确运行，还要保证系统功能可以达到预期目标，所以本章将从系统功能性测试与系统性能测试这两个方面对手机游戏后期美术资源管理系统进行综合全面的衡量，并且对本系统的当前运行情况做出了陈述并对其存在的缺陷提出了改善性的建议。

6.1 系统功能测试

本小节列举了三个测试用例，针对第三章系统需求分析所提需求，进行测试用例的设计，每个测试用例分别包括执行步骤、预期结果、实际结果、测试结论等内容，并根据三个测试结果进行了详细的相关说明，三个测试用例如表 6-1、表 6-2 和表 6-3 所示。

表 6-1 按 ETC 格式检索测试用例表
Table 6-1 Search by ETC format use case table

测试类型	测试情况
测试用例编号	24571
测试优先级	高
测试用例标题	检验按 ETC 格式检索资源功能
测试类型	功能性测试
用例设计者	XXX
设计日期	2016 年 3 月 11 日
前置条件	用户拥有至少一个图集，图集中至少拥有一个资源，系统界面处于按 Android 系统检索资源页面，且页面内容能够正确显示
测试方法	选择任意图集，选择 ETC 格式进行对这个图集的检索，查看检测结果与实际情况是否一致，信息是否正确。
执行步骤	点击“按系统标准检索”按钮，继续点击“Android 系统检索标准”按钮，在右侧内容显示页面中选择需要检索的图集，选择按 ETC 格式进行检索，等待系统检索完成并反馈结果，对比系统检索反馈结果与实际情况是否一致,信息显示是否正确。
预期结果	按 ETC 格式进行检索所得到的资源情况与用户所进行检索的图集资源情况完全一致，信息显示完全正确。
实际结果	实际检测结果与预期结果完全一致。

表 6-1 按 ETC 格式检索测试用例表（续）

Table 6-1 Search by ETC format use case table (continue)

测试类型	测试情况
结论	应用本系统按 ETC 格式检索图片功能检索资源成功，所得数据与实际数据完全一致，本系统可以给用户提供正确的按 ETC 格式进行美术资源检索的功能。
对应开发人员	XXX
测试日期	2016 年 3 月 11 日

表 6-1 是对手机游戏后期优化美术资源管理系统的核心功能进行测试，当用户拥有至少一个图集，图集中至少拥有一个美术资源的时候，用户可以对某个图集资源进行按系统标准进行检索中的 Android 系统检索中的按 ETC 格式进行检索。由检测结果观察到，在对某个图集进行按 ETC 格式进行检索后，系统所反馈给用的信息与图集中实际拥有的 ETC 格式资源完全一致，并且反馈的资源信息完全正确，因此该功能可以满足用户的需求。

表 6-2 压缩资源为 ETC 格式测试用例表

Table 6-2 Compress resource to ETC format test case table

测试类型	测试情况
测试用例编号	24614
测试优先级	高
测试用例标题	检验压缩为 ETC 格式功能
测试类型	功能性测试
用例设计者	XXX
设计日期	2016 年 3 月 11 日
前置条件	用户拥有至少一个图集，图集中至少拥有一个资源，系统界面处于压缩为 ETC 格式页面，且页面内容能够正确显示
测试方法	选择任意图集，在图集中选择任意资源，选择目标格式为 ETC，然后对这个资源进行压缩，查看压缩完成后与实际情况是否一致，信息是否正确。
执行步骤	点击“压缩资源按钮”按钮，再次点击“压缩为 ETC 格式”按钮，在右侧内容显示页面中选择图集，在图集中选择资源，然后选择压缩格式为 ETC，点击“压缩”按钮，等待系统压缩完成并反馈结果，对比系统压缩反馈结果与实际情况是否一致，信息显示是否正确。

表 6-2 按大小现在检索测试用例表（续）
Table 6-2 Search by size use case table (continue)

测试类型	测试情况
预期结果	压缩资源为 ETC 格式操作后的资源情况与用户所进行压缩的图集集中的资源情况完全一致，信息显示完全正确。
实际结果	实际压缩结果与预期结果完全一致。
结论	应用本系统压缩某一资源成为 ETC 格式操作成功，所得资源格式数据与实际数据完全一致，本系统可以给用户提供正确的压缩美术资源格式为 ETC 的功能。
对应开发人员	XXX
测试日期	2016 年 3 月 11 日

表 6-2 是对手机游戏后期优化美术资源管理系统的核心功能进行测试，当用户拥有至少一个图集，图集中至少拥有一个美术资源的时候，用户可以对某个图集集中的资源进行压缩资源中压缩资源格式为 ETC 操作。由检测结果观察到，在对某个图集集中的资源进行压缩后，系统压缩资源所得到的最终格式信息与数据库中资源的信息完全一致，并且反馈信息完全正确，因此该功能可以满足用户的需求。

表 6-3 修改图集信息测试用例表
Table 6-3 Modify atlas information use case table

测试类型	测试情况
测试用例编号	24670
测试优先级	高
测试用例标题	检验修改图集信息功能
测试类型	功能性测试
用例设计者	XXX
设计日期	2016 年 3 月 11 日
前置条件	用户拥有至少一个图集，系统界面处于图集管理页面，且页面内容能够正确显示
测试方法	选择任意图集，对本图集的信息进行修改，再次登录系统并查看图集信息，观测图集信息是否被正确修改。
执行步骤	点击“管理”按钮，然后点击“图集管理”按钮，最后点击“修改图集信息”按钮，在右侧内容显示页面中选择需要修改的图集，点击“修改”按钮，在修改图集信息页面的文本框内输入新的图集信息，点击“保存修改”按钮，再次的登录本系统并查看被修改图集的信息，观测信息是否是最新的。

表 6-3 修改图集信息测试用例表（续）

Table 6-3 Modify atlas information use case table (continue)

测试类型	测试情况
预期结果	图集信息被正确修改。
实际结果	实际修改结果与预期结果完全一致。
结论	应用本系统的修改图集信息功能对图集信息进行修改成功，修改后的数据可以正确显示，本系统可以给用户提供正确的修改图集信息功能。
对应开发人员	XXX
测试日期	2016 年 3 月 11 日

表 6-3 是对手机游戏后期优化美术资源管理系统的基本功能进行测试，当用户拥有至少一个图集的时候，用户可以对某个图集进行信息修改的操作，由检测结果观察到，在对某个图集修改后，再次登录本系统查看的图集信息已变成最新的图集信息，并且信息完全正确，因此该功能可以满足用户的需求。

6.2 系统性能测试

系统性能测试可以验证手机游戏后期美术资源管理系统的可靠性、易用性、易于维护性等特点，以下将简要对系统界面测试和系统性能测试两方面进行简要的测试分析。

（1）界面测试

使用性能测试：其判断标准就是用户看上去舒服、用起来方便。由于本系统采用了页面框架等结构，左边均为菜单部分，右边均为对应的内容操作页面，因此显得整体结构清晰，使用效果简洁方便。

跳转测试：链接测试用于检查系统的每个界面的可访问性。测试的内容基本可以分为以下三个方面：首先是检查所有界面按钮跳转是否按照所设计的正常运行，跳转到了指定的界面；第二是检查界面中的所有按钮跳转所指向界面是否是存在的；其三是检查整个美术资源管理系统中是否有孤立界面，也就是所有的按钮跳转都无法到达的界面。

数据输入测试：对系统中所有的界面中存在数据输入的地方，都需要进行合法与非法数据的输入测试和数据的临界值测试，以检查美术资源管理系统的数据处理能力。

数据处理和显示测试：主要验证数据处理的准确性，即为检查本管理系统数据的存取是否正确。

针对以上几个测试原则，本人对手机游戏后期美术资源管理系统进行了测试，测试结果表明，所有界面均简单直观；所有按钮均可以正确跳转；数据验证均符合设计，临界值符合要求；数据存取正确，显示无误。

（3）性能测试

性能测试主要是通过对本系统的服务器施加一定的访问上的压力，来查看其对不同网络环境的适应性能如何，从而依此测试结果来提出改进意见。本管理系统是用于中小型企业公司内部网络环境下，因此只在实验室内局域网的条件下进行了测试，测试结果良好，可以获得验证，本管理系统具有良好的性能、易用等特点。

6.3 本章小结

本章对系统的测试情况进行了详细的描述，从系统功能测试和系统性能测试两方面对系统进行测试并做了简要分析，最终通过对手机游戏后期美术资源管理系统的测试表明，本系统设计与建设不仅达到了预期的目标，并且取得了良好的效果

第七章 总结与展望

本章主要对论文研究的手机游戏后期优化美术资源管理系统的工作进行相关总结与展望。通过本章，对论文中所提出的系统功能模块的划分、开发流程以及设计、测试过程所取得的成果和不足进行总结与展望。

7.1 工作总结

针对手机游戏开发行业的特点，在手游开发后期优化过程中使用计算机管理系统进行科学性的管理，可以大大提升管理效率与管理水平。基于此点，本文展开了针对手游后期优化资源管理的研究，分析了手游后期优化美术资源管理系统的主要功能需求、数据库设计和系统架构设计等问题。具体工作主要包括如下几个方面：

- （1）介绍了手机游戏开发行业的特点，从而分析了当前手游后期优化资源管理的具体需求，重点讨论了美术资源管理的系统目标，以及管理系统应采取的体系结构。
- （2）从理论上阐述了 unity3D 引擎的项目优化具体内容和方法，美术资源优化的压缩方式比较和美术资源优化压缩目标格式的选择。
- （3）根据手机游戏后期优化美术资源管理系统的特点，给出了手机游戏后期优化美术资源管理系统的功能性分析和系统的非功能性分析。
- （4）对手机游戏后期优化美术资源管理系统的设计进行了介绍，重点描述了本系统的系统架构设计、功能结构设计、数据库设计以及详细设计。

7.2 工作展望

尽管我们在前期做了大量的有关美术资源管理系统的调研工作并对系统做了详细的需求分析分析。但是由于系统建设仓促，算法未经雕琢，我们还是发现了一些不足：系统界面有待优化、系统功能有待丰富、系统测试仓促等等。总的来说，我们将把本系统的未来工作放在以下几个方面：

- （1）组织测试人员进行全面的系统功能性测试。
- （2）对于测试人员根据测试结果提出的改进建议，详细记录在案。
- （3）针对系统改进意见，继续修改完善美术资源管理系统。
- （4）调查使用过本系统的用户，针对其提出的针对功能方面的修改意见进行统计整理。

- （5）针对统计结果进行功能性的需求分析，继续完善美术资源管理系统的功能。
- （6）探索相关系统界面设计的优秀方案，修改人机交互的方式并优化系统界面。
- （7）做好系统数据的备份工作以及服务器的防范灾害工作。

参考文献

- [1] 童娜《浅析游戏行业的变化趋势》，《大众文艺：学术版》，2014年3期, P257—P258
- [2] 马继华《3G时代的手机游戏产业》，2010年1期 P110—P113
- [3] 戈晶晶《“后短信时代”SP的出路》，2004年19期 P18—P22
- [4] 冯迪. 基于J2ME平台的手机游戏设计与实现[M]. 电子科技大学出版社, 2007
- [5] 刘晓龙《浅谈手机游戏的开发与改进》2011年8期 P250—P250
- [6] 《unity4.X从入门到精通》，中国铁道出版社，2013
- [7] 《Unity3D 游戏开发》，北京：人民邮电出版社，2012。
- [8] 张楠楠《浅谈 Unity3D 虚拟现实系统在房地产动画中的应用前景》，2015年6期
- [9] 李俊雅陈永翔《虚拟仿真居室游览系统的设计与实现》，2010年15期，P4103—P4104
- [10] 黄裕兴《基于 Unity3D 引擎的 Android 手机益智类游戏开发》2014年1期 P138—P138
- [11] 3D Computer Graphics: A Mathematical Introduction with OpenGL, by Samuel R. Buss, June 2003 (sample code, etc.)
- [12] Essential Mathematics for Games and Interactive Applications, by Jim Van Verth and Lars Bishop, Sept. 2003

致 谢

首先要感谢我的毕业设计指导老师 XXX 老师。缺乏经验的我在毕业设计论文撰写的过程中遇过困难、走过弯路，感谢 XXX 一次又一次不厌其烦地悉心指导，从论文开题知道、总体结构、论文内容、到论文格式，每一处细微的地方都仔细提出了修改建。老师的指导过程不仅是对我在论文撰写上的帮助，更是对我在以后生活、学习上一笔宝贵的财富。

同时感谢培育我四年的软件学院以及各位老师，老师们的谆谆教导使我从对软件工程一无所知到今天可以为软件行业贡献力量，感谢学院和老师们在成长过程中铺砌了道路。

也要感谢单位研发部的领导和同事，是他们带我 from 学校走向社会，走出了自己在软件行业的第一步。没有他们的帮助我不可能完成自己的任务，不可能如此迅速的成长。

最后要感谢我的同学们，他们不仅在毕业设计过程中给予了我很多帮助，更是在我四年大学生活中陪我走过了每一步。

感谢所有支持、关心和帮助过我的人。

附 录

附录 A 文献翻译

原文:

Patent Name: METHOD AND SYSTEM FOR LABELING A DOCUMENT FOR STORAGE, MANIPULATION, AND RETRIEVAL

Patent Number: 5,680,223

Date of Patent: *Oct. 21, 1997

Inventors: Martin F. N. Cooper, Fremont; Walter A. L. Johnson, Santa Clara; Dick Wah Lo, Milpitas; Z. Erol Smith, III, Palo Alto, all of Calif.

Filed: May 15, 1995 Related U.S. Application Data

References Cited: U.S. PATENT DOCUMENTS

4,587,633 5/1986 Wang et al. 364/900
 4,833,720 5/1989 Garcia-Serra . 382/11
 4,893,333 1/1990 Baran et al. . 379/100
 4,972,496 11/1990 Sklarew 382/187
 5,060,980 10/1991 Johnson et al. 283/70
 5,099,340 3/1992 Kamada et al. . 358/403
 5,126,858 6/1992 Kurogane et al. . . 358/450
 5,172,245 12/1992 Kita et al. 358/403
 5,231,510 7/1993 Worthington .. 358/400
 5,251,265 10/1993 Dtlhle et al. . 382/119
 5,267,047 11/1993 Argenta et al. 358/400
 5,444,840 8/1995 Froessl 395/145
 5,448,375 9/1995 Cooper et al. . . 358/403
 5,465,167 11/1995 Cooper et al. 358/468

FOREIGN PATENT DOCUMENTS:

@83267 9/1988 European Pat. 01f. .

ABSTRACT

Storage, manipulation, and retrieval of files, for example data representations of scanned documents, is facilitated by establishing a relationship between an arbitrary, image domain file label and a computer recognizable text domain file name for the file. Selection of the arbitrary, image domain file label is interpreted as a selection of the related file. The arbitrary, image domain file name is assigned by way of a paper form or the like, and may be assigned at the time of document storage. The arbitrary, image domain file label facilitates the meaningful naming of a file for storage when a keyboard or other typical

text entry apparatus is unavailable, such as when inputting a document for storage by way of a facsimile machine. Character recognition is not performed on the arbitrary, image domain file label, so the burden on the processing resource is minimized, while errors from inaccuracy are eliminated. Selection of a file for processing by way of its arbitrary, image domain, le label may be by appropriate indication on either a screen display or a printed form.

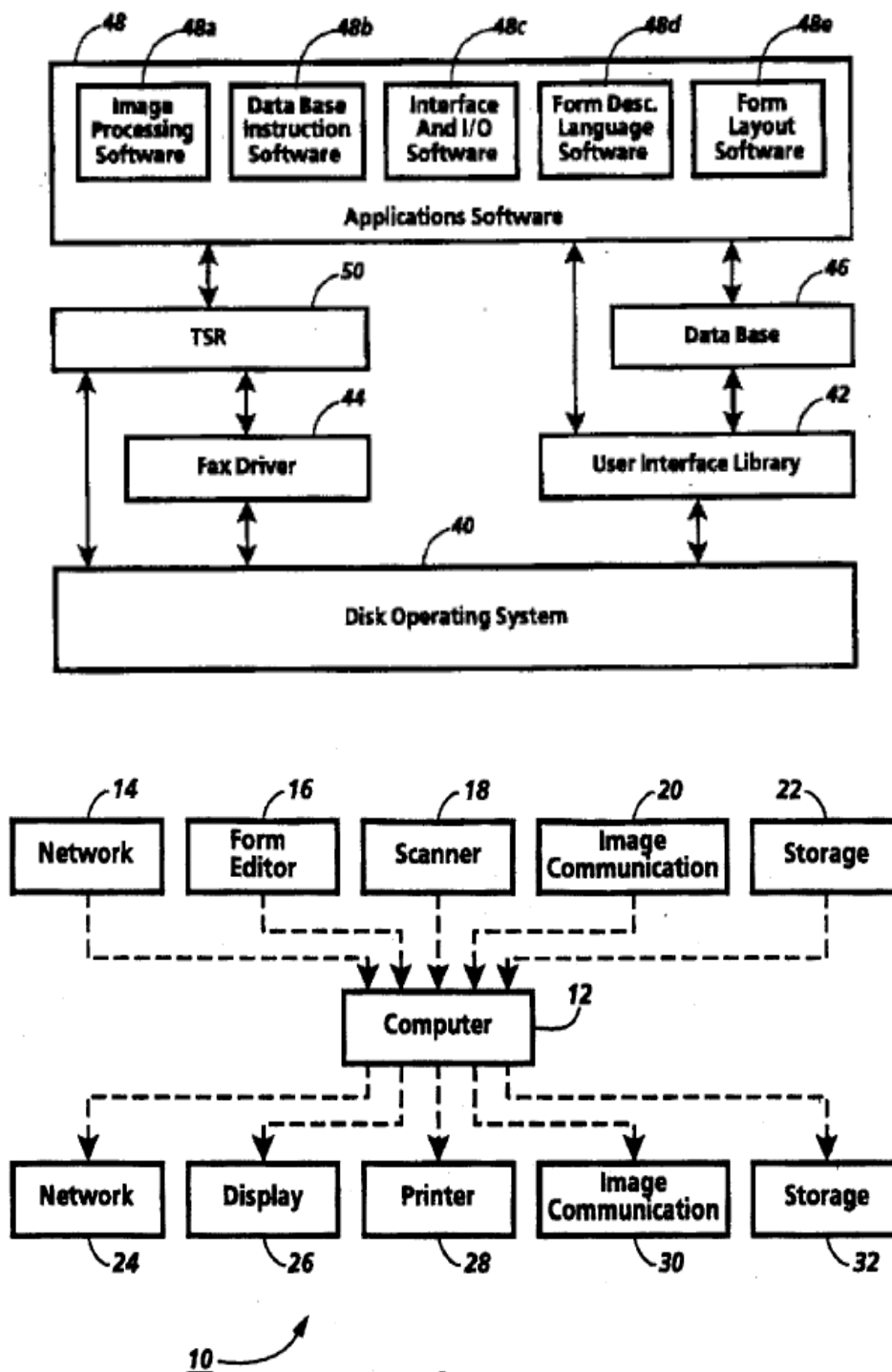
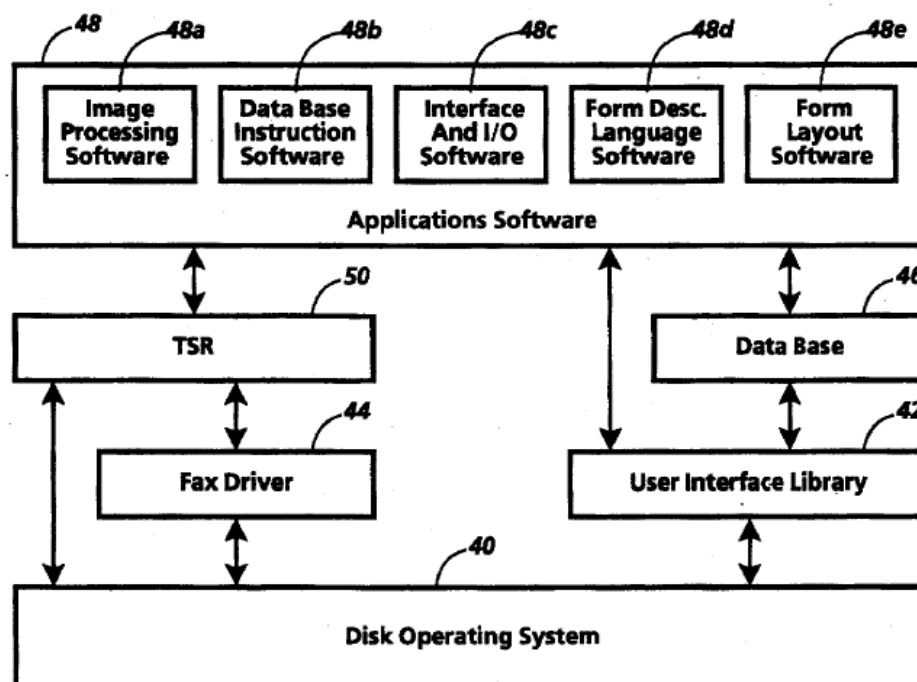


Fig. 1

**Fig. 2**

The present invention relates generally to document processing methods and systems, and more specifically to a method and system for labeling a document with an arbitrary, image domain document label for document storage, manipulation, and retrieval. Scanning documents for processing on a digital computer, such as a personal computer ("p.c."), a workstation, or other digital data processing resource is now routine. Furthermore, remote document storage, manipulation, and retrieval is becoming more commonplace today given the improving interfaces between computers and telecommunication devices such as fax machines. For example, a user can now "fax" a document to his computer for the purposes of storing the document on the computer, redistributing the document via the computer, etc. What ties these two different document processes together is that they both involve apparatus peripheral to the data processing resource. The present invention is concerned with facilitating the use of such peripheral apparatus, specifically the naming and referring to files stored on the data processing resource. For purposes of the present discussion, the digital data processing resource such as the p.c., workstation, and the like will be referred to herein as a computer. Document as used herein shall be understood to mean a carrier, such as paper, for carrying markings, as well as the markings, if any, applied to the carrier. A file as used herein shall be understood to mean a collection of data, for example that representing a scanned image of a document, stored or accessible to a computer. The term electronic representation of data will be used herein, although the representation of the data (i.e., data

representation) may be electronic, magnetic, optical, or other appropriate representation. Furthermore, the data may be in analog or digital format. Finally, document storage, manipulation, and retrieval will be understood to represent all actions that a user may perform on a document and its electronic representation, including those requiring communication between a peripheral apparatus and the computer. For example, this includes document scanning and transmission to the computer from a “remote” scanner, retrieving a file from the computer, transferring a document from one computer to another computer, etc. These definitions will simplify the explanation herein of the background and details of the present invention, although it will be understood that their use should not be interpreted as limiting the spirit and scope of the present invention.

Fundamentally, in order to perform any task on a document requiring communication between a peripheral apparatus and the computer, the document must be represented by data, i.e., an electronic representation of the document must be generated. Typically, the generation of an electronic representation of a document will be performed by a document scanner, which generates a description of the on/off state of the picture elements (“pixels”) comprising the image, and packages the representation as a file. The form of the electronic representation may, for example, be a bitmap of the document or a coded collection of data representing the document. Once an electronic representation of the document (hereafter referred to as an “electronic document”) is generated, there must be a way of uniquely identifying it. This requirement is most commonly handled by the disk operating system resident on the computer. For convenience, virtually every disk operating system permits, and in fact requires either the user or the computer to assign a file name to the file containing the electronic document for subsequent identification of the file. According to known document storage, manipulation and retrieval systems, the user selected file name must be in a format which is recognizable by the computer, for example encoded text such as EBCDIC or ASCII which may be entered from a keyboard. Electronic documents transmitted to a computer for storage and/or processing from a peripheral device are typically named at the time of transmission to or receipt by the computer in association with the task of document storage. For example, a user may enter via a keyboard attached to the sending or receiving device an encoded text name for the electronic document. Alternatively, the sending or receiving device may automatically assign an encoded text name to the electronic document according to a preestablished rule for name assignment. Typically, the task of document storage involves establishing a destination for the file in a memory media, such as a physical location on a magnetic disk, in RAM, etc., and a system identification (“systemID”) of that destination. As part of the storage process, the disk operating system establishes and maintains a correspondence between the assigned file name and the system ID. The file name, when assigned by the user, is often

a mnemonic device or other label allowing a user to identify from the file name the general or specific contents of the file. When the file name is assigned by the system, it is most often a generic name such as, for example, the user's name,

the name of the device from which the file was transmitted, the date and time of creation of the file, etc. Thus, a user is typically more likely to be able to identify the contents of a file when the user assigns the file name than when it is assigned by the system. There are known systems that permit document retrieval using peripheral apparatus, such as a fax machine. One disclosure of such a system is U.S. Pat. No. 4,893,333. According to this reference, a prestored document is identified for retrieval by way of indicia imparted on the form, for example, so-called bar codes, fill-in check boxes or fill-in fields. The idea of identifying a form absent such indicia by use of appropriate image processing software is also disclosed therein. Furthermore, performing certain operations (store, retrieve, forward, etc.) on documents by way of a peripheral device, is provided when the document is capable of being identified by way of dual-tone DTMF telephone signals, as disclosed for example, in U.S. Pat. No. 4,918,722, or in the User Handbook, Version 3.01, for the Xerox® Fax Master 21™ software product.

One problem continually encountered in the art is that not all peripheral devices are accompanied by a keyboard allowing the user to enter an appropriate file name, for example for assigning a file name for file storage, accessing prestored files, etc. A typical stand-alone scanner comprises optical imaging components, software for processing images, and possibly paper document handling mechanisms. Typical facsimile devices include the above as well as a numerical keypad, but rarely include all of the keys of a full alpha numeric keyboard. In general, present peripheral apparatus limit the ability of the user to assign a meaningful file name to files and access previously stored files. Furthermore, when identifying pre-stored and pre-named files by way of filling in check boxes or fill-in fields, at least one check box or fill-in field must be appropriately marked for each character in the file name. This leads to time consuming and error prone document identification. For example, if check boxes are employed to identify a file, a great many such check boxes must be provided to allow identification of alphanumeric file names. If fill-in fields are employed, the processing apparatus which identifies the document must ultimately perform character recognition on the indications in the fill-in fields. Finally, virtually every system for establishing file names requires not only that the file name be in a format which is recognizable by the computer, but that the character set used in the file name be the native character set of the computer. For example, it is generally not possible to assign a file name to a file using a foreign language character set or graphics unless the processing apparatus is capable of recognizing the character set or graphics. This precludes such operations as assigning a file a file name with Kanji characters when the computer

is capable of recognizing only the Latin characters set.

SUMVIARY OF THE INVENTION

The present invention overcomes the problem of the limits imposed on a user in entering a file name by providing a method and system for assigning a meaningful user selected file label to files which uses existing peripheral devices. Specifically, for a file having an assigned file name, a relationship is established between an image domain file label and the file name assigned by the computer, so that the label may be employed to assist the user in identifying the file. The present invention builds on the methods and systems of the prior art by providing a relationship between the assigned file name and the image domain file label for a file. This allows establishing a meaningful file label for a file which can stand in the place of the less meaningful assigned file name. Furthermore, by establishing this relationship, a user may more easily and directly identify a desired file in a system lacking a text entry device than heretofore provided by the prior art. One aspect of the present invention involves storing a file on or by way of a computer. According to this aspect, the file is initially a document consisting of a can 'ier means such as paper, plastic, etc., having markings such as printing or writing thereon. A special cover form is employed which includes a region in which the user imparts an image domain label (for example a handwritten name or illustration) for the file. The document, prefaced by the cover form, is scanned by a scanning means whose output is an electronic data file containing the image of the form and the document. This data file is transmitted to a computer, where it is assigned a file name and stored as a file either in the computer's memory or in a memory media associated with the computer. Associated with the transmission of the data file to the computer will be an instruction to the computer to store the file (the instruction being read from the form or other input device). The computer establishes a location in which to store the file and creates a file name for the file. The computer maintains the association between the location of the file and the file name according to methods well known in the art. Next, the computer distinguishes the data representing the form and the data representing the document, locates representation of the image domain label imparted on the form, and establishes a relationship by way of database entries between the data representing the image domain label and the data representing the document. When the computer is called on to access the document, it displays or prints the image domain label in such a manner that selection of the image domain label is interpreted by the computer to mean selection of the document. Another aspect of the present invention involves accessing for sending, retiieving, deleting, etc., a previously stored electronic document having related to it an image domain file label. According to this aspect, a user would request a listing of the labels of an appropriate set of files which are stored on or accessible to the computer. In response to the request for the listing, the computer generates a display of the image domain in file label, if any, and possibly other

indications, for each file. The display may be an image formed on a computer display, a printed paper document, etc. From this display, the user selects the item(s) of interest by selecting the image domain file label, for example by highlighting the file label on the computer display or imparting a check mark in a check box field on a paper or other printed document of the display. Based on the preestablished relationship between the image domain file label and the file name, the computer is able to interpret the user's selection as a selection of the associated file. Closely related to the above is the aspect that an image domain label may be assigned to a file and used to identify that file without resort to character recognition software such as optical character recognition ("OCR"). That is, there is no requirement to convert the image domain file label into a machine recognizable format. This reduces the demands on the processing resources of the computer, increases the speed at which the computer can process instructions involving the image domain label, allows use of characters other than those supported by the character set of the computer (e.g., Kanji characters used on a standard DOS machine), allows use of non-textual labels (such as figures or relevant non-textual marks), and allows the user to select the image domain label without requiring the user to duplicate that label.

Yet another aspect of the present invention is that the file to which the image domain label is assigned need not be an electronic document. For example, the file may be data representing one or more instructions, or a program of instructions, which the computer will follow to accomplish specific tasks. That is, the underlying subject matter having the associated image domain label may be a computer program which may be referred to, loaded, and/or run in or by the computer by referring to the image domain label. Alternatively, the file to which the image domain label is related may be one or more of many other types of files, such as binary files in formats utilized by other data processing resources. In fact, the term "file" is used herein in its broadest sense to refer to the element to which the image domain label is assigned, and shall be understood to mean any data item or portion of a data item which is appropriate for assignment of an image domain label. Further aspects of the present invention and the manner in which it addresses the above problems, as well as others, will become more readily apparent from the following detailed description when taken in conjunction with the accompanying drawings, detailed description, and claims.

译文：

专利名：为了实现文档存储、操作与检索的方法与系统

美国专利号：5,680,223

授权日期：1997年10月21日

发明人：Martin F. N. Cooper, Fremont; Walter

A. L. Johnson, Santa Clara;

Dick WahLo, Milpitas;

Z. Erol Smith, III, Palo Alto, all of Calif.

协作单位：

参考相关专利：美国专利

4,587,633 5/1986 Wang et al. 364/900

4,833,720 5/1989 Garcia-Serra . 382/11

4,893,333 1/1990 Baran et al. . 379/100

4,972,496 11/1990 Sklarew 382/187

5,060,980 10/1991 Johnson et al. 283/70

5,099,340 3/1992 Kamada et al. . 358/403

5,126,858 6/1992 Kurogane et al. . . 358/450

5,172,245 12/1992 Kita et al. 358/403

5,231,510 7/1993 Worthington .. . 358/400

5,251,265 10/1993 Dtlhle et al. . 382/119

5,267,047 11/1993 Argenta et al. 358/400

5,444,840 8/1995 Froessl 395/145

5,448,375 9/1995 Cooper et al. . . 358/403

5,465,167 11/1995 Cooper et al. 358/468

国外专利

@83267 9/1988 European Pat. 01f. .

摘要

对扫描文件的存储、操作、以及检索是通过在计算机中建立图像域的文件标签与相关的计算机可识别的字符域的文件名相关联的方式实现的。选取任意的图像域标签都会被翻译到相应的对字符域标签的选取。任意的图像域标签能辅助系统为文件制定可以更有意义的文件名，即便是外设无法输入很多的特定字符。由于引入了，字符识别也不是必需的步骤了。因而对计算资源的占用将被最小化，同时准确性更高。通过图像域标签来选取文档是一个更加方便而快捷的方式。

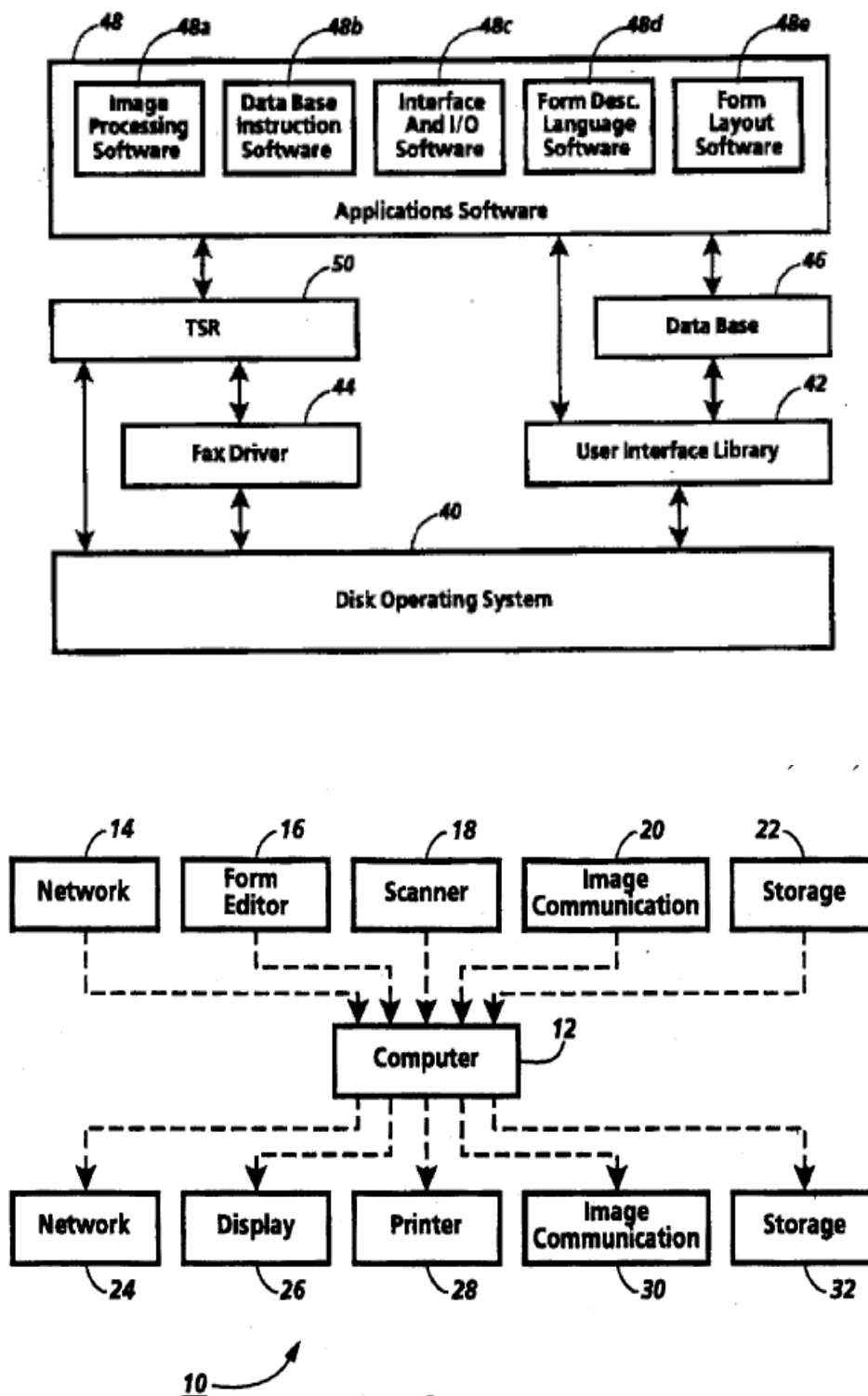


Fig. 1

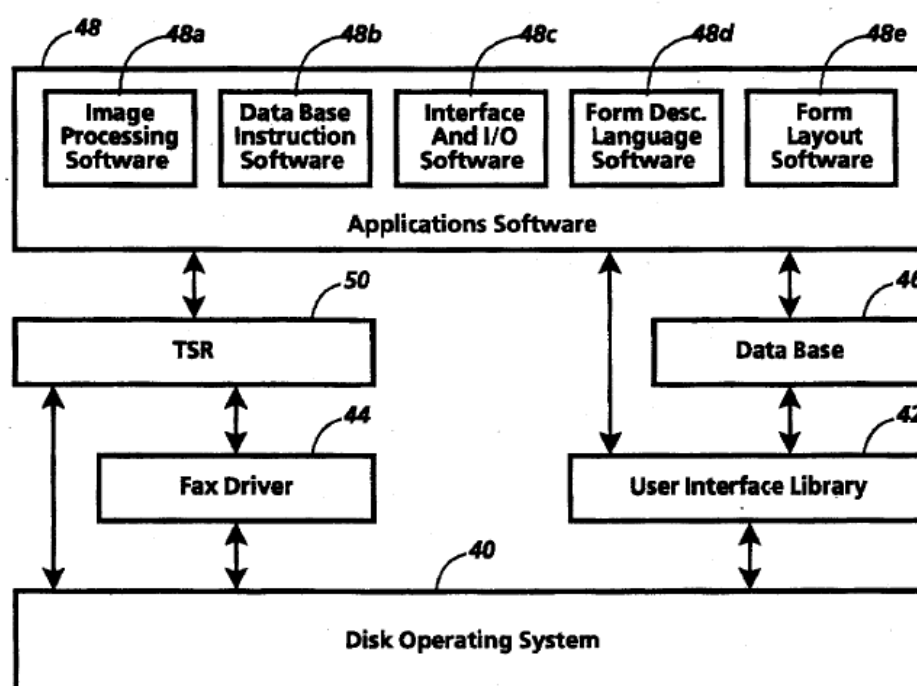


Fig. 2

为了存储、处理、检索而标记文档的方法与系统发明的背景：此发明经常和文件处理的方法与系统联系起来，更具体来说，是和对有任意图像域的图像文件进行标识，进而实现文件存储、处理与检索的方法与系统。

在像个人电脑、工作站或者其他的数字信号处理资源的数字计算机上面为了处理文件而扫描是很常规的。更进一步说，远程的文件存储，处理和检索如今正变得越来越普遍。比如用户可以用将文件传真到他的计算机上，可以通过他的计算机发布它的文件。将这两个过程联系起来的是数字处理资源之外的外围设备。此发明考虑促进这些外围设备的功用。

本段讨论的是像个人电脑、工作站和其他可以被人们称为计算机的各种计算资源。文档，可以被人们认为是一种容器，就像可以承载各种标记的纸张。文件被认为是一种数据集。“数据的电子呈现”这个词意味着我们的数据将会以电子或者磁抑或是光学或者其他相近的形式呈现。而且，数据应该以模拟或者数字形式展示。最后，用户对文件的存储，处理与检索可以看作是用户对文件的所有操作行为。以上这些顶一会贱货我们后面对发明的解释。

实现对文档的各种作业操作需要外围设备与电脑的通信。文档必须以计算机上的电子数据的形式呈现。一般的数字文档的生成一般来说要是有一个扫描仪完成的。扫描仪通过开和关的状态记录图像的每一个像素，并以像素来组成图片并将其打包成文件的形式。电子文档的形式可能是一个文档的比特图或者数据编码。

文档一夕生成就必然会有一个独特的方式来标识它。这个工程施工过计算机上的硬盘操作系统。为了方便，硬盘操作系统会允许用户和计算机自己给文件命名。就目前已知的文件存储、操作和检索系统来说，用户指定的文件名必须是像 ASCII 那样可以被计算机识别的字符。从外设转送到计算机等待存储与处理作业的电子文档通常是在被传输或者被接收的时刻被命名的。一般的文档存储的过程包含了一个为文件在存储媒介中

建立一个目的地的过程。作为存储工作的一部分,硬盘操作系统会建立与维持文件名和系统 ID 的映射。

用户指定的文件名是一个助记方式,或者可以帮助用户确定文件的内容或者相关信息。目前有已知的系统可以允许通过传真机来实现文件检索。有专利披露这种系统。一个预存储的文件为检索而被标识这一过程将会通过对文件形式给予标记的方式来进行。更进一步说,有一种系统可以通过外设进行某种特定的文件操作,这种系统会采取双音多轨电话的形式来实现这一功能。此系统被 Xerox 公司的相关专利披露。

一个常见的问题是外设不一定经常会有一个可以输入字符的外设。有些设备不具有可以输出全套的字符的键盘。一般来说,目前的外设限制了用户给文件定一个有意义名字和检索到任意一个位置文件的能力。当通过搜索框来标志一个未存储或者未命名的时,文件名中的所有的字符必须被妥善的标记出来。这将导致一个耗时而且容易出错的过程。最终,实际上没有一个系统给文件名命名的过程中不仅需要文件名是可以被计算机识别的格式,而且所需的字符也是计算机原生的字符系统。

发明概述:

本发明克服了用户在利用方法和系统的过程中外围设备对用户键入有意义的文件名的种种限制。特别的,对于一个有用户指定文件名的文件,用户指定的用户名将会与计算机制定的用户名相互匹配。

本发明的另外一大特色在于提出了一种将制定的文件名与图像文件的文件域名相关联的有联系的命名方式。这样,文件可以避免使用一个无意义的文件名。与此同时,用户也可以更容易地通过文件名来了解文件信息,更容易地发现和定位到我们需要的文件。

本发明的一个方面涉及了通过或者在电脑上面存储文件。开始文件相当于一个像纸张与塑料一样的写满了符号与标记的载体。一个带有文件域标识的特殊封面会出现在文件中。以这一封面为序言的文档会被扫描,这意味着文档将会以数字形式呈现给用户。文档将会被传送到计算机上,而计算机正是给文档命名和存储的位置。与文件传输到计算机相关的,一个指令也会被送到计算机处。之后计算机给文件安排了一个存储文件的物理空间并且保存好了指定的文件名,并将文件在存储空间的位置与文件名关联。接下来,计算机将识别文档的文件数据格式标识和文档内容,定位图像域标签,并且在数据库中建立关联文件数据域域信息与文件数据的数据条目。

另一个与本发明相关的方面涉及为了发送、检索和删除而读取文件,那些之前存储好的、与特定图像域标签相关的文档。根据这一特性,用户将会请求一系列的存储于计算机的特定文件集合的标签。计算机为了响应用户的请求,将会在展示出文件标签的图像域,以及其他的可能需要的文件描述。展示结果的方式可能是在屏幕上以图像方式展示出来,也可能会以打印的纸质文档形式表示出来。接下来用户就可以在展示的列表中通过选取文档图像域的方式选出自己需要的文档。根据之前就存储好的文件关联,计算机可以很容易得来将用户的选择转化为相应的文件项。

与之前所说的内容紧密相关的,还有这样一个方面。那就是系统使用了与文档相关联的图像域文件标签。这样就可以不依赖光学字符识别软件。这样就不需要将图像域文件标签转化为机器可以识别的形式。这将大大减少对计算机资源的需求,提高了响应速度。更进一步来将,在这个发明采用的技术这一层次上,有图像域标签标记的文档甚至可以不是电子数据。进一步来讲,当文件细节被包含进入标签之后,文档标签将更容易被大家所理解。