

专 业 学 位 硕 士 学 位 论 文

基于 J2ME 的手机 3D 游戏引擎的研发

**Research and Development Of 3D Game Engine For Mobile
Based on J2ME**

作 者 姓 名: 赵 晖

学 科、专 业 : 软件工程

学 号 : 40317431

指 导 教 师: 李明楚

完 成 日 期: 2005-10-20

大连理工大学

Dalian University of Technology

摘 要

随着通信和计算机技术的日益进步,游戏不再仅仅可以在台式计算机上玩。手机硬件技术的进步,使手机也逐渐成为一个好的游戏平台。手机虽然在计算能力、存储空间、视觉效果、声音效果上都不如台式计算机,但是,手机游戏面对的是庞大的手机使用人群从而拥有美好的未来。当人们在等待公车、购物排队以及一些比较短的间歇时间内,拿出手机享受一下手机游戏带来的快乐,这是现代通信技术与计算机技术给予人们的又一项休闲娱乐。

本文吸取了前人在网络游戏引擎方面的研究成果,重点对网络游戏引擎的系统结构方面做了深入的研究,先从宏观上说明了游戏引擎的组成部分,把这些分为系统、控制台、支持、渲染器/引擎内核、游戏界面、游戏本身以及工具/数据等部分,论述了每个组成部分的工作原理。通过对国外先进引擎的分析学习,完成了一个手机 3D 游戏引擎的系统架构的设计。

本文所论述的游戏引擎是用于控制所有游戏功能的主程序,而手机 3D 游戏引擎又具有小型化、多模块、简单化这几个特点,针对这几个特点本文将引擎设计的内容分为以下几个模块:用户界面、渲染、流程控制、碰撞检测以及摄像机这五个模块。最后协同配合大连泰康科技有限公司的同仁,在“3D 高尔夫”的项目工作中,设计了完整的游戏流程及一个小型的手机 3D 游戏引擎,并完成了引擎中最主要的五个功能类的实现。

关键词: 手机游戏; 系统架构; 游戏引擎

Research&Development Of 3D Game Engine For Mobile Based On J2ME

Abstract

With the fast development of telecommunications and computer technologies, games are not only played on computer but also mobile. With the development of hardware, mobile has become a remarkable game platform. Though being under the limitations of computing, RAM, graphics and sound effect, mobile phone has a good future because of the huge user group. Enjoying games while waiting bus and shopping, it is another amusement that modern technologies can offer.

This paper has drawn forefathers' research results in network game engine, have done deep studying to the systematic structure of the network game engine especially. From on macroscopic explained the game engine constituent first, divides into the system, Console, Support, Renderer/Engine Core, Game Interface, Game, and Tools/Data. Complete the design of a 3D game engine's system architecture by large numbers of literature searches and learning the foreign knowledge that is advanced.

This paper elaborates game engine is used for controlling the main program of all game functions, and the mobile 3D game engine has has the miniaturization, the multi-modules, simplification these characteristics. In view of these characteristics, the content designs which the engine divides into Screen, Renderer, Navigator, Collide and Camera. Finally coordinates the colleague of Dalian Taikang Science and Technology Limited Company, in "3D golf" Project work, have designed the integrity game flow and a small mobile 3D game engine, have finished the realization of five main functions in the engine.

Key Words: Mobil-phone Game; System Architecter; Game Engine

独创性说明

作者郑重声明：本硕士学位论文是我个人在导师指导下进行的研究工作及取得研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得大连理工大学或者其他单位的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的贡献均已在论文中做了明确的说明并表示了谢意。

作者签名： 赵晖 日期： 2025.12.25

大连理工大学学位论文版权使用授权书

本学位论文作者及指导教师完全了解“大连理工大学硕士、博士学位论文版权使用规定”，同意大连理工大学保留并向国家有关部门或机构送交学位论文的复印件和电子版，允许论文被查阅和借阅。本人授权大连理工大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，也可采用影印、缩印或扫描等复制手段保存和汇编学位论文。

作者签名: 赵 晖

导师签名: 李 慧

2005 年 12 月 25 日

1 绪论

1.1 课题来源

大连泰康科技有限公司研发项目《3D 高尔夫》手机游戏中引擎架构部分。运用光学、力学、运动学等科学原理开发实时的通用 3D 游戏引擎。像动画, 电视游戏, 网络游戏的开发一样, 手机游戏的开发需要一个引擎, 有了这样引擎之后, 手机游戏的开发就变得简单可行。从长远的观点出发, 开发一个 3 维手机游戏的引擎是 3 维手机游戏制作的第一步, 运用这个引擎去生产 3 维手机游戏是第二步。

1.2 选题背景

随着网络游戏的发展和手机硬件的性能的不断提高, 手机游戏也会以很快的速度追赶网络游戏, 也就是说网络游戏今天的成功可以说是手机游戏的明天的希望。由于手机集众多功能于一体, 成为人们永远离不开的宝物, 因此手机的数量将远远超过计算机的数量, 人手一机在不远的将来将成为现实^[1]。鉴于手机的数量增加和性能的提高, 人们对手机的服务内容的要求也会越来越多, 像手机游戏这样一种老中少皆宜的服务内容无疑会被视为力所能及的商品去消费, 而这种消费又是周期性很快的消费, 因此它的市场是可想而知的, 并且是不受时间约束的。手机游戏将是手机多媒体时代最受欢迎, 最有价值的应用游戏是多媒体应用中集大成的业务, 图像、音乐、互动都能在游戏中实现, 充分满足用户各方面的需求。手机已经与现代生活方式紧紧地结合在一起, 他们是最普遍携带的个人用品中仅次于钥匙和钱包的东西, 手机游戏则每个人都可以访问的到——随时, 随地。尽管每个手机游戏都不贵, 但是巨大的使用量(如: 每人每星期一个新游戏)将使得这个市场商机无限。

1.3 研究现状

目前国际上商用的手机 3D 游戏引擎只有一款。Bobber Tec 公司的 DRG 3D 游戏引擎技术, 是唯一一个可以实现在 GBA 上真正运行由 PC 平台上的 QUAKE I 和 QUAKE II 引擎开发的的游戏的技术。这次新发行的版本增加了对许多 60 系列手持设备的支持(包括 7650、3650 和 N-GAGE 以及其它很多型号), 并具有支持所有 DRG 3D 平台的跨平台特性。在 PC 上进行的开发及测试工作已基本完成, 现连带一套客户工具一同转移到目标操作平台; 由于新的工具兼容各种现有的 QUAKE 类游戏的编辑器, 使得对其学习使用的过程得以缩短, 取得事半功倍的效果^[2]。DRG 3D 新增的跨平台兼容性使得开发者

可以多次利用相同的内容和效果来创造复杂的 SKU（库存单位），而不需要在维持目标个体图案的质量和特征时为一个传统接口或转换花费时间和精力。

北京数位红软件应用技术有限公司用了两年的时间总结开发出一套基于移动设备的游戏引擎：Dragon Bone Game Engine for Mobile Phone（龙骨），它可以支持目前大部分的高端移动设备，它不仅可以使程序开发人员及游戏项目节省了大量的时间，另外还可以让美术人员最直观的看到其设计效果。它最大的价值就是可以使开发人员的一套代码在不需要改动程序的情况下编译运行在不同设备平台上（改动不超过 20%）^[2]。目前数位红及其他公司已经使用它开发游戏超过了 30 款。引擎提供了非常完善的 9 个模块与 3 个增强模块，可以完全满足各类游戏开发的需要。游戏引擎可以直接嵌入到开发者的程序中，不需要额外的支持程序即可使所开发的游戏独立运行。游戏引擎简单、易用，编译后体积很小，占用内存也很少，可以节省玩家的内存空间及无线下载费用。

网络游戏造就的“盛大奇迹”，鼓舞着众多知名企业和年轻的创业团队投入到手机游戏产业。手机游戏，特别是手机网络游戏的市场规模，给人以极大的想象空间。一个新兴产业在暗暗中迅速发展着，这就是手机游戏产业。

根据信息产业部截止到 2004 年 10 月的统计数据，PC 的市场保有量为 6900 万台，相应的 PC 游戏用户达到 2300 万，市场规模为 25.7 亿元。PC 网络游戏市场还在迅速发展中^[3]。即使以 2004 年的网络游戏普及率来简单推算手机游戏市场，就有让人吃惊的市场潜力。2003 年中国手机游戏市场规模为 3.07 亿，随着整个产业对手机游戏的重视，游戏被列入国家 863 计划，2004 年手机游戏取得较快的发展，2004 年全年手机游戏市场实现 8.84 亿的产值。业界预计，随着 WAP 和 KJAVA/BREW 等游戏业务的进一步成熟，2005 年国内手机游戏市场规模将达到 13.28 亿，比 2004 年增长 56.7%；2006 年手机游戏市场将增长到 22.84 亿，比 2005 年增长 71.9%。中国手机游戏产业的发展已经受到了国际资本的重视，众多风险投资机构纷纷抢滩中国手机游戏市场。2004 年 1 月，软银（Softbank）向杭州联梦娱乐投资 200 余万美元。5 月 17 日，杭州联梦以“联梦活力世界”（8100.HK）的名义在香港创业板借壳上市。2004 年 4 月，魔龙（moloan）获得 IDG 投资 500 万美元。

全球范围看，手机游戏增长势头远远超过其他移动数据业务，前景十分看好，已作为移动数据业务中的杀手级应用。手机游戏用户数量在 2003 年达到 1.8 亿，至 2009 年将超过 10 亿。来自欧洲的权威调研资料显示，随着手机游戏在欧洲开始起步，在未来 5 年内，手机游戏将成为 30 亿欧元的大市场^[3]。另据 IDC 预计，仅在美国，整个手机无线游戏业务的硬件、软件及服务所带来的收入在 2006 年将达到 40 亿美元，预计到 2006 年，全球的无线游戏市场总值将达 170 亿美元。如此大的市场前景，为运营商、游戏开

发商、手机厂商以及 SP 等产业链上每一个环节都带来了商机。诺基亚公司还宣称，到 2006 年，来自手机游戏的收入要占到其总收入的 40%^[3]。

虽然目前手机游戏还无法与 PC 游戏相比，其显示功能和键盘输入、声音处理能力也形成一定制约。但是随时随地可操作可联网的特性大大拓展了人们娱乐休闲的时间和空间，以游戏功能为主的手机终端在市场上渐被认可，多媒体功能正向手机的标配功能演进，制约手机游戏发展的技术瓶颈将被打破。当前的手机游戏产业尽管仍然存在很多制约其发展的瓶颈，但手机游戏已显示出非比寻常的发展速度，其市场潜力十分巨大，将成为移动增值服务市场新的增长点。

1.4 研究内容

电子和电脑游戏一直以来就是人们所喜欢的娱乐方式，它们在带给人们欢乐的同时还起到益智、锻炼人的反应能力等作用。而手机游戏是信息产业的兴起给人们带来的又一项新的娱乐方式。这些游戏操作简单、娱乐性强、老少皆宜，非常适合手机用户在闲暇时休闲娱乐。在最近几年，手机游戏成绩斐然，成为 IT 产业中增长最快的部分之一。

纵观 IT 产业的历史，计算机游戏已经成为技术创新背后的动力之一^[3]。计算机游戏者渴望更加强大的硬件计算能力，渴望不受不同的软件的限制。游戏开发者和玩家常常是前沿计算机技术的最早的采用者。随着技术对游戏经验的促进，游戏市场持续增长。游戏行业中最新的热点就是可移动性。

游戏开发可以说是集合了计算机行业中最先进的硬件技术和最新的编程思想，比如现在游戏都采用了面向对象的编程技术，大部分高质量图像的游戏都支持或者要求图形加速卡，同时游戏编程也有自己的基本理论和结构设计方法^[2]。在游戏开发中，核心的技术是构架游戏引擎，它是决定游戏质量的关键。游戏引擎是指为制作游戏需要而事先编制好的一系列同游戏初始化、图形处理、角色控制、碰撞检测等有关的 API 函数，它相当于游戏程序员的开发工具包，有些商业化的游戏引擎同时还提供了关卡和角色编辑器。这样游戏程序员就不必从最底层开始编写游戏，只需要调用游戏引擎中相关的 API 函数。同样在游戏里，剧情的进行、形形色色的角色衬托、各种场景的变换，都是由游戏引擎事先就约定俗成一种模式、一个大体的框架，然后依靠一些复杂的数据库来组织完成的。现阶段各大游戏开发商都以开发 3D 游戏为主要目标，3D 游戏引擎是网络游戏市场的主流。

本文主要从事在项目底层的游戏引擎设计上的工作，预期能够完成一个规模较小的引擎，力求实现一个完整的小型手机 3D 游戏引擎的架构。

1.5 本文的组织结构

本文将从以下几部分展开论述：

第一部分，主要对本文的研究背景、内容以及相关领域的研究现状进行说明。

第二部分，介绍本文所使用的开发技术——面向移动的 J2ME 技术。

第三部分，本文的设计、实现部分。详细描述了对手机 3D 游戏引擎的结构分析，以及 3D 高尔夫游戏系统引擎的设计与实现。

2 面向移动的 J2ME (Java 2 Micro Edition) 技术

随着无线应用与嵌入式系统的愈来愈受到人们的重视, SUN 也推出了 J2ME 来迎合这一趋势。SUN 认为, 未来的信息设备将比我们今天的更为 customizable 和 personal。这一点我们已经可以从手机和 PDA 的流行程度看出。所以为了开发面向这些设备的应用软件, SUN 推出了 J2ME(Java2 Platform, Micro Edition)。

J2ME(Java 2 Micro Edition)技术在 1999 年的 JavaOne Developer Conference 大会上正式推出, 是 Java2 的一个组成部分, 它与 J2SE、J2EE 并称。根据 Sun 的定义: J2ME 是一种高度优化的 Java 运行环境, 主要针对消费类电子设备的, 例如蜂窝电话和可视电话、数字机顶盒、汽车导航系统等等^[4]。它将 Java 语言的与平台无关的“Write once, run anywhere”特性移植到小型电子设备上, 允许移动无线设备之间共享应用程序。用 Java 编写的程序可以运行在不同的终端上, 开发商无须为不同的终端建立特殊的应用, 制造商也只需要简单地使它们的操作平台可以支持 Java 便可。

2.1 J2ME 技术架构

(1) J2ME 的体系结构

如同各种 SUN 所展现的 Java 技术, J2ME 的每种相关技术都经由一定的步骤来制定。这个过程就叫做 JCP (Java Community Process, <http://www.jcp.org>)。所有和 J2ME 相关的规范都可以在 <http://jcp.org/jsr/tech/j2me.jsp> 找到。J2ME 体系结构是基于设备的系列和类别的。J2ME 把设备分为两类, 一种是联接设备 (Connected Device), 一种是有限联接设备 (Connected Limited Device)^[5]。前一种对应于那些有电源的, 电力充裕, 较大的设备。例如: 电视机, 冰箱等, 后一种对应于主要使用电池, 小型的设备, 例如: 手机, PDA 等。这里本文只涉及第二种, 有限联接设备 (Connected Limited Device)。

为了支持资源受限设备系列所要求的那种灵活性和可定制部署, 人们将 J2ME 体系结构设计成模块化的和可伸缩的。J2ME 技术在一个完整的应用程序运行时模型中定义了这种模块性和可伸缩性, 在该模型中的四个软件层都构建在设备的主机操作系统上。J2ME 平台是由配置 (Configuration) 和简表 (Profile) 构成的。配置是提供给最大范围设备使用的最小类库集合, 在配置中同时包含 Java 虚拟机。简表是针对一系列设备提供的开发包集合。在 J2ME 中还有一个重要的概念是可选包 (Optional Package), 它是针对特定设备提供的类库^[6]。如图 2.1

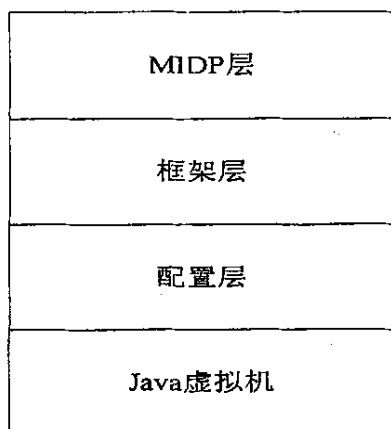


图 2.1 J2ME 的技术架构

Fig. 2.1 Technological Framework Of J2ME

从上图可以看出 J2ME 体系一共分四层：①Java 虚拟机层（Java Virtual Machine Layer）：这一层是 Java 虚拟机的一个实现，它是为特定设备的主机操作系统定制的，而且支持一个特定的 J2ME 配置（configuration）；②配置层（Configuration Layer）：配置层定义了 Java 虚拟机功能的和特定类别设备上可用的 Java 类库的最小集。从某种程度上说，一个配置定义了 Java 平台功能部件和库的共同性，开发者可以假设这些功能部件和库在属于某一特定类别的所有设备上都是可用的。用户不太会见到这一层，但它对框架（profile）实现者非常重要；③框架层（Profile Layer）：框架层定义了特定系列设备上可用的应用程序编程接口（API）的最小集。框架在一个特定的配置上面实现。应用程序是针对特定的框架编写的，因此可以移植到支持该框架的任何设备上。一个设备可以支持多个框架。用户和应用程序供应商看到最多的就是这一层；④MIDP 层：移动信息设备框架（Mobile Information Device Profile（MIDP））是一个 Java API 集合，它处理诸如用户界面、持久存储和联网这样的问题。Java 虚拟机层（Java Virtual Machine Layer）、配置层（Configuration Layer）和框架层（Profile Layer）一起构成了连接限制设备配置（Connected Limited Device Configuration（CLDC））。MID 框架（MID Profile）和 CLDC 提供了一个标准的运行时环境，这个环境允许把新的应用程序和服务动态地部署在终端用户设备上。

(2) J2ME 的实现

SUN 把 J2ME 的实现分为两层：configuration 和 profile。

设备的配置 configuration 是指针对某一类设备的最小的 Java 平台。configuration 包括虚拟机（virtual machine），核心的类库与 API。configuration 层定义一个 Java 虚拟机的特性与 Java 类库的最小子集。也就是说，configuration 层提供了开发人员一个最基础、

最核心的 Java 平台^[7]。有限连接设备配置 Connected Limited Devices Configuration (CLDC) 就是为有限连接设备定义了一个基本的 J2ME 运行环境。CLDC 的核心是虚拟机和核心类库。虚拟机运行在目标操作系统之上, 对下层的硬件提供必要的兼容和支持, 核心类库提供操作系统所需的最小的软件需求。

由于 CLDC 要面向尽可能多的设备, 而这些设备所使用的硬件又各不相同。因此 CLDC 规范中并没有指明需要某种硬件支持, 只是对设备的最小内存进行了限制。CLDC 规范中要求硬件必须达到以下要求: 至少 160KB 的固定内存以供虚拟机和 CLDC 核心类库使用; 至少 32KB 的动态内存以供虚拟机运行时使用(堆栈等)。这里所说的固定内存是指拥有写保护, 不会因关机而抹去的 ROM。对于具体的设备的具体实现, 这些需求也可能有变化。这里所规定的 160KB 是 CLDC 规范中的要求, 实际也可以是 128KB 左右^[7]。

和硬件类似, CLDC 上运行的软件也是多种多样的。例如, 有些设备支持多进程操作系统或者支持文件系统; 而有些功能极其有限的设备并不需要文件系统。对于这些不确定性, CLDC 只定义了软件所必须的最小集合。CLDC 规范中要求操作系统不需要支持多进程或是分址空间寻址, 也不用考虑运行时的协调和延迟; 但是必须提供至少一个可控制的实体来运行虚拟机。

在应用方面 CLDC1.0 版本定义了以下功能: ①Java 核心语言与 Java 虚拟机的特性; ②核心 Java 类库; ③输入/输出; ④对网络的支持; ⑤对安全性的支持; ⑥对国际化的支持。CLDC 不包含的功能有: 对应用程序生命周期的管理、用户界面、事件处理、高级应用程序模式(这里指用户与应用程序的交互)等, 这些功能要靠其上层的 Profile—MIDP 来实现^[7]。

Profile 层也包含一组 API, 主要针对于特定的某一族系的设备而定义。profile 层在特定的 configuration 层上实现; 而程序员则负责在特定的 profile 上编写应用程序。对于手机, PDA 等 Connected Limited Device。它们的 profile 层称为 MIDP (Mobile Information Device Profile)^{[6][7]}。MIDP 是植基于 J2ME 架构的技术套件, 主要提供一组无线通讯平台的环境, 可用来开发可移动小型资讯设备上所需的应用系统, 如移动电话与 PDA。MIDP 提供手机所需的各种核心功能, 包括使用者介面、网络连接、资料存储等, 是一个完整且标准的 Java 应用程序界面(API)的总集合。于是 MIDP 与 CLDC 合在一起就构成了一个完整的 J2ME 架构。

图 2.2 显示了开发手机应用的结构。

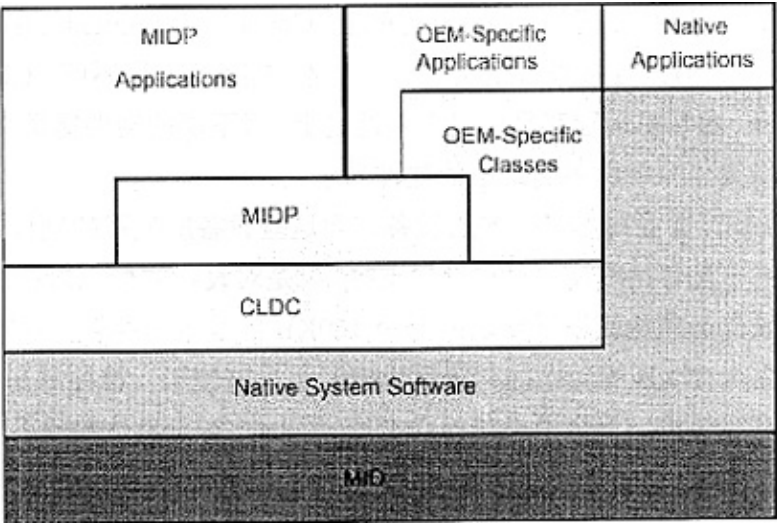


图 2.2 结构图

Fig.2.2 The Diagram Of The Structure

对于 MIDP 上的应用程序 SUN 也有独特的叫法，它们被称为 MIDlet。正如 Java 中的 applet。它是被 Application Management Software (AMS) 管理的^[8]。AMS 负责 MIDlet 的安装、下载、运行和删除等操作。在被 AMS 管理的同时，MIDlet 可以和应用管理软件通信通知应用管理软件自己状态的变化，通常是通过方法 `notifyDestroyed()` 和 `notifyPaused()` 实现的。

MIDlet 有三个状态，分别是 `pause`、`active` 和 `destroyed`^[8]。在启动一个 MIDlet 的时候，应用管理软件会首先创建一个 MIDlet 实例并使得他处于 `pause` 状态，当 `startApp()` 方法被调用的时候 MIDlet 进入 `active` 状态，也就是所说的运行状态。在 `active` 状态调用 `destroyApp(boolean unconditional)` 或者 `pauseApp()` 方法可以使得 MIDlet 进入 `destroyed` 或者 `pause` 状态。值得一提的是 `destroyApp(boolean unconditional)` 方法，事实上，当 `destroyApp()` 方法被调用的时候，AMS 通知 MIDlet 进入 `destroyed` 状态。在 `destroyed` 状态的 MIDlet 必须释放了所有的资源，并且保存了数据。如果 `unconditional` 为 `false` 的时候，MIDlet 可以在接到通知后抛出 `MIDletStateChangeException` 而保持在当前状态，如果设置为 `true` 的话，则必须立即进入 `destroyed` 状态。图 2.3 说明了 MIDlet 状态改变情况：

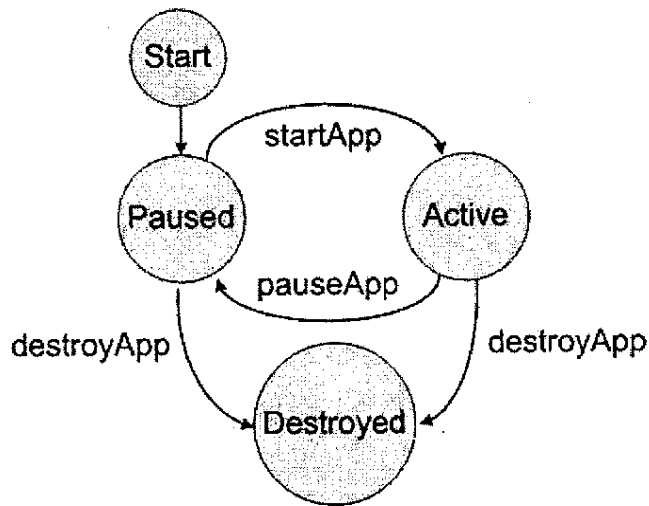


图 2.3 MIDlet 状态图

Fig. 2.3 The Diagram Of MIDlet State

下面通过一个例子来验证 MIDlet 应用程序的生命周期：

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
```

```
public class HelloWorld extends MIDlet{
    private Display display;//构造函数
    public HelloWorld(){
        display = Display.getDisplay(this);
        System.out.println("Constructor");
    }
    public void startApp(){
        System.out.println("startApp is called.");
        Form f = new Form("HelloTest");
        display.setCurrent(f);
    }
    public void pauseApp(){
        System.out.println("pauseApp is called.");
    }
    public void destroyApp(boolean unconditional){
        System.out.println("destroyApp is called.");
    }
}
```

编译该程序后运行，控制台上就会依次输出：

Constructor startApp is called.

当我们退出该程序时，控制台就会输出：

destroyApp is called.

2.2 MIDP 手机程序开发流程

完整的 MIDP 手机程序开发流程如表 2.1：

表 2.1 手机程序开发流程

Tab. 2.1 Procedure Of The Mobile-Phone

流程	工具	输入	输出
编译	javac.exe 编译	源文件 (*.java)	未混淆的类文件
混淆	第三方提供的工具	未混淆的类文件 (*.class)	混淆后的类文件 (*.class)
预审	preverify.exe 预审核	混淆后的类文件 (*.class)	经过预先审核的 类文件 (*.class)
包	jar.exe 打包	经过预先审核的 类文件 (*.class)	包文件 (*.jar)
编写描述文件	文本编辑工具		描述文件 (*.jad)
安装运行	传输工具 (IR/BT/数据线/OTA)	包文件 (*.jar) 和 描述文件 (*.jad)	在仿真器或手机 上正式运行

编译就是将我们所编制的 *.java 文文件，编译成为二进制的 *.class 文件（计算机只认识二进制！）。javac.exe 是由 SUN 公司编写的一个编译器，它可以把 *.java 文件编译成为 *.class 文件。*.class 文件是与类的个数相对应的，如果一个 *.java 文件中定义了三个类，它将被编译成三个 *.class 文件。

由于 class 文件格式透明的缘故，java 文件很容易被反汇编。因此，如果不希望源代码外泄，就一定要进行混淆（obfuscate）。所谓混淆，就是利用工具，将方法名、类名改成没有实际意义的特定的字符及代号，增加阅读的难度。这样就充分的保护了我们自己的知识产权。而且混淆还有个意想不到的好处，就是减少程序的大小。这是由于混淆器将设定的方法名、类名变成没有意义的短字符或代码，无形中减少了程序的大小。对于手机程序设计来讲尤为重要，每 K 的减少都意味着可以获得更多的空间。混淆器都是第三方软件开发商提供的，许多都是开源的，可以免费使用。常见的混淆器见表 2.2：

表 2.2 混淆器
Tab. 2.2 Obscure

名称	地址	特点
JODE	http://jode.sourceforge.net/	开源
ProGuard	http://proguard.sourceforge.net/	开源
RetroGuard	http://www.retrologic.com/	开源，中国移动百宝箱 强制使用
DashO	http://www.preemptive.com/	商业软件，一般专业公 司使用，昂贵
ZKM	http://www.zelix.com/	商业软件可试用
JBuilder	http://www.borland.com/	集成开发环境中内附混 淆功能，但 JBuilder 的价格 也不便宜。

从混淆程度，功能性来看 ZKM 和 JAX 都不错，但 JAX 是 IBM 开发的产品，因此也继承了大部分 IBM 产品的最大特色，就是“功能超强，但不易使用”，可能光是看完 JAX 的设定文件就是一个很大的问题。所以从使用上的方便性来考虑，还是向大家推荐 ProGuard 和 Retroguard，除了使用方便外，最重要的是它们都是 Open source 的软件。

ProGuard 是一个非常好的混淆器。它的特点是除了可以混淆保护你的代码，还可以自动检测和删除没有用到的类、方法和数据成员，还可以去掉包的层次关系，最大限度的缩小你的 jar 包。使用 ProGuard 的另外一个显著意义就是，因为它可以自动删除不用的部分，大可以放心的使用模块级重用，而不是通常 J2ME 程序员所采用的代码级重用，大大提高开发效率^[8]。但是一个问题是 ProGuard 还不能集成到 JB 和 WTK 等开发环境（不过 WTK2.0 自带），而要自己写命令行来使用，这个是制约 ProGuard 使用的最大障碍。不过 NETBEANS4.1 已经集成了 ProGuard。RetroGuard 也是一个很不错的 Java 混淆器，而且在 JBuilder7 的企业版中也带了这个混淆器，并且，在 SUN 发布的标准开发工具中也提供和这款混淆器相配套的设置（SUN 本身并不提供 RetroGuard，但是当你选择需要混淆时，它会提示你到 RetroGuard 的网站上自己下载，并且告诉你要把下载到的 RetroGuard.jar 放在哪里才可以用）。以 SUN 的 Java Wireless Toolkit 举例，当你在工程中选择 Package->Create Obfuscated Package 项时，如果它找不到 RetroGuard.jar，就会提示你下载这个包，并告诉你要把它放在 JWT 安装目录中的 bin 目录下。放好后，就可以正常进行混淆了。

在完成编译后,我们必须要对*.class 文件进行预审核,这和传统的 Java 程序(Applet、Servlet)是不同的。因为 class 在传输过程中容易损坏或是被篡改,传统的 Java 程序在运行前,都在本地机器上对.class 进行 Byte Code 的审核。而对于手机这样的资源有限设备而言,在手机上进行大量的此类的审核是极为浪费资源(如占用 CPU 的时间、消耗电力等)。因此,我们必须先在 PC 机上使用 preverify.exe 进行一部份预选审核工作。这样,在手机上进行的审核工作就大量减少了。

MIDP 可执行文件后缀名为 jar。利用 jar file.class 就能将通过预审核的*.class 文件,打包成 MIDP 认可的可执行文件。后缀名为 jad 的文件是 jar 文件的描述文件。有了 jar 及 jad 文文件后,我们就可以把它们放到仿真器或手机上运行了。至于如何把它们放到手机上,根据手机的功能不同,有四种方法可以选择:①使用数据线,将 PC 与手机相连,下载文件;②使用红外线;③使用蓝牙;④使用 OTA 空中下载(利用短信/WAP) [7]。

2.3 J2ME 的 3D 技术

3D 图形技术在各个领域已经越来越多的被应用了,当然这也包括了 J2ME 领域。在 J2ME 中为我们提供了 JSR184 这样一个可选包,该套 API 实现手机上 3D 图形的编程。同时也伴随着移动设备硬件的发展,现在也出现了越来越的支持该可选包的手机了,例如 Sony Ericsson 的 K 系列、S 系列等等。

(1) JSR184 简介

JSR184 标准(M3G: Mobile 3D Graphics)为 Java 移动应用程序定义了一个简洁的 3D API 接口, J2ME 程序可以非常方便地使用 M3G 来实现 3D 应用比如游戏等等^[9]。M3G 被设计为非常轻量级的,整个 API 的完整实现不超过 150kb。M3G 是 J2ME 的一个可选包,以 OpenGL 为基础的精简版,一共有 30 个类,运行在 CLDC1.1/CLDC2.0 上(必须支持浮点运算),可以在 MIDP1.0 和 MIDP2.0 中使用^[8]。目前,支持 M3G 的手机有 Nokia 6230/3650/7650/6600、Siemens S65/CX65/S55/M55、Sony-Ericsson K700i/P800/P900、Moto 220/T720 等。M3G 只是一个 Java 接口,具体的底层 3D 引擎一般由 C 代码实现,比如许多手机厂商的 3D 引擎采用的便是 SuperScape 公司的 Swerve 引擎,这是一个专门为移动设备设计的高性能 3D 引擎。

类似于 Microsoft 的 D3D, M3G 支持两种 3D 模式:立即模式(immediate mode)和保留模式(retained mode)。在立即模式下,开发者必须手动渲染每一帧,从而获得较快的速度,但代码较繁琐;在保留模式下,开发者只需设置好关键帧,剩下的动画由 M3G 完成,代码较简单,但速度较慢^[9]。M3G 也允许混合使用这两种模式。

如果熟悉 OpenGL, 那么 M3G 是很容易理解的。在 M3G 中, Graphics3D 是 3D 渲染的屏幕接口, World 代表整个 3D 场景, 包括 Camera (用于设置观察者视角)、Light (灯光)、Background (背景) 和树型结构的任意数量的 3D 物体。3D 对象在计算机中用点 (Point, Pixel)、线 (Line, Polyline, Spline)、面 (Mesh) 来描述, 具体存储和运算 (如旋转、投影) 都是矩阵运算和变换^[9]。在现实生活中, 我们通过眼睛来观察这个世界, 我们生活在一个 3 维坐标系构成的世界。在 Mobile3D 中同样的有 World 类让你随心所欲的构造你想要的世界, 在 3D 制图中它被称为“场景”; 另外还有 Camera 类来作为你的眼睛, 你可以来设置它的位置角度等参数, 来显示出不同的图像。那么在 Mobile 3D 中是如何实现 3D 图像的显示呢? 首先你要建立或者载入 3D 模型, 然后根据需要设置场景中的环境、渲染方式等一系列的参数, 再后生成并设置一架摄影机并且调整好你想要的光线, 调整好你所需要的位置和角度。OK 还需要什么呢? 按下快门, 这一步就是被叫做“渲染”一切就搞定了。

尽管 3D 模型可以在程序中创建, 但是非常繁琐。因此, M3G 提供一个 Loader 类, 允许直接从一个单一的.m3g 文件中读入全部 3D 场景, 导入后你就可以直接获得 World 的实例对象, 这里有可能包括各种模型、摄像机、灯光、环境、材质以及三角面的各种信息^[14]。也就是说通过*.m3g 文件, 我们可以直接获得 World 的实例化对象。M3G 文件的创建可以通过下面这三种途径获得。

首先, 最新的 Discreet's 3D Studio Max 版本已经内建了一个 M3G 的导出器。我们只需要点击导出按钮, 你就能导出你建立的场景、动画、骨骼、材质等成为 M3G 文件。然而, 很多人发现 Discreet 的导出器比较麻烦并且有一些错误, 所以为了得到更好的效果, 我们使用第二种方法。

HI 公司是索尼爱立信公司的 JSR184 接口的提供者, 他们制作了针对 3 个十分流行的 3D 建模软件例如 3D Studio Max, LightWave 和 Maya 的插件。

Blender, 这是一个具有可用的 M3G 导出器的免费的 3D 建模工具, 然而它在开发的比较早, 并且有一些错误。

那么我们怎样在我们的程序中调入这些非常有用的文件呢使用一个简单的方法就能从一个 M3G 文件中载入所有的对象。一个可以是字符串类型的 URL 另一个方法是一个原始字节流中的偏移, 下面是如何使用的例子:

```
Object3D[] objects = Loader.load("file.M3G");  
Object3D[] objects2 = Loader.load(byteArray, offset);
```

这个 Load 方法通常返回的是一个 Object3D 类型的数组, 对于这点有一个非常重要的原因。最好的就是使得 Loader 类可以载入 M3G 文件更多的数据。它可以转换成任

何一个从 Object3D 类中继承过来的类。SUN 的 WTK2.2 已经内置了 M3G 的实现包, 如果安装了 WTK2.2, 就可以在模拟器上运行 3D MIDP 程序。

(2) 三维空间中对象的移动

当一个对象在 3D 空间移动时, 需要计算得到三个不同的速率: x 、 y 和 z 的速率。然而, 大多数的游戏可以将 x 和 z 速率合成一个, 就是前进速率^[10]。(例如, 如果前进速率是 1.0, 那么就使用正弦和余弦分别得到 x 和 z 轴上速率) 如何使用三个相区别的速率来描述一个在 3D 空间中移动的对象。完成这个最简单的方法, 就是通过使用三个向量。坐标向量: 第一个向量是一个对象的坐标向量。这是一个至关重要的向量, 它通过保持对象的 x 、 y 和 z 坐标的轨迹保存了对象在 3D 空间的当前位置。否则, 就无法确定在什么地方渲染这个对象; 速率向量: 第二个向量是速率向量。它定义了对象的坐标体系在下一次移动时需要移动多少个单位。因此它也需要 x 、 y 和 z 三部分。基本上你需要做的就是, 每次把 x 速率加到 x 坐标上, y 速率加到 y 坐标上, z 速率加到 z 坐标上。那么, 如果你有一个太空船, 每帧在 y 轴上径直向上一个单位, 那么速率向量就是 $(0,1,0)$; 加速度向量: 最后一个向量是加速度向量。虽然在这里我们用不到, 但是它仍然是很重要的。加速度向量工作原理和速率向量一样, 但是它作用的是速率向量而不在坐标系上。这意味着加速度向量不直接操作坐标系, 而是在每帧增加速率, 给玩家一个加速的感觉。正常情况下, 一个物体不可能从停顿状态立即达到最高速度。这需要时间, 并且为了使一个 3D 游戏更接近于现实, 需要加速度。假定, 一个太空船想从每帧 1 个单位开始移动并且加速到每帧 5 个单位。它可能具有一个速率向量 $(0,1,0)$ 和一个加速度向量 $(0,0.1,0)$, 这也就意味着在 40 帧后, 太空船将会达到每帧 5 个单位的速率。

三维空间中的对象通常是不透明的, 并且会和另外一个对象发生碰撞。一个彗星可能撞击一个行星, 一个司机可能会驾车闯进一幢大楼。然而, 在 3D 游戏中, 它并不像真实世界中那么明显。这是因为只渲染了对象, 并且渲染的对象也仅仅是被转化为屏幕上像素的 3D 模型的数学表示, 而没有考虑过两个对象之间有多少不同^[10]。例如如果移动太空船到一些可能存在的行星中, 它将会继续穿过这个行星, 然后出现在另外一面。这都是因为没有碰撞检测。

光线交叉是一种十分直观和简单的碰撞检测形式。至少对常规和简单的碰撞情况是这样。它的工作原理是: 希望检测碰撞的对象从它的中心 (或者身体的其它位置) 在一个给定的方向上 (通常是对象的速率向量) 放射出光线。光线在 3D 世界中传播, 直到它实际撞到某些物体上 (像激光一样)。当它撞上某些物体时, 就报告碰撞, 并且报告在它撞上一个物体前还有多少远可以移动。根据这个距离, 程序将决定这个碰撞是否是会实际发生。

M3G 为我们提供了一个异常简单的方式来计算碰撞。在我们的世界中的每一个 Group (Group 是 Node 的集合,可能是在世界中存在的任何事物)有一个叫做 pick 的方法。它的作用就是完成所有碰撞的方法^[10]。它是这样工作的,假如说想检查在一个房间中主人公是否和一面墙发生碰撞,或者是否和这个房间中一个怪物发生碰撞。那么需要做的就是创建两个 Group。一个为所有的墙,一个为所有的怪物。然后,只需要简单的向每一组投射光线,从人物出发并且朝着它的方向,然后检测是否会与某些对象发生碰撞。

3 手机游戏 3D 高尔夫系统的分析与设计

3.1 系统需求分析

目前电脑(PC)在网络游戏终端中占主导地位,使用它来玩网络游戏的玩家高达 9 成,其次就是手机,而在未来会玩网络游戏的人当中,而以手机作为网络游戏终端的玩家比例将必然呈现稳定的增长趋势,这是由于玩家们出于对网络游戏终端即开即用、操作简便、便于携带和经济实用等方面的需求,手机网络游戏版也相应成为游戏软件方面的新诉求。现在市场上充斥着很多练级的角色扮演游戏,大部分玩家对于休闲类游戏的兴趣是越来越大,而休闲类游戏一般是棋牌类游戏为主,可以说到目前为止,国内还没有一款大型 3D 在线竞技体育类网游^[11]。所以,将此类游戏定位于手机网络游戏版的市场前景还是很客观的。同时,众所周知,高尔夫球这项运动,对于绝大多数人来说,目前的实际情况是它非常的昂贵,普通人无法经常性的消费。所以,对于大部分只是在名字上知道高尔夫球而不很了解这项运动的人来说,这款游戏也是一个绝佳的机会。最后,这个游戏属于对高雅的高尔夫球运动的模拟,它的清新,健康,对于很多对网络游戏纯粹练级持反对或者厌恶态度的人来说,也无可挑剔。

3.1.1 系统主题

本系统定位于体育竞技类游戏,体育竞技类的游戏不注重故事情节,它只关注比赛,赛事的安排策划和具体实时操作是这类游戏的两大主线^[12]。相比角色扮演类游戏内容来说,体育竞技类游戏更容易上手,老少皆宜,因此更容易获得众多玩家的青睐。

打高尔夫球是一项具有特殊魅力的运动。它是人们在天然优雅的自然绿色环境中,锻炼身体,陶冶情操,提高技巧的活动。勤劳勇敢的苏格兰渔民,在 13 世纪就创立了这项运动,他们在海边用木棍把圆形的石子击入挖好的小坑中。这项运动在 1411 年成立的圣安德鲁大学流传起来。随后传入了苏格兰皇室、由于形式的不断成熟,正式成为了一项皇室运动。后来,它以它那难以抵抗的诱惑力很快流传到了欧洲的其他国家。时至今日,高尔夫已遍布全球,其中以日本、美国和英联邦诸国最为流行,不过由于场地和器械的限制,它仍然是一项平民无福消受的“贵族运动”。二十世纪,高尔夫运动传入我国。一九三一年,上海成立了高尔夫球游戏中心。同年,中、英、美商人合办高尔夫球俱乐部,在南京陵园体育场旁开辟高尔夫球场。

进入八十年代,高尔夫运动在我国得到很快发展。1985 年,中国高尔夫球协会成立,1986 年 1 月,中国首次国际高尔夫球赛“中山杯”职业、业余选手混合邀请赛,在中山市温泉高尔夫球场举行。以后,每年举办一次。最近几年,高尔夫球运动在我国已迅速

普及和发展起来。作为一种时尚或某种身份的隐约暗示,高尔夫球已逐渐渗透到我们的都市生活之中,并令不少人向往。。"高尔夫",本是英语 golf 的译音。在英语中, golf 一词是由绿 (green)、氧气 (oxygen)、阳光 (light) 和友谊 (friendship) 这四个单词的打头字母所组成的。针对高尔夫运动的这些特点,游戏的设计就必须同时具备观赏性、竞技性、交互性这三种特点。

(1) 观赏性: 高尔夫球场的形状和大小无统一标准, 一般设在风景优美地区和公园的草地上, 游戏中的球场以大连的海边高尔夫球场为设计原点, 加入南方风情的背景, 细致各种场景元素例如天空, 水面, 小岛 (周围的水要透明效果), 房子, 绿地, 沙地, 树, 洞口的小旗子等, 各种天气情况比如晴天的阳光明媚, 阴雨天, 雪天, 整个场景的雾蒙蒙。场景随着气候不同而变化。人物, 物品, 建筑物在不同天气的不同阴影, 营造出一种强烈的现实感。力求玩家在竞技的同时还能欣赏到球场美丽的风景。

(2) 竞技性: 在竞争激烈的职业高尔夫世界, 只能靠自身实力来夺得冠军。不断与自我作斗争, 永不消逝的体力, 惊人的集中力……冠军只属于最强者。一款好的游戏应当做到寓教于乐, 将健康的理念容于游戏中^[13], 使玩家在娱乐的同时体会到顽强的体育精神。

(3) 交互性: 将玩家之间的交流溶于游戏中, 在比赛竞技的同时体会到友谊 (friendship) 的意义。

3.1.2 系统特点

区别于以往的手机体育竞技类产品本系统的主要特点是:

(1) 社区型高尔夫 Online Golf Game

- ①可以在虚拟世界中进行交易, 使得玩家门享受更活跃的沟通乐趣。
- ②社区成员之间可以举行竞赛, 也可以参加其他巡回赛。
- ③选好玩家感兴趣的高尔夫项目后开始游戏。



图 3.1 玩家在广场

Fig. 3.1 User On The Square

图 3.1 显示的是几个玩家在广场散步聊天的场景, 体现了游戏的交互性。

(2) 富有实际感的职业高尔夫选手的击球动作

- ①与电视节目媲美的精致精美的人物设计。
- ②具有职业高尔夫选手击球时的真实感，根据玩家不同控制，可以实现各种击球动作。
- ③实现沙坑球、草痕、雪、雨等各种特殊效果。



图 3.2 击球动作

Fig. 3.2 Shot

图 3.2 显示的是向远处击球的一个动作，具有身临其境的真实感。

(3) 人性化的界面，精准的控制

- ①只要 4 个键，即可享受游戏乐趣，“一点即通”式的用户界面
 - ②各种直接、间接游戏信息的自动提示功能
 - ③随着级别的提升，游戏控制更灵活
 - ④根据球道（Fairway）、地堡、Ruff 等地形出现不同控制菜单。
- (4) 同实际击球一样的感觉，大大增加了 3D Golf Game 的现场感
- ①根据玩家的力气、当天的练习密度和疲劳值表现出不同的击打动作。
 - ②根据当天的天气情况，轨道也跟着变化，同实际击球一样的感觉和球场。
 - ③在任何高尔夫路线上都可以保持稳定的姿势，可以根据球位使用最恰当的姿势。



图 3.3 姿势

Fig. 3.3 Posture

上图显示了在球进到坑里的时候所选择的击球姿势，并且在这种状态下也能保持姿势的稳定。

(5) 独特的角色育成系统

①玩家可以通过积累经验值、账户充值、打社区比赛等方式换取游戏币，用于购买道具和变换服装。

②随着经验的增长玩家的力量、技巧等也都随之增长。



图 3.4 服装

Fig. 3.4 Clothing

玩家的红色球衣是需要用游戏币购买的。

3.1.3 技术分析

手机游戏受硬件设备限制不能象 PC 游戏一样随心所欲，虽然高配置的手机越来越多，但低端手机暂时还是主导着消费市场。J2ME 程序由于其特殊的运行环境限制，在设计与开发中应将尽量少地占用内存和 CPU 作为基本准则。对程序的优化采用使用混淆器的解决方案，系统的开发环境为 NETBEANS4.1，NETBEANS4.1 内部已经集成了 ProGuard 混淆器，除了可以混淆保护代码，还可以自动检测和删除没有用到的类、方法和数据成员，还可以去掉包的层次关系，最大限度的缩小 jar 包，代码能减少 30% 以上的容量。

在开发中另一个会占用较大内存空间的是图片，一张 3k 的图片可以占用 20 多 k 的内存。对于图片的优化在开发中采用压缩的方法来解决，在 photoshop 里图片制作完成后选择“存储为 web 所用格式”根据里面的选项进行压缩，特别是颜色这一项越小越好。然后进行再次压缩，使用可以压缩 png 格式的工具软件 xat.com Image Optimizer。能达到 70% 的压缩率且图像不会失真。游戏中大地图的绘制采用由小块地图拼出的方法，用一个数组来保存地图各个位置的地形和状态，然后统一的在 paint 方法中刷出整张地图来。针对硬件设备的不同，设计三种不同的地图压缩包：2D 地图；近景 3D 远景 2D 地图；纯 3D 地图，让用户根据实际条件进行选择下载。

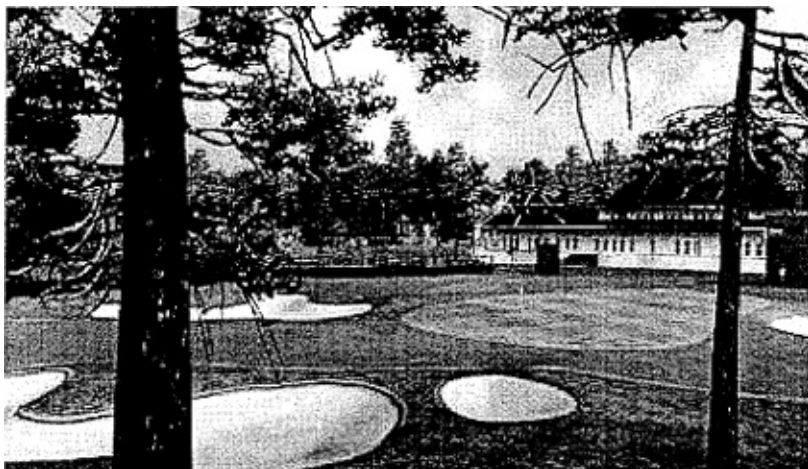


图 3.5 3D 地图

Fig. 3.5 3DMap

3.2 开发平台的分析

3.2.1 主流开发平台分析

目前开发客户端应用程序的选择有：WAP、QUALCOMM 的 BREW 平台和 Sun Microsystems 的 J2ME 标准。

(1) WAP

WAP (Wireless Application Protocol) 无线应用协议是一个开放式标准协议，利用它可以把网络上的信息传送到移动电话或其他无线通讯终端上。

WAP 是由爱立信 (Ericsson)、诺基亚 (Nokia)、摩托罗拉 (Motorola) 等通信业巨头在 1997 年成立的无线应用协议论坛 (WAP Forum) 中所制定的。可以把网络上的信息传送到移动电话或其它无线通讯终端上。它使用一种类似于 HTML 的标记式语言 WML (Wireless Markup Language 不是无线标记语言)，相当于国际互联网上的 HTML (超文件标记语言) 并可通过 WAP Gateway 直接访问一般的网页。通过 WAP，用户可以随时随地利用无线通讯终端来获取互联网上的即时信息或公司网站的资料，真正实现无线上网。它是移动通信与互联网结合的第一阶段性产物。WAP 能够运行于各种无线网络之上，如 GSM、GPRS、CDMA 等。WML 是无线标注语言 (Wireless Markup language) 的英文缩写。支持 WAP 技术的手机能浏览由 WML 描述的 Internet 内容。通过 WAP 这种技术，就可以将 Internet 的大量信息及各种各样的业务引入到移动电话、PALM 等无线终端之中。

(2) J2ME

Java 虽然发展迅速,但是 Java 2 体系庞大,不适合嵌入式平台,于是 SUN 公司针对嵌入式领域提出 Java 2 缩微版(J2ME),J2ME 可以看作是针对于嵌入式领域设计的精简 Java 版本,较小的 Java 内核、非常有限的用户界面、小屏幕等特性。

J2ME 是应用于移动通讯设备、嵌入式设备或消费性电器而推出的一项技术,是致力于消费产品和嵌入式设备的最佳解决方案^[14]。J2ME 在设计其规格的时候,先将所有的嵌入式装置大体上区分为两种:一种是运算功能有限、电源供应也有限的嵌入式装置(比方说 PDA、手机);另外一种则是运算能力相对较佳、并且在电源供应上相对比较充足的嵌入式装置(比方说空调机、电冰箱、电视机上的机顶盒)。因为划分了这两种类型,所以 J2ME 引入了一个叫做 Configuration 的概念,然后把运算功能有限、电源有限的嵌入式装置定义在 Connected Limited Device Configuration (CLDC)规格之中;而另外一种装置则规范为 Connected Device Configuration(CDC)规格。也就是说,J2ME 把所有的嵌入式装置利用 Configuration 的概念区隔离成两种抽象的型态。

J2ME 简单的可以理解为 Java 的移动版本,其 MIDP(Mobile Information Device Profile,移动信息装置简表)架构就是专门为手机与 PDA 而设计^{[6][14]}。目前 MIDP 有 1.0 和 2.0 之分,后者在兼容性以及游戏的多媒体表现力如音效和振动方面有更好的效果,此外通过测试软件显示,支持 MIDP2.0 的 Java 游戏在图形运行能力对比 1.0 版本也有更优秀的表现。目前支持 MIDP2.0 手机主要为 03 年下半年以后发布的手机如索爱 K700、摩托罗拉 V300、诺基亚 6230 等。

同时,以 J2ME 平台开发的程序叫做 MIDlet,可以直接从在线下载后离线运行,或从 PC 通过数据线、红外或者蓝牙传输到手机上^[14]。由于 J2ME 程序都很小,只有几十 K,所以又被称为 K-JAVA。由于 K-JAVA(J2ME)是开放平台,因此任何程序设计师甚至程序编写爱好者都能到 SUN 网站下载 J2ME 开发工具,为手机写 JAVA 程序。此外,J2ME 的 MIDlet 文件都是由 .jad(JAVA Description File)和 .jar(JAVA Package)呈现,不同的手机品牌在使用安装时会有不同的方式,如索爱的手机则仅需安装 JAR 文件即可;而三星和诺基亚手机则需要同时下载 JAD 和 JAR 文件。

由于 J2ME 的技术先进,而且又能跨平台运行,因此受到了众多手机厂家的支持,甚至成为现在用户购买手机的一项比较重要的技术指标。所以到目前为止,手机的游戏应用平台中,以支持 J2ME 的最多,而且 J2ME 软件开发商也有越来越多的趋势,J2ME 已成为 2002 年下半年之后的行动游戏平台主流。尤其在进入 04 年之后,基于 JSR184 的 Mobile 3D Graphics API for J2ME 的 3D 游戏开始出现,象索爱 S700/K700/Z500 以及西门子 S65 等新机都开始支持在手机上运行 3D JAVA 游戏。

(3) BREW

BREW 是 Binary Runtime Environment for Wireless 的缩写, 是 QUALCOMM 公司为无线数据应用程序开发和执行提供的通用内容。由于需要更少的内部应用程序开发和集成任务, OEM 可以更加快速地推出新设备。用户可以选择和下载适合自己个人喜好的无线软件。通过这种方式, 用户将推动新的无线数据应用程序和服务市场的发展。BREW 运行时环境是一个较薄的软件接口层, 位于芯片系统软件 and 应用程序间的无线设备上, 使最终用户可以无线下载程序并在启用 BREW 的设备上运行。BREW 具有较高的闪存和 RAM 使用效率, 它所占闪存较少, 而且在应用程序运行时会为其动态分配 RAM^[15]。

BREW 平台是一种开放的端到端的解决方案, 也是唯一能够使应用开发商、设备厂商以及网络运营商从数据业务中获益的途径。BREW 是一种技术, 同时也是一种灵活的“一揽子”业务系统。

3.2.2 各平台在应用上的比较

从实际应用来看这三种平台各有其不同特点

WAP(移动、联通): 优点是目前市场上的绝大多数手机都支持 WAP 功能, 技术相对简单, 开发成本低, 实现方便。缺点是通信数据流量大。界面设计风格简单, 不利于提供个性化服务。安全性较差, 不能实现实时在线购物、付款。

对于 J2ME(移动、联通)来说, 越来越多的手机开始支持 JAVA。通信流量小, 速度快。可以进行丰富的界面设计, 非常有利于提供个性化服务。安全性较高, 能够实现实时在线购物、付款。技术相对简单, 开发成本低, 实现方便。但缺点是程序运行相对速度较慢。

BREW(联通)的优点是通信流量小, 速度快。界面风格多样, 非常有利于提供个性化服务。安全性好, 能够实现实时在线购物、付款。程序运行速度快。但致命的缺点是支持的手机比较少。技术比较复杂, 开发成本较高。

从日前的手机市场方面来分析, 支持 WAP 的手机普及率是最高的, 其次是 J2ME, 支持 BREW 的手机实在太少, 而且联通对支持 BREW 平台不太放心, 而且有计划推出自己的私有应用平台。因此, 对于当前的市场而言, 选用 BREW 开发不太合适。从安全性方面来分析, WAP 技术的安全性太差, 没有有效的安全措施可以去实现;而相对来说, J2ME 和 BREW 的安全性可以做的很好, 安全措施的实现来说也比较灵活, 开发者可以自己定义加密、解密的方法。因此, 从安全性方面来讲, 不建议用 WAP 开发。从产品的性能方面来分析, WAP 技术的通信流量大, 因此容易造成通信速度慢, 无线终端(手机)和服务端端的负荷较重;同样相对来说 J2ME 和 BREW 的通信流量较小。因此, 从产品的性能方面来讲, 也不建议用 WAP 开发。从产品的开发成本来看, WAP 和 J2ME

的实现技术较为简单，开发所需人月数少，开发成本低，而且两者的扩展性和移植性也比较好。而 BREW 的技术较为复杂，开发难度大，而且相对来说扩展性和移植性也比较差。以 J2ME 技术来实现平台的搭建，其性价比最高，开发难度也小很多。

所以以目前的市场和技术，选用 J2ME 技术来开发无线应用是合适的。

3.3 系统总体设计

在此系统中总体设计方面的分工主要是游戏流程的设计与功能模块的划分。

3.3.1 游戏流程设计

系统设计的游戏流程如图 3.6 所示，从流程图我们可以看出手机游戏是要满足随时暂停、随时结束、随时存储功能的，而引擎是实现这些功能的基础。其中系统的大部分素材放在服务器的数据库内，每次运行游戏前，客户端程序自动到服务器上检索最新版本下载到本地，系统进入游戏主菜单。系统在界面上不提供保存按钮，在每一关过后系统自动根据当前用户 ID 临时保存游戏的进度，在退出游戏前系统自动提示询问是否保存游戏进度，如选择保存，则将临时保存的游戏进度正式保存；如不保存，则将临时保存进度删除。

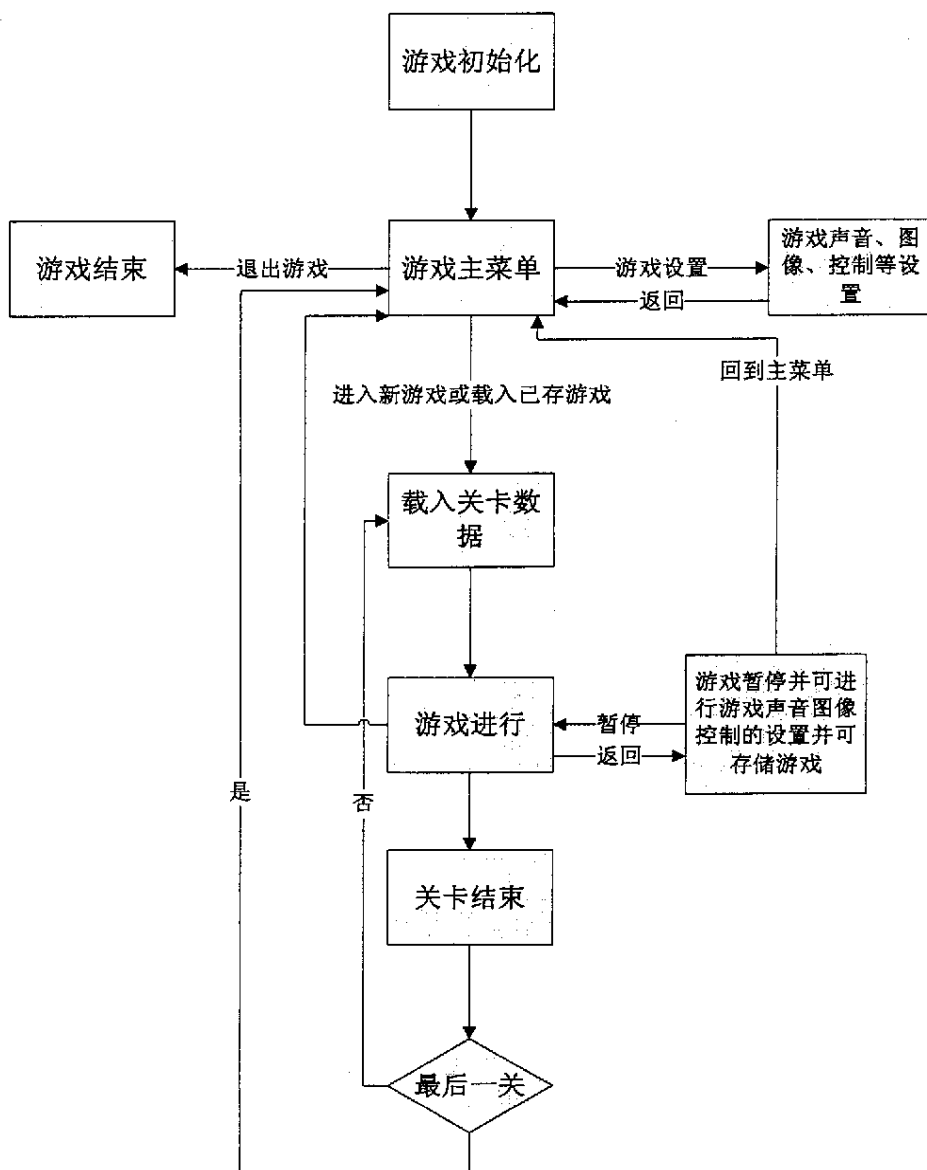


图 3.6 游戏流程

Fig. 3.6 Procedure Of The Game

3.3.2 游戏功能模块划分与描述

系统主要分为五大功能模块：单人模式、多人模式、交易区、游戏统计、系统设置。

(1) 单人模式

单人模式里面又包括 4 种选择：巡回赛模式、对战 NPC 模式——NPC 是指由服务器或电脑控制的角色、训练模式、在线模式。

①巡回赛模式

进入该模式后,要选择一个人物和球场,打满这个场地全部的 18 个洞。这个模式是最贴近真实比赛的模式了,需要极高的耐心和技术。比赛结束后,系统会统计出你的总杆数和积分,排出名次(和 NPC 比),击打杆数最少就可以获得冠军。如果你获得了第一名,会奖励你一个奖品(在九个奖品里面抽选一个,要看运气哦)和相应的经验值,每当经验值暂满之后,就可以参加提升等级的考试,考试通过就可以得到新球场和新的称号。

②对战 NPC 模式

1 对 1 的对战模式。刚开始的对手只有几个人,随着不断战胜他们还会触发隐藏人物的出现,有些人物在和他们比赛的时候还会提出苛刻的要求:强制你选择能力弱的角色、强制你选择有利他的球场、强制选择恶劣的天气。和巡回赛不一样,只有一半九个球洞,以每个洞为一个胜负点来决定输赢,每一个洞击打杆数最少的一方获得该胜负点,这时得分就会增加一分,人物头像旁边会显示“1UP”,对手会减一分,头像显示“1DW”。随着分数增加,会变成“2UP”“3UP”“4UP”“2DW”“3DW”...如果被追成平局,双方就都会显示“EVEN”。达到任意三个条件就可以打败这个人物:全部 9 洞结束,领先对方一分;领先的分数超过还未打完的洞数,如:领先两分,但是只剩下一个球洞,比赛就会提前结束;领先 4 分,就算还剩下 5 个球洞也会提前结束比赛。

③训练模式

给初学者用来修炼的模式,可以任意调整自己想要的环境。可以选择通常练习和推杆练习。通常练习:是模拟正式比赛的全过程练习,包括开球和最后的推杆。过程中按下 SELECT 键可以弹出另一个菜单,可以调整天气和球手,以及球杆和球;推杆练习:有长距离和中距离推杆,推杆的练习不像通常练习那么自由,规定 10 个球场,只给一次机会,不管是否进洞都直接跳到下一洞,全部推完后会统计出命中率。



图 3.7 推杆

Fig. 3.7 Push Rod

④在线模式

这个模式是各种级别的玩家的对抗比赛,依靠网络和全国各地的高手切磋的模式。可以 8 人无线对战。比赛胜者可获得相应的游戏币、经验值。第一次登陆游戏,首先要

创建人物。共有 8 种不同性格的人物可供选择。人物创建完之后选择你要进入的球场——练习场、初级球场、比赛场。进入每一个球场后,系统将会列出相关游戏室列表以及等候人列表,玩家自己也可以创建游戏室,游戏室的室长可以设定球场、击球时间、球数等。图中的两个玩家正在等待游戏开始。



图 3.8 等待中

Fig. 3.8 Waiting

(2) 多人模式

和亲朋好友一起享受天伦之乐的模式,通过红外联网,近距离的几个人可以一起对战,无需网络支持。操作方法与单人模式相同。

(3) 交易区: 交易区分为商店和二手街两个部分。

①商店

出售商品的地方,在这里可以购买各种各样的道具,从球杆、球到服装、鞋帽、甚至发型、饮料等在这里都可以买到,让玩家随心所欲的塑造个性化的人物。拥有适合自己的高尔夫球用具。

②二手街

是玩家与玩家之间的交易区,可以将自己的闲置道具放到信息发布板上,可以游戏币交易也可以以货易货。

(4) 游戏统计

游戏统计是一个个性化的设计部分,包括 4 个部分,玩家可以查看自己的道具装备,查看自己的成绩、回放游戏中的精彩表演(在此系统设计了自动记录功能,自动记录一般只记非常精彩的球),以及查看在线排行榜等。

(5) 系统设置

在系统设置中可以设置主界面的壁纸、选择自己喜欢的礼仪人物、以及选择自己喜欢的背景音乐。图 3.9 给出了在模拟机上实现的游戏画面,为击球前的姿势。



图 3.9 游戏画面

Fig. 3.9 Game Picture

3.3.3 游戏角色设定

游戏一共设计了 8 种不同形象的角色，每种角色都有相应不同的技术特点和性格特征，让玩家在竞技的同时也能体会到身临其境的真实感。如图 3.10 所示



图 3.10 游戏角色

Fig. 3.10 Game Role

这八种角色的性格特点分别是：(1)具有反抗性格，最年轻但充满魅力；(2)中性的气质，勇于接受挑战；(3)性格有些张扬却有很强的爆发力；(4)高贵典雅，球场上的焦点；(5)拥有奇怪力量的人；(6)智慧，谦逊的性格；(7)喜欢沉默，忍耐力超强；(8)可爱调皮的女孩子。

3.3.4 系统数据库设计

系统后台数据库共设计 8 张表：

(1) 目录

表 3.1 目录

Tab. 3.1 List

Name	Code	Comment
用户信息表	Userinfo	
成绩表	Grade	
素材表	Material	
更新记录表	Updaterecord	
客户机 IP 列表	Computer_ip	
进度保存表	Save_date	进度保存及临时进度保存
交易记录表	Deal_record	保存交易记录

(2) 用户信息表：

表 3.2 用户信息表

Tab. 3.2 User Info

Name	Code	Data type	Primary
自增 ID	ID	int	true
用户名	NAME	char(20)	false
密码	PASSWORD	int(6)	false
道具	TOOLS	image	false
积分	GRADE	int	false
货币	MONEY	int	false
客户机 IP	IP	char(15)	false
客户机版本号	CVERSION	char(20)	false

用户信息表用于存储用户的相关信息。包括用户名、密码、道具、积分、货币、以及便于管理员管理的客户机 IP、客户机版本号等相关信息。

(3) 成绩表：记录用户在各项比赛中的积分

表 3.3 成绩表

Tab. 3.3 Grade

Name	Code	Data type	Primary
自增 ID	ID	int	true
用户名	NAME	char(20)	false
积分	GRADE	int	false

(4) 素材表

方便数据库对可更新的素材进行维护，包括增加和删除素材，但不允许修改，可以删除后再增加。在进行素材维护的过程中，需要对素材的存储路径、文件名和最新版本号进行设置，这样才能使用户在自动更新时进行正确的更新操作。

表 3.4 素材表

Tab. 3.4 Material

Name	Code	Data type	Primary
自增 ID	ID	int	true
文件名	FILENAME	char(20)	false
二进制素材内容	CONTENT	image	false
当前版本	CVERSION	char(20)	false
保存路径	PATH	char(100)	false

(5) 更新记录表

数据库更新程序的流程是先由客户机发送更新请求，传给服务器本机的 IP 地址，服务器到“客户机 IP 列表”中校验这个 IP 是否可以更新，如果表中无此 IP，则不能执行操作，如果 IP 地址找到，根据这个 IP 地址对照更新记录表中的每一个文件的客户机版本号是否与素材表中的当前版本相符，如不符，则执行更新操作。

表 3.5 更新记录表

Tab. 3.5 UpdateRecord

Name	Code	Data type	Primary
自增 ID	ID	int	true
客户机 IP	IP	char(15)	false
文件名	FILENAME	char(20)	false
客户机版本号	CVERSION	char(20)	false

(6) 客户机 IP 列表

客户端更新时起校验作用。

表 3.6 客户机 IP 列表

Tab. 3.6 Computer_IP

Name	Code	Data type	Primary
客户机 IP	IP	char(15)	true
备注	COMMON	varchar(100)	false

(7) 进度保存表

退出游戏前系统自动根据当前用户 ID 临时保存游戏进度，并询问是否保存游戏进度，如选择保存，则将临时保存的游戏进度正式保存，如不保存，则将临时保存的进度删除。

表 3.7 进度保存表

Tab. 3.7 Savedate

Name	Code	Data type	Primary
自增 ID	ID	int	true
进度记录	DATA	image	false
创建日期	CREATE_DATE	datetime	false
是否为临时记录	RECORD	int	false

(8) 交易记录表

记录用户每次的交易信息

表 3.8 交易记录表

Tab. 3.8 DealRecord

Name	Code	Data type	Primary
自增 ID	ID	int	true
道具	TOOLS	image	false
交易日期	DEAL_DATA	datetime	false
交易金额	DEAL_SUM	int	false

4 3D 高尔夫游戏引擎设计与实现

4.1 手机 3D 游戏引擎结构分析

游戏的引擎(Game Engine)就好比汽车的发动机,发动机是汽车的心脏,决定着汽车的性能和稳定性,汽车的速度、操纵感这些直接与车手相关的指标都是建立在发动机的基础上的,游戏引擎也是如此^[16]。玩家所体验到的剧情、关卡、美工、音乐、操作等内容都是由游戏的引擎直接控制的,它扮演着中场发动机的角色,把游戏中的所有元素捆绑在一起,在后台指挥它们同时、有序地工作。不混淆游戏引擎和整个游戏,其实就像把一个汽车的发动机和整个汽车分清楚一样简单。我们能够从汽车里面取出发动机,利用另外一个外壳,构建成另外一部汽车,游戏也是这样。游戏引擎其实也可被定义为所有的非游戏特有的技术。简单地说,引擎就是用于控制所有游戏功能的主程序,从计算碰撞、物理系统和物体的相对位置,到接受玩家的输入,以及按照正确的音量输出声音等等^[17]。

无论是 2D 游戏还是 3D 游戏,无论是角色扮演游戏、即时策略游戏、冒险解谜游戏或是动作射击游戏,哪怕是一个只有 1M 大小的小游戏,都有这样一段起控制作用的代码,引擎。经过不断的进化,如今的游戏引擎已经发展为一套由多个子系统共同构成的复杂系统,从建模、动画到光影、粒子特效,从物理系统、碰撞检测到文件管理、网络特性,还有专业的编辑工具和插件,几乎涵盖了开发过程中的所有重要环节。

从全局看游戏引擎具有如下所示的体系结构:

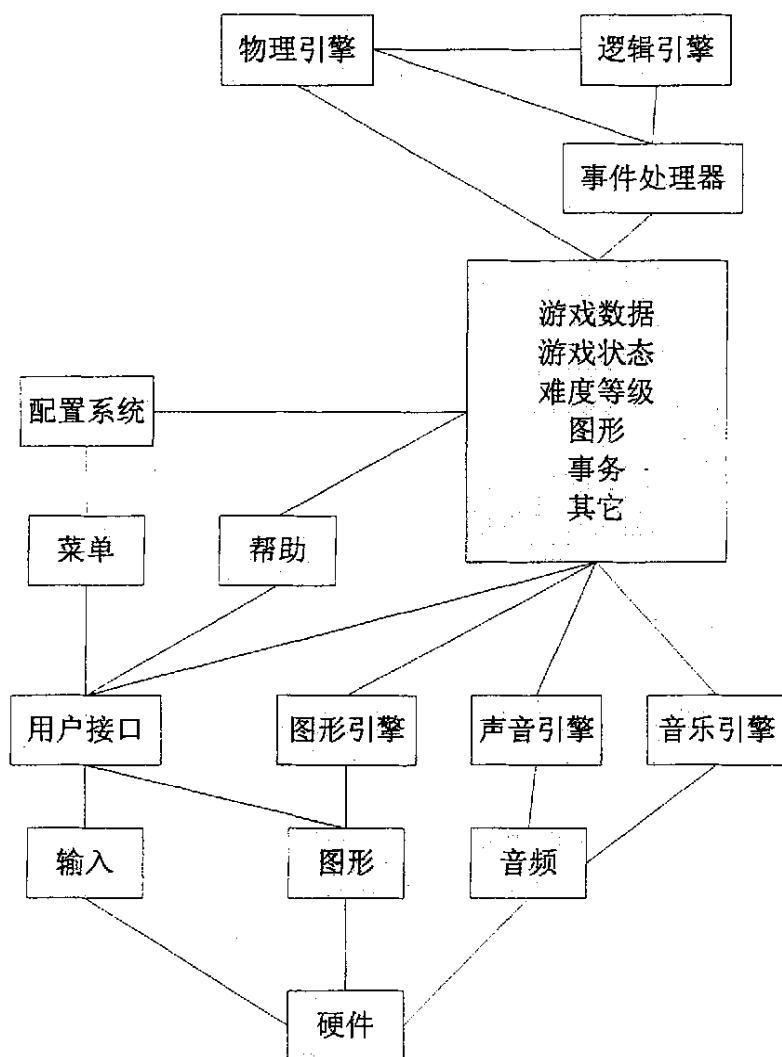


图 4.1 引擎结构

Fig. 4.1 Structure Of The Engine

从图中可以看到引擎主要包含用户接口、图形引擎、声音音乐引擎、配置系统、数据管理、事件处理器、物理引擎、逻辑引擎这几部分

(1) 和其他层相比，用户接口是相当简单的。它只是监视用户各种输入状态的变化，并把它们存储在游戏数据当中，为玩家显示出菜单以及相应的在线帮助(在线帮助目前倒可能做的相当复杂)^[18]。用户接口中有一个用于处理输入的系统，专门负责处理所有的输入(键盘、鼠标、游戏手柄和游戏摇杆)，并把它们统一起来以允许控制的抽象化。举例来说，当游戏逻辑进行到某处的时候，用户接口需要检测玩家是否希望他们的位置朝前移动。与其直接依次去检测每种输入设备，不如调用一次这个系统，去透明地完成

检测所有的输入设备的工作。这样还有一个好处,就是使得玩家可以轻松地进行输入配置,并且可以很容易地实现由多种输入控制同一动作的功能。

由于用户接口在开发时,很大程度上是类似甚至雷同的,而工作量极大。因此,为了节省开发时间和精力,便于移植,用户接口应该是可重用的。

(2) 图形引擎的概念非常直观易懂。游戏通过图形引擎来把图像在屏幕上显示出来。通常,图形引擎会用 3DMAX、MAYA、M3G 或者其他软件渲染来实现这个部分。如果想更酷的话,可以用 API 实现各种图形引擎,然后在它们上面设计一个图形抽象层,这样就给了用户更多的选择和机会来提高兼容性和性能。当然,实现这个目标并不是一件很容易的事情,因为不是所有的 API 都具有相同的特性集。

首先是光照和阴影效果,即场景中的光源对处于其中的人和物的影响方式。游戏的光照和阴影效果完全是由图形引擎控制的,折射、反射等基本的光学原理以及动态光源、彩色光源等高级效果也都是通过图形引擎的不同编程技术实现的^[19]。

其次是动画系统。目前游戏所采用的动画系统可以分为两种:一是骨骼动画系统,一是模型动画系统^[20]。前者用内置的骨骼带动物体产生运动,比较常见;后者则是在模型的基础上直接进行变形。图形引擎把这两种动画系统预先植入游戏,方便动画师为角色设计丰富的动作造型。角色的各种动作都不会对背景造成任何破坏。

渲染是图形引擎最重要的功能之一。当 3D 模型制作完毕之后,美工会按照不同的面贴上纹理(材质),这相当于为骨骼蒙上皮肤;最后再通过渲染,把模型、动画、光照、阴影、特效等所有效果实时计算出来并展示在屏幕上。渲染在图形引擎的所有部件当中是最复杂的,它的强大与否直接决定着场景和实体最终的输出质量^[20]。一些比较复杂的显示因素包括:全局的小地图、多个视图、画面重叠和特殊效果等。针对不同的游戏,图形引擎经常需要重做或者经过重大修改。这往往消耗了太多的时间和财力。

(3) 声音、音乐引擎负责加载和播放游戏当中的各种声音和音乐。3D 音效是手机游戏可玩性的一个重要因素,这使得整个工作变得稍微复杂了一些。声音和音乐引擎主要有三个功能:改进玩家的体验和感受、体现周围的环境和气氛以及对将要发生的事情提供某些线索。

其中各种声音和音乐主要有以下几种类型:Wave(质量高、占用内存多但速度快)、MP3(质量高、经过压缩而速度较慢)、Midi(质量差、只占用很少的内存、用途有限但可移植)、CD(质量很高、快速、占用大量内存、局限于背景音乐)等。相关素材可以通过以下几种方式得到:从网络上下载、对样本声音和音乐进行修改,以满足自己的需求或者是创建自己的声音和音乐。

(4) 配置系统负责读取配置文件、命令行参数、或者其他被用到的设置方式^[21]。其他子系统在初始化和运行的时候会向它查询相应的配置。这样就可以很容易地改变分辨率、颜色位深、快捷键设置、声音支持选项等等,甚至是在游戏已经加载了以后。游戏引擎应该尽可能是可配置的,这样可以简化测试并使得玩家更易于把游戏设置成他们自己所喜爱的方式。通过控制合变量和函数,玩家可以在不重新启动的情况下改变游戏和引擎的设置。对于开发人员来说,能在开发时输出调试信息也是一件很好的事情。这样在很多时候,开发人员如果需要检查一些变量,把它们值输出到控制台上也比使用调试器方便得多。一旦引擎开始运行,如果有错误发生,不必退出这个应用程序,开发人员可以很轻松地处理它并输出一条错误信息。如果开发人员不想让最终用户看到或使用控制台,可以很容易地将它屏蔽掉,没人能看得出它的存在。配置系统必须能够根据玩家的选择,适应各种不同规格的硬件;同时,还应该保存玩家的设置,满足他们不同的喜好和偏爱。

在线帮助和菜单系统共同组成了用户接口,集成在游戏当中(玩家是从不阅读帮助手册的)。在线帮助一般在屏幕上显示一些简单的文本,在内容上是静态的。针对不同的上下文环境和玩家,显示不同的内容。在线帮助还可以是角色形式的助手,更为生动活泼。随着游戏规模的不断膨胀,在线帮助也趋向于变得越来越复杂,当然也就要花费开发人员更多的时间和精力。

(5) 游戏是通过数据文件或者数据库来定义的^[22]。实际上,游戏在很大程度上是严格受数据库控制的,有人甚至认为,“游戏就是一个有着有趣的接口的数据库”。数据文件包括资源定义文件、游戏定义文件和界面定义文件。数据文件可以采用文本格式,这样带来两个好处:一是容易处理版本的升级和更新;二是可以减少前期对编辑工具的功能要求,许多数据可以手工编写而不需要专门的编辑工具。这样,开发人员可以用文字编辑器处理大部分数据,然后有时间时再设计一个强大的工具。当然,最终提供给用户的是处理后的数据文件。这中间需要有一个转换模块。游戏部件通过游戏数据和支持部件进行通讯。图像库、视频库、声音库和音乐库这些数据,可以通过编辑工具形成的资源文件来定义。每种资源包括定义的管理和一些操作接口。图像库中的图像包括多种格式(BMP、GIF等),最好允许图像组合定义。每一幅图像可能包括许多小图,需要定义它的小图的位置,当然也可以自动生成。游戏需要的特殊定义就是通过于图来定义的,比如行走、身体性质、中心定位点、触发区等,还可以根据自己的要求扩充各种性质的定义。视频库中的视频包括 AVI 及 FLC 等格式。声音库中包含 WAV、MP3 和 MID 等资源。调试版本中最好不要将资源打包,而是指向正常的文件名;发行版本中再将资源打包。这样修改起来容易一些,否则最好有一个自己的资源管理器。

除此之外,资源管理的对象还包括内存管理,比如设置时间值以释放长期不用的内存资源。游戏数据还包括难度等级的描述、游戏状态、事件队列、用户喜好等。

(6) 绝大多数游戏采用了基于事件的模型。事件可能由玩家的输入信息、实体之间的碰撞、定时器(由逻辑引擎进行控制)以及游戏令牌(即实体)产生。令牌有不同的状态,能够对事件做出反应并产生新的事件。游戏引擎使用状态图(有限状态机)和交互矩阵来描述实体的行为。

基于事件的模型是这样的:一个事件产生:玩家或者实体对该事件做出反应,逻辑引擎和物理引擎对游戏状态进行相应的改变;在屏幕上显示新的游戏状态,等待新的事件发生^[23]。如此循环,不断反复。在游戏时间的推进机制上,有四种不同的方式:①采用双缓冲,和现在 50fps(帧每秒)或者更低的帧速率相比,老游戏的帧速率可达到 100fps,运行速度太快。另外,可能导致在速度上的急剧下降;②采用三缓冲,当事件处理的时间太长时,会导致速度下降。事件处理花费的时间越长,会导致延时越大,从而可能造成恶性循环。如果能够使得事件处理的时间依赖于消逝的时间,则可以避免这一点;③采用固定的频率来进行完整的事件处理循环,而在进行事件处理循环的空闲进行绘制;④系统每次采取时间很短的步骤,只处理少量的事件;而在这个步骤的外面进行绘制。这种方式必须保证游戏状态一直是有效的,并且实现起来比较困难。

(7) 逻辑引擎负责为所有的实体(即游戏中的运动对象)赋予一定的人工智能(AI),并处理所有的游戏模拟,避免出现自相矛盾的情形^[22]。加强游戏的人工智能可以提高游戏的可玩性。物理引擎可以使物体的运动遵循固定的规律,力求比较逼真地模拟现实世界,给玩家以比较真实的体验。例如,当角色跳起的时候,系统内定的重力值将决定他能够跳多高,以及他下落的速度有多快。另外,球的飞行轨迹、在草地上的弹跳方式都是由物理引擎决定的。碰撞检测是物理引擎的核心部分,它可以检测游戏各物体的物理边缘。当两个 3D 物体撞在一起的时候,这种技术可以防止它们互相穿过,这就确保了当实体撞在墙上的时候不会穿墙而过,也不会把墙撞倒,因为碰撞检测会根据实体和墙之间的特性确定两者之间的位置和互相的作用关系。物理引擎还应该确保实体的运动反映出不同地形的影响。这主要包括两个方面:对于陆地上地形的起伏和海上波浪的翻滚,实体的运动速度和姿态做出相应的变化。

对于 J2ME 的游戏引擎来说,可谓是麻雀虽小五脏俱全,虽然 J2ME 游戏可能连一幅图片的大小都不到,但不能因此而放松对引擎的要求,一方面是理论的需要,另一方面引擎一旦开发成功可以大大减少程序与美工的工作量和提高产品质量和数量。游戏引擎是用于控制所有游戏功能的主程序,所以开发优秀的游戏最重要的部分是开发出优秀的游戏引擎。那么,什么样的手机游戏引擎才一是优秀的呢?引擎应该尽量简单。引

擎应该只包含游戏最重要部分，也就是游戏必须的部分。这样在开发引擎的时候才能把精力集中在游戏的核心部分，好的引擎是游戏成功的前提。对于手机游戏而言，尽量少地占用内存和 CPU 是基本准则。这在游戏引擎开发的时候就要尽量注意。

4.2 手机 3D 游戏引擎设计实现

基于 3D-GOLF 游戏的流程引擎的内容主要设计为五个方面：游戏画面(Screen)、渲染(Renderer)、流程控制(Navigator)、碰撞检测(Collidect)和摄象机(Camera)。这五个模块封装为一个 Engine 包，在应用程序中直接导入包，调用类名.方法即可。其中，对于手机游戏来说流程控制与碰撞检测是最重要的两个类，此次在项目中的主要工作也是完成这两个类的设计与实现。引擎的设计重在其可复用性以及交互方面的功能，出于篇幅及技术保密等方面的考虑，本文仅涉及重点设计部分。

4.2.1 游戏画面

系统中的游戏画面主要由启动画面、主画面、游戏画面、设定画面、帮助画面这五部分组成。

(1) 启动画面

这个画面只在开始加载时用到。在屏幕上显示一段游戏主题动画。这个类的定义是：

```
public class Splash extends Alert {  
    public Image Imgwelcome;  
    public Splash(String s);  
}
```

由于这个类只在系统一开始用到而已，因此用 Alert 实现最合适，Alert 是一个比较特殊的屏幕型对象，它会先发出一段声音，然后将自己显示在屏幕上，过一段时间之后，它会自动跳回之前的画面，由于它这种与众不同的特性，所以我们用它来做应用程序的启动画面。

Imgwelcome：是将图片加载到应用程序中

(2) 帮助画面

帮助画面的功能是说明游戏的规则、评分标准、以及游戏的操作规范，都是文字说明的性质，类的定义是：

```
class HelpScreen extends Form {  
    public HelpScreen(String c);  
}
```

HelpScreen 类只实现文字说明的功能，因此用 Form 来实现，HelpScreen 会根据构造函数传入的具体值而显示具体内容。

(3) 主画面

即为游戏开始的主菜单。类的定义：

```
class MainScreen{
    private static Displayable instance;
    synchronized public static Displayable getInstance();
    private MainScreen(){
        append();
    }
}
```

getInstance(): 主画面在系统中最好只有一个,除了可以保持本身状态,还可以降低内存使用,因此这里采用了 Java 中的 Singleton 设计模式,Singleton 模式主要作用是保证在 Java 应用程序中,一个类 Class 在任何给定的时间只有一个实例存在。

append(): 将选项加入 MainScreen 之中

(4) 设定画面

由用户来选择游戏的相关信息,例如背景音乐、难度等级、道具配置等。类的定义是:

```
class SetScreen{
    private static Displayable instance;
    synchronized public static Displayable getInstance();
    private SetScreen();
}
```

同样,设定画面在系统中也是只有一个,方便保持自身状态以及降低内存的使用,因此也采用 Singleton 设计模式来实现。

(5) 游戏画面

类的定义是:

```
class GameScreen{
    private GameScreen();
    public void commandAction(Command c,Displayable s);
    private void worldInit();
    private void draw_3D(Graphics g);
    private void setLights();
    private void draw_Feature();
}
```

游戏场景是在 3D 建模器中做好之后导入近来的,导入之后再赋予它一定的特效。

`worldInit()`: 导入 3D 场景并进行调整。

`draw_3D(Graphics g)`: 在场景中加入 3D 的实体, 并进行渲染。

`setLights()`: 给实体加入光线的效果。

`draw_Feature()`: 在游戏中设定的几种特效。

4.2.2 渲染 (Renderer 类)

受现有手机硬件条件限制, 手机上的 3D 游戏还不能向 PC 机上的 3D 游戏那样把模型、动画、光照、阴影、特效等所有效果实时计算出来并展示在屏幕上, 同时受屏幕大小的限制一些特效渲染前后的区别不大, 因此这一部分仅实现了几个简单的功能。

```
abstract class Renderer{
    private int[] transformation;
    public abstract void drawLine(int x1,int y1,int x2,int y2);
    public abstract void clearArea(int x,int y,int w,int h);
    public abstract void setDrawRegion(int x,int y,int w,int h);
    public void drawRectangle(int x,int y,int w,int h);
    public void drawScene(ViewPort view);
}
```

(1) `drawLine`: 画一条线, `x1`、`y1` 为起点坐标, `x2`、`y2` 为终点坐标。

(2) `clearArea`: 清出一块屏幕, `x`、`y` 为要清除区域左顶点坐标, `w`、`h` 为要清除区域的宽度和高度。

(3) `setDrawRegion`: 在屏幕上定义一块画的区域, `x`、`y` 为要画区域左顶点坐标, `w`、`h` 为要画区域的宽度和高度。

(4) `drawRectangle`: 画一个长方形, `x`、`y` 为要画区域左顶点坐标, `w`、`h` 为要画区域的宽度和高度。

(5) `drawScene`: 从一个顶点给物体加入光线。

4.2.3 流程控制 (Navigator 类)

在引擎构建的初期阶段, 首先需要对整个系统进行分析与设计。我们根据游戏的相关信息定义了系统流程, 系统流程决定了程序中画面的切换。3D 高尔夫游戏所设计的系统流程如图 4.1 所示。

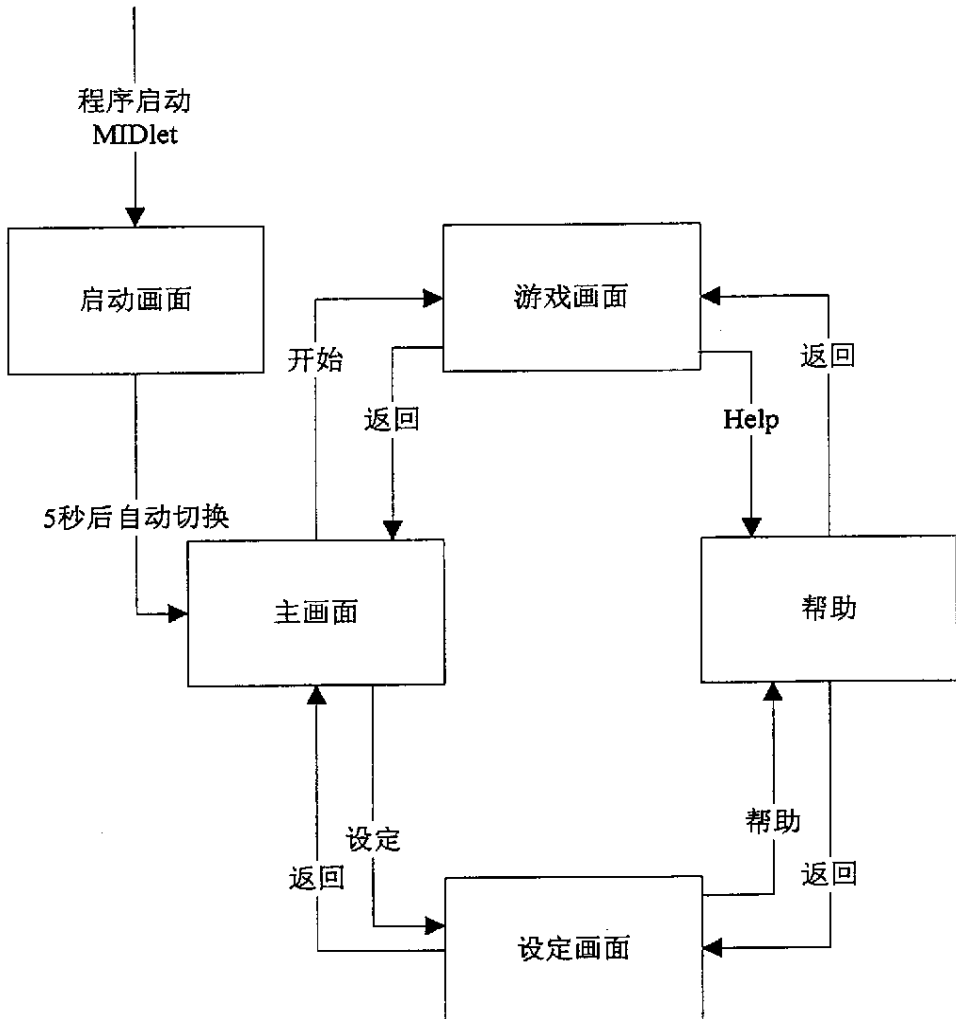


图 4.1 系统流程

Fig. 4.1 Systematic procedure

以上这些画面之间的切换由流程控制器来完成整个系统的流程控制。流程控制器以 Navigator 类来实现。类的定义是：

```

class Navigator{
    public static int current;
    public static void show();
    public static void flow(String cmd);
}
  
```

- (1) current: 使用 current 变量来记录各切换画面。
- (2) show(): 此方法用来统一显示出画面。

(3) flow():用来统一判断程序的流程。利用 current 变量与 cmd 变量来判断目前程序的运作方向。flow()方法内部会使用 show()方法来转换画面。这里不处理启动画面,因为它都使用 Alert 实现,Alert 具有时间一到就自动跳回前一个画面的特性,因此不另外对其做流程控制。

4.2.4 碰撞检测 (Collidect 类)

碰撞检测是物理引擎的核心部分,要整体实现物理引擎是十分复杂的,它的执行会大大加重游戏的负载,因此,对于手机游戏来说本研究只对核心的碰撞检测部分加以实现

类的定义是:

```
class Collidect{
    static boolean Hit_squareness(int obj_1_xpos,int obj_1_ypos,
    Image obj_1_img,int obj_2_xpos,int obj_2_ypos, Image obj_2_img)
    static boolean Hit_Sphere(int obj_1_xpos,int obj_1_ypos,
    Image obj_1_img,int obj_2_xpos,int obj_2_ypos,Image obj_2_img)
}
```

Hit_squareness()方形检测和 Hit_Sphere()圆形检测,通过给定物体的坐标来与图片高度及宽度做比较判断是否碰撞。

4.2.5 摄像机 (Camera 类)

在游戏中 Camera 的作用就相当于我们的眼睛,如何真实的感受到 3D 游戏的身临其境 Camera 的设计起了决定性的作用。

类的定义是:

```
class Camera{
    public Camera(World w, int fov);
    public World getWorld();
    public boolean inView();
    public void captureScene();
    public int getModelDistance(Model m);
    public static void Lookat();
    public static void seeat();
    public static int get_xScreen();
    public static boolean move_Camera();
    public static void camRotateMesh();
    public static void AutoLookAtHowe();
```

```
}
```

(1) Camera(): Camera 的构造函数。

(2) getWorld(): 返回一个通过摄像机所看到的世界

(3) inView(): 确定一个物体在摄像机的视野里是否是可见的。

(4) captureScene(): 返回可见世界的坐标改变。

(5) getModelDistance(): 返回摄像机和物体之间的近似距离。

(6) Lookat(): 这个函数功能是让摄像机定着一个物体看, 摄像机以自己为中心转动。

(7) seeat(): 与 lookat 函数类似, 没有竖直方向的转动, 只保留了水平方向。

(8) get_xScreen(): 函数功能为从 camera 向目标位置发出一条射线, 求出与屏幕相交点的二维坐标的 x 坐标。

(9) move_Camera(): 摄像机位置改变之后, 如果摄像机位置在地下, 将摄像机位置移到相对于物体的另一面

(10) camRotateMesh(): 这个是让摄像机盯着一个物体看。摄像机以目标物体为中心转动。

(11) AutoLookAtHowe(): 函数的功能为自动转向, 让摄像机对着球洞, 而且摄像机还能看着物体。

结 论

手机已经与现代生活方式紧紧地结合在一起，手机游戏将是手机多媒体时代最受欢迎、最有价值的应用。在游戏里，剧情的进行、形形色色的角色衬托、各种场景的变换，都是由游戏引擎事先就约定俗成一种模式、一个大体的框架，然后依靠一些复杂的数据库来组织完成的。

本文的主要工作是采用了面向移动的 J2ME 技术，针对手机游戏引擎小型化、多模块化的特点，在游戏开发的基础上架构了一个基于 J2ME 的小型 3D 游戏引擎。在游戏的流程控制、摄像机处理以及渲染等方面都做了一定的实现，对新游戏的开发具有不同程度的可复用性。

游戏引擎架构是一项极其复杂，而且工作量极大的工作，手机游戏引擎受其硬件条件限制，在开发中所涉及的实际问题和技术细节也很多，许多技术都是在开发过程中逐步去学习的。要真正解决手机游戏引擎中的所有问题，达到真正意义上的游戏引擎，还需要进一步的研究和探索。

参 考 文 献

- [1] 秦立超. 国内手机无线游戏市场前景分析. 信息技术. 2002, 10:23-27.
- [2] 潘功峰. Java 技术在手机领域中的应用. 计算机时代. 2003, 7:42-46.
- [3] 成良玉. 手机游戏将成为市场新宠. 计算机时代. 2004, 11:37-39.
- [4] Michael Juntao Yuan. J2ME 应用程序开发. 北京: 清华大学出版社, 2004.
- [5] 王森. Java 手 A/PDA 程序设计入门. 北京: 电子工业出版社, 2003.
- [6] Jamea Keogh. J2ME: The Complete Reference. McGraw-Hill, 2003.
- [7] 许向峰. 基于 J2ME 的 Java 手机软件开发. 微型机与应用. 2002, 1:38-42.
- [8] 王森. JAVA 手机程序设计. 北京: 中国铁道出版社, 2003.
- [9] 李朔峰. 基于 J2ME 的手机游戏的开发. 电脑知识与技术. 2005, 6:26-31.
- [10] 李振军. 基于 MIDP 的手机游戏开发方法与实现. 计算机应用. 2004, 1:46-49.
- [11] 黄心渊. 体育类手机游戏总动员. 电脑技术. 2004, 10:19-21.
- [12] Dante Treglia. 张磊译. 游戏编程精粹 3. 北京: 人民邮电出版社, 2003.
- [13] 荣钦. 游戏设计概论. 北京: 科海电子出版社, 2003.
- [14] 张小玮. J2ME 无线应用平台开发. 清华大学出版社, 2004.
- [15] 陈天洲. 深入 BREW 手机游戏开发. 清华大学出版社, 2004.
- [16] David Bourg. Physics for Game Developers. USA: O'Reilly, 2001.
- [17] John Chismar. 3D Studio MAX3 Media Animation. 电子工业出版社, 2000.
- [18] 夏旺盛. 3D 引擎构架概述. 现代计算机. 2003, 6:42-48.
- [19] Richard Rouse. Game Design: Theory and Practice. Canada: Wardware Publishing Inc, 2001.
- [20] David H. Eberly. 3D Game Engine Design. New York: Morgan Kaufman Publishers, 2001.
- [21] 张卫山. 基于 J2ME 的手机网络 RPG 游戏引擎. 计算机工程与应用. 2005, 6:63-70.
- [22] Eric. Mathematics for 3D Game Programming. Computer graphics forum. 2003, 4:27-34.
- [23] 张淮声. 基于 3D 结构和图像的复杂场景建模绘制. 计算机工程与应用. 2003, 10:60-67.

附录 A MIDlet 主程序

启动画面我们是使用 Alert 来实现的，在流程的转换上由于其本身的特性，所以我们采用“先显示主画面，再显示启动画面”的设计方式。

```
Public class FlowControl extends MIDlet
{
    boolean init=true;
    public FlowControl()
    //初始化 display 变量和 midlet 变量
    {
        Navigator.display=Display.getDisplay(this)
        Navigator.midlet=this;
    }
    public void startapp()
    { //初始化 current 变量
        Navigator.current=Navigator.MAIN_SCREEN;
        Navigator.show(null);

        if(init)
        {
            Alert splash=new Alert("片头动画");
            splash.setType(AlertType.CONFIRMATION);
            splash.setTimeout(5000);
            Navigator.display.setCurrent(splash);
            init=false;
        }
    }
    public void pauseApp()
    {
    }
    public void destroyApp(Boolean con)
    {
    }
}
```

由于 `startapp()` 不会只执行一次，它会在 MIDlet 从停止状态转换到运作状态时被调用，因此我们利用 `init` 变量来判断 `startapp()` 是否是第一次执行，只有第一次执行才会显示启动画面。

致 谢

在大连理工大学软件学院学习的两年半时间内，学院的各位老师不论是在学习上还是生活上都给予了我极大的关怀和帮助。我的导师李明楚教授更是以其严谨的治学态度和为人之道时刻影响着我的求学之路，本文是在他的悉心之道下完成的，从课题的选定到完成，他都给予了我详细的指导。在此，我对李老师表示最尊敬的致意，并真诚的祝愿他身体健康、工作顺利、家庭幸福。

感谢泰康科技有限公司的戴维博士，给我提供了大量的实践和锻炼机会，使我在实际动手能力和社会阅历方面都有了很大的提高。在这里我真诚的祝愿他工作顺利、万事如意。

感谢公司的同事陈国梁、李平，他们在学习和研究上给了我很大的帮助！

感谢我的父母和亲人，虽然他们不在我身边，但是他们的关心、爱护、鼓励和支持永远陪伴我，使我得以顺利地今天完成学业。在此，我要深深地祝福他们，祝他们身体健康，生活快乐！

感谢所有关心我和帮助我的朋友们！

作者: 赵晖
学位授予单位: 大连理工大学

参考文献(23条)

1. 秦立超 国内手机无线游戏市场前景分析 2002(10)
2. 潘功锋 Java技术在手机领域中的应用[期刊论文]-计算机时代 2002(7)
3. 成良玉 手机游戏将成为市场新宠 2004(11)
4. Michael Juntao Yuan J2ME应用程序开发 2004
5. 王森 Java手A/PDA程序设计入门 2003
6. Jamea Keogh J2ME:The Complete Reference 2003
7. 许向锋 基于J2ME的Java手机软件开发[期刊论文]-微型机与应用 2002(1)
8. 王森 JAVA手机程序设计 2003
9. 秦玉燕 基于J2ME的手机游戏的开发[期刊论文]-电脑知识与技术(技术论坛) 2005(9)
10. 李振军 基于MIDP的手机游戏开发方法与实现 2004(01)
11. 凌夫 体育类手机游戏总动员[期刊论文]-电脑技术-Hello-IT 2004(10)
12. Dante Treglia. 张磊 游戏编程精粹3 2003
13. 荣钦 游戏设计概论 2003
14. 张小玮 J2ME无线应用平台开发 2004
15. 陈天洲 深入BREW手机游戏开发 2004
16. David Bourg Physics for Game Developers 2001
17. John Chismar 3D Studio MAX3 Media Animation 2000
18. 夏旺盛 3D引擎构架概述 2003(06)
19. Richard Rouse Game Design:Theory and Practice 2001
20. David H Eberly 3D Game Engine Design 2001
21. 张卫山. 刘伟 基于J2ME的手机网络RPG游戏引擎[期刊论文]-计算机工程与应用 2005(33)
22. Eric Mathematics for 3D Game Programming 2003
23. 高月芳. 张淮声. 张佑生 基于3D结构和图象的复杂场景建模绘制[期刊论文]-计算机工程与应用 2003(10)

相似文献(3条)

1. 学位论文 林琳 JM3D-Core手机游戏引擎设计与渲染的实现 2006

游戏引擎是一个处理游戏底层技术的平台,使用游戏引擎,游戏开发人员可以不用花过多精力去处理系统架构、内存管理、图像绘制等一些底层技术,可以直接使用引擎提供的接口进行游戏开发,从而大大缩短开发时间。随着手机性能的提升和3D图形技术在手机游戏领域的应用,3D手机游戏必将成为游戏领域的一个亮点,因此开发一款适应性强的3D手机游戏引擎,能够使今后3D手机游戏的开发更加高效和方便,是未来手机游戏开发的重点。

本文通过对手机游戏引擎的发展现状及JM3D-Core手机游戏引擎开发的意义、优势等介绍,分析了该引擎的开发在3D手机游戏发展中必要性,并在此基础上设计了基于J2ME平台的3D手机游戏引擎模块。主要对物理系统中的碰撞检测部分,图形引擎中的渲染部分以及人工智能模块、网络通信模块作了详细设计。碰撞检测部分依据能在2D范围内解决的碰撞就不要使用3D方式检测的设计原则,采用了区域检测和向量检测交互进行的方式,大大提高了检测速度和检测精度,能够提高3D手机游戏的逼真性。人工智能的设计上主要是针对非玩家角色进行了简单、复杂两种行为设计。网络通信模块采用Socket技术对客户端进行了详细设计,至于服务器端由于J2ME技术本身的限制,预计采用其他技术在计算机上实现。图形引擎模块中渲染部分的设计实现是本文的重点,从原理入手,通过对光线投射和光线跟踪算法的比较分析,设计了运用光线投射技术来开发适用于J2ME手机平台上的第一人称视角的3D游戏渲染引擎,以提高渲染速度。

在该项目中,通过使用光线投射技术实现了渲染引擎的设计开发,并且可以在普通性能的手机上实现基本的场景渲染,其渲染速度和消耗的内存都可以控制在技术目标所设定的范围中,但是渲染的速度、性能仍需要在今后的开发中进一步提升和完善,使渲染的效果达到更高的层次。

2. 学位论文 蒋剑忠 基于J2ME的手机游戏引擎库设计与实现 2009

游戏引擎就是用于控制所有游戏功能的主循环,从计算物体碰撞、物理系统和物体的相对位置,到接受玩家的输入,以及按照正确的音量输出声音、提供各种工具和插件等等。每一款游戏都有自己的引擎,真正能获得他人认可并成为标准的引擎并不多,所以游戏引擎的研究成为游戏开发领域发展的一个重要方向。

随着电信各运营商3G业务的形成,手机游戏的巨大商机开始展现在人们面前。传统游戏的开发公司已从卡带游戏、PC游戏向手机游戏扩张,并与移动通信服务提供商进行更加紧密地合作,手机游戏已成为目前最具活力的增值业务。本文通过分析、设计、实现RPG(角色扮演)手机游戏引擎库,并通过应用引擎库开发三款RPG手机游戏。本论文阐述了基于J2ME平台上分析与研究手机游戏引擎库的设计与实现。论文的主要工作有:

1. 分析手机游戏的基本概况和国内研究状况,并阐述了J2ME开发平台及相关技术研究。

2. 从设计目标、数据分离法到背景引擎渲染详细分析RPG手机游戏引擎库系统架构。
3. 应用RPG手机游戏引擎库分析结果实现Gamepack包程序控制与ALLDATA数据部分，完成引擎库的设计与实现。
4. 应用上述引擎库开发三款RPG手机游戏，验证其可复用性、独立性、移植性。

3. 学位论文 郭庆唐 [基于短距离无线技术的手机战棋策略网络游戏引擎的设计与实现](#) 2007

随着生活节奏的加快，人们不再经常有整片的时间来玩电脑游戏，转而利用零碎时间玩手机游戏，因此手机游戏越来越受到大家的青睐。而游戏引擎作为对底层功能的封装，可以大大缩短同类游戏的开发周期。故本文进行了“基于短距离无线技术的手机战棋策略网络游戏引擎的设计与实现”的研究。

首先，本文认真分析和总结了当前存在的手机网络游戏引擎的主要特点和通用架构，并详细介绍和比较当前主流的几种短距离无线通信技术的各自特点，为后文设计出一种基于短距离无线通信技术的适用于手机、PDA等移动平台的战棋策略游戏引擎做理论准备，这一部分内容是论文的第一章。然后，本文分手机战棋策略游戏引擎系统架构和网络模块两部分对该游戏引擎的设计过程进行了具体的阐述和分析。这一部分是本文的核心，设计方法紧密结合本人在专业的手机游戏公司的实习经验。最后，本文详细阐述了如何利用该引擎快速地开发一个无线局域网模式下的手机战棋策略对战游戏，并将该游戏引擎运用在一个具体的游戏——《火焰之纹章二》，从模拟器上的运行效果可以看出：该引擎初步实现了无线局域网内的多人联网游戏。

在无线局域网中，玩家可以与自己熟悉的伙伴进行合作或对战游戏，而且无需支付昂贵的网络费用，所以利用短距离无线传输技术构建无线局域网游戏将会是手机网络游戏发展的一个方向。本文设计的游戏引擎能给这一类开发带来一定的意义和价值。

本文链接：http://d.g.wanfangdata.com.cn/Thesis_Y824917.aspx

授权使用：郑州大学(zzdx)，授权号：6fa41f4a-af52-48c6-b9d8-9e2200c9fe7e

下载时间：2010年11月2日