

数 码 艺 术 设 计 系 列

# J2ME

## 手机游戏设计案例教程

张晓蕾 主编

杨 旭 刘宝芹 罗如燕 副主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

数码艺术设计系列

# J2ME 手机游戏设计案例教程

张晓蕾 主编

杨 旭 刘宝芹 罗如燕 副主编

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

本书通过 16 个案例和一个较完整的商业游戏项目的设计,较全面地介绍了使用 J2ME 进行手机游戏设计的方法和技巧。在按案例进行讲解时,充分注意知识的相对完整性和系统性。本书起点低、跨度大、循序渐进、通俗易懂,使读者在阅读后,能够较快入门,并且在技术上得到较大的提高。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

## 图书在版编目(CIP)数据

J2ME 手机游戏设计案例教程 / 张晓蕾主编. —北京: 电子工业出版社, 2009. 12

(数码艺术设计系列)

ISBN 978-7-121-09911-3

I. J… II. 张… III. ①JAVA 语言—程序设计—教材 ②游戏—应用程序—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2009) 第 211175 号

责任编辑: 吕 迈

印 刷: 北京市顺义兴华印刷厂

装 订: 三河市双峰印刷装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 22.75 字数: 582 千字

印 次: 2009 年 12 月第 1 次印刷

印 数: 4 000 册 定价: 35.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zlts@phei.com.cn](mailto:zlts@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

# 前 言

随着游戏开发技术的发展,手机游戏成为游戏中的新秀。手机游戏开发相对于传统游戏开发,具有低预算、低成本、开发团队小、易于管理,以及不需要图形图像设备这类特殊的设备的优点,并且基于开放的标准,以及手机本身的特点,为支持多人游戏铺平了道路。

Sun 公司的 J2ME 技术对游戏的开发提供了强大的支持,使得开发者只需要相对较少的工作就可以完成完整的游戏程序设计。J2ME 是 Sun 公司针对嵌入式、消费类电子产品推出的开发平台,它与 J2SE 和 J2EE 共同组成 Java 技术的三个重要的分支。

本书共 5 章:

第 1 章介绍手机游戏设计的基本概念、J2ME 平台的体系结构、MIDlet 生命周期的概念和游戏策划设计,为以后的学习打下良好的基础。

第 2 章介绍手机游戏开发环境的配置,包括在 Eclipse 环境和 NetBeans 环境下进行手机游戏开发的开发环境配置和开发方法。

第 3 章介绍 MIDP 游戏开发的基础知识,包括高级用户界面、低级用户界面和图形图像处理等。

第 4 章介绍 MIDP 2.0 中新增加的游戏 API、音效设计和记录管理等内容,使得手机游戏的开发更加方便。

第 5 章通过一个较完整的商业手机项目的实现过程,全程讲解了手机游戏的设计方法。

本书采用任务驱动的案例教学方式,通过对手机游戏案例的开发,来介绍如何使用 J2ME 设计手机游戏程序,具有通俗易懂、实用和技巧性强的特点。

本书由张晓蕾任主编,杨旭、刘宝芹、罗如燕任副主编。本书章节的主要执笔者是:第 1 章、第 3 章由张晓蕾编写;第 2 章由刘宝芹编写;第 4 章由罗如燕编写;第 5 章由杨旭编写。其他执笔者还有:石涛、罗红霞、沈昕、王爱赫、关点、郑淑晖、董鑫、王浩轩、肖柠朴、耿文杰、刘宇昕、耿大伟、鹿胜利、孟昭勇、于站江、杨继萍、郭政、于建海、崔元如、郑鹤、罗丹丹。

由于技术的不断变化以及操作过程中的疏漏,书中难免有不妥之处,恳请广大读者批评指正。

作 者

2009 年 10 月

# 目 录

|                               |     |
|-------------------------------|-----|
| 第 1 章 J2ME 手机游戏开发基础           | 1   |
| 1.1 J2ME 手机游戏开发的基础知识          | 1   |
| 1.1.1 J2ME 手机游戏概述             | 1   |
| 1.1.2 J2ME 手机游戏开发的技术简介        | 2   |
| 1.2 手机游戏的策划                   | 7   |
| 1.2.1 手机游戏策划基础                | 7   |
| 1.2.2 游戏《太空之战》的策划             | 12  |
| 思考与练习 1                       | 17  |
| 第 2 章 J2ME 编程环境               | 18  |
| 2.1 J2ME 手机游戏的基本开发环境          | 18  |
| 2.1.1 JDK 的安装与配置              | 18  |
| 2.1.2 J2ME WTK 的安装与应用         | 21  |
| 【案例 1】第一个 MIDlet 项目           | 29  |
| 2.2 J2ME 集成开发环境               | 37  |
| 2.2.1 NetBeans 的安装与应用         | 37  |
| 2.2.2 Eclipse 的安装与配置          | 42  |
| 2.2.3 厂商模拟设备的安装与设置            | 54  |
| 【案例 2】NetBeans 环境下的 J2ME 项目开发 | 62  |
| 【案例 3】通过游戏生成器创建简单游戏           | 73  |
| 【案例 4】Eclipse 环境下的 J2ME 项目开发  | 75  |
| 思考与练习 2                       | 80  |
| 第 3 章 J2ME 游戏程序设计基础           | 81  |
| 3.1 J2ME 程序设计基础               | 81  |
| 3.1.1 MIDlet                  | 81  |
| 3.1.2 MIDP 用户界面简介             | 84  |
| 3.1.3 Display                 | 87  |
| 3.1.4 MIDP 事件处理机制             | 89  |
| 【案例 5】屏幕画面切换 1                | 91  |
| 【案例 6】屏幕画面切换 2                | 94  |
| 3.2 MIDP 的高级 UI 对象            | 97  |
| 3.2.1 TextBox                 | 97  |
| 3.2.2 Alert                   | 98  |
| 3.2.3 List                    | 99  |
| 3.2.4 Form                    | 101 |
| 【案例 7】高级 UI 设计的游戏菜单界面         | 106 |

|       |                             |     |
|-------|-----------------------------|-----|
| 3.3   | MIDP 的低级 UI 对象 .....        | 110 |
| 3.3.1 | Canvas 画布 .....             | 110 |
| 3.3.2 | Graphics 图形处理 .....         | 124 |
| 3.3.3 | 绘制文本 .....                  | 130 |
|       | 【案例 8】低级 UI 设计的高亮菜单 .....   | 132 |
| 3.4   | 图像处理 .....                  | 138 |
| 3.4.1 | 游戏图像处理基础 .....              | 138 |
| 3.4.2 | 创建 Image 图像 .....           | 145 |
| 3.4.3 | 图像的绘制 .....                 | 149 |
| 3.4.4 | 双缓冲绘图 .....                 | 153 |
|       | 【案例 9】图像的应用 .....           | 156 |
|       | 思考与练习 3 .....               | 161 |
| 第 4 章 | J2ME 手机游戏设计 .....           | 162 |
| 4.1   | 游戏引擎框架设计 .....              | 162 |
| 4.1.1 | 【相关知识】MIDP 2.0 中的游戏类 .....  | 162 |
| 4.1.2 | 【相关知识】GameCanvas 类 .....    | 163 |
| 4.1.3 | 【相关知识】游戏中的多线程应用 .....       | 166 |
| 4.1.4 | 【相关知识】游戏引擎框架结构 .....        | 169 |
|       | 【案例 10】游戏引擎框架的实现与应用 .....   | 177 |
|       | 【案例 11】《太空之战》游戏菜单设计 .....   | 186 |
| 4.2   | 游戏精灵与动画 .....               | 193 |
| 4.2.1 | 游戏精灵 .....                  | 193 |
| 4.2.2 | Sprite 类 .....              | 195 |
|       | 【案例 12】行走四方——精灵动画 .....     | 206 |
| 4.3   | 游戏场景地图与图层管理 .....           | 220 |
| 4.3.1 | 游戏地图设计 .....                | 220 |
| 4.3.2 | 图层管理 .....                  | 225 |
|       | 【案例 13】行走四方——卷动地图设计 .....   | 230 |
|       | 【案例 14】行走四方——地图障碍物设计 .....  | 238 |
| 4.4   | 游戏音效设计 .....                | 254 |
| 4.4.1 | J2ME 音效设计基础 .....           | 254 |
| 4.4.2 | 音频处理 .....                  | 257 |
|       | 【案例 15】行走四方——音效设计 .....     | 259 |
| 4.5   | 记录存储管理 .....                | 263 |
| 4.5.1 | 记录管理系统 RMS 概述 .....         | 263 |
| 4.5.2 | 管理记录存储 .....                | 264 |
| 4.5.3 | 记录存储的枚举、过滤与排序 .....         | 267 |
|       | 【案例 16】行走四方——场景的存储与加载 ..... | 270 |
|       | 思考与练习 4 .....               | 280 |

|                            |     |
|----------------------------|-----|
| <b>第 5 章 综合案例——太空之战</b>    | 281 |
| 5.1 游戏架构设计                 | 281 |
| 5.1.1 游戏简介                 | 281 |
| 5.1.2 游戏的架构                | 281 |
| 5.2 【综合案例 1】游戏引擎框架设计       | 283 |
| 5.2.1 设计效果                 | 283 |
| 5.2.2 项目创建与素材准备            | 284 |
| 5.2.3 游戏 MIDlet 的设计        | 285 |
| 5.2.4 游戏音效设计               | 287 |
| 5.2.5 闪屏画面设计               | 290 |
| 5.2.6 游戏启动画面设计             | 292 |
| 5.2.7 游戏菜单设计               | 296 |
| 5.2.8 游戏引擎框架设计             | 296 |
| 5.2.9 游戏图标设计               | 298 |
| 5.3 【综合案例 2】游戏关卡框架与卷动背景的设计 | 299 |
| 5.3.1 设计效果                 | 299 |
| 5.3.2 项目创建与素材准备            | 300 |
| 5.3.3 卷动背景设计               | 300 |
| 5.3.4 图像文字设计               | 302 |
| 5.3.5 游戏关卡框架设计             | 305 |
| 5.3.6 对 GameWorld 的修改      | 310 |
| 5.4 【综合案例 3】玩家飞机与子弹设计      | 311 |
| 5.4.1 设计效果                 | 311 |
| 5.4.2 项目创建与素材准备            | 311 |
| 5.4.3 玩家飞机设计               | 312 |
| 5.4.4 玩家子弹设计               | 315 |
| 5.4.5 将玩家飞机添加到关卡中          | 323 |
| 5.5 【综合案例 4】敌人飞机与子弹设计      | 324 |
| 5.5.1 设计效果                 | 324 |
| 5.5.2 项目创建与素材准备            | 325 |
| 5.5.3 敌人飞机设计               | 325 |
| 5.5.4 敌机子弹设计               | 337 |
| 5.5.5 对玩家飞机的修改             | 342 |
| 5.5.6 将敌机添加到关卡中            | 343 |
| 5.6 【综合案例 5】爆炸效果与奖励设计      | 345 |
| 5.6.1 设计效果                 | 345 |
| 5.6.2 项目创建与素材准备            | 345 |
| 5.6.3 奖励设计                 | 346 |
| 5.6.4 爆炸效果设计               | 349 |
| 5.6.5 将奖励和爆炸应用到关卡中         | 352 |
| 思考与练习 5                    | 356 |

# 第 1 章 J2ME 手机游戏开发基础

## 1.1 J2ME 手机游戏开发的基础知识

### 1.1.1 J2ME 手机游戏概述

#### 1. 手机游戏概述

电子游戏发展至今已经有五十多年的历史，从最初不经意的小游戏发展到现在，电子游戏产业已经成为全球数码娱乐产业的重要组成部分，在带来大量争议的同时，也带来了巨大的财富。

随着游戏开发技术的进步，以及手机硬件技术的发展，电子游戏在手机上的实现得到了良好的支持，现在可以随时随地玩你喜欢的游戏。但是，当前的手机游戏还仍然处在起步阶段，受限于软、硬件技术以及手机本身的局限——如运算能力、存储能力、屏幕显示能力等，手机还无法像普通计算机一样可以运行需要复杂运算、华丽图像的游戏。因此，目前的手机游戏显得比较简单，有些像当年 DOS 环境下的游戏，如图 1-1-1 所示。



图 1-1-1 手机游戏画面

不需要为手机游戏的简陋而感到悲伤，就如同 DOS 游戏时代一样，手机游戏时代的来临也正意味着个人设计者的新机遇——不像已经发展成熟的计算机游戏，需要大量的游戏设计人员，成立庞大的开发部门来进行游戏的开发。恰恰相反，这是一个个人英雄主义的新时代，它不需要由大量的美工、程序员和先进的图形图像设备组成的团队，花费数年的时间来开发一个游戏；只要用很少的几个人，不大的投入，就可以进行手机游戏的开发了，这对手机游戏开发者来说是件好事。手机设备还处在起步阶段，它留给我们很大的成长空间。

总而言之，手机游戏和传统游戏开发有着极大的区别，如：低预算、低成本，开发团



队小，易于管理；基于开放的设计标准，易于实现与扩展；手机本身作为通信器材的特点，为支持多人同时游戏铺平了道路。

2. J2ME 简介

J2ME (Java 2 Micro Edition, Java 2 微型版) 是 Sun 公司针对嵌入式、消费类电子产品推出的开发平台，与 J2SE 和 J2EE 共同组成 Java 技术的三个重要的分支，分别应用于不同领域的程序开发：J2SE (Java 2 Standard Edition, Java 2 标准版) 提供了 Java 的基础类库，主要用于桌面应用程序的开发；J2EE (Java 2 Enterprise Edition, Java 2 企业版) 主要用于企业级应用；而 J2ME 则是为开发信息电子设备的嵌入式应用程序而设计，如图 1-1-2 所示。

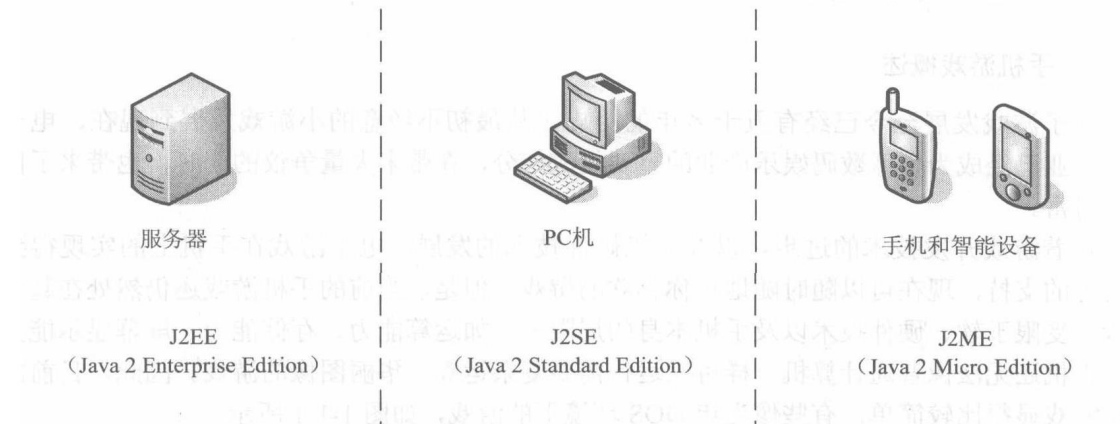


图 1-1-2 Java 的不同版本适用于不同平台

J2ME 实际上是一系列规范的集合，由 JCP (Java Community Process) 制定相关的 JSR (Java Specification Request) 并发布，各个厂商会按照规范在自己的产品上进行实现，但是必须要通过 TCK 测试，以确保兼容性。比如 MIDP 2.0 规范就是基于 JSR118 制定的。下面，我们来了解一下 J2ME 平台的体系结构。

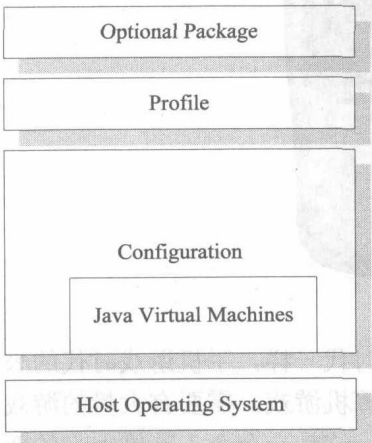


图 1-1-3 J2ME 体系结构示意图

1.1.2 J2ME 手机游戏开发的技术简介

J2ME 开发平台是由配置 (Configuration) 和简表 (Profile) 构成的。配置包含了 Java 虚拟机和基础类库。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：可选包 (Optional Package)，它是针对特定设备提供的类库，比如某些设备是支持蓝牙的，针对此功能，在 J2ME 中制定的 JSR82 (Bluetooth API) 提供了对蓝牙的支持。图 1-1-3 是 J2ME 的结构示意图。

1. CDC 与 CLDC

(1) CDC 与 CLDC。目前，J2ME 中有两个最主要的配置，分别是 CLDC (Connected Limited Devices Configuration，有限连接配置) 和 CDC (Connected Devices Configuration，连接配置)。它们是根据设备的硬件性能进行区分的，例如处理器、内存容量等。

- CDC。CDC 是为像数字电视机顶盒和汽车导航系统这样的智能设备而创建的，它有一个单独的简表，还有一个称为 CVM (Compact Virtual Machine) 的高性能虚拟机，它具有与 J2SE 几乎完全一样的开发性能。

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。

CDC 的硬件参数规范如下：

2MB 以上内存；

具有网络连接能力，通常为无线网络；

需要实现 Java 虚拟机规范的全部功能；

32 位或者 64 位的处理器。

- CLDC。CLDC 是为小型设备——特别是手机设备的开发而创建的，是用来开发在无线设备上运行的应用程序所要用到的一组最小的 API。CLDC 描述了所有无线移动设备所需的一个最小级别的功能集合。

在 CLDC 的参考实现中包括了一个称为 KVM (K Virtual Machine) 的虚拟机，所有的 CLDC 程序均在 KVM 上运行，就像 Java 程序运行于 JVM (Java Virtual Machine) 上一样。

(2) CLDC 1.0 规范。2000 年 5 月，JCP (Java Community Process) 公布了 CLDC1.0 规范，即 JSR30 (Java Specification Request 30)。作为第一个面对小型设备的 Java 应用开发规范，CLDC 是由包括 Nokia, Motorola 和 Siemens 在内的 18 家全球知名公司共同协商完成的。

CLDC 实质上只是定义了一个设备制造商用来实现 Java 运行环境的标准，第三方开发者只要遵循这个标准，就可以确定这个软件与符合标准的运行平台是否是相兼容。

由于这个标准是在 2000 年的时候指定的，而现在手机终端的处理能力和内存容量发展很快，如果还按照这个标准来评判可能就不准确了。因此，这里只是列出标准，供读者参考。

CLDC 的硬件参数规范如下：

至少 160KB 的内存；16 位或者 32 位的处理器；有限或非持续网络连接；简单有限的用户界面，通常只限于文本；低能耗，通常由电池供电。

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有至少 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。

这里所说的固定内存是指拥有写保护，不会因关机而丢失的 ROM。对于具体设备的实现，这些需求也可能有变化。规定的 160KB 固定内存是 CLDC 规范中的要求，实际上最小内存也可以是 128KB。

CLDC 1.0 规范包含了一个基本的 J2ME 运行环境，其中包括虚拟机 KVM 和核心的 Java 类库。作为专门针对于小型设备的配置，CLDC 对 J2SE 类库进行了大量的简化，其类库只保留了 Java 规范中定义的最核心的 3 个包，即 `java.io`、`java.lang` 和 `java.util`，并重新定义了一个新的包 `javax.microedition` 用于小型设备的应用开发。这里可以通过前缀来区别：`java.`表示核心的 Java 包，`javax.`表示标准 Java 扩展包。

从这里可以看到，CLDC 的类库可以分为两种：一种是从 J2SE 标准类库中继承的；另

一种是扩展的、专门为 CLDC 设计的（这部分类也可以被映射到 J2SE 中）类库。

对于第一种 CLDC 类库，包括了 J2SE 的 3 个最核心的包 `java.io`、`java.lang` 和 `java.util`。而且这 3 个包和 J2SE 相比，也只是 J2SE 相对应包的一个很小的子集。例如，`java.util` 包中的类和接口就由 J2SE 中的 47 个减少到 10 个。

对于后一种 CLDC 类库，只有描述标准连接框架的 `javax.microedition.io` 包，它和 MIDP 中定义的包一起放于 `javax.microedition` 包中。

`java.lang` 包包含了 Java 语言 API 的核心部分继承下来的类，但是 CLDC 只继承了 J2SE 中部分的类，而且一些类中的接口并没有完全实现。

`java.io` 包是 J2SE 的子集，只提供了相当有限的 8 位输入/输出功能。而且，一些抽象类，像 `FileInputStream` 等也被省掉了，原先从这些抽象类继承的类直接从它们的父类 `InputStream` 和 `OutputStream` 继承。

`javax.microedition.io` 包括了通用连接框架（GCF, Generic connection framework）中的类和接口，其中最关键的是 `Connector` 类和 `Connection` 接口。`Connector` 类中定义了静态方法生成特定类型的 `Connection`，利用这个 `Connection` 可以访问网络和各种其他设备。

`javax.microedition` 中其他的包定义了 CLDC 中没有定义的功能，如对应用程序生命周期的管理、用户界面（UI）、事件处理模式、永久性存储和用户与应用程序的交互等。这些功能的定义是由简表（即 MIDP）来完成的。

CLDC 标准为了能够涵盖尽可能多的设备，其类库只包含了最小的 Java 平台特性和 API。面对严格的内存限制和当前各种各样的小型设备，CLDC 不可能覆盖全部设备。因此在 CLDC 的规范中，不可避免地造成对某些设备要求过高或是对另一些设备要求又太低的现象。

为了确保与其他 Java 平台的兼容性，绝大多数的 CLDC 类库是从 J2SE 和 J2EE 中继承的，是 J2SE 和 J2EE 的子集。由于目标设备的特殊性，CLDC 类库在安全、输入/输出、用户界面、网络和存储管理等方面没有全部使用 J2SE 来实现；CLDC 对其中的部分类库进行了重写，如网络连接类库等。

（3）CLDC 1.1 规范。为了适应硬件水平的不断提高和应用的需要，JCP 又发布了 JSR139，即 CLDC 1.1 规范。相对于 1.0 版本，CLDC 1.1 规范并没有本质上的变化，它在兼容性和可用性上做了一些改进，并增加了一些 1.0 版本没有的新特性，下面介绍其中的一部分内容。

- 增加对浮点数据的支持，核心类库中增加了 `java.lang.Float` 类和 `java.lang.Double` 类，这是与 1.0 版最主要的区别；
- 部分支持弱参考（weak references）；
- 对错误处理有了更加明确的定义，并增加了 `NoClassDefFoundError` 类；
- 对于 `Thread` 类，CLDC1.1 允许为线程命名，允许中断线程，增加了新的构造方法；
- 设备的最小内存被提高到 160 KB 至 192 KB。

本书中的游戏开发，都是基于 CLDC 1.1 的，要了解 CLDC 1.1 的详细内容可以去 Sun 官方网站下载 CLDC 1.1 的规范，网址如下：

<http://www.jcp.org/en/jsr/detail?id=139>

## 2. MIDP

在上面的 CLDC 标准中可以发现，它缺少一些在编程时所需要使用的组件，比如用户界面。因为这些组件是与设备相关的，因此，它们由 Profile（简表，也称为设置文件）来实现。

Profile 是以 Configuration（配置）为基础的，用于在 Configuration 上加入设备的相关组件，与 Configuration 一起构成了完整的运行环境。Profile 对设备横向分类，针对特定领域细分市场，内容主要包括特定用途的类库和 API。

MIDP（Mobile Information Devices Profile，移动信息设备简表）就是 CLDC 上的重要简表，它是以 MID（Micro Information Devices，微型信息设备）为目标的。

相对于普通的计算机来说，MID 的计算能力很弱，它的功能相对也较少。MIDP 设定了 MID 应该满足的目标设备，通过设立一个最低的常用基准目标平台来表现它，这个平台是大多数移动设备，特别是手机可以满足的。MIDP 部分解决了无线终端产品上 Java 应用程序的兼容问题。

MIDP 在软硬件两方面对 MID 进行了规范，包括基本内核、存储能力、显示能力、网络能力、输入方式等。

随着技术的发展，MIDP 的版本也在不断更新，每个版本都规范相应的能力特性。

（1）MIDP 1.0。在 MIDP 1.0 中，提供了如下功能：专门为 MIDs 开发特别设计的工具应用程序；适合小屏幕和输入受限的用户界面；永久存储（非易失性内存），也被称为记录管理（RMS，Record Management System）；网络连接（通过对 CLDC 连接框架的实现）；定时器。

这些功能分别在如下的包中提供：

`javax.microedition.io`，J2ME 的 I/O 操作类。CLDC 中关于网络连接的类也包括在 `javax.microedition.io` 中。

`javax.microedition.lcdui`，J2ME 的 GUI 类，提供界面编程功能。由于手机一般使用 LCD 显示器，因此称为“LCD UI”。

`javax.microedition.rms`，为 J2ME 提供持久存储。

`javax.microedition.midlet`，J2ME 程序 MIDlet 的基类，J2ME 程序都从 MIDlet 开始执行。

（2）MIDP 2.0 和 MIDP 2.1。MIDP 2.0 版本在网络、音效、用户界面、游戏开发和安全等方面有不小的提高。例如，支持播放 MIDI 声音文件和 WAV 声音文件，图层管理，增强的画布工具，PNG 图像的 Alpha 透明等。

这些功能分别在如下的包中提供：

`javax.microedition.lcdui.game`，一个简单的游戏引擎包。

`javax.microedition.media`，声音编程技术。

`javax.microedition.media.control`，声音控制技术。

MIDP 2.1 是对 MIDP 2.0 的补充，它提供了一些增强功能，如支持 JPG 图像、多种格式的 PNG 图像，MIDlet 最少可创建 10 个线程，增强的网络通信能力，Canvas 画布触摸事件等。

## 3. MIDlet

MIDlet 即 MIDP 小应用程序，它是在 MIDP 中定义的一种新的应用程序模型，是专门运行在移动设备上的，基于 CLDC 配置的程序，它被 AMS（Application Management

Software, 应用管理软件) 所管理。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 Destroyed()和 Paused()来实现的。

MIDlet 有三种状态, 分别是 pause (暂停)、active (活动) 和 destroyed (销毁)。在启动 (Start) 一个 MIDlet 的时候, 应用管理软件会首先创建一个 MIDlet 实例并使它处于 pause 状态, 当 startApp()方法被调用的时候 MIDlet 进入 active 状态, 即运行状态。在 active 状态调用 destroyApp(boolean unconditional)或者 pauseApp()方法可以使 MIDlet 进入 destroyed 或 pause 状态。图 1-1-4 说明了 MIDlet 的状态改变情况。

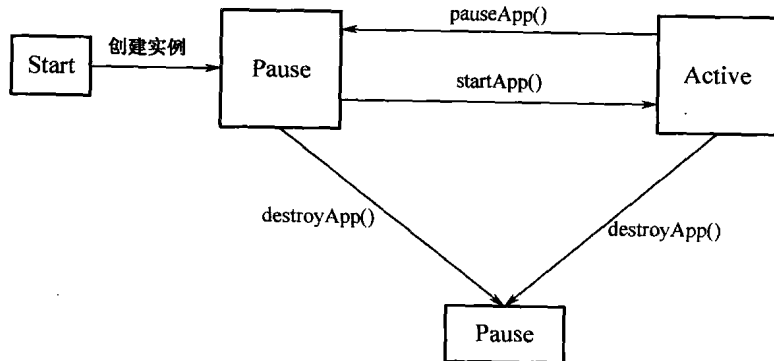


图 1-1-4 MIDlet 状态改变情况

下面的代码可用于验证 MIDlet 应用程序的状态改变情况。

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

public class Midlet extends MIDlet {

    private Display display;    //定义显示对象

    //构造函数
    public Midlet () {
        display = Display.getDisplay(this); //获取显示对象
        System.out.println("构造 MIDlet");
    }

    public void startApp() {
        System.out.println("MIDlet 开始运行");
        Form from = new Form("MIDlet");    //创建 Form 窗口实例
        display.setCurrent(from);           //设置 from 为当前显示对象
    }

    public void pauseApp() {
```

```

        System.out.println("停止 MIDlet");
    }

    public void destroyApp(boolean unconditional) {
        System.out.println("销毁 MIDlet");
    }
}

```

J2ME 程序的编辑需要专门的开发环境，可在后面学习了环境配置后，再来运行该程序。编译并运行程序，控制台上就会输出如下内容，以表示程序开始运行：

构造 MIDlet。

MIDlet 开始运行。

当退出该程序时，控制台就会输出如下内容，表示程序运行结束：

销毁 MIDlet。

## 1.2 手机游戏的策划

### 1.2.1 手机游戏策划基础

#### 1. 游戏策划的基本概念

在游戏设计中，最初的起点就是游戏的策划，游戏策划的优劣，对游戏有着重大的影响。

作为游戏开发小组中的策划者，只要有了一个新的想法或念头，就意味着可能有一个新游戏的诞生。在这个创意被充分讨论之后，再加上对其操作过程的趣味性及市场销售的可行性的预测等因素的准确判断，一个完整的策划方案才可能产生。

一个游戏从一个想法到成为产品需要经历很多的关卡，合格的策划应该在一开始就知道这个想法能否行得通，在经过了严格的论证并初步产生了产品的轮廓后，才能把自己的想法提出来，这也是一个游戏是否可行的一个自我论证过程。主要包括以下几个部分的内容：技术可行性分析、经济可行性分析和人力状况分析。

整理完设计思路并整理出一份合格的可行性报告，这就为全部策划开了一个好头。下面就是写一个完整的项目建议书，这是拿来给部门负责人看的。

在经过充分的讨论后，策划人员必须将讨论的重点写成文字，也就是提出完整的策划方案，经决策者同意认可后，才能进行下一步的工作。

游戏策划方案就像一部电影的剧本，它必须完整地涵盖整个游戏的故事、流程、内容、方式、游戏画面、角色造型、场景规划、人工智能、硬件配备、市场评估等，主要包括：故事设计、脚本设计、玩法设计、角色设计、关卡设计、AI 设计、音效设定、场景设定、市场定位和收益评估等多方面的内容。

对整个游戏过程的详细描述及实施规划都应记录在案。当进入创作过程之后，策划还必须随时和美术设计师和程序设计员保持联系，以免游戏程序的编写失控。策划应能对游戏设置的内容与精神了如指掌，与各个小组及时沟通，并且控制整个游戏制作的进程。

策划书的内容首先包括了制作这个游戏的意义，又分析了这个游戏的市场前景，既谈到游戏情节定位，又理顺了游戏开发的全过程，内容之大、之多是其他策划书所不能比拟的。

## 2. 游戏设计的基本内容

要设计一个游戏，必须要确定以下几个重要内容：游戏类型、游戏的背景、游戏模式、游戏的程序技术和市场定位。

(1) 游戏的类型。首先要确定设计的游戏是属于哪一种类型，是 RGB、RTS、ACT 还是 FPS。

(2) 市场定位。不论所设计的游戏构想如何的好，如果没有清楚地定位市场，那么到时制作完成的游戏可能会面临销售不佳的窘状，所以在设计游戏之前得知道所定位的游戏族群。比如，青少年可能对角色扮演、战略、射击、体育等游戏感兴趣，男孩可能偏重于比较激烈的战略游戏，而女孩可能偏重于比较休闲的游戏。

(3) 游戏背景。游戏背景就是确定游戏发生的时空架构，是中国的还是外国的；是古代的还是现代的；是有真实历史背景的还是虚幻架空的。比如，《轩辕剑》游戏系列定位于中国古代（已发售的版本跨越了从先秦到隋唐的各个历史时期，也包括了一些古代神话传说）的时空背景。《魔兽世界》则发生在一个魔法与科学并存的虚幻世界——艾泽拉斯（Azeroth）大陆。《星球大战》是发生在未来的太空之中的科幻游戏。

时代背景对于游戏美工来说是非常重要的，因为决定一个时代背景意味着确定了在设计角色造形和场景、建筑时的依据。

(4) 游戏模式。当决定了游戏类型及游戏背景之后，接下来就要构思游戏中所要呈现的模式，包括游戏中的各种元素、游戏规则等。

假设定位的背景是架空的魔幻世界，类型定位为 RPG，则要考虑游戏的场景、故事流程、支线剧情、角色行为、武器道具、升级方式等。假设定位的背景是二战时期，类型定位为 FPS，这时需要考虑游戏的进行方式和相关元素，这些元素可能包括游戏场景（在陆地、空中还是海上）、武器弹药、人物的游戏方式等。假设背景定位在古代中国，而类型定位为 SLG，这时游戏需要的元素可能包括城池、将领、士兵、金钱、粮食和矿产等内容，游戏方式则可能包括战争、人材的搜索、税收等。

(5) 程序设计与测试。无论对一个游戏想得多好，架构设计多庞大，如果限于程序开发人员本身的技术或当前的软硬件开发技术无法达成的话，那么一切还是流于空谈。所以在设计一个游戏之前必须要先去征询程序开发人员的意见，游戏能否在现有的技术范围内实现，是使用 2D 还是 3D，使用什么样的游戏引擎来实现等。

(6) 维护与更新。在游戏发行之后，还需要继续注意游戏的维护与更新。这样做，一方面是为了游戏运行良好、稳定，修正错误；另一方面是可以扩展游戏的可玩性，对游戏内容进行扩充，延长游戏的生命周期，从游戏中获取更多收益。

## 3. 手机游戏的分类

顾名思义，所谓手机游戏就是可以在手机上进行的游戏。随着科技的发展，现在手机的功能越来越多，越来越强大。而手机游戏也远远不是印象中的什么“俄罗斯方块”“贪吃蛇”之类画面平淡，规则简单的游戏，进而发展到了可以和掌上游戏机媲美，具有很强的娱乐性和交互性的复杂形态了。

(1) 按游戏技术特征分类。手机游戏可以根据游戏本身的不同，而分成文字类游戏和图形类游戏两种。

- 文字类游戏。文字类游戏是以文字交换为游戏形式的游戏。这种游戏一般都是通过玩家按照游戏发给手机的提示，来回复相应信息进行的游戏。

文字类游戏主要分为两种：短信游戏和 Wap 浏览器游戏。

短信游戏是玩家和游戏服务商通过短信来交流的，以达到进行游戏的目的。由于短信游戏是通过文字来表达，造成短信游戏的娱乐性较差。但是短信游戏却是兼容性最好的手机游戏之一。只要手机可以发短信，就可以畅快地享受短信游戏。

Wap 是一种手机拨号上网的网络服务，而 Wap 浏览器游戏就好像用计算机上网，并通过浏览器浏览网页来进行简单的游戏，它也属于一种文字游戏，其方法和短信游戏类似，玩家可以根据 Wap 浏览器浏览到的页面上的提示，通过选择各种不同的选项的方法来进行游戏。Wap 游戏也有短信游戏不够直观的缺点。

文字类游戏都有着共同的一个特点，即游戏是通过文字描述来进行的。游戏过程中，需要玩家进行过多的想象，使得游戏相对比较单调。

- 图形类游戏。图形类游戏更接近计算机游戏。玩家通过动画的形式来发展情节进行游戏。由于游戏采用了更为直观且更为精美的画面，因此图形类游戏的娱乐性往往较文字类游戏高，因此广受玩家的欢迎。

(2) 按游戏类型分类。按游戏类型分类的方法分类，与传统的计算机游戏分类相似，常见的手机游戏可分为如下几种。

- 角色扮演类游戏 (RPG, Role-Playing Game)。角色扮演类游戏是由玩家扮演游戏中的一个或数个角色，有完整故事情节的游戏。玩家可能会与冒险类游戏混淆，其实区分很简单，RPG 游戏更强调的是剧情发展和个人体验，注重剧情。图 1-2-1 是角色扮演类游戏《诛仙传说 II》和《天使之恋》的游戏截图。



图 1-2-1 角色扮演类游戏《诛仙传说 II》和《天使之恋》游戏截图

- 冒险类游戏 (AVG, Adventure Game)。冒险类游戏是由玩家控制游戏人物进行虚拟冒险的游戏。与 RPG 不同的是，AVG 的特色是故事情节往往是以完成一个任务或解开某些谜题的形式出现的，而且在游戏过程中刻意强调谜题的重要性。图 1-2-2 是冒险类游戏《音速小子决战冒险岛》和《疯狂冒险岛》的游戏截图。



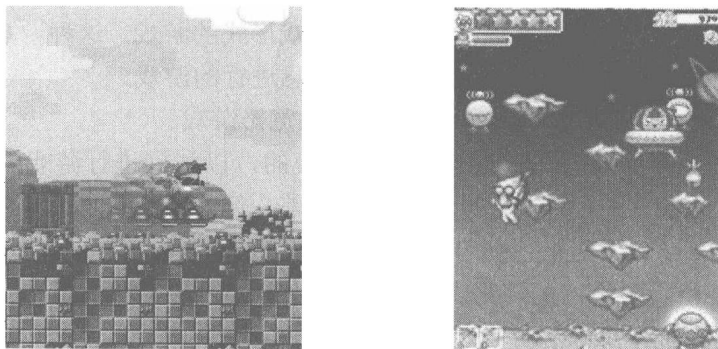


图 1-2-2 冒险类游戏《音速小子决战冒险岛》和《疯狂冒险岛》的游戏截图

- 动作类游戏（ACT，Action Game）。动作类游戏是由玩家控制游戏人物用各种武器消灭敌人以过关的游戏，不追求故事情节。图 1-2-3 是动作类游戏《三国猛将》和《鬼武者 激斗版》的游戏截图。

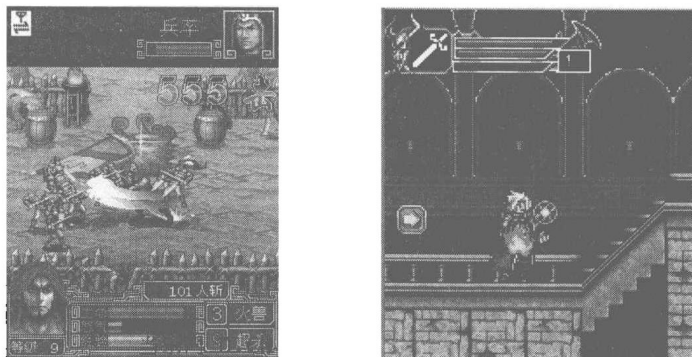


图 1-2-3 动作类游戏《三国猛将》和《鬼武者 激斗版》的游戏截图

- 射击类游戏。射击类游戏可分为两类。

第一人称视角射击游戏。严格来说它属于动作游戏的一个分支。由于其在世界上的风靡，使之成为一个单独的类型，典型的有《毁灭战士》（DOOM）、《雷神之锤》（QUAKE）、《虚幻竞技场》、《反恐精英》（CS）等。

射击游戏。射击游戏区别于 CS 之类的模拟射击，它是指纯射击游戏，由玩家控制各种对象（如飞机、坦克、潜艇等）完成任务或过关的游戏。图 1-2-4 是射击类游戏《眼镜蛇攻击》和《荣誉勋章》的游戏截图。



图 1-2-4 射击类游戏《眼镜蛇攻击》和《荣誉勋章》的游戏截图