

基于 Android 的五子棋游戏设计

软件工程

【摘要】 本论文主要阐述以面向对象的程序开发语言 `eclipse` 为开发工具，基于智能手机 `Android` 之上设计一个五子棋游戏。五子棋起源于中国古代的传统黑白棋种之一，它不仅能增强思维能力 提高智力，而且富含哲理，有助于修身养性。本系统是本系统中设定了人机对战和人人对战两种模式。人机对战实现的是一些简单的算法，人人对战只是进行了输赢方的判定。它与 `javame` 的五子棋设计有许多的不一样，`Android` 本来就是全触摸的形式，而且操作非常简单。

【关键词】 机器人， 生命周期，算法，线程

The five in a row game based on Android

【Abstract】 This paper mainly expounds the object-oriented program development in `eclipse` for language development tool, smartphone `Android` based on above design a five in a row game. Gobang originated in China ancient traditional black white kind, it not only can enhance one of thinking capacity, improve intelligence and high in philosophy, helps to cultivate one's morality raises a gender. This system is the system that set the man-machine against and everyone against two patterns. The realization man-machine against some simple algorithm, everyone was against just win the judgement of the party. It has with the `javame` gobang chess design many are dissimilar, `Android` is the entire touching form originally, moreover operates is simple.

【Key words】 `Android`, `Activity`, arithmetic, `Thread`

目 录

绪论	5
1 开发概述	6
1.1 背景	7
1.2 课题来源	7
1.3 国内外研究现状	8
1.4 论文主要工作	8
2 Android 的开发环境	9
2.1 操作系统	9
2.2 Android 开发环境	9
2.3 Android 开发环境变量配置	9
2.3.1 jdk 的配置	9
2.3.2 sdk 的配置	11
2.4 eclipse 设置	11
2.5 Android 开发环境配置最终步骤	11
3 Android 应用程序核心	11
3.1 应用程序基础	12
3.2 应用程序组件	12
3.2.1、活动 (Activities)	13
3.2.2、 服务 (Services)	15
3.2.3、广播接收者 (Broadcast receivers)	16
3.2.4、内容提供者 (Content providers)	17
4 相关技术概念	17
4.1 坐标布局 (AbsoluteLayout)	17
4.2 Intent	19

4.3	AlertDialog.....	21
4.4	相关的算法研究:	21
4.4.1	相关的数据结构.....	21
4.4.2	分析规则.....	21
4.4.3	胜负判断.....	23
4.4.4	算法实现描述.....	24
5.	实现运行情况, 以及各个模块的截图	26
5.1	概要设计:	26
5.2.1	游戏主界面:	27
5.2.2	新游戏界面:	28
5.2.3	帮助界面	28
5.2.4	关于五子棋介绍界面	29
5.2.5	五子棋的退出:	30
	总结	31
	参考文献	32
	附录	33

绪论

智能手机(Smartphone)，具有独立的操作系统，像个人电脑一样支持用户自行安装软件、游戏等第三方服务商提供的程序，并通过此类程序不断对手机的功能进行扩充，同时可通过移动通讯网络来实现无线网络接入。目前，全球多数手机厂商都有智能手机产品，而芬兰诺基亚、美国苹果、加拿大 RIM（黑莓）、美国摩托罗拉、中国台湾宏达（htc）更是智能机中的佼佼者。

智能手机具有五大特点：1. 具备了无线接入互联网的能力，即要有 GSM 的支持网络下的 GPRS 或者是 CDMA 网络支持下的 CDMA 1X 或 3G 网络。2. 人性化，可以根据个人需要扩展机器的功能。3. 它是具有开放性的操作系统，还可以安装非常多的应用程序，使得智能手机的功能可以得到无限的扩展。4. 具有 PDA 功能，其中包括个人信息管理、日程记事、任务安排、多媒体应用、浏览网页。5. 强大的功能，扩展性能强，第三方软件支持多。

Android 的诞生：从时间要追溯到二零零三年，在参与到 Android 的传奇之前，它的创始人 Andy Rubin 在周刊中这么说过：“在开发对于用户位置以及状态更加敏感的移动设备方面存在着巨大的潜力”，他还说过：“很快他们就会在消费类产品中使用这些位置信息，如果开发者足够聪明的话。”现在看起来他自己都没想到他的预言是多么的准确：实际上，移动领域将会发生巨变，从而催生出我们从未见过的强大的智能手机。到二零零五年有很多关于 Google 即将进军移动设备制造领域并专注于制造基于位置服务的手机的谣传，但最大的障碍就在于 Google 没有自己的可以达成这个目标的平台。

在二零零五年七月，Google 收购了 Android，这么一个小规模收购行动还是立刻引起了移动业界的兴趣，它们怀疑这是 Google 进入移动市场的标志。尽管如此，这个当时只是一个小规模的移动软件开发公司正是由 Andy Rubin、Rich Miner、Nick Sears 以及 Chris White 四个人共同创建的。现在看来，他们的怀疑是正确的。差不多两年半以后，在二零零七年十一月五日，Google 发布了 Android，它这个的发布给移动业界带来了一场可怕的风暴，，基于 Linux 的全新移动操作系统。这不单单是一个全新的移动系统，非常重要的一点是：它是开源的。

这是当时新成立的开放手机联盟整合下的成果，诸如 Google、HTC、Intel、Sprint-Nextel、T-Mobile 以及 NVIDIA 都是这个联盟的成员，该联盟的目的就是要创立移动设备的开放标准。第一款搭载 Android 系统的手机是 2008 年 9 月由 T-Mobile 发布的 G1，它拥有 GPS 功能，3.1M 摄像头，以及一系列的 Google 应用。G1 的面市，正式拉开了 Google 以及整个移动业界激动人心的大冒险的序幕。

Android 操作系统版本

Android 1.1 (发布于 2009 年 2 月)

Android 1.5 “Cupcake” 这是第一个主要版本，发布于 2009 年 5 月。这个升级之后，它的更新给用户带来了新的体验。用户界面得到了极大的改良，并且增添了以下功能。Android 1.6 “Donut” 2009 年 9 月发布。搭载 Android 1.6 的 HTC Hero 获得了意想不到的成功。使得原来用许多都没有用过 Android 的人都对 Android 非常的关注，并且也让竞争者苹果和微软更加警惕了。

Android 2.0/2.1: 2009 年 10 月 26 日，又一个主要版本升级以创纪录的速度放出。这次，大版本升级到了 Android 2.0/2.1 “Eclair.”。

Android 2.0.1 SDK 在 2009 年 12 月 3 日发布出来，后来是 2010 年 1 月 10 日发布的 2.1 版本。很多围观群众和用户可能会有点奇怪：“Android 为什么会被用甜点作为它系统版本的代号？”，其实这个命名方法开始在 Android 1.5 发布出来的时候。每一个版本代表的甜点的尺寸就越变越大，然后按照字母的数序：小蛋糕，甜甜圈和松饼。在这之前人们预计 2.2 版本的代号可能是“馅饼”，但这个最终被证明是错误的，“FroYo”（冻酸奶）才是 Android 2.2 这个伴随 Google Nexus One 发布的新版的最新代号。而下一个版本的代号将会是“姜饼”。Android 2.2 “Froyo”：Nexus One 的用户已经开始接受 Android 2.2 “FroYo” 的无线更新，其它机型的用户可能会在短时间内获取到这一更新的。这一版本的更新出来会让大量让其他手机用户垂涎三尺。

有关的消息众口一词声称 Android 3.0 “Gingerbread”，Google 将会在今年秋天的某个时候发布出来，它将带来给用户新的体验。

从 Android 的发展旅途中，我们看到 2003 年 Andy 他是一个有远见的人，看到了 2005 年的起步和被 Google 的收购，也看到了系统的进步以及新的机型的快速发布。所有这一切都让我们相信，Android 将会走向世界第一移动操作系统宝座的步伐是势不可挡的。Android 平板电脑也在今年的夏天开始发布以及 Google TV 发布以后（也许还有 Google Music?），Android 可能真的会将继续带着我们走向移动设备梦想的彼岸。

1 开发概述

在手机游戏的早期，大多数开发者都是想要玩转设备、学习技巧和快速攻关的狂热分子。但是，随着更多的开发者加入这个行业，手机游戏已经进化成涉及到专业设计师、构造师、开发者、产品经销商、律师、和艺术家在内的主导产品。这是行业走向成熟的必然趋势。尽管在某人的业余时间内要开发和推销简单的游戏更加困难，但是更多的专业商店将为许多人提供更多的就业机会。

从手机软件的平台种类，游戏玩家的心理，未来移动通信服务终端的发展，3G 手机网络对手机软件游戏的影响，还有从 J2ME 手机游戏开发平台介绍的手机游戏的趋势和未来这种游戏的运营模式，这几方面的内容来看，手机游戏是未来游戏发展的趋势，其移动性、互连性，必定会带来网络游戏的风暴，这就使我们对这一个新的领域有了一定的概念，对指导我们建立一个手机软件开发培训的知识结构有一定的帮助，也是我们的一个发展契机。

1.1 背景

在我们的生活中有许多的游戏，但是游戏不同，它也会给玩家带来不同的感受和生活理念。游戏是我们日常生活的一个必不可少的娱乐项目，也带动了游戏相关产业的发展。游戏平台是经过从游戏机到电视游戏，然后从到 pc 游戏，到手机游戏，中间经历了很多的技术的更新和变革。但随着新技术的发展和我们的生活水平的提高，手机开始逐渐变为普及，因为手机是非常小巧便捷的，而且方便的特性成为我们日常生活中必备通讯工具。而基于手机平台的游戏也就拥有的市场空间是巨大的，也因随着 3g 的开通和应用，手机游戏必然会迎来新一轮游戏热潮！

1.2 课题来源

现在的移动电话是小型的计算机，它的处理能力与台式机的标准处理能力相比很有限，但是足够运行一个小型的游戏。现在的手机的一个特性就是它们还是网络计算机，能够高速发送和接收数字数据。除了语音数据以外，它们还可以发送和接收其它类型的数据。所以这样的网络游戏也可以在手机上实现。短信类游戏就好像“虚拟宠物”那样。短信游戏是通过玩家和游戏服务商通过短信中的文字的内容来交流，达到进行游戏的目的的一种文字游戏。由于短信游戏的整个游戏过程都是通过文字来表达，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。只要您的手机可以发短信，您就可以畅快的享受短信游戏给您带来的快乐了。图形类游戏更接近我们常说的“电视游戏”，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直接表现，因此图形类游戏的游戏性和代入感往往较文字类游戏高。因此广受玩家们的欢迎。嵌入式游戏是一种将游戏程序预先固化在手机的芯片中的游戏。由于这种游戏的所有数据都是预先固化在手机芯片中的，因此这种游戏是无法进行任何修改的。也就是说，您不能更换其它游戏的，只能玩您手机中已经存在的娱乐游戏，而且您也不能将它们删除。手机在网络

游戏行业在多年的技术经验与运营经验的积累与总结之后，到了 2009 年迫不及待的需要创新出新的游戏模式和崭新的运营模式来推动手机游戏市场的爆发和发展。其中比较

吸引人们的是手机的休闲类的游戏。应该可以说它是与 PC 网络游戏完全的脱离开了，而且充分发挥了自身的优势手机终端。休闲类的游戏恰巧与之相反，只需要很少的网络流量的网络环境，通过少量多次的游戏形式带给玩家流畅的游戏体验满足玩家瞬间提升的快感，但是同时却很难具有长期的“沉迷”性。

本题目是基于 Android 平台开发的一个小型的程序。其实都差不多，在有 JAVA 的智能手机上都可以运行。其实手机是我们生活中不可缺少的工具，这些小游戏可以带给我们快乐和放松我们的心情。无论什么样手机，我想买起来的时候上面就有一些它自带的各种小游戏，其中有一个独特的小游戏也是非常重要的。

1.3 国内外研究现状

Android 是 Google 开发的基于 Linux 平台的开源手机操作系统。它包括操作系统、用户界面和应用程序——移动电话工作所需的全部软件，而且不存在任何以往阻碍移动产业创新的专有权障碍。Google 与开放手机联盟合作开发了 Android，这个联盟由包括中国移动、摩托罗拉、高通、宏达电和 T-Mobile 在内的 30 多家技术和无线应用的领军企业组成。Google 通过与运营商、设备制造商、开发商和其他有关各方结成深层次的合作伙伴关系，希望借助建立标准化、开放式的移动电话软件平台，在移动产业内形成一个开放式的生态系统。

Android 作为谷歌企业战略的重要组成部分，将进一步推进“随时随地为每个人提供信息”这一企业目标的实现。我们发现，全球为数众多的移动电话用户从未使用过任何基于 Android 的电话。谷歌的目标是让（移动通讯）不依赖于设备甚至平台。出于这个目的，Android 将补充，而不会替代谷歌长期以来奉行的移动发展战略：通过与全球各地的手机制造商和移动运营商结成合作伙伴，开发既有用又有吸引力的移动服务，并推广这些产品。

开放手机联盟的成立和 Android 的推出是对现状的重大改变，在带来初步效益之前，还需要不小的耐心和高昂的投入。但是，我们认为全球移动用户从中能获得的潜在利益是值得付出这些努力的。如果你也是一个开发者，并对我们的想法感兴趣，就请再给我们一星期的时间，届时谷歌便能提供 SDK 了。如果你是一名移动用户，只需再等一段时间，我们的一些合作伙伴计划在 2008 年下半年推出基于 Android 平台的电话产品。如果你已经拥有一部你了解并喜爱的电话，请登录 mobile.google.com，确保你已经安装谷歌手机地图、Gmail 以及其他一些专为你手机开发的精彩应用。谷歌将继续努力，让这些服务变得更好，同时也将添加更有吸引力的特性、应用和服务。

1.4 论文主要工作

此游戏主要是采用 Android 的 View 框架编写，部分框架运用 Android 中的其他组件。主要运用了算法来实现对棋局的分析，棋盘输赢的判定。

UI 方面:此游戏当进入游戏的界面时，调用的是一个用相对布局-AbsoluteLayout 的

XML 布局文件。其中包括了一个 ImageView 和 4 个 Button. 分别实现了不同的功能。第一个 Button 是进入新游戏, 点击按钮进入新游戏, 来选择对手。第二个 Button 是帮助, 用了一个 TextView 来说明怎样来进行游戏。第三个 Button 是游戏的关于, 用了一个 AlertDialog 来介绍了一下游戏的这个游戏的好处。第四个 Button 是游戏的退出按钮。在游戏中, 我用画线来不满了整个的手机屏幕, 在上面用了 TextView 来进行了文字的说明和提示。

本篇文章, 主要目的是为了探讨更加深入的理解 Android, 为移动终端上的游戏软件的打造提供一个思路。

2 Android 的开发环境

本章将主要介绍如何构建 Android 开发环境, 包括获取 AndroidSDK、Eclipse 以及进行 Android Eclipse 插件设置等。

2.1 操作系统

目前只支持 XP(32-bit)、Vista(32/64-bit)、Mac OS X 10.4.8 或更新版本(x86)、Linux(只在 Linux Ubuntu Dapper Drake 上面做过测试)。

系统 XP sp2/sp3 32bit 来做讲解, 其他系统的可以 google 一下差异, 反正都大同小异的。

2.2 Android 开发环境

无论是官方还是我本人, 都建议使用 eclipse 进行开发, 当然, eclipse 版本要在 3.3 以上, 这是官方说法, 3.3 以下我也没测过, 3.3 (Europa)、3.4 (Ganymede) 都可以, java 版本或者 EE 版本也都没问题的, 可以到 eclipse 官方下载最新版本。因为开发是基于 java 的, 所以 jdk 是少不了的, jdk 要求是 5.0 或者 6.0, 我还是用 5.0 比较多, 可以到 sun 官方下载。

2.3 Android 开发环境变量配置

2.3.1 jdk 的配置

首先要到 sun 官方下载 jdk, 注意的是下载 jdk 而不是下载 jre, 因为 jre 是提供 java 的运行环境而不是开发环境。sun 官方网站是: <http://java.sun.com>

下载后安装是下的解压包那就解压到某目录, 例如解压到是 C:\Program Files\Java\jdk1.5.0_14

右键点击我的电脑 -> 属性 -> 高级, 点击下面的环境变量, 在下面的“系统变量”处选择新建在变量名处输入 JAVA_HOME 变量值中输入刚才的目录, 比如我的就 C:\Program

Files\Java\jdk1.5.0_14 。效果如图 2-1 所示：

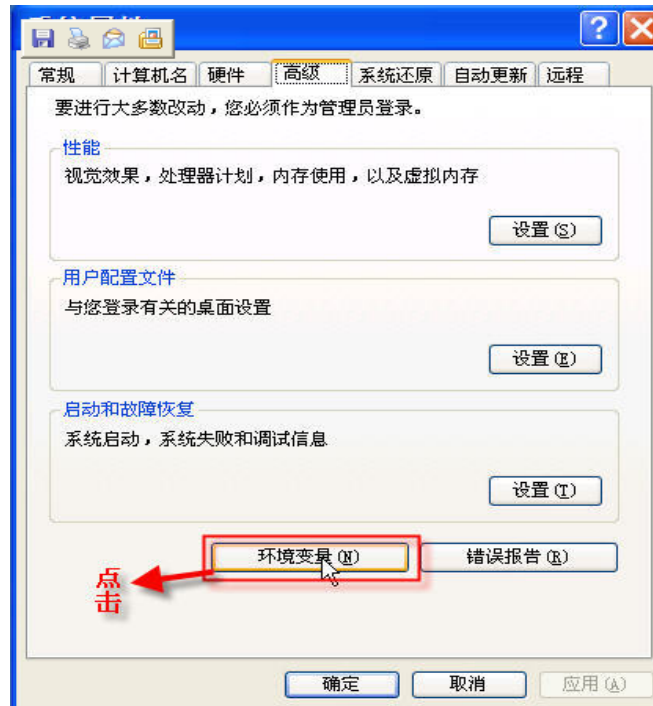


图 2-1 进入环境变量

确定后再新建一个变量名为 classpath，变量值为 %JAVA_HOME%/lib/rt.jar;%JAVA_HOME%/lib/tools.jar，确定后找到 PATH 的变量，双击或点击编辑，在变量值最前面加上 %JAVA_HOME%/bin；注意里面的，确定、确定、确定后即可。具体如下图 2-2 所示：

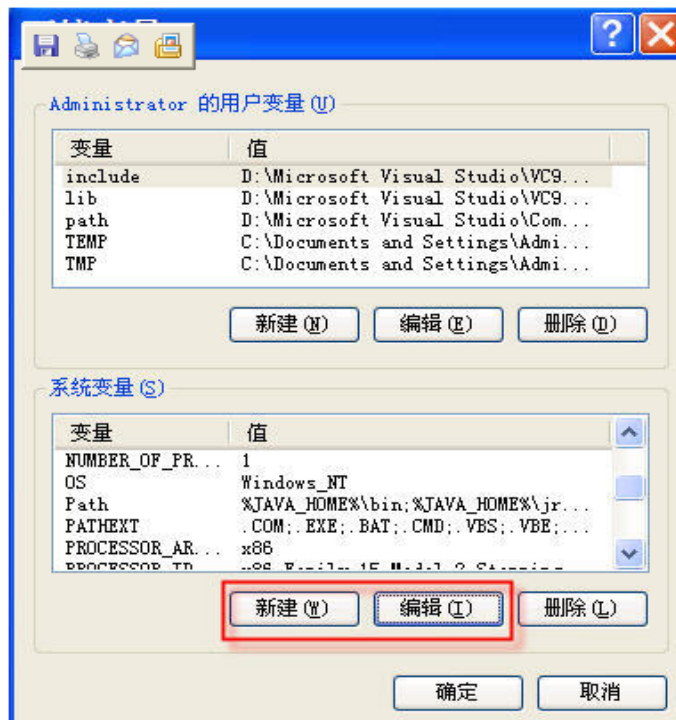


图 2-2 环境变量的设置

然后开始 - 运行 cmd, 输入 java 或者 javac 看看有没有效果, 如果出来一大堆其它的东西, 那就好, 如果出现类似 xxx 不是内部或者是外部命令之类的, 那就说明你的设置有问题, 看看那里出问题然后加以改正。

2.3.2 sdk 的配置

因为我们要开发的 android 应用程序, 所以下的 sdk 是 Google 那里下载, 地址在 http://developer.android.com/sdk/1.5_r3/index.html 以后更新后估计地址会变, 所以大家可以从 <http://developer.android.com/index.html> 进入 android 主页, 然后点击 sdk 就到了这个下载页面, 我是 XP, 当然下载了第一个 Windows 的版本。下载后解压到一个目录, 如是 D:\SDK\android-sdk-windows-1.5_r3\android-sdk-windows-1.5_r3 然后同样要设置环境变量, 同 jdk 设置, 新建一个 SDK_HOME, 值为 D:\SDK\android-sdk-windows-1.5_r3\android-sdk-windows-1.5_r3, 然后找到 path, 在前面加上 %SDK_HOME%\tools; 就 OK 了。和 jdk 一样, 我们也可以测试一下环境变量是否设置成功, cmd 窗口中输入 android, 出现一堆东西, 就完成了。

2.4 eclipse 设置

当然, 以上都搞定后, 还不能马上用 eclipse 开发 android 应用程序, 还必须添加 eclipse 插件才行, 幸好这步骤非常轻松, 打开 eclipse 后, 点击菜单栏的 help -> install new software... 然后点击 Add 会出来一个 Add Site 的窗口, 这里 Name 输入 android(自己取个就是了), location 中输入插件的网络地址:

`https://dl-ssl.google.com/android/eclipse/`

然后就等待连接了, 然后 install, 把 "Android DDMS" 和 "Android Development Tools" 都选中, finish 就可以了, 可能有段时间会去计算插件占用资源情况, 慢慢等会。完成后会提示重启 eclipse 来加载插件, ok 重启, 就可以用了。

2.5 Android 开发环境配置最终步骤

为什么要有这个步骤, 是因为运行了 example 的时候总会出现 "Failed to find an AVD compatible with target", 所以在这里写一下该步骤来提供大家的方便。

avd 必须在运行实例之前前创建好, 这个时候可以直接在 cmd 中输入 android list, 就会列出几个由 avd 创建的 api, 如果有 id1 id2 id3 分别对应 sdk1.1 sdk1.5 GoogleAPIs。

然后就运行 android create --name myname(取一个英文的名字) --target 3(如果要用 sdk1.5 就输入 2, 1.1 输出 1)

3 Android 应用程序核心

本章将会详细叙述 Android 的应用程序所包括的基本组件，还有这些组件之间的沟通方式—Intent (除了 Contentprovider), 同时讲解一下 Android 应用程序的生命周期 (Activity)—这是其他智能收与 Android 手机平台最显著不同的地方。最后将介绍在 Android 的应用程序中如何利用下载外部资源。

3.1 应用程序基础

Android 的应用程序全部都是用 Java 编程语言所写的。编译后的 Java 代码——包括应用程序要求的任何数据和资源文件，是通过 aapt 工具捆绑成一个 Android 包，归档文件是以 .apk 为后缀。这个文件是分发应用程序和安装到移动设备的中介或者是工具，用户下载这个文件到他们自己的设备上。一个 .apk 文件中的所有代码就会被认为是一个应用程序。

aapt 是 Android Asset Packaging Tool 的首写字母的缩写，这个工具包是含在 SDK 的 tools/ 目录下的。查看、创建、更新与 zip 兼容的归档文件 (zip、jar、apk)。它也能将资源文件编译成二进制包。

虽然你可能不会经常直接使用 aapt，但构建脚本 (build scripts) 和 IDE 插件会使用这个工具打包成 apk 文件，构成一个 Android 的应用程序。

下面是更详细的使用细节，首先打开一个终端，进入 tools/ 目录下，运行命令：

Linux 或 Mac 操作系统 `./aapt`，如果是 Windows: `aapt.exe`。在这里注意的是 tools/ 目录指的是 android SDK 目录下的 `/platforms/android-X/tools/`。在很多的方面，每个 Android 的应用程序是生活在它自己的世界。默认的情况下，任何一个应用程序是运行在它自己的 Linux 进程中。而当应用程序中的任何代码需要被执行时候，Android 将启动进程；当它不再需要和系统资源不会被其他应用程序所请求时，Android 就会关闭进程这个进程。每一个应用程序都有自己的 Java 虚拟机 (VM)，所以应用程序的代码是独立于其他所有的应用程序代码的运行。在默认情况下，每一个应用程序分配一个唯一的 Linux 用户 ID。权限被设置为每一个应用程序的文件仅对用户和应用程序自己本身可见。虽然有一些方法也可以暴露他们给其他应用程序。也有可能的是设置两个应用程序共享一个用户 ID，这种情况下，他们能够看到对方的文件。为了节省系统资源的占用，具有相同 ID 的应用程序最好可以安排在同一个 Linux 进程中，共享同一个 VM。

3.2 应用程序组件

Android 的一个主要特点是，一个应用程序可以利用其他应用程序的元素 (假设这些应用程序允许的话)。例如，如果你的应用程序需要显示一个图像的滚动列表，且其他应用程序已经开发了一个合适的滚动条并可以提供给别的应用程序用，你可以调用这个滚动条来工作，而不用自己开发一个。你的应用程序不用并入其他应用程序的代码或链接到它。相反，当需求产生时它只是启动其他应用程序块。

对于这个工作，当应用程序的任何部分被请求时，系统必须能够启动一个应用程序的进程，并实例化该部分的 Java 对象。因此，不像其他大多数系统的应用程序，Android 应用程序没有一个单一的入口点（例如，没有 main() 函数）。相反，系统能够实例化和运行需要几个必要的组件。有四种类型的组件：

活动 (Activities)

服务 (Services)

广播接收者 (Broadcast receivers)

内容提供者 (Content providers)

然而，并不是所有的应用程序都必须包含上面的四个部分，你的应用程序可以由上面的一个或几个来组建。当你决定使用以上哪些组件来构建 Android 应用程序时，你应该将它们列在 AndroidManifest.xml 文件中，在这个文件中你可以声明应用程序组件以及它们的特性和要求。关于 AndroidManifest.xml 在 Android 开发之旅: HelloWorld 项目的目录结构的 1.6、AndroidManifest.xml 简单介绍了一下，你可以参考一下，下篇也将介绍它。

3.2.1、活动 (Activities)

一个活动表示一个可视化的用户界面，关注一个用户从事的事件。例如，一个活动可能表示一个用户可选择的菜单项列表，或者可能显示照片连同它的标题。一个文本短信应用程序可能有一个活动，显示联系人的名单发送信息；第二个活动，写信息给选定的联系人；其他活动，重新查看旧信息或更改设置。虽然他们一起工作形成一个整体的用户界面，但是每个活动是独立于其他活动的。每一个都是作为 Activity 基类的一个子类的实现。

android.app.Activity 类：因为几乎所有的活动 (activities) 都是与用户交互的，所以 Activity 类关注创建窗口，你可以用方法 setContentView(View) 将自己的 UI 放到里面。然而活动通常以全屏的方式展示给用户，也可以以浮动窗口或嵌入在另外一个活动中。有两个方法是几乎所有的 Activity 子类都实现的：

onCreate(Bundle)：初始化你的活动 (Activity)，比如完成一些图形的绘制。最重要的是，在这个方法里你通常将用布局资源 (layout resource) 调用 setContentView(int) 方法定义你的 UI，和用 findViewById(int) 在你的 UI 中检索你需要编程地交互的小部件 (widgets)。setContentView 指定由哪个文件指定布局 (main.xml)，可以将这个界面显示出来，然后我们进行相关操作，我们的操作会被包装成为一个意图 (Intent)，然后这个意图对应有相关的 activity 进行处理。

onPause()：处理当离开你的活动时要做的事情。最重要的是，用户做的所有改变应该在这里提交（通常 ContentProvider 保存数据）。

生命周期如图 3-1

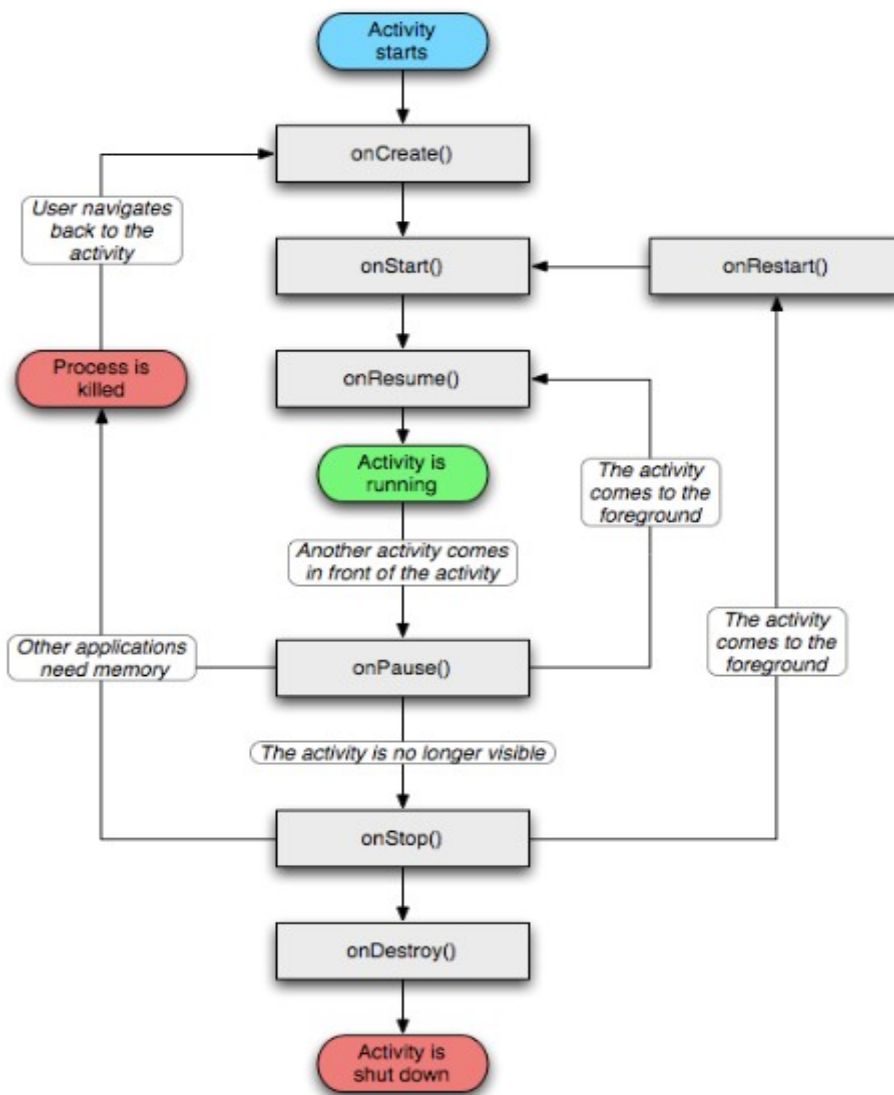


图 3-1 Android 的生命周期

每一个应用程序可能只包含一个活动，或者是像短信应用程序一样，它可能会包含几个活动。这些活动分别是什么，以及有多少，这个当然取决于它的应用程序自己的设计。根据一般情况来说，当一个应用程序被启动的时候，会被标记为第一个的活动，应该展示给用户。从一个活动转移到另外一个活动由当前的活动完成从而开始下一个。

每一个活动都会有一个默认的窗口。根据一般情况来说，窗口是会填满整个屏幕的，但是它也有可能比屏幕小或者是浮在其他窗口上，这个就是程序员代码的编写情况了。一个活动还可以使用额外窗口。比如弹出式对话框，或者是当一个用户选择屏幕上一个特定的项时，会跳出一个窗口来显示给用户重要的信息。

窗口的可视化内容是由继承自接口为 View 基类，一个分层的视图—对象提供。每个可视化控件是窗口内的

一个特定的矩形空间。父视图包含和组织子女视图的布局。叶子视图一分层的底层。绘制的矩形会直接控制和响应用户的操作。所以，一个视图活动是与用户发生交互的地方。比如说，一个视图显示一个小的图片和当用户点击图片时发起一个动作行为。Android 有一些现成的可视化控件你可以使用，其中包括按钮（buttons）、图片按钮（ImageButton）、文本域（text fields）、可编辑文本控件（EditText）、菜单项（menu items）、仿真时钟（AnalogClock）滚动条（scroll bars）、开关按钮（ToggleButton）、复选按钮（check boxes）等等。

通过 `Activity.setContentView()` 方法来设置一个视图层次在我们的活动窗口中。内容视图（content view）是层次结构的根视图对象。层次结构如下图 3-2 所示：

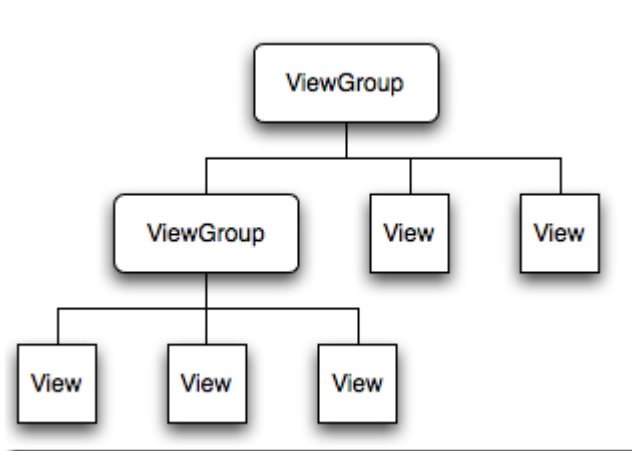


图 3-2 View 的层次结构

`Activity.setContentView()` 方法: `public void setContentView (int layoutResID):` 根据布局资源的设置活动界面。资源将会被夸大，添加布局资源文件中的所有的最高层的视图（top-level views）到活动。

3.2.2、 服务（Services）

与 Activity 相反，Service 是没有可视化的用户界面，但是 Service 的特点是能长时间在后台运行的。所以，也可以这样去理解，Service 是具有一段较长生命周期而且没有用户界面的程序。比如说一个服务可能要播放背景音乐但是用户做可以去做其他一些事情，或者它也可能从网络获取数据，也或者是计算一些东西并会提供一个结果给需要的活动（activities）。每一个服务都继承自 Service 基类。

每一个服务类都惠子在 `AndroidManifest.xml` 中有自己相应的 `<service>` 声明。服务必须可以通过 `Context.startService()` 或者 `Context.bindService()` 启动因为 service 是有两个不同的生命周期的。

一个非常典型的例子是媒体播放器播放一个播放列表中的歌曲文件。该播放器应用程序将可能有一个或者是多个活动（activities），可以允许用户选择要听的歌曲和开始播

放。然而，音乐播放本身就不会被一个活动处理，因为用户是希望保持音乐一直播放的

，当用户离开播放器的时候而去做其他事情。为了保持音乐能够继续播放，媒体播放器活动可以启动一个服务运行在后台的运行。系统将会保持音乐播放服务运行，甚至媒体播放器离开屏幕的时候。

可以连接到（绑定到）一个持续运行的服务（并启动服务，如果它尚未运行）。连接之后，你可以通过服务暴露的接口与服务交流。对于音乐服务，这个接口可以允许用户暂停、倒带、停止和重新播放。

像活动（activities）和其他组件一样，服务（services）运行在应用程序进程中的主线程中。因此，他们将不会阻止其他组件或用户界面，他们往往产生其他一些耗时的任务（如音乐播放）。

3.2.3、广播接收者（Broadcast receivers）

一个广播接收者是这样一个组件，它不做什么事，仅是接受广播公告并作出相应的反应。许多广播源自于系统代码，例如公告时区的改变、电池电量低、已采取图片、用户改变了语言偏好。应用程序也可以发起广播，例如为了他其他程序知道某些数据已经下载到设备且他们可以使用这些数据。

一个应用程序可以有任意数量的广播接收者去反应任何它认为重要的公告。所有的接受者继承自 `BroadcastReceiver` 基类。

`BroadcastReceiver` 类：是接受 `sendBroadcast()` 发送的意图（intents）的基类。可以用 `Context.registerReceiver()` 动态地注册这个类的实例，或者通过 `AndroidManifest.xml` 中 `<receiver>` 标签静态发布。注意：如果你在 `Activity.onResume()` 注册一个接受者，你应该在 `Activity.onPause()` 注销它。因为当暂停时你不会收到意图，注销它将削减不必要的系统开销。不要在 `Activity.onSaveInstanceState()` 中注销它，因为它将不会被调用，如果用户移动到先前的堆栈。

有两种主要的可接受广播类型：

正常广播（由 `Context.sendBroadcast` 发送）是完全异步的。所有的广播接收者以无序方式运行，往往在同一时间接收。这样效率较高，但是意味着接受者不能使用结果或终止广播数据传播。

有序广播（由 `Context.sendOrderedBroadcast` 发送）一次传递给一个接收者。由于每个接收者依次执行，因此它可以传播到下一个接收器，也可以完全终止传播以便他不会传递给其他接收者。接收者的运行顺序可由匹配的意图过滤器（`intent-filter`）的 `android:priority` 属性控制。

广播接收者不显示一个用户界面。然而，它们启动一个活动去响应收到的信息，或者他们可能使用 `NotificationManager` 去通知用户。通知可以使用多种方式获得用户的注意——闪烁的背光、振动设备、播放声音等等。典型的是放在一个持久的图标在状态栏，用户可以打开获取信息。

3.2.4、内容提供者（Content providers）

内容提供者（content provider）使一个应用程序的指定数据集提供给其他应用程序。这些数据可以存储在文件系统中、在一个 SQLite 数据库、或以任何其他合理的方式。内容提供者继承自 `ContentProvider` 基类并实现了一个标准的方法集，使得其他应用程序可以检索和存储数据。然而，应用程序并不直接调用这些方法。相反，替代的是它们使用一个 `ContentResolver` 对象并调用它的方法。`ContentResolver` 能与任何内容提供者通信，它与提供者合作来管理参与进来的进程间的通信。

内容提供者是 Android 应用程序的主要组成部分之一，提供内容给应用程序。他们封装数据且通过单个 `ContentResolver` 接口提供给应用程序。只有需要在多个应用程序间共享数据是才需要内容提供者。例如，通讯录数据被多个应用程序使用，且必须存储在一个内容提供者中。如果你不需要在多个应用程序间共享数据，你可以直接使用 `SQLiteDataBase`

当 `ContentResolver` 发出一个请求时，系统检查给定的 URI 的权限并传递请求给内容提供者注册。内容提供者能理解 URI 想要的东西。`UriMatcher` 类用于帮组解析 URIs。

需要实现的方法主要如下：

`query(Uri, String[], String, String[], String)` 返回数据给调用者

`insert(Uri, ContentValues)` 插入数据到内容提供者

`update(Uri, ContentValues, String, String[])` 更新内容提供者已存在的数据

`delete(Uri, String, String[])` 从内容提供者中删除数据

`getType(Uri)` 返回内容提供者中的 MIME 类型数据

更多的关于 `ContentResolver` 信息，请查看相关文档。

每当有一个应该由特定组件处理的请求，Android 可以确保该组件的应用程序正在运行，如果没有就启动它，而且一个适当的组件实例可用，如果没有就创建。

4 相关技术概念

本章节主要介绍一下此论文中用到了的 Android 的布局文件，可视化控件以及它们的具体实现。

4.1 坐标布局（AbsoluteLayout）

坐标布局（`AbsoluteLayout`）的子控件需要指定相对与此坐标布局的横、纵坐标的值，将会像 `FrameLayout` 那样被排在左上角。

项目中进入游戏的布局文件代码如下：

```

<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@android:color/black">
<ImageView
    android:id="@+id/menu"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/menu"
    android:layout_x="55px"
    android:layout_y="10px"
/>
<Button
    android:id="@+id/b1"

    android:layout_width="110px"
    android:layout_height="40px"
    android:layout_x="100px"
    android:layout_y="150px"
    android:text="新游戏"
/>
<Button
    android:id="@+id/b3"
    android:layout_width="110px"
    android:layout_height="40px"
    android:layout_x="100px"
    android:layout_y="200px"
    android:text="帮助"
/>
<Button
    android:id="@+id/b4"
    android:layout_width="110px"
    android:layout_height="40px"
    android:layout_x="100px"
    android:layout_y="250px"
    android:text="关于"
/>
<Button
    android:id="@+id/b5"
    android:layout_width="110px"
    android:layout_height="40px"
    android:layout_x="100px"
    android:layout_y="300px"
    android:text="退出"
/></AbsoluteLayout>

```

4.2 Intent

Android 中提供了 Intent 机制来协助应用间的交互与通讯，Intent 负责对应用中一次操作的动作、动作涉及数据、附加数据进行描述，Android 则根据此 Intent 的描述，负责找到对应的组件，将 Intent 传递给调用的组件，并完成组件的调用。Intent 不仅可用于应用程序之间，也可用于应用程序内部的 Activity/Service 之间的交互。因此，Intent 在这里起着媒体中介的作用，专门提供组件互相调用的相关信息，实现调用者与被调用者之间的解耦。

(1) action: 一般要执行的操作，如 ACTION_VIEW, ACTION_EDIT, ACTION_MAIN 等

(2) Data 也就是执行动作要操作的数据 Android 中采用指向数据的一个 URI 来表示，如在联系人应用中，一个指向某联系人的 URI 可能为：content://contacts/1。对于不同的动作，其 URI 数据的类型是不同的(可以设置 type 属性指定特定类型数据)，如 ACTION_EDIT 指定 Data 为文件 URI，打电话为 tel:URI，访问网络为 http:URI，而由 content provider 提供的数据则为 content: URIs。

(3) category: 提供了有关的行动来执行的其他信息。例如，CATEGORY_LAUNCHER 意味着它应该出现在作为顶层应用程序启动器，而 CATEGORY_ALTERNATIVE 意味着它应在的其他行动用户可以对一个数据列表中。

(4) type: 指定一个明确的意图的数据类型 (MIME 类型)。一般情况下，类型推断，从数据本身。通过设置这个属性，您禁用，评价并强制显式类型。

(5) component: 指定组件类使用的意图明确的名称。通常这是由看在意图 (动作，数据/类型和类别) 其它信息和匹配一个组件，它可以处理它的。如果这个属性被设置，那么评价没有执行，这部份会被用来完全一样的。通过指定这个属性，其他所有属性成为可选的意向。

(6) extras: 这是任何其他信息包。这可以用来提供扩展信息的组件。例如，如果我们有一个行动，发送一封邮件，我们也可以包括在这里的数据提供一个额外的部分主题，正文等。

下面是转载来的其他的一些 Intent 用法实例(转自 javaeye)

显示网页

1. Uri uri = Uri.parse("http://google.com");
2. Intent intent = new Intent(Intent.ACTION_VIEW, uri);
3. startActivity(intent);

显示地图

1. Uri uri = Uri.parse("geo:38.899533,-77.036476");
2. Intent intent = new Intent(Intent.ACTION_VIEW, uri);
3. startActivity(intent);
4. //其他 geo URI ?例
5. //geo:latitude,longitude;
6. //geo:latitude,longitude?z=zoom;
7. //geo:0,0?q=my+street+address;
8. //geo:0,0?q=business+near+city;
9. //google.streetview:cbll=lat,lng&cbp=1,yaw,,pitch,zoom&mz=mapZoom;

路径规划

1. Uri uri =
Uri.parse("http://maps.google.com/maps?f=d&saddr=startLat%20startLng&daddr=endLat%20endLng&hl=en");
2. Intent intent = new Intent(Intent.ACTION_VIEW, uri);
3. startActivity(intent);
4. //where startLat, startLng, endLat, endLng are a long with 6 decimals like: 50.123456

打电话

1. //叫出拨号程序
2. Uri uri = Uri.parse("tel:0280000456");
3. Intent intent = new Intent(Intent.ACTION_DIAL, uri);
4. startActivity(intent);
1. // 直接打电话出去
2. Intent intent = new Intent(Intent.ACTION_CALL, uri);
3. Uri uri = Uri.parse("tel:0280000456");

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/498130022016006074>