



eoeandroid.com

做中国最棒的Android开发社区

【eoe 特刊】第 8 期: 开发技巧



eomarket



eoemobile.com

发布版本: Ver1.0.0(build 2009.09.28)

© Copyright 2009 eoeAndroid.com. All Rights Reserved

本期封面: 摩托罗拉 最新 Android 机 CLIQ

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

eoeAndroid 做中国最棒的 Android 开发社区

本期简介

当掌握了一门开发语言以及一个平台的开发模式以后,基本上就可以进行开发了。但是写程序是门很有意思的事情,特别是现在使用Java这样的语言开发,我们又了强大的开发工具,我们又了更好的UI交互的平台,所以也诞生了很多使用的开发技巧,如果能够掌握这些小技巧无疑能为我们的开发提速不少。

本期就为大家搜罗了一些小技巧,可能数量还不是很多,各位看官各取所需吧!

同时,此期特刊由网友和特刊组成员热血完成,在本刊中将为您一一介绍,为作者和读者搭一桥梁,相信在大家支持下,特刊组将不断推出大家需求的文档。

上传 eoeMarket 让社区帮你推广

hi,所有的开发者们,你们好

eoeMarket 是社区的一个软件发布平台,凡是上传到 eoeMarket 的原创软件,经过审核,对于优秀的软件 eoeMarket 都会帮助你们进行推广,让更多的用户使用,并且得到更多的用户反馈。

如果你们觉得自己的软件好,需要推荐的话,也可以给我发邮件: haiyangjy@eoemobile.com.



《Google Android 开发入门与实战》简介

- 作者: 靳岩 姚尚朗
- 出版社: 人民邮电出版社
- 书号: 9787115209306
- 出版日期: 2009 年 7 月
- 开本: 16 开
- 页码: 400

内容特点

国内第一本原创 Android 图书

完全基于 Android 最新的 SDK1.5

全书除了大量小型案例之外还包含了 5 个 Android 平台下的完整商业实例及源码分析, 分别是 RSS 阅读器、基于 GoogleMap 的个人 GPS、豆瓣客户端、在线音乐播放器、手机信息助手

随书附赠的光盘中包含 300 分钟的详细教学视频以及 Android 开发必备的开发资源

读者对于此书内容的疑问可以访问 <http://www.eoeandroid.com> 社区, 作者团队将会及时解答

1.   

本期简介.....	2
上传 EOEMARKET 让社区帮你推广.....	2
《GOOGLE ANDROID 开发入门与实战》简介.....	3
1.最常用的 ECLIPSE 快捷键&模拟器快捷键.....	6
1.1ECLIPSE 快捷键.....	6
1.2 模拟器快捷键.....	8
2.ECLIPSE 使用技巧.....	10
2.1 密技篇:	10
2.2 外挂篇:	10
2.3 一般插件安装.....	11
2. 4 安装 MYECLIPSE.....	12
2. 5 自定义注释.....	14
2.6 查看 JDK 源代码.....	14
3.JAVA 开发小技巧.....	15
4. ANDROID 最佳实践.....	19
4.1 为性能设计:	19
4.2 为响应灵敏性设计.....	22
4.3 为无缝设计:	24
5.多资源文件的引用.....	27
6. ANDROID 调试 LOGCAT 技巧.....	29
7. 用 ANDROID 运行最简单的 C 程序.....	31
8.开发技巧杂集.....	33
8.1 一些源于 CSS 的组合实现技巧.....	33
8.2 关于 SEARCH 搜索框的使用.....	33
8.3ANDROID 是否有网络连接.....	34
8.4ACTIVITY 全屏和 无标题栏.....	35
8.5 使用 VIEWSTUB 延迟展开视图.....	35
8.6 删除窗口背景.....	35
8.7 使用主题.....	36
8.8 预先缩放图像到视图大小.....	36
8.9 使用 GETSTRING (INT RESID, OBJECT... FORMATARGS)的技巧.....	37
8.10 横竖屏切换时不重新加载 ONCREATE.....	38
8.11ANDROID TRANSLUCENTTHEME 半透明主题.....	38
8.12 模拟器玩 PING.....	38
8.13 通过 GSM CALL 命令打电话\发短信.....	39
9. 在 ANDROID 中使用 MAP 需要注意的技巧.....	40
10.每个人应该会用的四个 GOOGLE ANDROID 小技巧.....	42

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

10.1 小技巧 1: 随时使用 MPEG-4 H.264.....	42
10.2 小技巧 2: 硬检索关键字.....	42
10.3 技巧 3 使用手机摄像头拍摄商品条码消费.....	42
10.4 小技巧 4: 快捷键.....	42
总结:	43
11.GOOGLE ANDROID SDK 1.6 发布以及重大性能提升!	44
系统新功能.....	44
新平台新技术.....	46
12.EOEMARKET.....	48
A 每日一句.....	52
B. APPSHARERS.....	53
13.介绍特刊组成员.....	55
14.其他.....	56
15. 游戏诞生记 真的要生蛋了.....	57
15.1 游戏诞生记一月总结.....	57
15.2 游戏诞生记 所有资源汇集贴.....	59
16. EOEANDROID 社区为大家提供一个优质的 ANDROID 人才平台.....	60

1.最常用的 Eclipse 快捷键&模拟器快捷键

收集 by fanth

1.1Eclipse 快捷键

Ctrl+I 快速修复(最经典的快捷键,就不用多说了)

Ctrl+D: 删除当前行

Ctrl+F 查找并替换

Ctrl+Shift+K 查找上一个

Ctrl+K 查找下一个

Ctrl+Z 撤销

Ctrl+C 复制

Alt+Shift+↓ 恢复上一个选择

Ctrl+X 剪切

Alt+/ 内容辅助

Ctrl+A 全部选中

Alt+? 上下文信息

Alt+↓ 当前行和下面一行交互位置(特别实用,可以省去先剪切,再粘贴了)

Alt+↑ 当前行和上面一行交互位置(同上)

Alt+← 前一个编辑的页面

Alt+→ 下一个编辑的页面(当然是针对上面那条来说了)

Alt+Enter 显示当前选择资源(工程,or 文件 or 文件)的属性

Ctrl+Alt+↓ 复制当前行到下一行(复制增加)

Ctrl+Alt+↑ 复制当前行到上一行(复制增加)

Shift+Enter 在当前行的下一行插入空行(这时鼠标可以在当前行的任一位置,不一定是最后)

Shift+Ctrl+Enter 在当前行插入空行(原理同上条)

Ctrl+Q 定位到最后编辑的地方

Ctrl+L 定位在某行 (对于程序超过 100 的人就有福音了)

Ctrl+M 最大化当前的 Edit 或 View (再按则反之)

Ctrl+/ 注释当前行,再按则取消注释

Ctrl+O 快速显示 OutLine

Ctrl+T 快速显示当前类的继承结构

Ctrl+W 关闭当前 Editor

Ctrl+K 参照选中的 Word 快速定位到下一个

Ctrl+E 快速显示当前 Editor 的下拉列表(如果当前页面没有显示的用黑体表示)

Ctrl+/(小键盘) 折叠当前类中的所有代码

Ctrl+×(小键盘) 展开当前类中的所有代码

Ctrl+Space 代码助手完成一些代码的插入(但一般和输入法有冲突,可以修改输入法的热键,也可以暂用 Alt+/来代替)

Ctrl+Shift+E 显示管理当前打开的所有的 View 的管理器(可以选择关闭,激活等操作)

Ctrl+J 正向增量查找(按下 Ctrl+J 后,你所输入的每个字母编辑器都提供快速匹配定位到某个单词,如果没有,则在 stutes line 中显示没有找到了,查一个单词时,特别实用,这个功能 Idea 两年前就有了)

Ctrl+Shift+J 反向增量查找(和上条相同,只不过是往后往前查)

Ctrl+Shift+F4 关闭所有打开的 Editor

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

Ctrl+Shift+F6 上一个编辑器

Ctrl+Shift+F7 上一个视图

Ctrl+Shift+F8 上一个透视图

Ctrl+F6 下一个编辑器

Ctrl+F7 下一个视图

Ctrl+F8 下一个透视图

Ctrl+Shift+X 把当前选中的文本全部变味小写

Ctrl+Shift+Y 把当前选中的文本全部变为小写

Ctrl+Shift+F 格式化当前代码

Ctrl+Shift+W 切换编辑器

Ctrl+Shift+P 定位到对于的匹配符(譬如{}) (从前面定位后面时,光标要在匹配符里面,后面到前面,则反之)

下面的快捷键是重构里面常用的,本人就自己喜欢且常用的整理一下(注:一般重构的快捷键都是 Alt+Shift 开头的了)

Alt+Shift+R 重命名 (是我自己最爱用的一个了,尤其是变量和类的 Rename,比手工方法能节省很多劳动力)

Alt+Shift+M 抽取方法 (这是重构里面最常用的方法之一了,尤其是对一大堆泥团代码有用)

Alt+Shift+C 修改函数结构(比较实用,有 N 个函数调用了这个方法,修改一次搞定)

Alt+Shift+L 抽取本地变量(可以直接把一些魔法数字和字符串抽取成一个变量,尤其是多处调用的时候)

Alt+Shift+F 把 Class 中的 local 变量变为 field 变量 (比较实用的功能)

Alt+Shift+I 合并变量(可能这样说有点不妥 Inline)

Alt+Shift+V 移动函数和变量(不怎么常用)

Alt+Shift+Z 重构的后悔药(Undo)

Ctrl+= 放大

Ctrl+- 缩小

作用域 功能 快捷键

Java 编辑器 打开结构 Ctrl+F3

全局 打开类型 Ctrl+Shift+T

全局 打开类型层次结构 F4

全局 打开声明 F3

全局 打开外部 javadoc Shift+F2

全局 打开资源 Ctrl+Shift+R

全局 后退历史记录 Alt+←

全局 前进历史记录 Alt+→

全局 上一个 Ctrl+,

全局 下一个 Ctrl+.

Java 编辑器 显示大纲 Ctrl+O

全局 在层次结构中打开类型 Ctrl+Shift+H

全局 转至匹配的括号 Ctrl+Shift+P

全局 转至上一个编辑位置 Ctrl+Q

Java 编辑器 转至上一个成员 Ctrl+Shift+↑

Java 编辑器 转至下一个成员 Ctrl+Shift+↓

文本编辑器 转至行 Ctrl+L

全局 出现在文件中 Ctrl+Shift+U

全局 打开搜索对话框 Ctrl+H

全局 工作区中的声明 Ctrl+G

全局 工作区中的引用 Ctrl+Shift+G

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

文本编辑器 改写切换 Insert

文本编辑器 上滚行 Ctrl+↑

文本编辑器 下滚行 Ctrl+↓

全局 保存 Ctrl+X

Ctrl+S

全局 打印 Ctrl+P

全局 关闭 Ctrl+F4

全局 全部保存 Ctrl+Shift+S

全局 全部关闭 Ctrl+Shift+F4

全局 属性 Alt+Enter

全局 新建 Ctrl+N

全局 全部构建 Ctrl+B

Java 编辑器 格式化 Ctrl+Shift+F

Java 编辑器 取消注释 Ctrl+\

Java 编辑器 注释 Ctrl+/

Java 编辑器 添加导入 Ctrl+Shift+M

Java 编辑器 组织导入 Ctrl+Shift+O

Java 编辑器 使用 try/catch 块来包围 未设置, 太常用了, 所以在这里列出, 建议自己设置。

也可以使用 Ctrl+I 自动修正。

全局 单步返回 F7

全局 单步跳过 F6

全局 单步跳入 F5

全局 单步跳入选择 Ctrl+F5

全局 调试上次启动 F11

全局 继续 F8

全局 使用过滤器单步执行 Shift+F5

全局 添加/去除断点 Ctrl+Shift+B

全局 显示 Ctrl+D

全局 运行上次启动 Ctrl+F11

全局 运行至行 Ctrl+R

全局 执行 Ctrl+U

全局 撤销重构 Alt+Shift+Z

全局 抽取方法 Alt+Shift+M

全局 抽取局部变量 Alt+Shift+L

全局 内联 Alt+Shift+I

全局 移动 Alt+Shift+V

全局 重命名 Alt+Shift+R

全局 重做 Alt+Shift+Y

1.2 模拟器快捷键

F1/PgUp Menu key 菜单键

F2/PgDown Star key 星号键

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

F3 Call key 发送拨号键

F4 End Call key 结束通话或者说红键

Home Home key Home 键

ESC Back Key 后退键

F7 Power button 电源键

F8 Disable/Enable all networking 禁止/启用所有网络

F9 Start tracing (only with -trace) 开始跟踪

F10 Stop tracing (only with -trace) 停止跟踪

Keypad 2468 Dpad arrows 方向键

Keypad 5 Dpad center 导航中建/OK 键

Keypad 79 Rotate device skin 旋转设备外观

Keypad + Volume Up key 音量增加键

Keypad - Volume Down key 音量降低键

Keypad / Decrease onion alpha 除法

Keypad * Increase onion alpha 乘法

Ctrl-K Switch between 'unicode' and 'raw' keyboard modes 切换 unicode 和原始键盘

2.Eclipse 使用技巧

收集 by fanth

部分原文出处: <http://zhangjunhd.blog.51cto.com/113473/18295>

2.1 密技篇:

一套 Eclipse 可同时切换, 英文、繁体、简体显示:

1.首先要先安装完中文化包。

2.在桌面的快捷方式后面加上参数即可,

英文-> -nl "zh_US"

繁体-> -nl "zh_TW"

简体-> -nl "zh_CN"。

(其它语系以此类推)

像我 2.1.2 中文化后, 我在我桌面的 Eclipse 快捷方式加入参数-nl "zh_US"。

"C:\Program Files\eclipse\eclipse.exe" -n "zh_US"

接口就会变回英文语系噜。

利用 Eclipse, 在 Word 编辑文书时可不必将程序代码重新编排:

将 Eclipse 程序编辑区的程序代码整个复制下来(Ctrl+C), 直接贴(Ctrl+V)到

Word 或 WordPad 上, 您将会发现在 Word 里的程序代码格式, 跟 Eclipse

所设定的完全一样, 包括字型、缩排、关键词颜色。我曾试过 JBuilder

GEL NetBeans 使用复制贴上时, 只有缩排格式一样, 字型、颜色等都不会改变。

2.2 外挂篇:

外挂安装: 将外挂包下载回来后, 将其解压缩后, 您会发现 features、plugins 这 2 个数据夹, 将里面的东西都复制或移动到 Eclipse 的 features、plugins 数据夹内后, 重新启动 Eclipse 即可。

让 Eclipse 可以像 JBuilderX 一样使用拖拉方式建构 GUI 的外挂:

1. Jigloo SWT/Swing GUI Builder :

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

<http://cloudgarden.com/jigloo/index.html>

下载此版本: Jigloo plugin for Eclipse (using Java 1.4 or 1.5)

安装后即可由档案->新建->其它->GUI Form 选取要建构的 GUI 类型。

2. Eclipse Visual Editor Project: <http://www.eclipse.org/vep/>

3. 点选下方 Download Page, 再点选 Latest Release 0.5.0 进入下载。除了 VE-runtime-0.5.0.zip 要下载外, 以下这 2 个也要:

EMF build 1.1.1: (build page) (download zip)

GEF Build 2.1.2: (build page) (download zip)

我只测试过 Eclipse 2.1.2 版本, 使用上是 OK 的!

3.0 版本以上的使用者, 请下载:

Eclipse build I20040324:

1.0.0 Stream Integration Build I20040325 Thu, 25 Mar 2004 -- 12:09 (-0500)

1.0.0 Stream Nightly Build N20040323a Tue, 23 Mar 2004 -- 13:53 (-0500)

注意: 3.0 以上版本, 仅 build I20040324 可正常使用。

安装成功后, 即可由新建->Java->AWT 与 Swing 里选择所要建构的 GUI 类型开始进行设计。VE 必须配合着对应版本, 才能正常使用, 否则即使安装成功, 使用上仍会有问题。使用 Eclipse 来开发 JSP 程序:

外挂名称

lomboz(下载页面) http://forge.objectweb.org/project/showfiles.php?group_id=97 请选择适合自己版本的 lomboz 下载, lomboz.212.p1.zip 表示 2.1.2 版, lomboz.3m7.zip 表示 M7 版本....以此类推。

lomboz 安装以及设置教学: Eclipse 开发 JSP-教学文件

2.3 一般插件安装

其实最简单的 Eclipse 插件安装方式就是将插件包解压缩, 然后将 features 和 plugins 复制到相应的 Eclipse 目录下的 features 和 plugins 目录中。但并不提倡这样做, 显得很杂乱, 不便于管理, 所以最好是单独放在一个目录下, 用一个文件指示插件所在的位置。具体做法:

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

例如我现在安装 Lombok 插件:

①解压插件包, 将其放在 Eclipse 目录下, 注意此目录应该和 eclipse.exe 在同一层。即:

D:\Dvp\eclipse3.1.2\eclipse\eclipse.exe

D:\Dvp\eclipse3.1.2\eclipse\features

D:\Dvp\eclipse3.1.2\eclipse\plugins

D:\Dvp\eclipse3.1.2\eclipse\configuration

(D:\Dvp\eclipse3.1.2\eclipse 是我的 Eclipse 解压目录)

与上述这些目录(文件)同层:

D:\Dvp\eclipse3.1.2\eclipse\lombok (假设解压文件名为 lombok)

②新建一个目录, 取名为 links:

D:\Dvp\eclipse3.1.2\eclipse\links

③在 links 目录下新建文件 lombok.link, 写入内容:

path=D:\Dvp\eclipse3.1.2\eclipse\lombok (D:\Dvp\eclipse3.1.2\eclipse 是我的 Eclipse 解压目录)

④要注意 lombok 目录下的目录层次,

D:\Dvp\eclipse3.1.2\eclipse\lombok\eclipse\features

D:\Dvp\eclipse3.1.2\eclipse\lombok\eclipse\plugins

(D:\Dvp\eclipse3.1.2\eclipse 是我的 Eclipse 解压目录)

这里一定要注意在 lombok 下先有 eclipse 目录, 因为 lombok 解压后就有 eclipse 目录, 所以直接可用。但有的插件解压后直接是 features 和 plugins 目录, 所以你就要手动添上 eclipse 目录。

2. 4 安装 Myeclipse

fEnterpriseWorkbenchInstaller_4.1GA_E3.1+eclipse3.1 的安装流程:

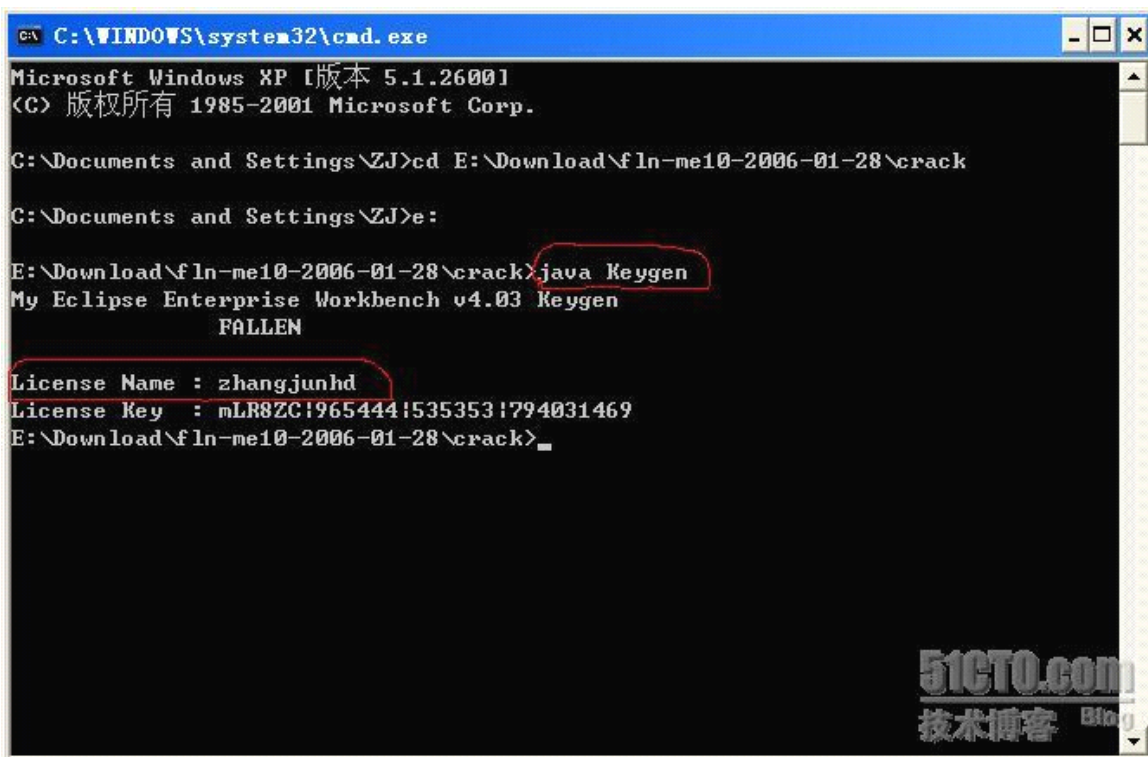
2.1 双击运行 EnterpriseWorkbenchInstaller_4.1GA_E3.1.exe

2.2 选择 eclipse3.1 的安装路径

2.3 破解

①运行当前路径下的 Keygen.class, 输入 License Name 后, 会给出 License Key。见下图:

本例中将 Keygen.class 放在 E:\Download\fln-me10-2006-01-28\crack, 所以先要转到当前目录, 注意 Keygen 中的 K 大写(常识)



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

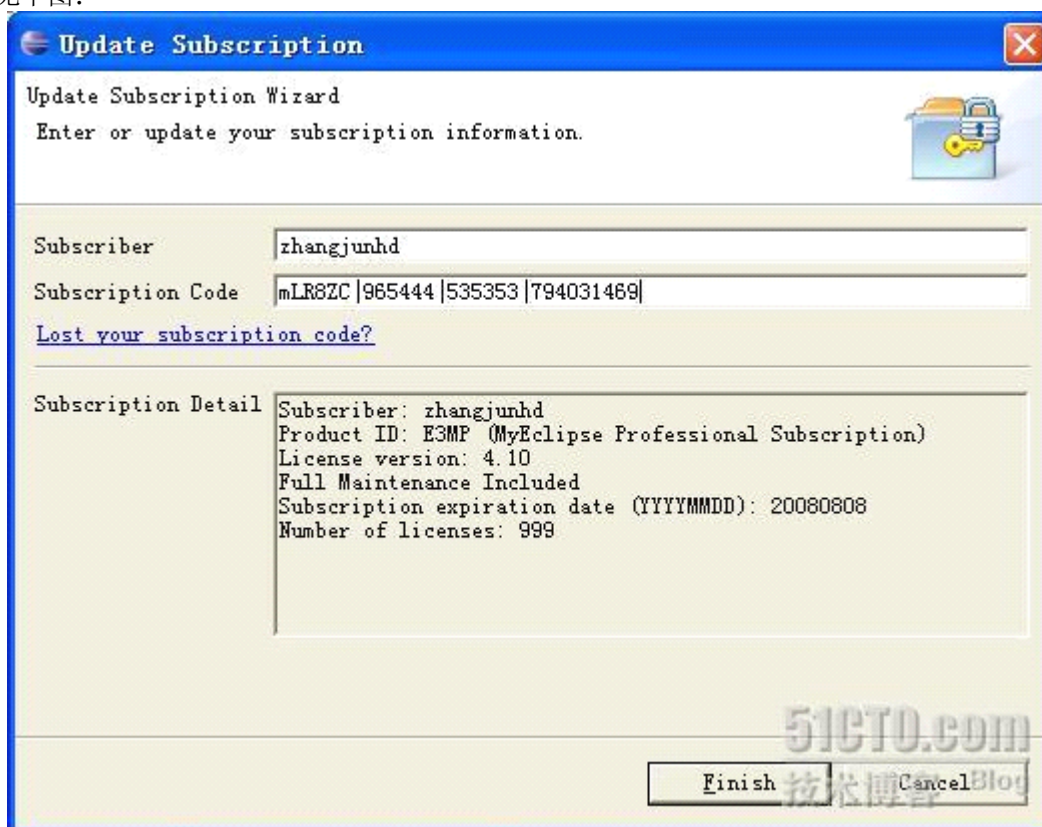
C:\Documents and Settings\ZJ>cd E:\Download\fln-me10-2006-01-28\crack

C:\Documents and Settings\ZJ>e:

E:\Download\fln-me10-2006-01-28\crack>java Keygen
My Eclipse Enterprise Workbench v4.03 Keygen
FALLEN

License Name : zhangjunhd
License Key : mLR8ZC!965444!535353!794031469
E:\Download\fln-me10-2006-01-28\crack>
```

②在 eclipse 中选择菜单 MyEclipse\Update Subscription, 输入 License Name 和 License Key。见下图:



Update Subscription

Update Subscription Wizard
Enter or update your subscription information.

Subscriber: zhangjunhd

Subscription Code: mLR8ZC|965444|535353|794031469

[Lost your subscription code?](#)

Subscription Detail:
Subscriber: zhangjunhd
Product ID: E3MP (MyEclipse Professional Subscription)
License version: 4.10
Full Maintenance Included
Subscription expiration date (YYYYMMDD): 20080808
Number of licenses: 999

Finish Cancel

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

2. 5 自定义注释

在目录 Window-Preferences-Java-Code Style 可自定义注释。例如为每个新建的 Java 文件加入作者和创建日期注释：

选择 Code Style-Code Templates-New Java files, 选择 edit, 加入

```
/*
```

*@author ZJ, \$ {date}

*

*Blog: [url]<http://zhangjunhd.blog.51cto.com/>[/url]

*

** /*

`{date}` 表示当前日期。

2.6 查看 JDK 源代码

按下 Ctrl 键不放，再单击你想看的代码名。（此时鼠标呈手型）

3.Java 开发小技巧

收集 by fanth

1.在连接字符串的时候尽量避免使用 `String = "str"+"str2"`; 而使用 `StringBuffer str = new StringBuffer("str");str.append("str2")`代替

2. 多使用 `PreparedStatement` 代替 `Statement` 这样可以避免在拼接字符串的时候出现"select * from tablename where col = '"+col+"'"单引号过多的情况 3.在拼接查询语句的时候加上"where 1=1 "道理: 当你有很多查询条件的时候, 你不知道那些个会有, 哪些个没有, 如果一个没有的话, 是不是就不用 where 子句了呢

3.如果只有一个条件的话, 是不是只要一个 where 子句就搞定了呢? 又如果有 2 个以上, 就需要用 and 来连接了。。基于上诉的情况, 勤劳的程序员们想出了一个好用的办法, 加个 where 1=1 后面不管有多少个条件有则加 and, 没有的话就什么都不用做, 相比较之前的 3 中情况的判断来说, 比较方便。这种 sql 开关代码, 可是编程中的大忌啊! 首先这么写的目的是预留开关, 给今后参数的改变带来方便, 但这样同时会造成数据库对该条语句的 explain plan 不能做到完全稳定不变, 而且也会造成代码不能被 sga ping 在内存里, 如果是频繁被调用的代码会造成很严重的性能问题, 深入的原理大家可以从 dba 方面的书籍中找到。这种开关代码在 10 来年前大行其道, 但现在随着各公司水平的提高基本都能根据自身需要实现个性化的 orm 层了, 所以早已经被证明是不好的设计。

4.多使用 `MessageFormat` 类

5.代码中尽量少出现"123".equals(str)这样的危险字符(我们公司是这样定义的)而要

`public static final String STR = "123"; STR.equals(str)`去代替.

6.方法的命名要能够表达出方法的功能

在 for 循环里要注意

```
for(int i = 0; i < 调用某个方法返回值;i++)
```

这种写法要改成

```
int y = 调用某个方法返回值;
```

```
for(int i = 0;i < y;i++)
```

(1) 使用 `Integer.valueOf()`代替 `new Integer()`;

(2) `if (result.size() > 0) return true;`

```
return false;
```

可以优化为 `return result.size()>0`

(3) 使用"`const`".equals(variable)代替 `variable.equals("const")` 避免 null point exception

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

(4) 在使用字串的拼接的时候,建议使用 `StringBuffer` 代替 `String`

(5) 在进行数据库操作的时候用 `PreparedStatement` 代替 `Statement`,可以避免因为引号过多而引起的错误

(6) 在进行复杂查询语句的拼接的时候,建议加上 `"where 1=1"`,当然在不考虑数据库的性能.

(7) `try{} catch(Exception e){} finally{if(conn!=null){conn.close();conn=null}}` 在使用数据库操作的时候尽量多用 `finally` 语句, 进行资源的释放。

(8) 在进行多异常捕获的时候, 最后建议加上 `Exception` 异常做没有考虑到的异常捕获, 比如

```
try{} catch(OtherException e){} catch(Exception e){} finally{if(conn!=null){conn.close();conn=null}}
```

(9) 给每个 `if(condition){}` 都加上大括号, 即使里面只有一句话,加强程序的可读性

(10)能用常量的东西都要用常量来完成, 避免使用硬编码,增加可维护性质。比如少用 `String str = "123"`

使用 `private static final CONST = "123" ; String str = CONST;`

(11) 定义的静态的常量用全大写, 方法名开头用小写, 类名用大写。在 `bean` 中定义的变量名用小写, 并且所有的名字

命名要体现出业务的特性。呵呵, 这里说到了规范。。。

(12) 多看看 `Apache` 下的一些 `Utils` 包吧! 方便实用, 必备工具!

(13) 可以用 `for(int i= 100;i>0;i--);`去代替 `for(int i = 0;i <100;i++)` 这样可以提高程序的运行速度

(14) 写注释有助于写出逻辑清晰的代码

(15) 用字符分隔多字符串时, 为了防止字符串中有设定的分隔符, 我采用如下字符进行分隔 (c#的, JAVA 也差不多)

```
char char2 = '\x0012'; string strreg = ""; strreg += char2;
```

(16) 一个数据有很多属性时,可以用反射取出所有属性,在制作 `HTML` 表单时,这个方法非常爽

(17) 在表单取值的时候, 多使用 `String A = B.trim();`来去空格

(18) 可以用 `for(int i= 100;i>0;i--);`去代替 `for(int i = 0;i <100;i++)` 这样可以提高程序的运行速度

(19) 把某非 `String` 类型转换成 `String` 类型的, 大多用 `.toString();` 但可以用 `String.valueOf(...);`

(20) 内部类爱好者 + 匿名内部类狂热分子

(21) 类名首字母应该大写。字段、方法以及对象(句柄)的首字母应小写。对于所有标识符, 其中包含的所有单词都应紧靠在一起, 而且大写中间单词的首字母。例如:

`ThisIsAClassName`

`thisIsMethodOrFieldName`

若在定义中出现了常数初始化字符, 则大写 `static final` 基本类型标识符中的所有字母。这样便可标志出它们属于编译期的常数。

Java 包(Package)属于一种特殊情况: 它们全都是小写字母, 即便中间的单词亦是如此。对于域名扩展名称, 如 `com`, `org`, `net` 或者 `edu` 等, 全部都应小写(这也是 `Java 1.1` 和 `Java 1.2` 的区别之一)

(22) 为了常规用途而创建一个类时, 请采取"经典形式", 并包含对下述元素的定义:

```
equals()
```

```
hashCode()
```

本文档由 `eoeAndroid` 社区组织策划, 整理及发布, 版权所有, 转载请保留!

toString()

clone() (implement Cloneable)

implement Serializable

(23) 对于自己创建的每一个类, 都考虑置入一个 main(), 其中包含了用于测试那个类的代码。为使用一个项目中的类, 我们没必要删除测试代码。若进行了任何形式的改动, 可方便地返回测试。这些代码也可作为如何使用类的一个示例使用。

(24) 应将方法设计成简要的、功能性单元, 用它描述和实现一个不连续的类接口部分。理想情况下, 方法应简明扼要。若长度很大, 可考虑通过某种方式将其分割成较短的几个方法。这样做也便于类内代码的重复使用 (有些时候, 方法必须非常大, 但它们仍应只做同样的一件事情)。

(25) 设计一个类时, 请设身处地为客户程序员考虑一下 (类的使用方法应该是非常明确的)。然后, 再设身处地管理代码的人考虑一下 (预计有可能进行哪些形式的修改, 想想用什么方法可把它们变得更简单)。

(26) 使类尽可能短小精悍, 而且只解决一个特定的问题。下面是对类设计的一些建议:

■一个复杂的开关语句: 考虑采用"多形"机制

■数量众多的方法涉及到类型差别极大的操作: 考虑用几个类来分别实现

■许多成员变量在特征上有很大的差别: 考虑使用几个类

(27) 让一切东西都尽可能地"私有"--private。可使库的某一部分"公共化" (一个方法、类或者一个字段等等), 就永远不能把它拿出。若强行拿出, 就可能破坏其他人现有的代码, 使他们不得不重新编写和设计。若只公布自己必须公布的, 就可放心大胆地改变其他任何东西。在多线程环境中, 隐私是特别重要的一个因素--只有 private 字段才能在非同步使用的情况下受到保护。

(28) 谨防"巨大对象综合症"。对一些习惯于顺序编程思维、且初涉 OOP 领域的新手, 往往喜欢先写一个顺序执行的程序, 再把它嵌入一个或两个巨大的对象里。根据编程原理, 对象表达的应该是应用程序的概念, 而非应用程序本身。

(29) 若不得已进行一些不太雅观的编程, 至少应该把那些代码置于一个类的内部。

(30) 任何时候只要发现类与类之间结合得非常紧密, 就需要考虑是否采用内部类, 从而改善编码及维护工作

(31) 尽可能细致地加上注释, 并用 javadoc 注释文档语法生成自己的程序文档。

(32) 避免使用"魔术数字", 这些数字很难与代码很好地配合。如以后需要修改它, 无疑会成为一场噩梦, 因为根本不知道"100"到底是指"数组大小"还是"其他 全然不同的东西"。所以, 我们应创建一个常数, 并为其使用具有说服力的描述性名称, 并在整个程序中都采用常数标识符。这样可使程序更易理解以及更易维护。

(33) 涉及构建器和异常的时候, 通常希望重新丢弃在构建器中捕获的任何异常--如果它造成了那个对象的创建失败。这样一来, 调用者就不会以为那个对象已正确地创建, 从而盲目地继续。

(34) 当客户程序员用完对象以后, 若你的类要求进行任何清除工作, 可考虑将清除代码置于一个良好定义的方法里, 采用类似于 cleanup() 这样的名字, 明确表明自己的用途。除此以外, 可在类内放置一个 boolean (布尔) 标记, 指出对象是否已被清除。在类的 finalize() 方法里, 请确定对象已被清除, 并已丢弃了从 RuntimeException 继承的一个类 (如果还没有的话), 从而指出一个编程错误。在采取象这样的方案之前, 请确定 finalize() 能够在自己的系统中工作 (可能需要调用 System.runFinalizersOnExit(true), 从而确保这一行为)。

(35) 在一个特定的作用域内, 若一个对象必须清除 (非由垃圾收集机制处理), 请采用下述方法: 初始化对象; 若成功, 则立即进入一个含有 finally 从句的 try 块, 开始清除工作。

1、去掉一个字符串数组中的重复项: (2006.04.11.)

原来采用的方法:

```
1 String[] rid = request.getParameterValues("noattRid");
2 //需要用一个临时的变量把过滤后的结果这个存起来
```

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

```

3 String[] ridFiltered = new String[rid.length];
4     int index = 0;
5     for (int i = 0; i < rid.length; i++) {
6         if (!this.isExistString(ridFiltered, rid[i])&&!rid[i].equals("")) {
7             ridFiltered[index++] = rid[i];
8         }
9     }
10
11 //还需要用一个方法来判断在结果中是否存在此项，如下
12 /**
13  * 用于查找某个字符串是否在一个字符串数组中
14  * @param dest String[]
15  * @param str String
16  * return boolean
17  */
18
19 public boolean isExistString(String[] dest, String str) {
20     boolean flag = false;
21     for (int i = 0; i < dest.length; i++) {
22         if (str.equals(dest[i])) {
23             flag = true;
24         }
25     }
26     return flag;
27 }

```

看看，要多麻烦有多麻烦，来看看新方法：

```

1 String[] s = request.getParameterValues("noattRid");
2 List list = Arrays.asList(s);
3 Set set = new HashSet(list);
4 //rid=(String [])set.toArray();
5 String[] temp = new String[set.size()];
6 Iterator ite = set.iterator();
7 for (int i = 0; ite.hasNext(); i++) {
8     temp[i] = (String) ite.next();
9 }

```

简简单单的几行代码即可搞定，无论是从程序的可读性、优雅性还是效率方面都有很好的提升。
 扩展：你同时还可以运用上面的方法还判断三个字符串是否相等（可进一步扩展至多个，但是似乎不太有实际意义），例程如下：

```

1 String[] s = {"one", "two", "two"};
2 List list = Arrays.asList(s);
3 Set set = new HashSet(list);
4 System.out.println(list.size() == set.size()); // false

```

本文档由 eoeAndroid 社区组织策划，整理及发布，版权所有，转载请保留！

4. Android 最佳实践

原文地址: <http://android.group.javaeye.com/group/blog/362954>

4.1 为性能设计:

1) 避免创建对象

对象的创建从来不是免费的。虽然 GC 使得内存申请代价不再高昂,但是申请总是比不申请来得昂贵。如果你在一个用户接口循环中申请对象,你将会强行执行周期性的 GC,在用户体验上出现一些小的“打嗝”,因此除非不得已,你应该避免创建对象实例,下面是一些例子可以帮助理解:当你在一组输入数据中抽取字符串时,尝试返回源数据的子串,而非创建一个副本。你将会创建一个新的 String 对象,但是它会和数据共享字符数组 char[]。

如果你有一个返回 String 的方法,而且你知道它的结果将会一直被追加到 StringBuffer,改变你的签名和实现,使这个函数里面直接追加,避免创建临时对象。

一个更激进的主意是将多维数组切成与之平行的一维数组:

一个 int 数组比 Integer 数组要好,但也有一个公认的事实就是两个平行的 int 数组要比一个 (int,int)对象数组要高效很多。对于其它原始数据类型亦如是。

如果你需要实现一个存储一组对象 (Foo,Bar) 的容器,请记住两个平等的 Foo[] 和 Bar[] 数组通常元比一个定制对象数组要好(当然,对于此有个例外,就是当你设计一个 API 供其它代码访问时;在那样的情况下,通常最好是为保证 API 的正确性而牺牲一点速度。但是在你的内部代码,你应该尽可能保持高效)。

通常来说,避免创建临时对象,如果你可以的话。更少的对象创建意味着更小频率的 GC,这对用户体验有直接的影响。

2) 用 Native 方法

当处理字符串时,要毫不犹豫地使用诸如 String.indexOf()、String.lastIndexOf() 之类的专门方法,这些是典型的用 C/C++ 代码实现的方法,它们可以轻易地比实现同样功能的 Java 循环快 10-100 倍。

对此建议的反面是调用一个 native 方法要比调用一个解析的方法,不要将 native 方法用于琐碎的计算,如果可以避免的话。

优先使用 Virtual 而非 Interface

假如你有一个 HashMap 对象,你可以声明它为一个 HashMap 或一个通用的 Map:

```
Map myMap1 = new HashMap();
```

```
HashMap myMap2 = new HashMap();
```

哪一个更好?

一般的会说你该选择 Map,因为它允许你改变其实现,对于通常的编程来说这是对的,但是对于嵌入式系统来说这并不是太妙。通过接口的引用来调用一个方法要比通过一个具体类型的引用调用 virtual 方法花多 2 倍的时间。

如果你已经选择了一个 HashMap,因为它正好适用你正在做的事情,那通过 Map 来调用就没有什么价值了。考虑到 IDE 可以为你重构代码,用 Map 来调用就没有太大价值了,即使你不知道你代码将去向何方(但是,再一次的,公共的 API 是又是一个例外:好的 API 较少考虑性能)。

3) 优先选择 static 而非 virtual

如果你不必访问一个对象的字段,使你的方法成为 static 方法。它可以被更快地调用,因为它不需要一个虚拟方法表间接调用。同时它也是一个好的做法,因为从方法的签名可以看出调用这个方法不会改变对象的状态。

4) 避免内部的 Getter/Setter

在一些像 C++ 的语言中,通常的做法是用 getter (如: i=getCount()) 代替直接地访问字段
本文档由 eoeAndroid 社区组织策划,整理及发布,版权所有,转载请保留!

(i=mCount)，在 C++ 这是一个很好的习惯，因为编译器通常能够内联这个访问，并且你需要限制或 debug 字段访问，你可以在任何时候增加代码。

在 Android，这是一个坏主意。虚拟方法调用代价是昂贵的，实例字段查找代价更高。沿用面一般向对象编程实践在公开接口中提供 getter 和 setter 是合理的，但在一个类中你应该直接访问字段。

Cache 字段查找

访问对象字段要比访问本地变量慢得多，如下面这段：

```
for (int i = 0; i < this.mCount; i++)
    dumpItem(this.mItems[i]);
```

应该写成这样：

```
int count = this.mCount;
Item[] items = this.mItems;
for (int i = 0; i < count; i++)
    dumpItems(items[i]);
```

（我们用一个显式的“this”来表明这是一个成员变量。）

有一个相似的指引就是，不要在 for 语句中的第二个从句中调用方法。例如下面这段代码将会在每次迭代中都会执行一次 getCount()，这是一个巨大的浪费，你可以将它的值 cache 为一个 int。

```
for (int i = 0; i < this.getCount(); i++)
```

```
    dumpItems(this.getItem(i));
```

通常，如果你将要访问一个实例字段多次，一个好的习惯就是创建一个临时变量。例如：

```
protected void drawHorizontalScrollBar(Canvas canvas, int width, int height) {
    if (isHorizontalScrollBarEnabled()) {
        int size = mScrollBar.getSize(false);
        if (size <= 0) {
            size = mScrollBarSize;
        }
        mScrollBar.setBounds(0, height - size, width, height);
        mScrollBar.setParams( computeHorizontalScrollRange(),
                               computeHorizontalScrollOffset(),
                               computeHorizontalScrollExtent(), false);
        mScrollBar.draw(canvas);
    }
}
```

这是对成员字段 mScrollBar 的四次分开查找，通过将 mScrollBar 缓存到本地变量，四次成员字段查找变成四次本地变量引用，这样更为高效。

同样地，方法参数作为本地变量拥有相同的性能特征。

声明常量为 final

考虑在类开头的如下声明：

```
static int intVal = 42;
static String strVal = "Hello, world!";
```

编译器产生一个叫<clinit>的类初始化器方法，它在类首次使用时被执行。这个方法将 42 存到 intVal，并为 intVal 从类文件字符串常量表中抽出一个引用，当这些值在后面被引用到时，它们以字段查找的方式被访问。

我们可以用 final 关键字改进之：

```
static final int intVal = 42;
static final String strVal = "Hello, world!";
```

这个类不再需要一个<clinit>方法，因为常量存到直接由 VM 处理的类文件静态字段初始化器，代码访问 intVal 将会直接使用 integer 值 42，而对 intVal 的访问会用一个相对廉价的“字符串常量”指令来代替一个字段查找。

声明一个方法或类为 final 并不能直接获得性能上的好处，但它确实能起到某些优化作用。例如，假如编译器知道一个“getter”不能被一个子类重写，它能够内联这个方法调用。

你也可以将本地变量声明为 final，然而这并无真正意义上的性能提升。对于要地变量，只有

本文档由 eoeAndroid 社区组织策划，整理及发布，版权所有，转载请保留！

在使代码更清晰(或你不得不, 如为了在匿名内部类中使用)时使用 `final`。

小心使用增强的 For 循环语句

增强的 For 语句可以用于实现了 `Iterable` 接口的 `Collection`, 对于这些对象, 一个 `iterator` 被申请用来进行接口调用 `hasNext()` 和 `next()`。对于 `ArrayList`, 你最好直接遍历它, 但对于其它 `collections`, 增强的 For 循环语句将会等同于显式的迭代用法。

尽管如此, 下面的代码展示了增强的 For 语句的可为人接受的用法:

```
public class Foo {
    int mSplat;
    static Foo mArray[] = new Foo[27];
    public static void zero() {
        int sum = 0;
        for (int i = 0; i < mArray.length; i++) {
            sum += mArray[i].mSplat;
        }
    }
    public static void one() {
        int sum = 0;
        Foo[] localArray = mArray;
        int len = localArray.length;
        for (int i = 0; i < len; i++) {
            sum += localArray[i].mSplat;
        }
    }
    public static void two() {
        int sum = 0;
        for (Foo a: mArray) {
            sum += a.mSplat;
        }
    }
}
```

`zero()` 在循环中每次迭代获取静态字段两次计算数组长度一次。

`one()` 将所有东西存到本地变量, 避免查找。

`two()` 用到了增强的 For 循环语句, 由编译器产生的代码拷贝数组引用和数组长度到本地变量, 使之成为一个遍历数组元素的一个很好的选择。它确实为主循环中产生了一个额外的本地载入/存储, 使得它比起 `one()` 有点慢并且长了 4 bytes。

总之, 增强的 For 语句对于数组表现良好, 但对 `iterable` 对象要小心使用, 因为有额外的对象创建。

避免 Enum 类型

Enum 非常方便, 但不幸的是当考虑到时间和速度时就让人痛苦。例如这个:

```
public class Foo {
    public enum Shrubbery { GROUND, CRAWLING, HANGING }
}
```

将编译成一个 900byte 的 .class 文件, 在首次使用时, 类初始化器在代表每个被枚举的值对象上激活 `<init>` 方法。每个对象都有其静态字段, 并且整个集合就存储在一个数组 (一个称为 “\$values” 的静态字段) 上, 对于仅仅的三个 `integer` 来说, 那是太多的代码和数据了。

这个:

```
Shrubbery shrub = Shrubbery.GROUND;
```

导致了一次静态字段查找。如果 “GROUND” 是一个 `static final int` 编译器将会将它看作是一个常量并内联它。

相反地, 当然, 是运用 `enum` 你可以得到一个更优雅的 API 和一些编译时的值检查。因此, 通常折衷的办法是: 为 API, 你应该千方百计地使用 `enum`, 但是当考虑到性能时尝试避免使用它。

利用内部类使用包作用方域

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

考虑下面的类定义:

```
public class Foo {
    private int mValue;
    public void run() {
        Inner in = new Inner();
        mValue = 27;
        in.stuff();
    }
    private void doStuff(int value) {
        System.out.println("Value is " + value);
    }
    private class Inner {
        void stuff() {
            Foo.this.doStuff(Foo.this.mValue);
        }
    }
}
```

在这里我们要特别指出的是这里定义了一个内部类（`Foo$Inner`），它可以直接访问外部类的私有方法和私有实例字段，这是合法的，代码的执行的结果是如预期般的“Value is 27”。

问题在于，`Foo$Inner` 是一个完全独立的类，这使得直接访问其私有方法是非法的，为了架起桥梁，编译器会产生如下两个虚拟方法

```
/*package*/ static int Foo.access$100(Foo foo) {
    return foo.mValue;
}
/*package*/ static void Foo.access$200(Foo foo, int value) {
    foo.doStuff(value);
}
```

当内部类代码需要访问外部类的 `mValue` 变量或激活 `doStuff` 方法时就会调用这些方法。这就意味着上面的代码清楚表明了你是通过访问器来访问成员字段的，而非直接访问。前面我们讨论过访问器是比直接访问是要慢的，所以这是一个由于某种特定语言方言所导致的隐性性能打击。

我们可以通过声明由内部类访问的字段和方法为具有包作用域而非私有作用域来解决这个问题。这样运行得更快并且移除了额外产生的方法（不幸的是，这也意味着 这些字段可以被同包下的其它类所访问，这个是违反了使所有的字段成为私有的标准 OO 做法的，再一次的，如果你是在设计一个公共的 API 的话，你可能要慎重地考虑这一优化策略）。

9) 避免使用 Float 类型

在 Pentium CPU 发布之前，对于游戏作者来说做很多整型计算是很正常的事。有了 Pentium 之后，浮点计算联合处理器成了内置功能，你的游戏通过交错整型和浮点操作比只有整型计算时运行起来要快得多。在桌面系统上通常的可以自由的使用浮点数。

不幸的是，嵌入式处理器很少具有硬件浮点支持，所以所有的“float”和“double”操作都是在软件上进行。某些基本的浮点操作可能会花费数微秒。

还有，甚至对于整型数，一些芯片支持硬件乘法但缺少硬件除法，在这种情况下，整型除法和取模运算是在软件上执行的——如果你是在设计一个哈希表或做很多数学运算这就是你需要考虑的事情。

4.2 为响应灵敏性设计

代码可能通过各种性能测试,但是当用户使用时还是会需要漫长的等待,这些就是那种响应不够灵敏的应用——它们反应迟钝,挂起或冻住周期很长,或者要花很长时间来处理输入。

在 Android 上,系统通过向用户显示一个称为应用无响应(ANR:Application Not Responding)的对话框来防止在一段时间内响应不够快。用户可以选择让应用继续,但是用户并不会想要每次都来处理这个对话框。因此应把你的应用设计得响应灵敏,使得系统不必显示 ANR 给用户。

通常地,当不能响应用户输入时系统显示一个 ANR。例如,如果一个应用在 IO 操作(经常是网络访问)上阻塞了,那么主应用线程就会无法处理正在进行的用户输入事件。经过一段时间,系统认为应用已经挂起,向用户显示一个 ANR,让用户可以选择关闭。

相同地,如果你的应用花太多的时间在构建详细的内存结构上,又或者在计算游戏的下一个动作上,系统会认为你的应用已经挂起。用上面的技术来保证这些计算是高效的一直都是很重要的,但是即使是最高效的代码运行也是需要花费时间的。

在这两种情况下,解决的办法通常就是创建一个子线程,在这个子线程上做你的大部分工作。这样让你的主线程(驱动用户接口事件循环)保持运行,并让你的代码免于被系统认为已经冻住。因为这样的线程化通常都是在类级别上来完成的,所以你可以认为响应性能问题是一个类问题(与上面描述的方法级别的性能问题)。

这个文档讨论了 Android 系统是如何决定一个应用没有响应的,并提供了指引来保障你的应用是响应灵敏的。

1) 是什么引发了 ANR?

在 Android 系统上,应用的响应灵敏性由 Activity Manager 和 Window Manager system services 所监控,当它监测到如下的其中一个条件时,Android 就会为特定的应用显示一个 ANR:

5 秒内对输入事件无响应。

一个 BroadcastReceiver 在 10 秒内没有执行完毕。

怎样避免 ANR?

考虑到上面对 ANR 的定义,让我们来研究一下这是为什么会发生以及怎样最好的组织你的应用以避免 ANR。

Android 应用正常是运行在一个单独的(如 main)线程中的,这就意味着在你应用主线程中正在做的需要花很长时间来完成的事情都能够激活 ANR 对话框。因为你的应用并没有给自己一个机会来处理输入事件或 Intent 广播。

因此任何运行在主线程中的方法应该做尽可能少的事情。特别地 Activity 在关键生命周期方法中如 onCreate()和 onResume()应当做尽可能少的设置。潜在地的耗时的操作(如网络或数据库操作,或高耗费数学计算如改变位图大小)应该在子线程里面完成(或以数据库操作为例,可以通过异步请求)。尽管如此,这并不是说当等待子线程完成的过程中你的主线程必须被阻塞——你不必调用 Thread.wait()或 Thread.sleep(),恰恰相反,你的主线程应该为子线程提供一个 Handler,以便子线程完成时可以提交回给主线程。以这种方式来设计你的应用,将会允许你的主线程一直可以响应输入,以避免由 5 秒钟的输入事件超时导致的 ANR 对话框。这些做法同样应该被其它任何显示 UI 的线程所效仿,因为它们属于同样类型的超时。

IntentReceiver 执行时间的特定限制限制了它们应该做什么:在后台执行的一些琐碎的工作如保存设置或注册通知。至于其它在主线程里被调用的方法,在 BroadcastReceiver 中,应用应该避免潜在的长耗时操作或计算,而是应该用子线程来完成密集任务(因为 BroadcastReceiver 的生命周期是短暂的)。对 Intent broadcast 作出响应,你的应用应该启动一个 Service 来执行长耗时的动作。同样,你也应该避免从 Intent Receiver 中启动 Activity,因为它会产生一个新的屏,偷走任何用户正在运行的应用的焦点。对 Intent broadcast 作出的响应,假如你的应用需要向用户显示什么东西,应该用 Notification Manager 来完成。

增强响应灵敏性

通常,在一个应用中,100 到 200 微秒是一个让用户感觉到阻滞的阈值,因此这里有些小技巧让你用来使你的应用看起来响应更灵敏。

如果你的应用正在后台对用户输入作出响应,显示正在进行的进度(ProgressBar 和

本文档由 eoeAndroid 社区组织策划,整理及发布,版权所有,转载请保留!

ProgressDialog 对此很有用)。

特别是对于游戏,在子线程中做移动的计算。

如果你的应用有一个耗时的初始化过程,考虑用闪屏或尽可能快地渲染主界面并异步地填充信息。在这两种情况下你都应该表明进度正在进行,以免用户觉得你的应用被冻住了。

4.3 为无缝设计:

即使你的应用是快速且响应灵敏的,一些设计仍然能对用户造成问题——因为与其它应用未计划的交互或者对话,意外的数据丢失,无意识的阻塞等等。为了避免这些问题,有助于理解你的应用运行的环境和可以影响你的应用的系统交互。总之,你应该竭尽全力地开发一个与系统和其它应用无缝交互的应用。

一个常见的无缝问题就是一个应用的后台进程(如 service 或 broadcast receiver)对某事件作出响应而弹出对话框,这看起来仿佛并无大碍,特别是当你在模拟器上单独地构建和测试你的应用时。然而,当你的应用在真正的设备上运行,后台线程显示对话框时,你的应用当时可能没有获得用户焦点。这就会出现你的应用会在活动的应用后面显示对话框,或者从当前应用中获得焦点并显示对话框的情况,而管论当时用户正在做什么(如正在打电话等)。那样的行为可能对你的应用或用户不起作用。

为了避免这些问题,你的应用应该利用适当的系统资源——Notification 类,来通知用户。利用通知,你的应用可以通过在状态条上显示一个图标来通知用户事件已经发生,而非获得焦点和打断用户。

另外一个无缝问题的例子就是,Activity 由于未能正确实现 onPause()及其它生命周期方法而无意中丢失了状态或用户数据。又或者,你的应用要暴露供其它应用使用的数据,你应该通过 ContentProvider 来实现,而非通过一个全世界都可读的原始文件或数据库。

这些例子的共同特点就是,它们都是关于如何跟系统和其它应用协作得更好,Android 系统的设计就是将所有的应用看作是一个松散耦合的组件联邦,不是一堆墨盒代码。这就使你作为一个开发者可以将整个系统视为只是一个更大一点的组件联邦。这样有得于你将应用与其它应用清晰和无缝的集成,所以作为回报,你应该更好的设计你的代码。

这个文档讨论了常见的无缝集成问题和怎样去避免它们。它包含了如下主题:
别丢弃数据

一定要记住 Android 是一个移动平台。看起来很显然,但是记住这个事实很重要,就是任何 Activity(如"正在打进来的电话"应用)在任何时候都有可能弹出来覆盖你的 Activity,这会激活 onSaveInstanceState()和 onPause()方法,并导致你的应用被杀死。如果当其它 Activity 出现时,用户正好在你的应用上编辑数据,你的应用被杀死的同时那些数据也很可能会丢失。当然了,除非你先保存了进行中的工作。“Android 方式”是这么做的:能接收和编辑用户输入的应用需要重写 onSaveInstanceState()方法并以恰当的方式保存它们的状态,当用户重新访问应用时,就能重新获得数据。

一个运用这个行为经典的例子就是邮件应用,当用户正在撰写邮件时另一人 Activity 开始了,应用应该将正在编辑的邮件保存为草稿。

不要暴露原始数据

如果你不想穿着内衣在大街上散步,同样你的数据也不应如此。尽管可能暴露某些应用可以方便其它应用读取,但这通常不是最好的主意。暴露原始数据要求其它的应用能够理解你的数据格式;如果你改变了格式,你将会破坏其它没有同时更新的应用。

本文档由 eoeAndroid 社区组织策划,整理及发布,版权所有,转载请保留!

“Android 方式”就是创建一个 `ContentProvider` 通过一个清晰的、深思熟虑的、可维护的 API 来暴露你的数据给其它应用。使用 `ContentProvider` 就像一个 Java 接口来分离和组件化两段紧密耦合的代码,这就意味着你能够修改你数据的内部格式而不用修改由 `ContentProvider` 暴露的接口,这样就不会影响其它应用。

别打断用户

如果能确定一个用户是带有目的性的运行一个应用才是安全的。那就是为什么你除非是直接响应当前活动的用户输入,不然就要避免产生 `Activity` 的原因。

那就是说,不要从后台运行的 `BroadcastReceiver` 和 `Service` 中调用 `startActivity()`。如果这样做将会打断任何正在运行的应用,并使用户恼怒。甚至你的 `Activity` 可能成为一个“击键强盗”接收一些用户正在为上一个 `Activity` 提供的输入,视乎你的应用所做的,这是这可能是个坏消息。取代直接从后台直接产生 `Activity` UIs,你应该用 `NotificationManager` 来设置通知,这将会出现在状态条上,当用户空闲时可以点击它们来看你的应用向他们显示了什么。

(注意,当你的 `Activity` 已经在前台时所有这些都没适用:这时,对于输入的响应,用户期望看到你的下一个 `Activity`。)

有太多事要做? 在一个线程里做

如果你的应用需要做一个代价高昂或长耗时的计算,你可能要将它移到一个线程里。这个将会防止显示“`Application Not Responding`”对话框给用户,最终导致你的应用完全终止。

默认地,在一个 `Activity` 中的代码和其所有的 `View` 运行在同一个线程上。这与处理 UI 事件的线程是同一个。例如,当一个键被按下时,一个 `key-down` 事件被添加到 `Activity` 主线程的队列。事件处理系统需要很快地让这个事件出列并处理这个事件。不然,系统数秒后将会认为应用已经挂起并替用户杀死这个应用。

如果你有长耗时的代码,让它在你的 `Activity` 上内联运行将会在使它运行在事件处理线程上,这很大程度上阻塞了事件处理句柄。这会延缓输入处理并导致 ANR 对话框。为了避免之,将你的计算移到一个线程中。在为响应灵敏性设计中已经讨论了如何做。

5) 不要过载一个单一的 `Activity` 屏

任何值得使用的应用都可能会有几个不同的屏幕。当设计你的 UI 屏幕时,请一定要运用多个 `Activity` 对象实例。依赖于你的开发背景,你可能像解释某些类似 Java Applet 的东西一样来解释一个 `Activity`, `Activity` 是你应用的入口点。然而,那并不是准确:一个 Applet 的子类是一个 Java Applet 的单一入口点,而一个 `Activity` 应该被看作一个潜在的进入你的应用的多个入口点。在你的“main”`Activity` 和任何其它你可能有的 `Activity` 之间的唯一不同就是,那“main”`Activity` 碰巧是那个唯一在你的 `AndroidManifest.xml` 文件中对“`android.intent.action.MAIN`”动作有兴趣的一个而已。所以,当设计你的应用时,把你的应用看成一个 `Activity` 对象的联邦。从长远来看,这会使得你的代码更具可维护性。

6) 扩展系统主题

当提到用户接口的观感时,协调是很重要的。用户为那些与他们所期望的用户接口相反的应用所震动。当设计你的 UI 时,你应当尽量避免出现太多你自己主题,相反地,用同一个主题。你可以重写或扩展那部分你必须的主题,但是至少你是基于与其他应用相同的 UI 基础上的。详细可以参阅“应用风格和主题”部分。

7) 设计你的 UI 可以与多屏分辨率一起工作

不同的基于 Android 的设备可能会支持不同的分辨率。甚至一些可能支持随时更改分辨率。保证你的布局和图片足够灵活对于在不同设备屏幕上正常显示是非常重要的。

幸运的是,这是很容易办到的。简单讲,你需要做的就是为你的关键分辨率提供不同版本的作品,然后设计你的布局适应各种不同的维度。(例如,避免作用硬编码位置而用相对布局。)如果那样做的话,剩下的系统会处理,你的应用在任何设备上看起来都很棒。

8) 假定网络是很慢的

Android 设备会有多种网络链接选项。所有的都会提供数据访问,虽然有一些会比另一些更快。其中速度最慢的就是 GPRS (GSM 网络的非 3G 数据服务)。即便具备 3G 能力的设备在非 3G 网络上也会花很多的时间,所以网速低将会是一个长期存在的事实。

那就是为什么你应该针对最小化的网络访问和带宽编写你的应用。你不能假设网络是快速

本文档由 eoeAndroid 社区组织策划,整理及发布,版权所有,转载请保留!

的, 所以你应该一直计划它是慢的。如果你的用户碰巧是在一个快速的网络上, 那很好 —— 他们的体验只会提升。你要避免相反的情况: 应用有时可用, 但有时慢得令人沮丧, 得看用户是在哪在什么时间, 这样的应用可能不会受欢迎。

别假定触摸屏和键盘

Android 可能支持多种外观形状。那就是说一些 Android 设备将会有完整的“QWERTY”键盘, 而其它的可能会有 40 键、12 键或其它键盘设置。同样地, 一些设备会有触摸屏, 但很多会没有。当构建你的应用时, 一定要记住, 不要假定特定的键盘布局——当然了, 除非你真的喜欢限制你的应用以到它只能在某些设备上运行。

一定要节省设备电池

如果移动设备经常局限于屋内, 那就不是很“移动”。移动设备是电池供电的, 而我们如果能让电池每充一次的电量使用得更持久一些, 每个人都会更开心 —— 特别是用户。其中的两个用电大户就是处理器和音频, 那就是为什么你写的应用应尽量做少的工作的同时尽可能频繁地使用网络的原因。

最小化你的应用使用的处理器时间就归结为书写高效的代码。从音频上最小化功耗, 要确保优雅地处理错误条件, 并仅获取你需要的东西。例如, 如果连接网络失败不要一直重试连接网络, 如果失败了一次, 很可能是用户没有接收信号。如果你立即重试, 那么你所做的一切只是在浪费电池能量。

用户是相当聪明的: 如果你的程序是高耗电的, 你可以相信他们会发觉的。在那一点上, 唯一可以确定的是你的程序将不会保持安装非常久。

5.多资源文件的引用

在实际开发中,我们可能会在初始化的时候导入多个相类似的资源文件,如。button0.png、button1.png、button2.png...这样文件我们是 getResource().getDrawable(R.drawable.button0), getResource().getDrawable(R.drawable.button1)...这样子明显不是我们想要的,其实我们可以如下来做:

```
for(int i=0;i<length;i++){
```

//注意: R.drawable.button0 不是 R.drawable.button 哦,还有,资源文件名在 Android 中不能有大写英文字母哦。

```
buttonDrawable[i]=getResource().getDrawable(R.drawable.button0+i);
```

```
}
```

上面的代码,在我们 findViewById()遇到类似的也要像大上面的方法。

字符串移动。

在实际开发中,也许会遇到当单行文字过多要使其显示字符串滚动,你可以在 TextView 的 scrollTo(int a,int b)和 scrollBy(int a,int b)方法组合 Handler 来实现。

RelativeLayout 和 LinearLayout 精美组合实现布局。

在实际开发中,一定会遇到很多样式的组合,那你一定要学会用 RelativeLayout 和 LinearLayout 组合使用,有的时候也需要 TableLayout 来帮忙。这里主要是说明一些基本的概念,在实际的开发中还要学会提取公共部分形成模块----继承关系,其他类似的样式,就在此基础上继承而来,这样来大大提高代码的复用率。

你只要把如下的布局搞定,你就没有问题了。

有兴趣的同学可以试试手,看看你掌握的怎么样,一般来说你能很容易地把上面的布局搞定,说明你的基础还是不错的,这没有什么好谦虚的。(增加难度:置顶和置下,实现其点击的范围扩大)。

跳转携带数据。

在从一个 Activity 跳转到另外一个 Activity 是很常见的,在跳转的时候一般都会携带上一个 Activity 的数据,怎么处理类似的问题,简单的说,用 Intent 实现跳转,用 Bundle 来携带数据,看下面的 demo code:

```
/**
```

本文档由 eoeAndroid 社区组织策划,整理及发布,版权所有,转载请保留!

*跳转到其他模块的入口函数

*

* @param targetClass

* 所要跳转的模块类名

* @param extrasData

* 跳转时要附带的数据

*/

```
private void gotoModel(String targetClass, Bundle extrasData) {  
  
    Intent intent = new Intent();  
  
    intent.setClassName(ROOT_APP_PATH, targetClass);  
  
    if (extrasData != null) {  
  
        intent.putExtras(extrasData);  
  
    }  
  
    startActivity(intent);  
  
}
```

6. Android 调试 logcat 技巧

收集 by

logcat 是 Android 中一个命令行工具，可以用于得到程序的 log 信息。

logcat 使用方法如下所示：

logcat [options] [filterspecs]

logcat 的选项包括：

- s 设置过滤器，例如指定 '*:s'
- f <filename> 输出到文件，默认情况是标准输出。
- r [<kbytes>] Rotate log every kbytes. (16 if unspecified). Requires -f
- n <count> Sets max number of rotated logs to <count>, default 4
- v <format> 设置 log 的打印格式，<format> 是下面的一种：
 brief process tag thread raw time threadtime long
- c 清除所有 log 并退出
- d 得到所有 log 并退出（不阻塞）
- g 得到环形缓冲区的大小并退出
- b <buffer> 请求不同的环形缓冲区 ('main' (默认), 'radio', 'events')
- B 输出 log 到二进制中。

过滤器的格式是一个这样的串：

<tag>[:priority]

其中 <tag> 表示 log 的 component，tag (或者使用 * 表示所有)，priority 如下所示：

```
V  Verbose
D  Debug
I  Info
W  Warn
E  Error
F  Fatal
S  Silent
```

事实上 logcat 的功能是由 Android 的类 android.util.Log 决定的，在程序中 log 的使用方法如下所示：

```
Log.v() ----- VERBOSE
Log.d() ----- DEBUG
Log.i() ----- INFO
Log.w() ----- WARN
Log.e() ----- ERROR
```

以上 log 的级别依次升高，DEBUG 信息应当只存在于开发中，INFO, WARN, ERROR 这三种 log 将出现在发布版本中。

对于 JAVA 类，可以声明一个字符串常量 TAG，Logcat 可以根据他来区分不同的 log，例如在计算器（Calculator）的类中，定义如下所示：

```
public class Calculator extends Activity {
/* ..... */
    private static final String LOG_TAG = "Calculator";
    private static final boolean DEBUG = false;
    private static final boolean LOG_ENABLED = DEBUG ? Config.LOGD : Config.LOGV;
/* ..... */
}
```

本文档由 eoeAndroid 社区组织策划，整理及发布，版权所有，转载请保留！

由此, 所有在 Calculator 中使用的 log, 均以"Calculator"为开头。

例如使用方法如下所示:

logcat &

< 得到一个 log 片段 >

W/KeyCharacterMap(130): No keyboard for id 0

W/KeyCharacterMap(130): Using default keymap: /system/usr/keychars/qwerty.kcm.bin

I/ActivityManager(52): Displayed activity com.android.contacts/.DialtactsContactsEntryActivity: 983 ms

I/ARMAssembler(52): generated scanline__00000077:03545404_00000A04_00000000 [29 ipp] (51 ins) at [0x25c978:0x25ca44] in 1764174 ns

I/ARMAssembler(52): generated scanline__00000077:03515104_00000001_00000000 [46 ipp] (65 ins) at [0x25d1c8:0x25d2cc] in 776789 ns

D/dalvikvm(130): GC freed 834 objects / 81760 bytes in 63ms

D/dalvikvm(52): GC freed 10588 objects / 425776 bytes in 94ms

其中 W/I/D 表示 log 的级别, "dalvikvm"/"ARMAssembler"等是不同组件 (component) 的名称, 后面括号里面的数字表示了发出 log 的进程号。

使用技巧:

- 1.使用 logcat &在后台运行
- 2.使用-d 得到所有 log
- 3.使用-f 或者重定向 (>和>>) 输出到文件
- 4.使用-s 设置过滤器, 得到想要的 log。

当然, 最重要的还是在程序中加入恰当的 log.

7. 用 **Android** 运行最简单的 **C** 程序

* 下载 GNU/ARM 编译工具

http://www.codesourcery.com/gnu_toolchains/arm/download.html

* 编写 c/c++代码.

* 使用 GNU/ARM Linux 工具创建一个应用程序, 不使用动态链接库

```
ex. arm-none-linux-gnueabi-g++.exe -static -o hello HelloAndroid.cpp
```

* 启动模拟器

```
$SDK_ROOT/tools/emulator.exe
```

* 在命令行窗口运行 **adb** 将编译好的 **hello** 程序放入模拟器的磁盘

```
adb push hello /system/sbin/hello
```

* 让 **hello** 文件变成可执行程序,不要使用 **chmod ugo+x**

```
adb shell chmod 777 /system/sbin/hello
```

* 运行 **hello** 程序

```
adb shell
```

```
cd /system/sbin/
```

```
hello
```

EXAMPLE HELLO WORLD CODE

```
//  
// HelloAndroid.cpp  
//  
//  
#include  
  
using std::cin;  
  
using std::cout;
```

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

```
using std::endl;

class MyName
{
public:
    void getname( void );
    void sayhello( void );

private:
    char name[ 255 ];
};

void MyName::getname( void )
{
    cout << "What is your name? ";
    cin >> name;
}

void MyName::sayhello( void )
{
    cout << "Welcome " << name << " to the world of Android" << endl;
}

MyName name;

int main( int argc, char *argv[] )
{
    name.getname();
    name.sayhello();

    return 0;
}
```

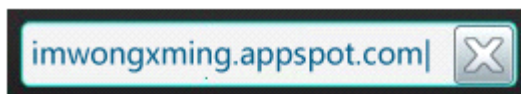
本文档由 eoeAndroid 社区组织策划，整理及发布，版权所有，转载请保留！

8.开发技巧杂集

转自: <http://imwongxming.appspot.com/page/1>

8.1 一些源于 css 的组合实现技巧

效果如下:



实现方式

LinearLayout 布局
EditText+Button 组合



RelativeLayout 比 LinearLayout 更好些

8.2 关于 Search 搜索框的使用

1.声明处理搜索的 Activity

```
<meta-data android:name="android.app.default_searchable"
android:value=".OnSearchAppsList" />
```

2.Activity 定义

```
<activity android:name=".OnSearchAppsList"
android:configChanges="orientation|keyboardHidden|navigation"
android:label="@string/Market">
<intent-filter>
```

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

```
<action android:name="android.intent.action.SEARCH" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

<meta-data android:name="android.app.searchable"
android:resource="@xml/searchable" />

</activity>
```

3.searchable.xml 定义

```
<?xml version="1.0" encoding="utf-8"?>

<searchable xmlns:android="http://schemas.android.com/apk/res/android"
android:label="@string/search_label"
android:hint="@string/search_hint" />
```

4.调用 Search 功能

```
onSearchRequested();
```

5.截获 参数

```
getIntent().getStringExtra(SearchManager.QUERY);
```

这后即可随意处理搜索请求了

8.3Android 是否有网络连接

```
public static boolean isNetworkAvailable(Context context) {
    ConnectivityManager cm = (ConnectivityManager) context.getSystemService(
    Context.CONNECTIVITY_SERVICE);

    NetworkInfo info = cm.getActiveNetworkInfo();

    return (info != null && info.isConnected());
}
```

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

8.4Activity 全屏和 无标题栏

```
// 全屏 (No Statusbar)
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
    WindowManager.LayoutParams.FLAG_FULLSCREEN);

// 无标题栏 (No Titlebar)
requestWindowFeature(Window.FEATURE_NO_TITLE);
```

8.5 使用 ViewStub 延迟展开视图

1) 定义 ViewStub

```
<ViewStub android:id = "@+id/stub_import"
    android:inflatedId="@+id/panel_import"
    android:layout="@layout/progress_overlay"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"/>
```

2) 展开视图

```
findViewById(R.id.stub_import).setVisibility(View.VISIBLE);

// 或者 获取视图
View importPanel = ((ViewStub)
    findViewById(R.id.stub_import)).inflate();
```

8.6 删除窗口背景

删除窗口背景

1) 编码实现

```
@Override
```

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

```
public void onCreate(Bundle icle){  
    super.onCreate(icle);  
    setContentView(R.layout.mainview);  
    // 删除窗口背景  
    getWindow().setBackgroundDrawable(null);  
    ...  
}
```

2) 自定义主题实现

```
<resources>  
  
<style name="NoBackgroundTheme" parent="android:Theme">  
  
<item name="android:windowBackground">@null</item>  
  
</style>  
  
</resources>
```

8.7 使用主题

```
<activity android:name="MyApplication"  
    android:theme="@style/NoBackgroundTheme">  
  
    ...  
  
</activity>
```

8.8 预先缩放图像到视图大小

预先缩放图像到视图大小

```
originalImage = Bitmap.createScaledBitmap(  
    originalImage, // 缩放图像
```

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

```
view.getWidth(), // 视图宽度

view.getHeight(), // 视图高度

true); // 线性过滤器

Adpater 使用 ViewHolder 模式

Adpater 使用 ViewHolder 模式

public View getView(int pos, View convertView, ViewGroup parent){

    ViewHolder holder;

    if (convertView == null) {

        convertView = mInflater.inflate(R.layout.list_item, null);

        holder = new ViewHolder();

        holder.text = (TextView) convertView.findViewById(R.id.text));

        holder.icon = (ImageView) convertView.findViewById(R.id.icon));

        convertView.setTag(holder);

    } else {

        holder = (ViewHolder) convertView.getTag();

    }

    holder.text.setText(DATA[pos]);

    holder.icon.setImageBitmap((pos & 1) == 1 ? mIcon1 : mIcon2);

    return convertView;

}
```

8.9 使用 `getString (int resId, Object... formatArgs)`的技巧

1.strings.xml

定义如下参数 `used_time`

```
<string name="used_time">用时%d 秒</string>
```

本文档由 eoeAndroid 社区组织策划，整理及发布，版权所有，转载请保留！

2.使用 getString 取得 used_time 值

```
getString(R.string.used_time,156)
```

输出结果: 用时 156 秒

在国际化时,很方便。

8.10 横竖屏切换时不重新加载 onCreate

为了防止横竖屏切换时 Activity 重新加载 onCreate,

只需要在 Activity 的配置文件里添加属性:

```
android:configChanges="orientation|keyboardHidden|navigation"
```

8.11 Android TranslucentTheme 半透明主题

1.Resource source file location: res/values/styles.xml

```
<style name="myTranslucentTheme" parent="android:style/Theme.Translucent">
<item name="android:windowBackground">@drawable/bg</item>
</style>
```

其中 bg 为一张半透明图片的索引

2. 使用自定义半透明主题

AndroidManifest.xml

```
<activity android:name=".WallpaperActivity"
android:label="@string/wallpaper_title"
android:theme="@style/myTranslucentTheme">
</activity>
```

8.12 模拟器玩 ping

Android 模拟器采用的是 Qemu, 而整个模拟的内核是 Linux, 虽然没有 xwindows

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

界面和 bash shell，但测试下基本的 Linux Shell 还是没有问题的。

首先我们进入 cmd 或 console 状态下，使用 cd 命令进入 Android SDK 的 Tools 命令，可以输入 adb shell 这时候，会提示* daemon not running. starting it now *，然后紧跟着显示* daemon started successfully *，当然这个之前 android 模拟器我们已经运行了，所以很快就找到了。这个时候我们的光标已经从>变为了#，现在已经在 linux shell 中了，虽然 Android 平台没有 vi 这样的编辑器，但是执行些简单的命令比如 cd、rmdir、mkdir 还是没有问题，我们直接在 Android 模拟器中测试 ping 本机，如# ping localhost。

8.13 通过 GSM call 命令打电话\发短信

通过 gsm call 命令可以像 Android 模拟器打电话，除了在 Eclipse ADT 的 DDMS 中通过按钮 Dial 外，还可以通过 DDMS 外壳调用 gsm call 命令直接拨打，我们首先需要启动 Android Emulator，cmd 环境下执行 telnet localhost 5554 就可以向 Android 模拟器拨号、发短信了，参数为 gsm call <PhoneNum> 和 sms send <senderPhoneNum> <message>

9. 在 Android 中使用 Map 需要注意的技巧

转自: <http://aidiji.com/viewtopic.php?f=27&t=776>

在 Android 中使用 GoogleMap 非常方便, 但是有些小技巧或者注意事项你必须牢记, 否则调试半天你也不会找到啥线索, 浪费时间不说, 更让你狂抓, 以下技巧都是我在实际项目中遇到的, 不断更新和维护, 如果你有类似技巧, 欢迎分享:

(以下技巧是基于 SDK 1.0 的)

一、申请 Apikey, 并放在正确的位置

这个应该都知道, 但是是申请得到的 key 放哪里很多人不知道, 可以放在

1、XML 布局文件中

代码: [全选](#)

```
<view android:id="@+id/mv"
    class="com.google.android.maps.MapView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:apiKey="01Yu9W3X3vbpYT3x33chPXXX7U1Z6jy8WYZXNFA"
/>
```

2、java 中

```
mMapView = new MapView(this, "01Yu9W3X3vbpYT3x33chPxxx7U1Z6jy8WYZXNFA");
```

二、记得导入 uses-library

由于 1.0 版本的修改, 使得 map 包不再是默认的了, 使用的时候需要在 manifest 中的 application 节点下加入

```
<uses-library android:name="com.google.android.maps" />
```

否则, 你将遇到可恶的 “java.lang.NoClassDefFoundError: ”, 切记!

三、需要给予一定的权限

因为要使用 GoogleMAP 的 service, 所以需要

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
```

如果需要 GPS 等应用, 还需要

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
```

四、Activity 需要继承自 MapActivity

类似如下代码;

代码: [全选](#)

```
package com.iceskysl.showmap;
import com.google.android.maps.MapActivity;
import android.os.Bundle;

public class ShowMap extends MapActivity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    @Override
    protected boolean isRouteDisplayed() {
        // TODO Auto-generated method stub
    }
}
```

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

```
        return false;
    }
}
```

10.每个人应该会用的四个 Google android 小技巧

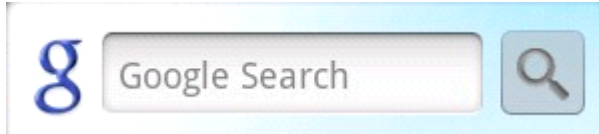
10.1 小技巧 1: 随时使用 MPEG-4 H.264

Google Android 系统内置了硬件视频加速器。硬件加速器最主要的就是使得 MPEG-4 H.264 的文件在你的显示器上显示的更加顺畅, 你能在标准的 Google android 系统中找到“视频播放器”的设置功能, 但很不幸的是这必须从

机器人商店 (android Market) 上下载, 作为 android 不配备视频播放器。另一方面, 视频播放器是免费的, 并且可以在几秒钟内下载到。

10.2 小技巧 2: 硬检索关键字

在您的 Google Android 手机屏幕的上边有一个像放大镜的小符号,



它将会迅速的带您到谷歌的搜索页面, 这是一个伟大的在线快速搜索功能, 尽而使关键字使用起来更加方便, 例如, 如果您在任何谷歌建立的网站, 如 YouTube 您可以单击同样的选项写入您要搜索的内容, 然后搜索, 谷歌拥有的任何搜索字词您在搜索框出现的类型的网站。

此功能的另一大应用是, 它是每个应用的具体框, YouTube 的可能只是一个典型的搜索框, 而谷歌的地图, 它会提示您包括如地址, 城市, 州, 邮编等信息。

为了快速启动搜索功能, 具体的谷歌网页确实没有什么比谷歌机器人的搜索功能更好。

10.3 技巧 3 使用手机摄像头拍摄商品条码消费

谷歌机器人设有免费下载的程序 (从机器人市场), 众所周知的 Google Adroid 应用商店。这个程序允许用户扫描 (使用他们的设备摄像头) 任何他们在商店的访问产品上的条码, 然后利用他们的设备数据, 他们可以连接到当地的商店, 发现携带已浏览的产品, 其中包括托马斯·提供价格其他本地商店。

如果您的消费较大的购物者, 这是一个必须有的应用, 且没有成本下载它不会给你什么损失

10.4 小技巧 4: 快捷键

如果你想利用特殊字符的优势, 删除整个文本行, 只使用资本 letters, 或其他重要功能, 您将学习如何操作与使用设备的每一个方便的快捷键。下面是几个我经常使用流行的选项列表。

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

1. 插入特殊字符: 按 Alt + 空格键
2. 突出显示文本: 按下 Shift 和滚动轨迹球
3. 打开全部大写: 按两次 Shift
4. 删除整行文字: 按 Alt + 删除
5. 移动光标到年底或文本的开头: 按 Alt 同时滚动轨迹球
6. 插入选项卡空间: 按 ALT + Q

总结:

在谷歌 Android 手机操作系统仍处于起步阶段, 但已经掌握的业务数量已经很大, 而且将一定会继续增长, 因为新的 ROM 更新变得可用。无论您想要使用机器人应用服务提供给您日常的需要或只是学习一些简单的技巧, 帮助您更轻松地了解您的设备, 我希望上述信息给你一些洞察你设备的能力, 多一点了解。我会继续发布更多的技巧和在不久的将来技巧。请继续关注

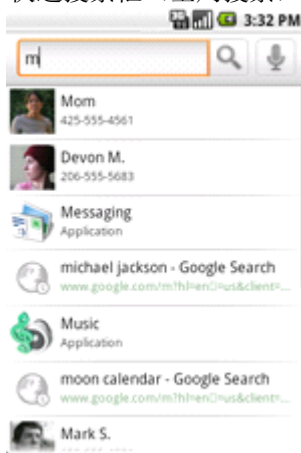
11.Google Android SDK 1.6 发布以及重大性能提升!

以下为新功能和技术的简要介绍:

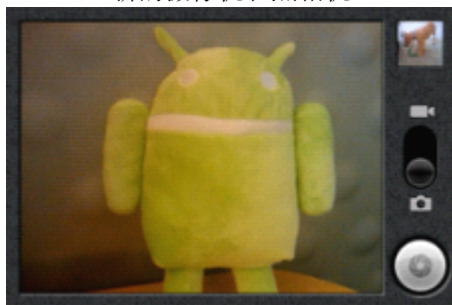
- 系统新功能
- [Android Market 升级](#)
- [新平台的新技术](#)

系统新功能

快速搜索框（全局搜索）

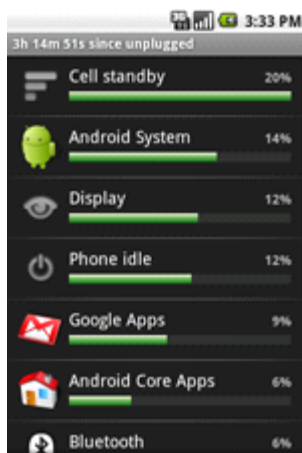


新的摄像机和照相机



电池用量指示

本文档由 eoeAndroid 社区组织策划，整理及发布，版权所有，转载请保留!



Android 快速搜索框

Android 1.6 包括了一个重新设计的内核库文件, 它提供了快速高效的功能让用户能搜索不同来源的资料——比如浏览器的书签、历史、联系人、网页! 这一切都是在桌面上! 这个系统还能学习你对搜索结果的选择, 所以经常使用的联系人或者程序会被自动提前(当你只输入几个字母的时候), 这个内核库还提供给开发者简单容易的方法, 让他们能在程序中使用这个功能。

照相机, 摄像机, 相册

一个升级后的用户界面提供给用户全新的照相机、摄像机和相册体验! 用户可以很快速的在他们之间切换。现在相册还可以一次选择多个相片。Android 1.6 也提供了一个更快速的拍照体验。比较旧版本, 现在打开相机的速度提高了 39%, 拍照速度也提高了 28%!

VPN, 802.1x

设置中的 VPN 控制板现在允许用户设置并且连接一下类型的 VPN 网络

- L2TP/IPSEC pre-shared key based VPN
- L2TP/IPsec certificate based VPN
- L2TP only VPN
- PPTP only VPN

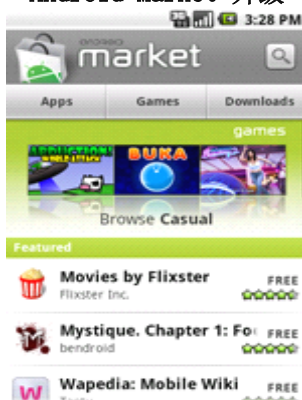
电池用量显示

一个新的电池用量屏幕可以让用户看到哪个程序或者服务在不断啃电池。如果用户发现某个服务或者软件很耗电, 他们可以通过调整设置、停止程序或者卸载来解决这个问题。

辅助工具

用户现在可以下载新的辅助工具服务, 并且能在设置中打开他们

Android Market 升级



本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

新的 Android Market 界面

那些使用 android market 菜场（我就喜欢这么叫）的设备，现在最新版本的菜场大大提升了用户体验，并且用户能更容易发现他们感兴趣的软件

- At the homescreen, users can choose among *Apps*, *Games*, and *Downloads*.
- 在桌面上，用户可以在程序、游戏、下载中选择
- Inside a category, users can explore titles that are *Top paid*, *Top free*, and *Just in*.
- 分类中，用户可以探索标题为：最受欢迎的付费软件、免费软件、最近添加
- For each title, users can now see screenshots submitted by developers in addition to reviews from other users.
- 对于每个软件，用于可以见到截图（终于有了！）和用户的评价

新平台新技术

扩展的搜索引擎

全新设计和扩展的 android 搜索引擎库允许第三方软件使用，也就是全局搜索工具啦！开发者需要让他们软件“有搜索能力的”并且在用户使用的时候提供建议。（YY 觉得是可以让程序通过全局搜索库来搜索你的手机文件）。用户可以在每个设置中单独设置程序的搜索权限！

文字到语音引擎

Android 1.6 用了一个叫 pico 的东西来提供多语言语音引擎。它允许软件能用各种各样的语言、各种各样的口音来“读”文字！支持的语言：英语（美国、英国），法语，意大利语，德语，西班牙语。如果你在用 Dream 的话，你需要从菜场下载 SpeechSynthesis Data，这个包包提供了所需要的引擎文件

手势

一个新的手势库文件可以让开发者创建、保存、打开、认识用户的手势，并且可以用于某些操作用户也可以用全新的手势创建器工具（包含在 SDK 1.6 中）在开发的软件中来创建手势库

复制工具

Android 1.6 提供了全新的辅助工具库，开发者可以利用它创建辅助插件来回应用户输入，比如窗口最小化响一个声音，页面回到顶的时候震动一下，当然也提供语音回馈！

扩展对屏幕色彩深度和分辨率的支持

Android 1.6 增加了对软件全新的分辨率和色彩深度的支持，开发者可以对软件指定分辨率和色彩等

对于 CDMA

Android 1.6 包含了对 CDMA 技术的支持！（电信的最爱！）

新版本的 OpenCore

Android 1.6 包括了新的 OpenCore 2 媒体引擎，有：

- 对 OpenMAX 编码的支持
- 对其他 AuthorEngine 中的音频格式的支持
- Improved buffering model supports shared buffers allocated in the decoder
- 改进了对一些流媒体的解码

本文档由 eoeAndroid 社区组织策划，整理及发布，版权所有，转载请保留！

2.6.29 Linux 内核

Android 1.6 升级 Linux 内核从 2.6.27 到 2.6.29.

新的 API 库文件

关于新的 API 的详细介绍, 请查看 [Version Notes](#). 关于所有的 API 更改的完整报告, 请查看 [API Differences Report](#).

新的 Android Market 重新设计的 Android Market, 新提供了屏幕截图, 图标显示, 添加描述等这些更直观的表现方式, 可以让开发者更好的展示自己的作品。搜索增加运动, 健康, 主题和动漫 4 个分类。

手势支持 新的 SDK 里面加入了一个 GestureBuilder 类, 可以让开发者定义自己的手势! 这下 Gphone 牛逼了, 什么多点触摸, 汉字识别都不在话下了。

语音识别 Dount 新包含一组 Text-to-Speech API, 支持了更多了语音识别, 包括英语、法语、德语、意大利语等。

手机搜索功能 有一个让人期待的功能, 可以把手机里的联系人, 音乐, 浏览历史, 书签什么的统统搜出来。

相机功能的一些改变 根据 Android 开发组的数据, 照相速度变快了 28%, 相机程序启动速度快了 39%。

应用程序耗电查看 对软件耗电情况一目了然, 更方便用户查看软件耗电情况。支持 VPN 可以配置和连接到各种 VPN 上。其他 支持更大的屏幕 WVGA, 支持, linux 内核升级到 2.6.29, 支持 OpenCore2 多媒体引擎。

12.eoeMarket

eoeMarket 将从本期开始在每一期特刊中, 为大家推荐一些 eoeMarket 平台中的一些精品的软件, 而且在后期, 我们甚至还会为大家推荐一些优秀的 Android 开发团队。

所以, 如果你有好的应用, 你应该尽快上传到我们的 eoeMarket。

因为是第一期, 可能还有很多朋友并不熟悉我们的 eoeMarket 软件平台, 所以我们会首先占用一定的篇幅给大家着重介绍一下 eoeMarket。

12.1eoeMarket 是什么?

eoeMarket 是由 eoeAndroid 社区推出的一个主要针对中文市场, 和 Android 开发者紧密合作、并给广大玩家提供优秀 Android 应用的集软件发布、搜索、安装于一体的平台。eoeMarket 平台致力于提供更好的 Android 软件服务, 拉近开发者和广大玩家的距离。

eoeMarket 包含两个部分:

web 端: 在 web 端可以发布, 搜索, 收藏应用

手机客户端: 在手机客户端可以浏览, 下载、安装应用

12.2eoeMarket 能给开发者带来什么?

A eoeMarket 对于开发者 在本地市场的 帮助

1. eoeMarket 的用户主要是中文用户, 用户的使用习惯我们更加清楚, 因此我们可以透过 eoeMarket, 制作出更适合中国人的软件。
2. eoeMarket 对于开发者而言, 不仅仅是一款简单的发布软件, 他更是一个平台, 一个具有交互功能的软件平台, 它能为开发者拉近与中国用户之间的距离, 在国内的环境下, 可以与更多的用户进行更好的交流。
3. 如果您的软件足够优秀, eoeMarket 可以帮您推荐给运营商和手机厂商, 如果您的软件可以成为运营商或者手机厂商指定的预装软件, 那您的软件所获得影响力将会成倍得增长!
4. eoeMarket 也是您展示实力的好机会, 优秀软件的作者, 也必定是优秀的您, 那么您更加可以透过 eoeMarket 来彰显自己的实力, 因此, 您能获得更多利润丰厚的外包项目。
5. eoeMarket 的定位决定了 eoeMarket 上的软件会被更多人下载和使用, 这是一次绝佳的给您的软件提供免费的反馈和建议的机会。这样您的软件会获得更多的用户的赞同, 同样也会获得更多的市场认同。

B eoeMarket 对于开发者 在国际市场的 帮助

6. eoeMarket 是针对国内用户的, 而国内有着更为挑剔的用户群, 而 eoeMarket 的环境更为宽松, 在您准备发布自己的软件时, 可以第一时间发布到 eoeMarket 中, 有了如此多的挑剔的玩家来检验, 可以让我们及早发现程序中的 bug, 或者不合理的地方。
当我们改掉这些不合理的地方后, 再上传到 Android Market 必定能征服更多挑剔的用户。

12.3 eoeMarket 能给玩家带来什么?

为方便国内用户更好地安装 Android 软件, 提供更多更好的适合我们中文本土化的应用软件, eoeAndroid 社区特地为广大的开发者和玩家提供了 eoeMarket 平台。
从此以后开发者可以轻松将自己做的最新最酷的软件上传到 eoeMarket, 广大的玩家朋友们也可以很方便地

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

在 eoeMarket 平台下载好玩、好用的本土化应用。

总而言之, eoeMarket 是 eoeAndroid 社区奉献给广大 Android 开发者和玩家的一份厚礼。

12.4 为什么使用 eoeMarket

1. 本土的服务器, 更快的访问速度。
2. 以中文应用软件及游戏为主。
3. 全部都是精选的软件, 不浪费您每一 KB 的流量。

eoeMarket 具有和 Android Market 同样优秀的界面, 只要大家使用过 Android Market, eoeMarket 可以很快上手。

最关键的是 eoeMarket 提供的应用软件是以本地化的中文语言为主的, 大部分软件和游戏都是让大家感到亲切友好的中文版, 再也不用担心下载下来的软件只有难懂的英文, 有 eoeAndroid 社区背后的支持,

eoeMarket 更会为大家推荐出更多更好的优秀软件。

不同于 Android Market 的杂乱, 在 eoeMarket 上, 看到的都是优秀的软件。

12.5 如何使用 eoeMarket

1. 如何注册? 通过 web 注册

您可以在您的电脑里边打开浏览器, 通过网址 <http://www.eoemarket.com/> 进行注册。



发布应用的分享平台.

首页	应用	游戏	主题	widget	下载客户端	关于eoeMarket	联系我们
----	----	----	----	--------	-------	-------------	------

为什么要注册?

注册后可以为您提供个性化的服务;
注册的帐号可以在web和手机客户端通用;
注册的帐号可以申请开发权限

注册新帐号

您的账户信息 (这是你登录eoeMarket的凭证)

登录名* (请使用3..40位的小写字母, 数字或 - _ @)

Email地址* (请输入可以正常接收邮件的地址)

密码* (请输入您的密码)

确认密码* (请再次输入您的密码)

注册帐号

[About](#) - [Features](#) - [Agreement](#) - [Contact](#)

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

过手机客户端注册

The image shows two side-by-side screenshots of the eoeMarket mobile application interface. Both screens have a dark background and a grey header bar with the 'eoeMarket' logo. The left screenshot, taken at 18:14, displays a login screen. It features a '用户名:' (Username) field with the text 'eoeMarket账号' and a '密码:' (Password) field. Below the fields are two buttons: '登录' (Login) and '取消' (Cancel). At the bottom, there is a link that says '快速注册eoeMarket帐号!' (Quickly register eoeMarket account!). The right screenshot, taken at 20:03, displays a registration screen. It features four input fields: '用户名:' (Username) with 'eoeMarket账号', '邮箱:' (Email) with '用来验证的邮箱地址' (Email address for verification), '密码:' (Password), and '确认密码:' (Confirm Password). Below these fields are two buttons: '注册' (Register) and '取消' (Cancel). The top status bar of both screenshots shows various mobile icons and the time.

2. 如何下载 eoeMarket 客户端软件?

通过 PC 安装

您可以通过网址: <http://www.eoemarket.com/download> 来下载客户端



通过手机浏览器进行安装

手机浏览器打开网址 <http://www.eoemarket.com/down> 进行安装。



3. 开发者如何上传软件?

注册后, 您可以选择成为开发者, 这样您就可以上传应用软件了。

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

开发者管理面板

您个人的应用发布平台...

快捷面板

[管理面板](#)

[发布新应用](#)

应用相关

[发布新应用](#)

[应用列表](#)

反馈相关

[应用反馈](#)

[用户留言](#)

统计相关

[统计首页](#)

[应用统计](#)

[标签统计](#)

[技能统计](#)

账户相关

[我的帐户](#)

[收支明细表](#)

[帐户充值](#)

[提现申请](#)

step1、请上传你要发布应用的APK文件（需要签名。）

选择APK文件: No file chosen

发布新应用步骤

step1:上传APK文件

选择需要发布应用的APK文件，上传成功后可以智能分析出您的应用中诸如版本，包名，icon等信息，免去您手工输入；

step2:填写app信息

该步骤您会看到智能分析出的应用信息，除此之外，需要您输入应用的名字，描述，同时需要您选择分类，输入标签、价格等信息。

12.6 eoeMarket 软件推荐

从这期开始小编会为大家推荐 eoeMarket 的优秀软件。如果您认为哪个软件好用的话，也可以直接告诉小编，发邮件到 eoemarket@eoemobile.com

eoeMarket 致力于为用户提供最好的中文化软件，所以我们选择的软件必须都是支持中文的。在这里给大家推荐两款软件：

A 每日一句

作者: xzf158

经过一个星期的努力开发与测试，每日一句基本功能已经完成，虽然还不够强大，但以后的不段升级中，相信会越来越好。

此版本添加新的功能：

- (1) 支持设置每天更新时间功能。
- (2) 添加多风格皮肤功能，有两种皮肤供大家选择。
- (3) 支持手动添加每日一句功能。
- (4) 桌面可选择上一条，下一条功能。

修改部分：

- (1) 修改设置菜单界面
- (2) Logo 图标与菜单图标
- (3) 优化整理代码程序代码，以方便修改升级

新版本效果图：

本文档由 eoeAndroid 社区组织策划，整理及发布，版权所有，转载请保留！



B. appSharers

AppSharers 是 eoeMobile 团队又一力作。通过非常简单的方法，就可以将自己手机中所有的应用软件分享给自己的朋友。Android 手机优秀的 e-mail 功能可以直接发送给你的朋友你共享的是哪些应用程序，此外你还可以通过发短信来分享。

只分享还是不够的，点开排行榜，还可以看到所有 AppSharers 的用户，共享最多的是哪些应用。想想当年 Twitter 怎么走红的，难道 AppSharers 会成为手机应用平台的下一个 Twitter 吗？



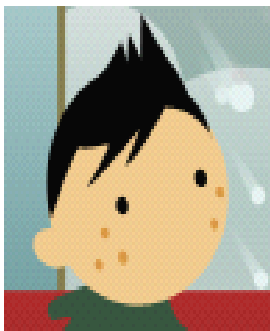


同样 AppSharers 还提供了一个网页版的页面

访问 AppSharers 网页版 请到

<http://www.appsharers.com/>

13.介绍特刊组成员



向上: 负责特刊的编辑

论坛 ID:xusaomaiss

QQ:63590240

Email:xusaomaiss@gmail.com

个人签名: 坚持向上出品必为精品, 向上人生路。



明叔:

论坛 ID:fanth

QQ:1093148692

Email: clm16668@gmail.com



王华: 负责特刊推广及人员联系组织

论坛 ID: 情话 love

QQ: 526155779

个人签名: 活自己, 潇洒自己, 做我们自己的特刊。

特刊组刚刚成立, 无论是技术, 人员等等很多方面都还不完善, 我们相信由于你的加入, 为特刊贡献出自己的那份力量, 会使我们的特刊变的更加强大!

因你的加入我们的团队又强大一分!

-----情话 献上

如果有对 android 开发有兴趣的朋友, 或是对我们的特快有兴趣的朋友都可以联系我, 你就有机会在我们的特刊上崭露头角啊!

联系情话 MSN:qinghua3344521@msn.com

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

大家如何有关于特刊的任何问题,都可与我们联系,希望能得到大家对特刊的反馈,同时也可以告诉我们你的需求,我们将尽量做到最好,做到您满意!

14.其他

BUG 提交 如果你发现文档中翻译不妥的地方,请到如下地址反馈,我们会定期更新、发布更新后的版本

<http://www.eoeandroid.com/viewthread.php?tid=753>

资源下载: 本期文部分中包含的源码请在如下地址下载

<http://www.eoeandroid.com/viewthread.php?tid=753>

关于 eoeAndroid

当前,3G 商业,传统互联网与移动互联网也呈现出全业务发展的融合趋势,电信与互联网行业已经踏入继单机计算时代、传统互联网时代之后的第三个纪元。

由于看好移动互联网和 Android 手机平台的商业前景,同时也拥有专业而独特的产品、技术服务能力,我们聚集了一群热爱 Android 的技术英才,组建了 eoeMobile 团队。

eoeMobile 是一支专注于 Android 平台应用开发、产品运营和相关商业与技术服务的团队,立志于建立中国最大的 Android 应用开发专业社区 eoeAndroid.com,想为 Android 在中国的发展尽自己的微薄之力。

本文档由 eoeAndroid 社区组织策划,整理及发布,版权所有,转载请保留!

15. 游戏诞生记 真的要生蛋了

今天不玩游戏, 我们作游戏!

我的游戏我制作!

游戏制作人不再是梦想

我做你玩的游戏



15.1 游戏诞生记一月总结

这里很高兴得向大家分享我们的一些经验教训。

(1) 先研究你的创意

首先我们做的是一款游戏, 这是一款软件程序, 但是他又不是一般的软件程序。正如之前所说的, 一款游戏要挖掘人们内心深处自己都不知道的对快乐的需求。这就体现了游戏制作的难度。似乎到现在有人还会想这个活动的一些困难之处。

诚然, 每一个人心中都有不一样的标准, 但是不一定每一个人心中的标准都无法和别人的相适应。

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

所以,甚至在我们开始策划案之前,就用了大量的时间来讨论,我们到底要做什么游戏,我们做成什么样子。而慢慢的,等到初稿形成以后,也在不断询问大家。可是这个时候只是问没太大的作用。这里大家也发现了,咱需要一专业的策划稿,所以就按照初稿的形态,一步一步细化,做出了大纲。

(2)让你的大纲变成你开发的龙骨

这里的大纲真的是非常重要,怎么说呢?一旦将你的大纲划分好,那就说明你对游戏的整体结构有了非常明确的划分。这样就可以更加深入细化得进行开发了。

大纲在之前封闭了一段时间,也是为了积攒能量,这个大纲很好,而且得到了很多前辈的赞同,但是同样,问题也很明显,就是对开发者做的还不够,有一些设定还是不够细。

当然了一步到位还是远远不够的,所以还要进行更加细的划分,特别是对于我们开发人员来说,我们的游戏还需要有一个引擎的架构。这样才能承载这些游戏的内容。

所以大纲的开发也在一直等待这一部分的完善。

(3)交织在一起的开发

我想真正做过游戏开发的朋友也许更加清楚,一款游戏的开发需要大家共同的努力,我们之间必须协同的工作。大家不是分开的,如果各自为政那反而会造成一些协调的困难。

而此时,我们需要在各个方面都还需要不断的努力,首先是我们的原画,毕竟人物和背景原型都是从这里诞生的。其次是需要一个实实在在的剧本,有了这个我们的音效师大饼才能继续他的创作,而且对字幕旁白才能有依有据。另外我们的架构如果想要跑起来,也都需要这么多的资源。Hoho,各位是不是热血沸腾了呢?

(4)我们的收获

加入到活动中的朋友,也都有自己的收获。不过最重要的,还是那种做东西的兴奋,当初共享给大家的初稿的时候。很多朋友看到初稿的蛋后,都觉得非常有意思,而我们要的就是那种感觉,因为真正能做好游戏的我们一定也是喜爱游戏的玩家。如果自己能让自己高兴起来那我们才算成功。

而大纲中的曾哥、春哥想必也能激发起大家无数的斗志吧。

最后也为那些正在游戏开发道路上的朋友和准备进行游戏开发的朋友说给一些建议:

A 你的游戏你必须喜欢

B 你的游戏请一定要让他变得好玩

C 在你想好了游戏的玩法以后,就要想办法来进行分解,分得一定要细

D 确定好每一部分的结构以后,按部就班一步一步稳扎稳打得来

15.2 游戏诞生记 所有资源汇集贴

1.

初稿

http://docs.google.com/View?id=df69hmkq_03nmt26m

2.

大纲

http://docs.google.com/View?id=df69hmkq_28gn5xbmht

3.

剧本

(未完成)

4.

原画

http://docs.google.com/View?id=df69hmkq_55dz5r4mhg

5.

音效

(等待剧本)

6.

流程图、引擎

https://sites.google.com/site/eoegametebiezu/dan-sheng-ji_you-xi-jia-gou-yin-qing-/liu-cheng-tu

此页面有流程图的 PDF 和 Visio 图

7.

代码

代码在 Google Code 上, 并使用 SVN 来进行协同的开发

1. 如何使用使用 SVN

<http://www.eoeandroid.com/viewthread.php?tid=3312&extra=page%3D2>

<http://www.eoeandroid.com/viewthread.php?tid=3313&extra=page%3D2>

2. 如何将 SVN 和 Google Code 进行绑定

<http://www.eoeandroid.com/viewthread.php?tid=3314&extra=page%3D2>

Google Code 项目地址:

1. 模块 demo 地址: <http://code.google.com/p/eoeegdemo/>

2. 项目代码: <http://code.google.com/p/eoeegg/>

eoeAndroid 社区与 Gphone 中文网联合推出的“[游戏诞生记](#)”邀您加入!

我们的讨论群:4276804

三言两语, 不顶事儿。如果看到这个帖子心动了, 最好的办法就是赶紧加入我们的开发吧!

eoeAndroid 专区 <http://www.eoeandroid.com/forumdisplay.php?fid=47>

Gphone 中文网专区: <http://forum.tgbus.com/forumdisplay.php?fid=230>

本文档由 eoeAndroid 社区组织策划, 整理及发布, 版权所有, 转载请保留!

16. eoeAndroid 社区为大家提供一个优质的 **Android** 人才平台

21 世纪什么最重要，人才啊！
2009 年什么难找，工作啊！

大公司一方面找人找不下，在 eoeAndroid 社区优秀的朋友却工作找不到这太不正常了。这是怎么回事儿呢？是因为大家没找对地方。

人才去招聘网站、人才市场，那里鱼龙混杂，早都被埋没了。

招人去招聘网站、人才市场，千篇一律的简历，早就看烦了。

如果你的公司想在 Android 领域发展，想招到最适合做 Android 开发的人。

如果你励志投身一家以 Android 为主的公司！

请来 eoeAndroid 人才平台！

1. eoeAndroid 是中国最优秀的 Android 开发社区，一定有潜在的大量的 Android 开发者！
来 eoeAndroid 招聘 Android 开发者，一定能找到最优秀的人才
2. eoeAndroid 是优秀开发者的聚集地，来 eoeAndroid 更能证明你自己的专业
找工作，来 eoeAndroid 人才平台！

招聘专区链接：

<http://www.eoeandroid.com/forum-26-1.html>