





Android开发高级进阶:

开发商业级Android应用程序的注意事项

主讲: 张运芳

zhangyunfang@eoemobile.com

www.eoemobile.com





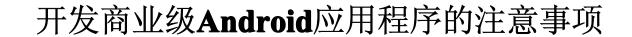
教学大纲

1.客户讨厌慢的程序。

-----性能,提高应用性能的各种方法总结。

- 2.程序响应方面的注意事项
- 3.程序友好性和使用流畅性方面的注意事项









1.客户讨厌慢的程序

有限内存 有限运算能力 有限电池寿命 有限存储 有限屏幕







Efficient Coding Design

两个基本的原则:

不要做不必要做的事情。 尽可能的节省内存的使用







Efficient Coding Design

用户体验三个核心特征:

- 1.快速
- 2.响应
- 3.无缝







快速

◆尽可能避免创建对象(Object)

例如: 当从原始的输入数据中提取字符串时, 试着从原始字符串返回一个子字符串, 而不是创建一份拷贝。

◆使用Native方法

例如: 当处理字符串的时候,尽可能多的使用诸如 String.indexOf()、String.lastIndexOf()这样对象自身带有的方法。 因为这些方法使用C/C++来实现的,要比在一个java循环中做同样的事情快10-100倍。







快速

◆缓冲属性调用

例如:

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

int count = this.mCount; Item[] items = this.mItems;

for (int i = 0; i < count; i++) dumpItems(items[i]);







快速

◆尽可能避免接口引用实例对象

例如:

Map myMap1 = new HashMap(); HashMap myMap2 = new HashMap(); 通过接口引用来调用会花费2倍以上的时间







快速

◆声明Final常量

例如:

static String action="com.eoemobile.action.SHOW_PHOTO"
static final String action="com.eoemobile.action.SHOW_PHOTO"
直接写入了类文件静态属性初始化中,初始化直接由虚拟机来处理。







快速

◆避免浮点类型的使用

嵌入式的处理器通常并不支持浮点数的处理

所有的"float"和"double"操作都是通过软件进行

一些基本的浮点数的操作就需要花费毫秒级的时间







一些标准操作的时间比较

Action	Time
Add a local variable	1
Add a member variable	4
Call String.length()	5
Call empty static native method	5
Call empty static method	12
Call empty virtual method	12.5
Call empty interface method	15
Call Iterator:next() on a HashMap	165
Call put() on a HashMap	600
Inflate 1 View from XML	22,000
Inflate 1 LinearLayout containing 1 TextView	25,000
Inflate 1 LinearLayout containing 6 View objects	100,000
Inflate 1 LinearLayout containing 6 TextView objects	135,000
Launch an empty activity	3,000,00





2.程序响应

在Android中,程序的响应性被Activity Manager和Window Manager 这两个系统服务所监视。

当出现下列情况是, Android会认为该程序无响应:

- ◆5秒内没有响应用户输入事件(如键盘输入)
- ◆一个BroadcastReceiver 执行十秒还没有完成







ANR--Application Not Responding

应用程序的执行在你的主线程里超过一段时间就会导致弹出ANR对话框

任何主线程的方法都不能做很复杂的处理。特别是Activity对象在它的关键生命周期函数里面不能处理太多,例如onCreate()和onResume()。

主线程应该提供一个Handler给子线程去返回完成信息







加强响应性

- ◆应用程序正在后台运行并响应用户输入,进程正在创建 ▶使用ProgressBar和 ProgressDialog
- ◆游戏编程 ▶在一个子线程里做位移运算
- ◆应用程序有一个比较耗时的初始化过程
 - ▶考虑显示一个开场动画
 - ▶尽可能快的显示主窗口然后异步的填充其他部分







3.程序友好性和使用流畅性

视图和布局

越简单越好

- ◆如果一个窗口包含很多视图
 - ▶启动时间长
 - ▶测量时间长
 - ▶布局时间长
 - ▶绘制时间长
- ◆如果视图树深度太深
 - ➤ StackOverflowException
 - ▶用户界面反应速度很慢







更多资源

- ◆Android专业开发社区 http://eoeandroid.com
- ◆Android blog http://blog.eoemobile.com
- ◆Android 应用上传和发布 http://eoemarket.com
- ◆Android中文学习教程《Google Android开发入门与实战》







谢谢!!

