android 人机界面指南

1Android 设计的依据

1.1 框架结构及流程

是什么使得 android 有着独特的用户体验？

- 后台处理支持多任务功能
- 正在进行和事件驱动的提示信息
- 通过 Widgets 和 live folders 来实现实时信息的预览
- 用户想用时，任一应用程序都可以挑选和选择
- android 不是关于程序的，它是关于活动，把任务分层，

1.2 架构基础

硬件平台

android 设备代表的是硬件和软件的完美组合。硬件辅助导航操作，并给 android 提供更多更好的功能。当菜单没有开启，要把屏幕最大化时，菜单按钮可以在屏幕上提供更多的内容。返回按钮允许使用返回堆（back stack）。

竖屏与横屏

一般来说，用户界面开发竖屏与横屏。在新横屏也仍存在于新的 Android 手机中。99％的 android 布局支持横屏。

焦点和菜单

在触摸模式里没有焦点，只有轨迹球。Android 平台里没有鼠标焦点。确定你从未显示焦点。主菜单应该包括全部功能；它们与活动联系一起形成整体。菜单上的图标按重要性排序。如果有多于 5 个图标，使用点击 more menu 菜单来查看那些不太重要的菜单项。上下文菜单（长按）集中在一个特定对象。总是把那些与所选项最相关的行为放在长按菜单的顶部。

需要记住的几点：

- 设计时要考虑速度和简洁
  尽量分层来分等级
  屏幕上的活动尽量最小
  使用下载进度条，下载数据时，而不是让用户等待去看一个加载完全的页面。
- 考虑活动流而不是线性行为

1.3 屏幕上的行为

android 设计了特定的行为方式。在你的应用程序里利用好这一点。应该坚持 android 行为的标准，避免混淆用户。

1.4 表达

细节使得产品集中在细节。程序的美学会帮助你集中注意在那些应用体验核心的关键任务上。API DEMO 是开始你的工具包的好地方。

2 用户界面原则

这部分试图讲述创造一个好的用户界面的一些基本的交互设计原则。这些原则是基本的，不止能应用于 android 的用户界面设计，也可以应用于其他。苹果建议开发者花费 60%的开发时间来进行设计工作。下面的用户界面原则将为好的设计提供一个基础。

2.1 隐喻

隐喻是构建一个基于操作任务心智模型的模块；用它们来传递应用程序的概念和功能。基于真实世界的应用对象可以帮助用户很快的理解该应用程序。当你设计你的应用程序时，要注意 andriod 中存在的隐喻，不要重新定义它们。同时，检查你的应用程序执行的任务，看是否有些自然隐喻你可以使用。

2.2 反映用户的心智模型

用户已经有了一个来描述你的程序正在进行的任务的心智模型。这个心智模型产生于真实世界经验、其它软件和一般电脑基本知识的结合。比如说，用户在真实世界里有写字、寄信的经验，也会产生特定的期待，像写一封新的信，选一个接受者，然后寄出信。一个忽略用户心智模型的电子邮件程序用起来会很困难和不舒服。这是因为程序强加给用户一个不熟悉的概念模型，而不是建立一个用户已有的知识经验模式。

在设计程序用户界面之前，试着去发现你的用户的心智模型,这样帮助用户去执行任务。心智模型中内在的隐喻，它代表了任务的概念组成。在写信这个例子中，隐喻包括信件、邮包和信封。在涉及到照片的任务的思考模式中，隐喻包括照片、照相机和专辑。我们要努力地发现用户的期望，包括任务组成、组织、窗口布局的工作流、菜单和工具栏组织、控制面板的使用。

要通过努力地何必把个下面的特征与用户心智模型相融合：

熟悉性

用户的心智模型主要是建立在经验的基础上

简单化

一项任务的心智模型通常是流线型，关注任务的基本组成部分。尽管对于一个给定的任务有很多可选的细节，但是基本的组成部分占大部分，并且不会占用用户的注意。

可利用性Availability

避免 components 在子菜单中过深的隐藏或者只在上下文菜单中可用。

发现性

通过提供怎样使用用户界面控件的线索来鼓励你的用户区发现一些功能。要鼓励通过做一些难以逆转或恢复的动作来发现。

2.3 直接操作

直接操作意味着人们感觉他们正在控制一些可触的事情而不是抽象的。直接操作的好处是当用户可以直接操作对象时，他们能更好的明白自己操作的结果。iPhone 利用多点触控来提供给用户一种深刻的直接操作的感觉。android 可以通过合理的使用单点触控来提供给用户大部分直接操作的体验。为了在你的程序里加强直接操作的感觉，要确保：

- 当用户在屏幕上操作对象时，那些对象仍是可见的。
- 用户操作的结果要立即可见

2.4 动画效果（Animation ）

在支持直接操作中，动画效果是很重要的。因为界面会像真实世界的物体那样予以反映，所以它会增加用户与设备的使用感。通过使用“狗耳朵”原则，可以给用户一种“突然感”。当一个狗停止跑动时会发生什么呢？它的耳朵会继续运行然后被反弹回来。要使你的用户界面有这种生动的感觉。比如说，当 iPhone 切换到另一个程序或者接一个电话时，会停止播放音乐。 另一个例子是 iphone 的 table views 和 android 的 list views 的不同。当用户使用滚动条到达列表的底端时，滚动条会突然停在 android 上，但是， 在 iphone 上，如果这时继续向下拖动滚动条，滚动条会有反弹的效果。iphone 提供真实世界的感受，但 android 没有，只是撞到墙上并立即停止。看起来是一件很小的事情，但在联系用户方面，却有很大不同。严肃的讲，使用 iphone 一会，你就会喜欢上动画效果。

2.5 看和点击

android 应用程序比人优胜的地方在于，它能更好的记住列表选项、命令、数据等等。使用列表格式中陈现选项，可以充分利用它的优势，使得用户可以很容易的浏览这些选项并进行选择。尽量减少文本输入。

2.6 用户控制

让用户（而不是程序）来触发和控制动作. 要使动作简单直接，使用户可以容易的理解和记住。不论什么时候，只要有可能，就使用用户已经熟悉的标准控制和行为。它们的关键在于提供给用户他们需要的功能，同时帮助他们避免危险和不可逆转的动作. 比如说，如果用户可能会突然损坏数据，那么你就要提示一个警告，但是如果用户他们选择继续，那他们就可以继续操作。

2.7 反馈和交流

在长时操作中，当用户操作时，他们需要及时的反馈和状态报告。你的程序应该提供一些可见的变化，这些变化根据每个用户的动作而变化。比如说，在列表中，当用户按下时，应该要高亮显示这个选项，使用户知道他们的触摸已被触发。动画效果是提供用户反馈的一种很好的方式。

2.8 容错性（Forgiveness）

要通过建立容错性来鼓励用户探索你的应用程序，就是说，使每个动作很容易可逆。当用户操作一项任务，当一项任务会引起不可逆操作而丢失数据时，要出现一个警告来提示用户。要能预期常规的问题，然后警告用户那些潜在的负面影响。

2.9 整体审美效果

整体美观是指信息被很好的组织，要根据视觉设计原则保持一致。也是关于整合了功能的应用程序的外观。外观对功能有很大影响。一个混乱或不合理的程序很难理解和使用。整体布局和用户界面元素

的设计都应该反映了用户使用应用程序任务时的心智模型。

2.10 控制你程序的复杂性

开发易用软件的最好的方法就是使设计尽量简单。你的程序任务越复杂，保持用户界面简单和被注意到就越重要。

3 Android 的交互设计

Google Android OS 是一个拥有很多有趣功能的独特平台。如果开发者们想让他们的应用程序与 android 体验融为一体，那么他们就应该尽量多使用这些功能。这部分将帮助开发者们更好的了解 android 平台的交互设计。

3.1 行为与任务

程序

一个 android 应用程序主要由包括一个或多个行为流成，这些行为是相关的，但没有严格的界限。

行为

行为是 android 程序主要组块. 你可以从你创造的行为 和其它 android OS 可用行为中装配一个应用程序。你创建的每个行为都应该被设计成只有一个用途，例如照相，查找通讯录，或读邮件。

显示用户界面的程序包括一个或多个行为。

当使用 android 设备时，随着用户在用户界面移动，就会一个接一个的触发这些行为，对他们来说应该是一个无缝的体验、一个行为接一个行为，一个任务接一个任务。这就像前面提过的行为流一样。

要记住到任应该连在一起来形成一个整体并且是联系的用户界面。如果你的行为不遵循基本的交互设计原则，与系统行为流关联时，那么用户可能会因为缺少连续性而感到困惑和沮丧。

一个行为处理一个特定的内容（数据）类型，并且接受一系列相关的用户行为。

行为流 Activity Stack

当用户从一个行为到另一个行为，通过程序，android 系统会以线性导航历史的形式记录用户已经访问过的行为。这就是行为流，也叫做返回流（back stack）。

总的来说，当用户开始一个新行为时，它就被加在行为流上，所以按返回键就会显示以前的行为.

然而，用户使用返回键只能返回到上一次行为直至首页，到首页后不能继续返回。

行为只是可以加到行为流 上的那些行为，但 views, windows, menus, dialogs 这些都不能添加到行为流。

任务

一个任务是一系列行为组成，使用这些行为来实现用户的使用目标，不论这些行为属于哪个应用程序的。

直到一个新任务被明确的定义，所有的用户开始的行为都被认为是目前任务的一部分。

启动任务的行为称之为根行为（root activity），一般来说，任务通常从应用程序启动、桌面快捷方式或切换到“最近的任务”

用户可以和启动任务时一样，通过从 root activity 选择一个 icon 来返回到任务。

中断任务

任务的一个很重要的功能是用户可以中断他们正在进行的任务，来进行另一个任务，同时，可以返回原来的任务，继续完成它。这就说，用户可以相继进行多个任务，然后在他们之间转换。

菜单设计

看菜单设计指南 http://androidappdocs.appspot.com/guide/practices/ui_guidelines/menu_design.html

icon 设计

看 icon 设计指南 http://androidappdocs.appspot.com/guide/practices/ui_guidelines/icon_design.html

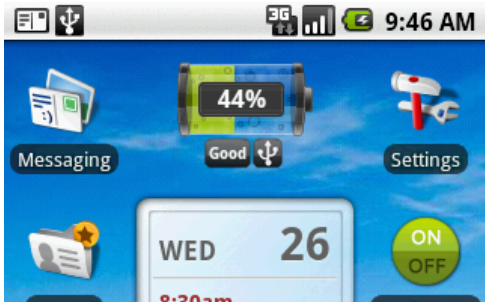
app widget design

看此部分指南 http://androidappdocs.appspot.com/guide/practices/ui_guidelines/widget_design.html

4 用户界面的设计

应该尽可能的使用 android 提供的标准界面元素，并且遵循建议的用法。用户习惯了标准视图和控制的外观和行为。如果你使用标准界面元素，当用户学习使用你的程序时，他们就可以依靠他们以前的经验来帮助他们学习。

4.1 状态栏



用法和行为 Usage and Behavior

状态栏包含了重要的信息，包括电池状态、时间、网络和信号强弱等. 它也会对用户显示提示图标。尽管你的程序可以隐藏状态栏 ，但是你应该认真考虑这种设计的结果。用户会期待可以在状态栏里看到信息。在 iphone 里，程序可以使状态栏变得透明，允许用户可以看到状态栏后的应用程序的窗口。在 android 中，你可以自定义状态栏的颜色。

建议

除非有充分的理由，否则不要隐藏状态栏。

利用提示系统，并在状态栏里显示提示的图标。

工具栏（toolbar）

4.2Tab Widget

用法和行为 Usage and Behavior

Tab Widget 会提供一个界面来导航不同视图。在程序里，tab 是可用的。它们可以作为应用程序除了菜单之外的另一种导航方式。在同一数据源中，或整个应用程序所有功能的不同子任务，每一个 tab 都应着不同的内容。在 Iphone 中，此功能不叫 tab widget ，而是叫 tab bar, tab bar 看起来要更友好，放在屏幕底部。

建议

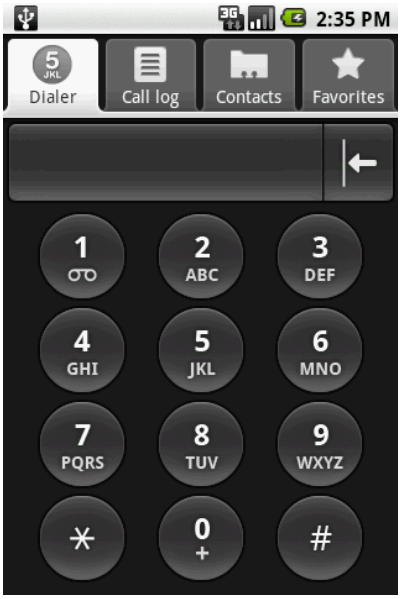
- bar widget 不同于 iphone 的 Toolbar，但这不意味着在列表页面必须要有 bar widget。
- 在你的 tab 里要用容易理解的灰阶图片。比如，android 的通讯录 app。
- 要文本结合图片，使得 tab 的功能更容易理解。要确保文本很小，可以合适于 tab.
- 是 tab 数量最好是 4 个，多于 4 个会使得读和点击变得不容易。如果程序只是在横屏下使用，那么就要多加几个 tabs.
- 如果你使用 tab，那么它们最大程度上对用户是可用的。如果在你的程序里移动它们，那么就会使界面变得不连续，用户会感到困惑。

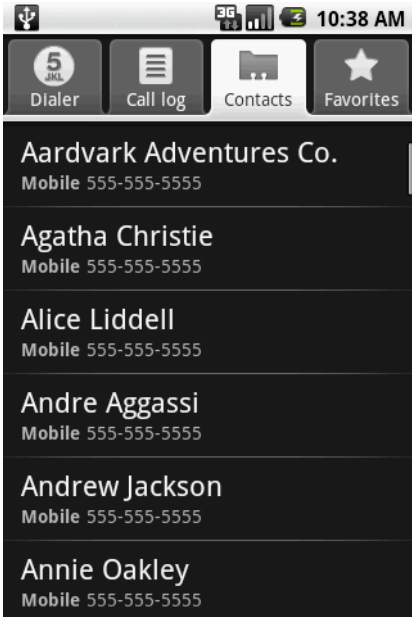
例子

Code: HelloTabWidget

http://binarysheep.com/AndroidCode/HelloTabWidget/

Application: Contacts

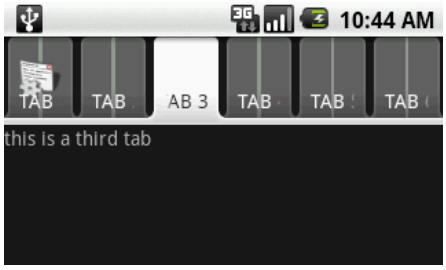




通讯录 app 充分利用了 Tab Widget. 每个 tab 都有一个简单的灰阶图片 和 t 文字，使得用户可以很快理解它们的功能。

注意这些 tabs 使用 dialer, call log, contacts ,favorites 这些暗喻与用户心智模型连接非常匹配。

Sample: Too Many Tabs

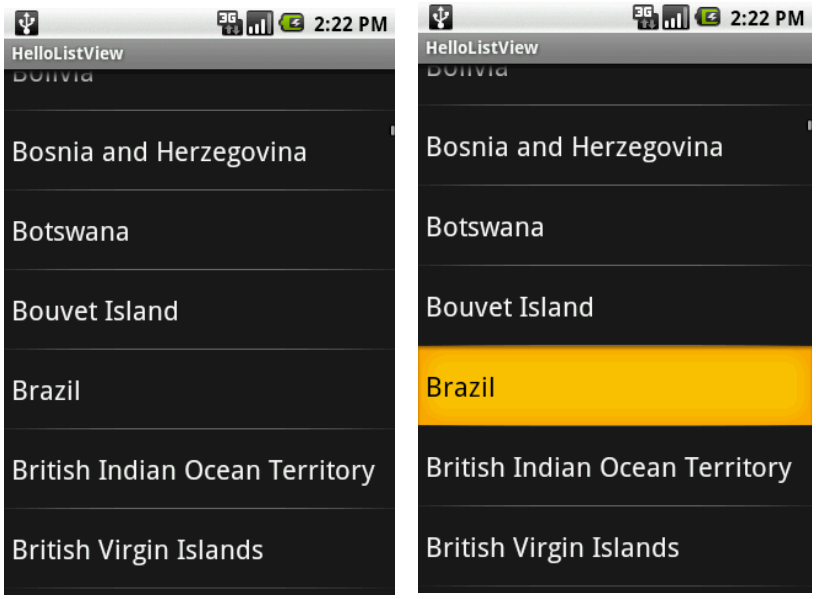


除非这个程序只在横屏模式里用，否则 tab 太多了。用户很难点击 tab 或明白 tab 的作用。

Sample: Tabs with List Views

???

4.3 列表视图



用法和行为 Usage and Behavior

Mainstay of Android programming. Can be used for drill down interfaces. Hook up to?

一个列表视图以单列多行的方式显示数据。每一列都包含了文本、图像和 Controls。列可以被分成几组。列表视图很有用，因为它们可以组织大量的数据。当用户选择了一个项目时，就会提供反馈。列表中的选项以短暂的高亮显示来显示已被选中，然后所选中的功能被触发. 你也可以注意到在上面显示的列表中，它的顶部使用了边缘的淡化效果（fading edges）. android 使用这项技术来显示当前页面是可以滚动的。列表视图是多功能的，可以被很多不同的用户界面使用。

- 选择项

  列表视图可以显示一列选项，以供用户选择。一个选中标记图像可以显示目前列表中被选中的项。

- 导航的分层 信息

  列表视图也可以被用来显示分层次信息，每一列都包含自己的子信息。Apple 把这项功能作为一个 drill down 界面，经常在 iphone 里使用。遗憾的是 android 没有像 iphone 那样的导航，可以在列表图顶端，分层 drill down 使导航变得容易。

- 查找索引信息

  列表视图可以用来根据一个标准来按顺序显示信息（比如根据字母）

- 按概念把信息分组

  我们可以把信息分组（例如工作、家庭、学校等）

建议

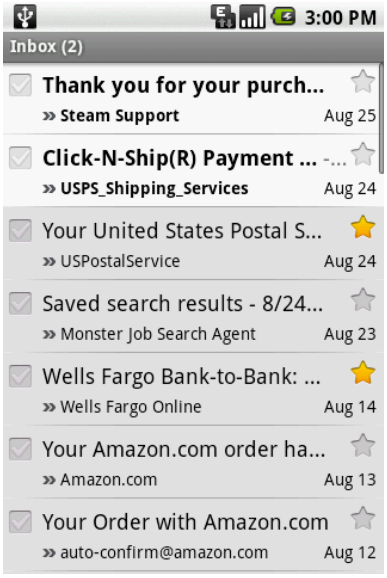
- 当选中信息时，用标识图片（checkmark image）来表示，而不是把整列都强调。
- 记住在竖屏模式下，list view widget 显示更多列。
- 在触摸模式里，没有当前选中的项目（current item selection）
- 如果每一列都需要显示综合信息项，就要考虑用三栏布局。

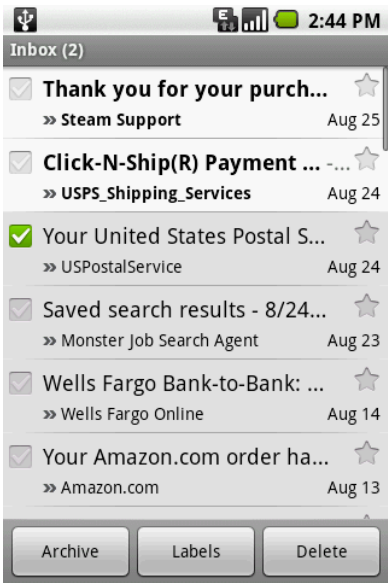
示例

Code: HelloListView

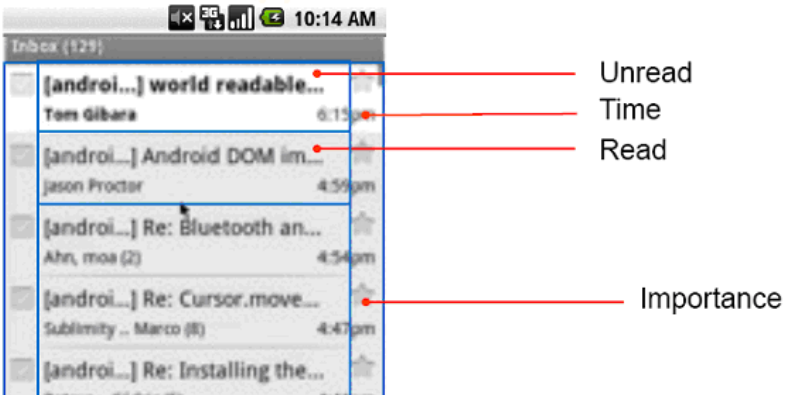
http://binarysheep.com/AndroidCode/HelloListView/

Application: Email

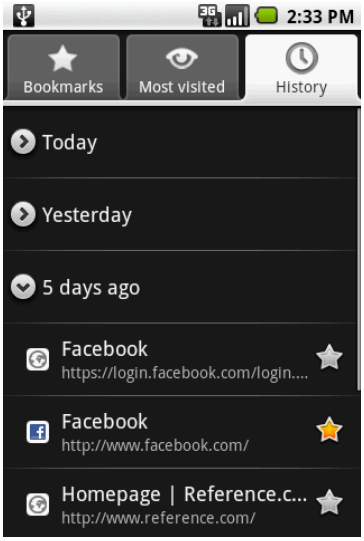




google 在 list view widget 方面做得很好。一个三列的布局用来提升程序的可用性，而没有使用零乱摆放的设计。在第一栏中可以激活一个多任务模型，会弹出一个三个按钮的工具栏。这个工具栏只有当列表视图里一个或多个选项背选中时才会出现。它可以允许用户在邮件箱里很快执行一个动作。用户也会发现推（push）菜单按钮会提供跟多邮件相关的选项。对于那些通过点击 checkbox 而发现它的用户来说，这些功能是很好的。第二栏显示标题和邮件的发送者。你会注意到已读和未读的信息，它们的颜色是不一样的。颜色是一种很好的方式来显示更多的信息，而不用使用任何屏幕的硬件 .第三栏允许用户提供一个更重要的邮件选项。它也会显示邮件信息收到的时间。

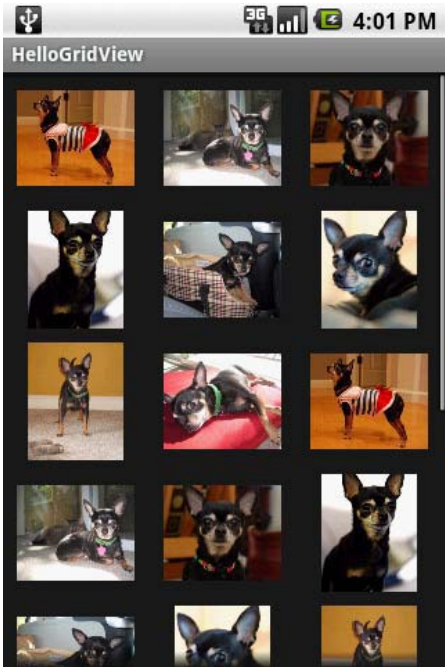


Application: Browser History



浏览器中的历史标签时有可能的历史数量非常多。Google 用一个可扩展的列表来精简页面和帮助用户更容易的导航。你也可以注意到他们使用和 Gmail 程序里相同的三栏布局。

4.4 网格视图（Grid view）



用法和行为 Usage and Behavior

表格视图是用二维滚动网格来显示项目。这些的菜单选项从列表适应器（ListAdapter）中获得。

建议

In development…

4.5 Spinner

用法和行为 Usage & Behavior

Spinner 是支持下拉列表最理想的 widget。当用户想在 spinner 中设置值时，他们可以点击 spinner widget 中的任意地方。列表视图格式里就会显示列表值。在使用 spiner 时，会将用度怕注意从原来页面吸引过来。为了帮助用户意识到他们还没有离开 UI，spinner 会在屏幕上显示列表栩栩如生，悬浮在 UI 的顶端。用户可以通过点击列中的任意位置来选中他们想要的值。然后 spinner 列表会返回到原来页面，被选中的值就会在 spinner 中显示。

建议

- 如果列表值不明显，就加一个题目来提醒用户。
- 不要多于 16 个值，这个用户就可以只用手指来滚动三次或少于三次。

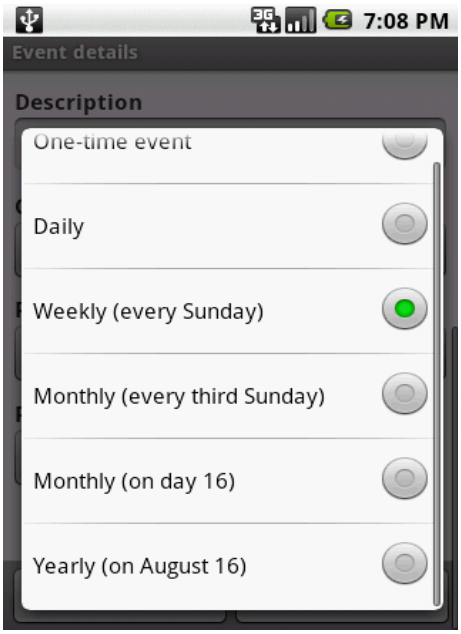
Examples

Code: HelloSpinner

http://binarysheep.com/AndroidCode/HelloSpinner/

程序：日历

当在日历上加一个新项目时，用户点击 Reminders Spinner，会出现一列可能的值。因为某种原因，在 Calendar UI 里的下一个 spinner 没有主题。在 android 里，如果没有标题的话，那列表值就不会扩展区填充标题空间。所有 spinner 都有标题或没有标题，这是保存标题一致性的好方法。



Auto Complete

5 开发过程

STEP 0 – 阅读设计规范

STEP 1 – 决定开发什么

- 创建一个应用定位说明
  - 描述成一个解决体系，而不是功能的罗列
  - 总结应用的设计目标
  - 定义主要的用户群
  - 用来指导发展和筛选功能
  - 示例：iphoto 的应用定位说明

    Desktop: 对那些业余的摄影者来说，方便数码照片进行编辑、组织和分享

    Iphone: 对iphone用户来说，方便使用，易使用数码照片进行分享

- 了解你的用户
  - 这个应用是为哪些用户设计的？

    选一个小的用户群（或单个用户）

    创建一个带有目标用户细节的人物角色，这个人物角色就像你自己一样。（例如：这个人是谁、典型的一天是怎么样的、他们是怎么操作任务的）
  - 开发你的程序要以用户和他们的能力为依据，而不是电脑。
  - 与你的用户交流这个非常重要

    让用户参与你的设计每个过程

    进行用户观察
  - 设计一个成功的程序的最好方法就是为自己建一个程序。
  -
- 提炼基本功能
  - 每个功能的基本描述
  - 选择尽量少数量的功能

    使用应用定位说明进行过滤。

    选择数量尽量少、但大多数用户频繁使用的功能，并且是适当移动状态下使用的。

STEP 2 – 访问APP Store或 Android Market

- 分析类似市场上相似产品的目标用户
  - 那些是否竞争产品，或与你的应用功能是否有交叉？

- ✧ 那些竞争产品是否有或高或低转移成本？
- 确定已经确立的使用规范
  - ✧ 移动用户的注意与学习跨度小
  - ✧ 开发的应用要容易学习，支持已经学习的使用方法
- 学习好的应用

STEP 3 - 探索可能的解决方案

- 分析和定义用户的心智模型
  - ✧ 发现用户使用应用相关任务时的心智模型或概念模型
  - ✧ 在电脑上用户怎么操作相似任务？
  - ✧ 用户在操作任务时，会有哪些概念、目标和手势？
- 应用human interface design principles
- 列出已经扩展功能的列表
  - ✧ 给每个功能一个具体的描述
- 尽量少，好的设计是一个解决方案，而不是一些功能。
  - ✧ 为你的80%用户设计，而其它用户可以进行自定义
  - ✧ 使用应用定位说明和人物角色过滤功能
  - ✧ 使用用户测试来获得反馈，发现哪些是功能是缺失的，哪些功能是多余的。
  - ✧ 好的应用是：各个功能相互融合，是提供了一个解决方案
- 不以迷恋第一个设计
  - ✧ 第一个设计绝不是一个最好的设计，尤其当你对这个平台是一个新手的时候。

STEP 4 - 绘草图

- 为你的应用设计10不同的方案草图
  - ✧ 最后两三个是比较难想得出的，但最有创意的想法往往来自于哪几个
- 使用那些草图来获得目标用户或朋友的反馈
  - ✧ 这样可以帮助你从10设计方案中选择出几个好的方案
- 质量来自于数量
  - ✧ 通过多个设计方案，帮你更快速的获得更好的设计方案

STEP 5 -使用Omnigraffle 画原型图

- 使用最佳的尺寸大小（像素）进行页面布局\
  - ✧ 开发者会精确地知道要创建多大

- 纸面原型，一个张纸呈现一张页面
  - ✧ 用户可以在使用纸面上的应用
  - ✧ 使用纸面原型进行观察性用户测试
- 原则：Fail early to succeed sooner

  注意：Omnigraffle只支持MAC系统的软件，也可选择使用Fireworks、Photoshop、Visio等相似软件。

STEP 6 –Do it all again

- 将“好的”扔到一边，重新开始（Its ok to throw it away and start again ）
- 重新设计，这样可以避免重新写代码
- 投入整个开发时间的60%到设计工作，目前Apples是这样做的
- 原则：Remember that nothing is precious

STEP 7 –代码开发

STEP 8 –beta测试

- 提交应用前进行测试
  - ✧ 你的应用只会在最新应用中出现一次
  - ✧ 如果因为小的Bug导致用户对你的应用评分较低，这个情况是很难去改变的
- 哪些人参与Beta测试？
  - ✧ 朋友
  - ✧ Amazon Mechanical Turk (https://www.mturk.com)
- 原则：提交前进行测试

STEP 9 –发布

- 准备进行维护与Bug修复
- 提炼应用的差异点
- 根据用户反馈进行迭代设计（必须准备好做这个事）