Android 4.0 设计规范

优先导读十大改变 & 规范全文翻译

在拜读和翻译了 Android design 设计指导后，对比 Android 4.0 与 Android2.3 及之前版本的 app 设计指导，总结了 Android 4.0 设计的 10 大改变：

1. 导航栏 (详见模式 PATTERNS>导航 Navigation)

由之前的物理按键导航（返回、菜单、搜索、主页）变成了嵌入屏幕的虚拟按键（返回、主页、最近任务）。



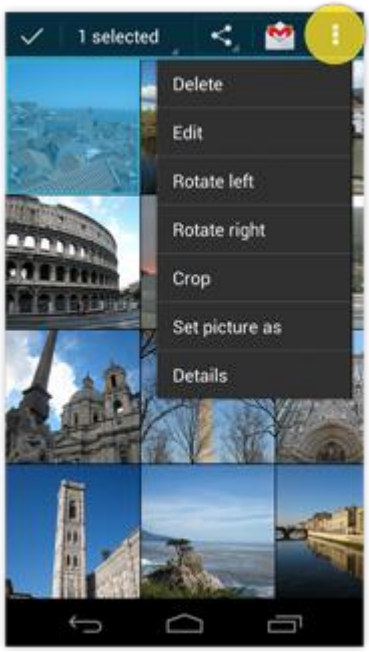
左侧为早期有 4 个物理按键的手机，右侧为新版只有 3 个虚拟按钮的手机

把菜单项和搜索项从导航栏去掉，把之前通过长按主页键才出现的最近任务直接展示在导航栏中。



最近任务的界面

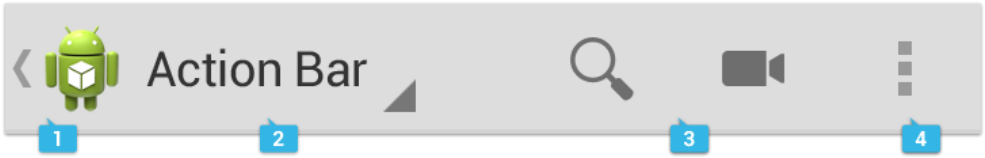
把菜单收起的同时，在软件界面的操作栏上增加了“更多操作”（action overflow），如下图中的黄色圆点所示。



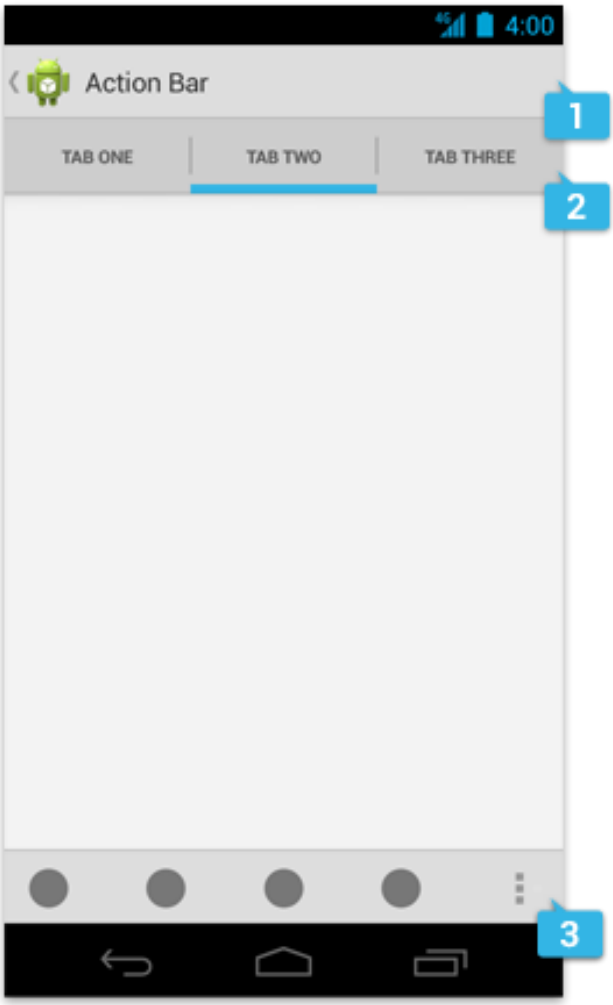
黄色圈部分为“更多操作”（action overflow）的位置和点击后内容的展示方式

2. 操作栏 (详见模式 PATTERNS>操作栏 Action bar)

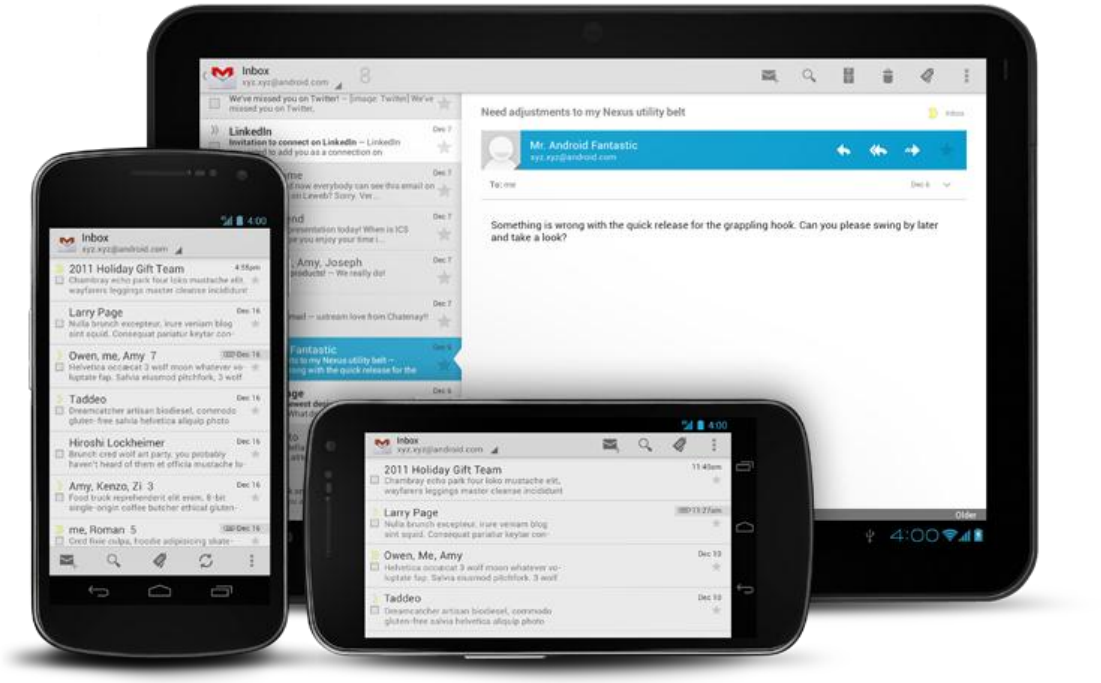
操作栏从之前的 app 图标＋操作，变成如下图所示：
1 向上＋2 Spinner＋3 重要操作＋4 更多操作。
其中，向上按钮，点击后是去到当前界面的上一个层级，非第一层级界面有此按钮，第一层级界面则无向上按钮；Spinner 是用于展示内容的下拉菜单，其内容包括视图的快速切换和显示相关内容的完整信息；更多操作（action overflow）是集合操作栏中不常用的和非重要操作的地方。



操作栏可以拆分成下图中的 1 主操作栏、2 顶部栏和 3 底部栏。如果需要，顶部栏可以承载选项卡 tabs，底部栏可以承载主要操作和被收起的更多操作（action overflow）。

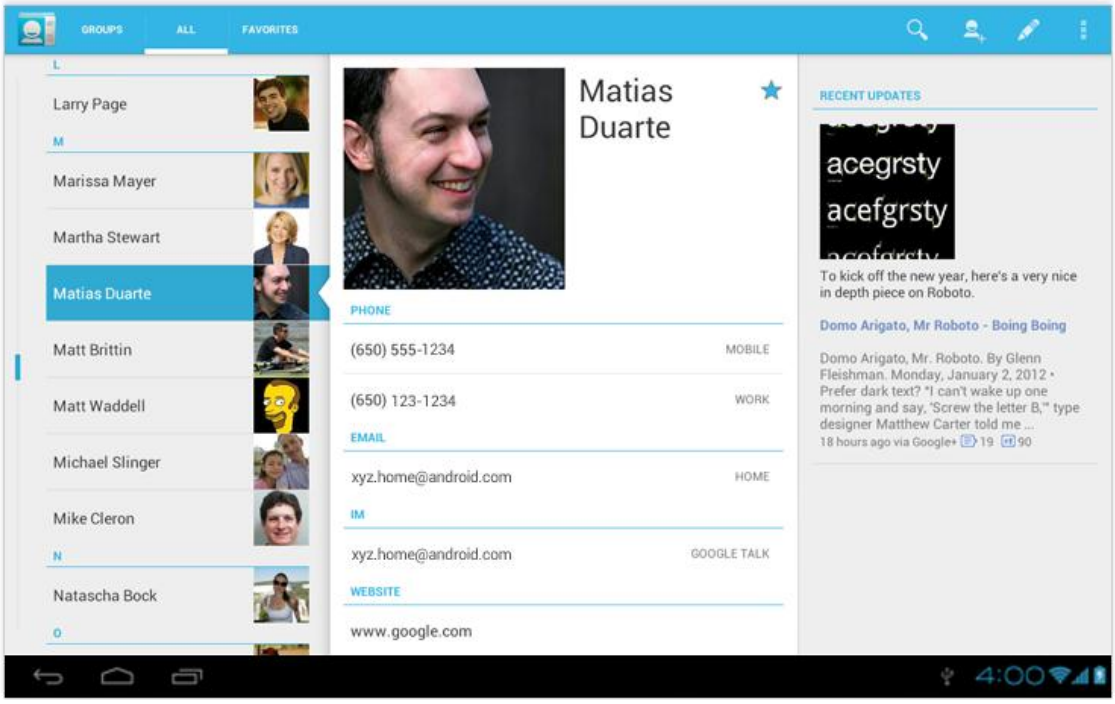


在空间足够的横向屏幕展示界面时，被拆分的操作栏，可以灵活合并在一起。如下图中的手机端横竖屏操作栏所示。



3. 多面板布局 (详见模式 PATTERNS>多面板布局 Multi-pane layouts)

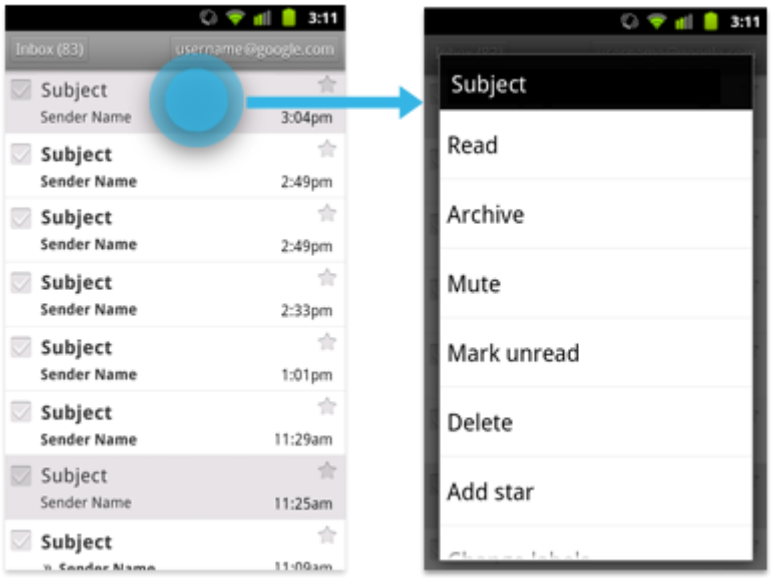
多面板布局更多的是针对平板电脑，把手机端的目录视图和详情视图两个层级的界面，甚至更多的页面，复合展示在同一个界面中，有效地利用平板电脑的屏幕空间， 扁平化层级结构，简化导航。这点在 iPad 上已经运用得相当娴熟了。



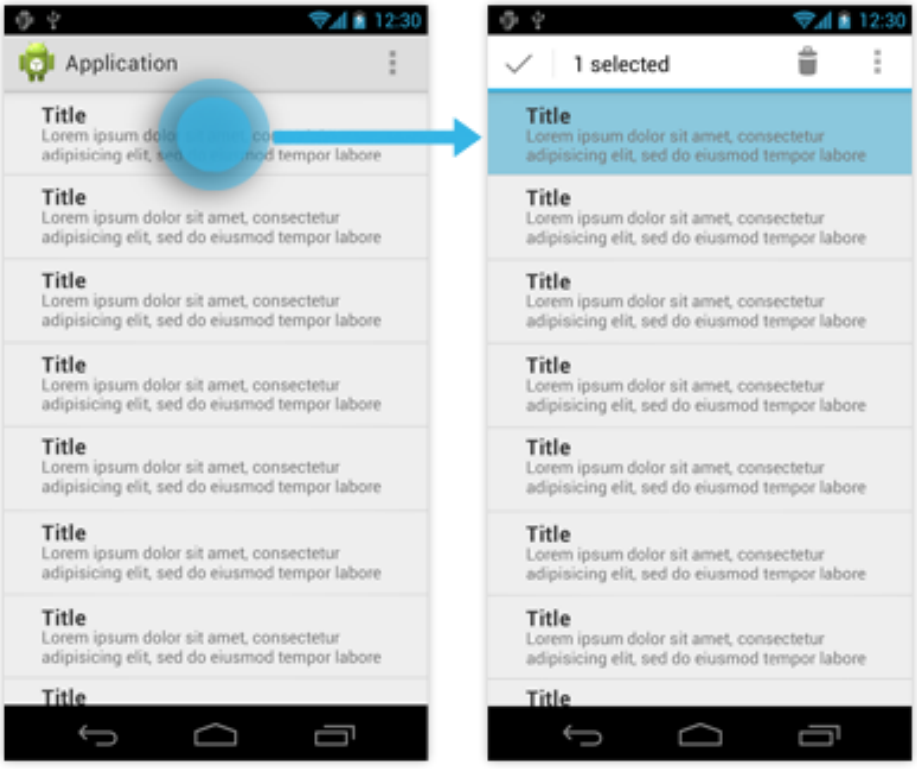
4. 选择 (详见模式 PATTERNS>选择 Selection)

Android4.0 中的长按与 Android2.3 及更早期的版本有很大的不同。早期版本长按操作后，是出

现情境菜单的浮出层。在 Android4.0 中，长按后在操作栏的位置会覆盖一个临时的情境操作栏，不再弹出情境菜单浮出层。在临时情境操作栏的环境下，当前界面的内容项允许被单个处理，也允许被批量处理。



长按 Android2.3 及更早期系统版本的内容项时，出现情境菜单的浮出层。



长按 Android4.0 系统版本的内容项时，情境菜单栏覆盖在操作栏上。

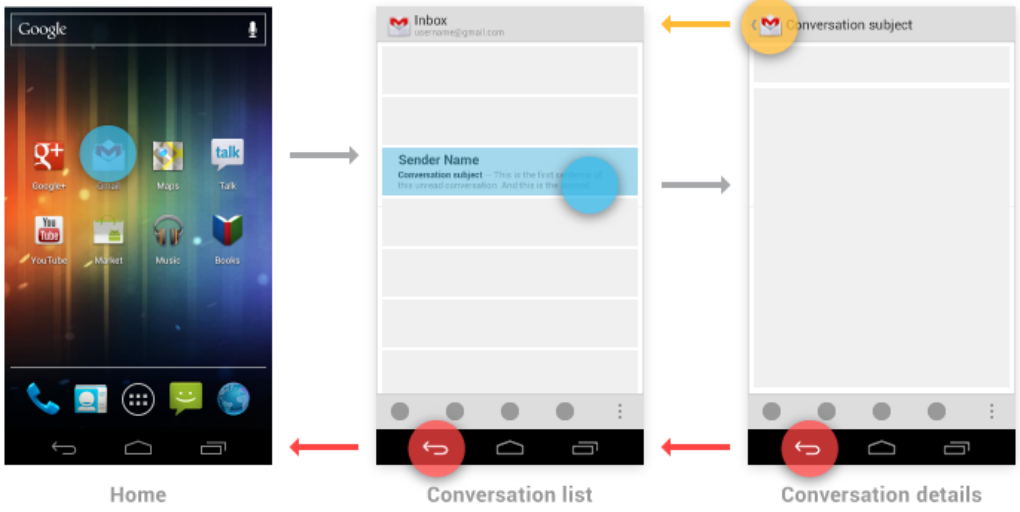


在情境菜单环境下，支持批量操作。

5. 返回和向上 (详见模式 PATTERNS>返回和向上 Navigation with Back and Up)

返回按键用在手机全局的虚拟导航栏中，基于用户最近查看的界面历史，采用时间倒序的方式，连接界面间的关系。向上按钮用在操作栏的左侧，基于层级结构，点击后是去到当前界面的上一个层级，若当前界面已经是最高一级，则没有向上按钮。

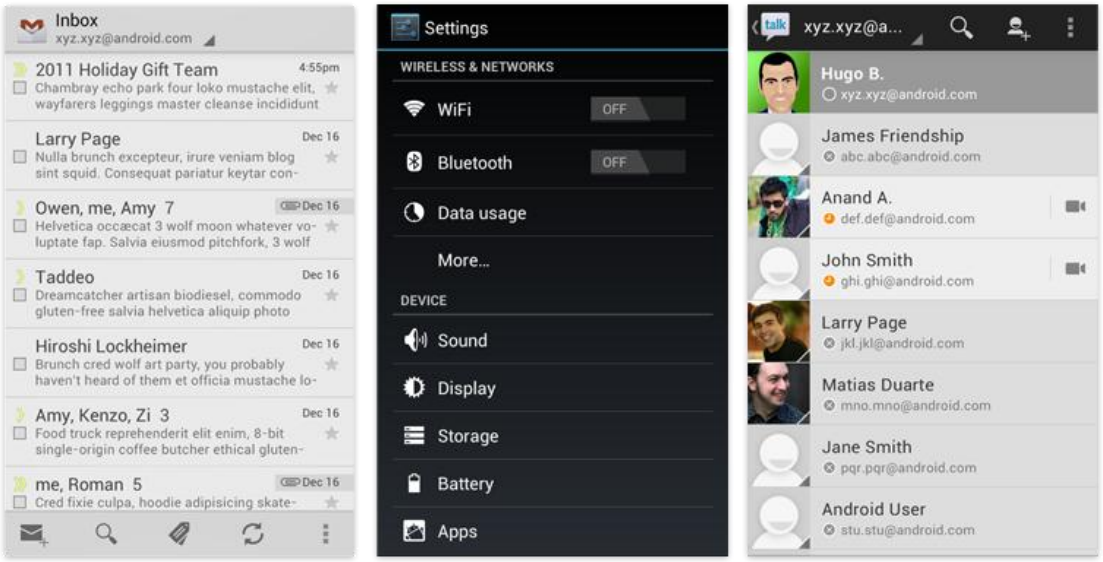




黄色部分为点击向上按钮后的页面路径，红色部分为点击返回按钮后的页面路径

6. 主题样式 (详见风格 STYLE>主题样式 Themes)

推出三套默认主题：Holo 浅色主题、Holo 深色主题、Holo 浅色底+深色操作栏主题。主推 app 在这三套默认主题的基础上做设计，以加快 app 研发效率，但只是建议使用，并没有完全强制。视觉设计师们可以重点看下 STYLE 和 BUILDING BLOCKS 这两章。



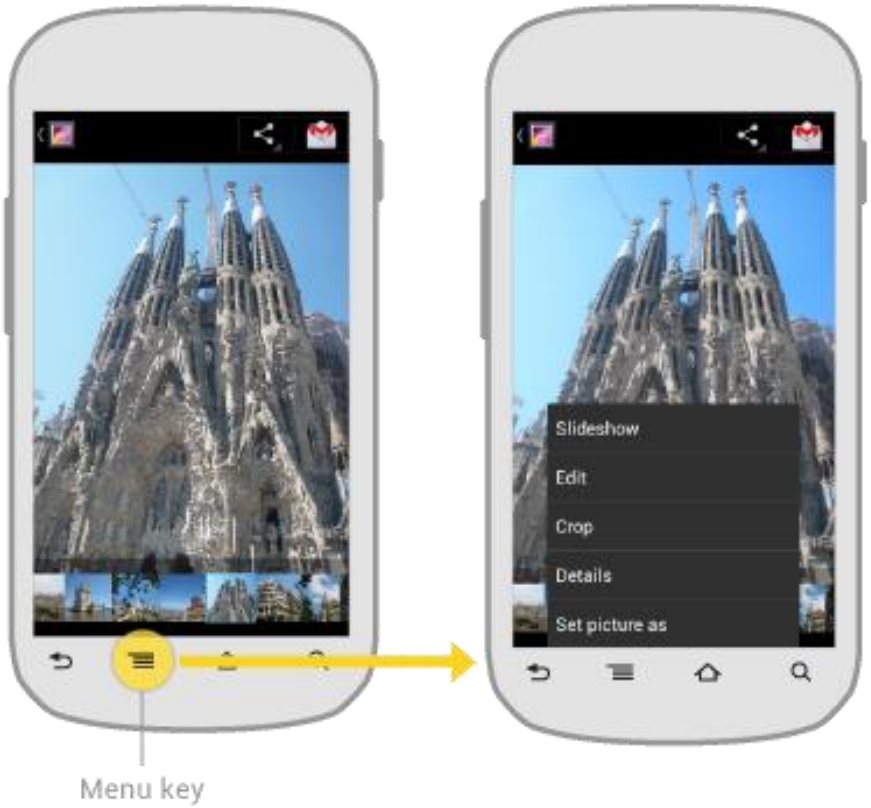
7. Widgets (详见开始吧 GET STARTED>UI 概览 UI Overview)

在原 apps 页里，用选项卡 tabs 的方式增加了 widgets 内容。一改用隐晦方式添加 widget 的操作和把 widgets 零散放在桌面呈现的方式。在 app 设计中，应该提高对 widget 设计的重视程度。



8. 兼容性 (详见模式 PATTERNS>兼容性 Compatibility)

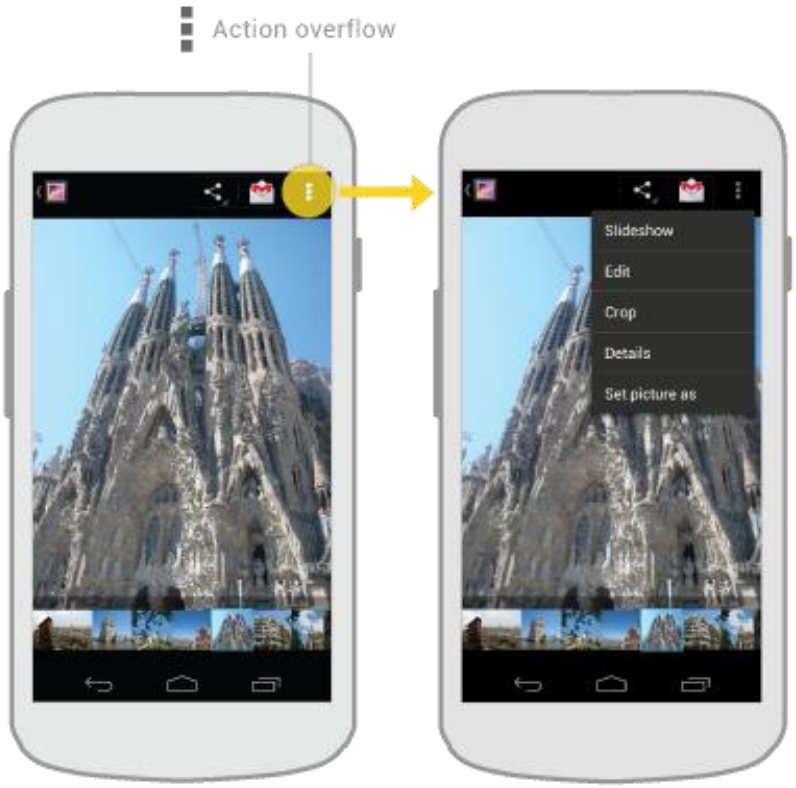
向后兼容，考虑到物理导航按键手机如何兼容 Android 4.0 系统和虚拟导航手机如何兼容 Android2.3 和更早期版本的 apps。



在物理导航按键手机上安装 android4.0 系统，点击物理按键，在屏幕下方出现更多菜单 action overflow 的内容。



在新的虚拟导航手机上安装为 android2.3 及更早版本系统设计的 app，会在屏幕下方的虚拟导航的右侧出现 action overflow 按钮（如上图的黄色部分）。



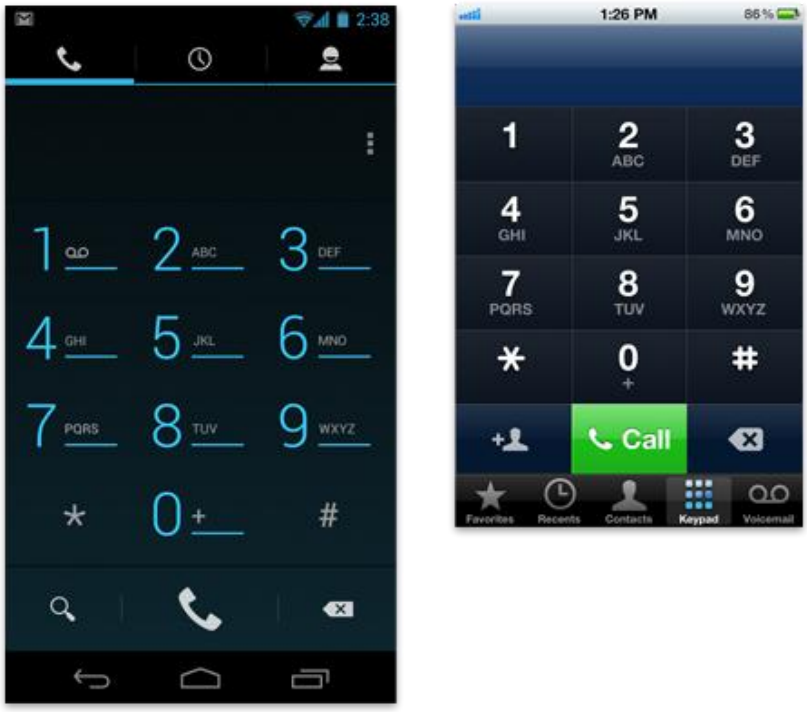
在新的虚拟导航手机上安装 android4.0，点击 action overflow 后的展示状态如上图。

9. 强调纯粹的 Android 应用设计 (详见模式 PATTERNS>纯粹的 Android)

在 Android4.0 app 设计准则中，特别强调为 Android 设计纯粹的 Android app，切勿使用其他平台特定元素的注意事项，有以下 5 个方面：

（1）强调视觉元素的样式要符合 android 系统；
（2）不用其它平台特有的图标；
（3）不在界面的下方使用选项卡 tabs；
（4）区分向上和返回，不在操作栏上使用返回样式的按钮；
（5）不在内容列表里使用向右箭头。



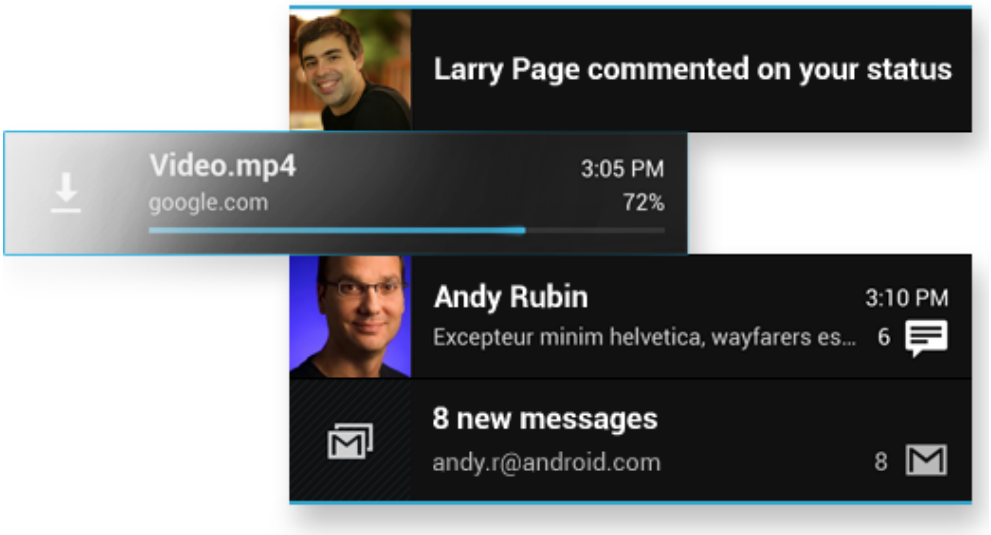


Android 系统里，选项卡（tabs）会固定放在屏幕上方，不会放在下方。这点与 ios 系统不同。

10. 其它细节

增加许多新的交互细节、信息展示和视觉样式等规范，详细见翻译。其中，有几点在此强调下：

（1）新增了横滑移除内容的交互手势。在部分模块中，支持向左或向右横滑移除内容的操作，如最近任务和消息通知抽屉。



（2）视觉的平面化，栅格风正在 4.0 中上演。

（3）在 writing style 中明确指出写 wording 时，要直接使用第二人称“你”。

以下部分为规范全文翻译。

Android Design



目录

开始吧 GET STARTED 13
创新的视觉 13
设计原则 14
UI 概览 20
设计风格 STYLE 23
设备和显示 23
主题样式 24
触摸反馈 25
标准和网格 26
字体 28
颜色 30
图标 30
写作风格 35
模式 PATTERNS 36
新的 Android 4.0 36
手势 37
应用结构 40
导航 48
操作栏 54
多面板布局 61
横划视图 64
选择 66
通知 68
兼容性 73
Android 就是 Android 76
积木/控件 BUILDING BLOCKS 79
选项卡 Tabs 79
列表 80
网格列表 81
滚动 83
Spinners（下拉菜单） 84
按钮 85
文本输入 86
滑块 87
反馈 88
选择控件 90
对话框 91
选择器 93

http://developer.android.com/design/index.html

开始吧 GET STARTED ▶

欢迎来到 Android Design，这里是你学习如何设计 Android Apps 的地方。



创新的视觉

雪糕三明治（Android 4.0）是 Android 设计的里程碑。我们通过扩展应用在蜂巢平板电脑和其他移动设备的新的设计方式，触碰到几乎系统的每一个像素。从最为基本的元素开始，我们使用了一个专门为高分辨率显示而设的新字体：Roboto。其他的大改变包括设备里框架级的操作栏（物理按键），及支持没有物理按键的新设备。

我们把核心 apps 和整个系统的设计焦点放到三个大目标上。在你设计 Android apps 的时候，请你也考虑这三点：

1. 迷惑我吧

漂亮不只是外表上的。Android apps 在多个层次上都漂亮且具有美感。它的转场快速清晰；排版和样式干脆利落及具有意义。App 图标本身就是件艺术品。就像一个制造精良的工具，你的 app 应该追求漂亮、简洁，并创造一种神奇的体验，这些都可以使你所向无敌。

2. 简化我的生活

Android apps 能让生活更简单且容易理解。当人们第一次使用你的 app 时，他们应该凭直觉就能了解到最重要的功能是什么。但不能只为用户的第一次使用而设计，Android apps 把正在进行的后台杂务如文件管理和同步都隐藏。简单的任务从来不要求复杂的步骤，而复杂的任务会为人量身定做。任何年龄与文化的人都可以感觉到很好的操纵感，不会因过多选择和无关紧要的动画而感到不知所措。

3. 让我感到惊艳
只是让 App 易于使用还不足够，Android apps 希望人们能不断尝试新东西和用创意的方式来使用 app。Android 让人们通过多任务、通知和跨 app 分享，把 apps 结合到新的工作流中。
同时，你的 app 还应该是很个人化的，让人们通过清晰和优雅的方式享受到伟大的科技。

设计原则----------

为了保持用户的兴趣， Android 用户体验设计团队设定了以下的原则。把它当做是你自己的创意和设计的思想。除非有恰当的理由，否则不要偏离它。

让我着迷

1. 惊喜
一个漂亮的界面，一个精心准备的动画，或一个适时的声音效果都是体验的乐趣。微妙的效果都可以贡献出强大的力量。



2. 真实对象比按钮和菜单更有趣
让人们直接触摸操作你 app 里的对象。它可以减少执行任务的认知负担，同时可以更多地满足情感需求。

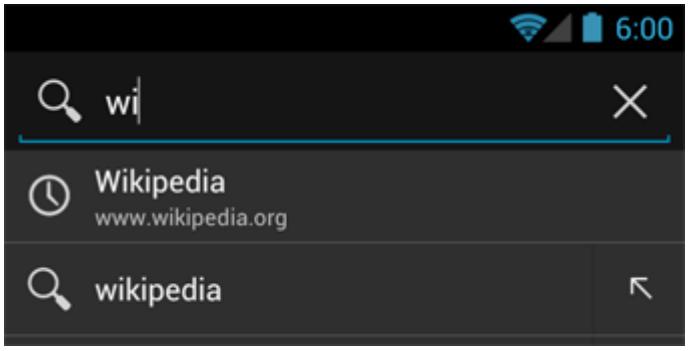


3. 个性化
人们喜欢增加个人的东西，它们让人感觉更有亲切感及控制感。提供实用、漂亮、有趣、可自定义，且不妨碍主要任务的的默认设置。



4. 记住人们的习惯

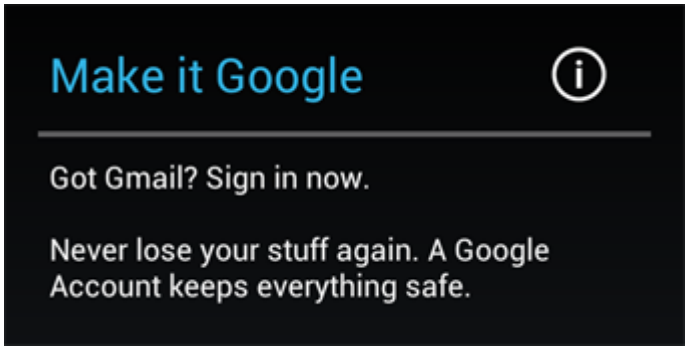
学习人们的使用习惯。跟随用户的使用行为，比一遍又一遍的重复问要好。



简单我的生活

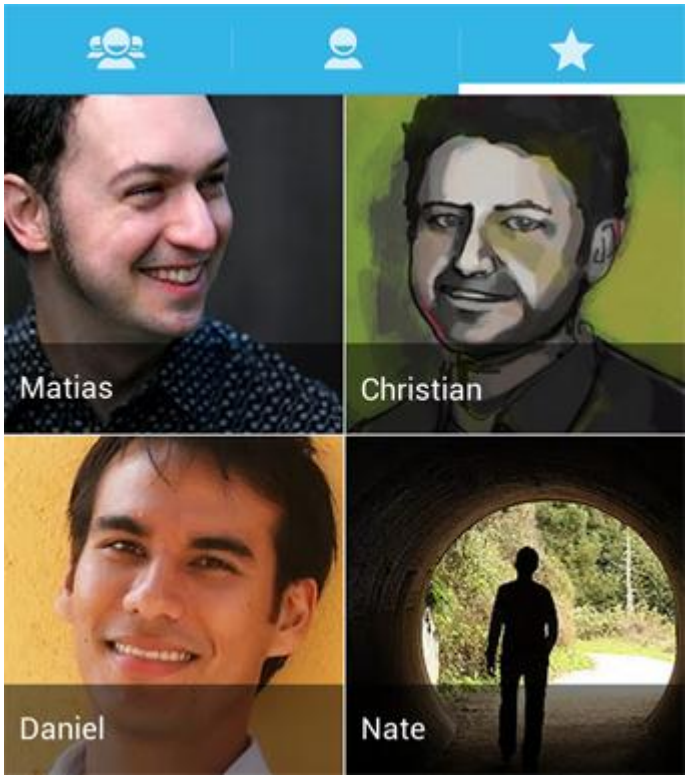
1. 尽量表达简洁

使用简单词、短句子。人们看到长句会不自觉的跳过。



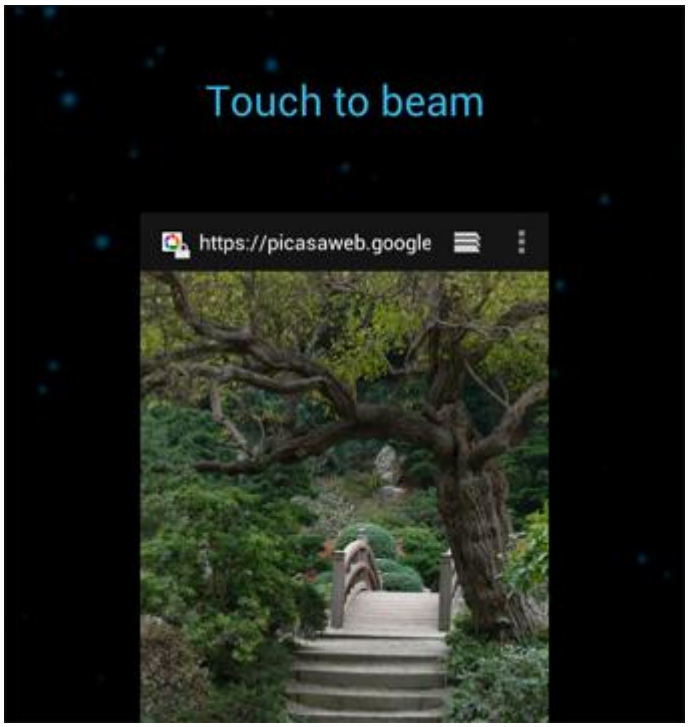
2. 图片比文字好理解

使用图片来解释想法。图片获得人们更多的注意力，比文字更有效率。



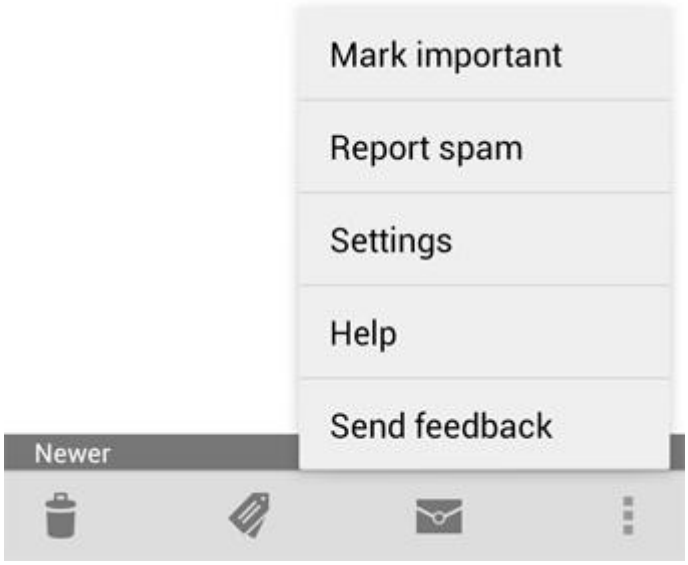
3. 为我作决定，但要让我有决定权

尽最大努力去猜而不是什么都问用户。太多的选择和决定会让人们不爽。为了预防你的猜测是错的，要提供后退。



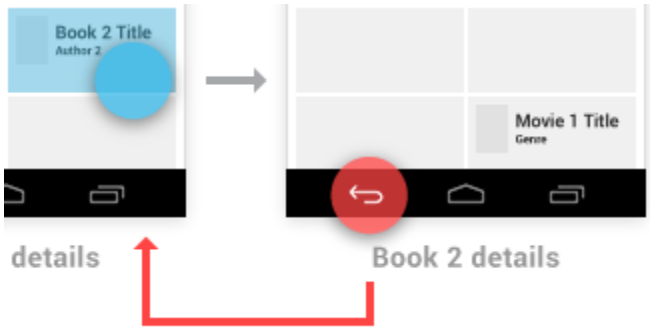
4. 只在需要的时候显示我需要的内容

人们看到太多选择会不知所措。把任务和信息打散成一个个小的、容易操作的内容。隐藏此时不必要的操作选项。



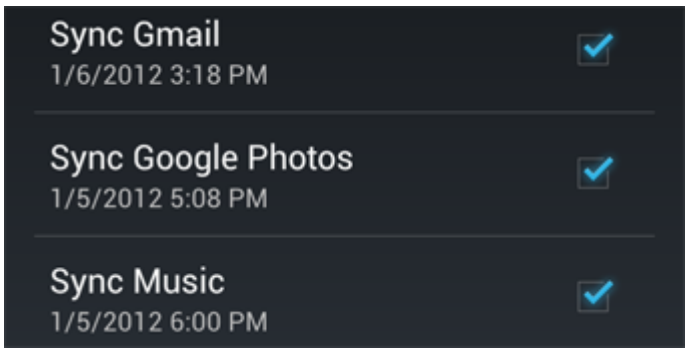
5. 让我知道我在哪

知道自己在哪可以给人们自信。让你的 app 每页看上去都有区别，使用转场显示各个屏之间的关系。在任务进程中提供清晰的反馈。



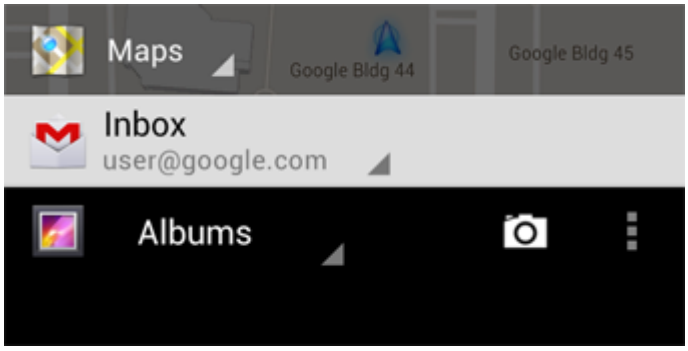
6. 永不丢失我的东西

为人们保存他们花时间创建的东西，并在任何地方都可以获取它们。记住设置，个性化触控，及创建电话、平板电脑和电脑之间的同步。这可以使升级成为世界上最容易的事。



7. 如果看上去一样，就应该操作也一样

为了更好的分辨功能，让他们视觉上的区别更大一些。避免那些看上去差不多的样式但操作起来却不一样。



8. 真的很重要时才打断我

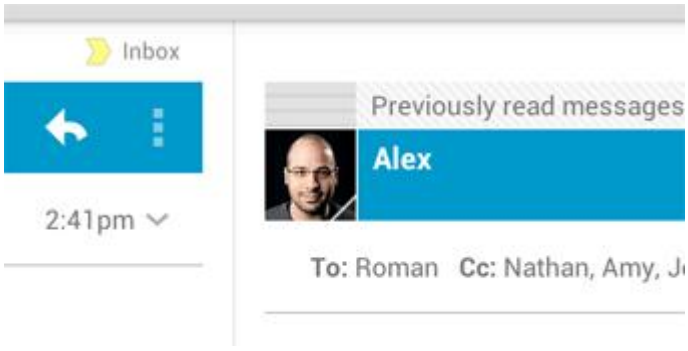
你要像一位贴心的助理一样，帮用户挡住那些不重要的骚扰。因为打断会令人费神且沮丧，所以人们希望保持专注，除非是非常重要和讲求时效的事情，才愿意被打断。



让我惊喜吧

1. 给我一些可以通用的操作方式

当人们自己搞懂一些事情的时候会感觉很赞。利用其它 Android apps 已有的视觉样式和通用方式让人们学习你的 app 变得很容易。例如，横划就是一个很好的导航的快捷切换方式。



2. 这不是我的错

当你让用户改正时要温和些。用户在使用你的 app 时会期望它很智能。如果出了问题，给出清晰的恢复指引，而不是详细的技术细节。当然，如果你可以在后台解决会更好。



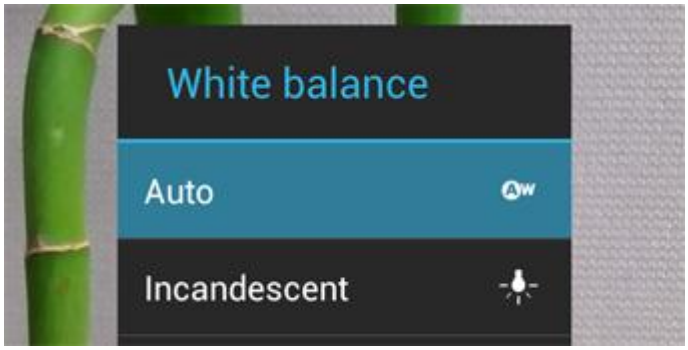
3. 不断地鼓励

把复杂的任务分拆成容易完成的一个个小步骤。即使操作有了很小的改变也要给出反馈。



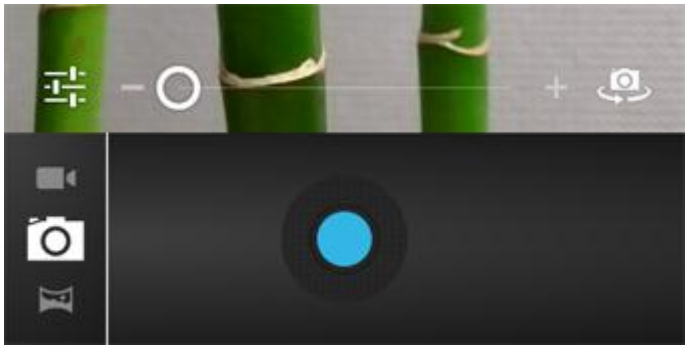
4. 帮我完成那些繁重的活

让新手顺利完成一些他们也没有预想自己能完成的任务，这可以让他们觉得自己也是个专家。例如，提供照片滤镜效果，只需通过几个简单的步骤，使一个非专业的照片看上去也很棒。



5. 让重要操作快起来

你的 app 中不是所有的功能都一样重要。决定哪些才是最重要的功能，让他们容易发现并且使用起来非常灵敏。就如照相机的快门和音乐播放器的暂停。



UI 概览----------

Android 系统 UI 提供的框架非常重要。其中包括了主界面（Home）的体验，设备的全局导航及通知栏。
你的 app 在保持 Android 体验的一致性和使用的愉快度上，有重要的作用。这一章的结尾，我们介绍了你的 app 要达到这个目标的实现要点。

先快速看看这些最重要的 Android 界面：
Home，全部 app 界面，及最近界面



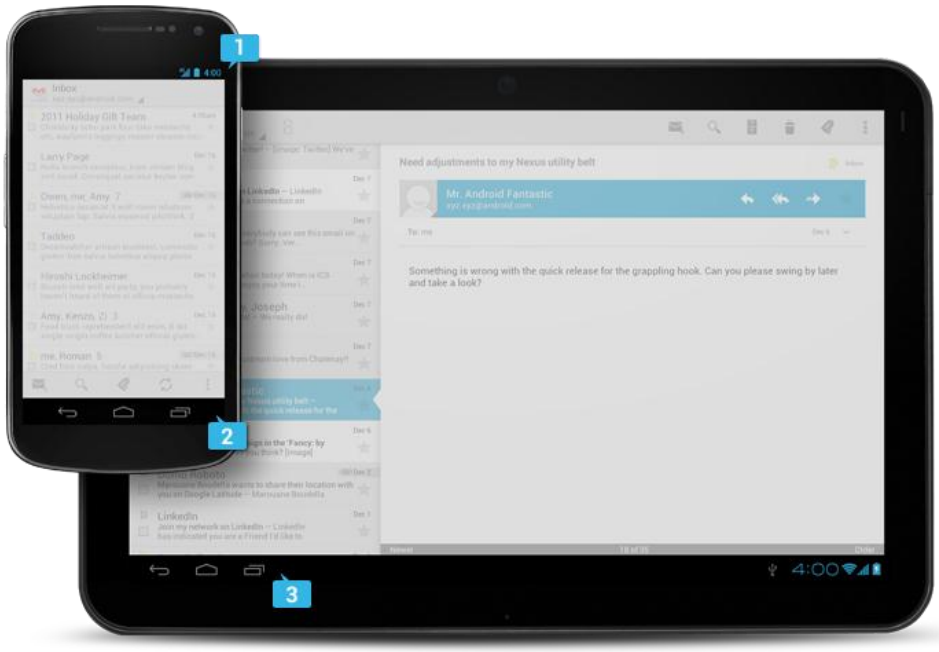
1. 主界面 Home
Home 是一个可定制收藏 app、文件夹和小工具的地方。通过左右横划来导航不同的 home 屏面板。
无论你横划到哪个 home 屏面板，底部始终有一栏为“我的最爱”，可以把最重要的 app 和文件夹放在这里展示。
点击底部栏“我的最爱”中间的“全部应用”按钮可以到达全部 app 和小插件的展示页面。

2. 全部 app 界面
这里可以让你浏览设备中全部安装了的 app 和小插件。
用户可随意拖拽 app 或小工具的图标，到达 Home 任意面板空的位置放下。

3. 最近任务界面
这里提供了一个在最近使用的 app 之间切换的快捷方式。它为多个同时进行的任务提供了一个清晰的导航路径。
最近按钮是在导航栏的最右边。展示了用户最近使用的 app，他们通过时间顺序组织，把最近使用的放在最底部。

UI 栏----------

UI 栏是专用于显示通知、设备的通讯状态以及设备导航的区域。通常，UI 栏会跟随所运行的 app 的需要而显示。如果是需要沉浸式体验的 app 如电影或图片，可以暂时隐藏 UI 栏，让用户尽情享受全屏内容不受打扰。



1. 状态栏

左边显示等待通知，右边显示如时间、电池水平、信号强度。向下划动状态栏显示通知详情。

2. 导航栏

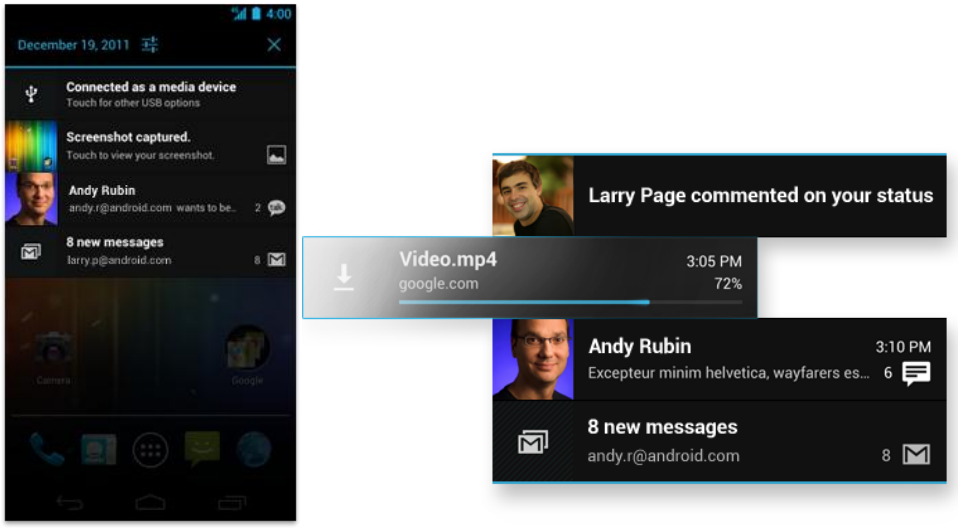
Android 4.0 的新特性。只在没有传统硬件导航栏的设备上显示。放置了设备的导航控件包括返回，Home 和最近任务，在 Android2.3 以及更早版本还有菜单键。

3. 系统栏

在平板电脑上使用，包含了状态栏和导航栏的元素。

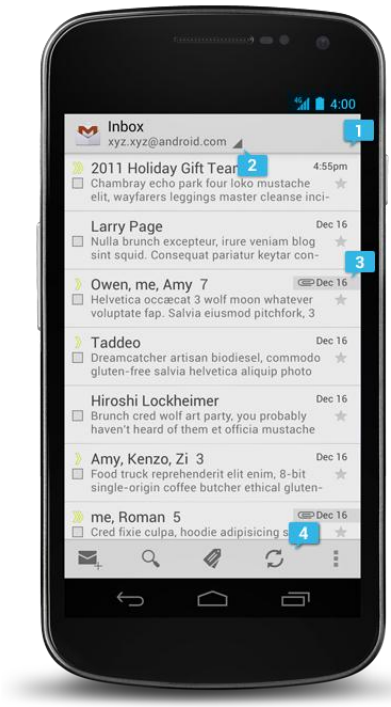
通知----------

通知是用户可以在任何时候从状态栏获取简短信息的地方。它提供了更新、提醒以及那些未重要到需要打断用户的重要信息。向下滑动状态栏可以打开通知抽屉。点击其中一个通知打开相应的 app。更多请看通知。



大部分的通知都是一行标题一行信息。这里推荐两行的布局。如有必要，你可以增加第三行。时间戳是可选元素。

App UI----------



一个典型的 Android app 包含了操作栏和内容区域。

1. 主操作栏

这是 app 的命令和控制中心。主操作栏包含了导航 app 层级和视图的元素，及最重要操作。

2. 视图控制

让用户切换 app 提供的不同视图。视图包含了内容不同的组织方式或不同的功能。

3. 内容区域

内容显示区域。

4. 次操作栏

次操作栏提供了一种方式，就是把操作从主操作栏，分配放置到次操作栏，它可以在主操作栏的下方或屏幕的底部。这个例子中，次操作栏把没放在主操作栏上的一些功能展示在底部。

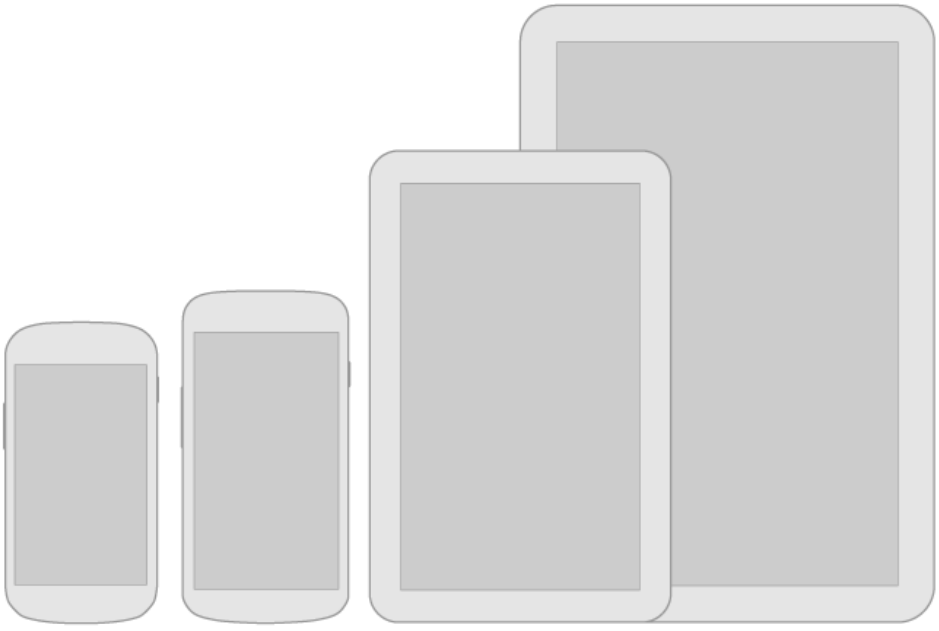
设计风格 STYLE ▶

打造在任何设备上都视觉诱人的 apps。



设备和显示

Android 赋予了成千上万的手机、平板电脑和其它设备动力，它们包含了各种各样的屏幕大小和构成元素。利用 Android 灵活的排布系统，你可以创建从大屏幕大到小屏幕优雅转换的 apps。



灵活

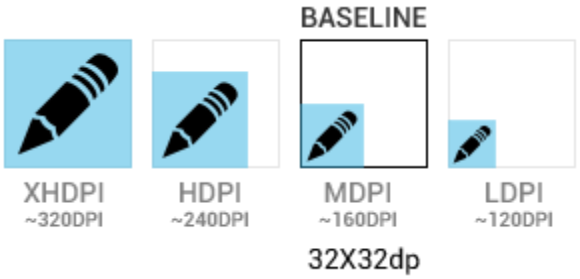
扩展和缩小你的布局来适应不同的高度和宽度。

优化布局

在大设备上时，利用额外的屏幕版面。创建包含多视图的复合视图，以展示更多的内容和更便捷的导航。

为全部设备做好准备

为不同屏幕分辨率提供资源，保证你的 app 在任何设备上都漂漂亮亮。



策略

当你为不同屏幕设计时，从哪开始？

一个方法是：在标准基础（MDPI）上开始，然后放大或缩小，以适应到其它尺寸。另一个方法是从设备的最大尺寸开始，然后缩小，并适应到你需要的小屏幕尺寸。

想了解关于此话题的更多内容，请看支持多屏幕。

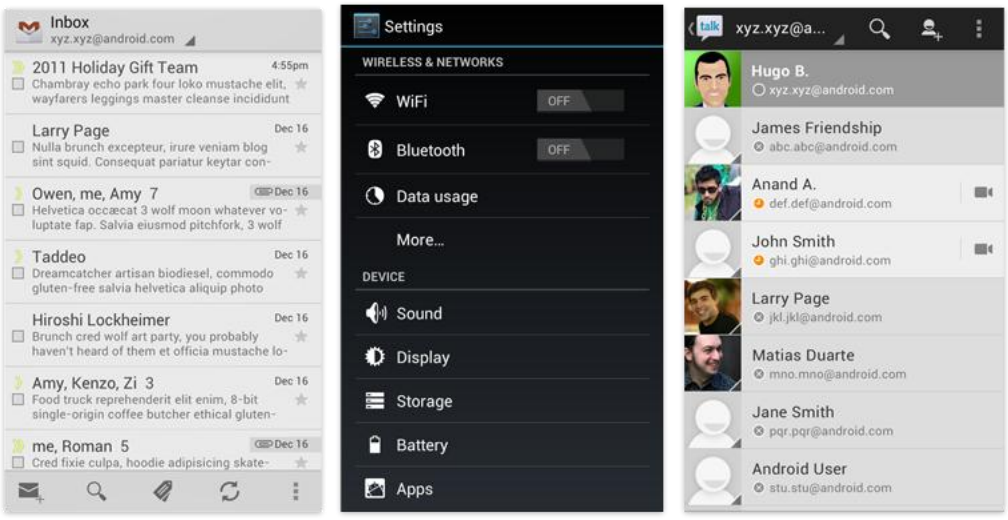
主题样式----------

主题样式是 Android 为了保持 app 或操作行为的视觉风格一致而创造的机制。风格指定了组成用户界面元素的视觉属性，如颜色，高度，空白及字体大小。为了各个 app 在平台上达到更好的统一效果，这次 Android 雪糕三明治系统为你的 app 提供了 3 套系统主题：

* Holo 浅色主题
* Holo 深色主题
* Holo 浅色底+深色操作栏主题

使用这些主题可以快速帮助你的 app 适应 Android 的整体视觉语言。

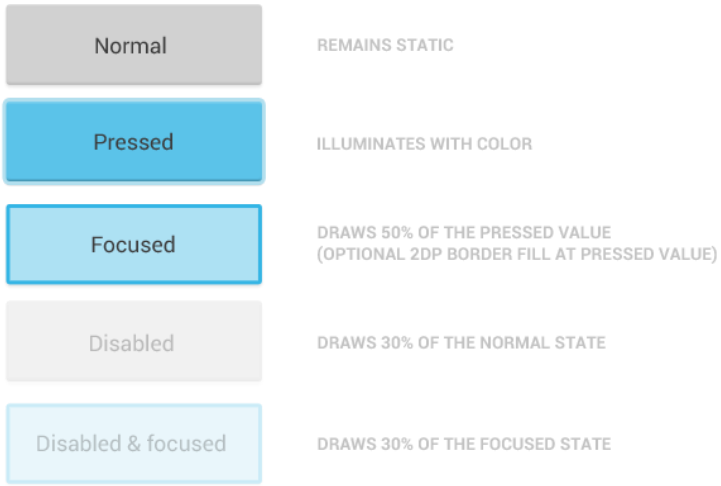
选择最适合你的 app 需求和设计的系统主题吧。如果你想要一个更独一无二的样式，从系统主题开始也是一个好主意。系统主题为你之后选用自己的视觉风格提供了一个坚固的基础。



触摸反馈

使用颜色和光作为触摸的反馈，加强手势行为的结果，暗示哪些操作能用，哪些操作被禁用。无论何时用户触摸了一个可操作区域，你的 app 都要提供视觉反馈。使用户知道哪些可操作。

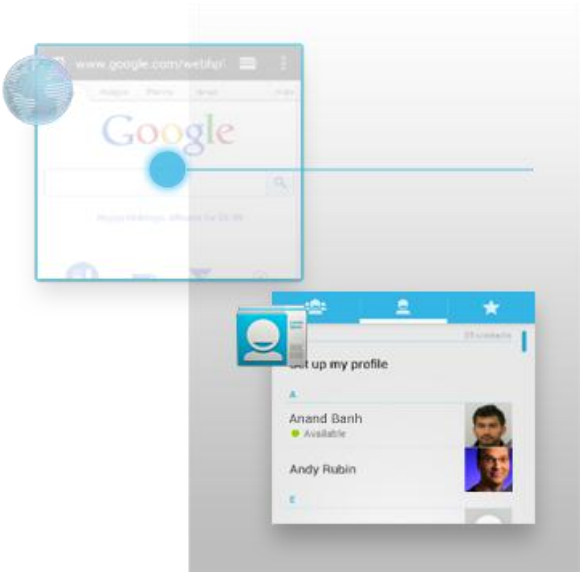
状态



大部分 Android 的 UI 元素都内置有触摸反馈，包括暗示触摸元素是否有效果的状态。

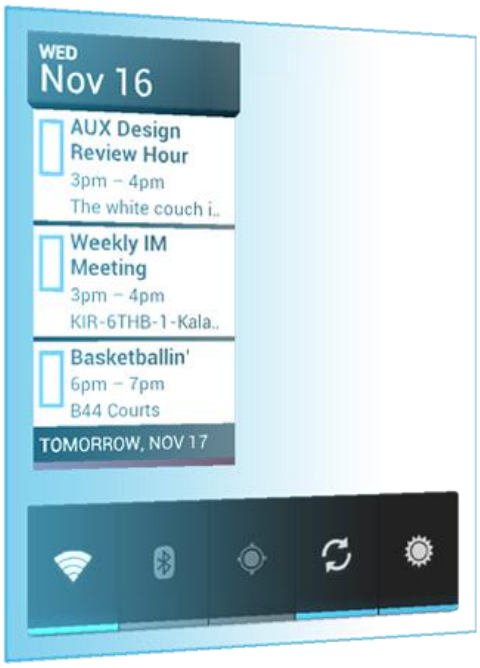
沟通

当有更复杂的手势时，帮助用户理解操作的结果。例如，在最近任务里，当你开始横划缩略图时，它会变暗淡。这帮助用户理解横划会引起对象的移除。



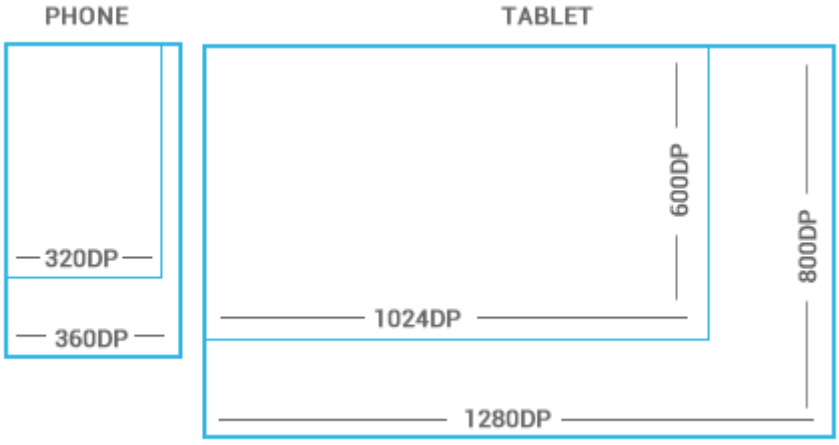
边界

当用户尝试滚动超过内容边界时，要给出一个明确的视觉线索。例如，如果一个用户在第一个 home 屏继续往左滚动，屏幕的内容就会向右倾斜，让用户知道再往这个方向的导航不可用。很多 Android 可滚动的 UI 部件（如列表和网格列表）都已经内置支持边界反馈。如果你要自定义，记得要做边界反馈。



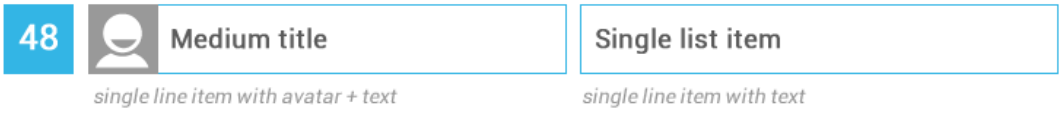
标准和网格

设备不仅有屏幕物理大小的不同，还有屏幕密度的不同（DPI，dots per inch）。屏幕物理大小是手机（小于 600DP）或平板电脑（大于或等于 600dp）的物理尺寸。屏幕密度是 LDPI，MDPI，HDPI，XHDPI。通过为不同屏幕大小设计不同的布局，为不同屏幕密度提供不同的位图图像，来优化您的 app 的用户界面。



48dp 规律

通常把 48dp 作为可触摸的 UI 元件的标准。

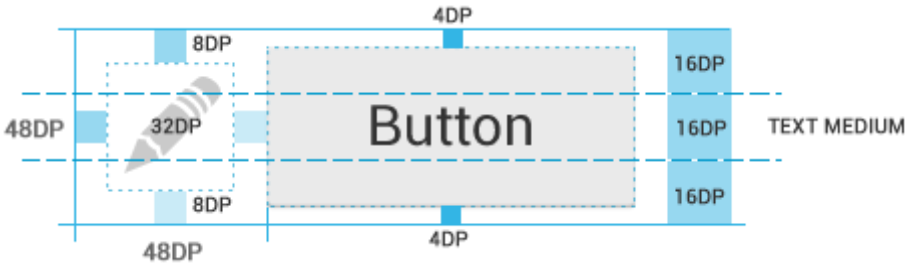


为什么是 48dp?

一般来说，48dp 转化为一个物理尺寸约 9 毫米。建议的目标大小为 7-10 毫米的范围，这是一个用户手指能准确并且舒适触摸的区域。

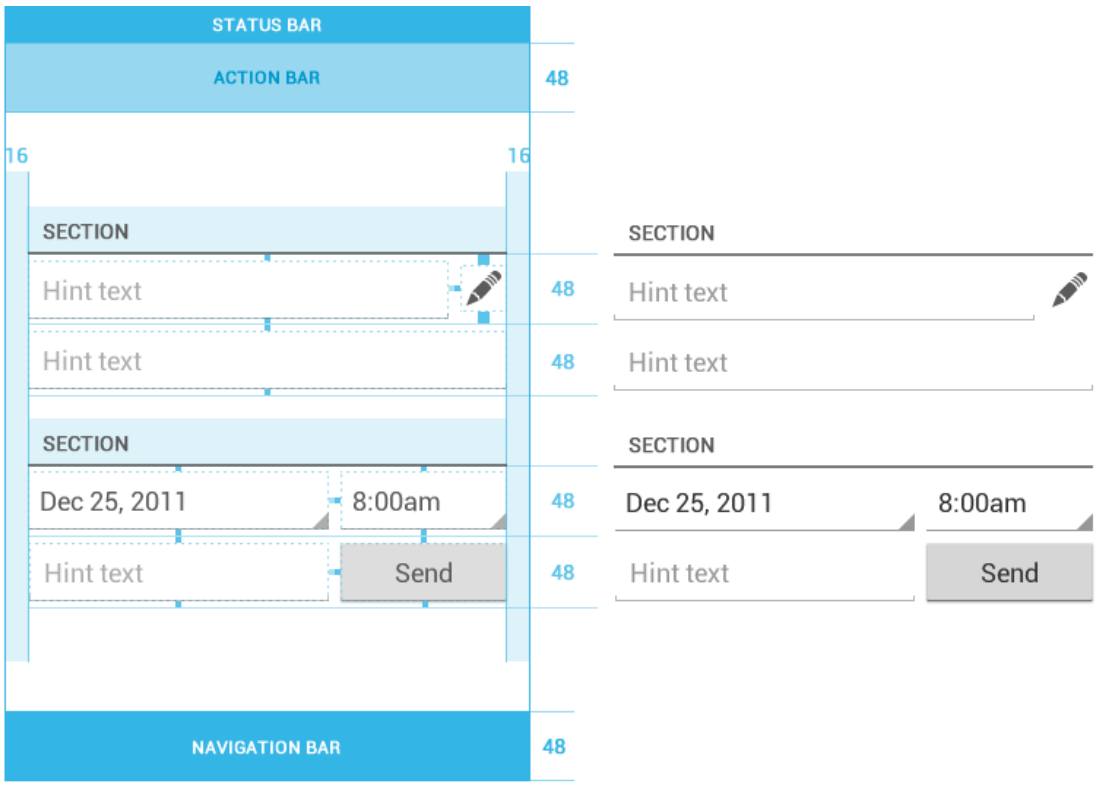
如果你设计的元素高和宽至少 48dp，你就可以保证：

1. 触摸目标绝不会比建议的最低目标（7mm）小，无论在什么屏幕上显示。
2. 在整体信息密度和触摸目标大小之间取得了一个很好的平衡。



注意空白

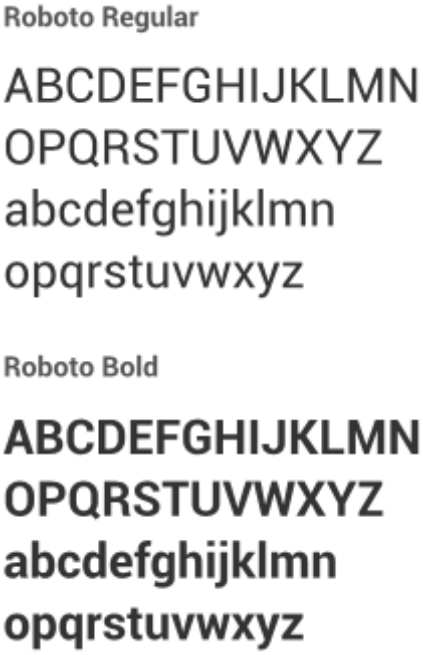
每个 UI 元素之间的间距是 8pd。



字体

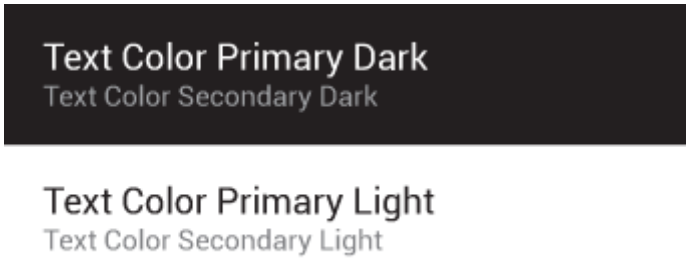


Android 的设计语言依赖于传统的排版工具，如大小，空间，节奏，以及与底层网格对齐。成功应用这些工具可以帮助用户快速了解信息。为了更好支持排版，Android 冰淇淋三明治介绍一种新的字体：Roboto，专门为高分辨率屏幕下的 UI 而设。目前 TextView 的框架默认支持常规，粗体，斜体，粗斜体。



默认文字颜色

Android UI 使用了以下默认颜色样式：textColorPrimary 和 textColorSecondary。
浅色主题使用 textColorPrimaryInverse 和 textColorSecondaryInverse.。框架里的文本颜色样式同样支持使用时不同的触摸反馈状态。



字体大小

使用不同大小字体对比，可以创建有序的，易理解的布局。然而，在相同的用户界面有太多不同大小的字体，会很乱。Android 框架使用以下的大小：



用户可以在设置 app 中选择系统级的文本缩放。

颜色

使用颜色是为了强调内容。选择适合你品牌的颜色，可以为视觉元素提供更好的对比。注意的是红绿颜色对红绿色盲不适用。



调色板

蓝色是 android 调色板里的标准颜色。每个颜色都有对应的一系列饱和度，供需要的时候使用。



图标



图标在屏幕中虽然只占一小块地方，但它为操作、状态和 app 提供了一个快速且直观的表现形式。

启动图标

启动图标是放在 home 或全部 app 界面上代表你的 app 的视觉表现。由于用户可以改变 home 页的壁纸，所以要确保你的启动图标在任意壁纸上都清晰可见。



尺寸和缩放
在移动设备上的启动图标必须是 48x48dp。
在应用市场上启动图标必须是 512x512dp。

大小
整体大小为 48x48dp

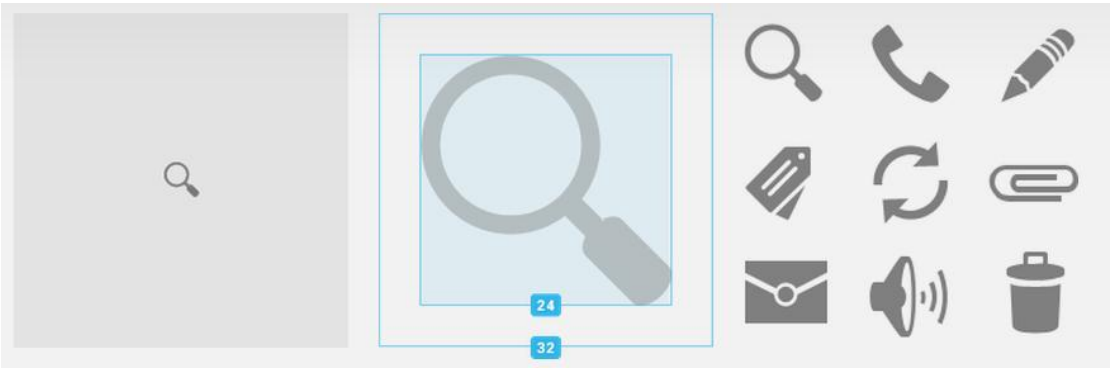
风格
可以使用一个独特的剪影。三维，前视角带有一点点从上往下的透视，让用户可以感觉到有一定深度。





操作栏图标

操作栏图标是平面的按钮，代表用户在你的 app 中可以使用到的最重要的操作。每一个图标都应该用简单的比喻来传达一个单纯的概念，并让大部分人都能一目了然。



尺寸和缩放

手机的操作栏图标是 32x32dp。

焦点区域和大小

整体大小 32x32dp

图形区域 24X24dp

风格

象形，平面，不用过于细节，流畅的曲线或尖锐的形状。如果你的图形太长条（如电话、书写笔），向左或向右旋转 45 ° ，以填补空间的焦点。描边和空白之间的间距应至少 2dp。

颜色

颜色: #333333

可用 60% 透明度

禁用: 30% 透明度



颜色: #FFFFFF

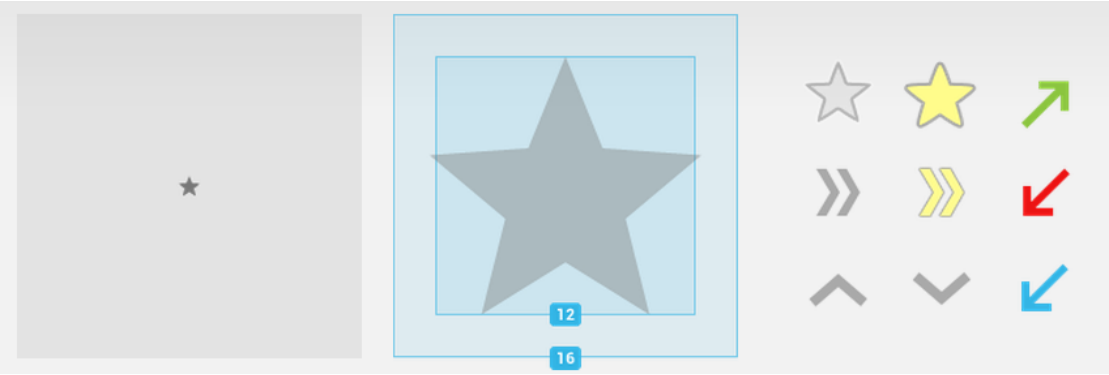
可用: 80% 透明度

禁用: 30% 透明度



小/场景图标

在你的 app 中，使用小图标提供操作或特定项目的状态。例如，Gmail app，消息前的星形图标，标记为重要消息。



尺寸和缩放

小图标为 16x16dp

焦点区域和大小

整体大小 16x16dp

可视区域 12X12dp

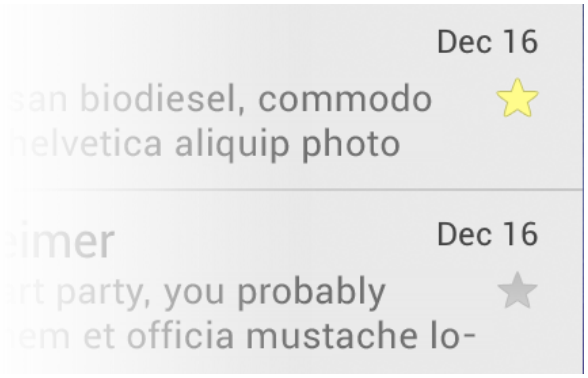
风格

中性，平面，简洁。使用填充形状比细描边更容易看到。使用单一的视觉隐喻，使用户可以很容易地识别和理解它的目的。

颜色

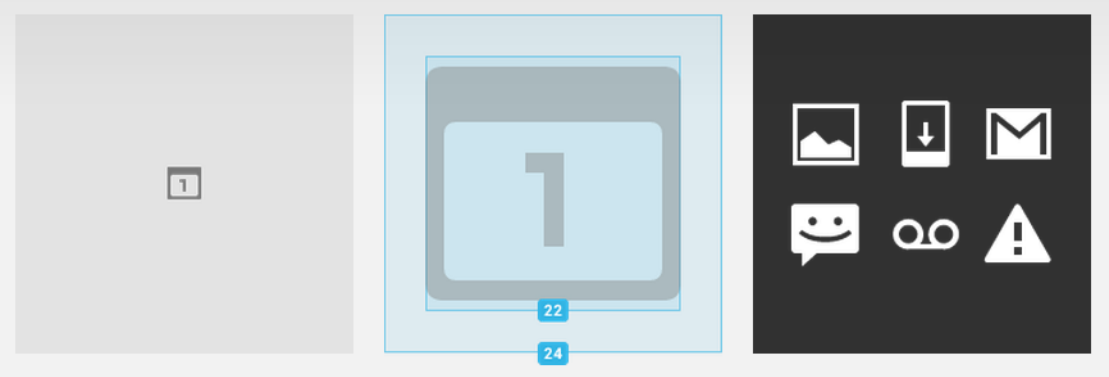
带目的性的使用颜色。例如，Gmail 使用黄色的星形图标表示标记消息。如果图标是可操作的，选择一个与背景对比大的颜色。





通知图标

如果你的 app 有通知，要提供一个每当有新通知时显示在状态栏的通知图标。



尺寸和缩放
手机的操作栏图标是 24x24dp。

焦点区域和大小
整体大小 24x24dp
可视区域 22X22dp

风格
保持风格的平面化和简洁，使用与启动图标相同的视觉隐喻。

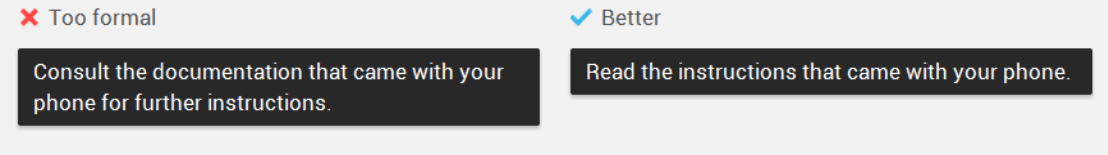
颜色
通知图标必须是完全白色。而且，系统可能会缩小或变暗图标。



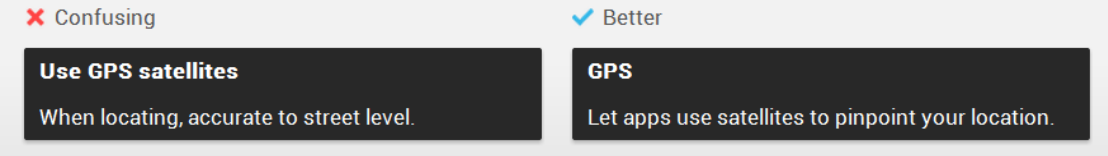
写作风格----------

当为你的 app 写句子时：

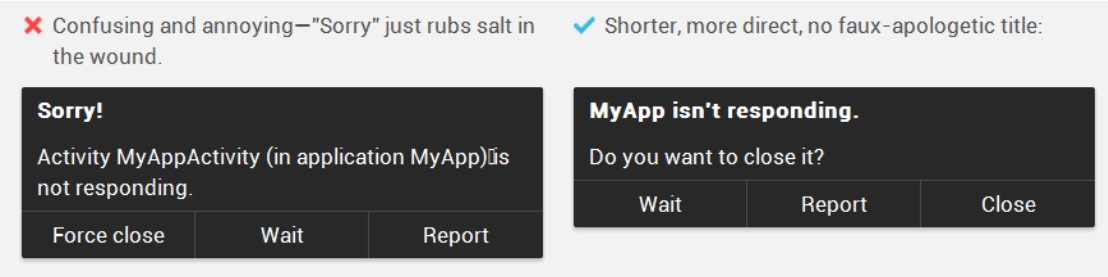
1. 保持简短。简明，简单，准确。从限制使用 30 个字符（包括空格）开始，除非必要绝对不增加字符。



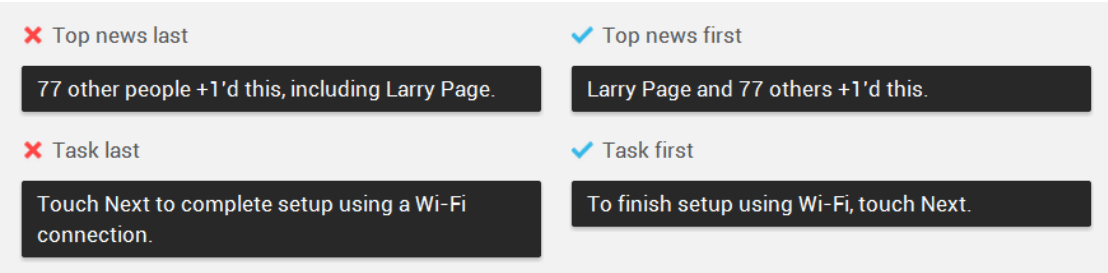
2. 保持简单。假装你正在与一个精明能干的人说话，但他不懂技术术语、英语不是很好。使用简短的话，主动动词，和普通名词。



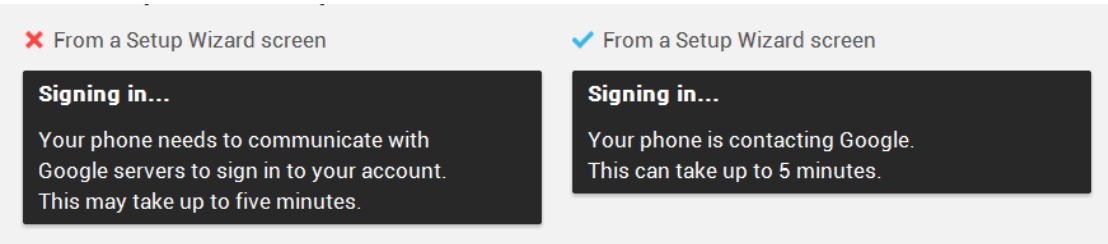
3. 保持友好。使用缩写。直接使用第二人称（“你”）。如果这不是你休闲对话时会用的句子，那大概它也不适用于此。避免唐突和骚扰，使用户感到安全，愉快，充满活力。



4. 先讲最重要的事情。前两个单词（约 11 个字符，包括空格）至少应包括一个最重要的信息。如果不是这样，重新开始。



5. 仅描述必要的。不要试图解释细微的差别。



6. 避免重复。如果一个重要的词在一个界面或一段文本内不断重复，想办法只用它一次。

模式 PATTERNS ▸

新的 Android 4.0

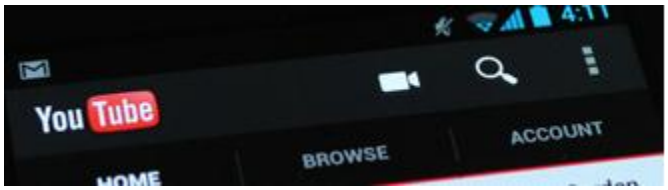
导航栏

Android 4.0 去除了传统手机物理按键的需求，以虚拟按键代替。虚拟按键包括返回，Home 和最近任务。通过阅读本章中的“兼容性”可以了解操作系统如何适应手机的物理按键，以及如何支持 android 3.0 以前的旧版本——那些导航按键包括 menu 的 apps。



操作栏

在 android 的 app 中，操作栏是最重要的结构元素。它提供了跨整个平台的一致性导航，并展示你的 app 的重要操作。



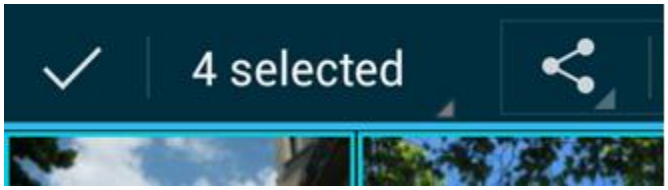
多面板布局

设计 app 时，适配不同设备和屏幕大小对 android 来说非常重要。多面板布局，允许你将在小屏幕上单独展示的界面，以富态视图的方式一同展示在平板电脑的同一个界面上。



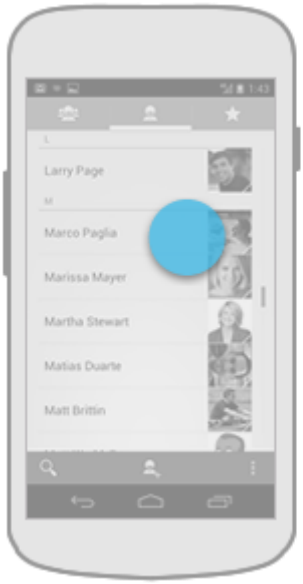
选择

长按的手势，在传统操作中是用来展示内容项的情境菜单，现在用于内容项的选择。当选择内容项的时候，情境操作栏可以展示你 app 的重点操作功能。



手势

手势允许用户通过操纵屏幕上的对象，与我们的 app 进行互动。以下的图例中，展示了 android 里支持的核心手势。



触摸

触发指定项的默认功能

动作：按下 抬起



长按

进入内容项选择模式。允许你在一个视图中选择一个或更多的项，用情境操作栏对所选资料进行操作。避免使用长按来展示情境菜单。

动作：按下 等待 抬起



横划

滚动满屏的内容，或者在同一层级中转换导航间的视图。

动作：按下 移动 抬起



拖动

在视图中重新排布资料的位置，或者移动资料到容器中（如，Home 界面中的文件夹）

动作：长按 移动 抬起



双触摸

缩放到内容，也用于文本选择的一个辅助手势。

动作：两次快速且连续触摸



Pinch open

放大内容

动作：两个手指按下 向外移动 抬起



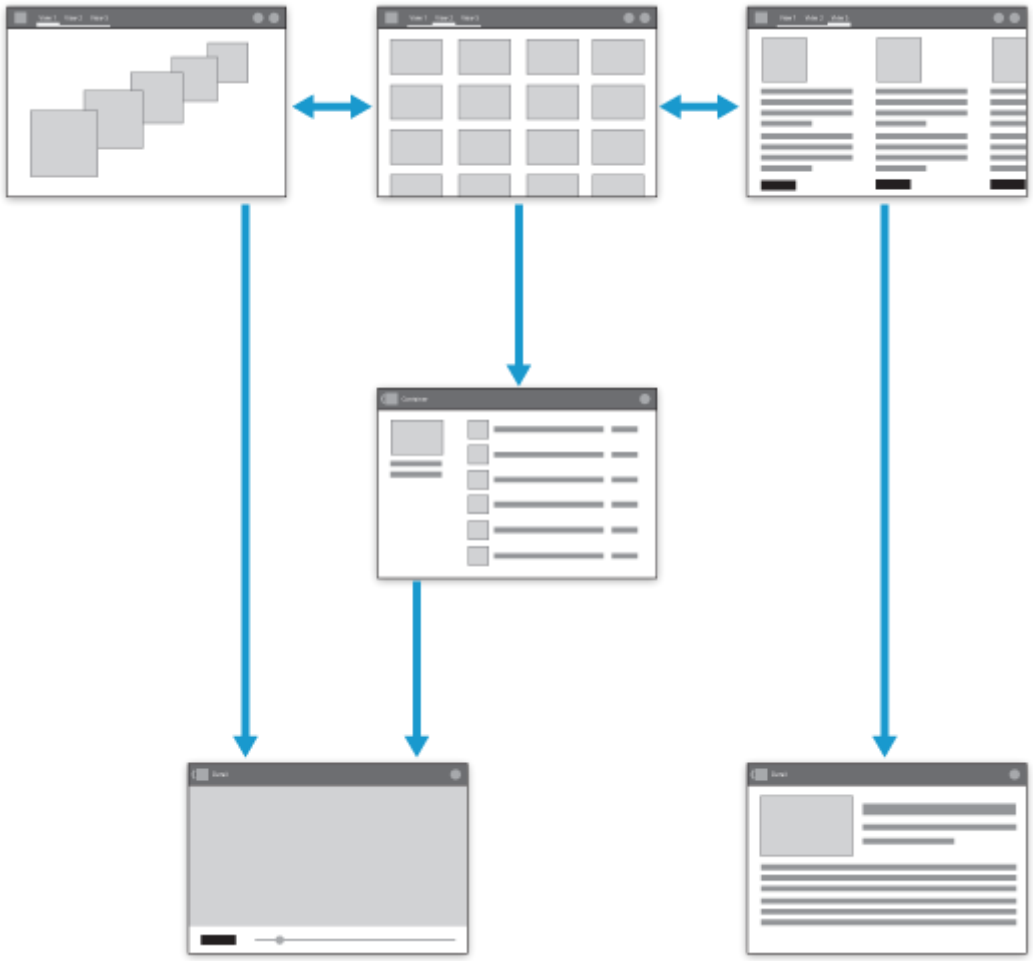
Pinch close
缩小内容。
动作：两个手指按下 向内移动 抬起

应用结构----------

app 结构
app 多种多样，以满足各种不同的需求。比如：
* 计算器或照相机，这种在一个屏幕里围绕一个核心功能的 app
* 电话，其主要目的就是在不同的操作中切换，而不是有更深层次导航的 app
* Gmail 或应用市场，这种结合了一系列深层级内容视图的 app

你的 app 的结构，主要取决于你想展示给用户的内容和任务。

常用结构
一个典型的 androidapp 包括顶级视图和详情/编辑视图。如果导航的层级结构是深而复杂的，那么目录视图可以用于连接顶级视图和详情视图。



1. 顶级视图

App 的顶级通常包括其支持的不同视图。视图既可以是展示相同内容的不同呈现方式，又可以是展示你的 app 的不同功能模块。

2. 目录视图

目录视图允许你进入更深层级的内容。

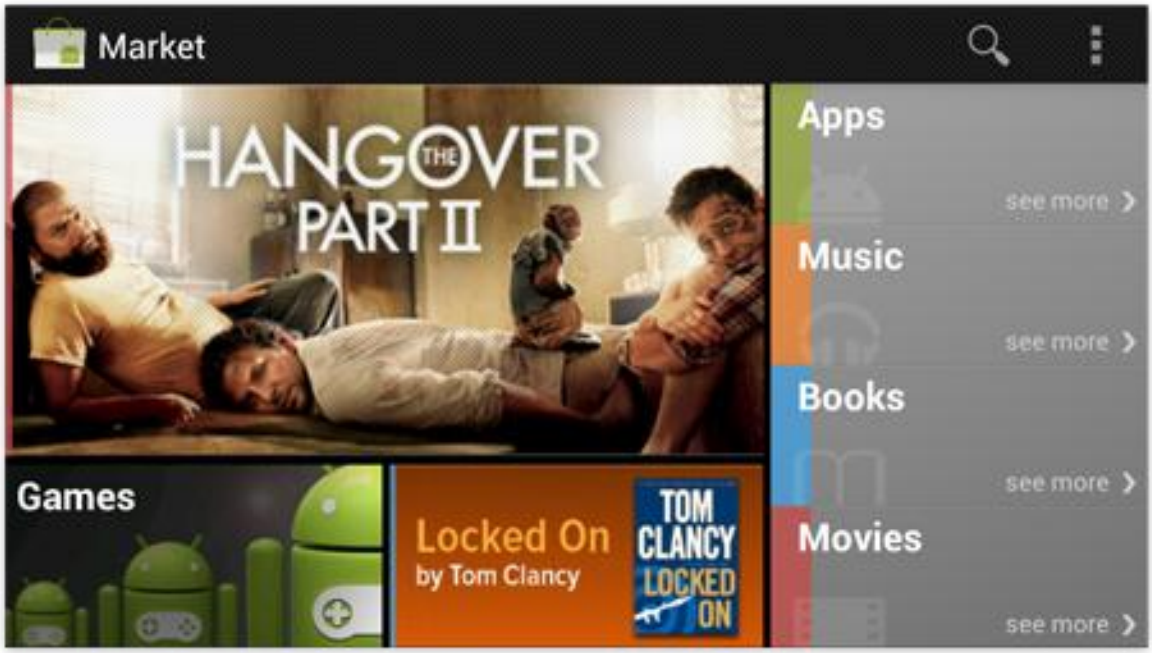
3. 详情/编辑视图

详情或编辑视图是你享受或创造内容的地方。

顶级

启动界面的布局需要特别注意。这是用户启动 app 后看到的第一个界面，所以，首屏应该同时满足老用户和新用户。

问问自己，“我的典型用户在 app 中最想做些什么？”，并根据这个来设计首屏的体验结构。



市场启动界面首先展示了进入 app、音乐、图书和电影的导航入口。同时定制了用户可能感兴趣的丰富的推荐内容。并在操作栏上给出了快速搜索的入口。

1. 把推荐内容置前

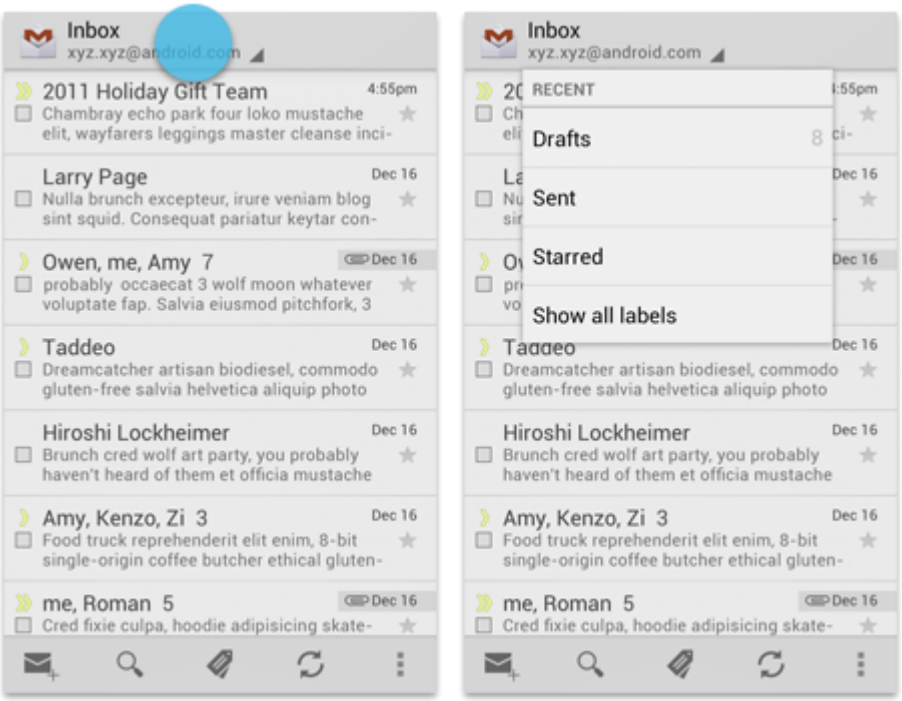
许多 app 把焦点集中在推荐内容的展示上。避免只在界面上出现导航，要让用户在你的首屏中快速获得他们想要的内容。选择具有视觉吸引力，且适当的内容类型和屏幕大小的布局。

2. 为导航和操作进行操作栏设置

在 app 所有的界面中，都应该显示操作栏，提供一致的导航和展示重要的操作。

层级操作栏应该考虑：

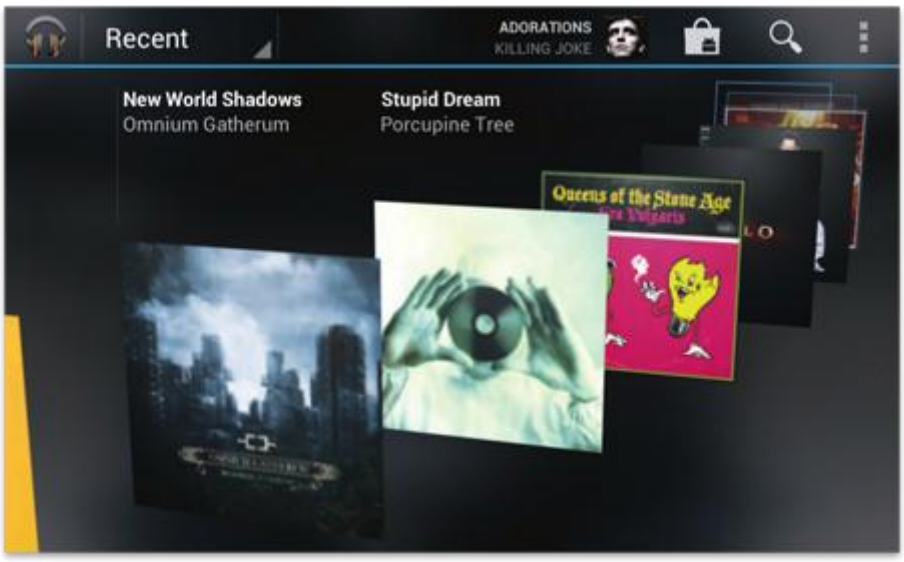
* 使用操作栏展示你的 app 图标或标题。
* 如果你的第一层级包括多种视图，或在多个账户中切换是一个重要的使用情景，确保用户在增加视图控制的操作栏上操作容易。
* 如果你的 app 允许用户创建内容，考虑在第一层级就给出创建内容的入口。
* 如果你的内容可以被搜索，就要在操作栏中包含搜索操作。人们可以通过搜索打破导航层级结构，进行全局性搜索。



电子邮件是效率型 App，所以需要高效和轻松浏览高密度信息列表。导航可支持不同帐号和最近使用标签之间的切换。在底部的操作栏中，会把常用的创建新邮件或搜索的图标优先放在前面。

3. 让人们一眼就能认出你的 app

为你的 app 创造识别性的重要性超越了操作栏。你的 app 通过它的内容版式和交互方式形成自己的一个识别点。尤其是富媒体应用，尝试创造出独特的布局，展示你的内容和超越单调的列表视图方式。



3D 旋转木马式的封面展示方式为音乐这个 app 建立起一个独特的识别。默认在最近试听中展示用户最后听过的一些歌曲。

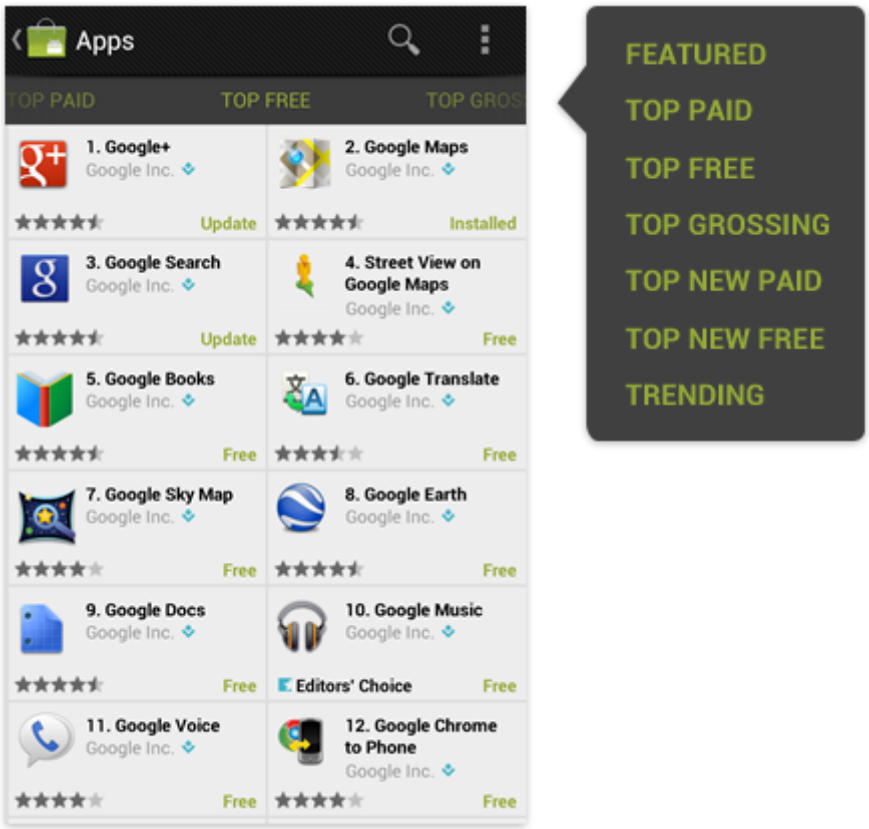
分类

一般来说，一个层级深的、数据驱动的 app 的目的是通过有组织的分类，引导用户可以进入查看和管理。扁平化你的 app 层级可以减轻用户对导航层级的负担。

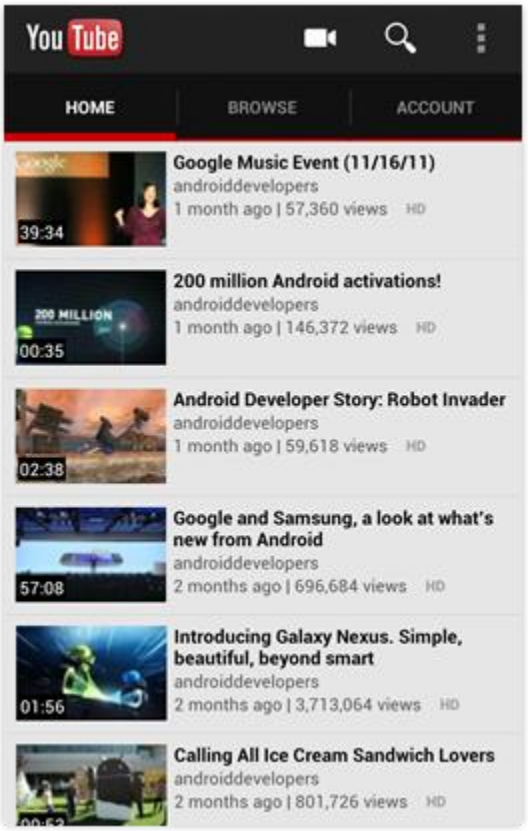
尽管 app 的层级结构取决于内容，但有几个方式，可以减少繁冗的导航认知。

1. 使用选项卡（tabs）结合分类选择和内容的展示
此方法适用于分类熟悉或者分类数量不多的情况。这种方法的优势是减少了一个层级，而且内容仍然在用户关注的中心区域。在丰富的内容分类之间的横向导航（如下图）更像是一种随意的浏览，而不是有明确步骤的导航。

如果分类是熟悉的、可预见的、或者是密切相关的，使用滚动选项卡 scrolling tabs（并非所有的项同时展示）。控制滚动选项卡数量，减少导航的滚动。经验法则：不超过 5-7 个选项卡。



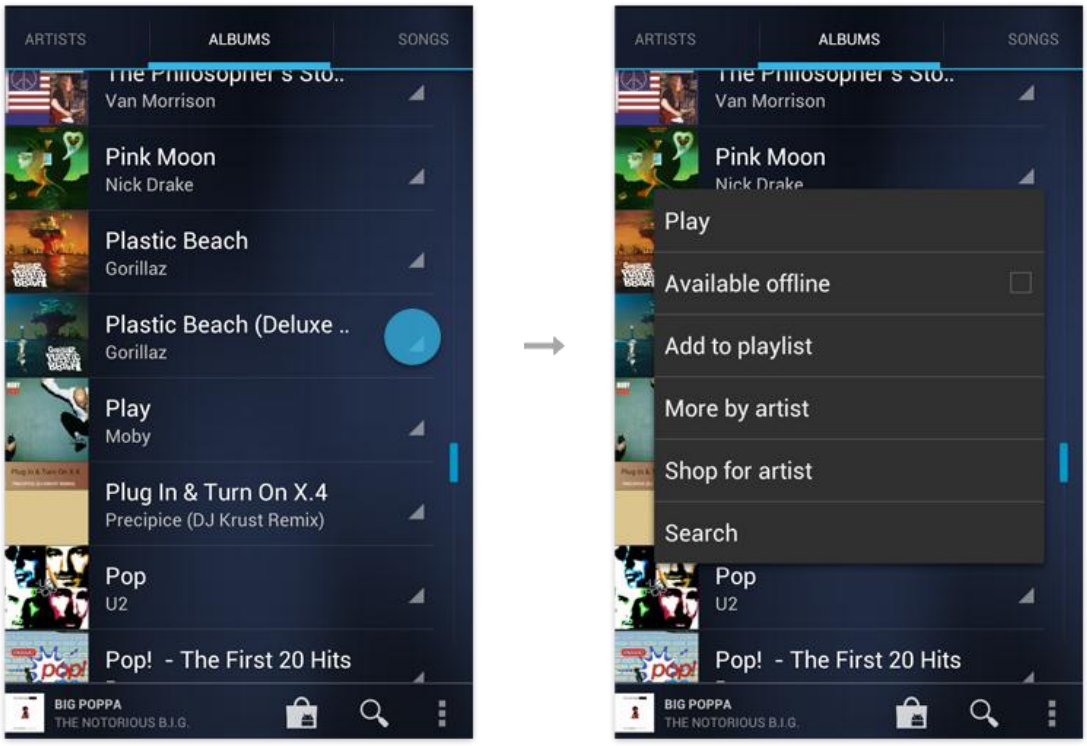
应用市场用选项卡同时显示类别的选择和内容。用户可以通过在内容上左右横划来选择分类。
如果在选项卡中的类别不密切相关，偏向于用固定选项卡，以便同时查看所有类别。



YouTube 使用固定选项卡来切换不同的、相对无关的功能区域。

2. 允许穿越层级的操作

利用快捷键让用户更快达到目的。允许用户在第一层级调用对二级内容直接进行操作的列表项，如下图。这让人们直接对内容进行操作，而不必一路遵循导航层级结构。



音乐允许用户在分类视图（专辑）里对歌曲项直接操作，从而去掉了进入歌曲详情页进行操作的导航需求。

3. 作用于多个内容项

尽管目录视图大多为引导人们进入内容详情页服务，但请记住，它也常常可以用于对内容集进行批量操作。

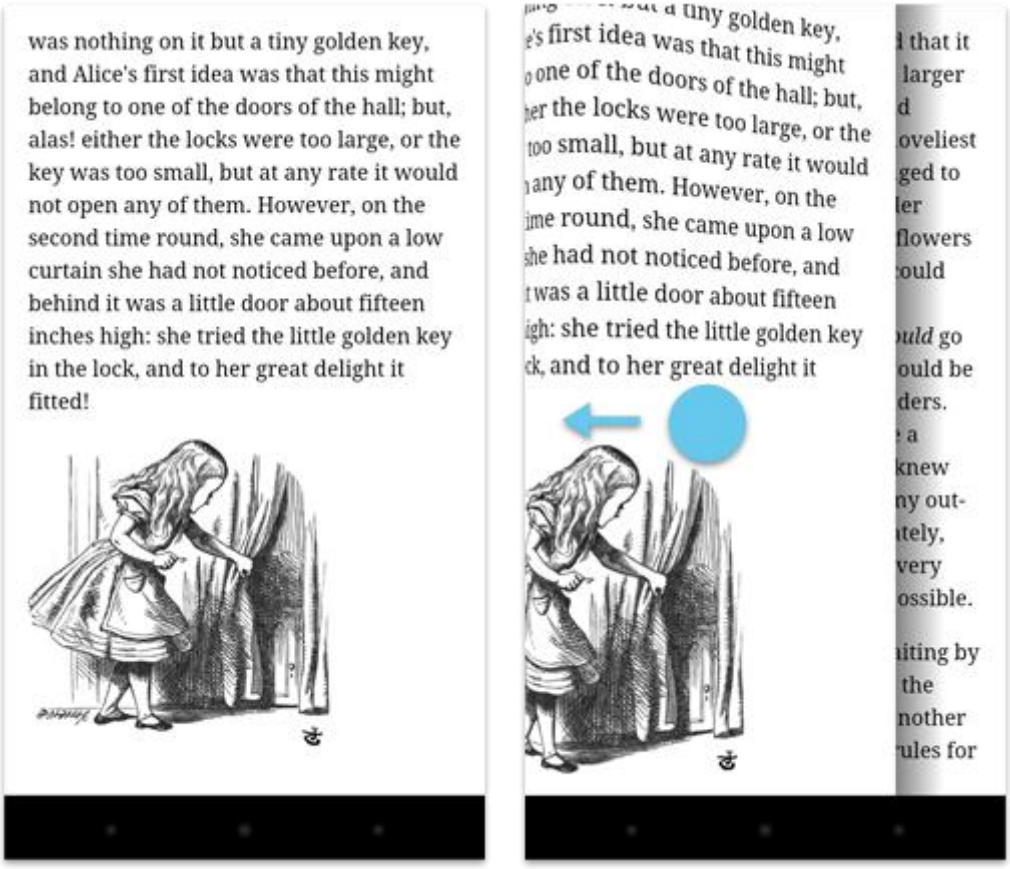
例如，如果允许人们在详情视图中删除项目，那就应该允许他们在目录视图中删除多个项目。详情视图里的操作同样适用于内容集。使用多项选择让 app 可以多内容项的目录视图中，同时对多个项进行功能操作。

详情页

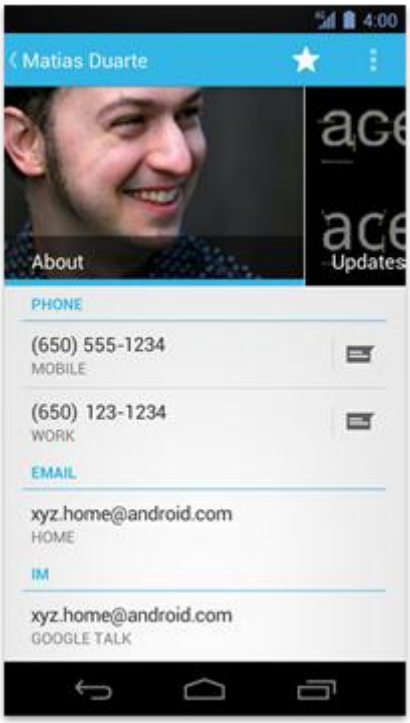
详情视图允许你查看和操作你的内容。详情视图的布局展示取决于内容的类型，在不同的 app 之间相差甚远。

1. 布局

考虑人们将在详情视图中执行的操作，并安排相应的布局。针对沉浸式内容，利用关灯模式允许用户全屏专心查看内容。



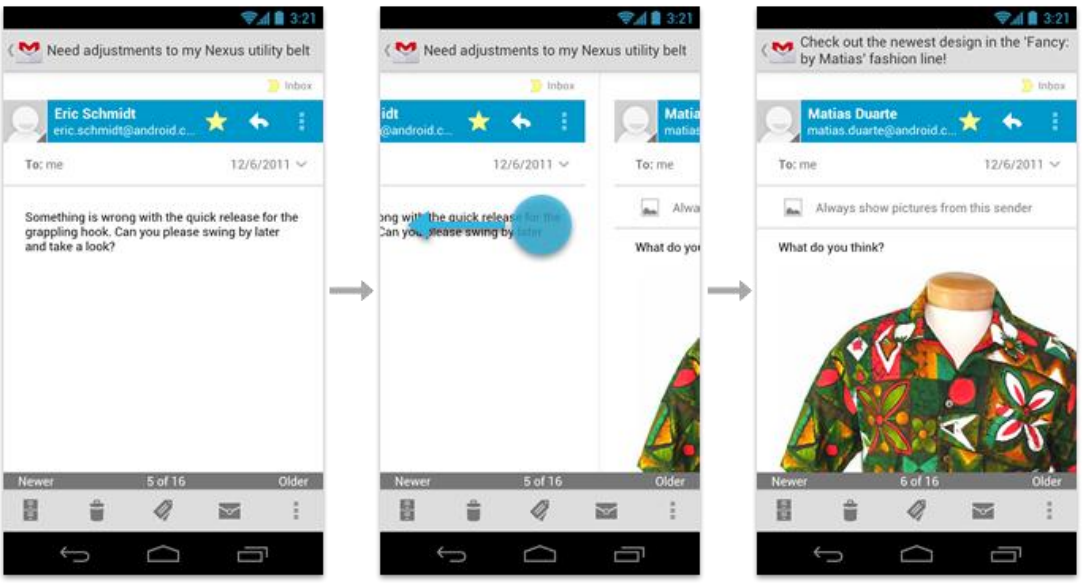
Google 图书的详情视图复制人们读书的真实经验。翻页的动画强调了这种读书真实经验的概念。利用 app 进入关灯模式，即隐藏所有系统界面的方式，营造一种沉浸式的体验。



Peopleapp 详情页的目的，就是为了展示通讯选项。列表视图利于高效扫视和快速访问电话号码、邮件地址和其他信息项。号码呼叫和信息发送被合并到同一行中。

2. 让详情页之间的导航高效

如果你的用户想要查看多个项目，请允许他们在详情视图间切换。使用横划查看或其他技术，如幻灯片模式，来达到这个目的。



Gmail 使用横划的查看方式，来实现详情页间的切换。



除了支持横划手势让图片由左至右切换外，Gallery 还提供幻灯片的控制模式让用户快速跳转到指定图片。

清单

* 在你的首屏中，找到一种方式展示有用的内容。
* 使用操作栏提供一致的导航。
* 使用横排导航和快捷方式，扁平化你的层级结构。
* 使用多项选择来允许用户对内容集进行操作。
* 允许通过横划查看的方式，在详情页中快速切换。

导航----------

“向上（去到上一级）”和“返回“导航

一致的导航是整体用户体验的重要部分。一些与用户行为不一致和出乎用户意料的操作，会让用户感到沮丧。Android 3.0 介绍了一些全局导航操作的重要变化。遵循以下“返回”和“向上（去到上一级）”的设计原则，让你的 app 的导航符合用户的预期和操作。

Android 2.3 和更早期的系统依赖于返回按钮去支持 app 中的导航。在 android 3.0 操作栏的介绍中，第二个导航机制出现：“向上”的按钮，包括一个 app 的图标和向左的符号。



对比“向上”和“返回”操作

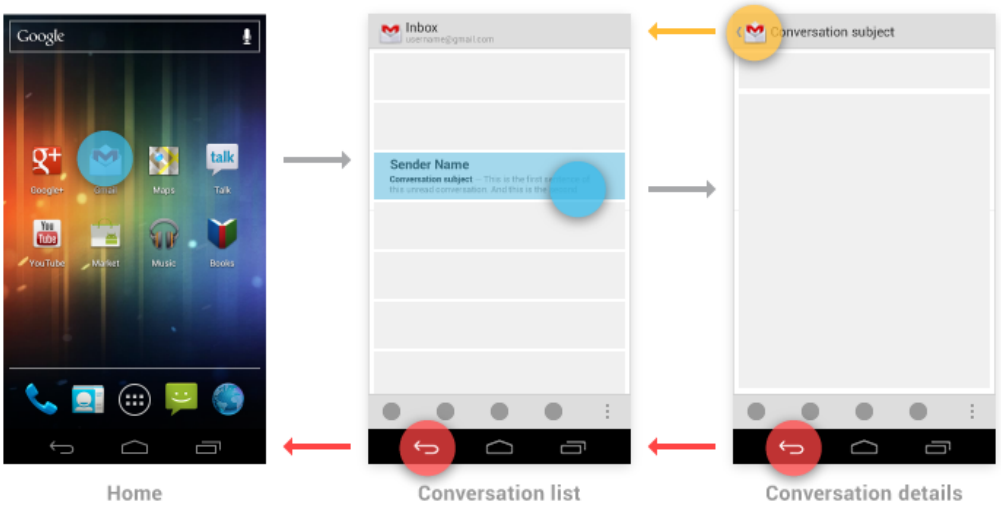
app 中的向上按钮，用在基于层级结构的界面关系间的导航。例如，如果界面 A 展示的是一个列表项，选择其中的一个项，会进入到界面 B（意味着界面 B 会展示更多的细节），那么在界

面 B 应该提供一个“向上”按钮去到界面 A。

如果此界面是一个 app 中最高的一个层级（如，首页），那就不应该出现“向上”按钮。

系统“返回”按键，是基于用户最近查看的界面历史的导航，采用时间倒叙的方式连接界面间的关系。

当上一个视图界面与当前界面有上下层级结构关系时，点击“返回”按键和点击“向上”按钮会有着同样的结果，这是一个常见的现象。然而，不同的是，“向上”按钮可以确保用户停留在同一个 app 中，而“返回”按钮可以让用户回到首屏，或者回到另一个 app，具有一个跳跃性。



返回按钮也能支持一些无直接联系的界面间的导航：
* 消除浮出窗（对话窗口、弹出层）
* 消除情境操作栏，以及把选中态从选中项中移除
* 隐藏屏幕上的键盘

app 中的导航

1. 通过多个入口导航进入到界面

有时一个界面在 app 的层级结构中，不会有一个严格的位置，用户能够从不同的入口进入到这个界面，比如，可以从 app 的任何一个界面中进入到设置界面。在这种情况下，“向上”按钮应该选择去到触发进入设置入口的相关界面，与“返回”的动作一致。

2. 改变界面视图

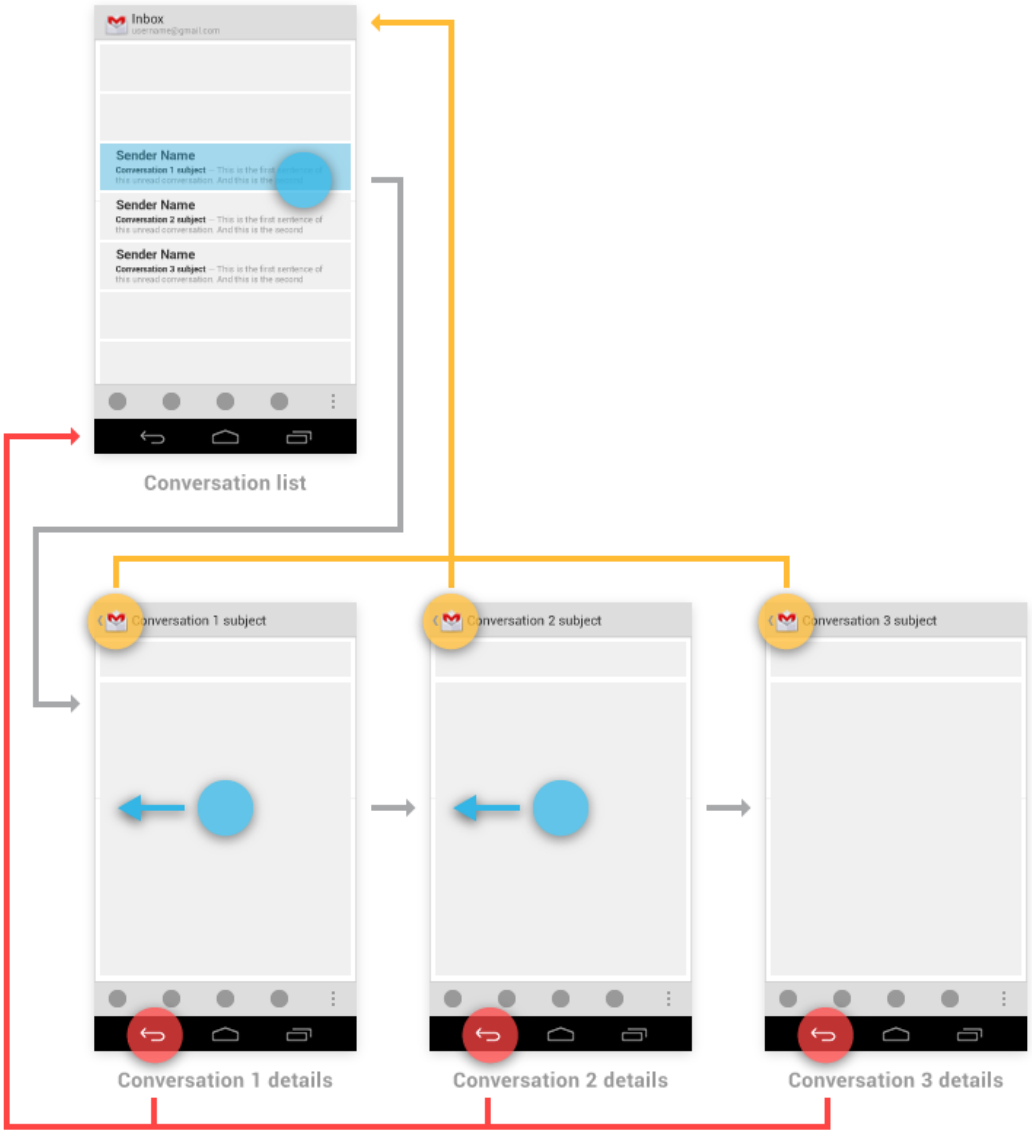
改变界面视图的选择不会影响“向上”和“返回”的操作：界面仍然在 app 层级结构的同一个地方，并且没有创造新的导航历史。

改变视图的案例如下：
* 通过使用选项卡（tabs）或左右横划切换视图
* 通过下拉列表（又称折叠选项卡）切换视图
* 列表筛选

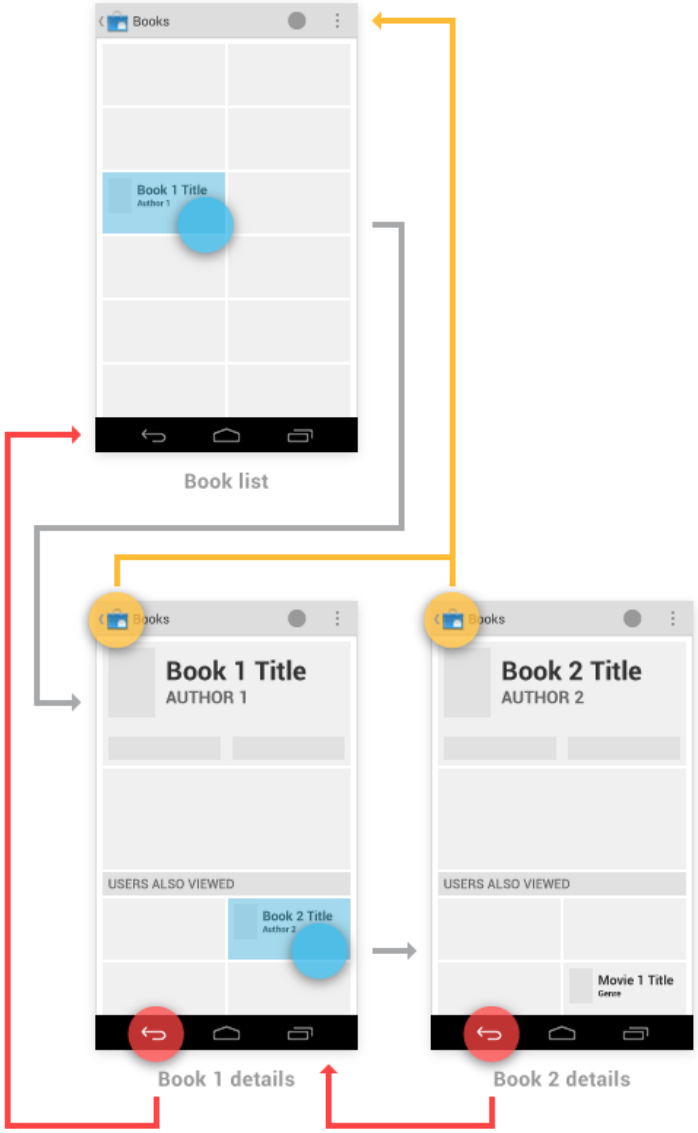
* 列表分类
* 更改显示特性（如，放大）

3. 在同一层级的界面间导航

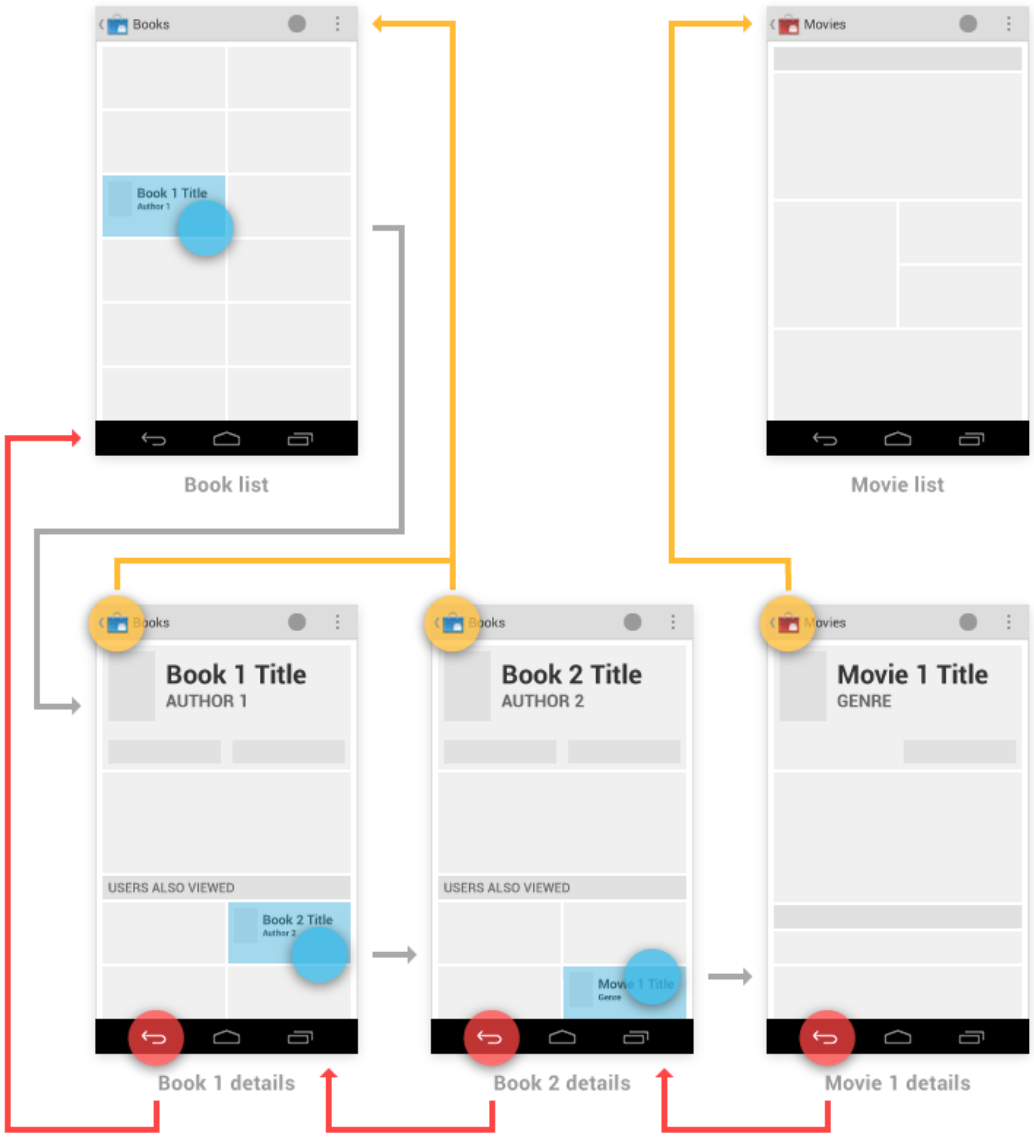
当你的 app 支持从一个列表项（目录视图）导航到其中一个项的详情视图，它通常需要支持到从列表的其中一个项中去到列表顺序中的另一个项。例如，在 Gmail 中，通过左右横划手势可以很容易地在同一个邮箱中，查看前后邮件的内容。如同改变屏幕的视图一样，这种导航不会改变向上或返回的动作。



然而，当用户浏览的详情视图没有绑定在一个统一的相关列表中时，会出现一个值得关注的问题，例如，浏览应用市场里同一个开发商的 apps，或者同一个歌手的专辑。这种情况下，点击链接会创造历史，导致“返回”按钮会逐个贯穿用户所查看过的相关内容界面。使用“向上”按钮应该绕过这些相关的界面，直接去到最近查看内容的上一级界面。



你完全可以基于你对详情视图的了解，使去到上一级的操作行为更聪明便捷。如果我们从应用市场的案例扩展，假设用户从最后的图书视图去到电影作品的详情视图。这种情况下，向上按钮能够去到电影界面的上一级，一个用户从未到过的界面。



app 外的导航

有两种类别的导航能通过 app 的层级结构，从你的 app 之外到达你的 app 之内的界面：

* app 间的导航，例如通过有目的的操作来实现。
* 从系统到 app 的导航，例如通过通知和系统主界面的插件。

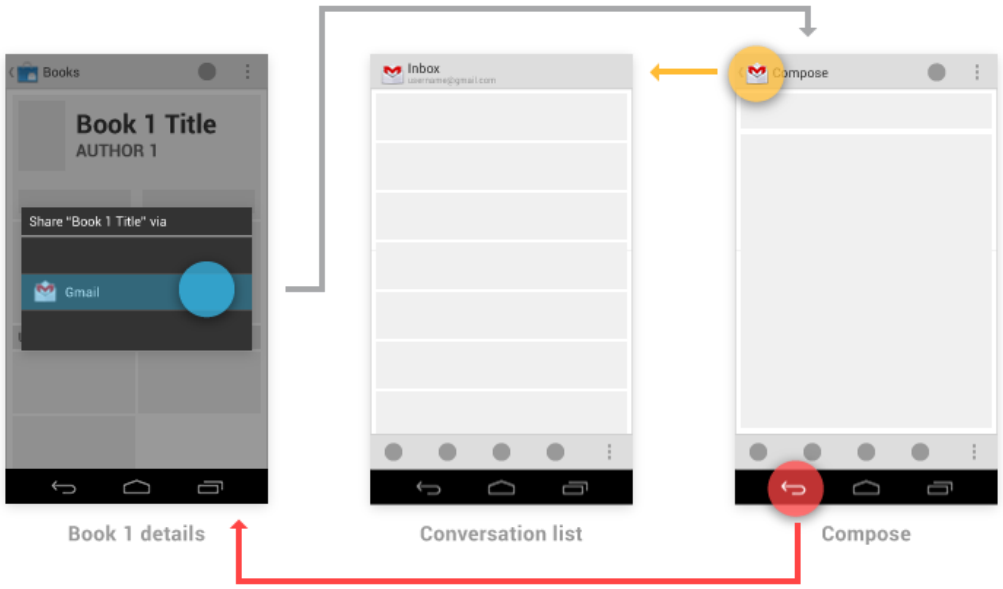
Gmail 提供了一些对应的案例。app 间导航的案例是，点击某个 app 的“分享”后直接跳转到 Gmail 的编写界面。从系统到 app 导航的案例是，新消息通知和主界面的插件都能够进入到收件箱界面，把用户直接带到对话视图中。

1. app 间的导航

当直接通过一些有目的的操作，从另一个 app 深入到你的 app 的层级结构中时，“返回”键将会返回到触发进入当前视图的界面。

例如，在应用市场中选择一本书并分享，用户会被直接引导到 Gmail 的编写界面。在那里，向

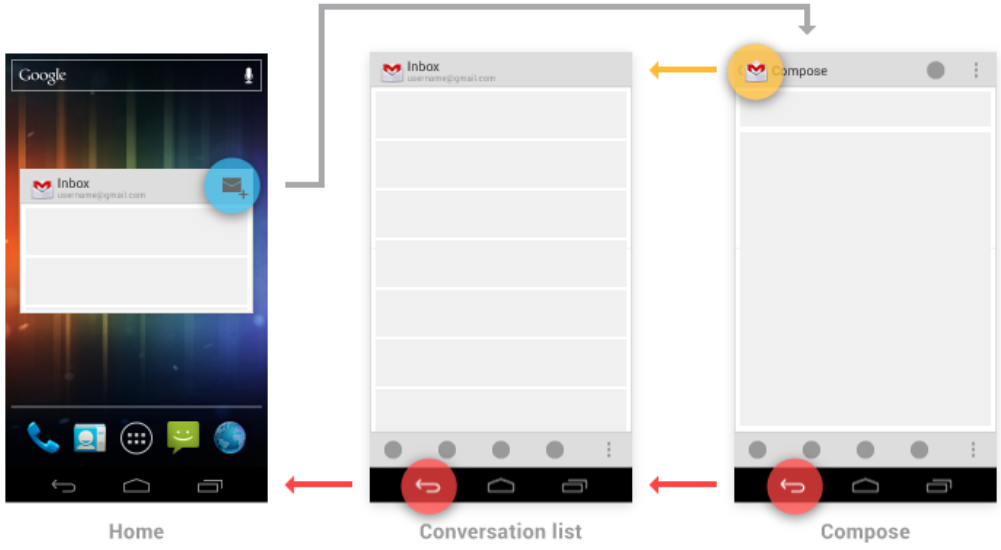
上操作是返回到收件箱（正好同时是编写界面的上一级和 app 的首页），返回操作则是去到应用市场。



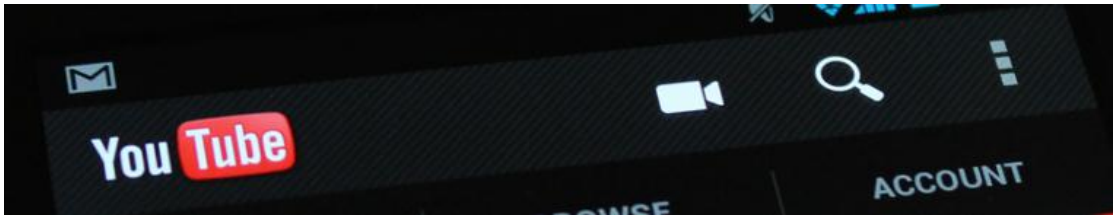
2. 从系统到 app 的导航
如果是通过系统的通知机制或主界面的插件到达你的 app，向上行为在这里的描述与上述的 app 间导航的一致，都是去到当前所在界面的上一个层级。

对于“返回”按键，需要给到用户一个心理预期，点击“返回”按钮的时候，可以遵循任务返回路径去到 app 的第一层级界面。这样，在退出之前，忘记如何进入你的 app 的用户可以安全的去到 app 的第一层级界面。

例如，Gmail 在 Home 界面的小插件有一个按钮可以直接去到他的编辑界面。操作并去到编辑页面后，返回按键首先会返回到收件箱，然后紧接着返回到 Home 界面。



操作栏----------



操作栏可以说是 android app 中最重要的结构元素。它在每个屏幕的的顶部都有一块专用的区域，通常会贯穿整个 app。

操作栏的主要目的是：
* 突出重要的操作（如搜索）并用可预见的方式访问。
* 支持一致的导航和 app 内视图的切换。
* 通过把少用的操作收起到更多操作里，以减少混乱。
* 为你的 app 提供一个专用的空间。

如果你刚开始编写 android 的 app，请注意，操作栏是最重要的设计元素之一。遵循以下的准则，将会大大帮助你的 app，让界面符合 android app 的核心。

常规的组织结构
操作栏被分为适应于大多数 app 的四个不同的功能区域。



1. app 图标
用 app 的图标建立起你的 app 的识别。如果你愿意，它可以被各种 logo 或品牌图标代替。重要：如果 app 当前不是在第一层级的界面，需要保证向上一级的符号展示在 app 图标的左边，让用户能去到上一个层级。更多关于向上一级的导航的讨论，见“导航”模块。



app 图标中没有和有向上一级操作的展示如上图。

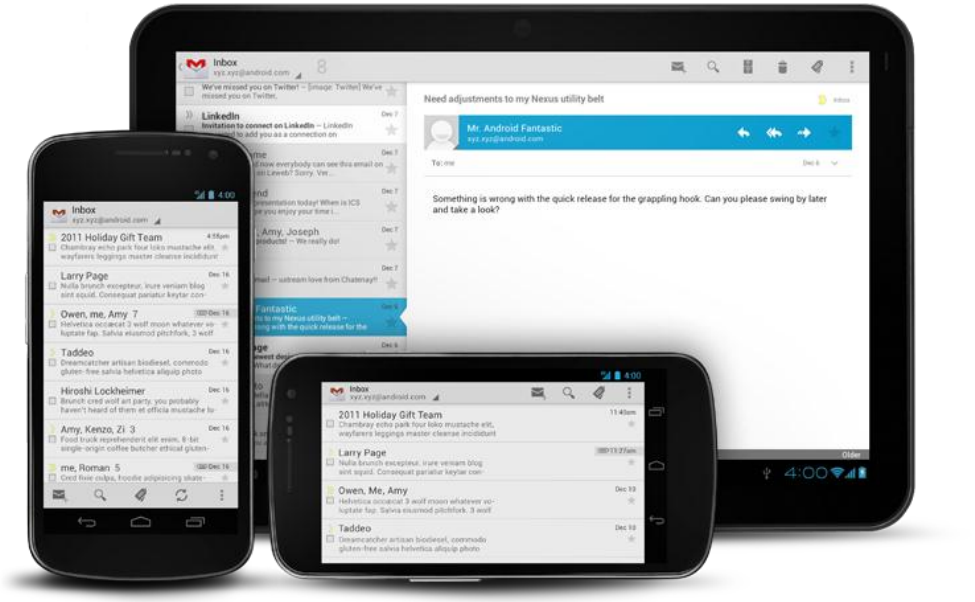
2. 视图控件
如果你的 app 可以用不同的方式展示内容，操作栏里的这个部分可以允许用户切换视图。视图切换控件的样式有下拉菜单或选项卡控件。
如果你的 app 不能支持视图切换，你也可以用这个空间来展示非互动的内容，如一个 app 的标题或更长的品牌信息。

3.操作按钮

在操作栏里，展示出你的 app 中最重要的操作。不能展示在操作栏里的操作，可以自动移到“更多操作”里。

4.更多操作
把不常用的操作放到更多操作里。

适应横竖屏旋转和不同的屏幕尺寸
最重要的界面问题之一，是需要在创建 app 的时候考虑如何在不同的屏幕尺寸下适应方向的旋转。
你可以通过使用次操作栏的方式适应这些改变，允许你在主操作栏的下方或屏幕底部的多个栏中拆分操作栏的内容。



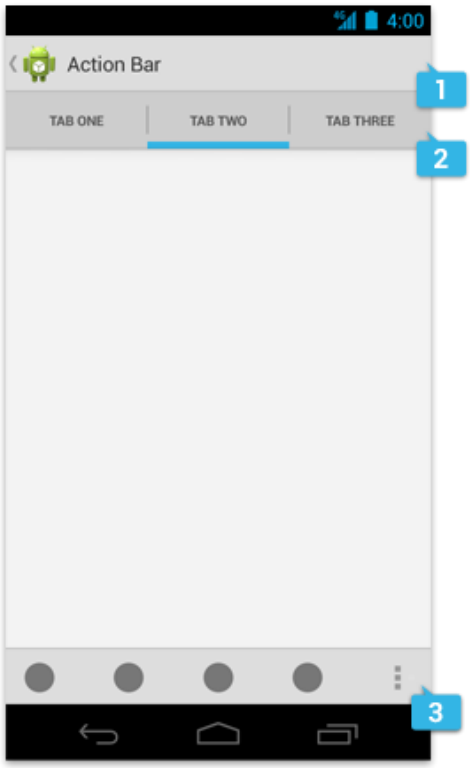
次操作栏，把操作按钮展示在垂直方向上的屏幕下方。

次操作栏的布局注意事项
在多操作栏上拆分操作内容时，你通常可能有三个放操作栏内容的位置：
1. 主操作栏
2. 顶部栏
3. 底部栏

如果用户能从特定的界面去到层级结构的上一层，主操作栏至少要包括向上的符号。

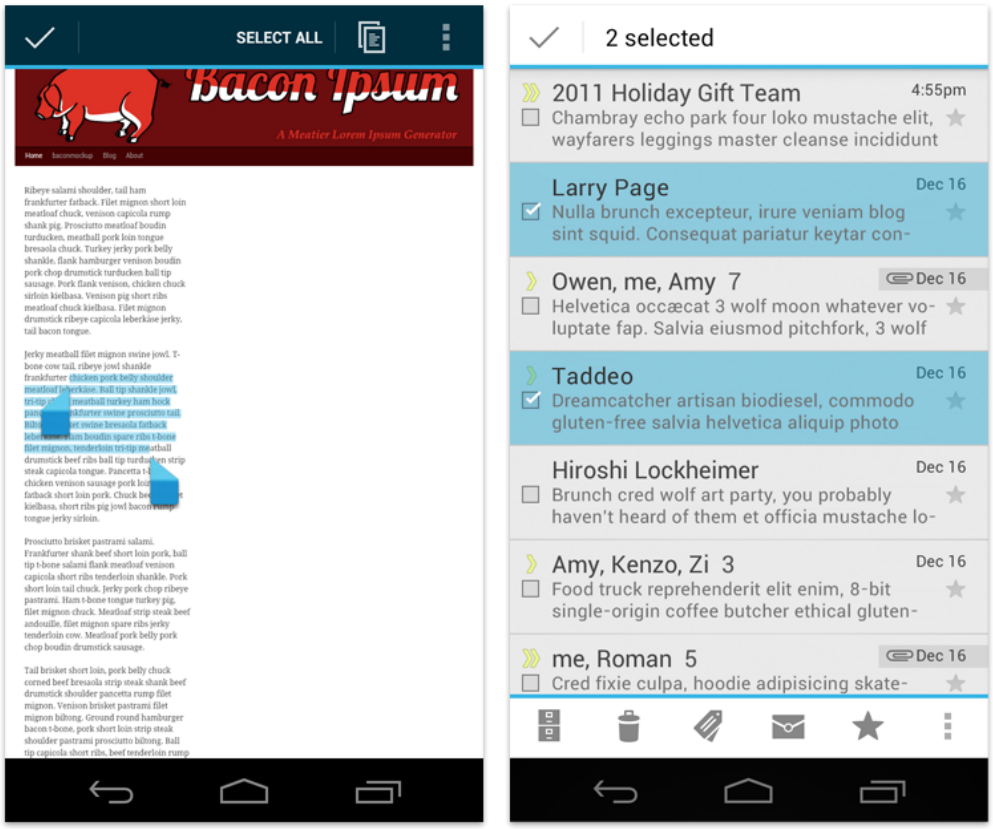
通过使用选项卡 tabs 或顶部栏中的下拉菜单 spinner，来允许用户快速切换你的 app 中提供的各种视图。

如果需要的话，可以在底部栏展示操作和更多操作。



情境操作栏（CAB）

情境操作栏（CAB）是一个在操作特定子任务期间，重叠在 app 操作栏上的临时操作栏。CABs 是通常用于涉及内容或文字选择的任务。



在浏览器和 Gmail 中的情境操作栏

选择一个内容项长按后会出现选择情境操作栏（CAB），并触发选择模式。

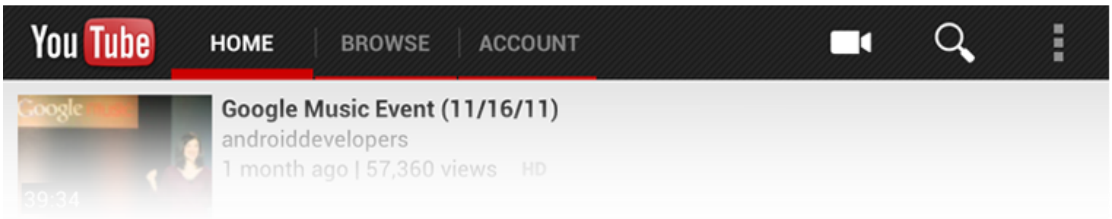
从这里，用户可以：
* 通过触摸内容项，选择添加元素。
* 从情境操作栏触发一个操作，会应用于所有被选择的内容项。然后，情境菜单栏会自动关闭。
* 通过点击导航栏里的返回按钮或者情境操作栏里的复选标记按钮，可以关闭情境操作栏。这将移除情境操作栏中所有的选中态高光。
允许你的用户通过长按被选内容来使用情境操作栏。你可以控制一个情境操作栏里的操作内容，以便插入你想要用户执行的操作。

更多相关信息，见“选择”模块。

操作栏的元素

1. 选项卡 tabs

选项卡展示 app 视图的同时，使其易于浏览和切换。如果你希望用户频繁切换视图，就使用选项卡的样式吧。



有两种类型的选项卡：滚动的和固定的。

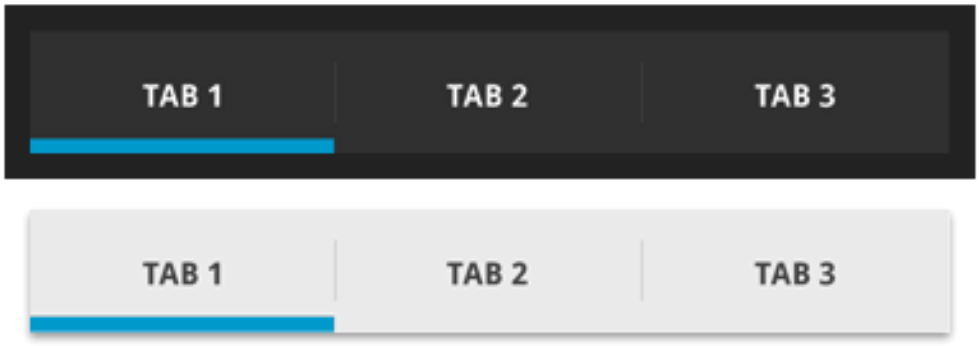
滚动选项卡

滚动选项卡总是撑满整栏的宽度，当前选中的视图项会在正中间的位置。滚动选项卡能通过水平滑动，去到更多的选项卡视图中。

如果你有很多视图，或会灵活的插入而不能确定有多少视图将会被展示，就使用滚动选项卡吧。滚动选项卡总是允许用户通过在内容区域左右横划来浏览视图，和横划切换选项卡一样。

固定选项卡

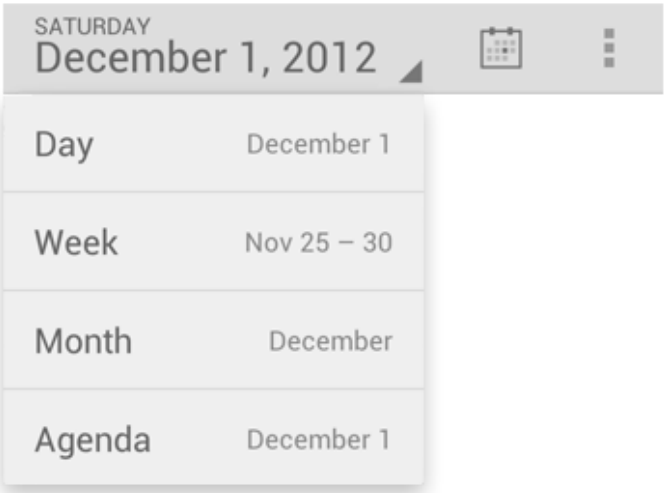
固定选项卡会一直展示在屏幕上，不能像滚动选项卡一样被移开。当屏幕方向改变的时候，主操作栏中的固定选项卡可以移动到顶部栏中。



默认的固定选项卡在 holo 深色主题和 holo 浅色主题下的显示。

2. Spinners 下拉菜单
Spinner 是用户切换 app 视图的下拉菜单。

以下情况使用下拉菜单而非主操作栏：
* 你不想放弃垂直屏幕上专属的选项卡栏。
* 你期待使用你的 app 的用户频繁切换视图。



Calendar app 操作栏上的 spinner。

3. 操作按钮
用操作栏中的操作按钮，展示你的 app 中最重要的功能。思考哪些按钮会最常使用，并相应的排列顺序。由于可用的屏幕大小有限，系统只会显示最重要的操作，把其它放在“更多操作”里。



Gmail 中使用的的操作按钮样式示例

使用 FIT 作为优先展示操作按钮的准则。
F-频繁
* 你的用户是否在访问这个界面的时候，十次中有七次会使用这个操作？
* 他们通常会连续使用几次吗？
* 每次使用都会繁复地增加额外操作吗？

I-重要
* 你希望大家都发现这个操作，因为它特别酷或有卖点吗？
* 是否在某些情况下需要很容易就能操作？

T-典型
* 在相似的 apps 中是否把这个操作作为典型第一层级操作？
* 在使用场景中，如果把它收在更多操作里，用户会感到惊讶吗？

如果符合 F，I 或 T，就放在操作栏里，否则，它属于更多操作。

4. 更多操作 action overflow

操作栏里的更多操作为你的 app 中不常使用的操作提供入口。更多操作的图标只会出现在没有硬件按键 menu 的手机上。



更多操作菜单固定放在右侧。

主操作栏上可以放多少个操作？操作栏的容量取决于以下规则：

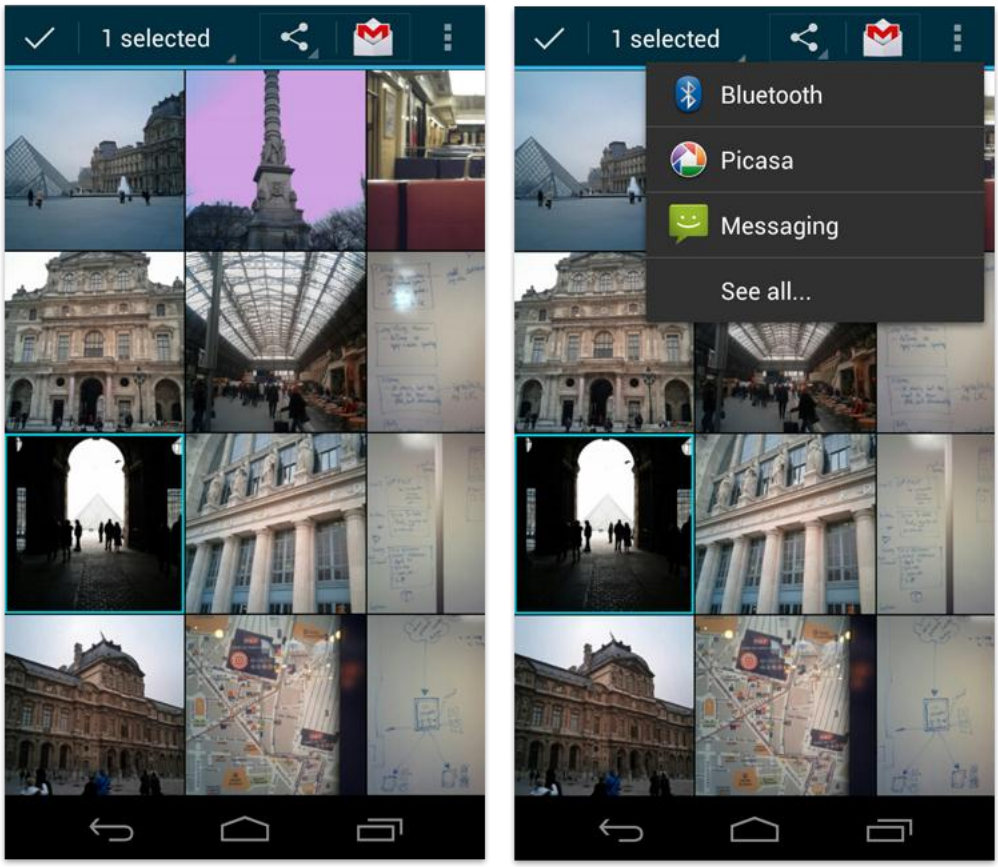
* 主操作栏上的操作按钮不会占用超过整栏宽度的 50%。底部栏上的操作按钮可以占用整栏。
* 屏幕宽度决定将有几个操作项可以放在主操作栏上：
* 小于 360dp＝2 个图标
* 360－499dp＝3 个图标
* 500－599dp＝4 个图标
* 大于 600dp＝5 个图标

Device	Orientation	Horz. Dp	Icons	Example
Nexus S	Portrait	320	2	oo
Galaxy Nexus	Portrait	360	3	oo=
Nexus S	Landscape	534	4	oooo
7" Tablet	Portrait	600	5	oooo=
Galaxy Nexus	Landscape	640	5	oooo=
10" Tablet	Portrait	800	5	oooo=
7" Tablet	Landscape	1024	5	oooo=
10" Tablet	Landscape	1280	5	oooo=

在上图表中，“o”代表操作栏上的图标项，“＝”代表更多操作。

5. 分享内容

每当你的 app 分享内容如图片或影片，在操作栏上使用一个分享操作接口。设计分享操作接口是为了通过展示最近使用的分享服务来达到快速分享的目的，这个接口放在下拉菜单 spinner 按钮的旁边，内容还包括其它的分享选择。



Gallery 中的分享接口使用了 spinner 来扩展分享的选择。

操作栏清单

当你打算拆分你的操作栏，问自己以下问题：

1. 对于这个任务，查看导航有多重要？
如果查看导航对于你的 app 来说很重要，那么使用选项卡 tabs（为了快速切换视图）或 spinners.

2. app 中哪些操作需要直接放在操作栏上，哪些可以移到更多操作里？
使用 FIT 准则来决定操作是展示在操作栏的顶层上还是收起放在更多操作里。如果顶层的操作数量超过了主操作栏可以容纳的数量，那么可以把操作放在底部栏里。

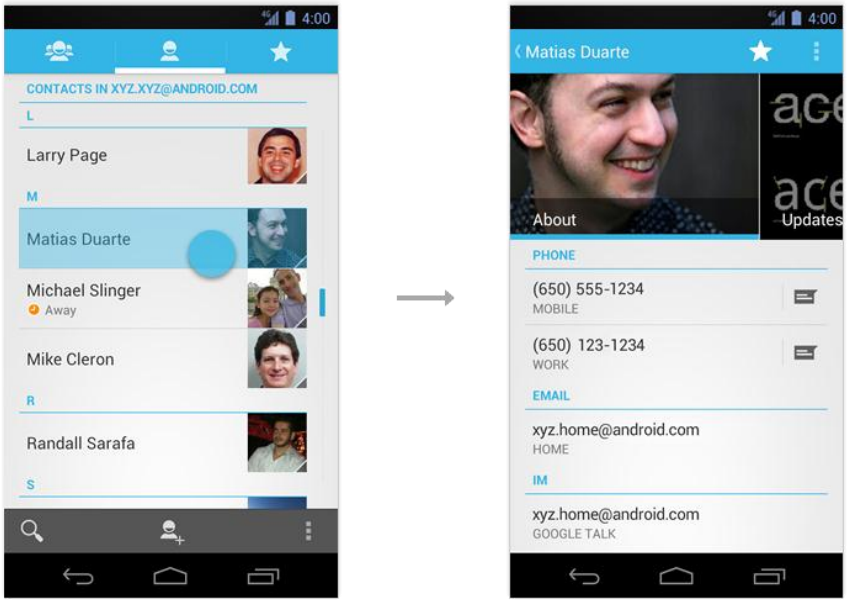
3. 还有哪些足够重要的内容需要长期展示？
有时，显示情境信息对于你的 app 来说也很重要。比如，收件箱中的未读邮件数量，播放器里的正在播放歌曲信息。精心地计划那些你想展示的重要信息，并根据计划调整你的操作栏结构。

多面板布局----------

在创建 Android 应用时，要记住 Androiod 设备有许多不同的屏幕尺寸和类型。要确保你的应用能通过调整其内容来适应不同的屏幕尺寸和方向，以达到一致的平衡和美观的布局。
多面板是一种让你的应用达到这种目标的很好方式。他们允许你在横屏有更多空间时将多个视图结合为一个复合视图，而在空间比较少时则将他们分开。

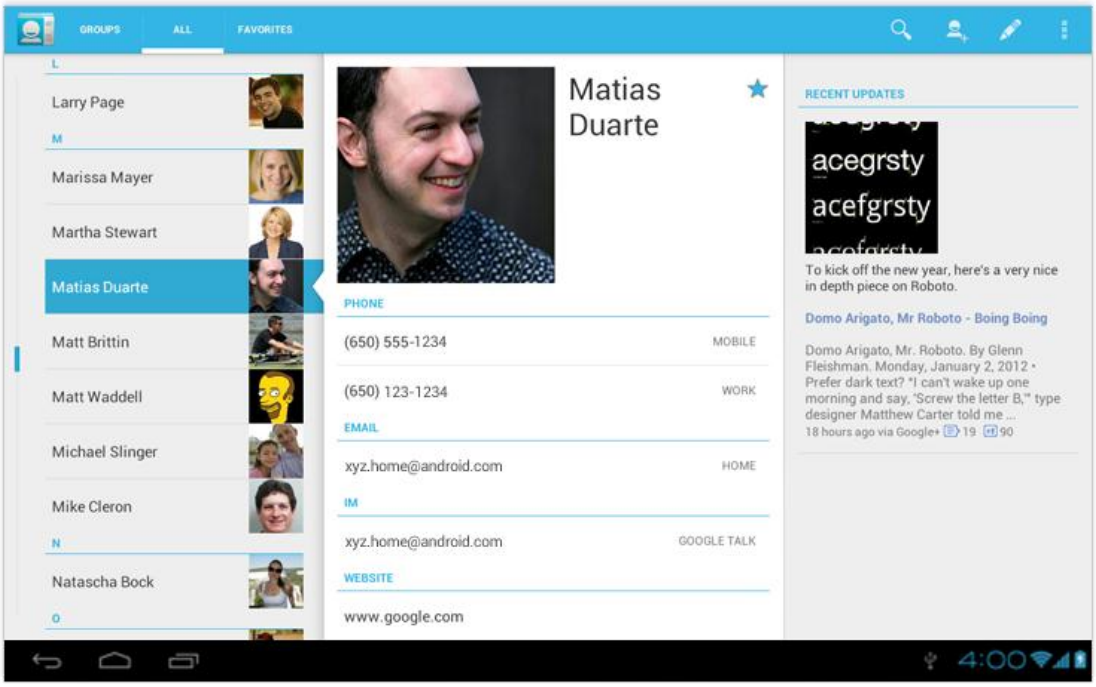
将多个视图结合为一个

在较少小的设备上你的内容通常被分为一个主表格或列表视图和一个详情视图。触碰主视图的条目会打开一个显示条目详细信息的界面。



List view　　Detail view

因为平板比手机有更多的屏幕空间，你能够使用面板来将相关的列表和详细列表融合到一个复合视图中。这就更有效地利用了空间并使得这个应用的导航更为简单。



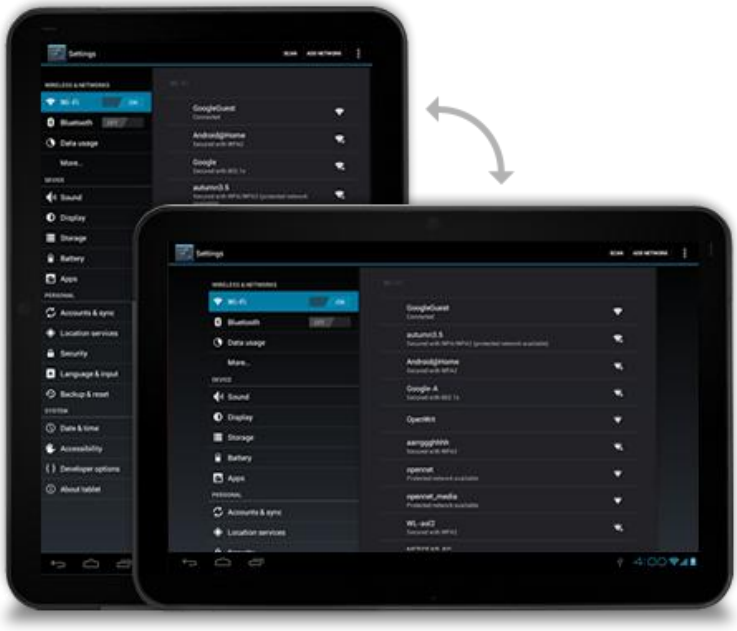
通常，使用右边的面板来显示你在左边面板选择的条目的详细信息。为建立面板之间的关系，要保持左边面板条目的选中状态。

复合视图和朝向变换

不管朝向如何改变，屏幕应该有相同的功能。如果你在某一个朝向使用了复合视图，当用户旋转屏幕时不要分解它。这里有一些技术帮助你在朝向改变之后调整布局以保持功能不变。

拉伸/压缩

调整左边面板列的宽度让两个朝向保持布局的平衡。



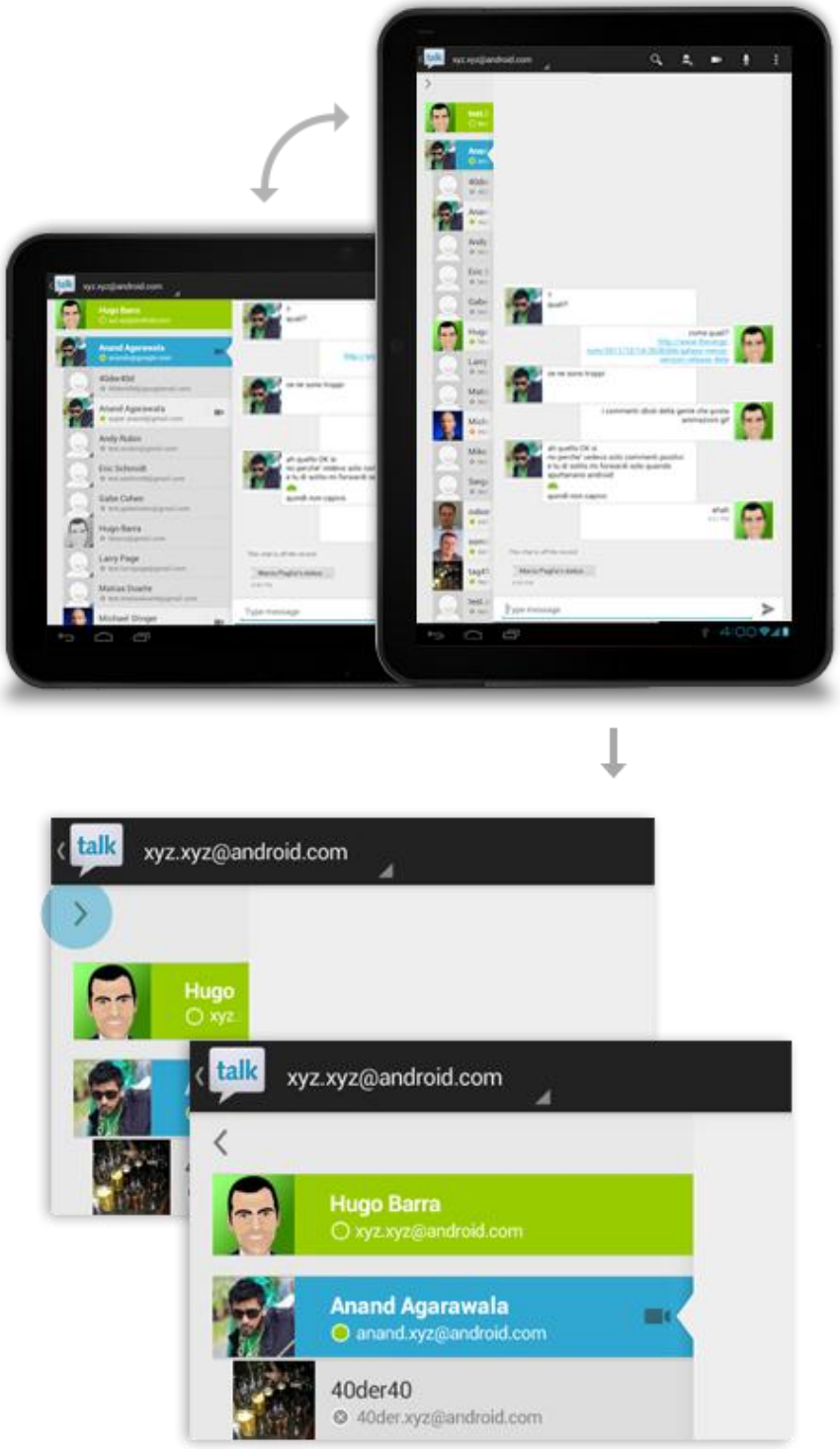
堆叠

重新排列屏幕上的面板来适应朝向。



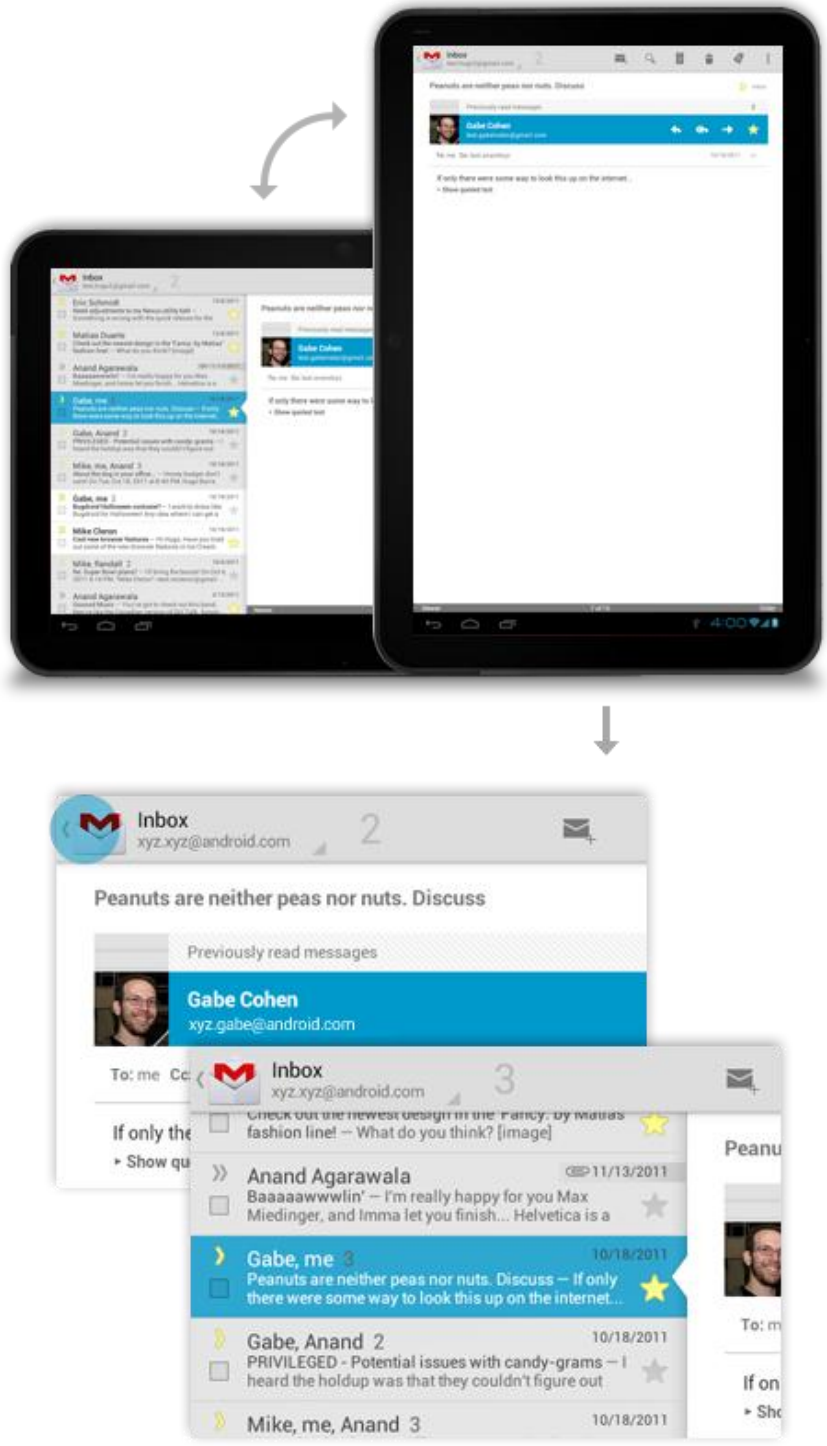
展开/折叠

当设备旋转时，折叠左边的面板视图，使其只显示最重要的信息。提供给用户可以将左边面板内容展开恢复为原有宽度的控件，反之亦然。



显示/隐藏

在旋转视图之后，以全屏视图显示右侧的面板。使用在操作栏上的“向上”图标显示左侧面板并可以导航到另一封电子邮件。通过触碰详细面板的内容来隐藏左侧面板。



清单

* 提前计划你的应用该如何适应不同的屏幕尺寸和屏幕朝向。
* 在屏幕朝向改变时找到最好的方法重新组织复合视图的面板。
* 寻找机会将你的视图整合到具有多面板的复合视图中。
* 确保你的屏幕在朝向改变后保持相同的功能。

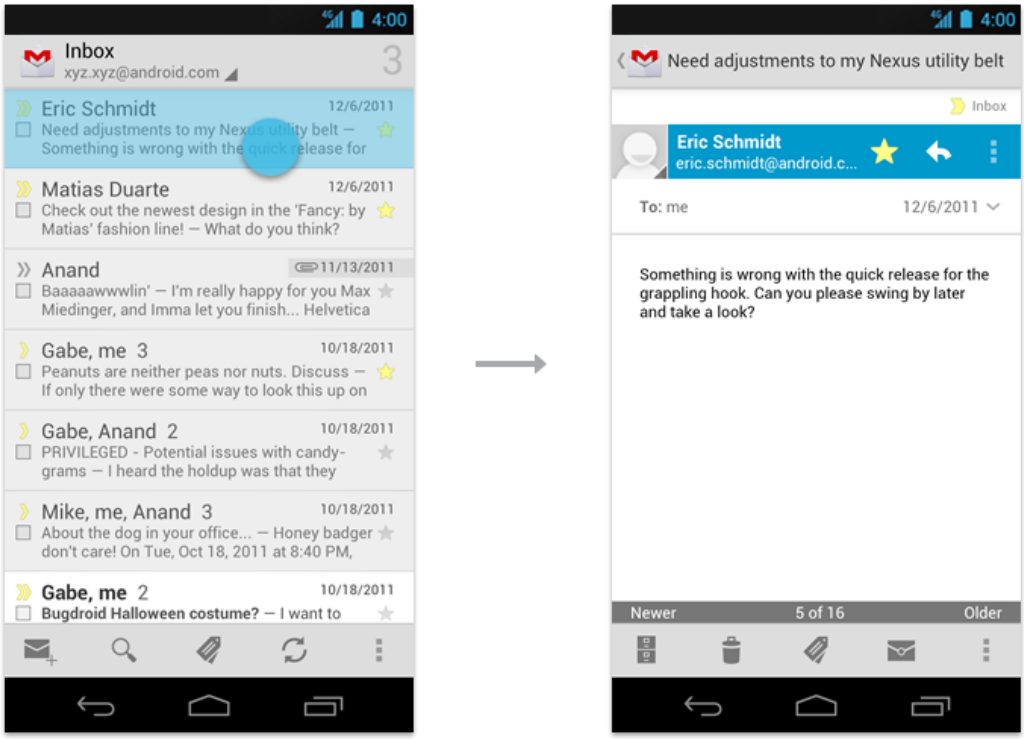
横划视图----------

有效率的导航是设计良好 app 的基石之一。虽然 app 通常以分层方式建立，但横向导航也能够扁平化垂直层次结构，并使得访问相关内容项更便捷。横划视图可使用户通过一个简单的

手势高效率地在内容项间切换，并因此带来浏览和享受更流畅的体验。

在详情视图间横划

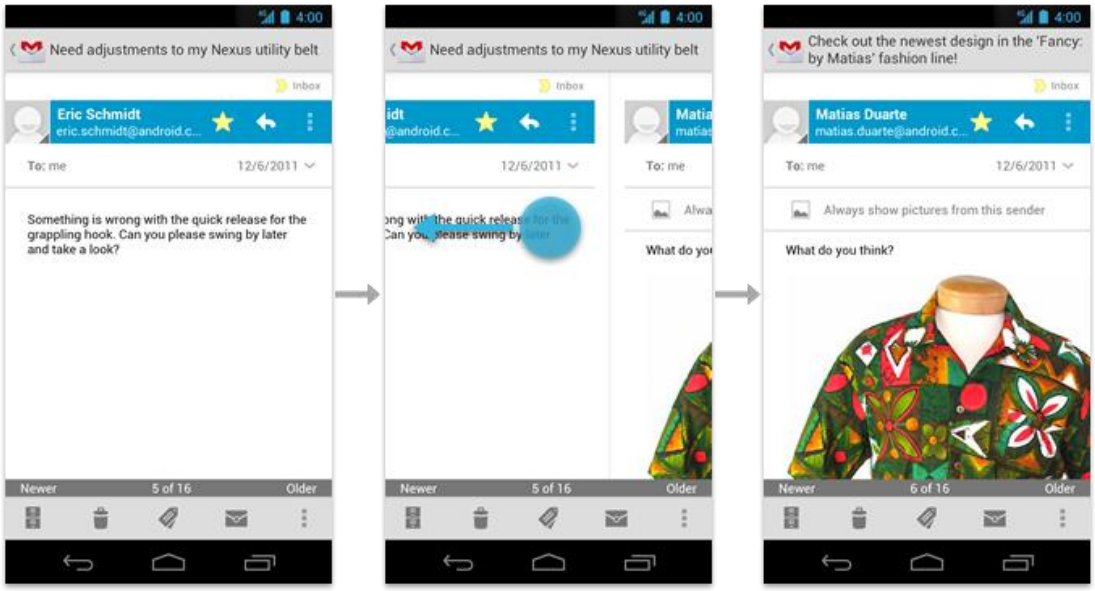
一个 app 的数据通常被组织成“主列表/详情”的关系：用户能够看到一列相关内容项，比如说图片，聊天记录或电子邮件，然后选择其中的一个内容项并在另一个界面查看详细内容。



主列表（左）和详情（右）视图

在手机上，因为“主列表”和“详情”视图虽然在不同的界面，这通常要求用户在列表和详情视图间来回切换，也被称为“pogo-sticking”。

在用户想要以连续的方式查看多个详情内容项时，要避免使用“pogo-sticking”，而是通过横划手势来导航到“下一个”/“前一个”详情视图。



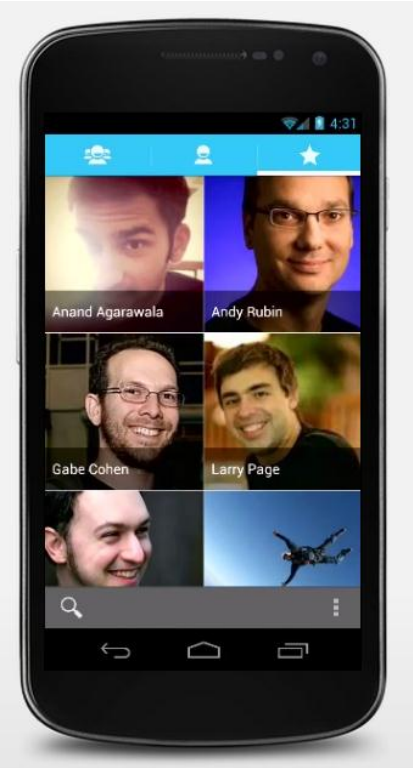
用横划手势在连续的电子邮件消息之间导航

在选项卡 tabs 之间横划

如果你的 app 使用了操作栏 **tabs**，请用横划在不同的视图间导航。

清单

* 用横划在详情视图或 tabs 之间快速导航。
* 视图间的过渡要伴随着用户的横划手势。不要等到手势完成才执行视图间的切换。
* 如果你之前用按钮来完成“前一个”/“下一个”的导航，那么用横划来代替这种方式。
* 考虑在你的详情视图中添加情景信息让用户知道当前可见的内容在列表中的位置关系。

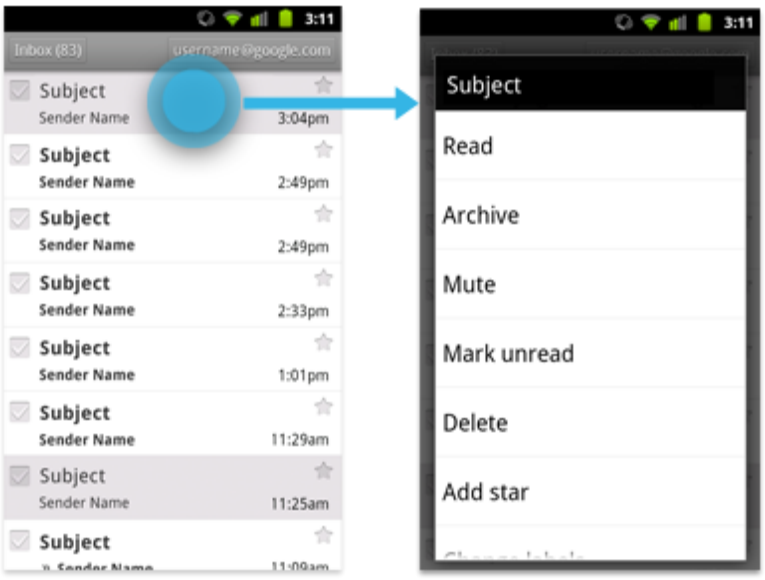


选择----------

Android 3.0 引入了长按手势，即是触碰位置一段时间，作为选择数据的全局手势。这个手势会影响到你在 app 中如何处理多项选择和情境操作。

有什么改变？

在之前的 Android 版本中，长按手势统一被用来调用被选内容项的情境操作。
这种模式在 Android 3.0 中发生了改变。通过把情境操作和选择管理集成到一个被称为情境操作栏(contextual action bar, CAB)的新元素中，长按操作现在被用来选择内容。



传统的长按操作用来显示情境操作菜单

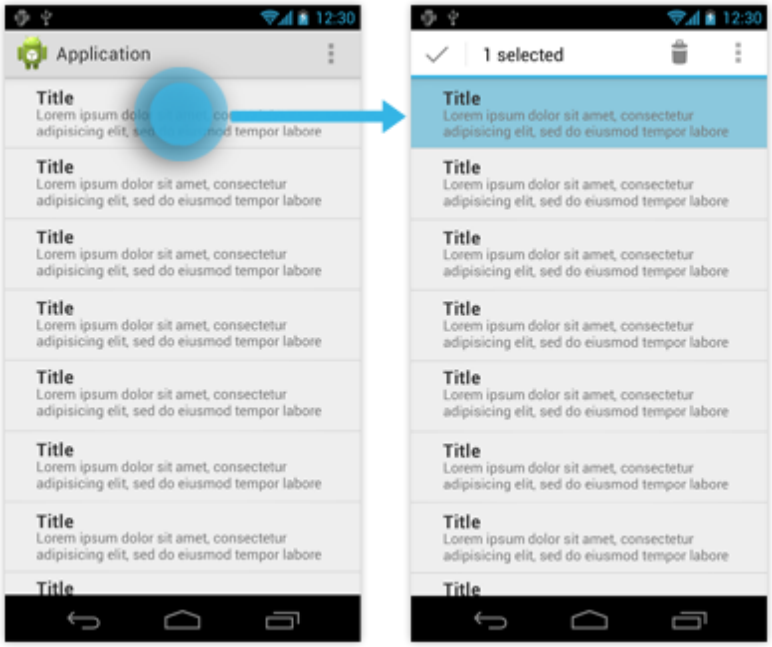
使用情境操作栏(CAB)

CAB 是当数据被选择时覆盖在你的 app 的当前操作栏上的一个临时操作栏。它在用户长按某个可被选择的内容项后出现。



从这儿用户能够：

* 通过触碰选择多个数据项。
* 通过 CAB 触发一个操作应用到所有高亮的数据项上。之后 CAB 会自动消失。
* 通过导航栏上的返回按钮或 CAB 上的√(checkmark)按钮取消 CAB。这会移除 CAB 和所有选择内容的高亮。



选择 CAB 操作

你可以决定哪些操作和元素在 CAB 中出现。根据操作栏中的指南来决定哪些操作项要在顶级呈现，而哪些可以移到更多操作中。

动态调整 CAB 的操作项

在大多数情况下，当用户添加更多的内容项到选中状态时，你需要动态调整 CAB 中操作项。应用于单个被选内容项的操作不必适用于多个被选内容项。



当多个数据项被选中时调整 *CAB* 中的操作内容

清单

* 如果你的 app 支持多个内容项选择，好好利用情境操作栏(CAB)。
* 将长按手势专门保留用于选择操作，而不要用它来显示传统的情境菜单。
* 如果你不支持一个列表中的多项选择操作，长按应该不起任何效果。
* 用安排 app 的操作栏的操作项的方式来安排在 CAB 中要呈现的操作项。

通知----------

你的app可以通过通知系统将重要事件信息告知用户，例如聊天应用或日历事件的新消息。要创建一个流畅、舒适、友好的 app，认真地设计你的通知系统很重要。通知代表着你的 app 的声音，并且促进了 app 的个性化。不想要或不重要的通知会烦扰用户，谨记要明智使用。

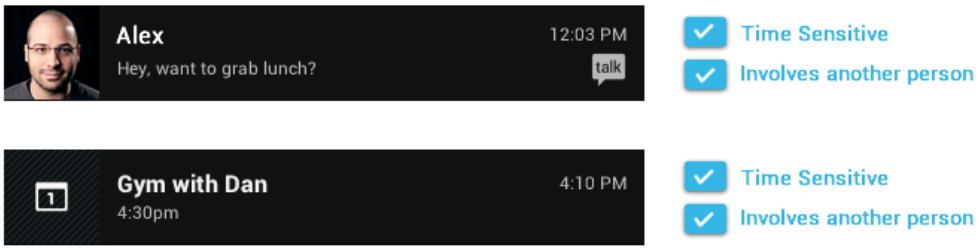
何时显示通知

要创建一个人们喜爱的应用，很重要的一点是要认识到用户的注意力和焦点必须被小心保护。用一个可能会引起软件开发人员共鸣的例子来说，用户不是一个能被调用返回值的函数。用户的聚焦更像是一种线程的资源，通知会立刻打断用户线程，用户会处理并关闭这个阻碍性的通知。

虽然 Android 的通知系统已被设计成当用户聚焦某个任务时能快速告知用户，但要有负责的态度去决定是否创建一个通知还是很重要的。

表现良好的 app 只在需要它发言时才发言，但确实有一些少量的情况是需要用一个自发的通知来打断用户。

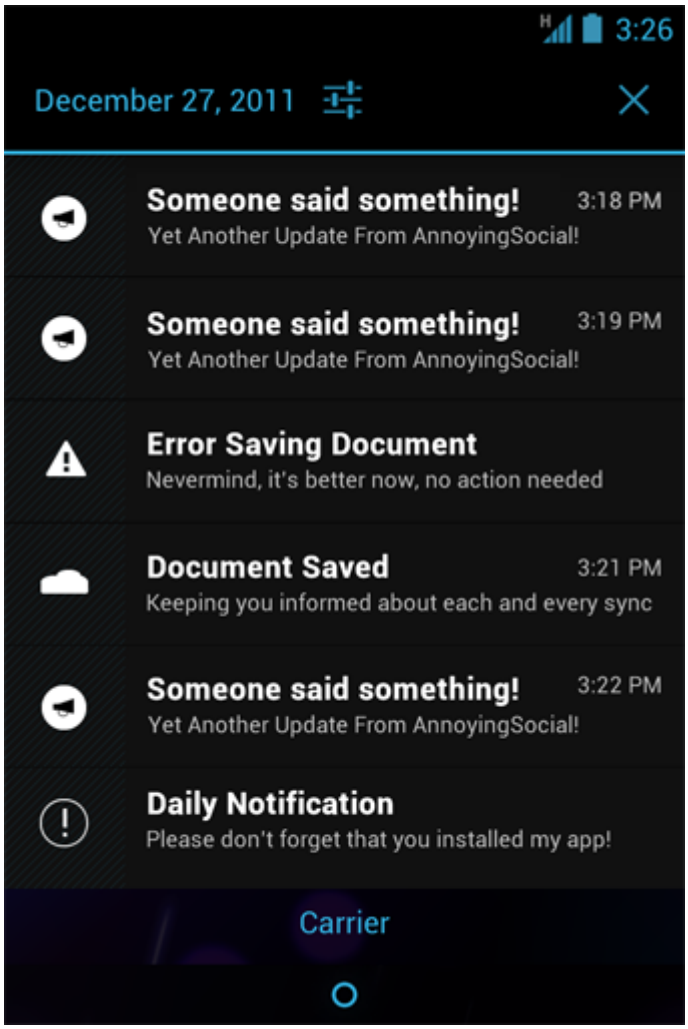
通知应主要用于对时间敏感的事件，特别是当这些同步事件涉及其他人。例如，传入的聊天信息是一种实时和同步的交流方式，有另一个用户正在积极地等待你的回应。日历事件又是另一个需要创建通知并抓住用户注意力的好例子，因为事件是紧迫的，并且涉及到其他人。



何时避免显示通知

但仍然有一些不应该使用通知的情况：

* 不要通知用户那些不是直接针对他们或不紧急的信息。例如某个社交网络的异步信息就不应该中断用户。
* 如果相关的新信息目前正显示在屏幕上，就不要创建一个通知，而是通过 app 本身的 UI 直接在情境中通知用户有新信息。例如一个聊天应用不应该在用户正在与其他人聊天时创建系统新通知。
* 如果系统可以简单地维护自身而不用涉及用户操作，就不要以一些低级的技术问题去打断用户，比如保存或同步信息，或者是更新 app 等。
* 如果在发生错误时 app 可以快速地将自己从错误中恢复而不需要用户执行任何操作，就不要用错误通知去打断用户。
* 不要通知一个用户不能自行启动或停止的服务。
* 不要只是为了让你的品牌出现在用户面前就创建多余的通知，这样的通知只会阻挠并可能疏远你的用户。提供用户少量的更新信息而又保持他们对你的应用的注意力的最有效方式是开发一种小插件，使用户可以选择放在他们的 Home 界面上。



设计指南

1. 使其私有化

对于其他用户发送过来的通知条目（比如一条消息或更新状态），请包含用户图片。

记住在通知里面要包含 app 图标作为第二图标，这样用户能够区分出是哪个 app 发出的通知。



2. 导航到正确的地方

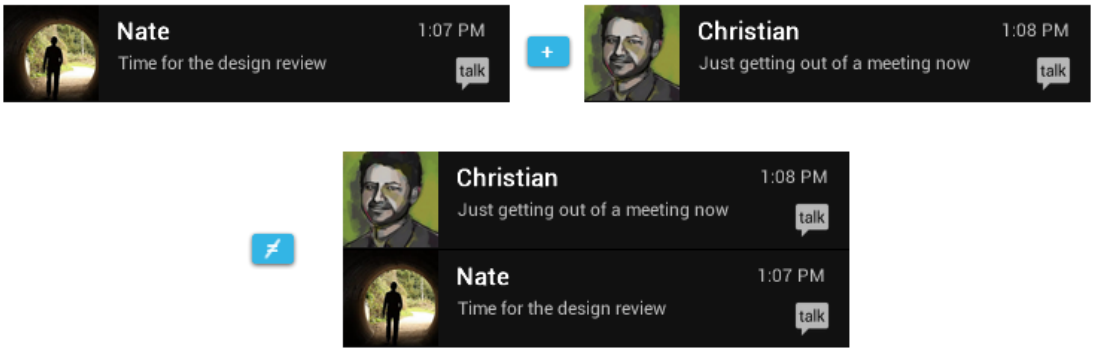
当用户触碰一个通知时，要把你的 app 在用户可以使用的地方打开，并且操作的是通知里反映的内容。在大部分情况下这会是一条单独内容条目的详情视图（比如一条消息），但如果通知有多条（详见下面的多条通知），那么这可能是一个总结性视图并且引用了多个内容条目。

如果是用户被带到了 app 非顶层级，要使他们可以通过系统导航栏返回键来到 app 顶层。更多信息请参阅导航中的系统到应用的导航章节。

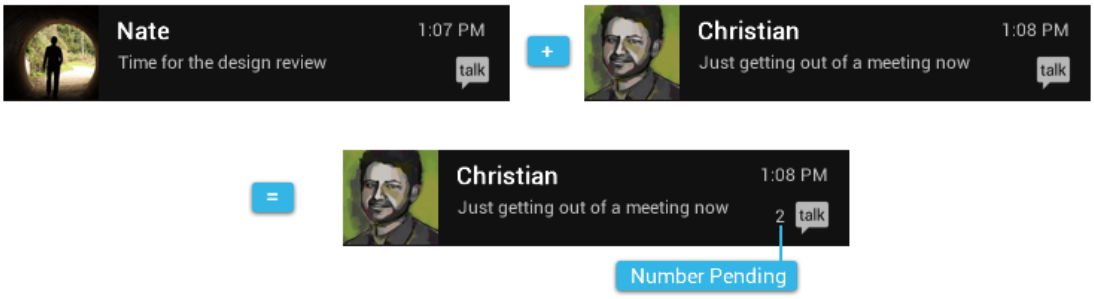
3. 对时间敏感事件的时间戳
默认的，Android 的通知系统包含了一个在右上角的时间戳。考虑时间戳是否在你的通知情境中有价值。如果时间戳没有价值，考虑事件是否足够重要到需要用一个通知来吸引用户的注意力。如果通知足够重要，再决定你是否想要选择不显示时间戳。
如果用户想要知道这个通知是多久之前发生的，就包含时间戳。使用时间戳的一个好例子是通讯通知（email，消息，聊天信息，语音邮件），用户可能需要时间戳信息来理解消息的上下文或撰写一个回应。

4. 合并多条通知
当你的 app 创建了一个通知而另一个相同类型的通知还在待处理状态时，不要另外再创建一个新通知对象，而是把他们合并显示。
一个多条的通知构建了一个总结性的描述并且允许用户知道多少同一类型的通知正在等待被处理。

不要：



要：

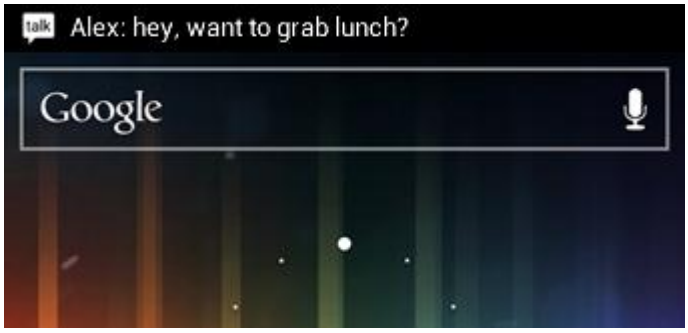


如果你的总体信息和详情信息显示是不同的界面，多条通知打开你的 app 的视图可能与单项通知打开的视图不同。
例如一个单项的电子邮件通知，打开通常应该就是邮件的内容，而一个多条通知打开后是一个收件箱的视图。

自我清理
就像是日历事件，某些通知会警告用户某个事件会在某个具体的时间点发生。在那个时间过去之后，这个通知可能对用户来说就不再重要了，你就应该考虑自动删除它。这同样适用于

用户已经看过的聊天对话或听过的语音电子邮件，用户不应该需要手工单独地去采取操作来移除这些通知。

提供通知的预览
提供一个简短的文本，可以在通知到来的时候让用户对通知内容有快速、简短的预览。ticker 文本只显示很短的一段时间，然后自动地隐藏。



让通知可选
用户应该总能控制通知。在你的 app 设置中添加通知设置条目，使用户可以选择取消通知。

使用不同的图标
用户在通知栏看一眼就应该能够区分出当前等待处理的是什么类型的通知。

要：
观察 Android 应用已经提供的通知图标并创建你的 app 图标，使其有足够的不同。

不要：
使用颜色来区分你的应用和其他的应用。通知图标通常是黑白的。

与通知交互
通知在通知区域以图标表示，并且能够通过打开通知抽屉的方式（向下划动）进入。



在抽屉内，通知时间顺序的排序是把最近的通知排在最顶部。触碰一个通知会打开相关的 app 去到与该通知相关的详细内容。向左或向右横划动一条通知会将其从抽屉中删除。

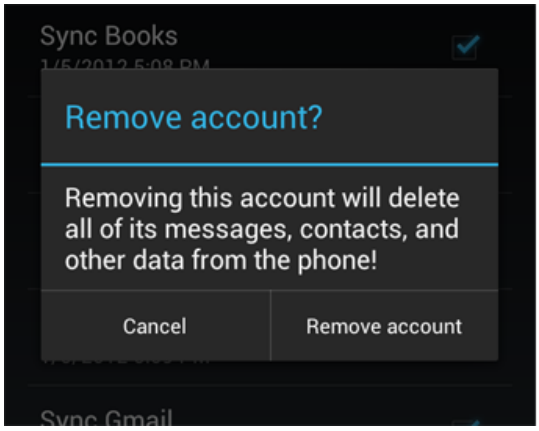
在平板中，通知区域集成在屏幕底部的系统栏中，通过触碰通知区域的任意位置来打开通知抽屉。



进行中的通知
进行中的通知让用户知道正在后台运行的进程。例如一个音乐播放器会在通知系统里宣布当前正在播放的曲目，并且在用户停止它之前持续如此。它们也能够展现一个长任务的反馈，比如下载一个文件或是对一个视频编码。进行中的通知不能够被手工删除。



对话框和信息提示条（toasts）是用于反馈而不是通知
对话框和信息提示只应该在用户操作你的 app 时用来进行即时的反馈。例如，对话框能用来确认用户准确地理解了某项操作，而信息提示条能够反馈一个操作已经被成功地执行。





兼容性

向后兼容

Android 3.0（平板电脑系统）中的重要变化包括：
* 反对使用硬件导航按钮（返回、菜单、搜索、主页），支持通过虚拟控制处理导航（返回、主页、最近使用）。
* 在操作栏中，用固定的模式来支持菜单的使用。

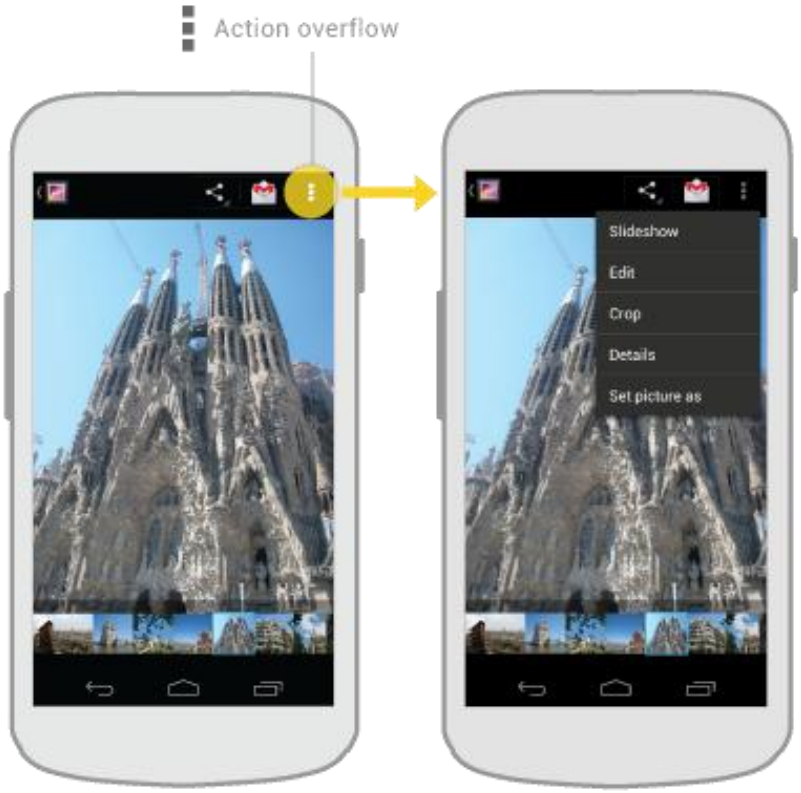
Android 4.0 把对平板电脑的这些改变带到手机平台。

Android4.0 适用于旧的硬件和 app

1. 具有虚拟导航栏的手机

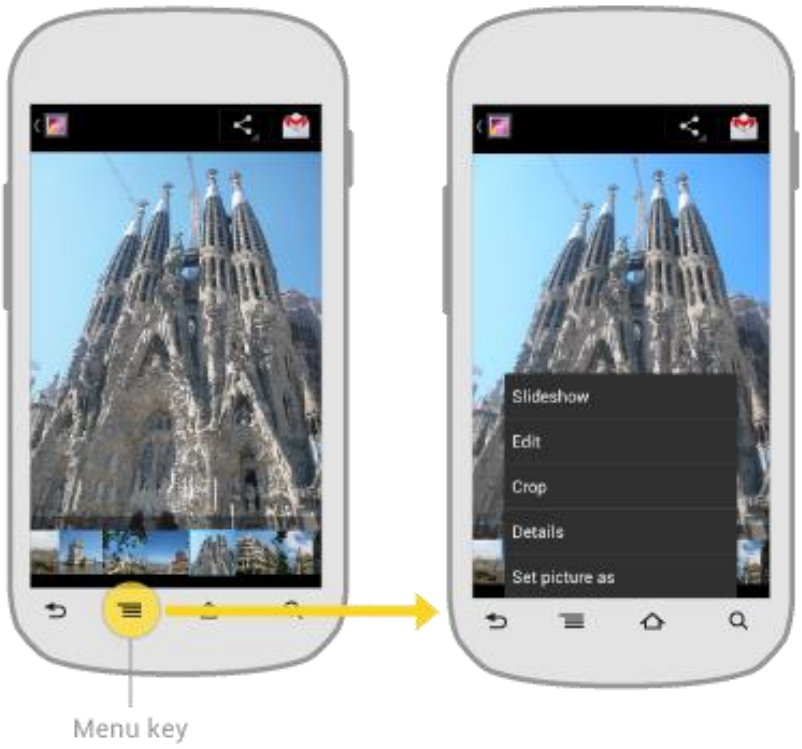
为 Android3.0 以上版本编写的 Android app 在操作栏里显示功能的操作。不合适或不足够重要的操作会放到更多操作里。

通过触摸操作栏里的更多操作菜单可以访问那些操作。



2. 具有物理导航按键的手机

有传统物理导航按键的 Android 手机不会在屏幕的底部展示虚拟的导航栏。相反，在菜单硬件按钮中可以使用更多操作菜单。点击 menu 的物理按键后出现的操作浮层样式于上一个例子的更多操作菜单一样，但是展示在屏幕的底部。



3. 旧式的 apps 在新的有虚拟导航栏的手机上

当你在有虚拟导航栏的手机上运行为 Android2.3 或更早版本系统设计的 app 时，更多操作菜单的控制会出现在虚拟导航栏的右侧。通过触摸会以传统 Android 菜单的样式来展示更多操作。

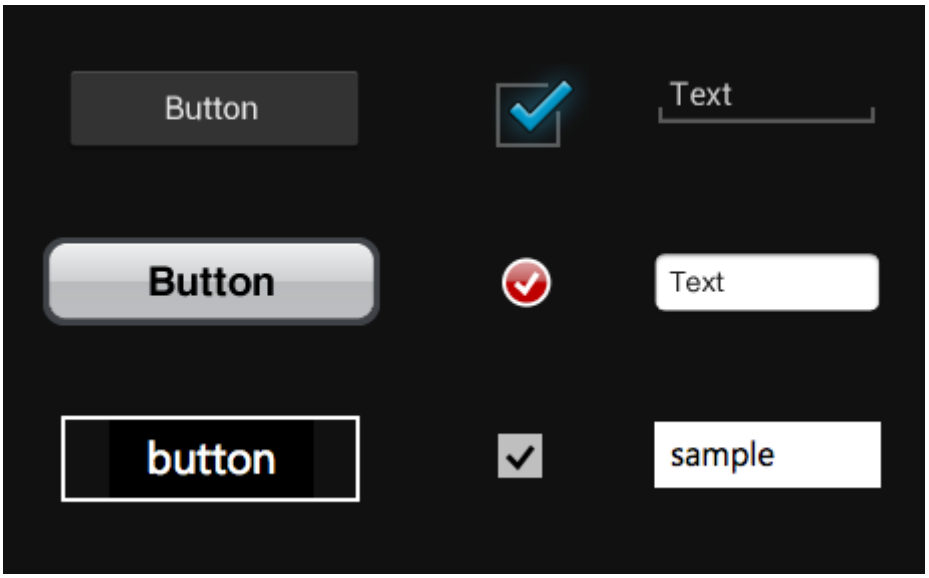


Android 就是 Android----------

很多开发者都想把自己的 app 发布在不同的平台上。如果你正在计划 Android 平台上的 app，请记住不同平台有不同的规则和习惯。在某一个平台上看似完美的做法在另一个平台上不一定会好。虽然“拿一个版本走遍全部平台”的想法可以帮你节省时间，但与平台的不一致很可能会让你疏远用户。请遵循以下一些准则避免常见误区：

不要模仿其他平台的 UI 元素。

每个平台都有与其他区别开、自己精心设计的一套 UI 元素。例如一些平台提倡在按钮上用圆角，一些提倡在标题栏使用渐变。某些情况下，这些元素可能有异曲同工之处，但设计却不同。当你创建你的 Android app 时，不要从其他平台直接挪用 UI 元素，不要模仿它们特定的行为。查看“积木/控件”这一章可以了解更多关于 Android UI 元素和在系统主题中风格指引。除此之外，还可以参考 Android 平台的 app 感觉一下这些元素是怎么应用到实际的 app 中。如果你想定制自己主题的 UI 元素，要注意参考你品牌的特性，而不是参考别的平台的习惯。



Android，iOS 和 Windows Phone7 的 UI 元素范例

不要延续其他平台的特定图标。

平台都会为自己的基本功能提供一套图标，如分享，新建一个文档或删除。
当你移植你的 app 到 android 时，请留意把这些 Android 平台的特有图标换上。
你可以在 Android SDK 上找到各种各样的图标用到你的 app 上。



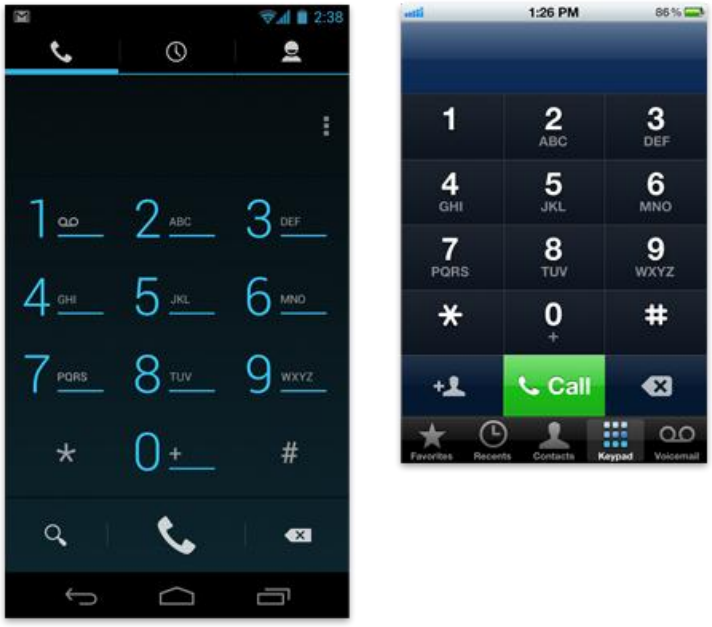
Android，iOS 和 Windows7 的图标范例

不要使用底部选项卡栏。

其他平台会使用底部为操作选项卡栏。但 Android 平台的习惯是在屏幕的顶部作为操作选项卡栏。而且 Android apps 还要利用底部栏来展示次操作栏。

你应该遵循这些指引来创造与 Android 平台其他 app 统一的产品体验，避免造成混淆。

想了解更多关于如果正确的使用操作栏，请查看操作栏。



Android 使用顶部操作选项卡栏而 iOS 使用底部为操作选项卡栏。

不要在操作栏使用“返回”按钮。

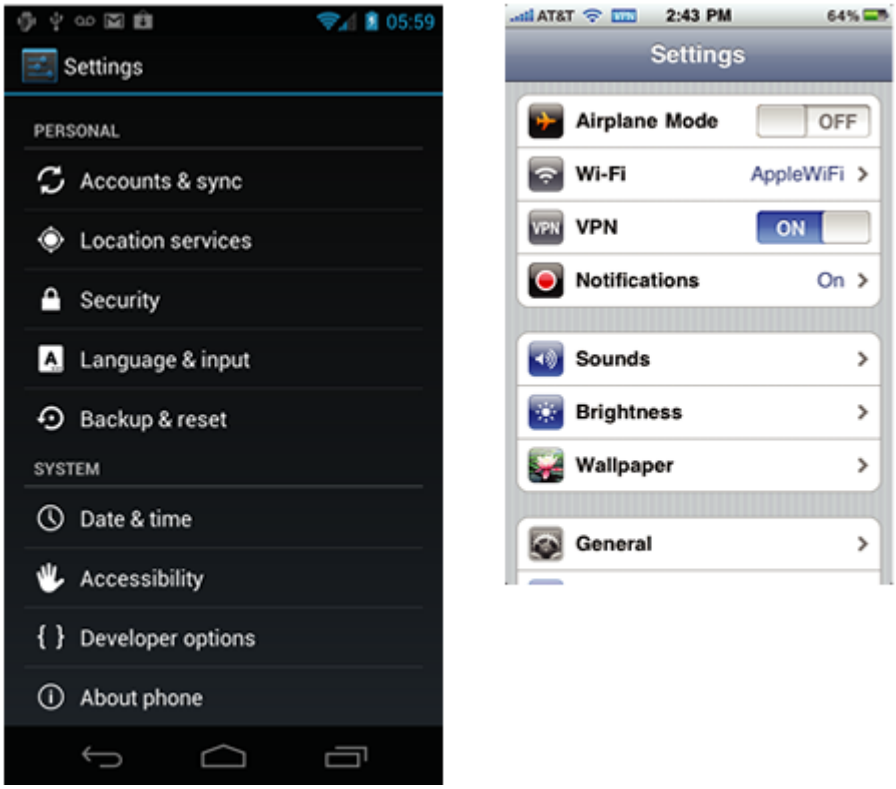
其他平台会在顶部操作栏使用“返回”按钮让用户返回到程序上一层级的页面。但 Android 里不同，分为“返回”和“返回上一层”两个概念，在导航栏的“返回”可以返回上一部操作，在操作栏上的为“返回上一层级”按钮。想了解更多，请查看模式中的导航。



Android 的顶部操作栏的“返回上一层级”，iOS 的称谓“返回”的返回上一层级按钮

不要在列表内容使用向右箭头。

其他平台一个常用的做法是在列表项目最右边有一个向右的箭头，表示查看详情。Android 不会使用这样的箭头。要避免使用这样的箭头，不要让用户觉得平台不统一和疑惑。



Android 的设置面板列表不带向右箭头 对比 iOS 的设置面板

设备的独立性

记住你的 app 会在不同屏幕大小的设备上运行。可以通过为不同屏幕大小和分辨率的设备创建视觉版本以及利用多面板的布局概念来适应你的 UI 到不同的设备上。

想了解更多，请查看设备和显示和多面板布局。

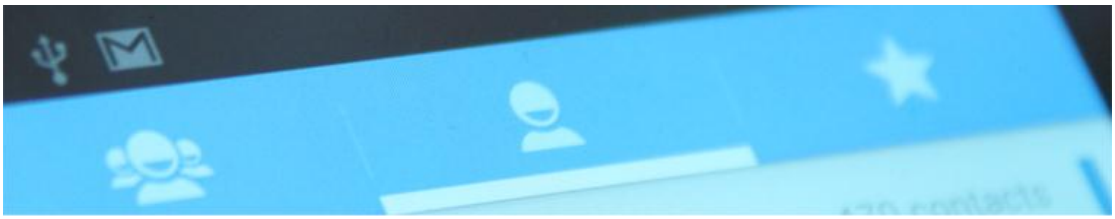
积木/控件 BUILDING BLOCKS ▸

为你创造出色的 app 准备的控件库。



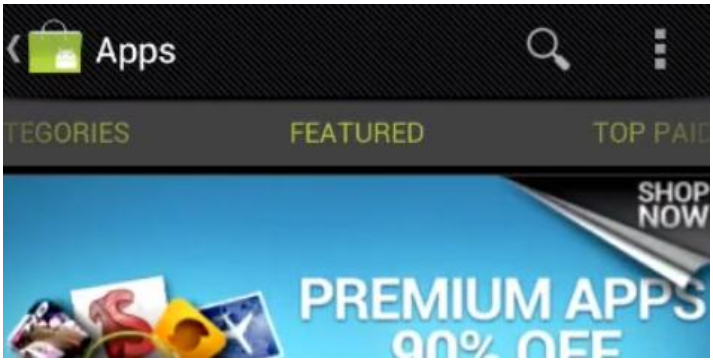
选项卡 Tabs

操作栏中的选项卡可以很轻松探索 app 中的不同功能，或浏览不同分类的数据集。



滚动选项卡

滚动选项卡控件可以包含比普通选项卡控件较多的项目。通过横划左和横划右来操控。



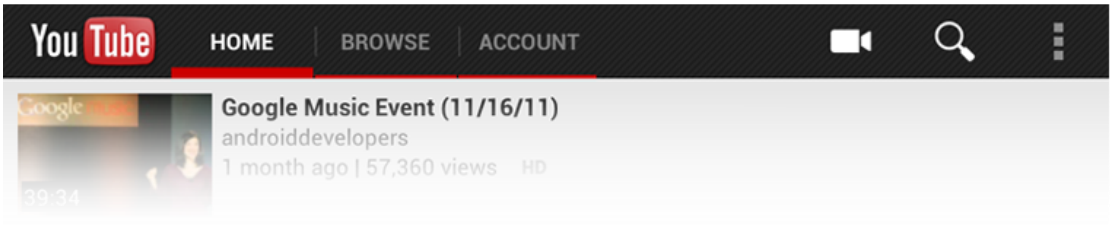
Android 市场中的滚动选项卡

固定选项卡

固定选项卡显示所有项目，触摸标签即可导航。



浅色和深色的固定选项卡



Youtube 的固定选项卡

堆叠选项卡

如果导航在你的 app 中必不可少，你可以再堆叠多一个单独的操作栏选项卡。这有利于在较窄的屏幕的快速切换。



列表

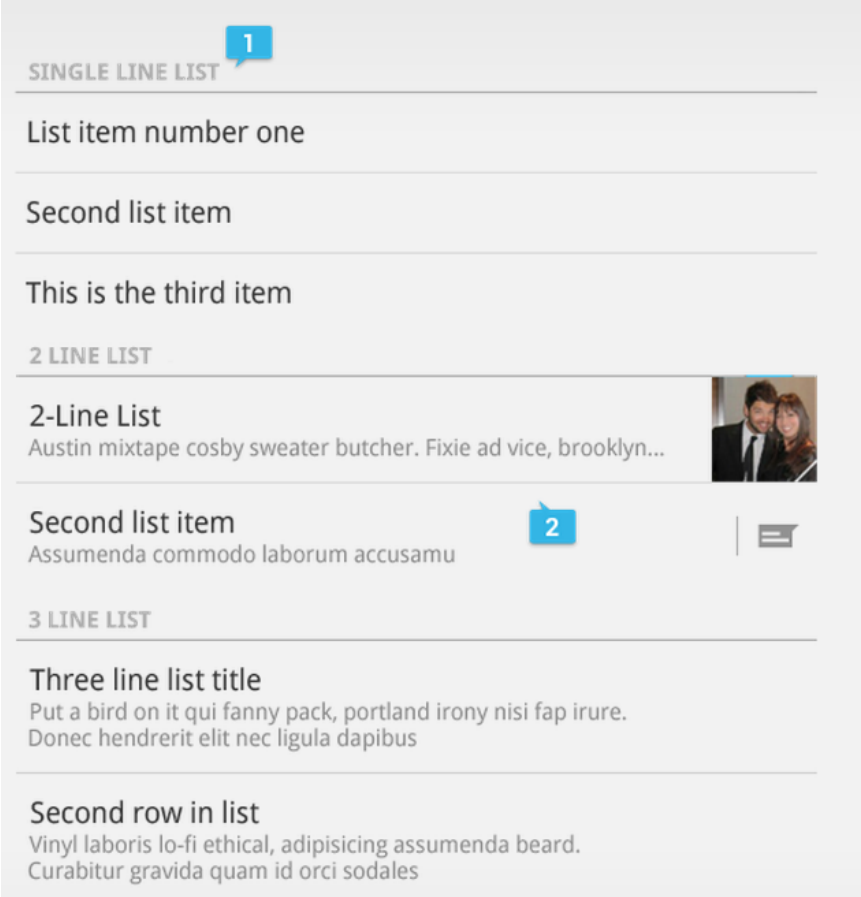
列表可以纵向展现多行内容。它可以被用作数据选取及向下排列的导航。

1. 章节分割

使用章节分割来分组组织你的内容，便于扫描。

2. 行

列表可以容纳不同的数据组织形式，包括单行，多行，包含图标，复选框和操作按钮。



网格列表----------

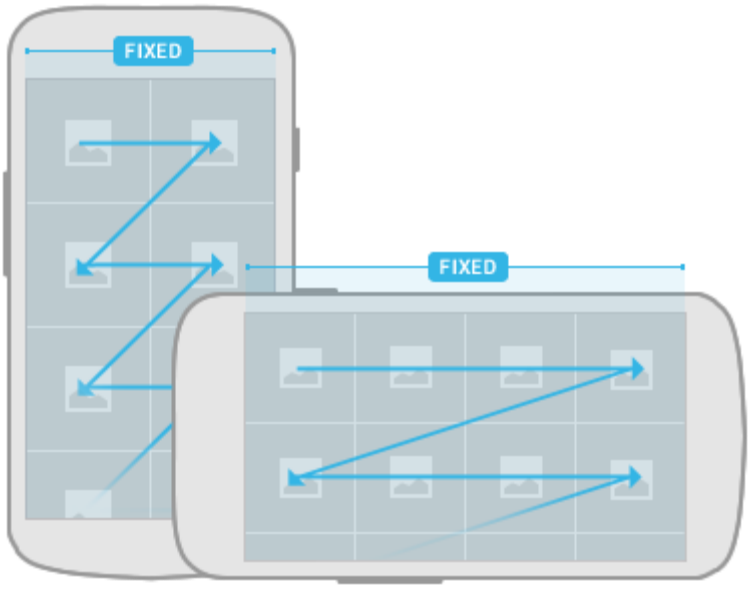
网格列表是标准列表的替代选择。它最适合那些使用图像展示的数据。对比简单的列表，网格列表既可以水平滚动也可以垂直滚动。



通用网格

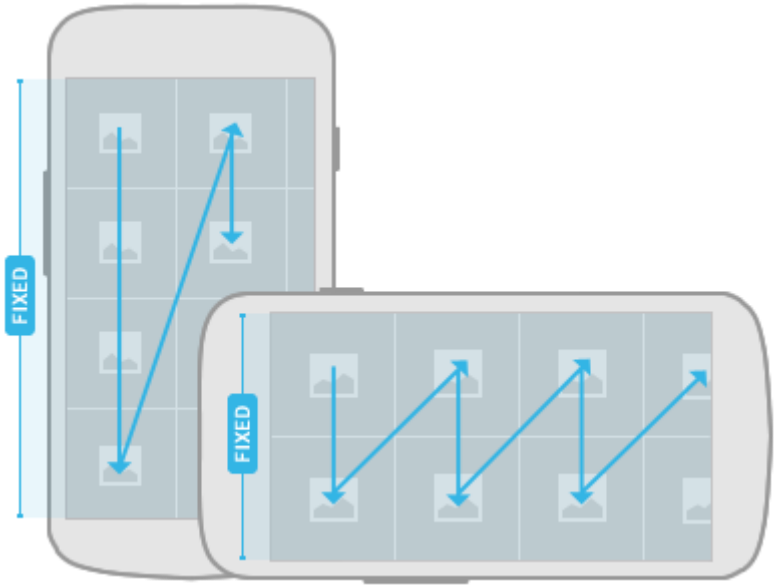
网格里的对象从两个方向组织，其中一个为滚动的方向。滚动方向决定了网格的组织顺序。滚动的方向是不确定的，通过切断内容（边缘只显示一部分内容）的方式可以让用户知道哪边才是滚动方向。

避免在两个方向都滚动。



1. 垂直滚动

垂直滚动的网格列表项是按照西方的阅读习惯进行排序：从左到右，从上到下。显示列表时，底部内容切断，只显示一部分，让用户可以清晰知道垂直滚动向下可以看到更多，当用户旋转屏幕时也要这样。



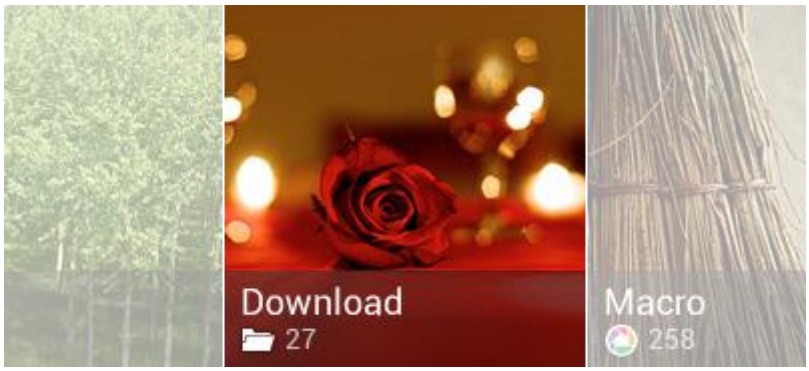
2. 水平滚动

水平滚动对比垂直滚动列表，排序稍有改变：先从上到下，再从左到右。同样使用相同的切断右边缘内容的方法暗示滚动方向。

不要同时使用滚动选项卡与水平滚动网格列表，它们会冲突。如果你使用滚动选项卡，请配合使用垂直网格滚动列表。

带标签的网格列表

可以使用标签显示网格内容的附加信息。



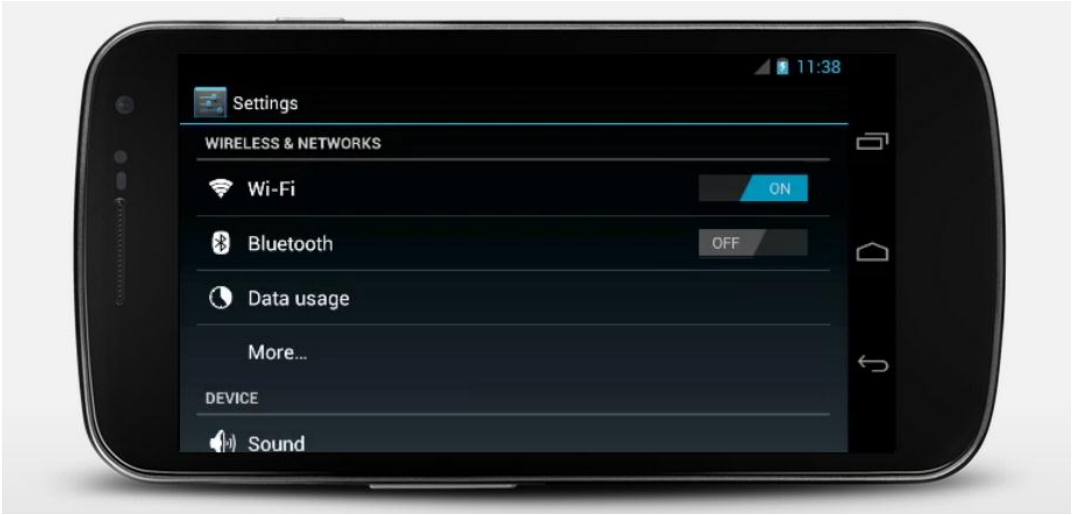
样式
使用半透明面板覆盖在内容上来展示你的标签。这可以使你控制好背景与标签的对比，使背景很亮的同时标签仍然很清晰。

滚动

滚动让用户使用一个滑动的手势查看更多内容。滚动的速度与手势的速度成正比。

滚动提示

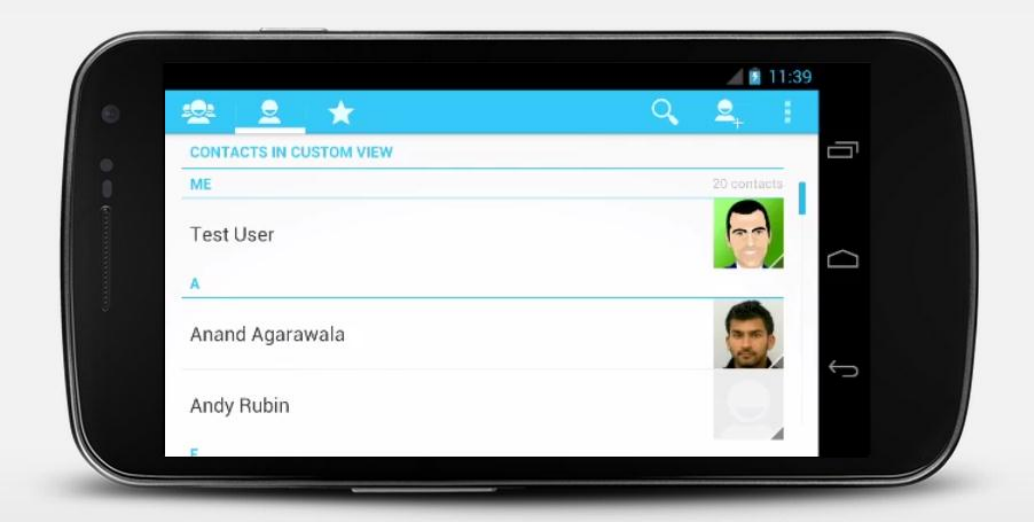
滚动时展示滚动提示（不滚动不展示），表明显示内容在全部内容中的位置。

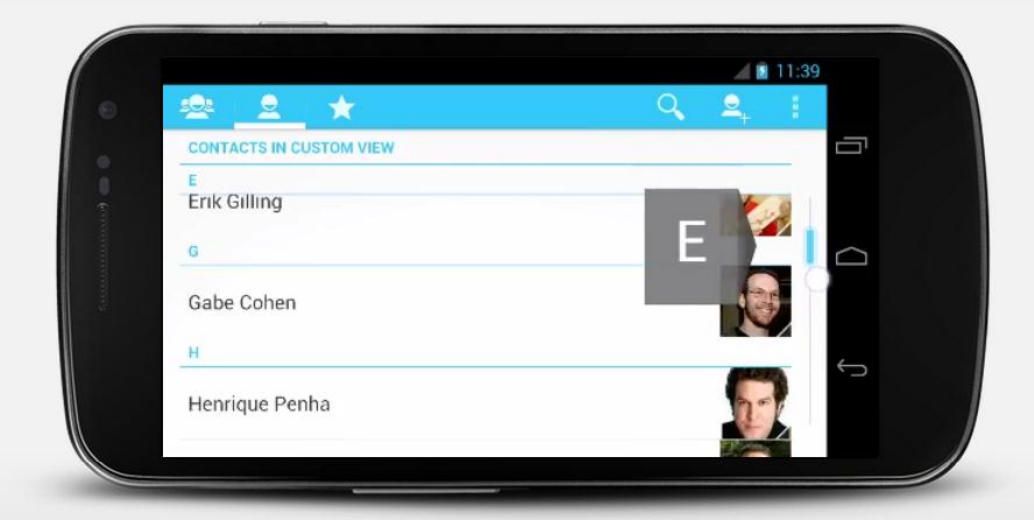




索引滚动

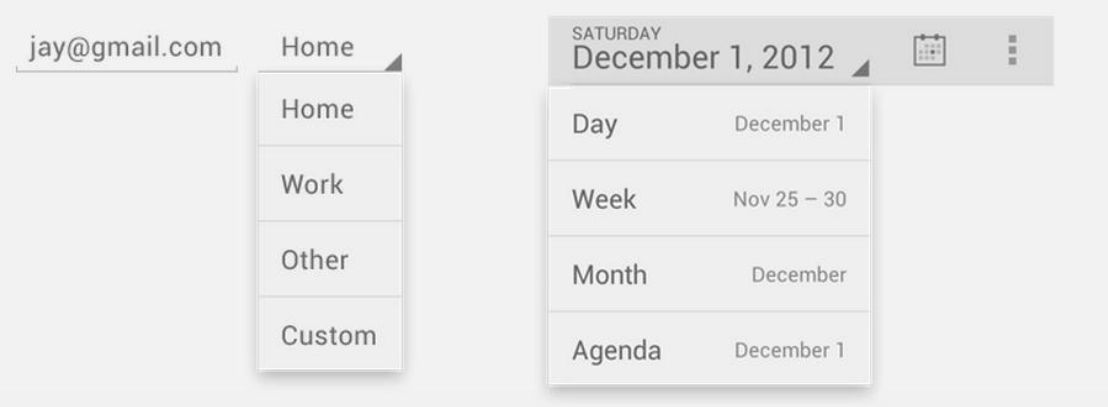
除了传统的滚动，带有字母列表的索引滚动也是一个快速找到对象的方法。索引滚动，即使在用户不滚动的时候也能看到滚动提示。触摸或拖动滚动条显示现在位置的字母。





Spinners (下拉菜单)

Spinner 提供了一个快速选择的方式。默认展示了选中的内容。触摸下拉框展示所有可选内容，用户可以从中选择。

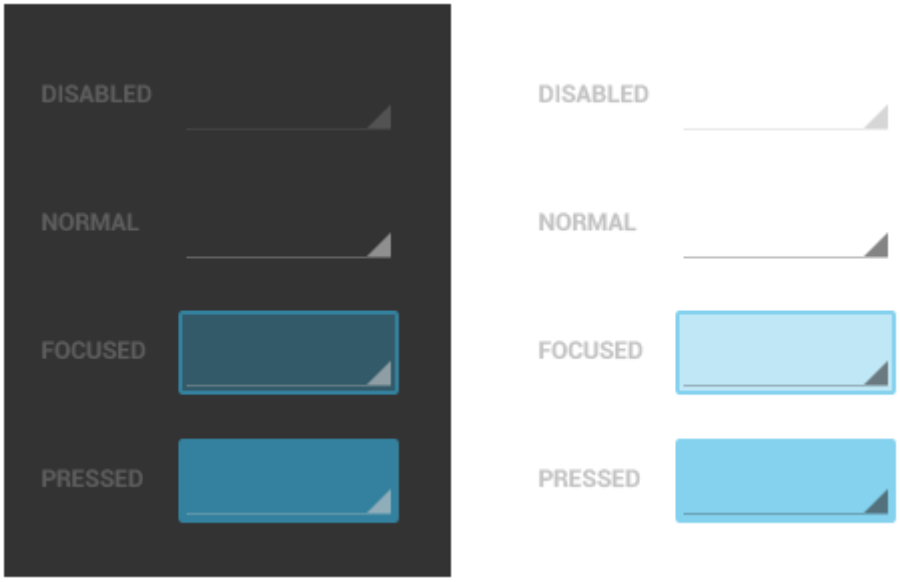


下拉菜单的形式

下拉菜单是一种很有用的选取数据的形式。它能与其他组成部分很好的融合。下拉菜单既是简单的数据输入，又可以跟其他输入控件结合使用。例如，在输入框填写一个联系人的地址，使用下拉菜单选择这个是他的家庭地址还是工作地址。

在操作栏的下拉菜单

使用操作栏的下拉菜单切换视图。例如，gmaiL 使用下拉菜单以允许进行帐户或常用的分类之间切换。下拉菜单对于切换你的 app 的视图很有用，但如果切换的内容很重要时，请使用选项卡。



深色与浅色主题的下拉菜单。

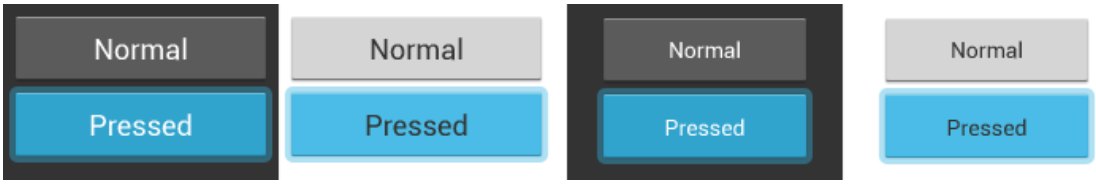
按钮

按钮包括了文本和图形，它传达出当用户触摸它时，行为就触发的信息。Android 支持两种不同的按钮：基础按钮和无边框按钮。两种都可以包含文本和图形。



基础按钮

基础按钮是传统的有边框和背景颜色的按钮。Android 支持两种样式的基础按钮：默认按钮和小按钮。默认按钮的字体稍大，适合在内容框外显示。小按钮的字体和最小高度较小，适合与内容一起显示。当需要与其他 UI 元素对齐时，使用小按钮。



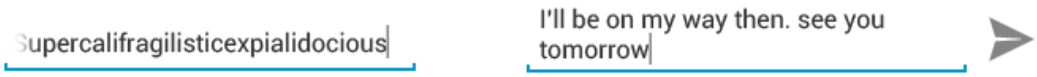
无边框按钮

无边框按钮类似于基础按钮，但他们没有边界或背景。它也可以同时带有图标和文本。无边框按钮在视觉上比基本按钮更轻巧，并能很好地与其他内容融合。



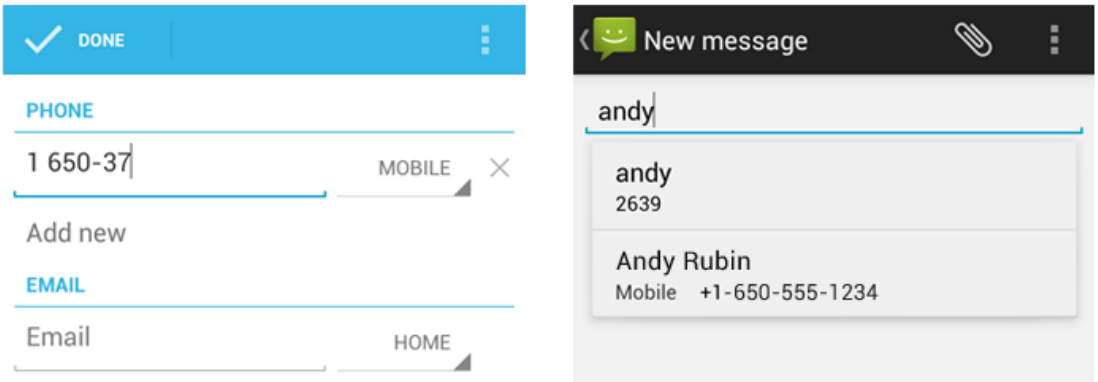
文本输入

文本输入可以是单行也可以是多行，触摸一个文本输入的区域，光标就会自动放置，自动显示键盘。除了输入，文本输入位置还有其他操作，例如文本选择（剪切，复制和粘贴）和自动检查的数据查找。



1. 单行和多行

单行的输入区域当输入到边缘时会自动把内容往左边滚。多行输入区域当输入到边缘时会自动换行。

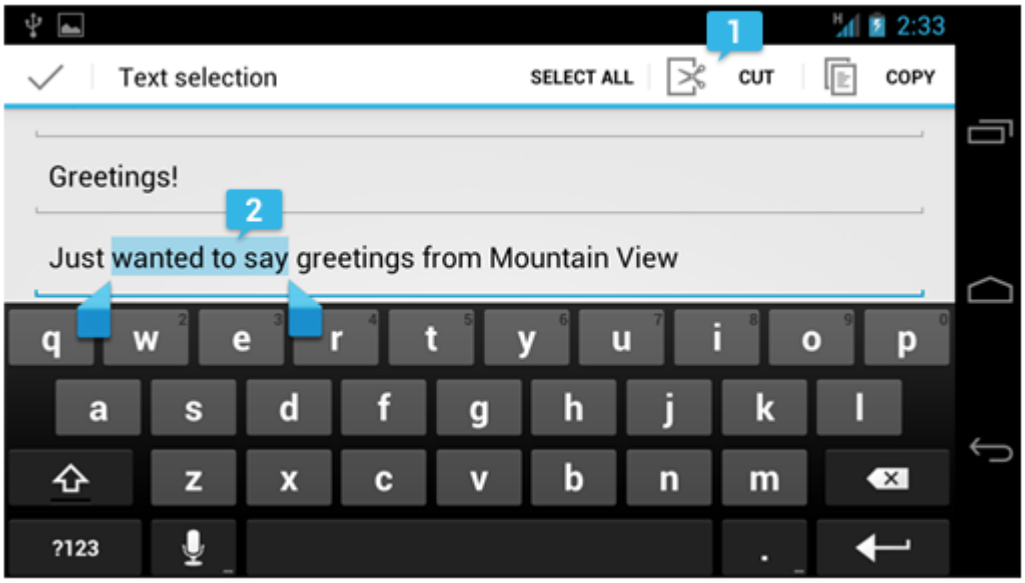


3. 文本输入类型

文本输入可以有不同的类型，如数字，信息或电子邮件地址。输入类型决定了允许的字符类型，字符类型决定了虚拟键盘。

3. 自动完成的文本输入
使用自动完成的文本输入，通过弹出窗展示实时的补充或搜索结果，使用户输入信息可以更准确，更有效。

文本选择
用户可以长按选择在文本输入中的任何字。这个操作触发一个文本选择模式，用于扩展的选择或操作选定的文本。选择模式包括：

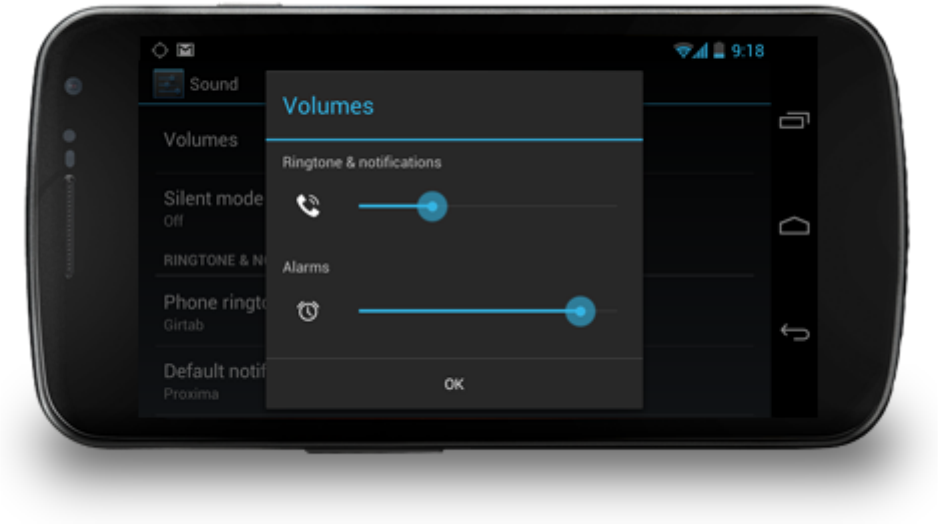


1. 情景操作栏
情景操作栏（CAB）显示可用来执行选择的选项：剪切，复制和粘贴，但 app 根据需要，可以插入其他命令。

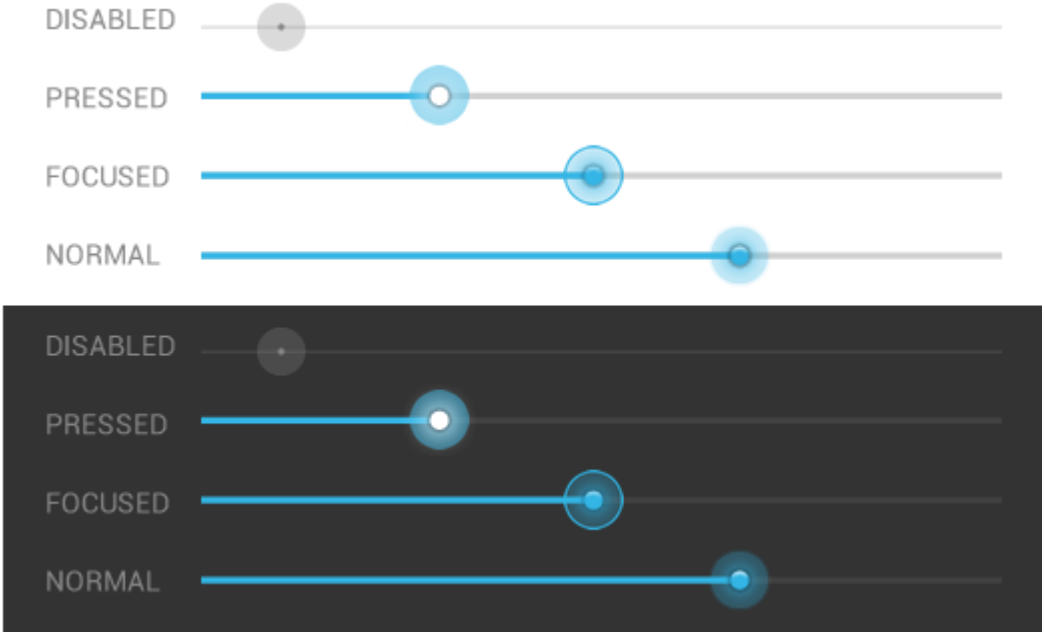
2. 选择手柄
可以拖动选择手柄选择更多或更少的文字。

滑块----------

交互式的滑块用于从一段范围内进行选择。最小值在最左，最大值在最右。滑块的自然交互使得如音量、亮度、色彩饱和度、强度等设置选用滑块为最好的选择。



例子
使用滑块设置铃声音量。该值可以通过硬件音量控制或此滑块完成。



反馈

当操作的过程会花费相当一段时间时，要提供还正在进行进程或已经完成的视觉反馈。

进程

可以的话，使用一个进度条告诉用户已经完成的百分比。



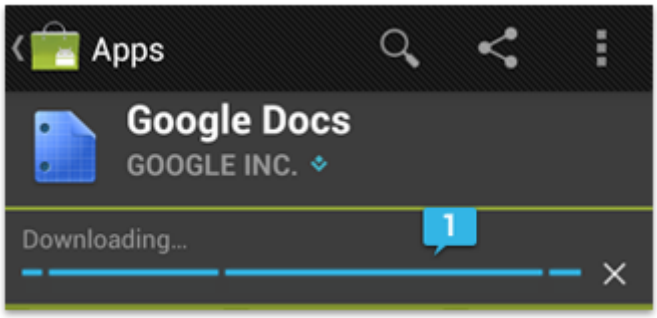
进度条应该从从 0%到 100%。避免把进度条设定到一个更低的值或使用一个进度条代表多个事件，这样做毫无意义。如果你不确定一个进程需要到长时间，使用一个不确定的进度条。



在浅色主题和深色主题下的进度条

活动

如果你不知道进程将持续多久，用一个不确定的进度指标。有两种款式可供选择：一个长条和一个圆。根据空间选用合适的。

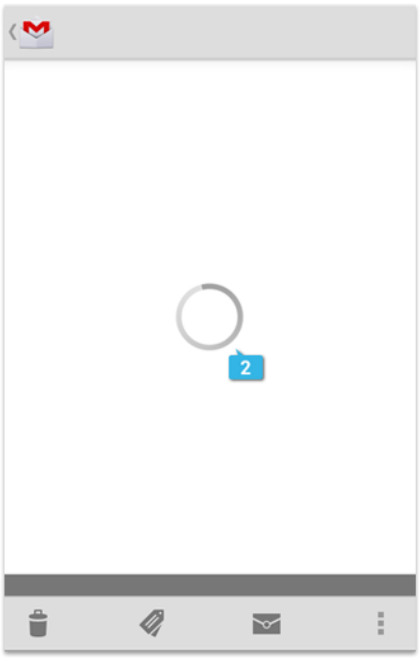


1. 不确定进度条（显示为深色主题）

一个不确定的进度条会用于程度的下载。因为应用市场还没链接到服务器，所以不能决定要多久才能下载完毕。

2. 进度圈（显示为浅色主题）

在 Gmailapp 正在加载邮件时会使用进度圈，因为它无法确定多久才能下载完该电子邮件。



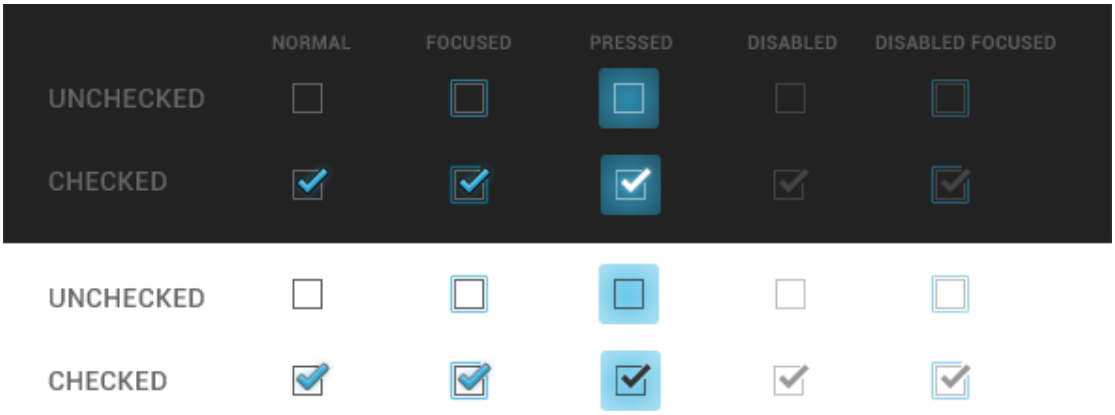
在每个界面上，你应该只使用一个进度指示，并应根据周围环境的大小适应。例如，最大的进度圈在空白的内容中显示良好，但在一个小对话框就不适用。

选择控件

有三种选择控件：复选框、单选按钮、开关。

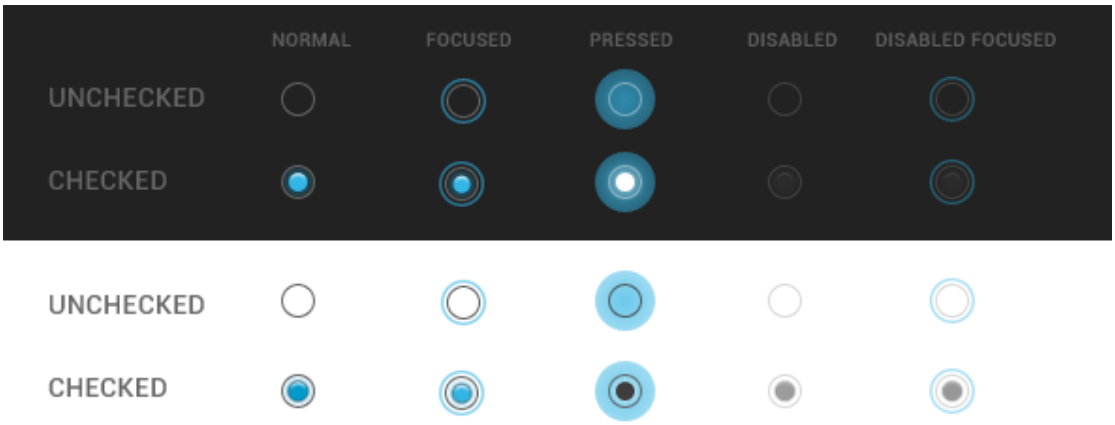
复选框

用户可以通过复选框在一组中选择多个选项。避免使用复选框进行开关操作。



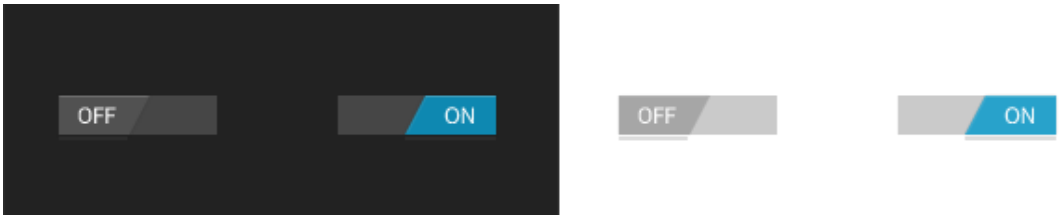
单选按钮

用户可以通过单选按钮从一组选项中选择一个。单选按钮适用于你需要用户看到所有选项的情况，如果它不需要看到所有选项，应该使用下拉菜单。



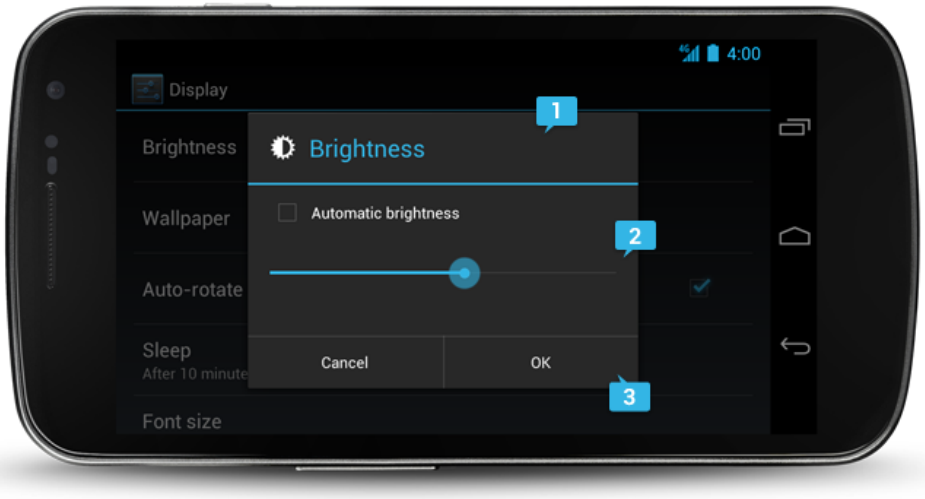
开关

开关可以从两个状态中选一。



对话框

在 app 需要询问用户做选择或更多信息任务才能进行下去时可以使用对话框。对话框的形式包括简单的选择确定、取消到复杂的要求用户调整设置或输入文本。



1. 标题区（可选）

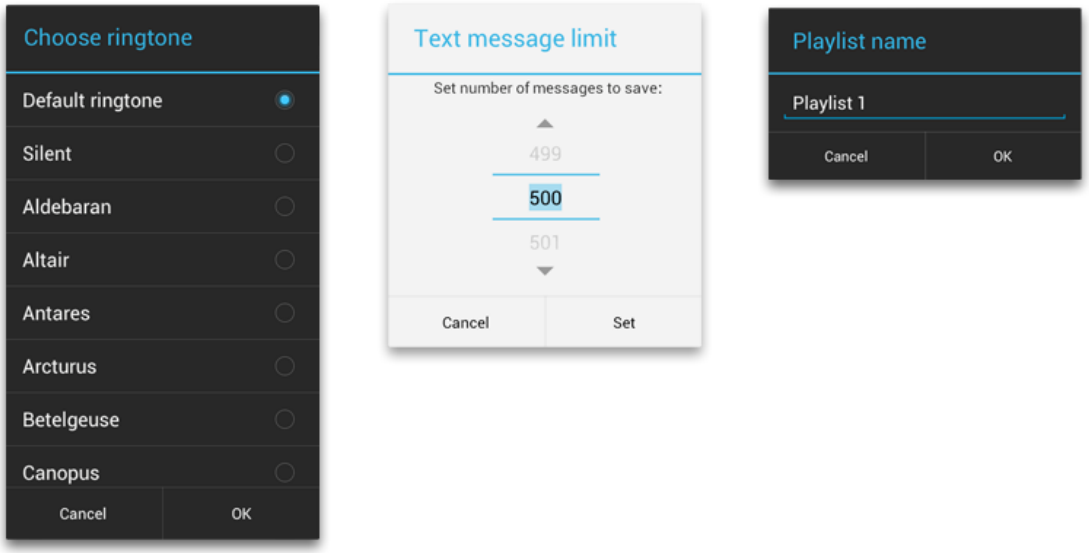
标题是关于这个对话框是什么。例如，它可以是一项设置的名称等。

2. 内容区域

对话框内容多种多样。对于设置对话框，它会可以帮助用户改变 app 属性或系统设置的元素包括滑块、文本输入、复选框、单选按钮。其他的还包括警报，内容可能是用户需做决定的背景资料介绍。

3. 操作按钮

操作按钮通常是指确定和取消，确定是首选或最可能选的操作。但如果选项包括关闭或等待这些不是确定或取消类所描述内容的操作时，这些按钮都应该使用主动动词。有一个规则是把否定的操作安排在左边的按钮，肯定的操作安排在右边。



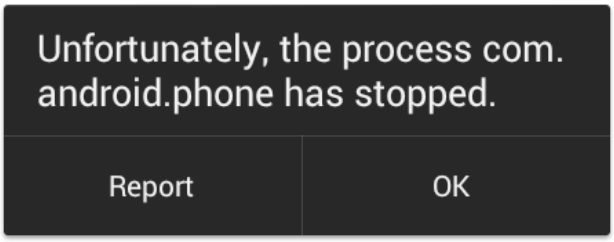
Android 典型对话框范例

警报

警报是通知获得用户批准或确认程序才能进行下去的对话框形式。

1. 没有标题的警报

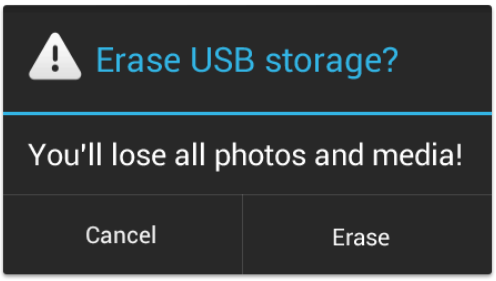
大部分警报不需要标题，通常情况下，决定不会有严重影响并可一两句话总结清楚。内容区域应该问一个问题（如“删除这个谈话吗？”），或是一个与操作按钮明显相关的明确陈述。



2. 带标题的警报

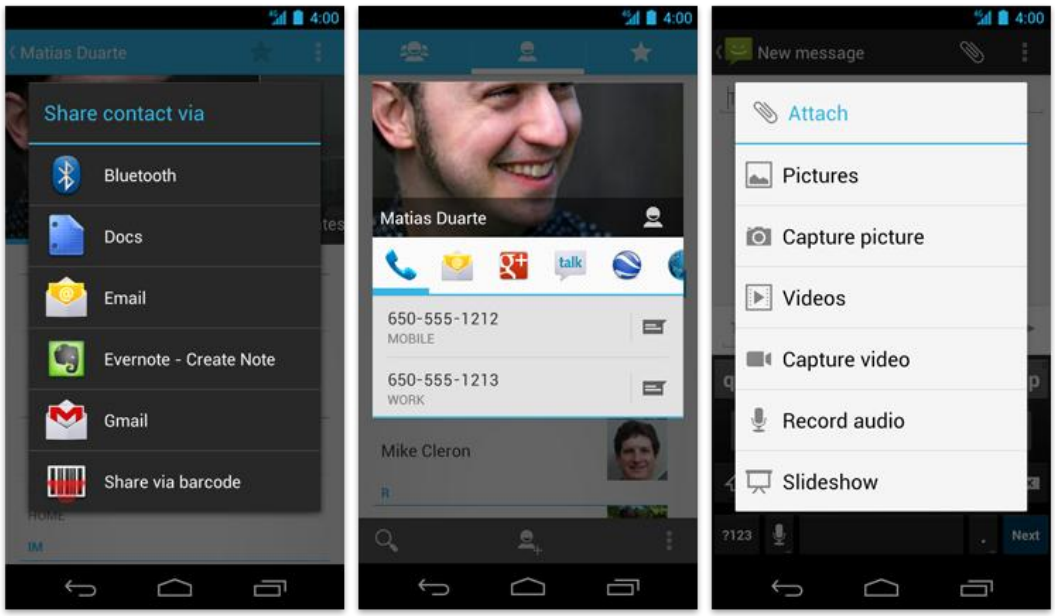
有节制地使用带标题的警报。只有在可能引致数据丢失、连接、收取额外费用等地高风险操作时才使用。标题需要一个明确的问题，在内容区需要一些附加的解释。

保持问题或陈述的简短：例如“清除 usb 的储存？”避免道歉。要做到即使用户只是随便扫过内容，只根据标题与操作按钮的文本也清楚知道该选哪个按钮。



弹出窗口

弹出窗口是对话框的轻量版，它只要求用户的一个选择。弹出窗口不需要确定或取消按钮。它只要求在众多选项中选取一个或点击弹出窗口以外的地方离开弹出窗口。



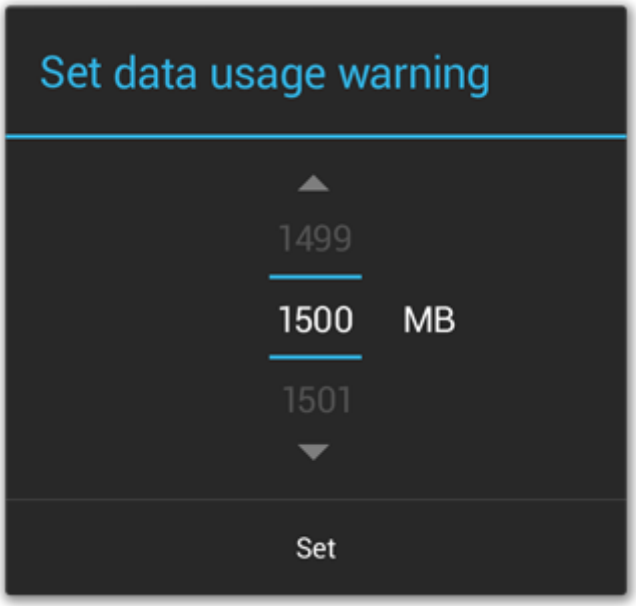
Toasts 信息提示条

信息提示条是一个操作以后的轻量反馈。例如，当你在写邮件的页面跳转到其他页面时，信息提示条弹出提示“邮件已存到草稿箱”，让你之后还可以继续编辑它。信息提示条在几秒后自动消失。



选择器

选择器提供了一个简单的方式从一定范围内选择一个值。操作可以触摸向上向下箭头按钮、通过键盘输入或向上向下滑动的手势选取。



空间的考虑
选择器可以内嵌在一个形式里，但由于它占的位置相对较大，最好把它放在一个对话框里。对于嵌入式的，可以考虑使用更为紧凑的空间如文本输入或下拉菜单。

日期和时间选择器
Android 提供了以下这些供使用的日期和时间选择器对话框。每个选择器可用于输入日期（月，日，年）或时间（小时，分钟，上午/下午）。在你的 app 中使用这些选择器可以保证用户的输入格式准确、有效。时间和日期的格式会根据使用的语言环境而自动调整。

