

Android Design 官方设计文档

翻译

译者：潘思霁（Pandroid）

友情提示：

- 1.Pandroid 的英语水平只有 CET4
- 2.部分翻译精简，部分专用词汇保持原文
- 3.感谢谷歌翻译和有道词典的帮助

via：[兔斯霽の LCL 之海](#)

目录

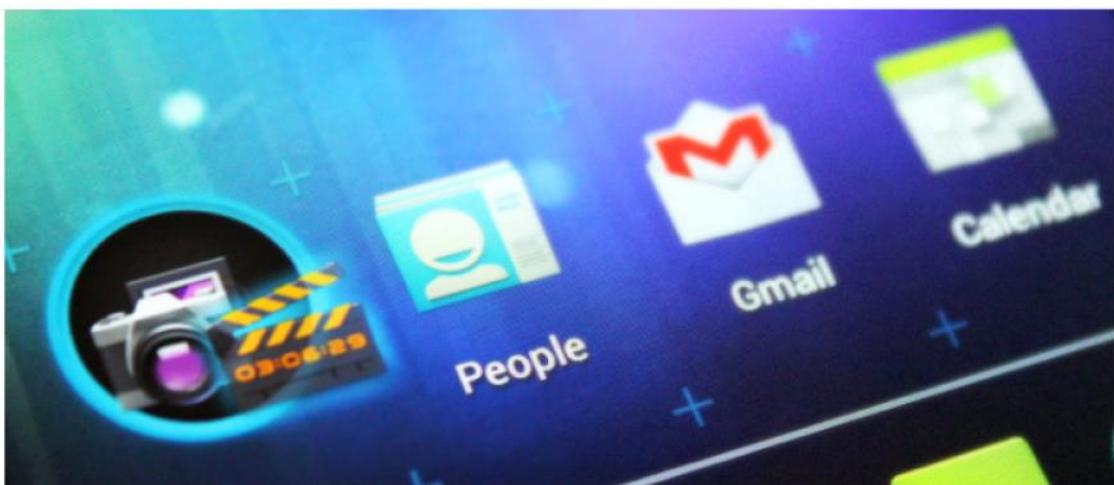
GET STARTED-让我们开始吧-----	3
Creative Vision-创造性的视觉体验-----	4
Design Principles-设计原则-----	5
UI Overview-用户界面概要-----	10
Style-我的风格-----	14
Devices and Displays-设备与显示-----	15
Themes-主题-----	16
Touch Feedback-触摸反馈-----	17
Metrics and Grids-度量与网格-----	19
Typography-排版-----	21
Color-色彩-----	22
Iconography-图标研究-----	23
Writing Style-文案书写风格-----	27
Patterns-模式-----	29
New in Android4.0-Android4.0 新特性-----	30
Gestures-手势-----	31
App Structure-应用程序结构-----	32
Navigation-导航-----	38
Action Bar-----	47
Multi-pane layouts-多面板布局-----	54
Swipe Views-滑动视图-----	58
Selection-选择操作-----	60
Notifications-通知系统-----	62
Compatibility-兼容性-----	66
Pure Android-纯粹 Android-----	68
Building Blocks-系统控件-----	71
Tabs-标签-----	72
Lists-列表-----	73
Grid Lists-网格列表-----	74
Scrolling-滚动-----	75
Spinners-下拉列表-----	76
Buttons-按钮-----	77
Text Fields-文本域-----	78
Seek Bars-----	79
Progress & Activity-进度和活动-----	80
Switches-开关-----	82
Dialogs-对话框-----	83
Pickers-选择器-----	86
Downloads-资源下载-----	87
Contact-联系作者-----	88

GET STARTED-让我们开始吧



欢迎来到 Android Design ,
在这里你可以学习如何设计出优秀的 Android 应用程序。

Creative Vision-创造性的视觉体验



冰激凌三明治 (Android 4.0) 是 Android 设计上的一个重要的里程碑。我们几乎考虑到系统每一个像素，采用新的设计方法，将蜂巢系统平板的特性融入到其他移动设备中。从最基础的开始，我们引入了为高分辨率显示而设计的字体——“Roboto”。其他重要的改变包括 “Action bars” 框架和无物理按钮设计。

当设计 Android 应用程序时，我们主要关注三个目标：

让我着迷

美丽是外在的表现。从各种标准来看，Android 应用程序应该是井然有序的，而且要符合人们的审美观。过渡动画快速明了；布局排版充满新鲜感且意义非凡；应用程序的图标包含艺术性。就像一个制作精良的工具，你的应用程序应该努力去整合美丽、简洁和目标，从而创造出梦幻般的体验——简单易用且功能强大。

简化我的生活

Android 应用程序让我们的生活更加简洁，使事物更加通俗易懂。当人们第一次使用你的应用程序时，他们应该能直观地掌握最重要的功能特性。设计工作不会就此停止，Android 应用程序需要不断地移除那些“琐碎杂事”，就像文件管理和同步一样。简单任务不需要复杂的步骤，而复杂任务的步骤需要充分适应人们的行为和想法。不同年龄和不同文化的人都能感到一切事物都在掌控之中，而不是被过多的选项和无关的打扰所淹没。

让我感到惊喜

让一个应用程序做到易用还完全不够。Android 应用程序允许人们尝试新的东西，且可以让人通过一些别出心裁的方法去使用应用程序。Android 能让人们通过多任务、通知消息和分享系统将各类应用程序整合到同一工作流程中。与此同时，你的应用程序应该是个性化的，能让人透彻地触及到优秀的技术。

Design Principles-设计原则

这些设计原则是 Android 用户体验团队提出来的，旨在让应用程序获得用户的喜爱。为了避免偏离设计目标，请您在设计应用程序时参考这些设计原则。

让我着迷

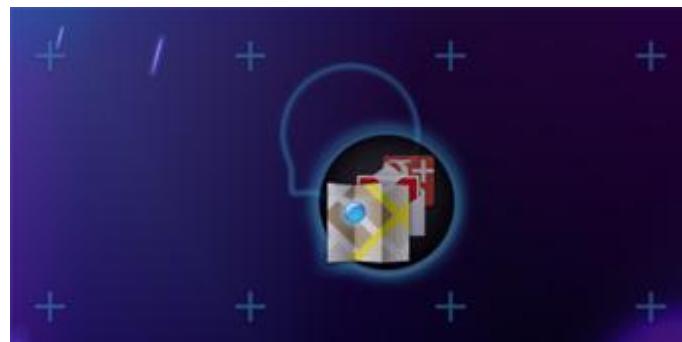
用一种惊奇的方式让我高兴

一个漂亮优雅的界面，一个精心设置的动画效果，一个恰到好处的音效，都是一种愉快的体验。精致的效果能让你的应用程序变得更加高效且让人觉得功能强大。



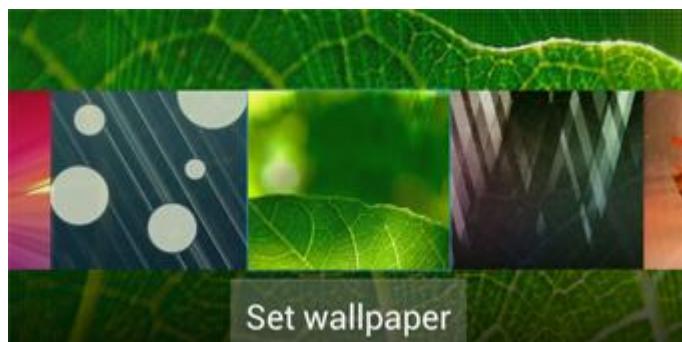
真实的物体比按钮和菜单更有趣

让人们在你的应用程序中能准确地触摸操作各类物体控件。这样能够帮助用户在执行任务时能快速理解意图，降低用户在认知事物时所需要的精力，让用户心情愉悦。



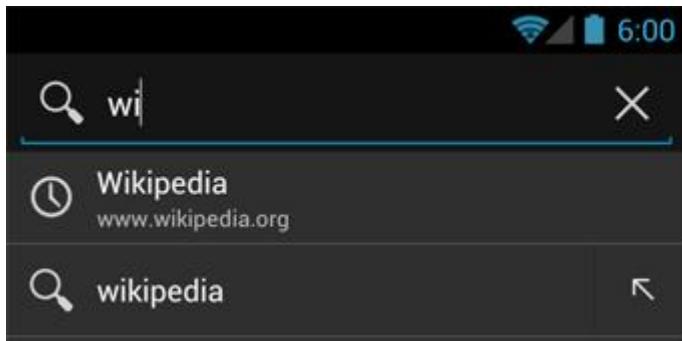
让它成为我自己的

人们喜欢在手机上添加上自己的特色，因为这样能让他们觉得亲切，并且有一种掌控感。所以你的程序需要提供明智的、优雅的默认选项以及可选的定制方案。



让它懂我

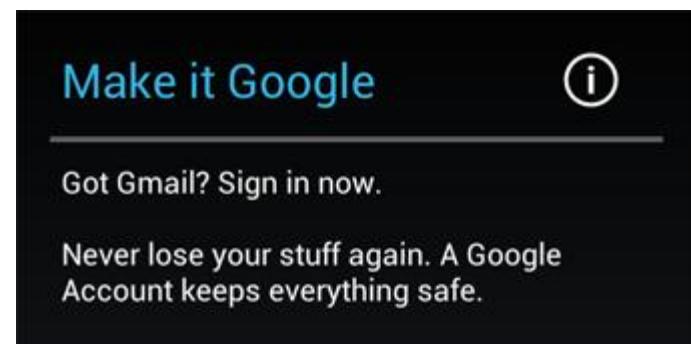
随时学习人们的偏好设置。将之前选择过的选项置于最前，而不是一遍一遍重复相同的选项。



简化我的生活

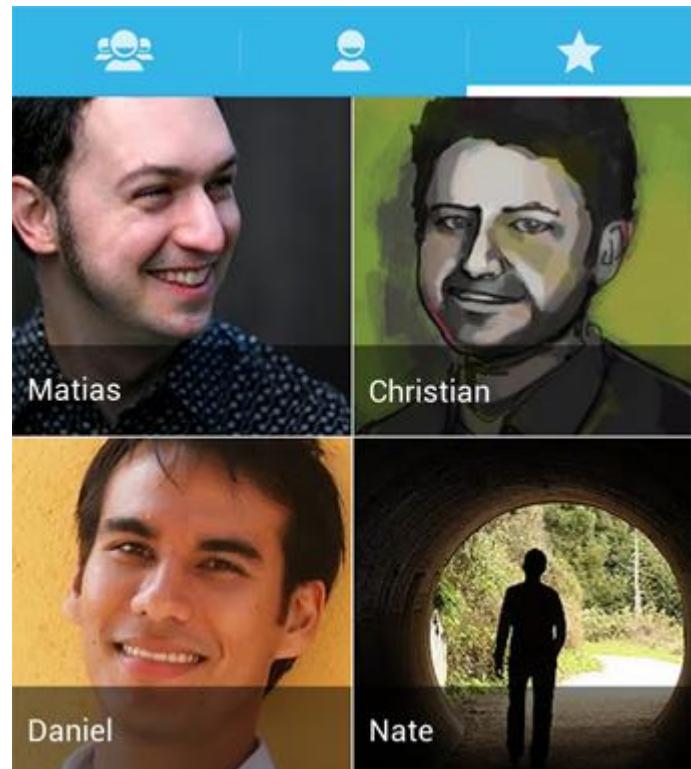
简洁明了

请使用简单的语句，因为人们往往会觉得跳过比较长的句子。

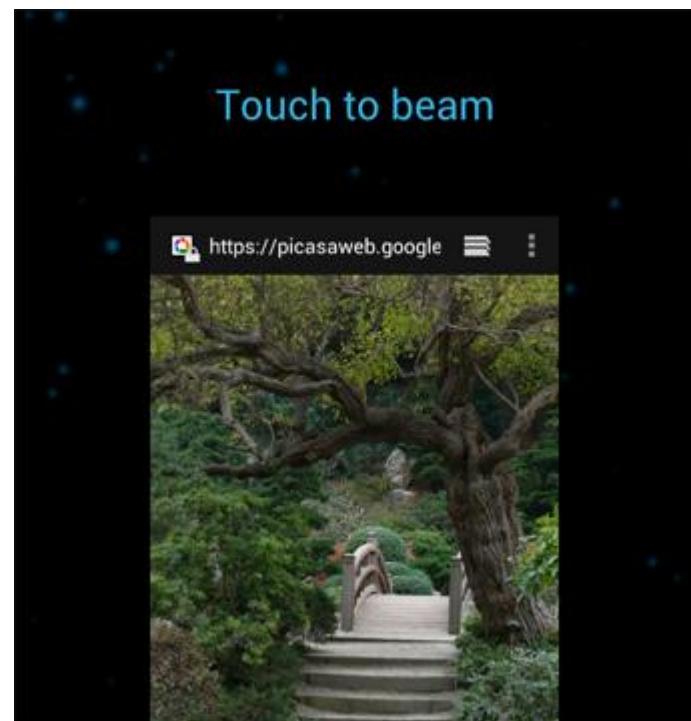


图片比文字更容易表达内容

请使用图片来表达你的想法。图片更能吸引人的注意，而且比文字更有效率。



为我做决定，但是最后我说了算
准确的预测和行动远比重复询问更加好，太多的选择和决定会让人感到懊恼。万一预测错误，可以取消操作。



Mark important

每

Report spam

Settings

Help

Send feedback

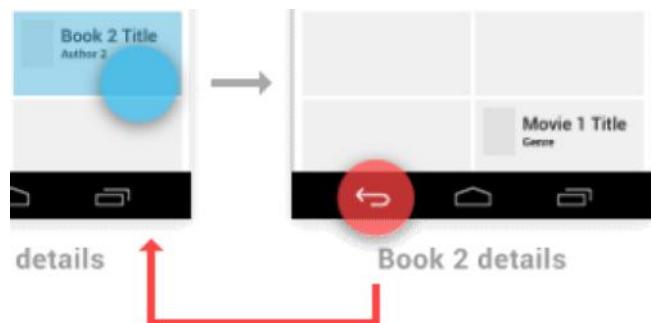


仅在我需要的时候显示

过多的选项同时出现会让人们“不堪重负”。所以，请将任务和信息拆分成简短易懂的小块，并且隐藏此场景中不必要的选项。

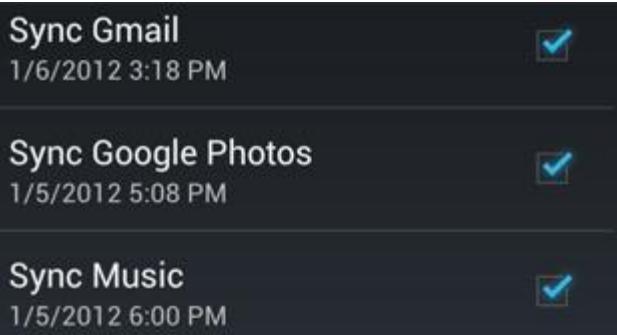
我应该知道我在哪儿

让人们知道他们现在所处的场景。在你的应用程序中显著地表明当前位置，并且使用过渡动画来表明每个界面的联系。最后一定要在任务进程中发送反馈。



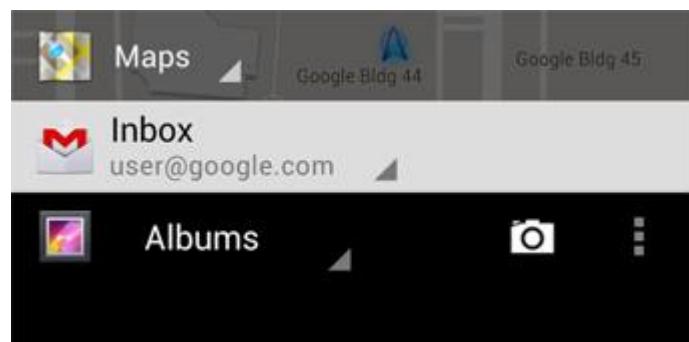
永远不要丢失我的东西

将人们创建的文件保存，并且让他们能在任何地方获取到这些文件。设定参数、个人特色和文件等能够跨设备获取，例如通过手机、平板、电脑。除此之外，还需要保持数据的及时更新和同步。



如果它看起来一样，那么它的动作也要一致

要帮助人们分辨功能的不同，那么就需要用特征明显、形象具体的标记来加以区分。要避免那些看起来相似但是动作却完全不同的设计。



只有重要的事情才需要通知我

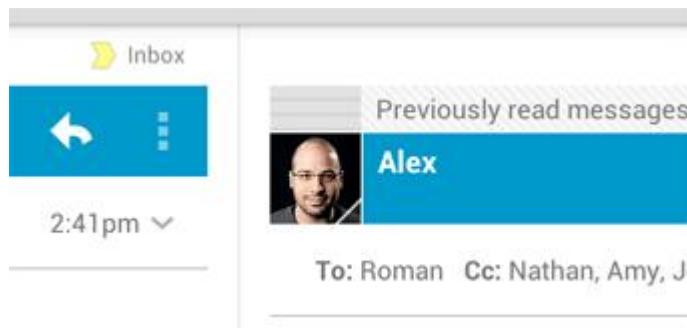
就像一个优秀的私人助理一般，帮人们阻拦不重要的小事情。人们想一直保持专注，除了紧急的、有时限的通知以外，其他任何打扰性的通知消息都会让人分心、沮丧。



让我感到惊喜

在任何时候给我技巧提示

当人们自己完成一些事情的时候，当时他们的感觉都会非常棒。采用视觉模式，并且依靠用户使用其他 Android 应用程序而留下的记忆，使你的应用程序简单易学。例如，滑动的手势就是一种非常好的导航方式。



这不是我的错

提示用户改正错误的语句文案要礼貌一些。在人们使用你的应用程序时，他们总是觉得自己是聪明的。当某些事情出错的时候，给出了明确的恢复指导后却又提示详细的 техnical details，这显然是不明智的。如果你不能修好它，那么最好别让这些细节提示显示在场景中。



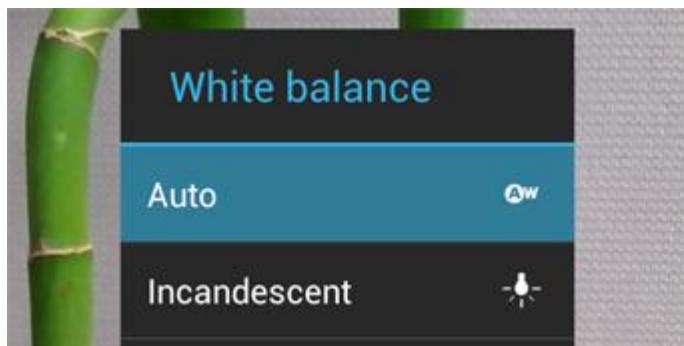
给予反馈鼓励

将复杂的任务拆分为小的步骤，这样任务就能比较容易完成。对每一个动作都要做出反馈，即使它微不足道。



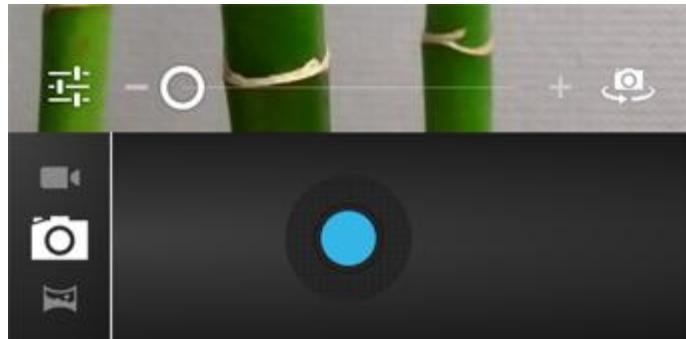
出乎意料的“大”东西

让初学者学用户会做一些他们认为自己不可能做到的事情，这样就能让他们有一种专家用户的感觉，使他们心情愉悦。例如，集成多种照片效果的快捷键，可以快速地让业余水平的照片变得让人眼前一亮。



突出重要的东西

每个控件动作的重要程度都是不一样的。你要仔细考虑把你认为重要的动作放置在最显眼的位置，以便用户能快速发现并且使用它们。就像相机程序的快门按钮，又或者是音乐播放程序中的暂停按钮。



UI Overview-用户界面概要

Android 系统在你创建应用程序时提供相应的框架。主屏幕的使用体验、程序的导航设计和通知系统都是比较重要的几个方面。

在保持 Android 统一的使用习惯和出色的用户体验中，你的应用程序会起到至关重要的作用。在本章的最后一节，我们会介绍如何一些基础知识以便帮助你实现这一目标。

下面我们快速地了解一下 Android 用户界面中比较重要的几个方面。

Home、All Apps、and Recents

Home screen

Home screen 是一个自由的空间，你可以使用图标、文件夹和桌面插件进行定制。Android 系统包含 5 个 Home screen，通过左右滑动来进行切换。

底部的 Favorites Tray 可以放置你最常用的程序和文件夹，他们不会随着主屏幕的切换而改变。

点击 Favorites Tray 中间的 All Apps 按钮就进入 All Apps screen，在这里你可以查看你所有的应用程序和桌面插件。



All Apps screen

All Apps screen 可以让你查看所有安装在设备中的应用程序和桌面插件。

用户可以拖拽一个应用程序或者桌面插件到任一主屏幕，当然也需要保证那个主屏幕有足够的空间。

Recents screen

Recents 可以帮助用户快速地在最近使用的应用程序间进行切换，这是一种简单明了的多任务导航方式。

Recents 按钮在 Navigation Bar 的最右侧，点击后就会显示用户近期使用的应用程序。Recents 中的程序按照用户使用先后顺序进行排序，最近使用的排在最下，然后依次向上。

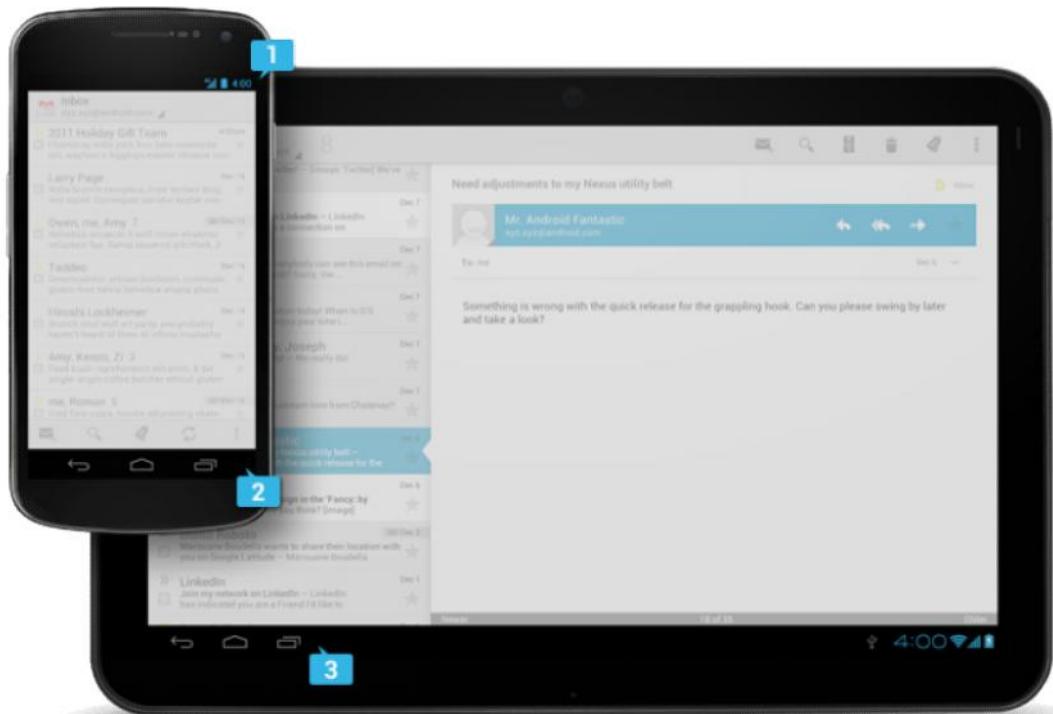
在切换应用程序是可以点击 Recents 按钮，左右滑动可以移除该应用程序。



System Bars

System bars（包含 Status bar、Navigation bar 和 Combined bar）用于显示通知信息、设备通信状态和导航。System bars 往往是和你的应用程序一同显示的。当应用程序需要显示一些特殊内容时，System bars 就是自动隐藏，比如显示图片或者视频。这样的话就可以让用户享受全屏的体验，且不会打扰用户。

下面就来介绍下 System bars 的三个部分：Status bar、Navigation bar 和 Combined bar



(如上图所示)

1.Status Bar

通知信息会显示在 Status bar 的左侧，而设备信息比如电量、网络信号、时间等会显示在右侧。从 Status bar 向下滑动就可以显示通知的详情。

2.Navigation Bar

目前，在少数搭载 Android4.0 的手机上，Navigation bar 已经取代了传统物理导航按键。它已经整合了设备的导航按钮 Back、Home 以及 Recents，而且为了支持 Android2.3 之前的程序，会将 menu 按钮显示在 Navigation bar 的最右侧。

3.Combined Bar

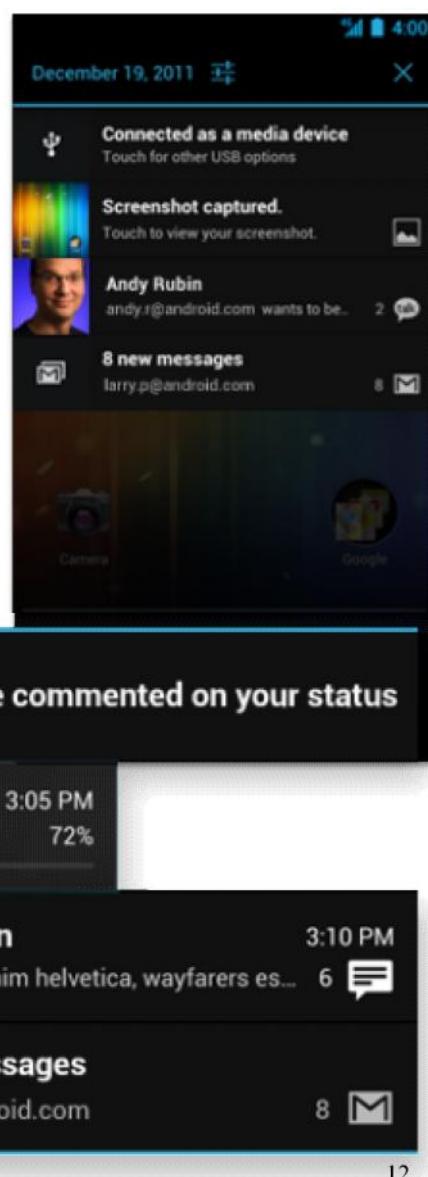
在平板电脑上，Status bar 和 Navigation bar 是一同显示在屏幕底部的一条上的。(PS：Android3.X。其实 Android4.X 已经整合了平板和手机，Combined bar 将不复存在)

通知

通知是一段简短的消息，可以帮助用户快速得从 Status bar 获得消息内容。通知可以显示程序更新、事项提醒等一些重要的信息，但是并不保证用户不会被非重要的通知打扰。从 Status bar 向下滑动即可打开通知中心，点击通知就可进入对应的应用程序。[更多关于通知](#)

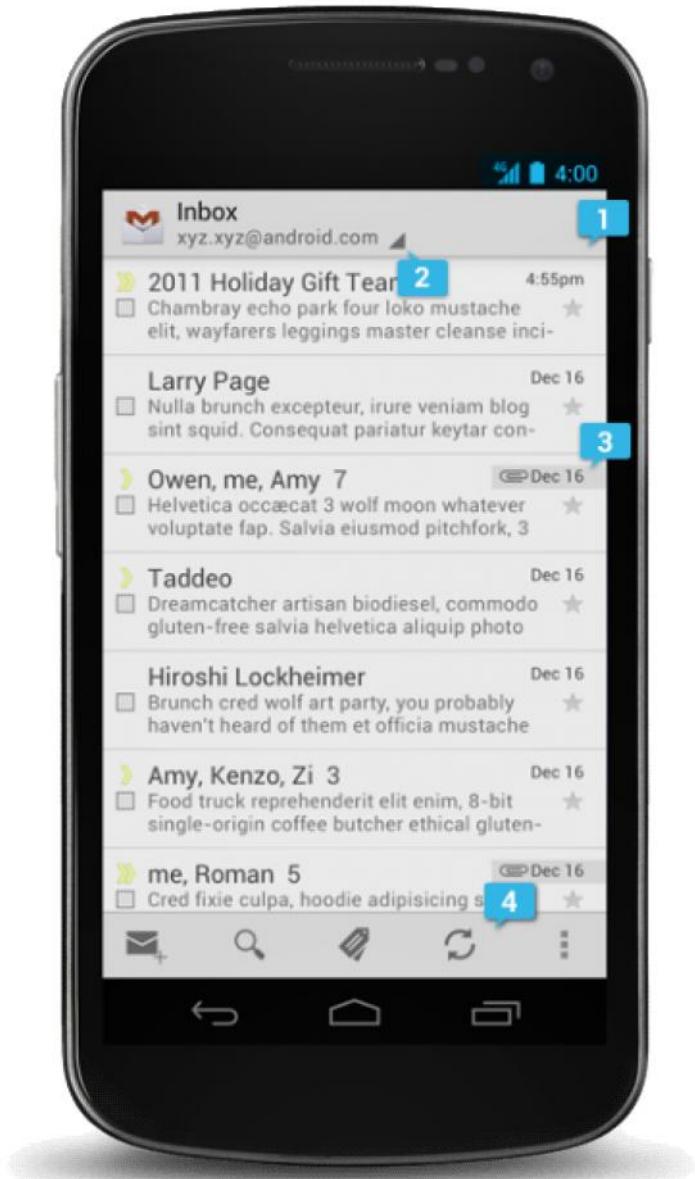
大多数的通知都有一行标题和一行消息，我们也是建议这样设计的。如果需要的话，你可以设计为三行显示。通知时间的显示是可选的，根据自己的需求进行选择。

向左或者向右滑动一条通知就进行删除操作。



典型的 Android 应用程序界面设计

一个典型的 Android 应用程序由 Action bars 和 Content area 组成。



1. Main Action Bar

Main Action bar 是你程序的控制中心，它包含了最基础的控件用于导航、切换视图以及显示最重要的动作按钮。

[更多关于 Action bar](#)

2. View Control

View Control 可以让用户在应用程序的不同界面（视图）之间进行切换。这些界面（视图）主要包括应用程序中不同的数据排布以及不同的功能。

3. Content Area

用来显示应用程序的内容。

4. Split Action Bar

Split Action bar 用于显示常用的动作按钮，它通常位于 Main Action bar 的下方或者是屏幕的底部。以图中为例，我们将所有重要的且不适合出现在 Main Action bar 上的动作按钮全部转移到了位于屏幕底部 Split Action bar。

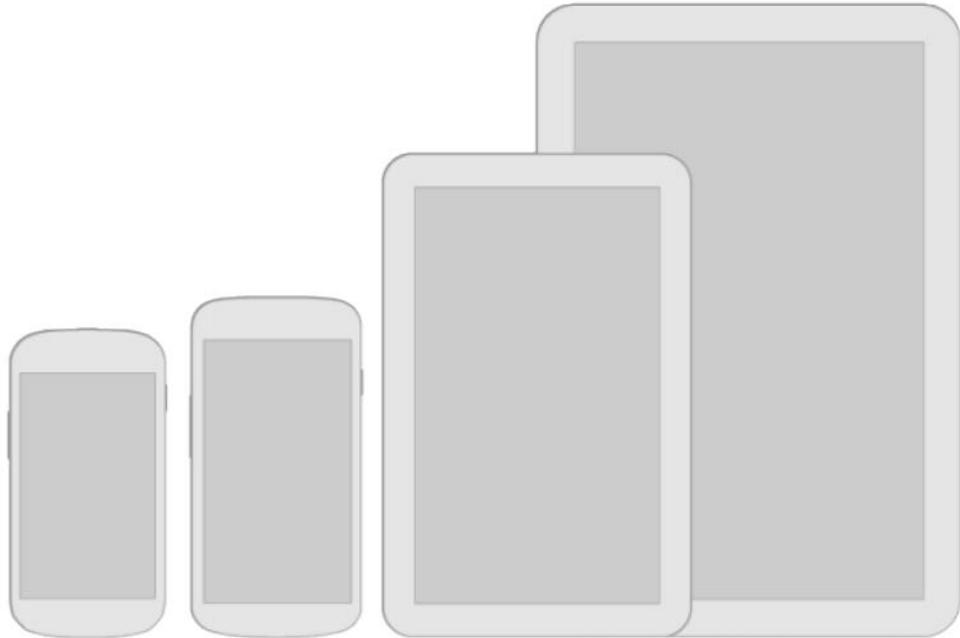
Style-我的风格



设计一个视觉出众的的应用程序，
让它在各种设备上都是那么引人注目。

Devices and Displays-设备与显示

包括手机、平板等等在内，数百万的设备上运行着 Android 系统，这些设备有着不同的屏幕尺寸和技术参数，所以我们的应用程序需要多分辨率支持。利用 Android SDK 灵活的布局系统，可以让你的应用程序适配不同屏幕大小的平板和手机。



灵活应用

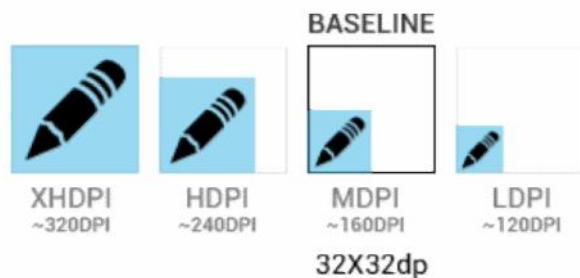
伸展或者压缩你的布局文件以便于适应各种高度和宽度。

优化布局

在大尺度的设备上，可以利用大尺寸屏幕将多视图、内容和导航设计在一个混合的界面（视图）上。

支持多种设备

为不同 DPI 的屏幕提供多种布局资源文件，以保证应用程序在各种设备上完美显示。



设计策略

那么我们应该从何开始为多分辨率显示设计呢？一种方法是设定一个基础的标准（比如中等尺寸，MDPI），然后根据屏幕比例缩小放大布局资源文件。另一种方法是以适应最大屏幕尺寸为标准，然后缩小显示比例并调整用户界面，以此来适配小屏幕设备。

更多详细信息，请参考[多分辨率显示支持](#)

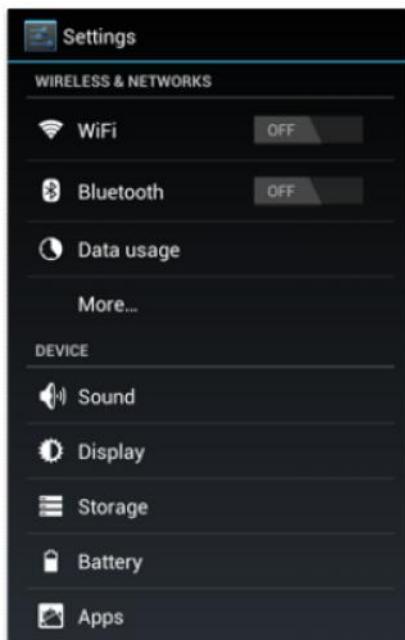
Themes-主题

主题的设置是为了保证应用程序和 activity 能够有统一的风格。而风格则指定了用户界面的基本视觉元素，例如 color、height、padding 和 font size。为了使各应用程序在视觉上更加统一，Android 提供了三种系统主题，你可以选择以下三种主题来开发程序（需要支持 Android4.0 以上）：

Holo Light

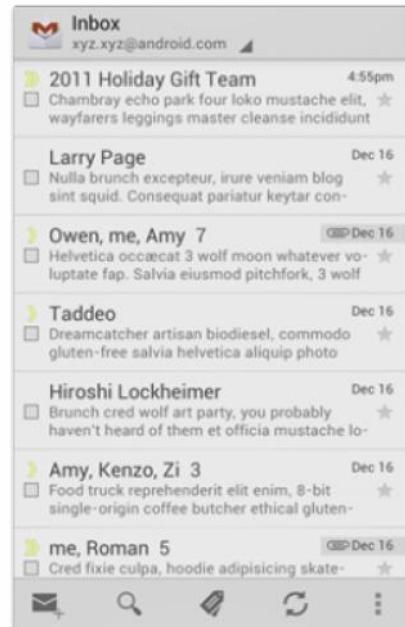
Holo Dark

Holo Light with dark action bars

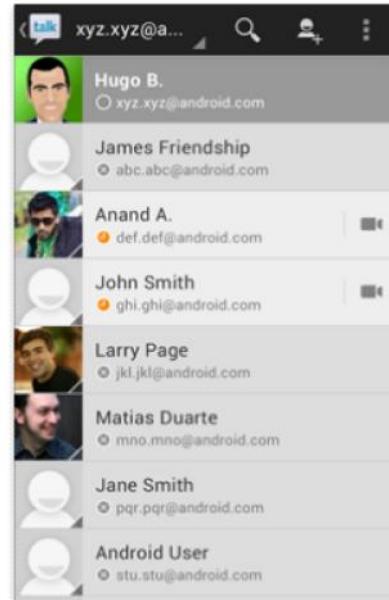


系统设置：Holo Dark

当你设计应用程序时请采用这些主题，他们可以使你的程序与 Android 系统保持一贯的视觉风格。



Gmail : Holo Light



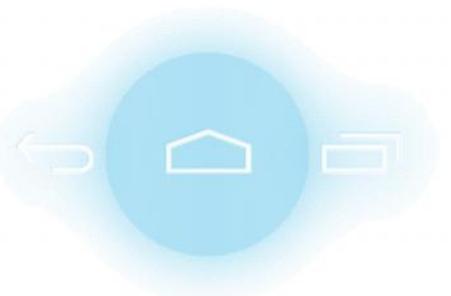
Gtalk : Holo Light with dark action bars

选取你觉得合适的主题，然后为自己的程序精心设计吧。如果你希望你的应用程序看起来与众不同，那么可以只采用其中一个系统主题作为你定制界面的开始点，这也不失为一种好办法。系统主题在顶部提供了一个固定的区域，你可以自由定制你想要的视觉风格。

Touch Feedback-触摸反馈

使用颜色和亮光来对触摸进行反馈，以强调该触摸手势会产生的结果，并且表明哪些 action 按钮是启用和禁用的。

当用户在你的应用程序中触摸可以可操作的区域时，请提供一个视觉上的反馈。这可以帮助用户理解哪些对象是可操作的，同时表明你的程序在“聆听”用户的操作。



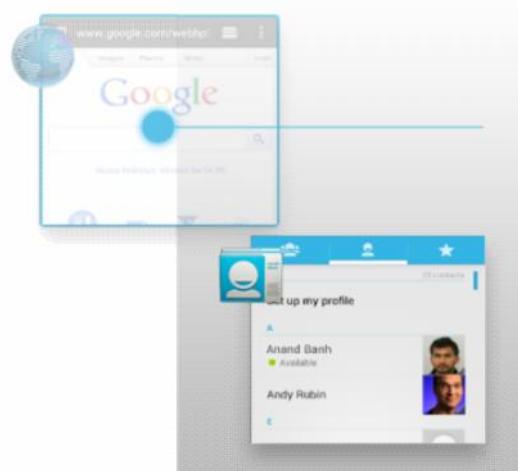
状态

大多数的 Android UI 元素都带有触摸反馈，包括控件的不同状态也表明它是否会产生反馈。

Normal	REMAINS STATIC
Pressed	ILLUMINATES WITH COLOR
Focused	DRAWS 50% OF THE PRESSED VALUE (OPTIONAL 2DP BORDER FILL AT PRESSED VALUE)
Disabled	DRAWS 30% OF THE NORMAL STATE
Disabled & focused	DRAWS 30% OF THE FOCUSED STATE

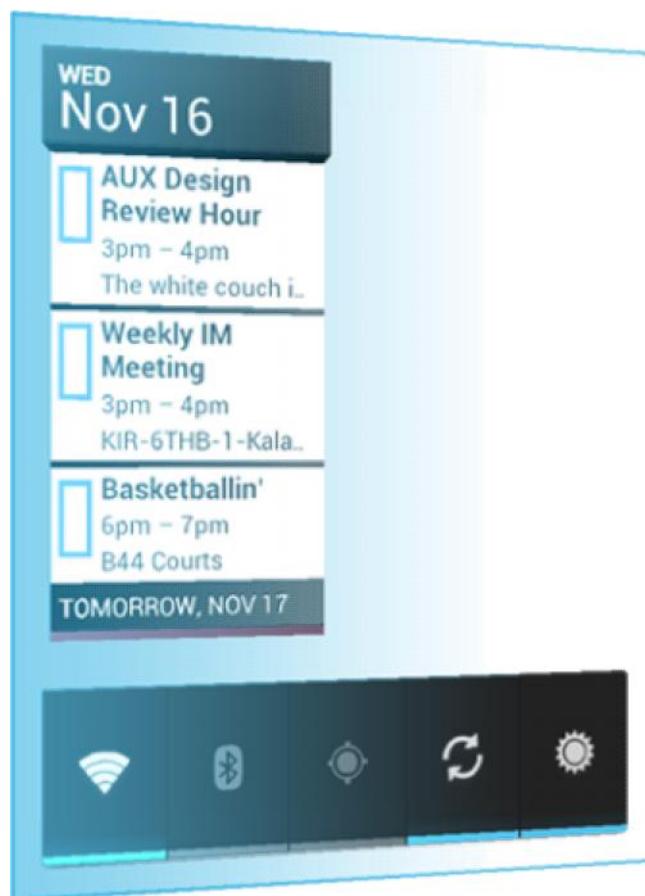
沟通

当你的应用程序中的控件对用户的手势做出反应时，你需要帮助用户理解该手势会产生什么样的结果。比如在 Recents 中，当你左右滑动程序的缩略图时，它开始变透明。这就告诉用户这里的滑动手势会移出一个任务。



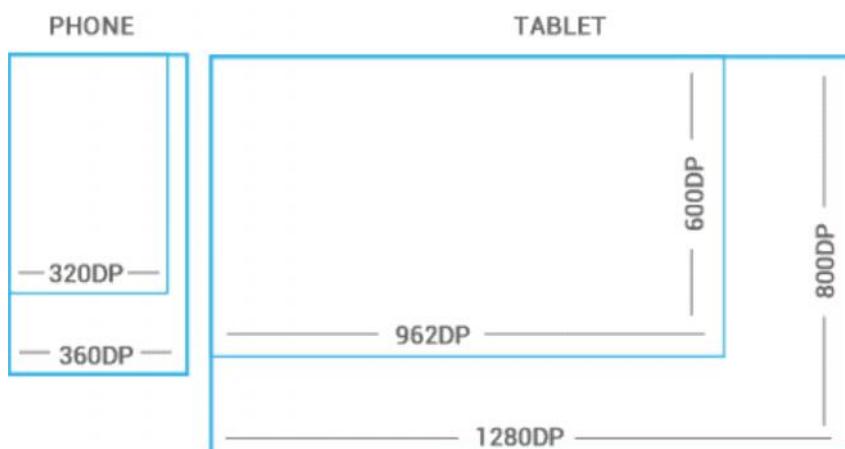
边界

当用户的手势在可滚动区达到操作上限（下限）边界时，请用特殊的视觉效果来提示用户。比如，当用户试图滚动 Home screen 第一屏时，屏幕的内容会产生向右倾斜的动画效果，这就表明无法进一步地进行滚动操作。许多 Android 可滚动的 UI 控件（比如 lists 和 grid lists）都支持这种边界反馈，在你定制应用程序界面的时候，请务必把边界反馈铭记于心。



Metrics and Grids-度量与网格

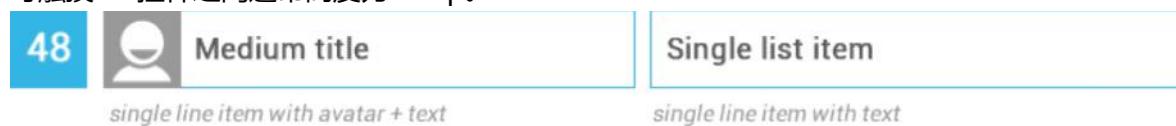
很多设备不仅在屏幕上大小上不同，同时在屏幕密度上 (DPI) 也有很大不同。为了将设计工作更加简单化，把每个设备都想象成一个特定的尺寸桶和密度桶。尺寸桶都是手机（小于 600dp）和平板电脑（大于等于 600dp），密度桶是多种屏幕密度：LDPI (120dpi), MDPI (160dpi), HDPI (240dpi), XHDPI (320dpi)。为了优化你的应用程序，你需要为不同大小的尺寸桶设计不同的布局文件，为不同的密度桶提供多种位图图像。



更多详细信息，请参考 [Screen Sizes and Densities Device Dashboard.](#)

48dp 的“节拍”

可触摸 UI 控件之间通常高度为 48dp。

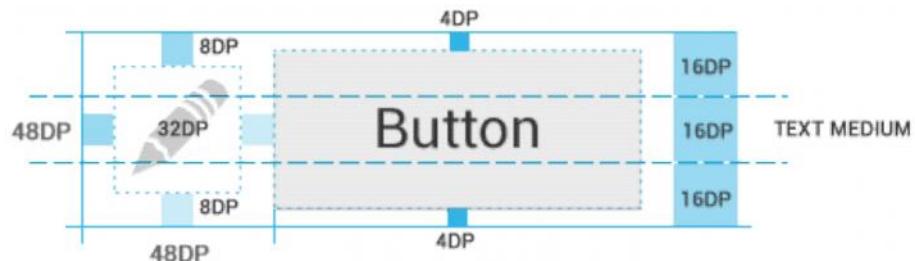


为什么是 48dp？

一般来说，48dp 转化成物理尺寸约为 9 毫米（稍微有些出入）。对于触屏对象来说这个尺寸刚刚合适，与建议的尺寸（7-10 毫米）非常接近。这样用户就能用手指准确地点击目标。

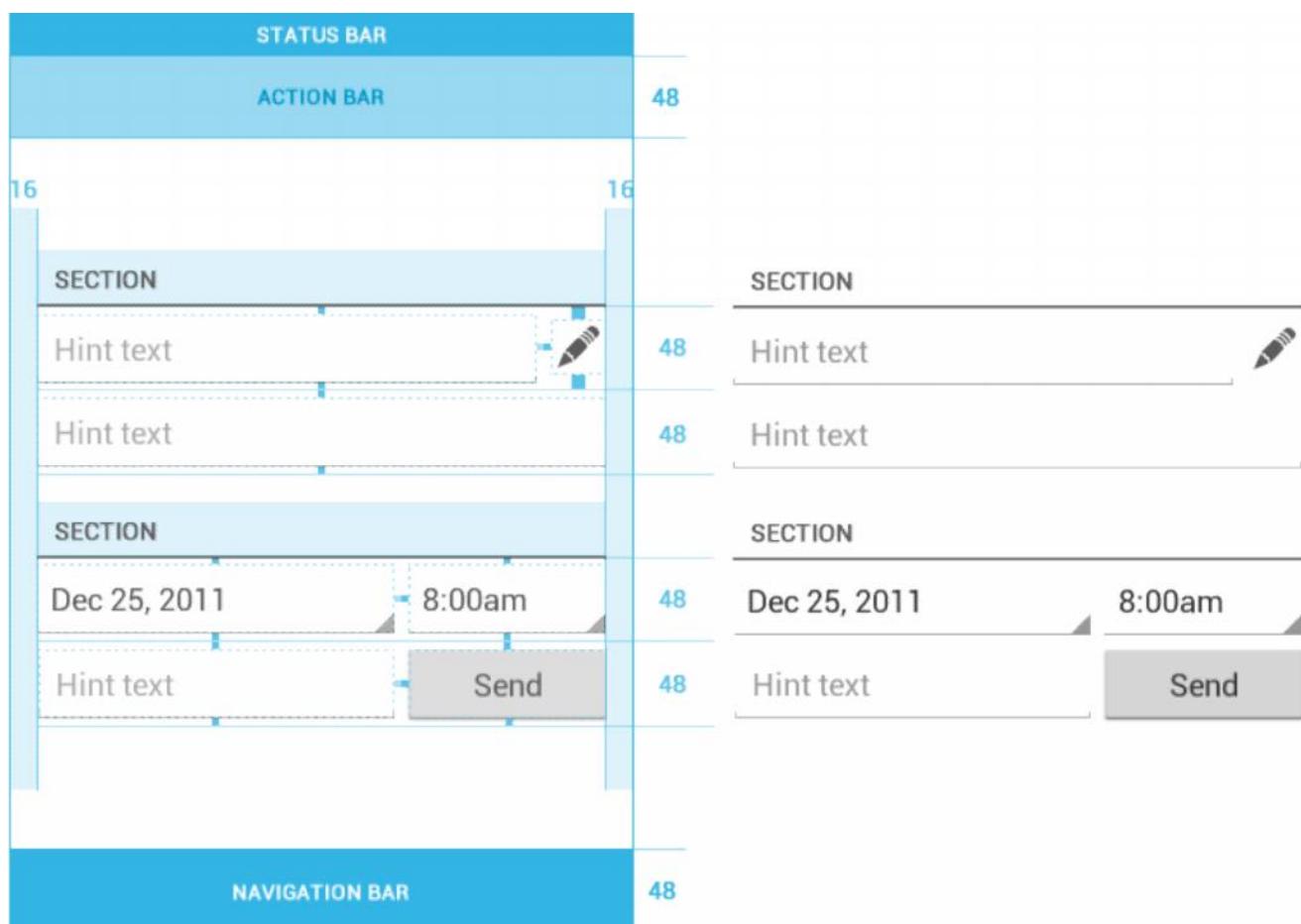
如果你设计的 UI 元素高宽尺寸至少在 48dp 以上，我们可以保证：

1. 你的控件大小用于不会小于 7 毫米的建议尺寸。
2. 你在 UI 信息密度和 UI 元素之间做出了一个出色的折中设计。



注意间隙，每个 UI 元素之间的距离是 8dp。

下面是一个例子



Typography-排版



Tips:

[Download Roboto](#)

[Specimen Book](#)

默认颜色

Android UI 使用如下默认颜色风格：
textColorPrimary 和 textColorSecondary。
Light 主题使用 textColorPrimaryInverse 和
textColorSecondaryInverse。该框架还支持 UI
控件内部文本颜色样式随触摸反馈而变化。

Text Color Primary Dark
Text Color Secondary Dark

Text Color Primary Light
Text Color Secondary Light

排版规模

在类型大小的对比中创建有序的、易懂的布局，
你还有很长的路要走。然而，在同一 UI 中不同的
尺寸大小可以被混合在一起。Android 框架使用
如图中尺寸有限的类型：

Text Size Micro	12sp
Text Size Small	14sp
Text Size Medium	18sp
Text Size Large	22sp

在 Setting 中，用户可以选择全系统适应的文本尺寸。为了支持这些辅助功能，尽可能选择与规模无关的像素(sp)。(在布局中推荐使用 sp 尺寸单位，这样可以更好地适应多分辨率)

Color-颜色

使用颜色的重点。为你的标牌选择合适的颜色，保证各视觉部分有良好的对比度。需要注意的是，色盲的用户可能无法分辨红色和绿色。



调色板

蓝色是 Android 系统的基准色，每一种颜色都有对应的更深的颜色，当它需要是可以作为补充。

[下载颜色样板](#)



Iconography-图标研究



图标是一个图形，它占用了屏幕上一小块部分，快速、直观地表示出一个动作、状态、或者是应用程序。

启动器

启动的图标，是你的应用程序在 Home 和 All Apps screen 上的可视化表现。由于用户可以更换桌面墙纸，所以请保证启动图标在任何背景中都清晰可见、容易识别。



尺寸&比例

启动图标在移动设备上必须为 48x48dp；在 Play 商店里必须为 512x512pixels

大小

全部尺寸 48x48dp

风格

使用一个独特的剪影。从前面看上去有立体感，然后有轻微从上往下看的角度。这样用户就能感觉到一些图标的深度

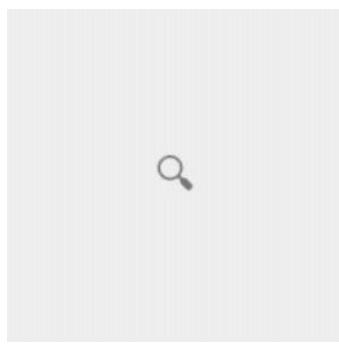


Action Bar

Action Bar 图标是图形按钮，用于代表应用程序中比较重要的一些 actions 操作。每个人都应该使用一个简单的象征性图形来表现一个概念，并且让大多数人都一目了然。

预先定义的字形应该用于某些常见的 actions 操作，例如“刷新”和“分享”。下面的下载链接提供了一个压缩包，里面提供的图标适用于各种规格屏幕密度，并且可用于 Holo Light 和 Holo Dark 主题。压缩包内还提供了一些无样式的图标，你可以修改他们以便匹配你自定义的主题。你可以使用 Adobe® Illustrator® 来修改源文件。

[下载 Action Bar 图标压缩包](#)



尺寸&比例

Action Bar 上的按钮为 32x32dp。



主要区域&大小

全部尺寸 32x32dp
图形区域 24x24dp



风格

象形的、平面的，由流畅的线条和尖锐的形状构成，但不能够写实。如果图像是瘦长的，那么就向左（右）旋转 45 度来填补空缺。

色彩

颜色:#333333

可用:60%不透明

不可用:30 不透明



颜色:#FFFFFF

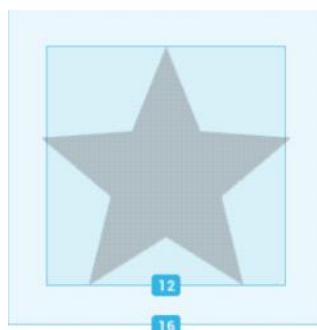
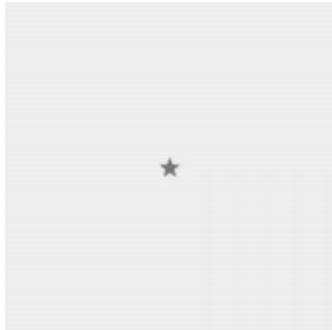
可用:80%不透明

不可用:30 不透明



小的 Contextual 图标

在应用程序的内部，使用小的图标来表现 action 操作或者为特定控件表明状态。比如在 Gmail 中，每一条消息都有一个可以表示重要程度的星状图标。



尺寸和比例

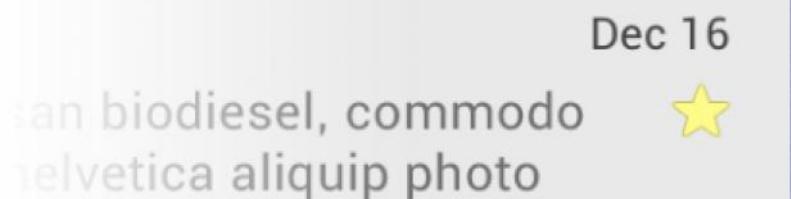
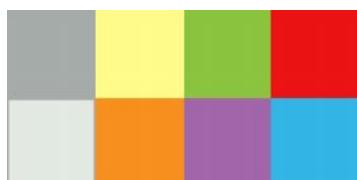
小的图标为 16x16dp

主要区域&大小

全部尺寸 16x16dp
图形区域 12x12dp

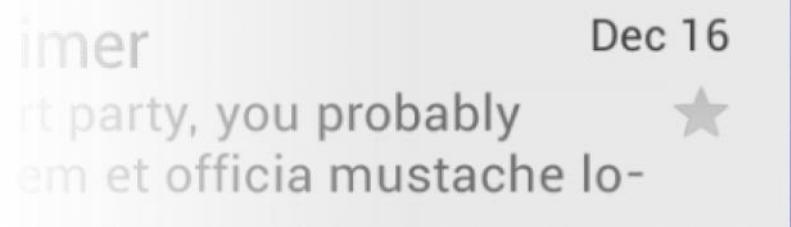
风格

中性色调、平面的、简洁的。使用一个简单的视觉图形，以便用户可以简单快速地识别和理解它的作用。



色彩

谨慎地、有目的地使用非中性色。
例如，Gmail 使用黄色的星形图案来表示一条消息已经被标注。如果图标是一个 action，那就选择一个能与背景色有强烈对比的颜色。



通知图标

如果你的应用程序会推动通知，那就请提供一个图标。当有消息推送过来时，图标就会显示在 Status Bar 上。



尺寸&比例

Action Bar 上的按钮为 24x24dp。

主要区域&大小

全部尺寸 24x24dp
图形区域 22x22dp

风格

保持平面简单的风格，
使用简单的视觉图像作
为你程序的启动图标

色彩

通知图标必须完全是全白的。而且
系统可能还会将它缩小、变暗。



Writing Style-文案书写风格

在为你的程序写文案的时候，请参照以下几点：

1. **言简意赅。** 保持简洁、简单、准确。使用 30 个以内字符（包括空格）来进行描述，不要使用不必要的语句。
2. **简单易懂。** 将你对话的人想象为一个聪明能干的人，但是他可能不懂技术语言或者他英语不是非常熟练。
3. **保持友好**，缩略语句。直接跟读者交谈时请用第二人称（“你”）。如果你的文案读起来跟你设想的完全不同，那么你可能是用了不正确的书写方式。文案不要很突兀、不要让人讨厌，而要让用户感到安逸、快乐和活力。
4. **把重要的放在前面。** 前两个词（大约 11 个字符，包括空格）至少要包括最重要的字符串信息。如果不是就重写文案吧。
5. **只需要描述必要的那些**，除此之外没有其他。不要试图去解释非常微妙的差别，不然就是失去很多用户。
6. **避免重复。** 如果一个重要的语句在屏幕和文本域中反复出现，那你需要精简这个用词。

举例

1. 言简意赅。from 设置向导

 Too formal	 Better
Consult the documentation that came with your phone for further instructions.	Read the instructions that came with your phone.

太正式

请参考随机附赠的文档来查看进一步说明。

更好的

阅读随机附赠的说明书。

2. 简单易懂。From 位置设置

 Confusing	 Better
Use GPS satellites When locating, accurate to street level.	GPS Let apps use satellites to pinpoint your location.

让人困惑

（使用 GPS 卫星）

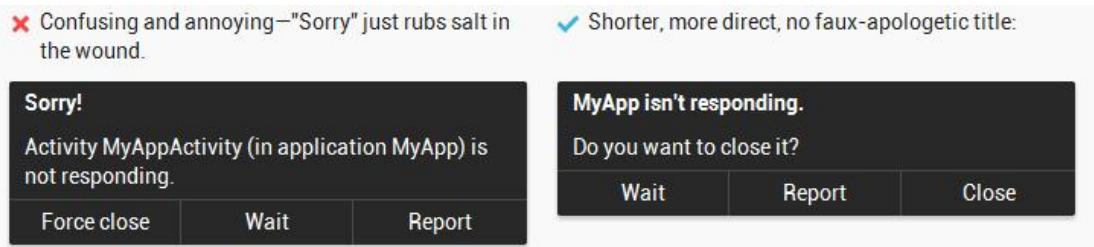
当定位时，精确到街道级别。

更好的

（GPS）

让应用程序使用卫星来确定你的位置。

3.保持友好。From 程序崩溃时出现的对话框



让人困惑和烦恼——一句“对不起”就像是在伤口上撒盐。
(Sorry !)
Activity MyAppActivity(在 MyApp 程序中)已无响应。

简短，直白，没有表示歉意的标题
(MyApp 无响应)
你想关闭它吗？

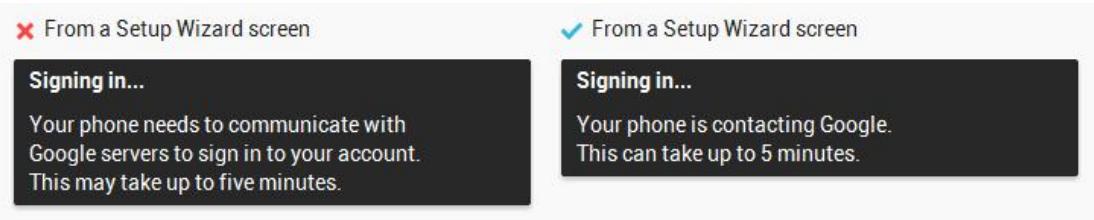
4.把最重要的放在前面



任务在后
点击下一步完成 Wifi 连接设置。

任务在前
要完成 Wifi 连接设置，点击下一步。

5.只需要描述必要的那些



你的电话需要连接 Google 的服务以便登录你的帐户。
这也许会花费一点时间。

你的电话正在连接 Google。
这将花费 5 分钟。

Patterns-模式

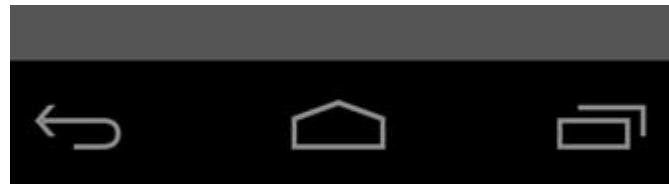


设计一个体验一致并可预测的应用程序

New in Android4.0-Android4.0 新特性

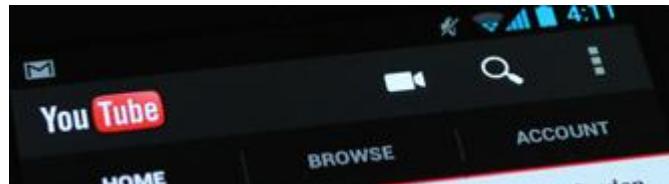
Navigation bar

Android4.0 放弃使用传统的物理按键，取而代之的是一个包括 Back、Home 和 Recents 按钮的虚拟 Navigation bar。如果想了解 Android4.0 是如何适应那些需要物理按钮的程序，请参考[兼容模式](#)。



Action bar

Action bar 是 Android 应用程序最重要的组成部分，它为整个程序提供了一致的导航，并且允许你放置 action 操作按钮。



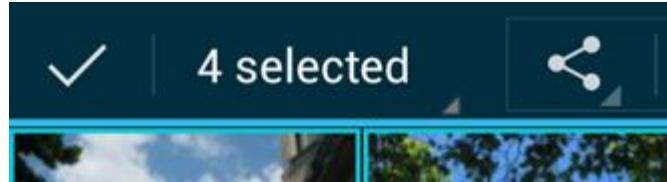
Multi-pane layouts (多面板布局)

在 Android 世界中，设计一个能够在不同参数和不同尺寸屏幕上操控自如的应用程序是非常重要的。Multi-pane 布局允许你整合不同的 Activity：一个应用程序在手机可能是多个单独的 Activity，但是该程序可以在平板电脑上将多个 Activity 联合显示在同一个屏幕上。（比如 Android 下的联系人程序）



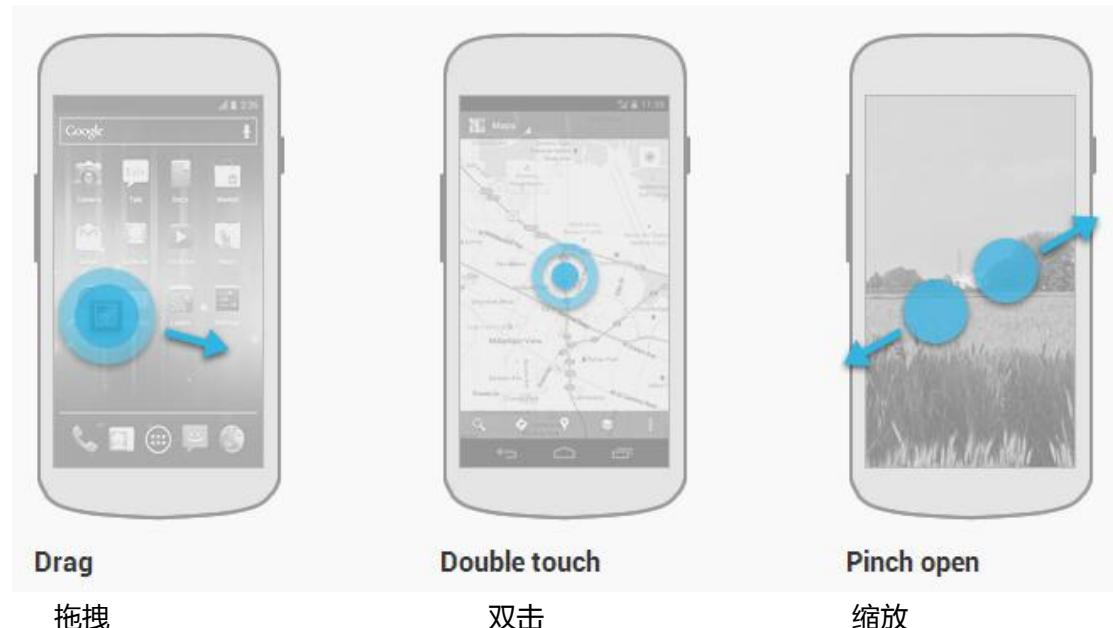
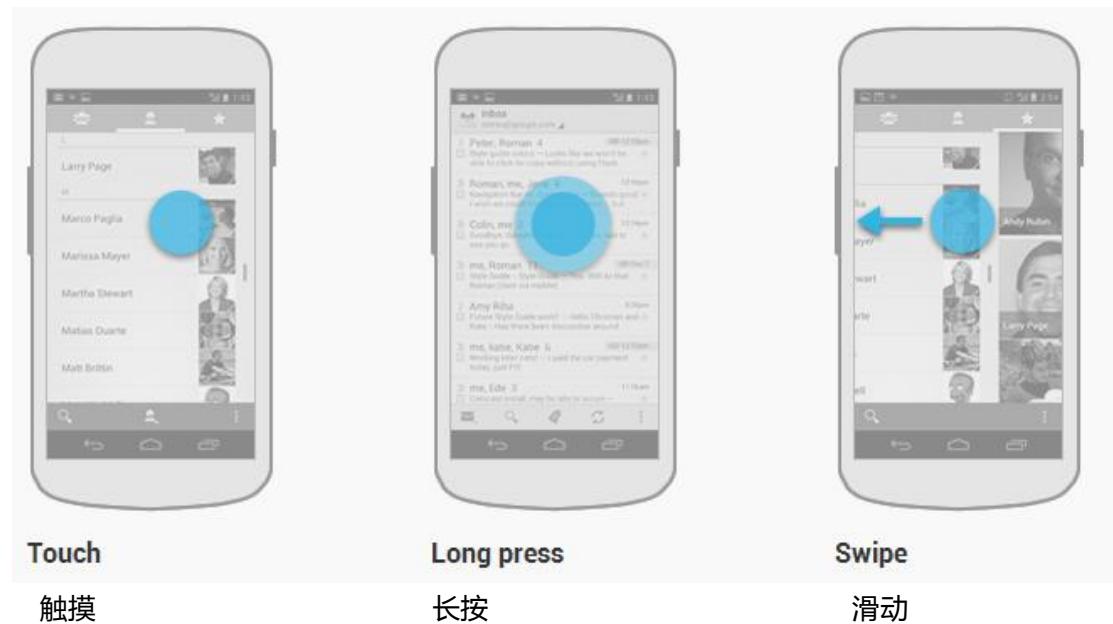
Selection

常用的长按手势将会启动 contextual action bar, bar 上提供了各种 actions (比如“全选”、“复制”），他们用来可以操作已经选中的数据条目。



Gestures-手势

手势允许用户通过屏幕上可触控的控件物体和应用程序进行交互操作。下面展示了 Android 支持的一系列主要操作手势。



App Structure-应用程序结构

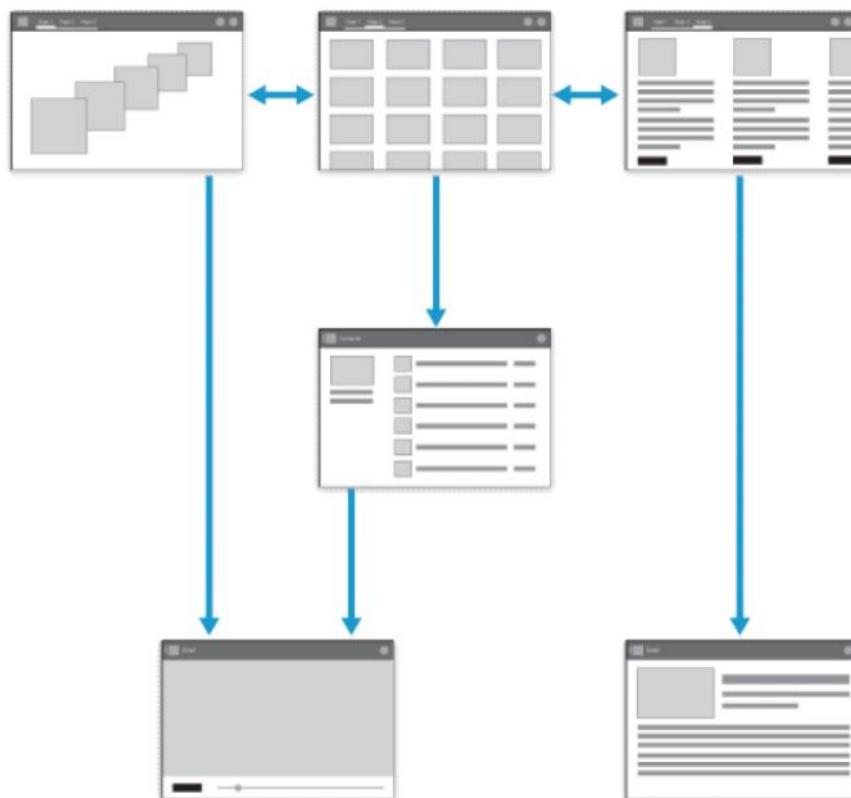
应用程序有很多不同的类型，所以它们能满足的用户需求也是不一样。例如：

- 1.照相机或者计算器一类的应用程序，主要呈现于最重要的单一的 activity 中。
- 2.拨号一类的应用程序，主要的目的就是快速地在不同的 activity (拨号、通话记录、最近联系人) 之间进行切换，并不需要层次很深的导航。
- 3.Gmail 或者 Play 商店一类的应用程序，整合了包含各种数据信息的 views，需要使用深层次的导航才能将这些内容高效地组织起来。

你的应用程序结构在很大程度上取决于程序的内容以及你想为用户呈现的 UI。

常用结构

一个典型的 Android 应用程序是由 top level 和 detail/edit views 组成。如果程序的导航层次很深而且繁多复杂，请使用 category view 作为 top level 和 detail views 的连接。



Top level views

应用程序的 top level 通常包括各种不同的视图 activity。这些视图可以用不同的方式来显示相同的数据。除此之外它们也会为你的应用程序呈现各种不同的功能。

Category views

Category views 可以让你更深一步地了解数据。

Detail/edit views

在 Detail/edit view 中你可以使用或者创建数据。

Top Level

需要特别注意应用程序启动的第一个 activity，因为这是人们在加载你的应用程序后在屏幕上看到的第一个界面。所以，需要给新用户和老用户带来相同的使用体验。

问问自己：“在我的应用程序中，用户最喜欢做什么？”以此来改善你的启动界面以达到更好的体验。

把内容放在前面

很多应用程序注重内容的呈现。不要使用 navigation-only 的界面，取代代之的是将程序的核心内容呈现在屏幕上，让用户能立刻了解程序的主旨内容。请使用视觉设计出众并且能适应各种不同类型、尺寸的屏幕的 layout (UI 布局文件)。

Play 商店的启动界面上包含了 Apps 、 Music 、 Books 以及 Movies 的导航。同时它也为用户量身定制了一些建议内容和可能感兴趣的内容。在 action bar 上用户可以很容易地使用搜索功能。

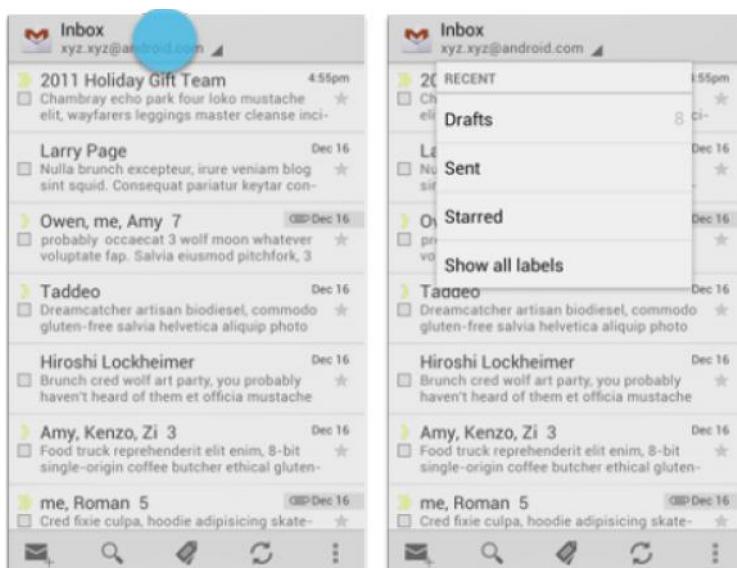


将 action bar 用于导航和放置 actions 操作按钮

你程序的每一个界面都必须显示 action bar 用于导航和放置重要的 actions。

在 top level，需要特别考虑 action bar：

1. 使用 action bar 来显示应用程序的图标和名称。
2. 如果程序的 top level 包含多视图，或者如果数据来源于多个可切换的账户，确保用户能够轻松地使用 action bar 上的视图切换控件。
3. 如果你的应用程序允许用户创建内容，请考虑 top level 的内容访问权限。
4. 如果程序的内容是可搜索的，请在 action bar 添加搜索按钮，方便用户检索查看。

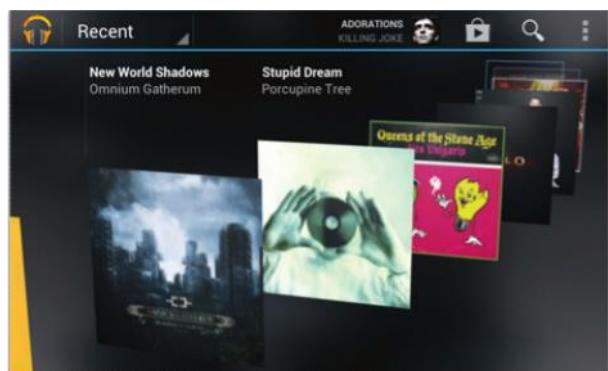


Gmail 是效率应用程序，一个高效的、快速阅读大量数据的列表视图能够很好的选择。导航控件支持不同账户和不同便签之间的切换。底部的 split action bar 提供了新建邮件以及搜索按钮图标。

为你的应用程序创建个性的控件

在应用程序的 action bar 上创建一个个性化的控件，然后重新通过此控件来传达出数据，信息等等，甚至是用户与程序进行交互的方式。尤其是对于多媒体应用来说，可以尝试独特的风格布局来展示内容和数据，超越之前单调的列表视图。

3D 旋转木马式的专辑封面是 Music 程序中一个个性的控件。最近播放视图下默认地将正在播放的歌曲加以关注。(还有 action bar 上会显示当前播放的歌曲)



Categories

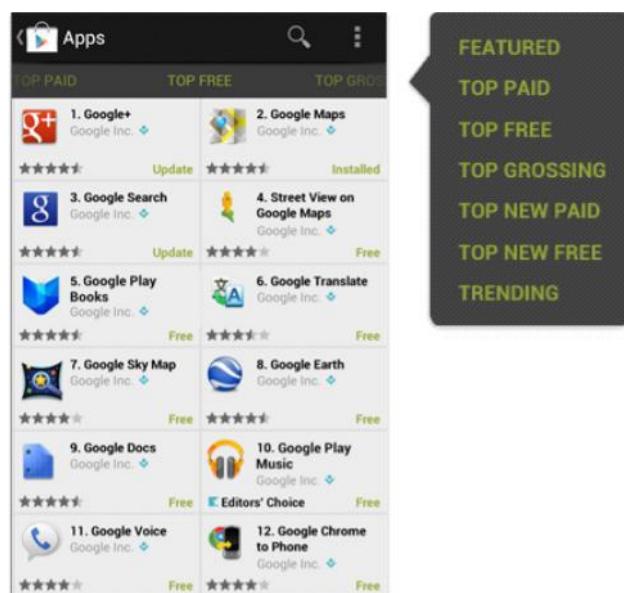
一般来说，一个目标层次深、以数据来驱动的应用程序是通过有组织的分类来导航到 detail level (可以查看管理数据) 的。要想减小导航的压力，那就请保持程序的导航层次不要太深。

从 top level 到 detail level，导航步骤的数量通常取决于应用程序的内容。下面有几种方法可以将导航任务化繁为简。

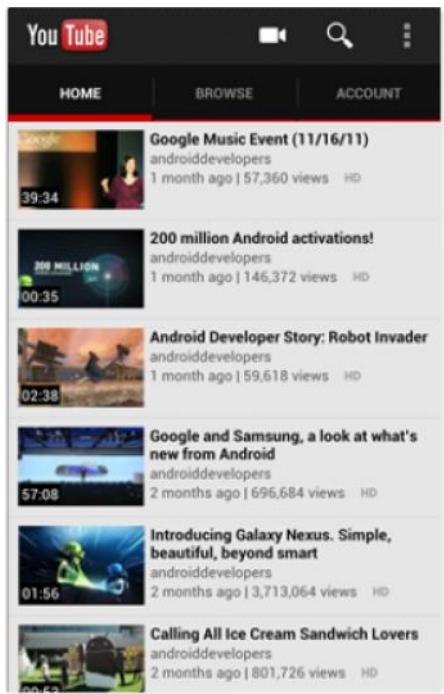
使用便签来整合选项和数据内容的显示

如果分类非常相似而且数量不多，那么标签栏就会非常好用。它的优势在于当一个层级移除后，用户的注意力仍然在数据上。在有很多分类的时候，横向的导航切换更加自然，比那些步骤明确的导航方式体验更加好。

如果分类都是近似的，可预测的，或关系密切的，请使用 scrolling tabs (在这里，并不是所有的事物都是同时可见的)。标签数量最好控制在 scrolling tabs 可承受的范围。按照经验来说标签不能超过 7 个。



Play 商店使用标签栏显示分类选项和内容。在内容区域向左/右滑动可以在不同分类之间进行浏览。

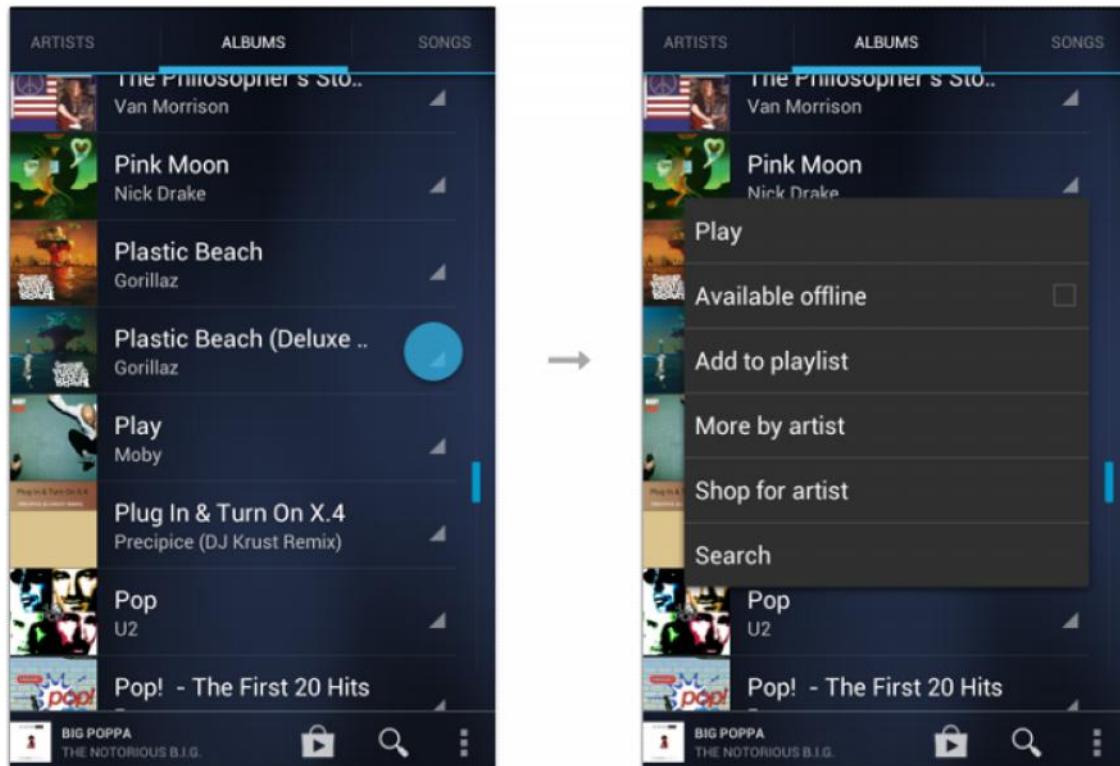


如果 分类的内容不是非常近似 , 请使用 fixed tabs , 一遍所有类别同时在视图中显示。

Youtube 使用 fixed tabs 来切换不同的 , 没有联系的功能区域。

允许切断层次结构

利用快捷方式人们可以快速地达到他们的目标。允许顶级调用的操作从内部数据项列表或网格视图显示突出的行为直接在列表视图项目可以使用下拉框或划分列表项。这让人们对数据调用操作,而不用导航沿着层次结构。



Music 程序允许用户在分类视图 (专辑) 中控制一个数据项 (歌曲), 这样就不需要设置导航到这首歌曲的详细信息界面。

操作多个数据项

尽管类别视图主要用于指导人们详细内容,请记住,经常有理由采取行动集合的数据。

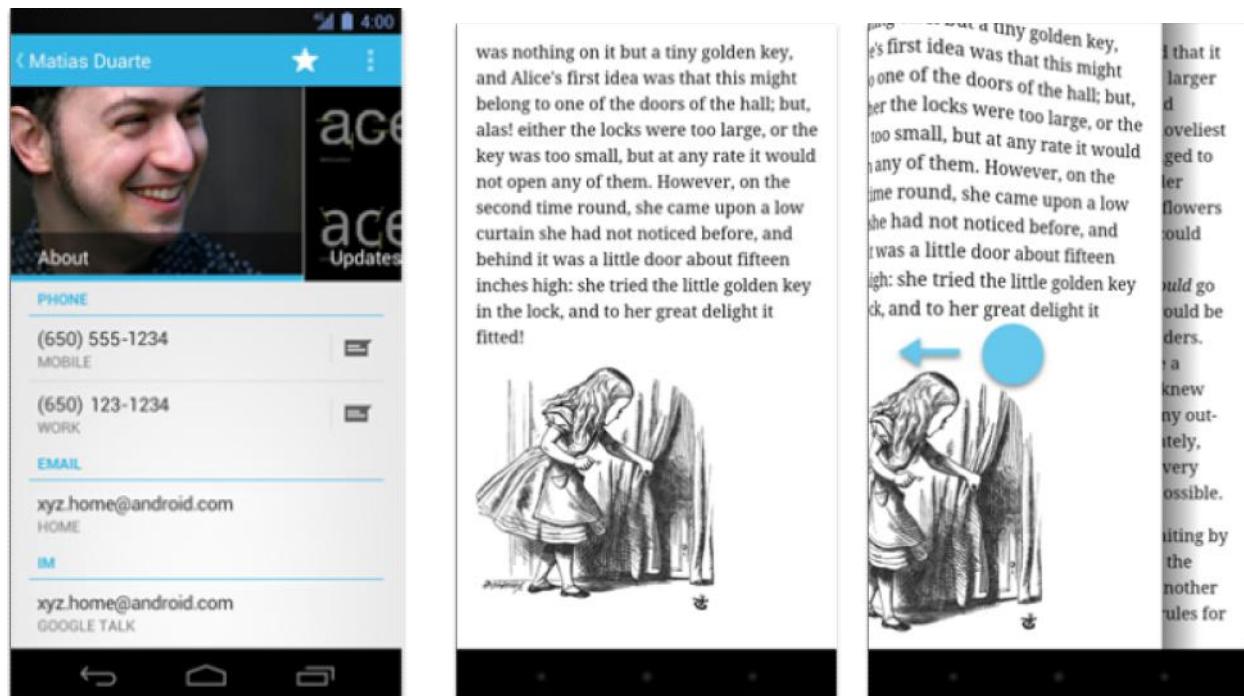
例如，程序允许人们删除 detail view 中的一个项目，你应该同时允许他们可以在分类中删除。分析哪些细节视图操作适用于数据项的集合，然后在多个分类视图中，让多选允许应用到多个项目。

Details

Detail view 用于显示应用程序的内容和数据，它的布局取决于要显示的数据类型。

布局

考虑到积极的人将执行在详细信息视图和安排相应的布局。对于身临其境的内容,利用熄灯模式允许查看内容，以及使用全屏显示防止用户分心。

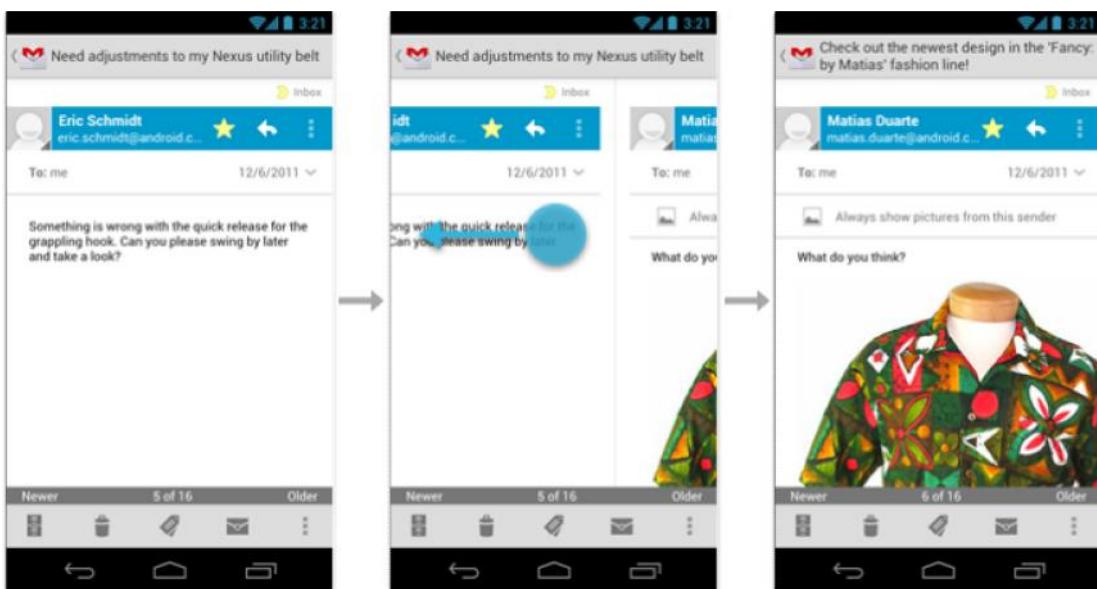


联系人应用程序的 detail view 目标是显示通信选项。这个列表视图,可以实现有效的扫描和快捷的电话号码、电子邮件地址和其它信息项。划分条目被用来结合调用和消息传递到一个紧凑的行项目。

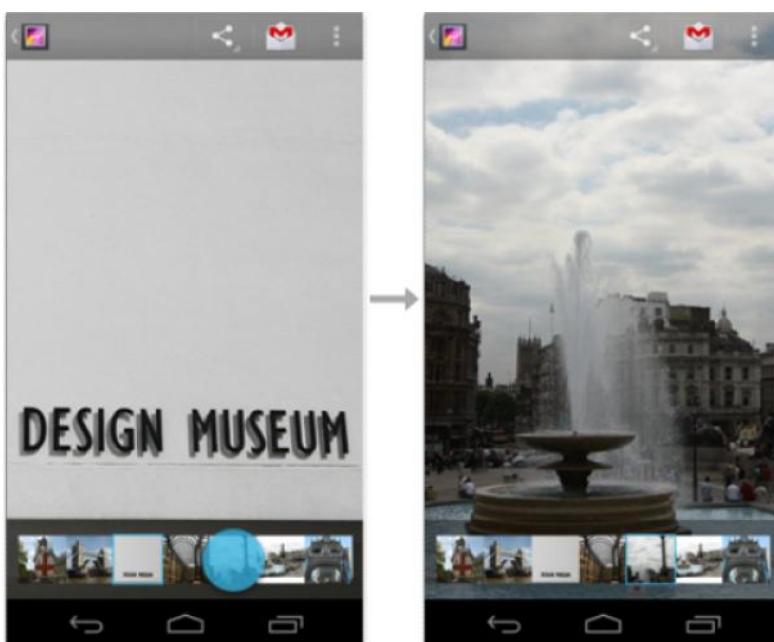
谷歌图书的 detail view 都是关于复制的阅读体验的一个实际的书。翻页动画强化一种观念，为了创建一个身临其境的体验，这个应用程序进入熄灯模式,它隐藏了所有 UI 交互系统。

让 detail view 之间的导航更有效率

如果您的用户可能想查看多个项目序列，允许他们从内部条目之间的 detail view 进行导航切换。使用滑动视图或其他技术，如 filmstrips，来实现这个。



Gmail 使用滑动视图在邮件 detail view 之间进行导航。



为了支持图片上的滑动手势操作，相册程序提供了 filmstrip 控件让人快速地跳转到指定照片。

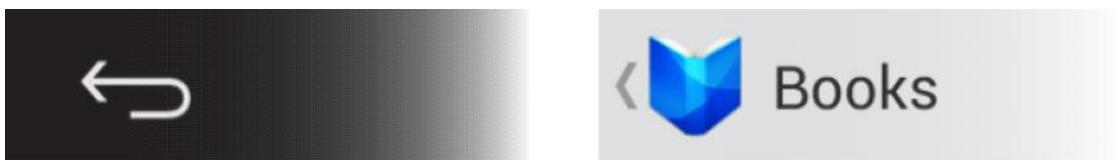
备忘清单

1. 寻找方法将有用的内容显示在启动界面上。
2. 使用 action bar 控件以保持同一的导航风格。
3. 使用横向导航和快捷图标，避免程序导航层次过深。
4. 使用多选操作允许用户操作数据集合。
5. 允许 detail views 之间的滑动导航。

Navigation-导航 (Back&Up)

统一的导航模式是用户体验最基础的部分。不一致的和意料之外的导航方法会让用户产生挫败感。Android3.0 引入了全新的导航机制来改变这一切。认真参考 Back and Up 的建议指导，这将帮助你设计出一个交互出色的导航方式。

Android2.3 以及更早的设备依靠设备系统 Back 按钮完成部分导航操作，而在 Android3.0 中新引入的 action bar 上，第二个导航按钮出现：Up 按钮，由程序图标和左侧的小箭头一同组成。



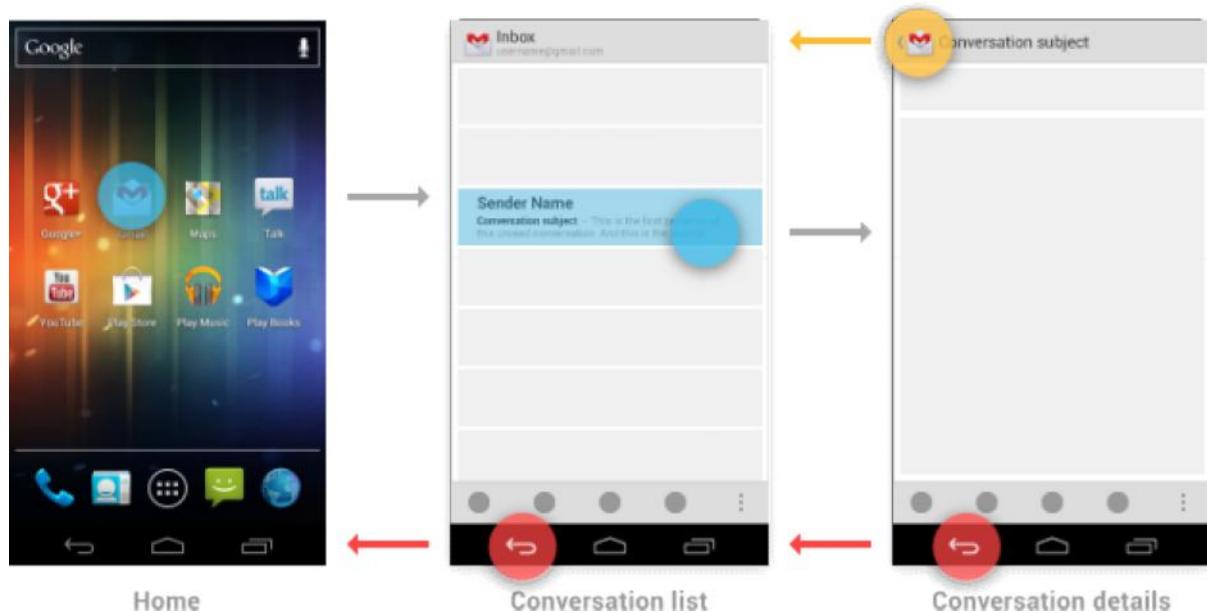
Up vs Back

Up 按钮用于同一个应用程序中不同 activity 之间的返回。例如，当用户点击了 activity A 上的条目跳转到了 activity B，那么在 activity B 中的 action bar 上应该提供 Up 按钮用于返回 activity A。

如果一个 activity 是程序的首页，那就不必提供 Up 按钮。

系统按钮 Back 同样用于导航，按照时间倒序，逐一切换用户浏览过的 activity。这通常是基于 activity 使用时间，而不局限于程序中的层次结构。

如果浏览过的的 activity 是在同一个程序中，那么 Back 和 Up 按钮的效果是一样的。但是有些情况下，Back 按钮可以让用户返回主屏幕，甚至是回到不同的应用程序。（Up 按钮只会在同一个程序内有效）



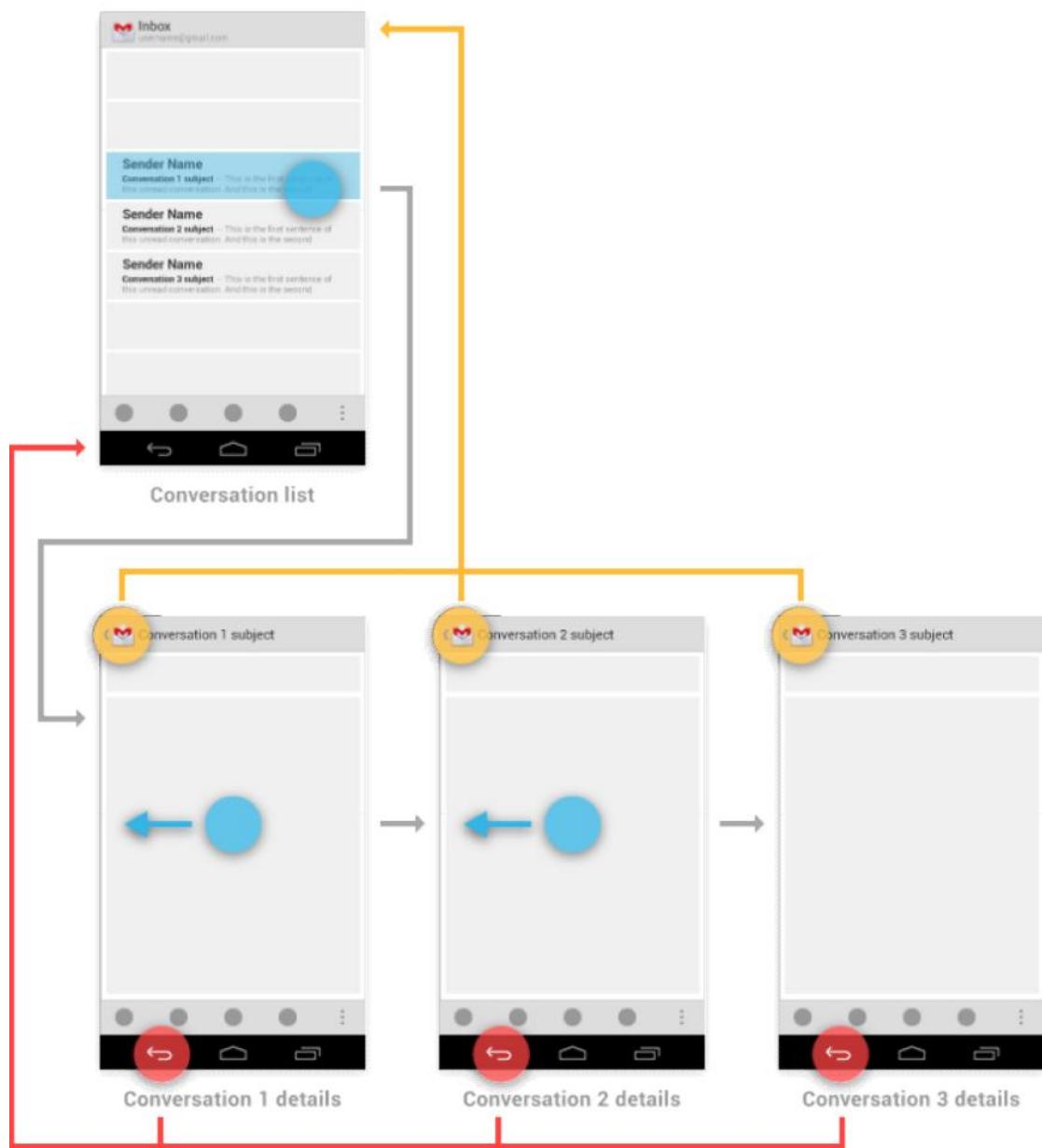
Back 按钮还能提供一些非导航类的功能：

1. 取消浮动窗口（对话框、弹出窗口）
2. 取消 contextual action bars，和选择的条目
3. 隐藏键盘

应用程序中的各种导航情况（此部分翻译精简，请理解图片，真是够反人类的）

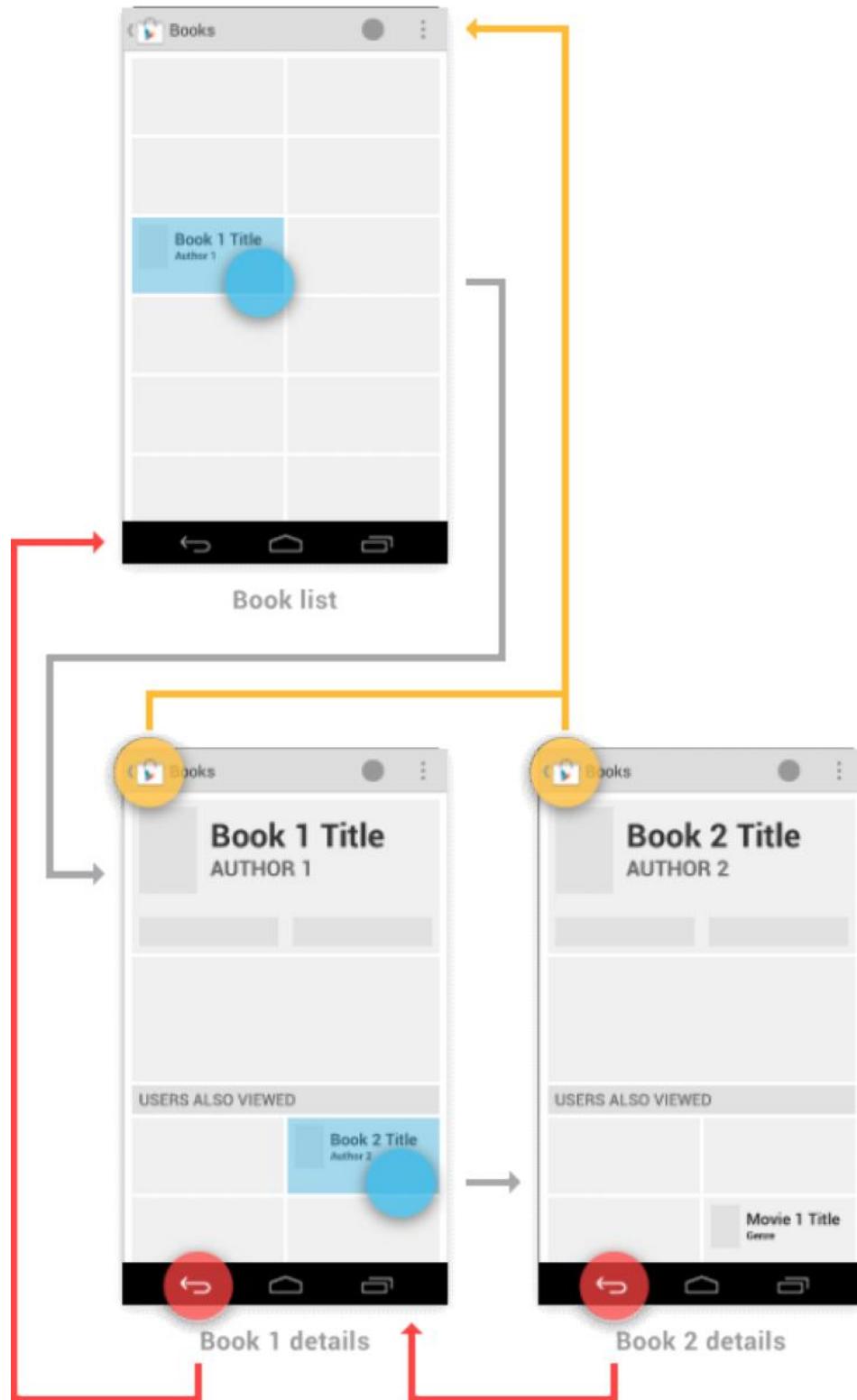
同一应用程序中相似 activity 之间的导航

Gmail 中，点击一个条目进入邮件详情，同时滑动切换邮件，此时 Back 和 Up 按钮的作用相同，都是返回 Gmail 首页，而不是返回“上一封邮件”。Back 是返回，但是由于每个邮件都是在导航上都属于同一层次，所以 back 返回首页。



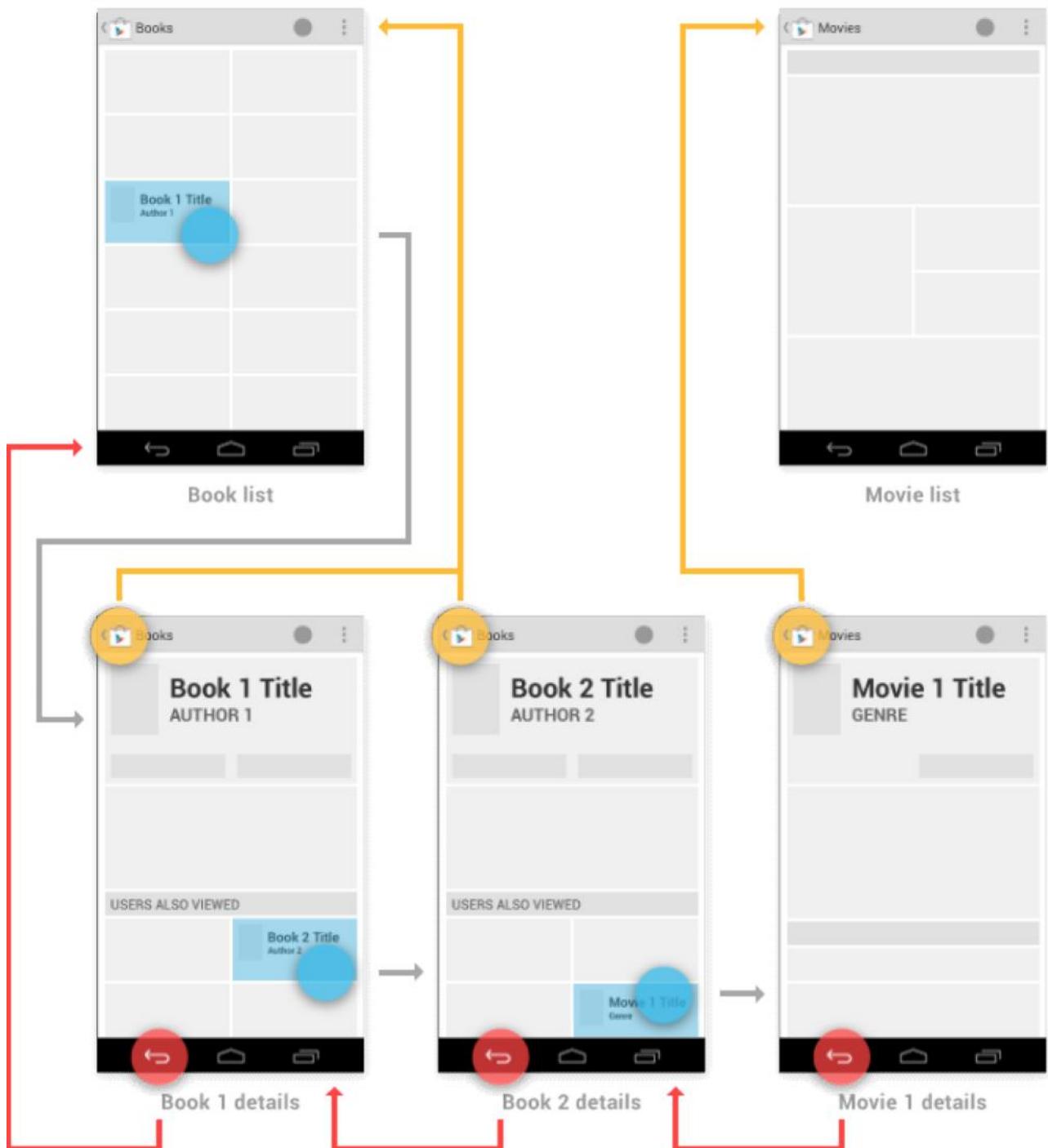
但是，Back 和 Up 按钮有时候作用也不同，这跟应用程序的采用的导航方式有关。

Play 商店中，从首页点击 item1 进入 activity1，然后再次点击 item2 进入 activity2。按 Back 按钮之后，先返回 activity1 再返回 activity2。Up 按钮直接返回首页，因为 Up 按钮是“返回上一层” activity1 层属于 detail level，所以返回到属于 top level 层的首页。



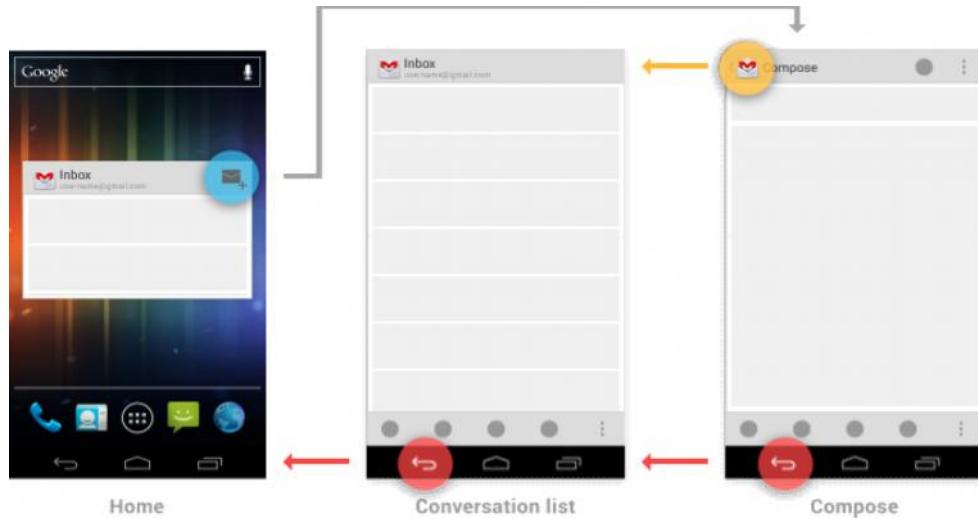
同一应用程序中不同分类之间的导航。

在 Play 商店中的 Book 分类点击 book1 进入 activity1，点击 activity1 上的 book2 进入 activity2，然后点击 activity2 上的 movie1 进入 activity3，那么如果按 Back 按钮，界面就会依次返回。如果在 activity1&2 按 Up 按钮，那么就会返回 Book 分类首页，因为 activity1&2 是从属于 Book 分类的 detail views；如果在 activity3 按 Up 按钮，就会进入到 Movie 分类的首页，这是因为 activity3 是 Movie 分类的 detail view。



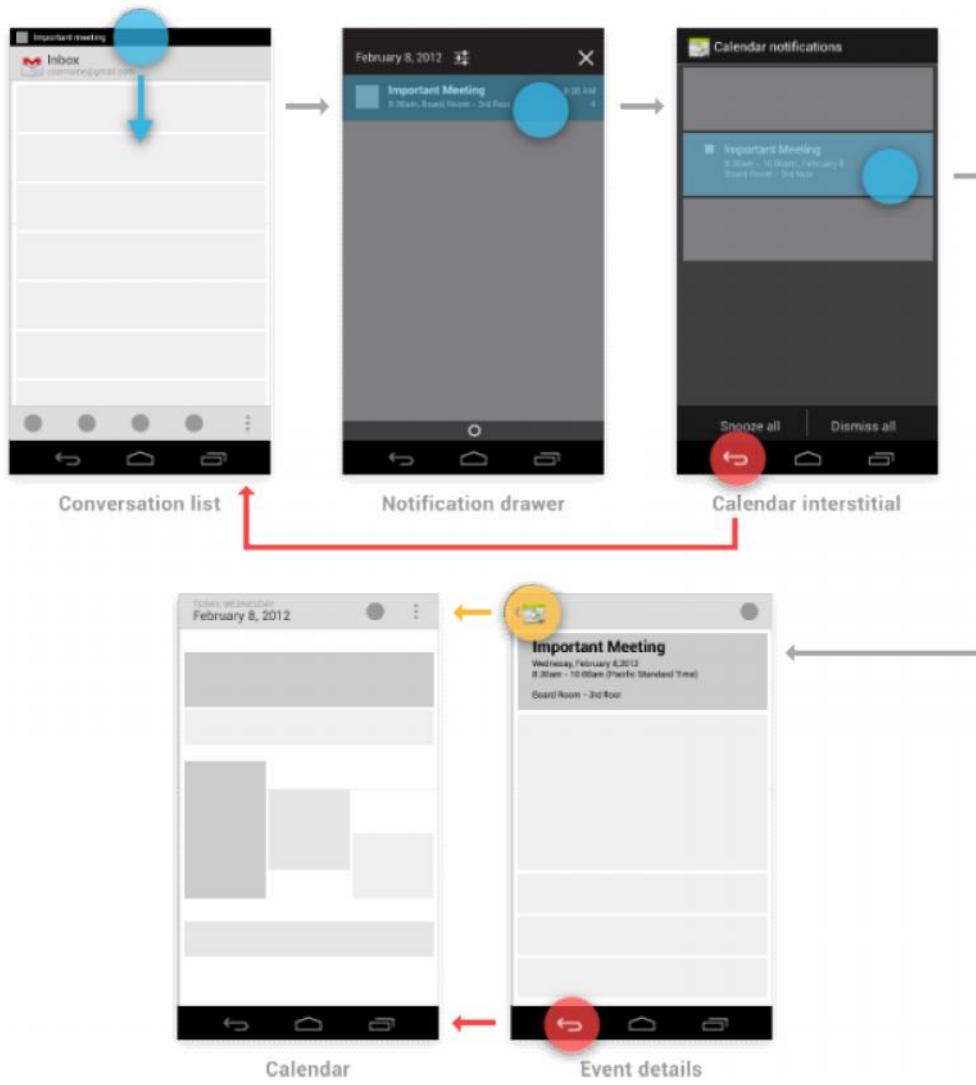
应用程序与桌面插件之间的导航

点击 Gmail 桌面插件的新建邮件按钮就会进入新建邮件 activity1，此时 Back 和 Up 按钮效果一样，按下后都是返回 Gmail 首页。



应用程序与通知系统之间的导航

在我使用 Gmail 的时候，有一条来自日程的通知，点击消息后就进入了日程通知 activity1 (已经在日历程序里)，如果此时点击 Back 按钮，就会回到 Gmail 首页。然后点击 activity1 上的条目进入日程详情 activity2，如果此时点击 Back 按钮，就会返回到日历的首页。



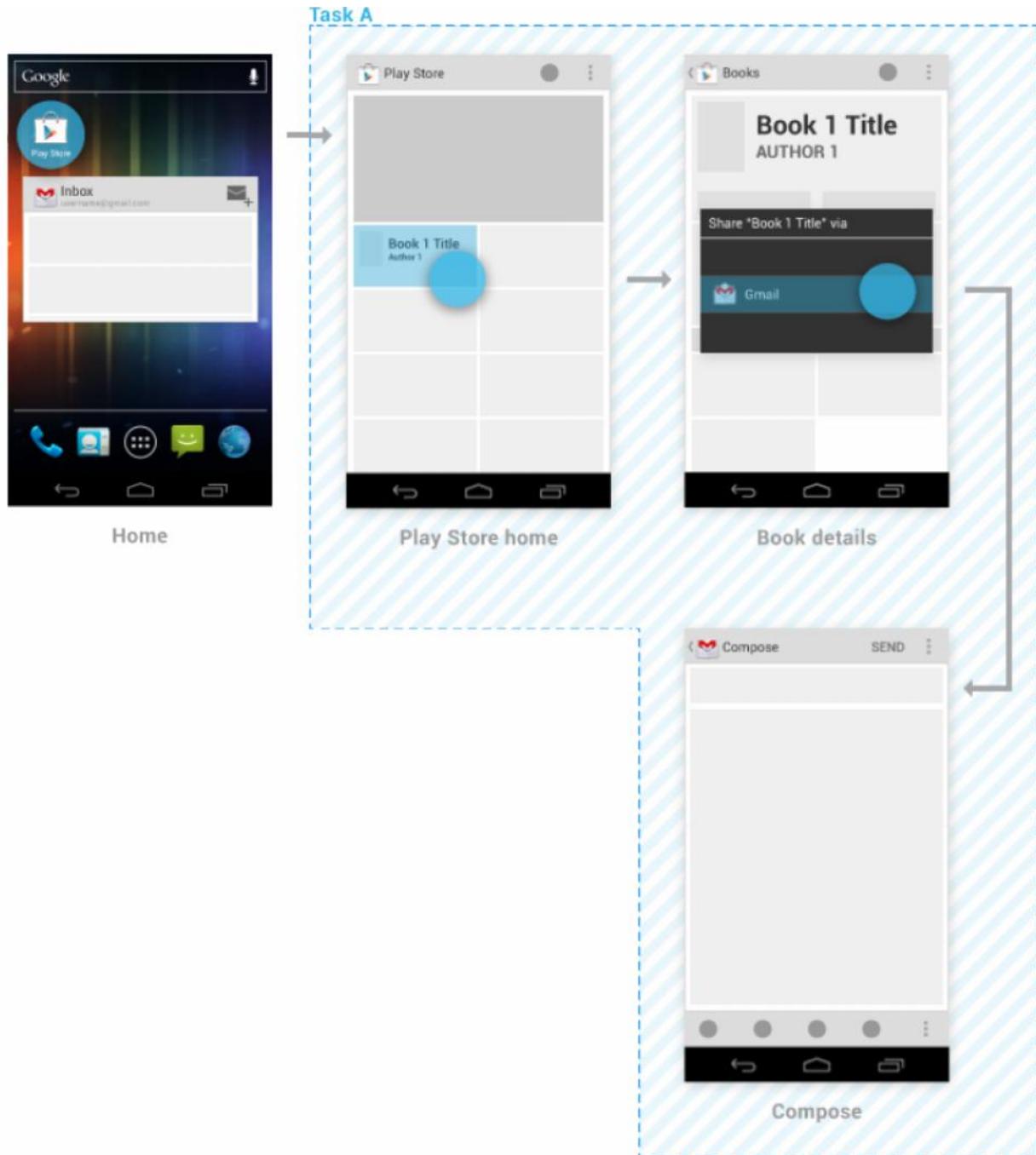
弹出式通知消息的导航

弹出式通知和传统通知中心的通知不一样。当使用 Gmail 时，弹出 Gtalk 聊天邀请，可以按 Back 按钮拒绝要求返回 Gmail 首页，也可以按 accept 进入 Gtalk 聊天界面 activity1。如果目前处于 activity1，Back 和 Up 按钮效果一样，都是返回 Gtalk 首页。



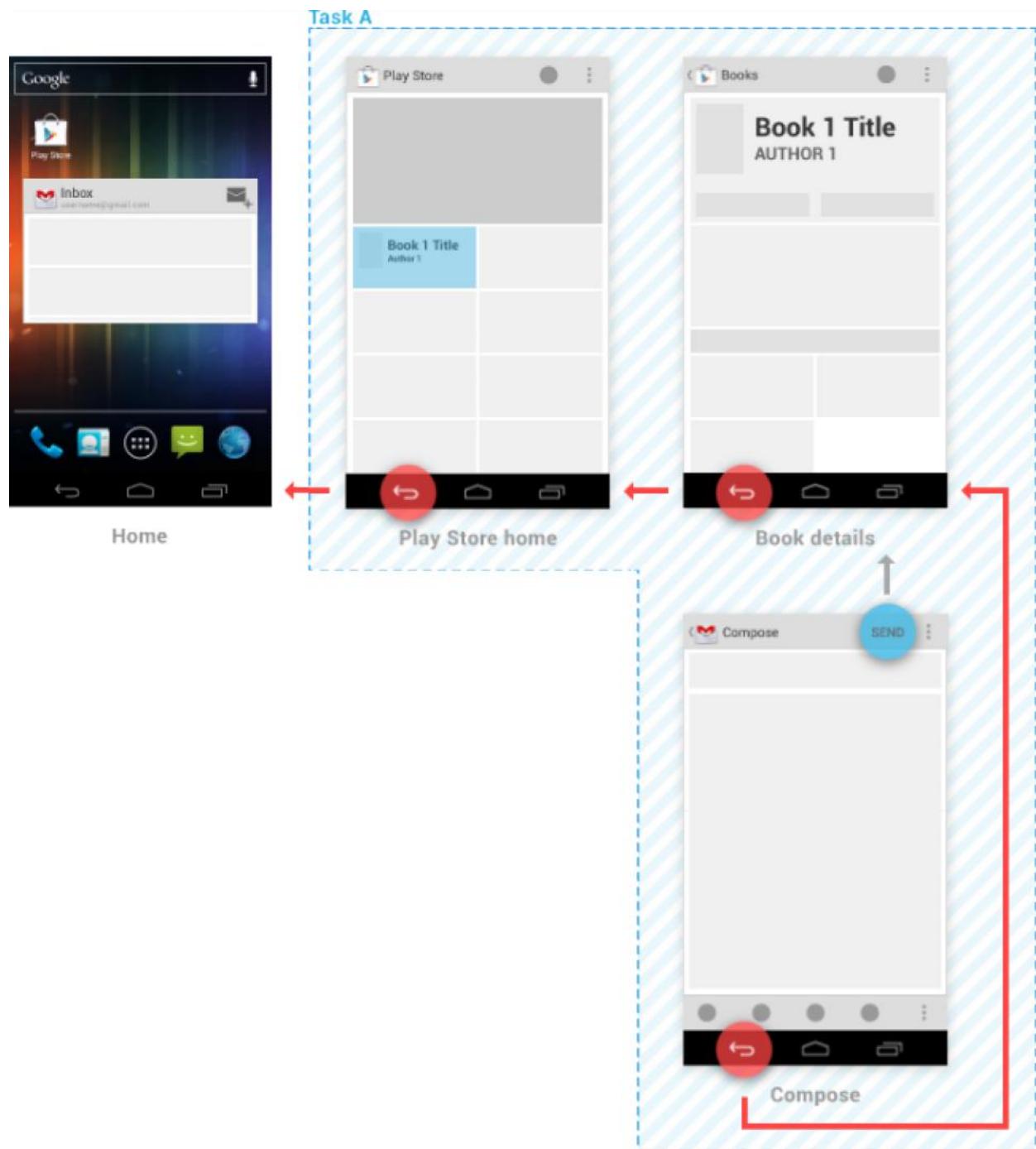
应用程序之间的导航

在 Home 屏点击 Play 商店进入商店首页 ,然后点击 item1 进入 activity1 ,然后采用 Gmail 分享 item1 , 就会进入新建邮件 activity2。



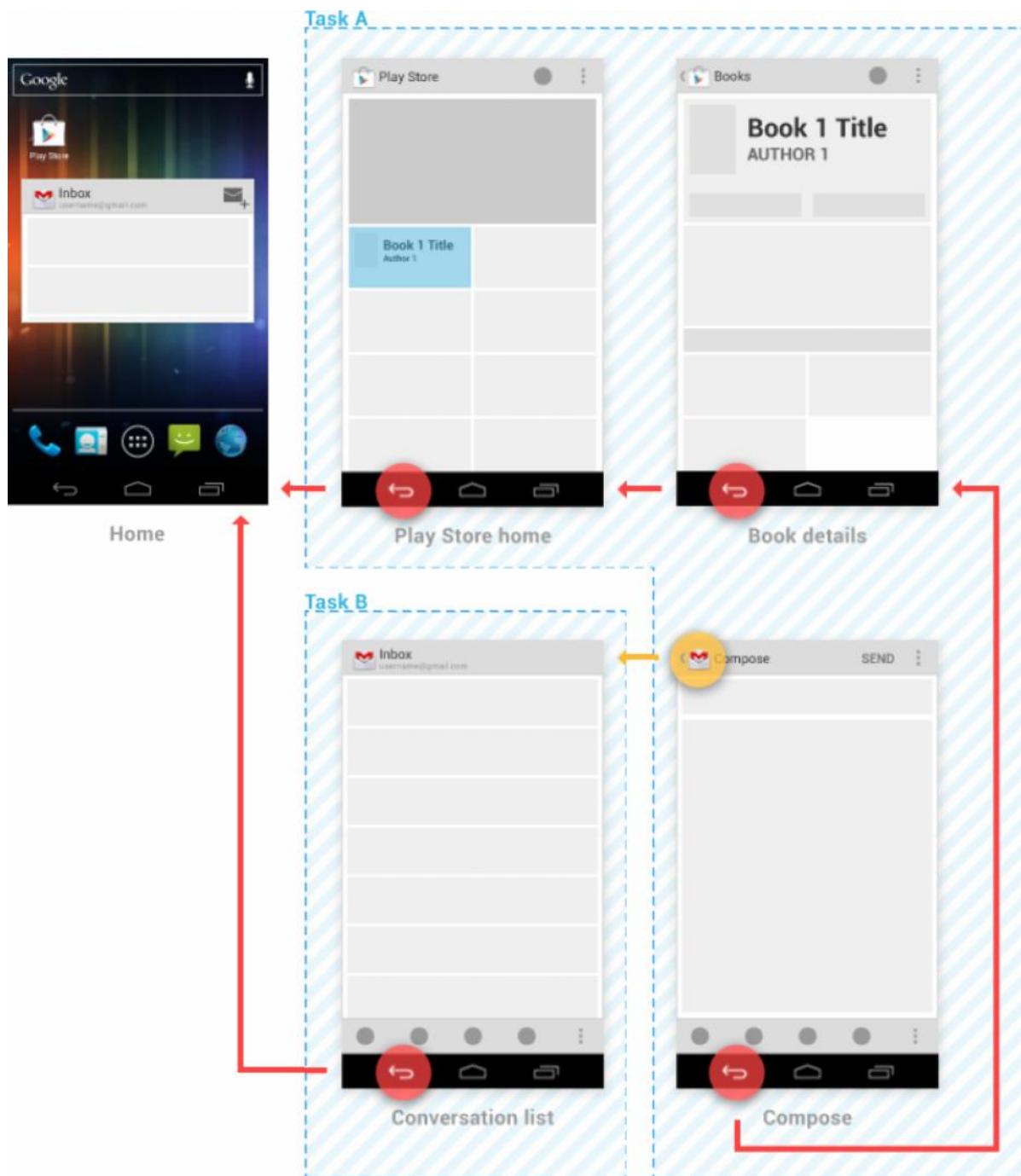
(接上页)

如果邮件发送成功，就会自动返回 item1 的 activity1



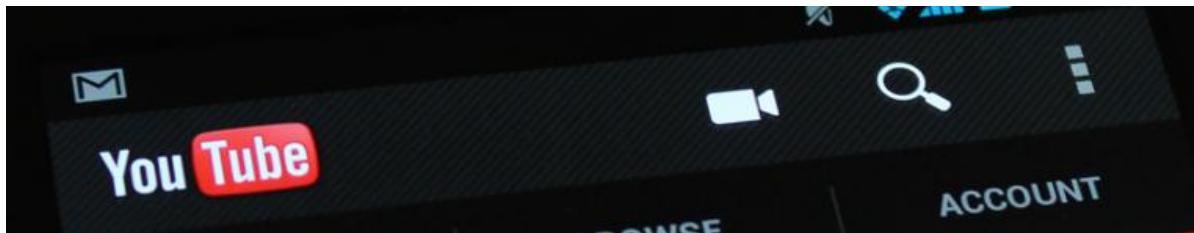
(接上页)

如果新建邮件时按 Back 按钮取消，那就会返回到 item1 的 activity1。按 Up 按钮，那就会进入 Gmail 首页。



(看完了，看懂了，迷糊了。所以，认真你就输了！)

Action Bar



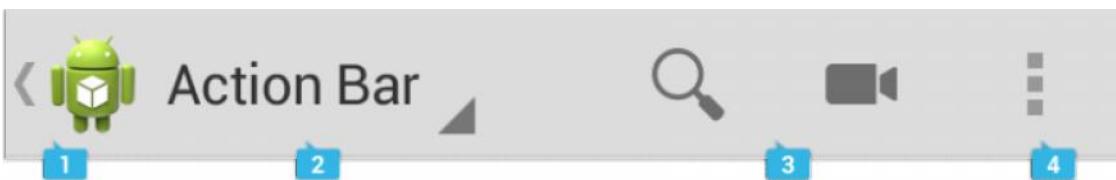
Action bar 可以说是 Android 应用程序最重要的构造元素，它位于个 activity 的顶部，一般始终会出现在程序中。

Action bar 的作用：

1. 将重要的 actions (比如新建、搜索等) 置于显著的位置方便用户快速使用。
2. 为程序提供导航功能和视图切换控件。
3. 在 action bar 上提供 action overflow 用于放置不常用的 action , 减少杂乱。
4. 为特殊控件提供特定的区域。

如果你是 Android 开发新手，请注意 action bar 是最重要的设计元素。请遵循这里列出的一些指导，这将帮助你设计出符合 Android 界面风格的应用程序。

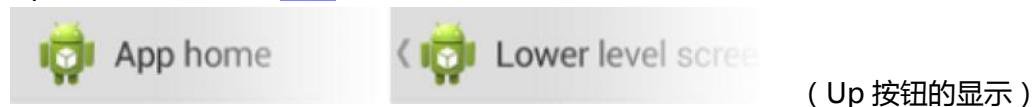
一般结构



Action bar 可以分为四个功能区域。

1. 程序图标

程序图标是程序本身的一个个人标签，它可以被替换成各种商标和品牌。重要：如果你的应用程序中的界面不是 top level (不是首页)，那就请在图标左侧设置 Up 按钮。更多关于 Up 按钮导航，请参考[导航](#)。



2. 视图切换控制

如果你的应用程序需要通过不同的 activity 来显示不同的数据内容 ,action bar 的这一部分提供了快速切换的控件，比如 drop-down menus 或者 tab controls。

如果你的程序不支持多视图，那你可以在这里显示一些非交互的数据，比如程序的名称或者品牌信息。

3.Action buttons

为你的应用程序提供了重要的 actions 按钮，不适合 action bar 的 actions 按钮会自动移到 action overflow。

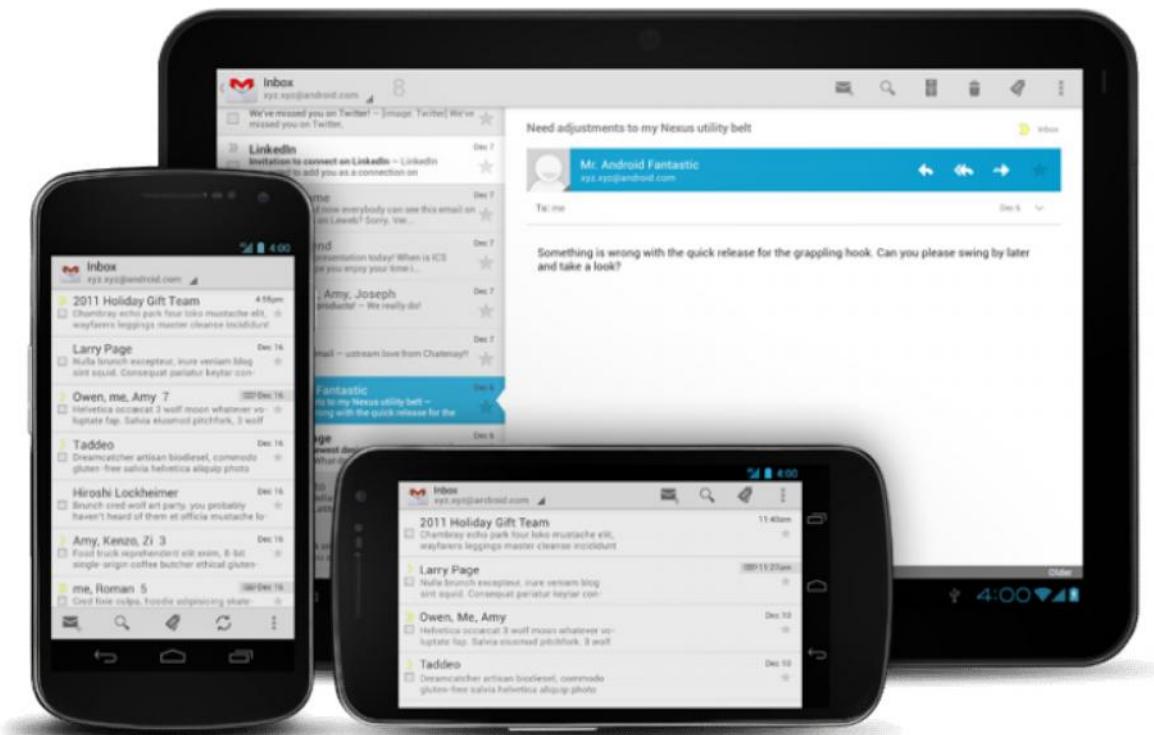
4.Action overflow

用于放置不常用的 actions buttons。

适应旋转和不同屏幕尺寸

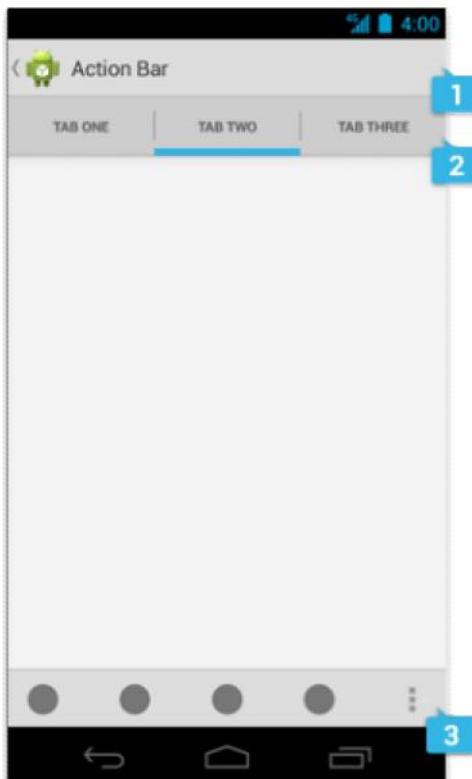
如何适应屏幕旋转和不同尺寸是 UI 设计中一个重要的问题。

为了解决这个问题，你可以使用 split action bars，它可以让 action bar 的内容跨区域显示于 main action bar 的下方和屏幕的底部。



如图所示，split action bar 在竖屏状态下显示与屏幕的下方，在横屏状态下与 main action bar 结合在一起。

Split action bar 布局思考



当你把内容分为多个 action bars 时，有三种通常的位置：

- 1.Main action bar
- 2.Top bar
- 3.Bottom bar

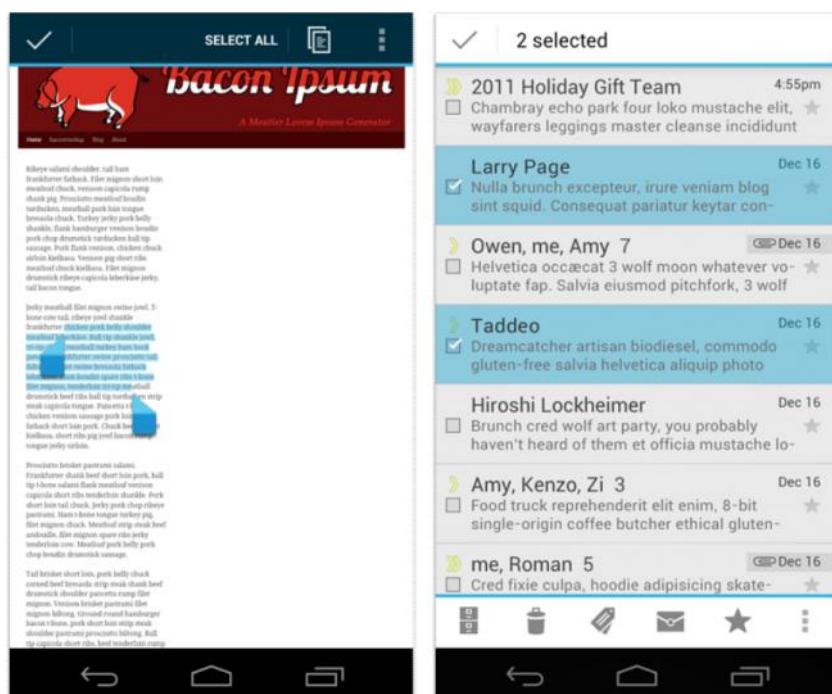
如果当然 activity 包含导航，请在 main action bar 提供 Up 按钮。

想让用户快速地在不同视图进行切换，请使用 tabs 或者 spinner。

想要显示 action buttons 和 action overflow，请使用 bottom bar。

Contextual Action Bars

Contextual Action Bars (CAB) 是一种临时性的 action bar , 它会遮挡原有的 action bar 并且显示其他特定功能 action buttons (比如 “全选”、“复制”) 。 CABs 通常用于选择文本或者数据之后的操作。



浏览器和 Gmail
中的 Contextual
Action Bars

长按一个可选的数据条目就可以显示 Contextual Action Bars，并触发数据选择模式。

用户可以进行以下操作：

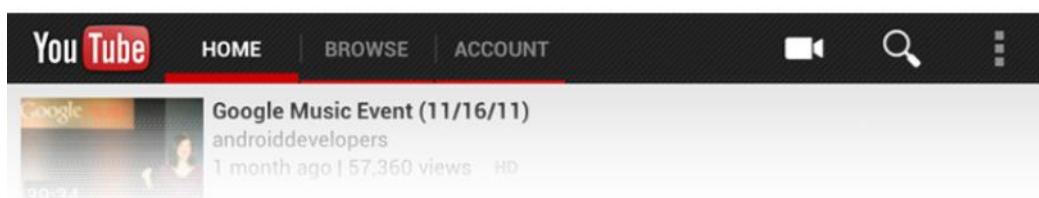
- 1.轻触一个可选元素就可选中。
- 2.可以对所有选择的条目进行统一的操作，然后 CAB 就会自动消失。
- 3.点击 navigation bar 的 back 按钮或者 CAB 的确认按钮就可以取消 CAB。同时取消已选中的数据条目。

允许你的用户通过长按操作触发 CAB，你可以控制 CAB 的内容，自定义 CAB 上的 action buttons。更多信息请参考[选择](#)。

Action Bar 基础元素

Tabs

Tabs 同时显示应用程序的多个视图，让用户快速地在不同视图间切换。如果你的程序需要经常切换显示视图，建议使用 tabs。

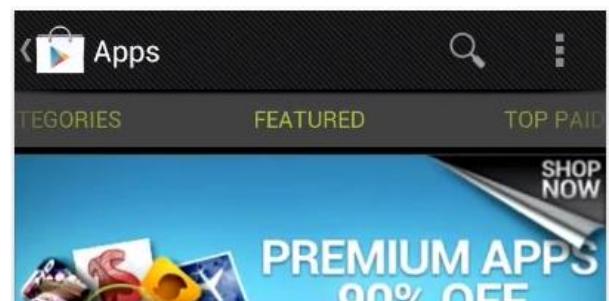


目前有两种 Tabs：fixed 和 scrollable。

Scrollable tabs

Scrollable tabs 总是占据整条 bar 的宽度，当前活动的视图对应的 tab 就会显示在 bar 中间。通过左/右滑动当前视图可以切换到其他标签和视图。

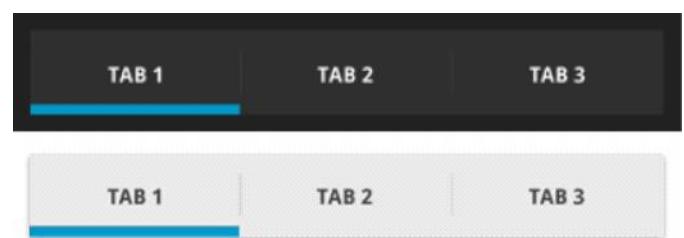
如果你的应用程序需要很多 tabs，而且不能确定 tabs 的数量(根据程序的需要可能自动会增减 tabs 数量)，那就请使用 scrollable tabs。左/右滑动内容区域就可在多个视图之间进行导航切换。



Play 商店中的 scrollable tabs

Fixed tabs

Fixed tabs 不能像 Scrollable tabs 一样，能将某些 tabs 移出可视区域。在旋转屏幕的时候，Fixed tabs 的位置可以在 Main action bar 和 bottom bar 之间自动进行转换。



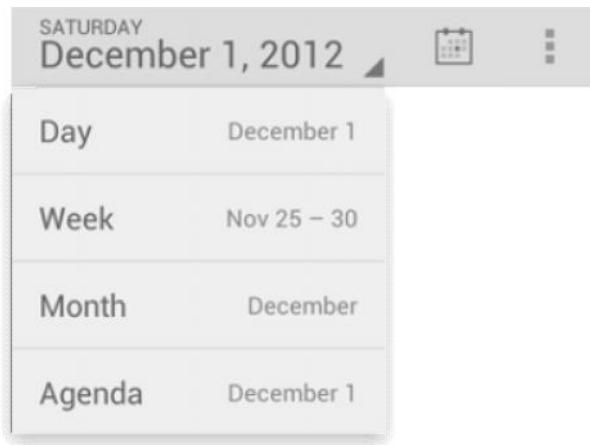
Holo dark 和 Holo light 主题的 fixed tabs

Spinners

Spinner 是一种 drop-down menu (下拉菜单)，它用于切换程序的不同视图。

如果遇到以下情况，请尽量选择 spinners：

1. 你不愿取消特殊的 tab bar。
2. 你希望用户尽可能少地切换视图。



Action buttons

Action buttons 是 action bar 上非常重要的元素，想想哪些按钮使用是最频繁的，把他们排列出来。考虑到可以屏幕区域，系统默认将那些最重要的 action buttons 放在 main action bar (或者 split action bar)上，然后将余下的放置于 action overflow 里。Action bar 和 action overflow 只显示那些对用户可用的 actions。如果一个 action 不可用的话，将它隐藏起来，而不是设置为不可用。



(一些在 Gmail 使用过的 action button 图标)

如何正确选择 actions 用于 action buttons，请使用 FIT 策划：

F-frequent (频繁的)

- 1.人们是否会经常使用，每次至少 7-10 遍？
- 2.他们会使用多少次？
- 3.采取额外步骤是否成为多余和累赘？

I-important (重要的)

- 1.你想让人知道这一 action，因为它非常酷。
- 2.这件事物在极少数情况下是否也需要？

T-typical (典型的)

- 1.在相似的一些程序中，它是不是典型存在的？
- 2.在特定的环境下，当它被置于 overflow 里之后人们是否会感到惊讶？

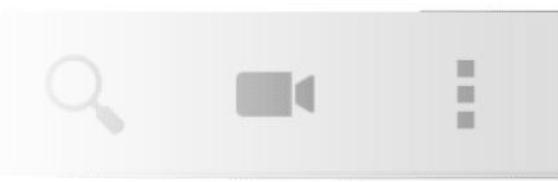
如果一个 action 适用 FIT，那就把它放到 action bar 上，否则就放入 action overflow。

将有含义的图形文字用于某些常见的 actions，比如“刷新”和“分享”。下面的下载链接是一个图标压缩包，里面提供多种屏幕密度以及 Holo dark、Holo light 主题的 action buttons 图标。除此之外还提供了一些无样式图标，你可以修改他们来匹配你的界面。请使用 Adobe® Illustrator® 来修改源文件。

[下载 action bar 图标](#)

Action overflow

Action bar 上的 Action overflow 用于放置一些不常用的 action buttons ,它只会在没有物理 menu 按键的设备上出现。那些有物理 menu 按钮的手机 按下 menu 按键就会打开 action overflow。



Action overflow 在 bar 的最右

那么多少 action bar 上放置多少 action buttons 才合适 ? 请参考以下规则 :

1.Action buttons 总占用的宽度不能超过 main action bar 的 50%。不过在 bottom bar 上 , action buttons 可以完全使用。

2.Action buttons 的数量取决于屏幕宽度像素大小 :

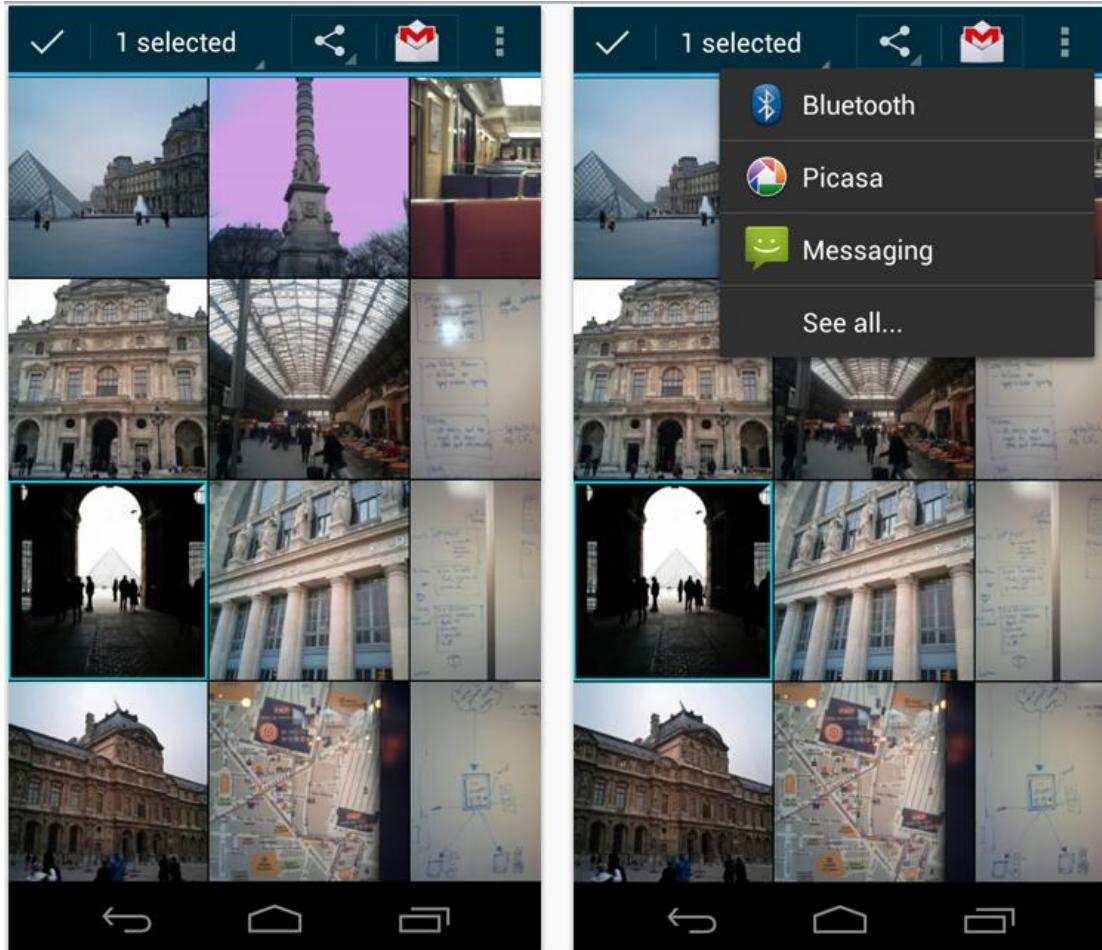
- a. 小于 360dp=2 个
- b. 360-499dp=3 个
- c. 500-599dp=4 个
- d. 大于 600dp=5 个

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=

其中 “o” 代表 action buttons , “=” 代表 action overflow

数据分享

如果应用程序允许分享数据，比如图像和电影，请在 main action bar 上使用 share action provider。Share action provider 是为了更快速地分析数据而设计，它提供了一个 spinner 用于放置多种分享选项。这些选项会根据使用频率自动排列。



(相册程序提供了分享机制，下拉条以及各种分享选项。)

Action bar 备忘清单

在设计 split action bar 的时候，请问问自己：

1. 视图导航最与任务来说有多重要？

如果视图导航在程序中处于很重要的地位，请使用 tabs(为了快速切换视图)或者 spinners。

2. 哪些 actions 需要直接放在 main action bar 上，那些需要移到 action overflow 里？

请参考上面的 FIT 规划来确定。

3. 有哪些重要的东西需要连续显示？

有时候程序需要一直显示一些重要的信息，比如未阅读的信息数目，又或者是当前播放的音乐。请认真设计哪些重要的信息是需要一直显示的。

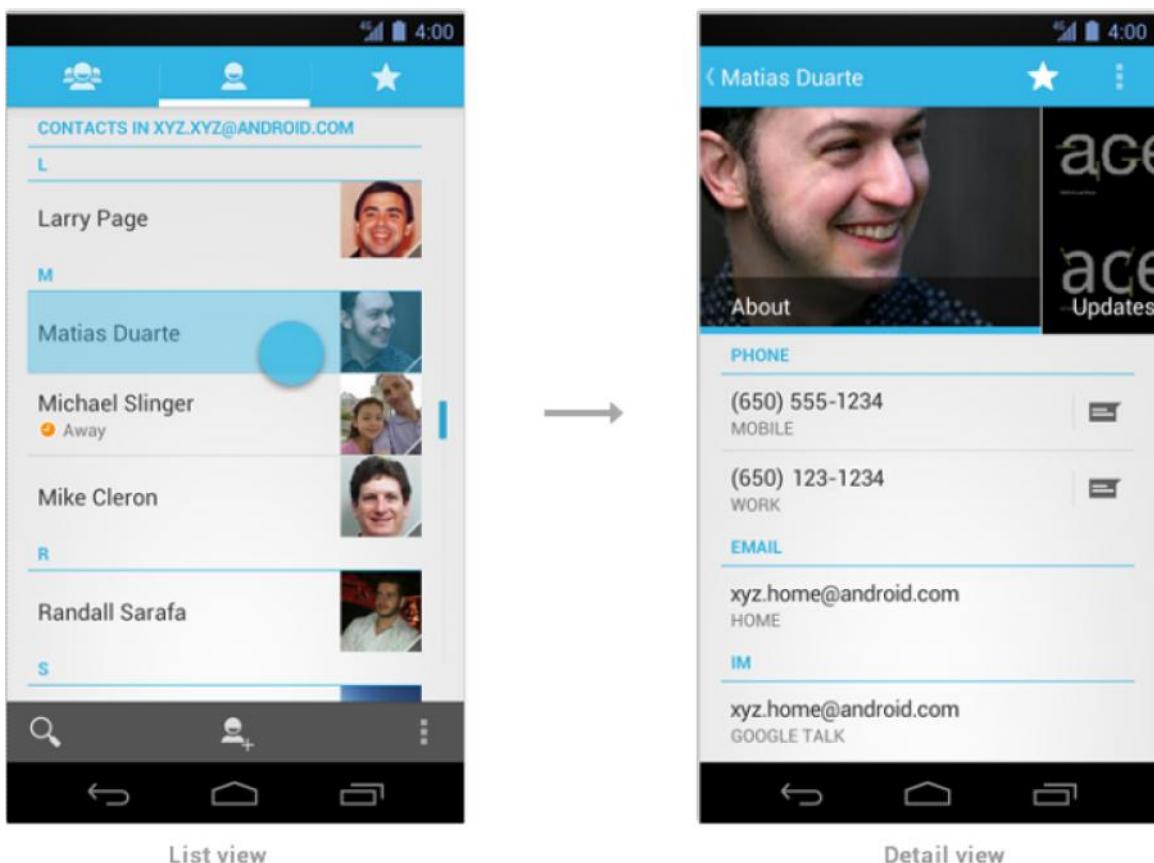
Multi-pane layouts-多面板布局

当你设计 Android 应用程序的时候 , 请始终记住你的应用程序需要适配各种尺寸各种类型的屏幕。确保你的程序能够提供一个美观合理的布局 , 并且能够适应不同的屏幕尺寸和设备的旋转方向。

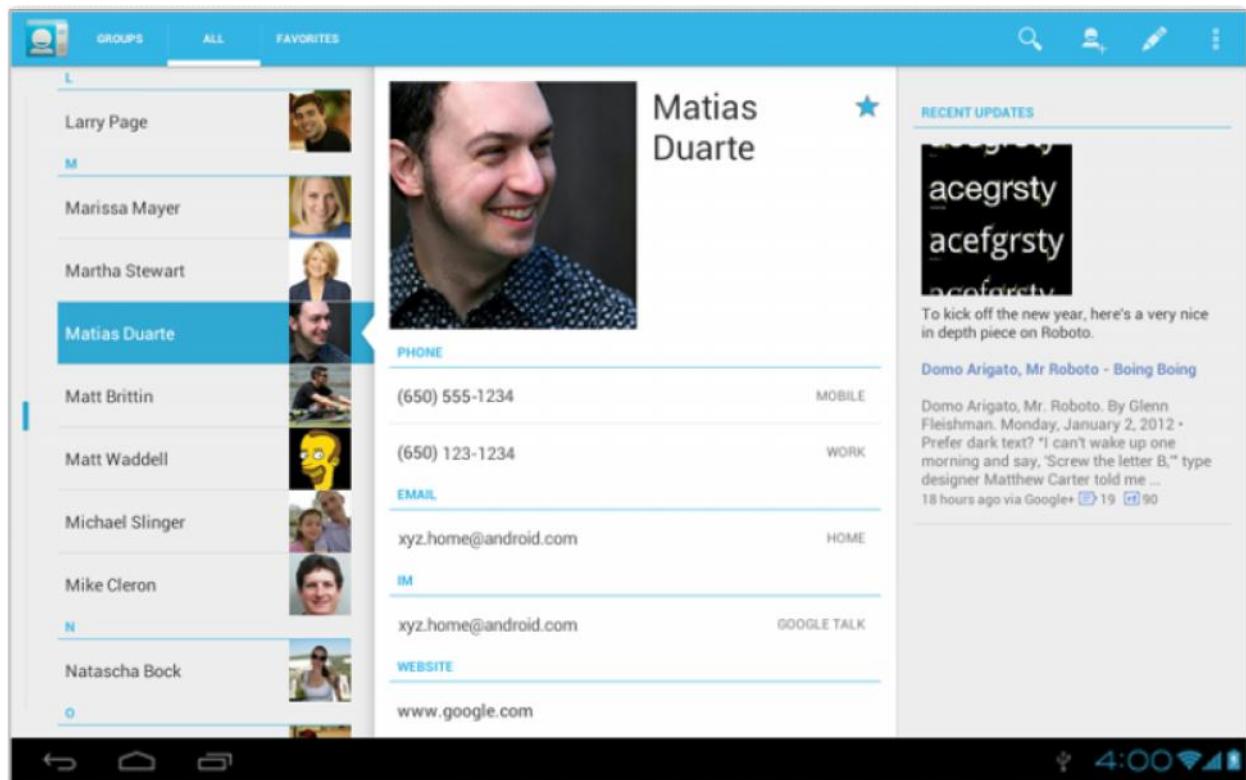
Multi-pane layouts 可以快速简单地实现这些。它允许你将多个视图结合在一起 , 当设备旋转到水平位置时 , 原来联合的视图又会在现有的屏幕区域分别显示出来。

多视图联合显示

在小屏幕设备上 , 应用程序的内容主要采用网格视图和列表视图进行显示。触摸一个项目然后进入不同的屏幕查看对应的数据和详细信息。



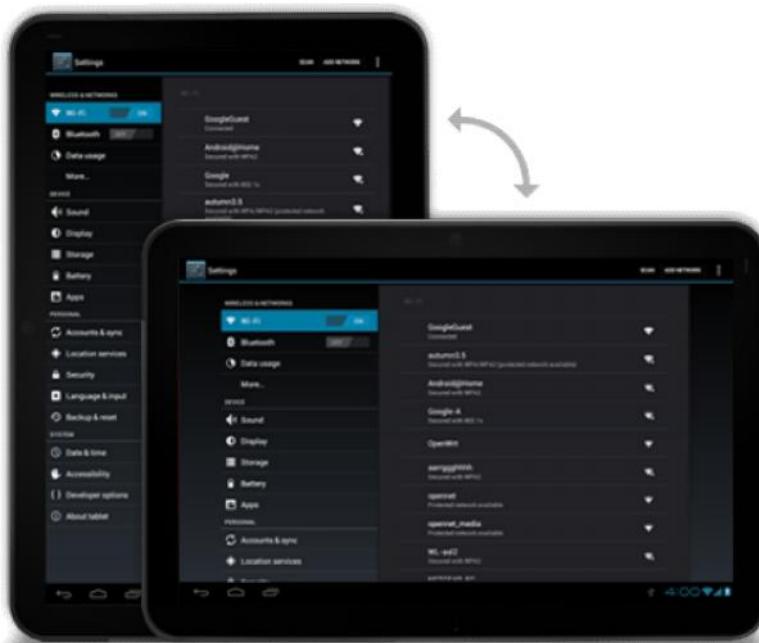
平板电脑比手机拥有更大的显示屏 , 所以你可以使用面包将相关列表视图和详细视图整合到一个复合的视图中。它能有效地利用多余的屏幕空间 , 让程序的导航变得更加简单。(接下页图)



一般来说，使用左边的面板显示分类或者数据条目等项目，右边的面板显示当前选中项目的详细信息。左面板的项目被选中时，请适当标注提示，这样保持左右面板之间的关系。（比如上图联系人程序中，选中的联系人用透明的蓝色条纹遮挡。）

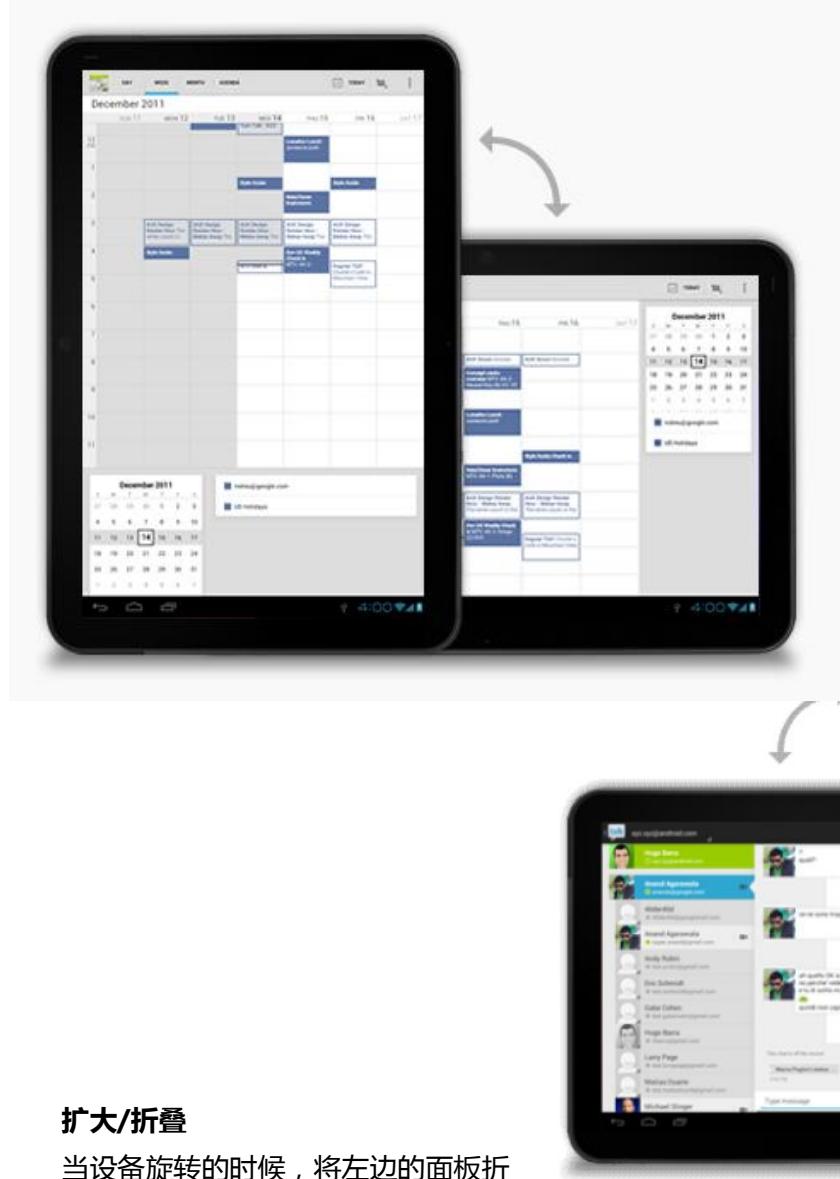
Compound Views 与方向旋转的关系

无论设备的屏幕旋转到何种方向，应用程序都应该有相同的功能。如果你设置在某个方向上使用 compound view，当用户旋转屏幕的时候也不要改变它们。下面是几条技术参考，帮助你调整布局旋转后的变化，同时保证功能上始终如一。



伸展/压缩

调整左面板的列宽，直到它与整个布局在各个方向上达到平衡状态。



堆叠

将各个面板重新组合排布以适应不同方向的屏幕。

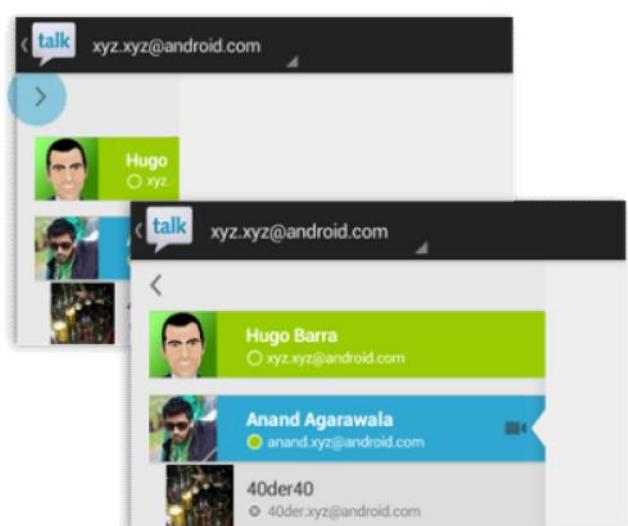
(日历程序中的各个面板在不同屏幕方向下排布也不一样。)



扩大/折叠

当设备旋转的时候，将左边的面板折叠且只显示最重要的信息。同时提供一个扩大按钮能够让左面板从折叠的状态中恢复过来。

(Gtalk 中提供了特殊按钮，可以将联系人列表自由折叠和还原。)





显示/隐藏

旋转设备后，右面板视图采用全屏显示。使用 action bar 上的 Up 按钮可以显示左面板，并且允许邮件之间的导航切换。当触摸详细信息面板上的内容时，左面板就自动隐藏起来。

(注意：目前 Gmail 的一些交互方式与上面提出的可能有出入。)

备忘清单

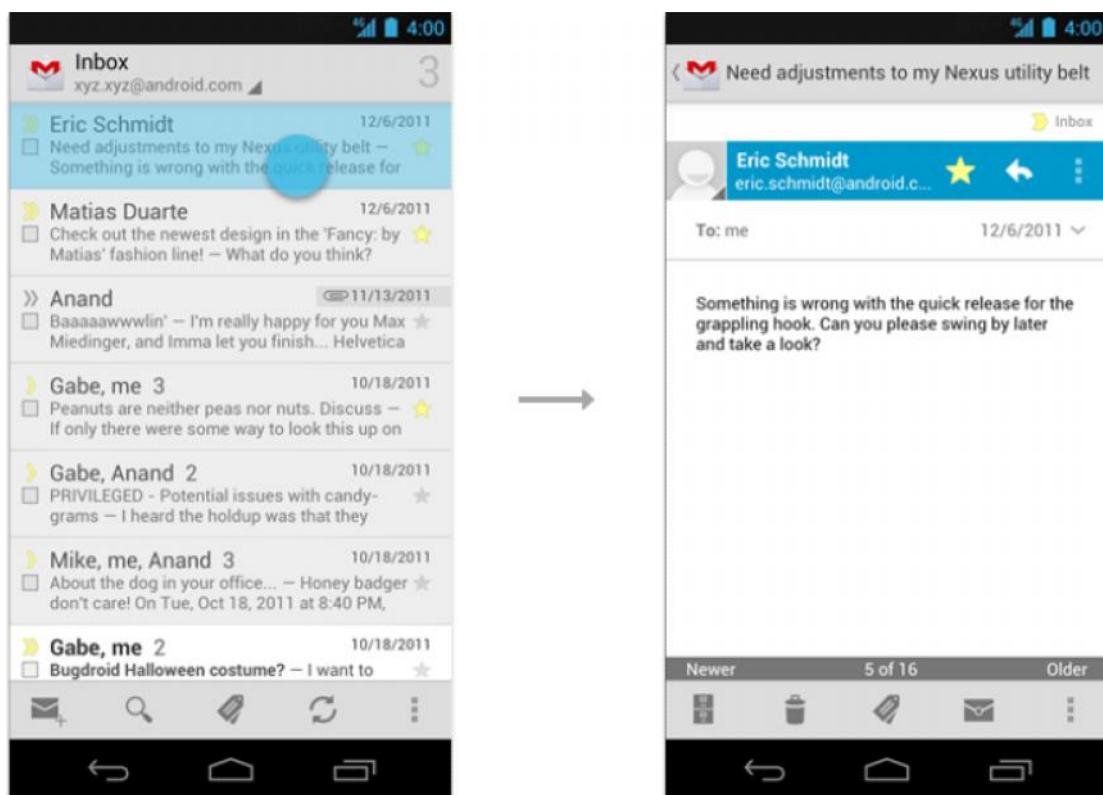
1. 请提前计划好如何让你的应用程序适应不同的屏幕大小和旋转方向。
2. 找出一种最合适的方法，让各个面板在屏幕方向改变时能够快速重新排布。
3. 寻找机会来巩固多面板复合视图。
4. 确保屏幕在旋转之后能提供功能校验。（— —||）

Swipe Views-滑动视图

高效的导航机制是出色应用程序必备的条件之一。虽然大多数的应用程序都是建立在层次上的，但是仍然有方法可以让垂直的层次结构使用水平导航。滑动视图允许用户通过简单的手势来进行切换，带来更舒畅的浏览体验。

Detail views 之间的滑动

应用程序的数据通常是通过主从关系而联系起来的：用户可以看到一个相关的数据列表，例如图像、聊天或邮件，然后点击其中任意一个项目就可以在新窗口中查看更详细的内容。

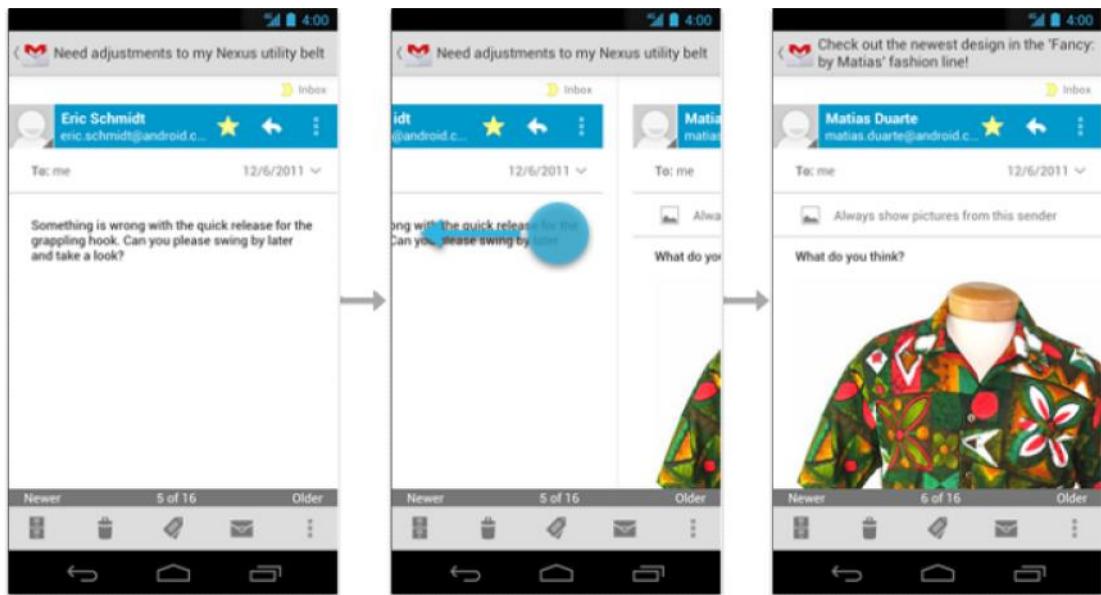


主要视图（左）和详情视图（右）

在手机上，主要内容和细节是在不同的屏幕上显示的，所以这就要求用户经常在主要内容列表和详细视图间频繁切换，又名“pogo-sticking”。

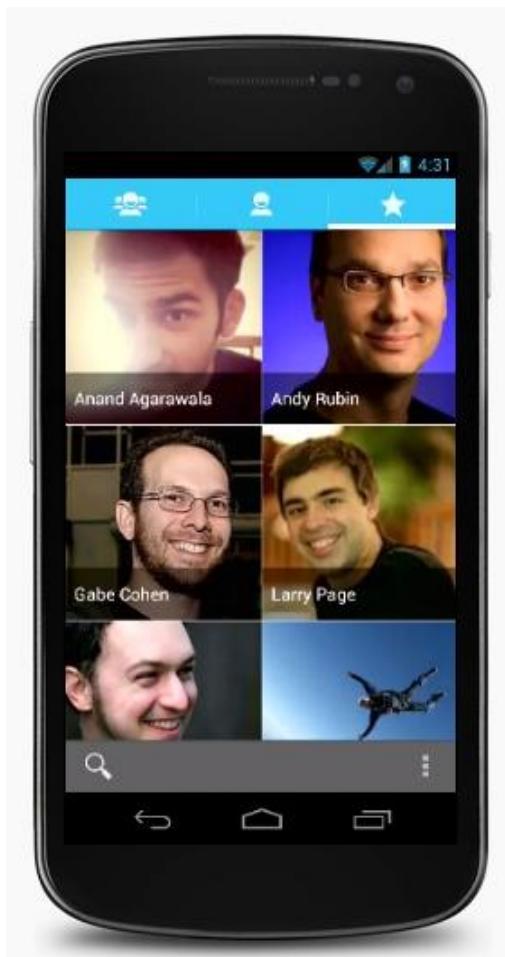
当用户想要查看多个连续视图的时候，使用滑动手势来进行详情视图的导航，从而避免了“pogo-sticking”。

（接下页图）



采用滑动手势的方式来进行连续邮件之间的切换导航。

Tabs 间的滑动操作



如果你的应用程序使用了 bar tabs，那就采用滑动手势来切换不同的视图。

备忘清单

1. 在 detail views 和 tabs 上使用手势导航。
2. 当手势启用的时候就视图就开始进行切换 不要等到手势结束的时候才开始。
3. 如果你使用了上一步/下一步按钮，请用滑动手势来替代他们。
4. 可以考虑在 detail views 上添加一些有关联的信息，这样可以告知用户当前可视项目在列表中的相对位置。

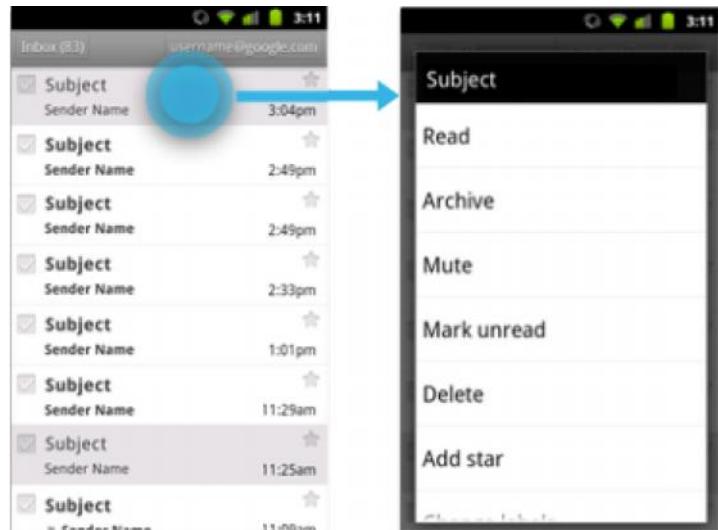
Selection-选择操作

Android3.0 引入了全新的长按手势：当触控在同一地方保持一段时间后，就可以全局地进行数据选择。这样或许会影响到你在应用程序中的多选操作等其他一系列操作。

哪些改变了？

在之前版本的 Android 系统中，长按手势普遍用来显示一个包含各种操作的 contextual menu，并且只能对单个数据项目进行操作。

而这种机制在 Android3.0 中发生了改变：长按手势用于选择数据，然后会出现 contextual action bar (CAB)。CAB 不仅能够提供相关操作，而且能够对选中的数据进行管理。



传统长按手势操作方式启动 contextual menu

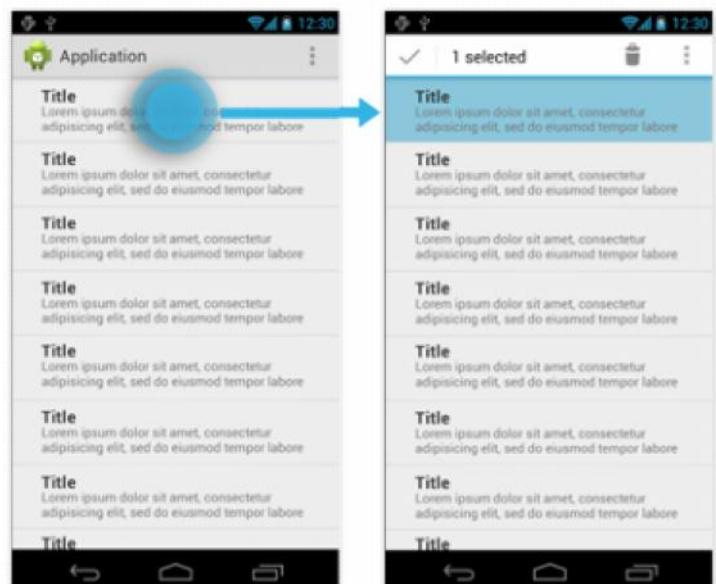
使用 contextual action bar (CAB)

当有数据被选中时，CAB 就会临时性遮挡 main action bar。长按一个数据项目就会出现 CAB。



用户可以：

- 1.一次选择多个数据项目。
- 2.可对选中的多个数据进行统一的操作，然后 CAB 自动消失。
- 3.点击 navigation bar 的 back 按钮或者 CAB 的确认按钮就可以取消 CAB。同时取消已选中的数据条目。

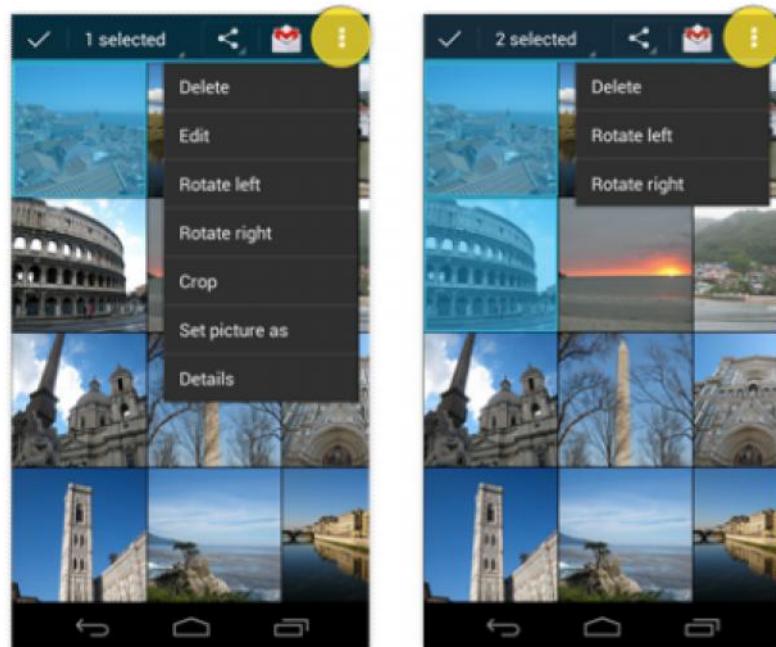


选择 CAB 上的 actions

你可以决定哪些元素和 actions 显示在 CAB 上。请参考 [Action Bar](#) 相关指导来决定哪些应该处于 top level view 哪些处于 action overflow。

动态地调整 CAB 上的 actions

由于用户可能会选择多个数据项目，所以 CAB 上的 actions 也需要根据所选数自动态地进行调整。因为有些 actions 不适用于多个数据操作。



备忘清单

1. 如果你的应用程序支持多数据选择，请使用 contextual action bar。
2. 保留长按手势进行数据选择，不要使用传统的 contextual menu。
3. 如果出现不支持多选操作，长按操作不要产生任何 actions。
4. 参考 main action bar 上 actions 的排布方法。

Notifications-通知系统

通知系统将重要的事项告知用户，比如聊天程序和日历的新消息。

为了创建一个体验出色的应用程序，设计好通知系统是一件非常重要的工作。通知代表了应用程序的“声音”，让程序更具个性化。不重要的通知千万不要发送给用户。

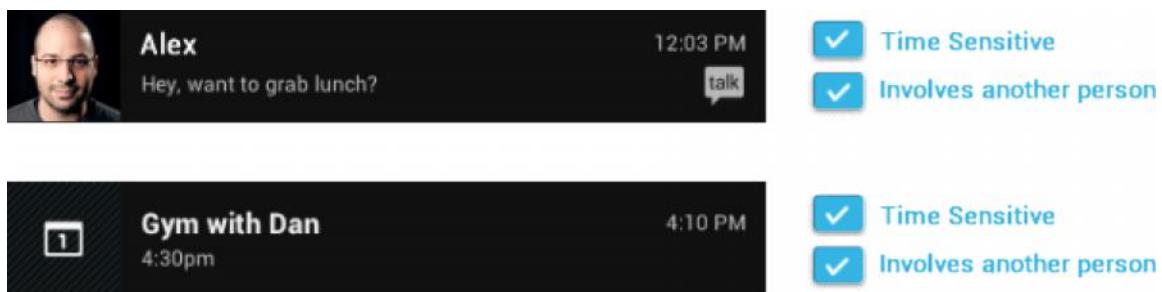
什么时候显示通知

为创建一个让人喜欢的程序，保证用户的关注不被打断是一件非常重要的事情。使用一个比喻可能引起与软件开发商的共鸣，用户不是一个可以调用的返回值的方法。用户的重点是资源更类似于一个线程，并创建一个通知，暂时阻止它们处理用户线程，然后取消打扰性的通知。

Android 通知系统能够将重要的事情快速通知给专心于其他的用户。

尽管一个优秀的应用程序只在该通知的时候发出通知，但有时候难免需要用不必要的通知打断用户。

通知应该用于对时间比较敏感的时间，特别是如果这些事件会涉及到其他人和事物。例如，一个实时的聊天请求和同步的信息交换：另一个用户在等待你的应答。日历事件是另一个很好例子：使用一个通知吸引用户的注意力，告知用户一些日程即将到来。而这些日程往往和其他事物有联系的。



什么时候不显示通知

遇到以下情况请不要使用通知：

- 1.与用户无关的，时间不敏感的不要通知。比如你的 facebook 程序网络发生异常，则不需要告知用户。
- 2.如果新消息已经显示在当前屏幕上，就不要在创建通知。使用应用程序本身界面来显示新通知。比如你正在使用 Gtalk 聊天，并且正处于聊天界面，那新的通知就直接显示在界面上而不要在通知系统产生通知。
- 3.低级的操作提示不要发送通知，比如数据的同步、更新。（不过目前很多程序都是对着干）
- 4.如果一些错误信息不需要用户进行确认，那就不要通知给用户。
- 5.不要使用通知来提示 services 的启动和暂停。
- 6.不要为了展示你的品牌而发送不必要的通知，它只会挫败和疏远你的用户。如果想要为用户提供少量更新信息，并且能与用户进行交互操作，最好的方法就是开发桌面插件（AppWidget），让用户在主屏幕上自由放置。

设计指导

个性化

如果是别人发过来的信息，请在通知中加入人的照片。同时还需要添加上应用程序的图标作为副图标，这样用户就可以知道是哪个程序推送的通知。



添加导航

当用户点击一条通知，就会跳转到通知内容所在的应用程序，在大多数情况下往往会跳转到 detail views (比如 Gtalk 聊天窗口)。但有时候通知信息是重叠的 (下面会提到) 包含了多条信息。这时候当用户点击通知后就会跳转到 detail views 的上一层和 top level 的下一层——categories view。所以在这里你需要添加一个 Up 按钮，让用户能够返回到 top level (程序主页面)。更多信息请参考[导航](#)设计模式

给时间敏感的事项加上时间戳

标准的 Android 通知默认将通知时间显示在每条通知的右上角。当然你也考虑时间戳对你程序的通知是否有价值。如果时间戳对你来说没有价值，那你就要考虑这条通知的重要程度能否引起用户的注意。如果这个通知足够重要，考虑是否取消时间戳的显示。

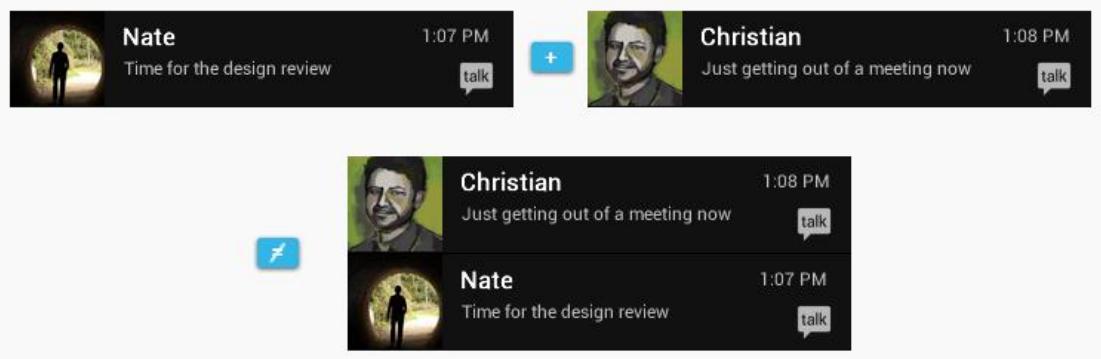
如果用户想要知道通知的时间，请添加时间戳。通信通知 (邮件、消息、聊天、语音) 适合使用时间戳，用户可以根据通知的时间来理解相关信息或进行下一步操作。

折叠你的通知

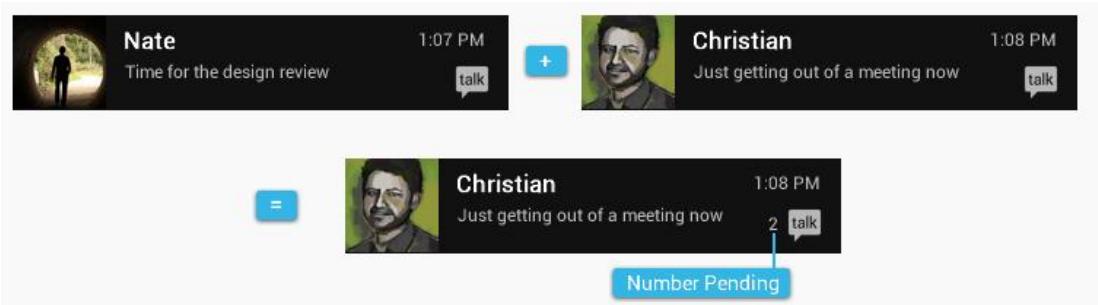
如果应用程序发送通知时，通知中心已经存在了相同的通知，那就不要再创建新的通知，而把相同的通知折叠起来。

折叠后的通知会对所有通知进行概要描述，同时告知用户有多少相同的通知。

不要这么做：



应该这样做：



如果概要和详情处于不同的视图，那点击折叠通知打开的应该是其他地方（categories views），而不是通知内容指向的 detail views。

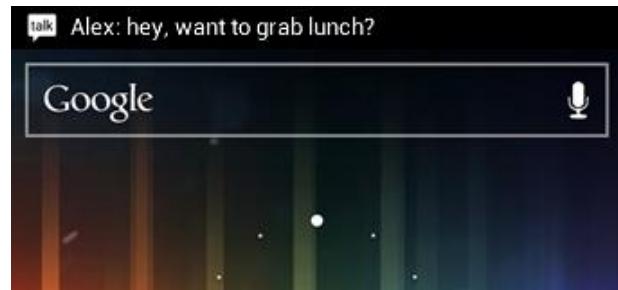
例如，点击单个邮件通知直接打开邮件的详细内容，但是点击折叠的邮件通知（包含多个邮件）则会打开收件箱。

自动清理

比如日历日程，通知会告知用户一个事件将要在特定的时间开始。当过了这个时间之后，这条通知就显得不那么重要，这时候它应该自动地消失。聊天对话和语言邮箱信息也是如此，完成对话后，通知应该自动地消失而不需要用户手动去清理。

快速查看通知

当收到通知的时候（比如短信），可以在 status bar 上用一段段文字显示通知的主要内容。显示完毕后文字自动隐藏。



通知应该是可选的

用户经常会收到各种通知，这难免会给用户造成困扰。所以在应用程序中，允许用户能够通过设置来关闭/开启通知消息。

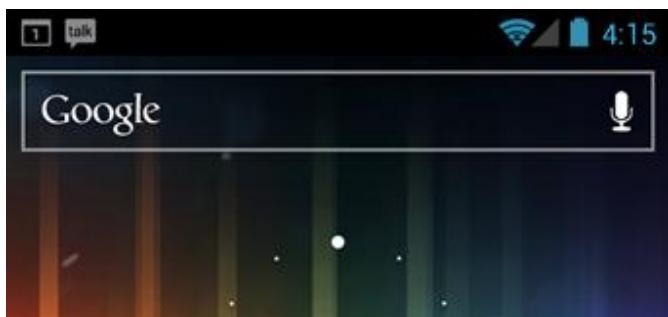
使用显而易见的图标

当用户查看通知中心的时候，他们需要分辨出通知是来自哪个应用程序的。Pictures are faster than words !!

建议这么做：参照应用程序的图标来创建一个对于的通知图标。

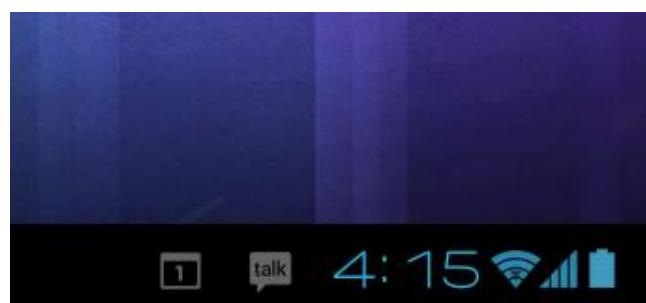
不要这样做：使用不同的颜色来区分。因为通知图标要求是单色的。（白色）

通知的交互模式



通知伴随图标显示在通知中心，可以下拉状态栏查看。在通知中心里，各通知消息按照时间进行排序，时间最靠前的排在顶部。点击任意通知就可以进行对应程序查看通知详细信息。左/右滑动一条通知就可以移除通知。

在平板电脑上（Android3.X），通知区域与 system bar 结合在一起，置于屏幕的底部。点击通知区域即可打开通知中心。



进行中的通知

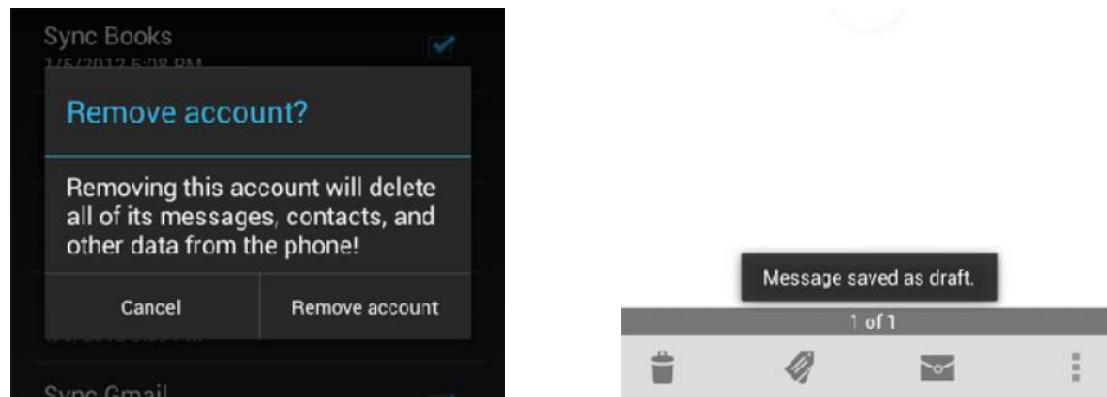
进行中的通知是为了告知用户某个进程还在后台运行着。比如，音乐播放器会在通知中心显示一条通知来告知用户当前正在播放的音乐，而且用户可以在这里控制音乐的暂停/播放。它们还可以被用来提示文件下载、编码视频等一些用时较长的任务。



进行中的通知不可以被移除。

Dialogs 和 toasts 不是通知而是用于反馈

在合适的情况下使用 dialogs 和 toasts，他们应该用于对用户的一些操作做出快速的响应和反馈。比如，dialogs 用来确认用户是否明白某个操作的重要性，toasts 可以反馈一个操作是否正确执行。



Compatibility-兼容性

Android3.0 中的显著变化：

1. 取消了传统物理按钮 (back、menu、search、home) , 采用了虚拟的 navigation bar 来取代之 (back、home、recents)。
2. 将 menu 依附于 action bars。

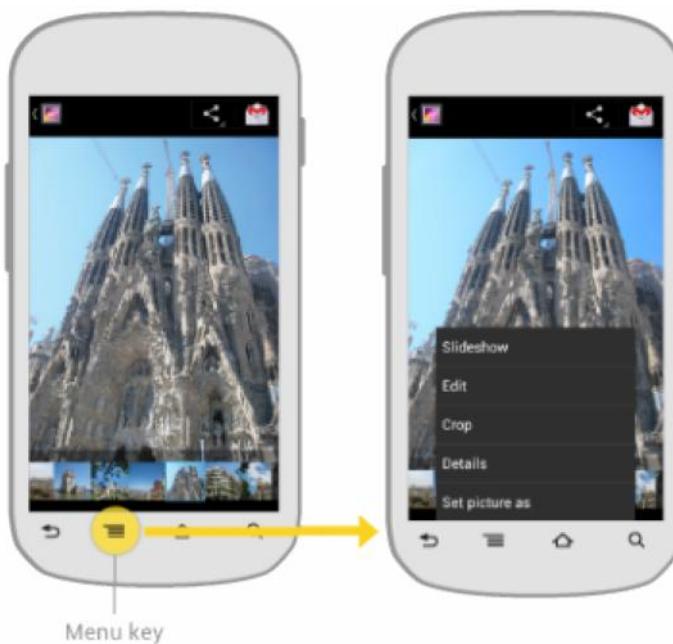
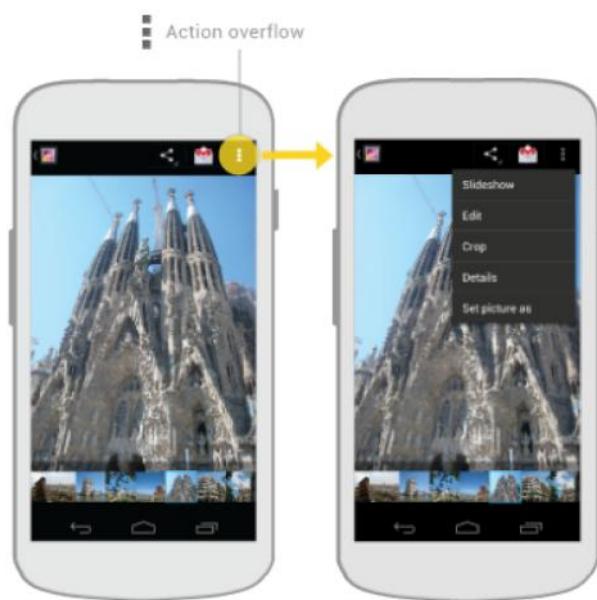
Android4.0 将这些特性从平板电脑带到了手机。

让老式硬件和应用程序适应

Android4.0

手机上的虚拟 navigation bar

为 Android3.0 及以上系统设计的程序都将 actions 放在 action bar 上。如果 actions 太多导致 action bar 无法显示，就将多余的、相对不重要的 actions 放在 action overflow 里。

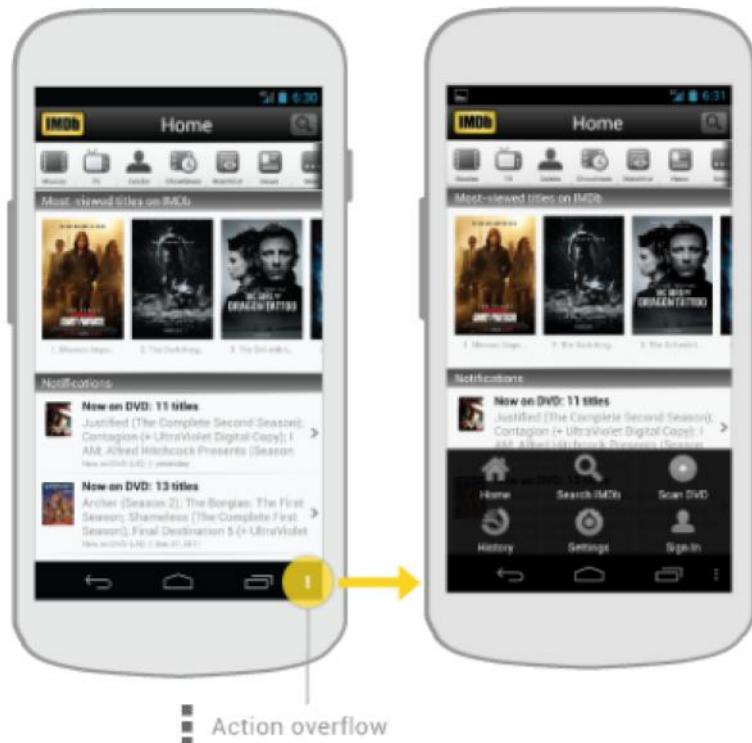


带有物理导航按钮的手机

在带有传统物理按钮的 Android 手机上，虚拟的 navigation bar 不需要显示，同时用 menu 按钮来代替 action overflow 的操作。操作的结果都是一致的，没有太大的变化，唯一不同的就是 action overflow 的内容是显示在屏幕的底部。

让虚拟 navigation bar 的手机适应旧的应用程序

当我们在只有虚拟 navigation bar 手机上，使用一个为 Android2.3 以及之前版本而设计的应用程序，一个虚拟出来的 action overflow 就会显示在 navigation bar 的最右侧。它的作用就是来弥补 menu 按键的缺失。



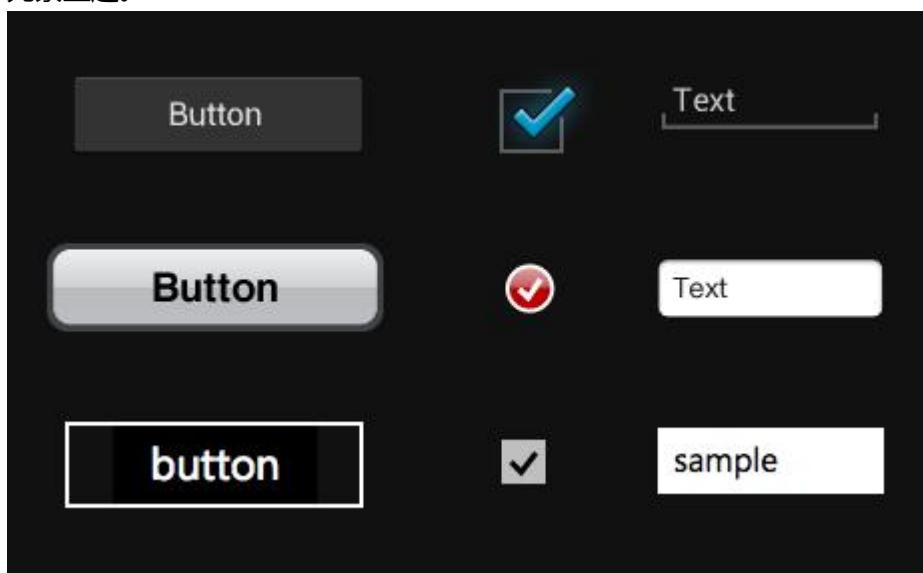
Pure Android-纯粹 Android

许多开发者想让应用程序跨平台。当你设计 Android 程序时，一定要记住每个平台都有不同的设计规则和惯例。也许一个应用程序在一个平台的表现很好，但是拿到另外一个平台也许就会表现得很差。“统一设计，多平台适配”，这样的方法也许会节省时间，但是可能会导致产品在平台上表现不一致，不符合当前平台的特性，影响用户体验。请参考一下设计知道，避免进入误区。

不要模仿其他平台的 UI

一个平台通常提供了适合该平台的 UI 元素和特定的主题。例如，一些平台提倡使用圆角按钮，而其他平台则建议将按钮设计得有梯度。在某些情况下，这些元素包含了相同的目的，但是在实际使用上可能会存在差异。

当设计 Android 应用程序时，不要在意其他平台的 UI 元素和交互模式。参考[控件](#)这一章节，了解 Android 基本控件样式和默认主题风格。同时可以参考 Android 平台现有的一些程序。如果你想自定义 UI 控件主题元素，请根据你的程序进行定制，千万不要参考其他平台的 UI 元素主题。



不要使用其他平台样式的按钮图标

每个平台通常提供了一套适用于该平台的图标，例如分享、搜索、新建或删除按钮。

如果你正在将其他平台的应用程序移植到 Android 上，请使用 Android 平台提供的素材。



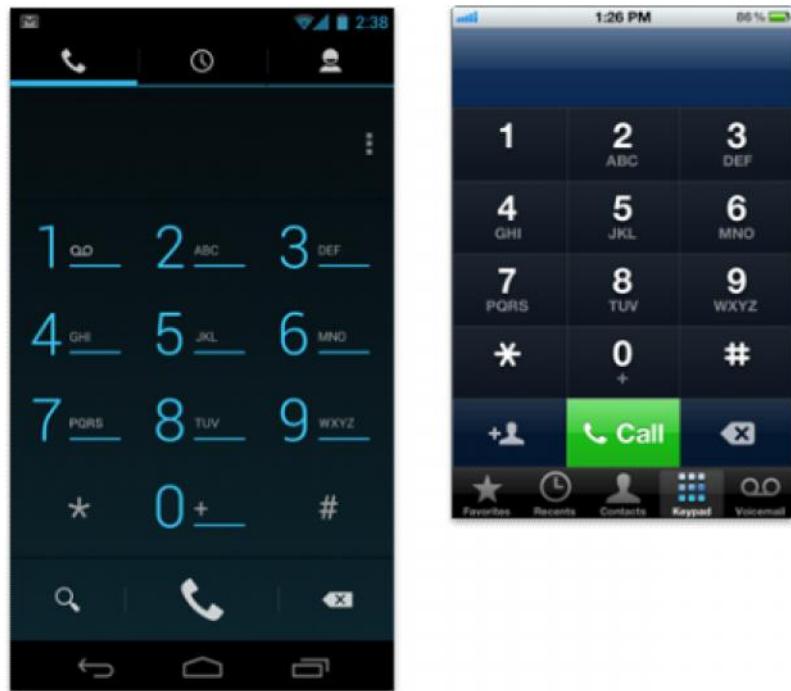
你可以在[下载页面](#)找到包含各种图标的压缩包。

不要使用底部标签栏

iOS 平台采用底部标签栏带进行视图切换。而每个平台都有自己的惯例，Android 的标签栏则处于屏幕的上方。其次，Android 程序中可能使用 split action bar 来显示一些重要的 actions。

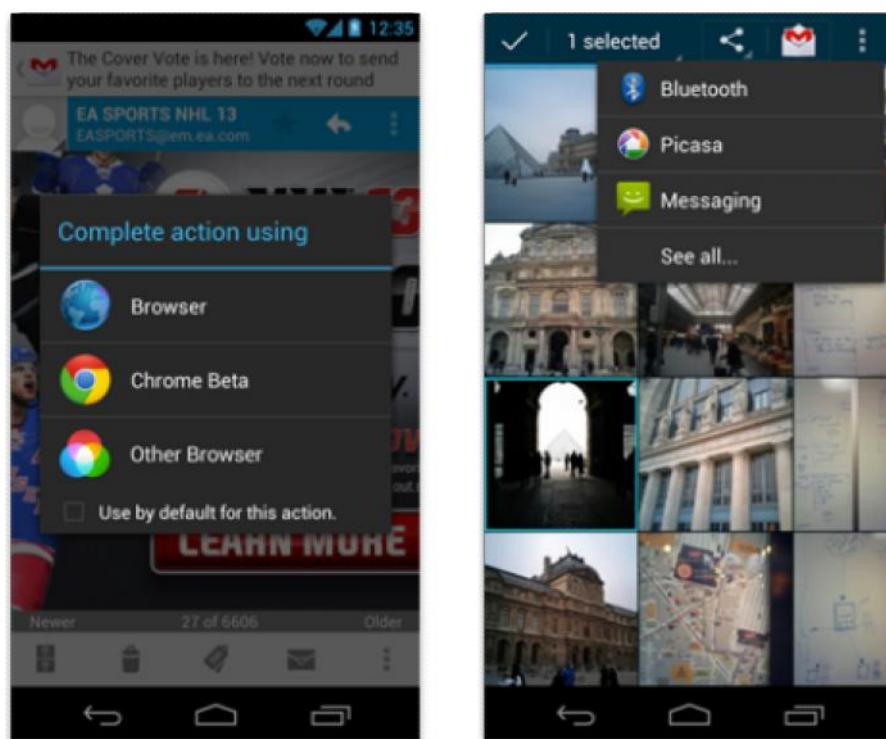
你应该遵循 Android 平台的设计指导来创建应用程序，这样可以保证应用程序在该平台上具有出色的体验，避免视图和操作的混乱。

更多信息请参考 [Action Bars.](#)



不要使用生硬的文字链接

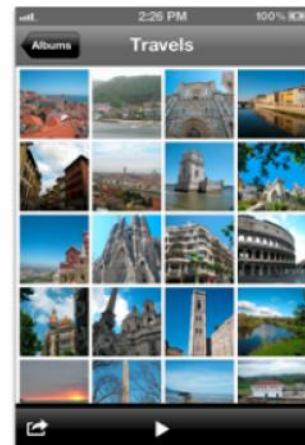
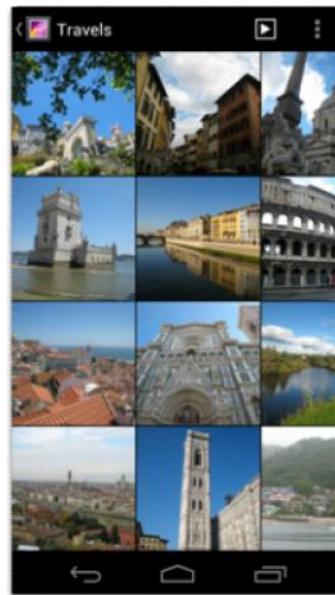
某些时候，你应用程序可能需要和其他程序进行数据交换，例如，你可能想要分享社交程序上的消息或者是分享一个网页链接。请不要使用生硬的文字、链接来实现，而要使用 Android 系统的 intent API 来加载一个可用程序列表，然后选择对应的程序。这样可以让用户通过他们喜欢的程序来完成任务。如果是分享操作，请考虑使用 Share Action Provider 来提供快速的分享方式。



不要在 action bars 上使用有标签的返回按钮

其他平台使用一个带有文字标签的返回按钮，用于明确导航层次结构。而在 Android 应用程序中主要使用 action bar 上带有图标 的返回按钮 和 navigation bar 上你虚拟 Back 按钮。

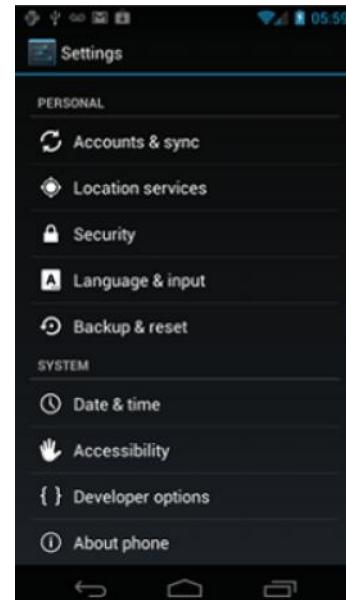
更多信息请参考[导航模式](#)。



不要使用线项目指向右边的符号标识

在其他平台上，常见的模式是右指向插入符的行项目，允许用户更深入更多的内容。

Android 不会使用向下钻取线项目等指标。避免他们与平台保持一致，以不有用户猜测到这些符号标识的含义可能是什么。



设备的独立性

记住你的应用程序将会运行在多种尺寸屏幕的设备上，请使用多面板布局等等方法为不同大小、密度的屏幕设计对应的布局。

更多信息请参考：[设备与显示、多面板布局](#)

Building Blocks-系统控件



用现成的 UI 元素创建一个出色的应用程序

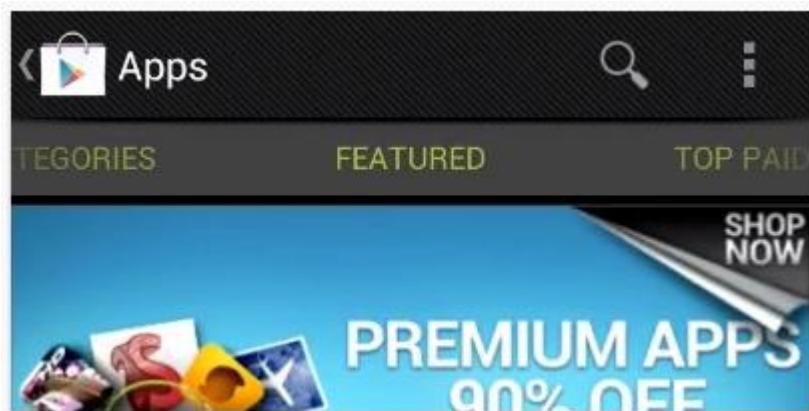
Tabs-标签



Action Bars 上的 tabs 能容易地发现和切换程序的多个视图和功能，也可以浏览不同分类的数据。

Scrollable Tabs

Scrollable Tabs 比标准的 tabs 能够容纳更多条目。左/右滑动视图可进行切换。



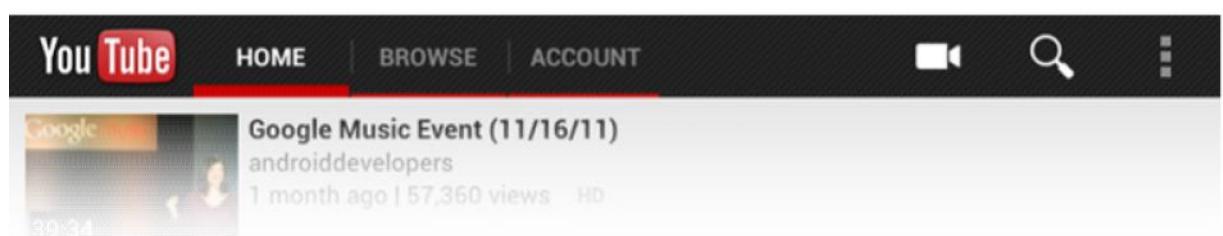
(Play 商店采用了 Scrolling Tabs)

Fixed Tabs

Fixed Tabs 同时显示所有项目，点击 tab 切换对应视图。



(Holo dark 和 Holo light 主题下的 Fixed Tabs)



(Youtube 中的 Fixed Tabs)

堆叠标签

必要的时候，你可以将 Fixed Tabs 和 action bars 结合在一起显示。比如旋转屏幕的时候，为了节省内容区域，直接把 Fixed Tabs 和 action bars 联合。



Lists-列表

列表用于在竖直方向上显示多行项目，它们可用于数据现在和滚动导航。

<p>1 SINGLE LINE LIST</p> <hr/> <p>List item number one</p> <hr/> <p>Second list item</p> <hr/> <p>This is the third item</p>	<p>1.分栏</p> <p>使用分栏将列表内容划分为多个分组。</p>
<p>2 2 LINE LIST</p> <hr/> <p>2-Line List Austin mixtape Cosby sweater butcher. Fixie ad vice, brooklyn...</p> <hr/> <p>Second list item Assumenda commodo laborum accusamus</p>	<p>2.列表项</p> <p>列表项在不同的安排下可以适应不同的数据类型，包括单行、多行项目、icons、checkboxes 和 action buttons。</p>
<p>3 3 LINE LIST</p> <hr/> <p>Three line list title Put a bird on it qui fanny pack, portland irony nisi fap irure. Donec hendrerit elit nec ligula dapibus</p> <hr/> <p>Second row in list Vinyl laboris lo-fi ethical, adipisicing assumenda beard. Curabitur gravida quam id orci sodales</p>	

Grid Lists-网格列表

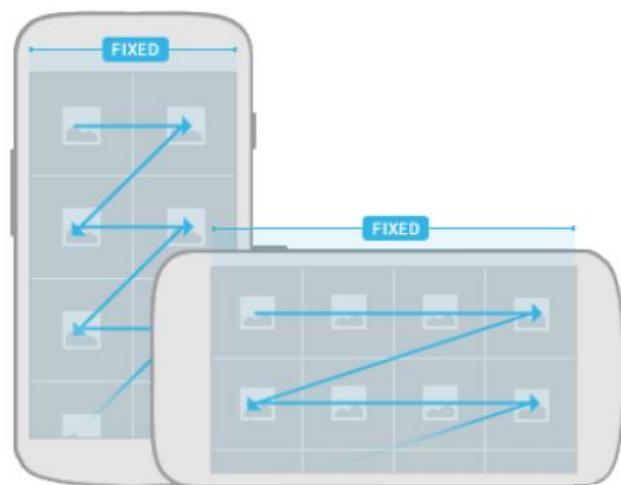


Grid Lists 是标准 list views 的替代，它们最适合显示图片类数据。与简单的 lists 相比，grid lists 要么垂直滚动，要么水平滚动。

一般的网格

网格列表中的数据项被安排在两个方向，其中一个在滚动的时候是固定不动的。滚动的方向决定了网格列表内的项目排布。因为滚动方向是不确定的，所以切断网格内容（在屏幕边缘显示一部分内容）来表明其他内容的位置，用户就能决定往哪个方向滚动。

不要创建在两个方向上都能滚动的网格列表。

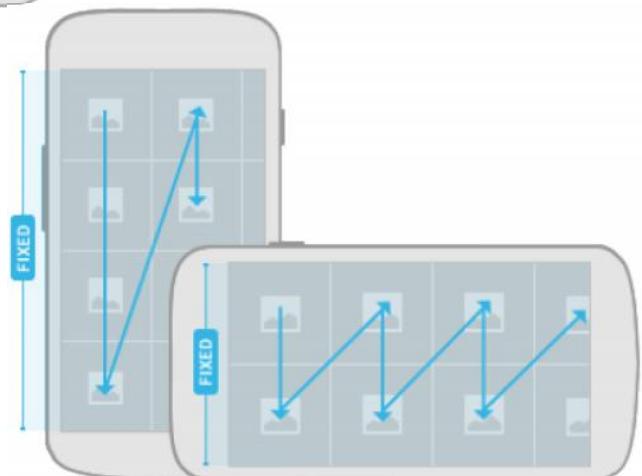


垂直滚动

垂直滚动的网格列表项目按照西式的阅读习惯来排布：从左到右，从上到下。显示列表的时候，将项目的一部分切断来告知用户向下滚动列表可以显示更多项目。在用户旋转屏幕后也要使用这种方式。

水平滚动

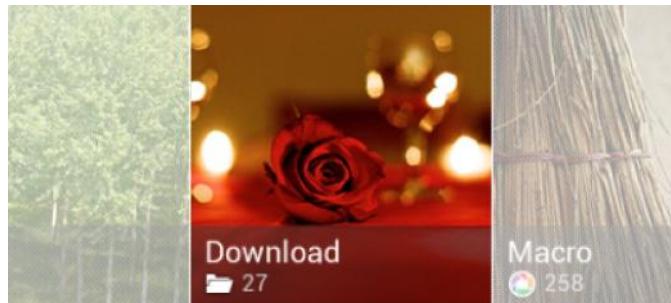
水平滚动列表在网格的垂直方向上是固定的。和垂直滚动列表相比项目的排布方式发生了变化：从上倒下，从左到右。同样使用切割项目的方法来表明还可以滚动的方向。



Scrolling tabs 不能与水平滚动网格列表同时使用，因为水平的切换视图的导航手势会与查看更多手势产生冲突。如果你想使用 scrolling tabs 来进行导航，那就使用垂直滚动的网格列表。

带标签的网格列表

在网格视图的项目上使用标签来显示额外相关的信息。



风格

在网格列表上层用一块半透明的面板来显示标签。这允许你控制对比度，保证标签更有“穿透性”，更容易被用户识别。

Scrolling-滚动

滚动允许用户使用滑动手势来查看更多的内容，滚动的速度和手势滑动的速度成正比。

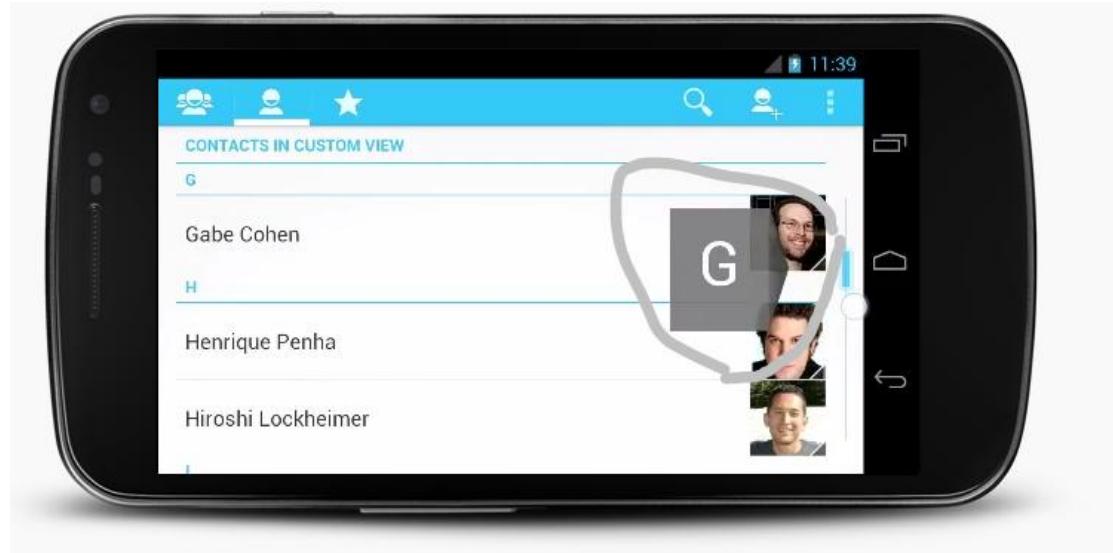
滚动指示

滚动的时候，滚动指示器会出现，以便告知用户当前处于视图的哪个位置。



索引滚动

除了传统的滚动外，一个较长的、按照字母顺序排列的列表使用了索引滚动：一种快速导航到特定项目的方法，以特定的字母开始。使用索引滚动时，滚动提示器（滚动条）会一直显示，即使它是静止的。点击或者拖动索引滚动提示器（滚动条），就可以放大显示相应字母。



Spinners-下拉列表

Spinners 能够让用户快速地从一列内容中选择一个值。默认状态下，spinner 会显示当前选中的值。点击 spinner 会弹出一个包含很多可选值的下拉菜单，用户可以从中选择新的值。

表格中的 spinners

Spinners 适用于表格中的数据选择，它们可以很好地与其他控件进行融合。表格的 spinners 可以输入简单的数据，同时也可以结合其他输入控件来使用。例如，一个文本域可以让你编辑一个联系人的邮件地址，而与之关联的 spinner 允许你选择是家庭还是工作地址。

jay@gmail.com

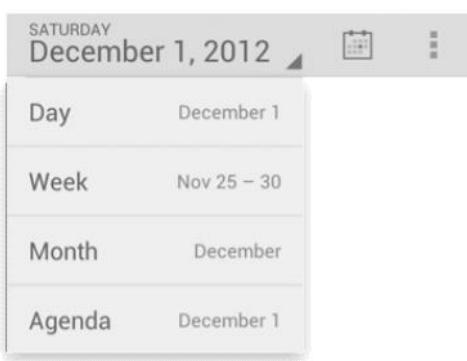
Home

Home

Work

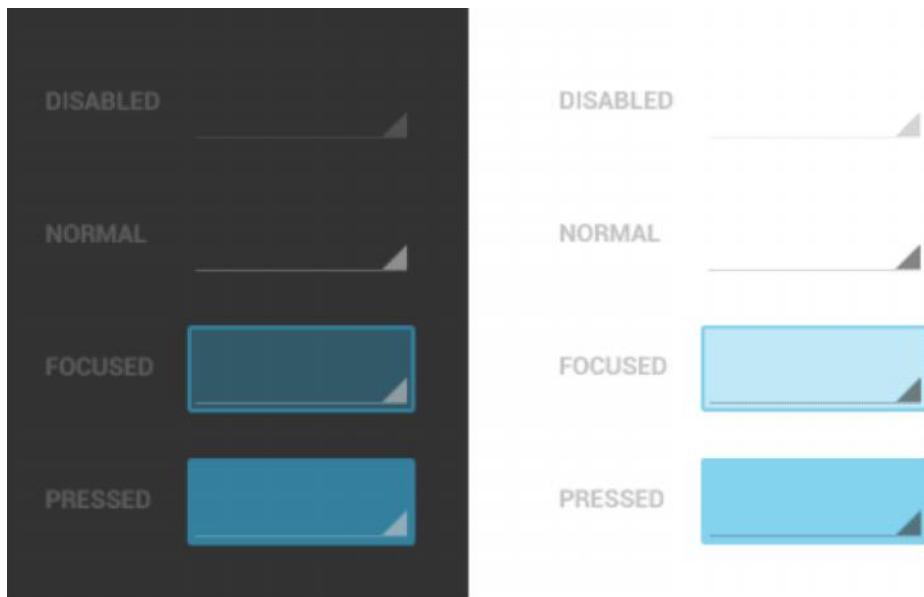
Other

Custom



Action bars 上的 spinners

在 action bars 上使用 spinner 来切换视图。例如，Gmail 使用一个 spinner 来实现账户和最近使用标签的视图切换。Spinners 在切换重要视图的时候会有很大的作用，但是这不一定经常发生。如果需要频繁的切换视图，请使用 tabs。



(Holo dark 和 Holo light 主题下的 spinners)

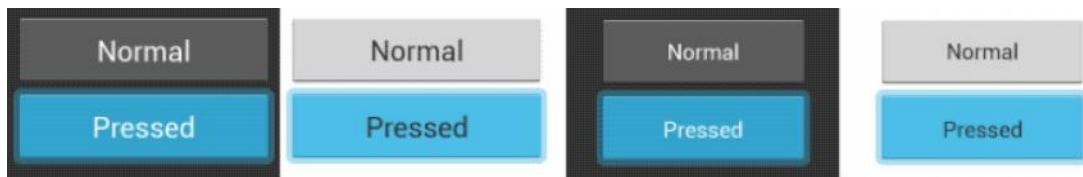
Buttons-按钮

一个带有文字和(或)图片的按钮能够清楚地表达出按下按钮后会产生的 actions。Android 支持两种类型的按钮：基础按钮和无边框按钮。两种按钮都能可以使用文本和(或)图片。



基础按钮

基础按钮是一种传统的带线框和背景的按钮。Android 支持两种风格的基础按钮：默认的和小的。默认的按钮有稍大的字体，用于优化表格内容的显示。小的按钮需要和其他的内容一同显示。它们有更小的字体和更小的高度。要使用小的按钮，你需要让它和表格中的其他 UI 元素对其。



(Holo dark 和 Holo light 主题下的默认按钮和小按钮)

无边框按钮

除了没有边框和背景，无边框按钮在其他方面和基础按钮都是一样的。你同意可以在无边框按钮中使用文本和图标。无边框按钮在视觉上比基础按钮更加轻量级，并且能很好地融合与其他内容。

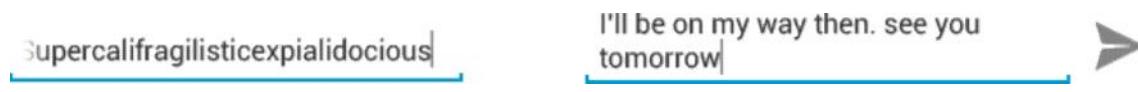


Text Fileds-文本域

应用程序中的文本域允许用户输入文本信息，可以单行使用和可以多行使用。点击文本域后显示光标，然后键盘自动显示。除了输入之外，文本域还允许其他各种操作，比如文本选择（剪切、复制、粘贴）以及数据自动填充。

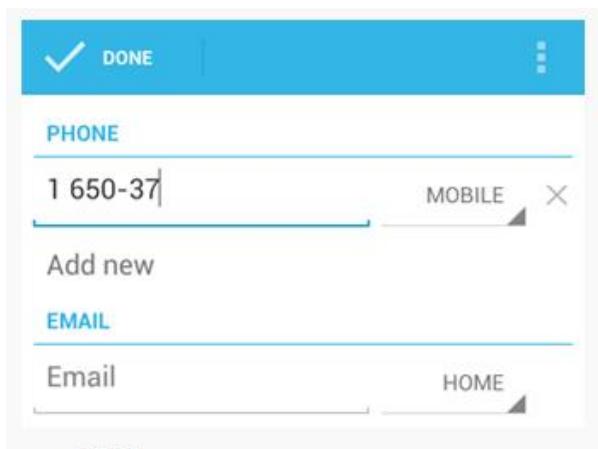
单行和多行

当单行文本域输入数据到达最右侧时，文本数据会自动像左滚动。多行文本域则会另起一行来显示文本，当光标处于较低的位置时会垂直滚动。



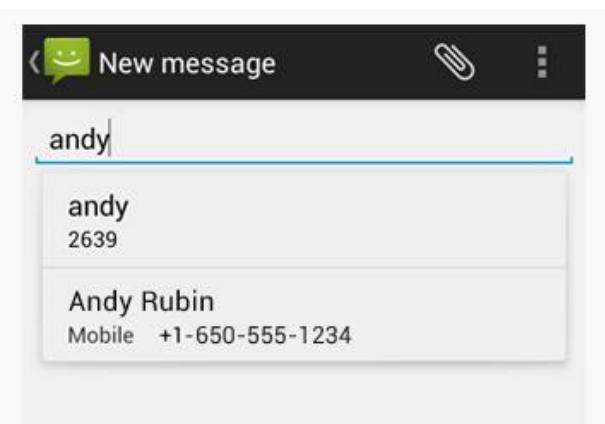
文本域类型

文本域包含很多种类型，比如数字、消息、或者邮件地址。类型决定了什么样的字符应该出现在文本域中，可促使键盘来优化它的布局，方便用户输入。



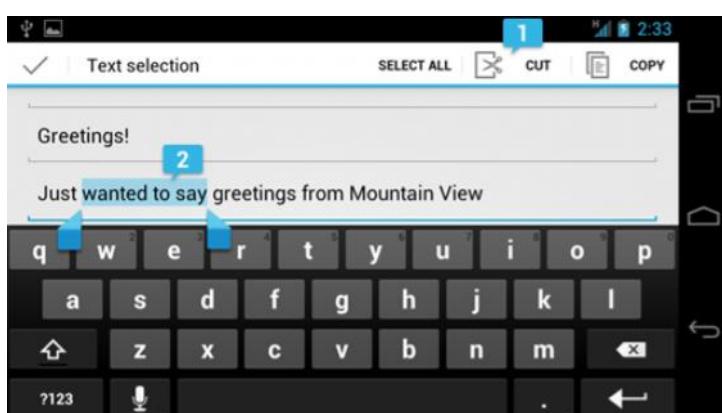
文本域自动填充

使用自动填充功能，在弹出的窗口中提示当前完成或搜索的结果，让用户更准确、更高效地输入文本信息。



文本选择

在文本域中长按任意字符就可以进行选择操作。这个 action 会触发文本选择模式，用于选择文本和操作当前选中的文本。选择模式包括：



1. Contextual action bar

Contextual action bar (CAB) 会显示当前适用的一些 actions 包括：剪切、复制和粘贴，当然可以按需增加其他 actions。

2. Selection handles

Selection handles 可以拖动选择文本，它在选择模式下会一直显示。

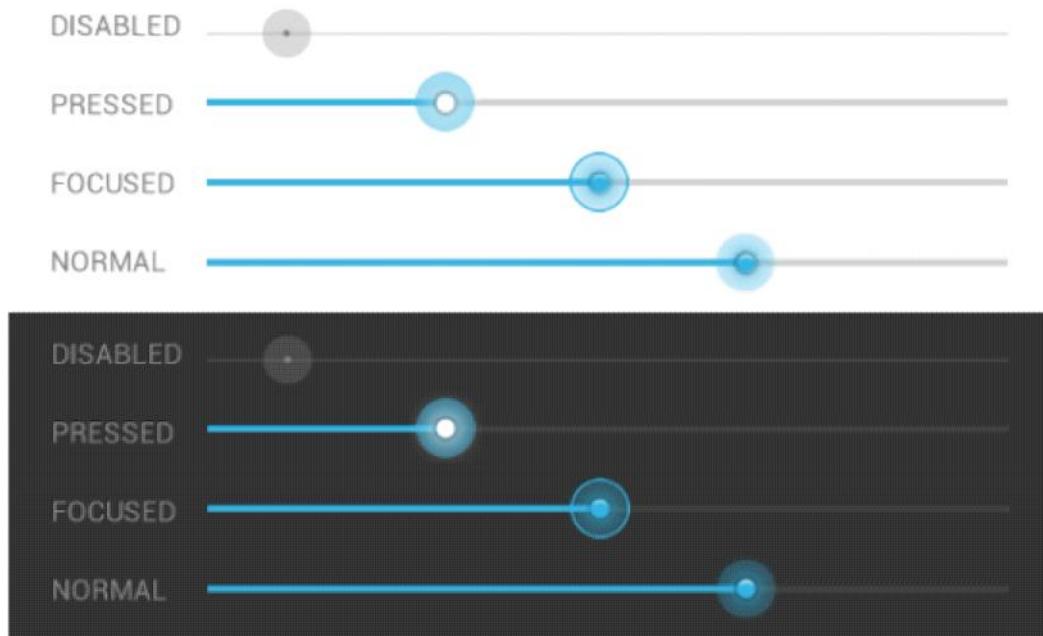
Seek Bars

交互式的滑块可以让用户通过手指的滑动，在一个连续或离散的数值域内设置一个值。最小值在左侧，最大值在右侧。基于滑块的交互特性，它非常适用于用来设置一些反映强度水平的数值，如音量、亮度或色彩饱和度。

举例

用滑块来设置闹铃的音量。音量值可以通过物理按钮设置，也可以通过一个简单的手势。





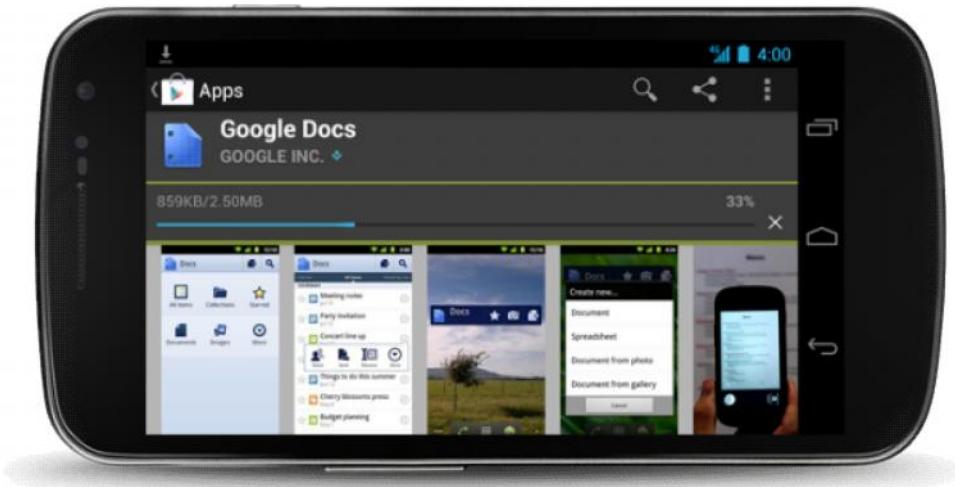
(Holo dark 和 Holo light 主题下的 Seek bars)

Progress & Activity-进度和活动

当一个用户喜欢的操作会持续很长时间时，提供一个可视化的反馈来告知用户当前的进度。

进度

如果你想知道进度完成的百分比，使用一个 determinate progress bar 来反馈进度信息，告知用户还需要多少时间完成。



Progress bar 应该从 0 到 100% 显示。避免设置一个比当前低的值，或者使用同一个 progress bar 来反馈多个任务进度，这样做是毫无意义的。如果你确定这样操作会执行多久，那就使用 indeterminate progress indicator (比如 android 旋转的小圈和 iOS 中的菊花，其实在 android 开发中他们都叫 progress bar)。



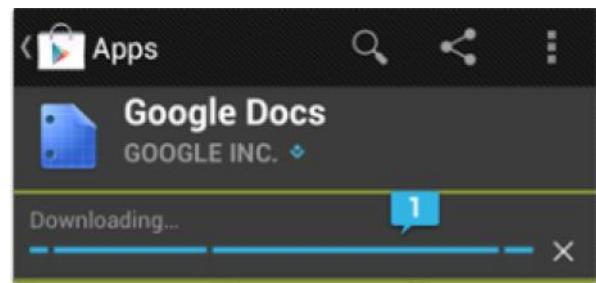
(Holo dark 和 Holo light 主题下的 progress bar)

活动

如果你不知道操作需要多久才能执行，那就使用 indeterminate progress indicator/它有两种风格：bar 和 circle。在合适的场景下使用他们。

1. Activity bar (with Holo dark 主题)

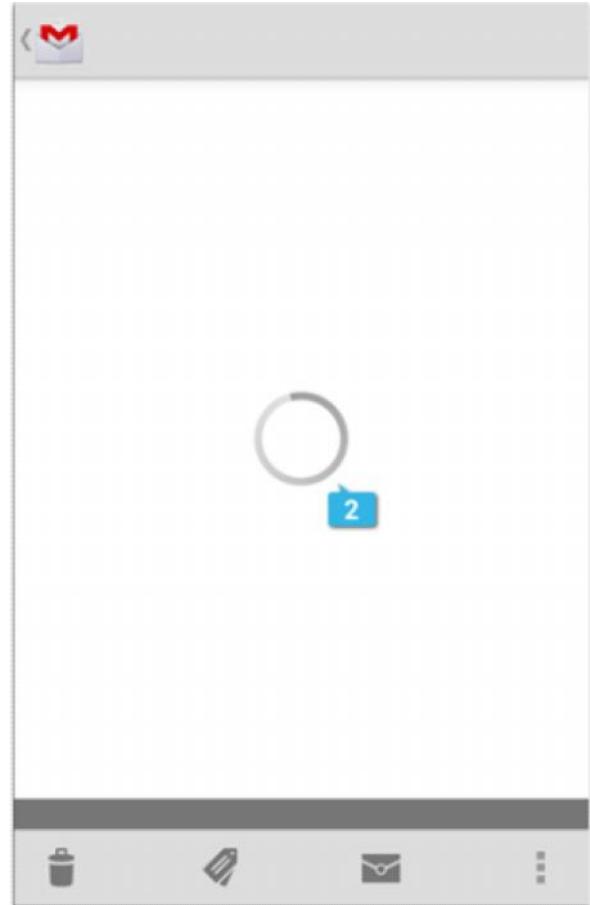
Indeterminate activity bar 常用于应用程序的下载。由于网络问题，play 商店可能无法确定多久才能开始下载。



2. Activity circle (with Holo light 主题)

Indeterminate activity circle 用于 Gmail 加载邮件的过程，因为无法确定邮件需要多久才能加载完毕。

每一个 activity 只允许使用一个 progress bar，同时根据视图内容适当调整它的大小。例如，大尺寸的 activity circle 适用于空白的内容区域，小尺寸的则适用于对话框。

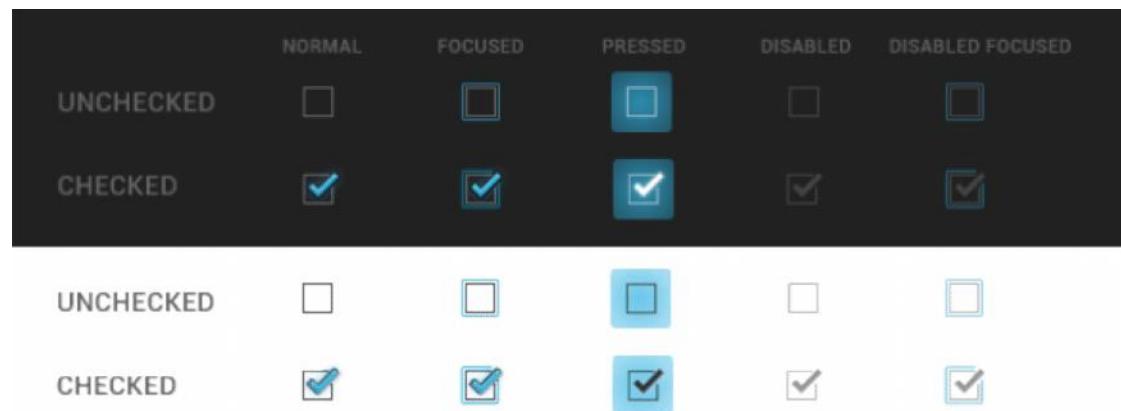


Switches-开关

开关允许用户选择选项。下面有三种开关样式：checkboxes、radio buttons、on/off switches。

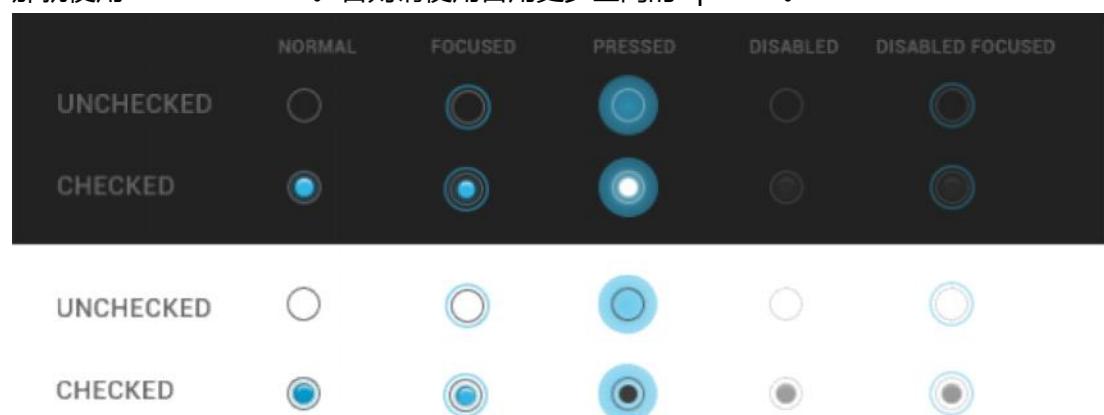
Checkboxes

Checkboxes 允许用户在一组选项多选。不要使用单个 checkbox 来控制一个选项的开关，而要使用 on/off switch。



Radio buttons

Radio buttons 只允许用户在一组选项中单选。如果你认为用户需要看到所有的单选选项，那就使用 radio buttons。否则请使用占用更少空间的 spinner。



On/off switches

On/off switches 用于切换单一设置选项的状态



Dialogs-对话框

对话框用来请求用户做出决定或提示额外的信息，以便能够继续执行一个任务。这一类的请求，可以是一个只包含取消或确定的简单对话框，也可以是一个包含复杂布局，甚至是要求用户调整参数、输入文本的复杂对话框。



1. 可选标题区域

标题用来介绍对话框的内容。例如，它可以定制用户修改或者决定的设置名称。

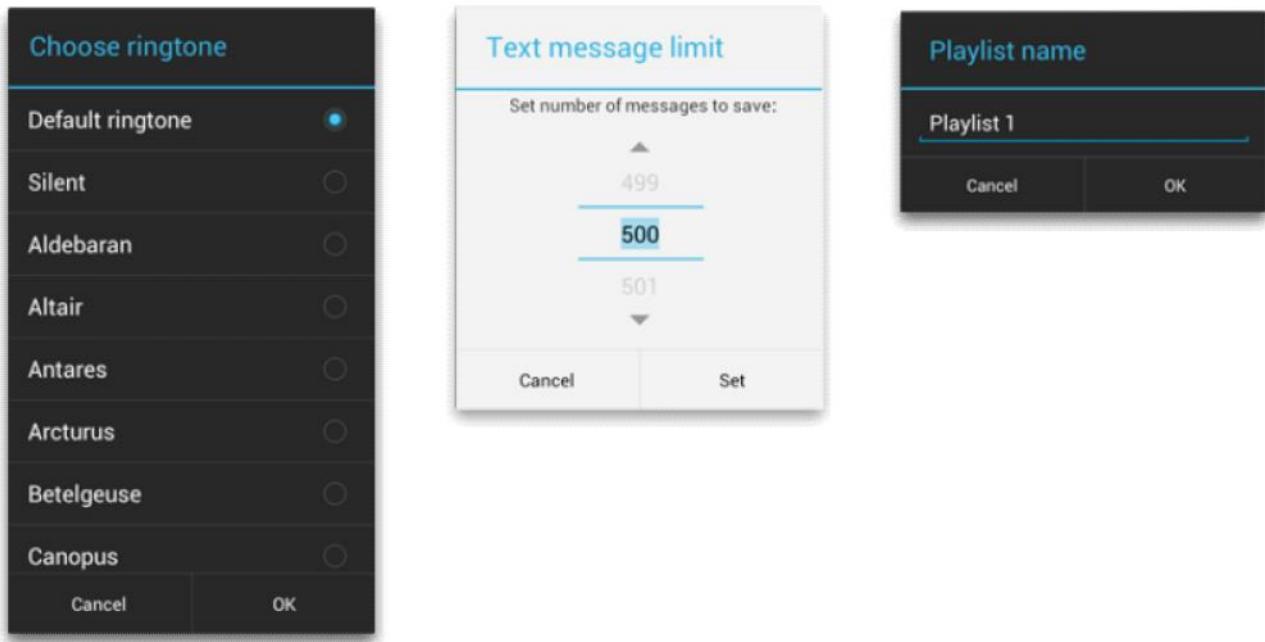
2. 内容区域

对话框内容的形式繁多。在设置对话框中，它可能包括了很多能够改变系统设定的 UI 元素，比如滑块、文本域、checkboxes 或 radio buttons。除此之外，比如“警告”对话框，它的内容可能完全是让用户做下一步决定的文案介绍。

3. Action buttons

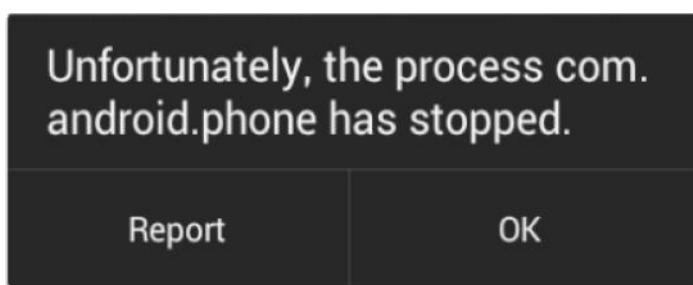
Action buttons 一般就是取消和（或）确定，确定是首选操作或者是最可能的操作。不过，如果这些选项包含诸如关闭或等待等特定 actions，而不是一个内容中描述的确定或者取消，那么所有的按钮文本都应该使用主动动词。有一个规则是这样的：否定的操作放在对话框的左边，确定的操作放在右边。

以下是一些常见的对话框



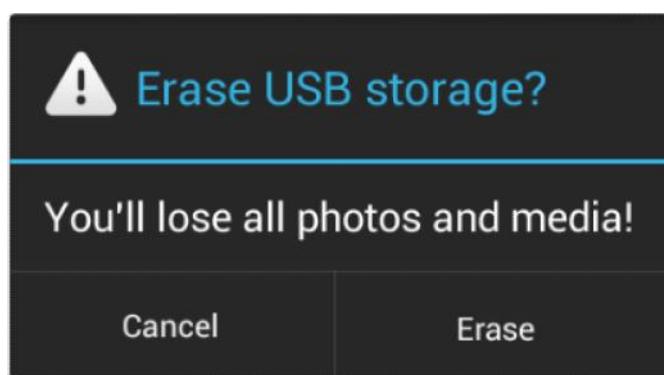
警告

警告可以提示用户：该任务执行前需要他们的确认。根据对话框内容的严重性和影响，他们的外观也会有不同的显示。



不带标题的警告

大多数的警告不需要标题。没有太大影响的决定可以用一两句话就能总结清楚。内容区域应该展示一个问题（比如“要删除这个对话吗？”）或者是一个与 action buttons 有很大联系的描述。



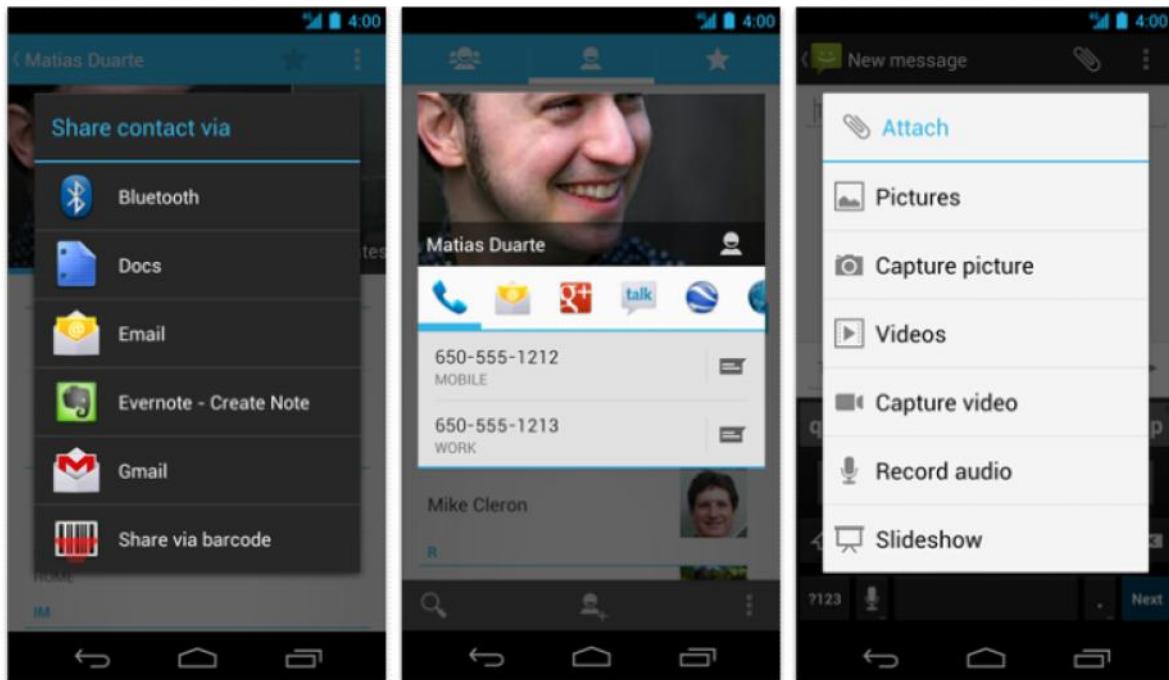
带标题的警告

尽量少使用带标题的警告，他们只是用于可能丢失数据、链接、额外费用等一些高危险的操作。使用的时候请显示一个明确的问题或者描述（标题）以及附加的解释说明（在内容区域）。

问题和描述要简短明确：比如，“清除 USB 存储？”。不要使用道歉的表述。即使让用户能够跳过内容描述，也能根据标题和 action buttons 上的文本来进行下一步操作。

弹出窗口

弹出窗口是一个“轻量”版对话框，用于让用户进行单一的选择。弹出窗口没有明显的按钮用于确定或取消操作。只需要选择一个项目或者轻触弹出窗口以外的区域就能取消弹出窗口。(系统的 Back 按钮同样适用)



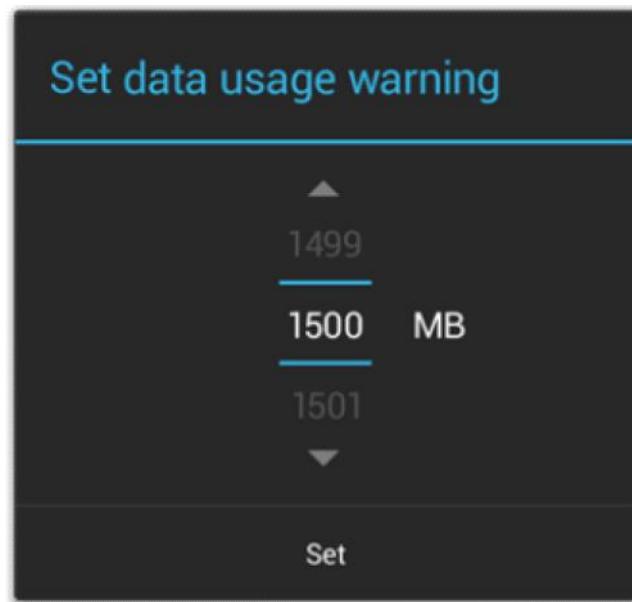
Toast

Toast 为各种 actions 提供轻量级的反馈信息。比如，在邮件发送之前离开编辑窗口，就会反馈一个“草稿已保存”的 toast，提醒你下次可以继续编辑邮件。Toast 会在一段时间后自动消失。(toast 可以选择显示时间的长短)。



Pickers-选择器

选择器让用户快速地选择一组数据。除了使用上/下箭头按钮之外，还可以用键盘直接输入数值，也可以通过滑动手势来改变数值。

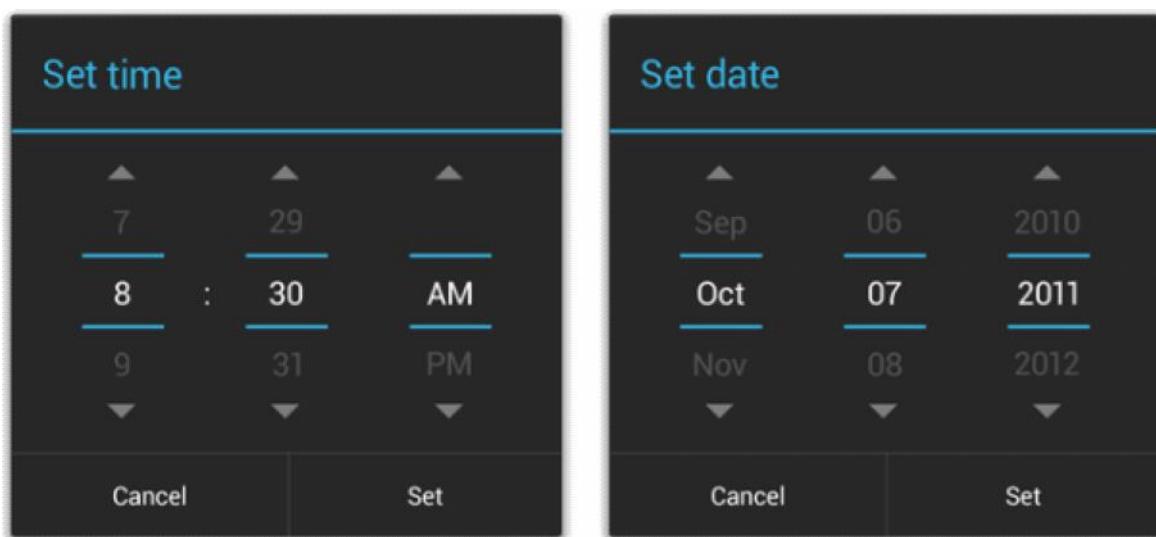


空间的考虑

选择器内嵌于对话框，因为他们会占据比较大的空间。如果你需要嵌入式的显示方式(picker 都是弹出式的)，那就使用更加合适的控件，比如文本域和 spinners。

Date picker & Time picker

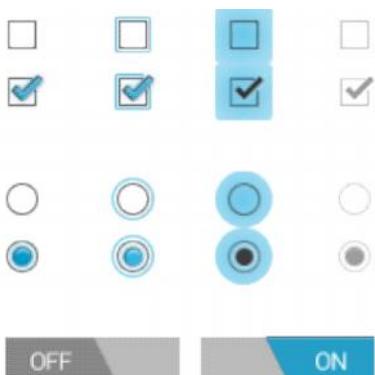
Android 系统提供很多可用的选择对话框。每个选择器都是一个可以输入日期（年月日）和时间（消失，分钟，AM/PM）的对话框。选择器可以帮助用户输入正确的日期和时间。Date piker 和 time picker 会根据不同的语言环境自动做出调整。



Downloads-资源下载

想要所有资源文件？我们已经为你将所有资源一个打包成了压缩文件。你也可以按照自己的需求下载对应的资源。

你可以毫无限制地使用这些资源。[下载全部资源](#)



模板和资源

使用这些模板和资源，为 Android4.0 设计精美的应用程序。这里包括了各种模板、色彩、空间以及系统图标。

[Adobe Fireworks PNG 模板](#)
[Omni OmniGraffle 模板](#)
[Adobe Photoshop 源文件](#)



Action bar 图标

图标包含了多种密度和多种主题的资源文件。

同时可以参考 [Action Bar Iconography](#)

[action bar 图标压缩包](#)



风格

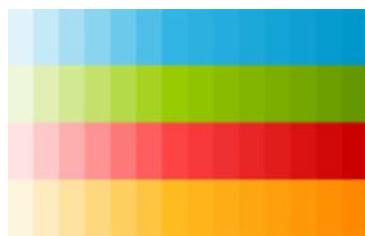
Roboto

Android4.0 引入的新字体。

更多参考 [Roboto](#)

[Roboto 字体](#)

[样本参考书](#)



色彩

蓝色是 Android 系统的基准色调。

更多参考 [色彩](#)

[颜色样本](#)

Contact-联系作者

潘思霽 (Pandroid)

pandroid@foxmail.com or lang7779698@gmail.com

Weibo@[兔斯霽](#)

Twitter@[Tusiji_LCL](#)

V2EX@[Pandroid](#)

Zhihu@[潘思霽](#)

Blog@[兔斯霽の LCL 之海](#)