

为Android 5.0设计Material风格的应用程序

(原创) 2016-01-01 伯乐在线-林申 安卓应用频道

(点击上方公众号，可快速关注)

我大爱 Material Design，Google 已经在这里给出了使用它的官方指导。如果你还没有看过的话，我建议你通读一下，因为它更加偏向讲述一个“好的设计”，而不是仅仅讲述如何去实现 Material Design。如果我是第一次看到，我一定会在角落那块地方或者对话框的按钮上点个大大的赞。

Kitkat 和 Jellybean 仍然主导着大部分市场（至少在我写这篇文章的时候），并且我们的应用还是需要支持运行着这两个版本系统的设备。Lollipop 已经出来一段时间了，我估计你也已经想到了一些办法把 Material Design 运用到你的应用上来，并且支持较老版本的系统。然后呢，这里也有一些办法，可以让你在几乎不用干任何设计活儿的情况下，使应用更加“material”：

1. 使用 App Compat v21 的主题和控件
2. Android Asset Studio 生成的图标
3. 使用免费开源库的中的组件
4. 设计模式：Google Support 自带的下拉刷新（Swipe to refresh）
5. 字体：Roboto

主题和控件，使用 App Compat v21

主题

Support 支持库的 V21版本已经伴随着 5.0 SDK 一起发布了。引用它便可以搞定应用“material 化”将近一半的工作。你将会得到不包含阴影的操作栏（ActionBar），并且在应用名称旁边也没有应用图标，还有由三个小圆点组成的更多（overflow）图标。菜单被放置在它们弹出的选项外边，这样当前被选中的菜单项就可以出现在所有项目或者控件等等元素的最上面。

新的 Support V21 包里总共有两套主题可用：暗色调（默认主题，通过 Theme.AppCompat 来启用）和亮色调（Theme.AppCompat.Light）

你的 styles.xml 现在需要这样写：

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="AppTheme" parent="AppTheme.Base"/>

    <style name="AppTheme.Base" parent="Theme.AppCompat">
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

colorPrimary 代表 ActionBar 的颜色, colorPrimaryDark 代表状态栏 (StatusBar) 的颜色, colorAccent 是染色控件 (tinted widgets) 的颜色. 比较可惜的是, colorPrimaryDark 属性在 Lollipop 之前的版本中是不能使用的.

你编写的 Activity 现在必须继承 ActionBarActivity , 如果要引用 操作栏 (ActionBar) 你需要调用 `getSupportActionBar();` 方法. 如果你的应用有导航抽屉 (navigation drawer) , 使用 `getSupportActionBar().setDisplayHomeAsUpEnabled(true);` 来获得. 可以看到, 当抽屉被打开的时候, 抽屉图标会变成一个返回箭头, 关闭的时候则相反.



抽屉图标变成返回箭头的时候还有一个萌萌哒的动画
控件

控件包括：

看到上面列表里面是不是忘了什么重要的东西？是的，按钮！

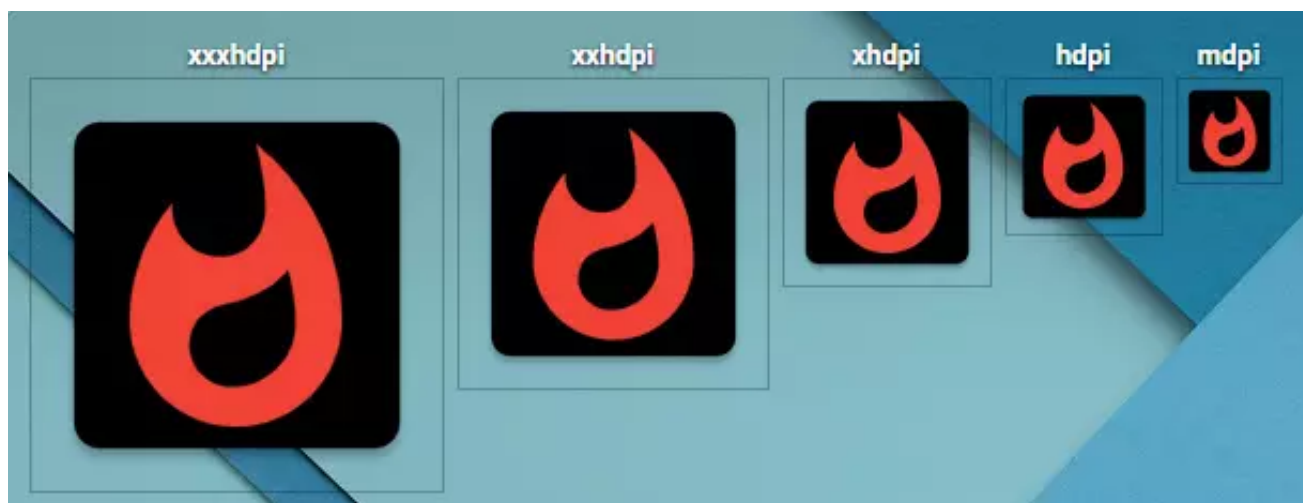
如何设置 App Compat library：Support Library Setup

Android Asset Studio 生成的图标

我做独立应用开发的时候从来没有错过这款工具，它包含一个图标生成器和一个点 9 图（nine patch）生成器。你可以用它从你的图像或文本中生成扁平化的应用图标、action bar 图标、剪切画，或者通用图标，（Android 的系统图标除外）。在生成通用图标的时候你还可以指定 DPI，但是 actionbar/tab 的图标暂时还不行，因为它们都是有标准大小的（xxhdpi, xhdpi, hdpi & mdpi）。可以看下这里：Android Asset Studio



使用 Android Asset Studio 生成的不同 DPI 一套图标

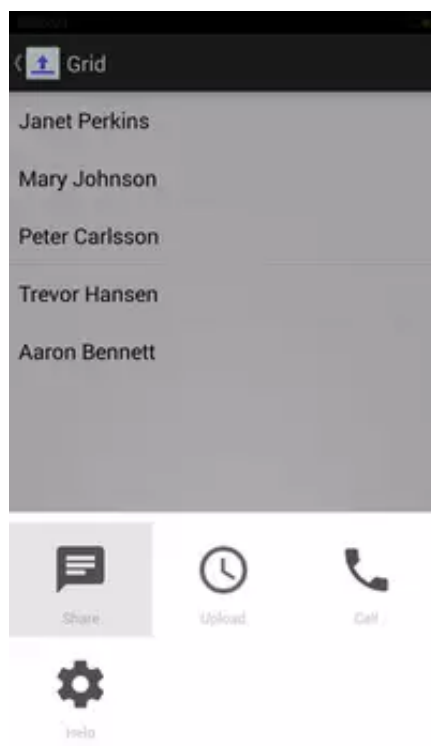
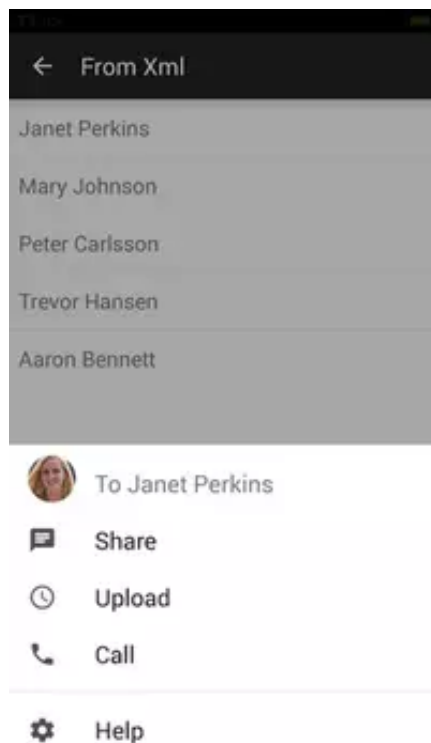


使用 Android Asset Studio 生成的应用程序（Launcher）图标

使用免费开源库的中的组件

不在 support 库中的一些组件，我们就得去看看其它开源库了。这里介绍一些我用过的挺棒的开源库

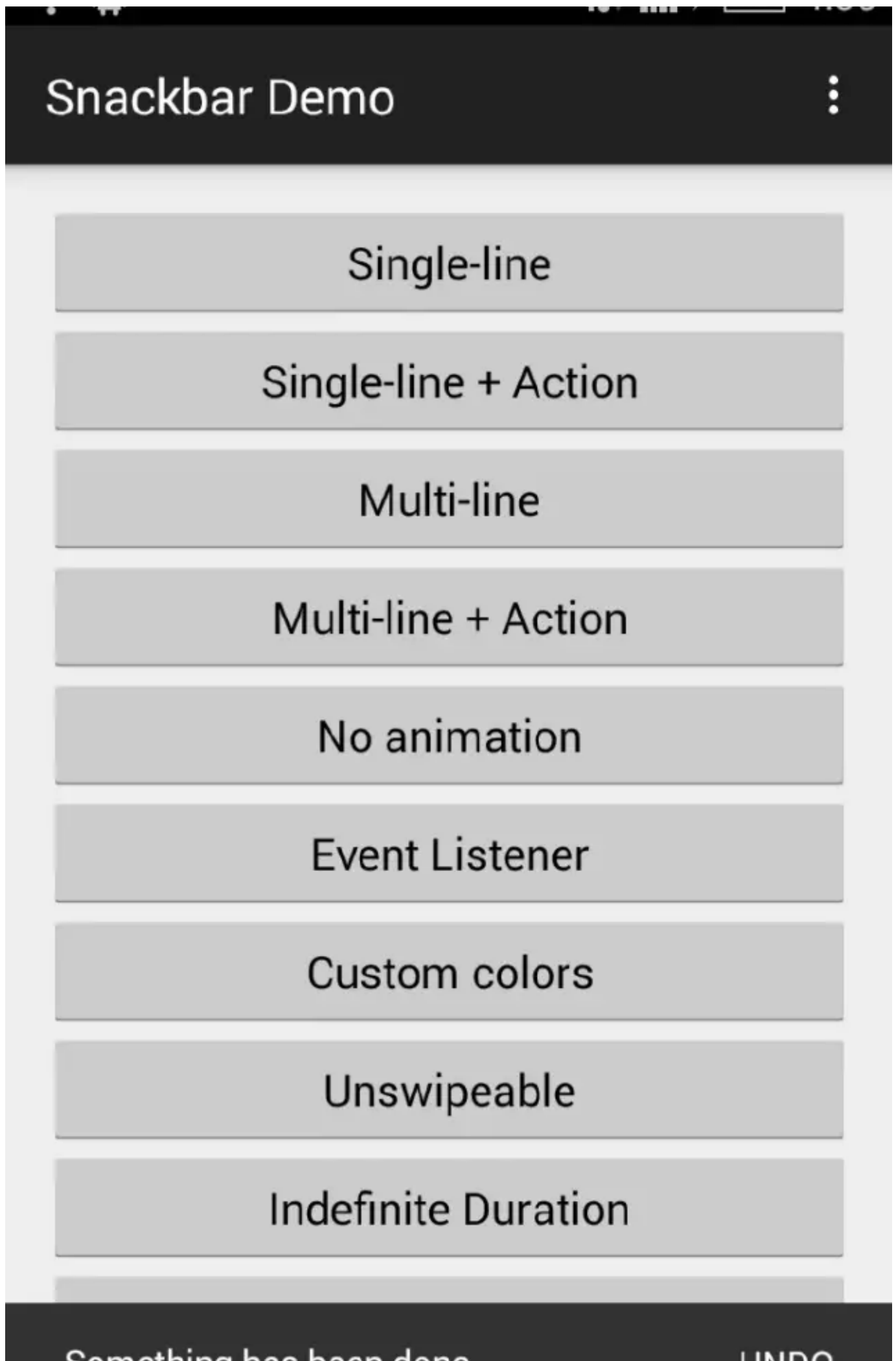
BOTTOMSHEET

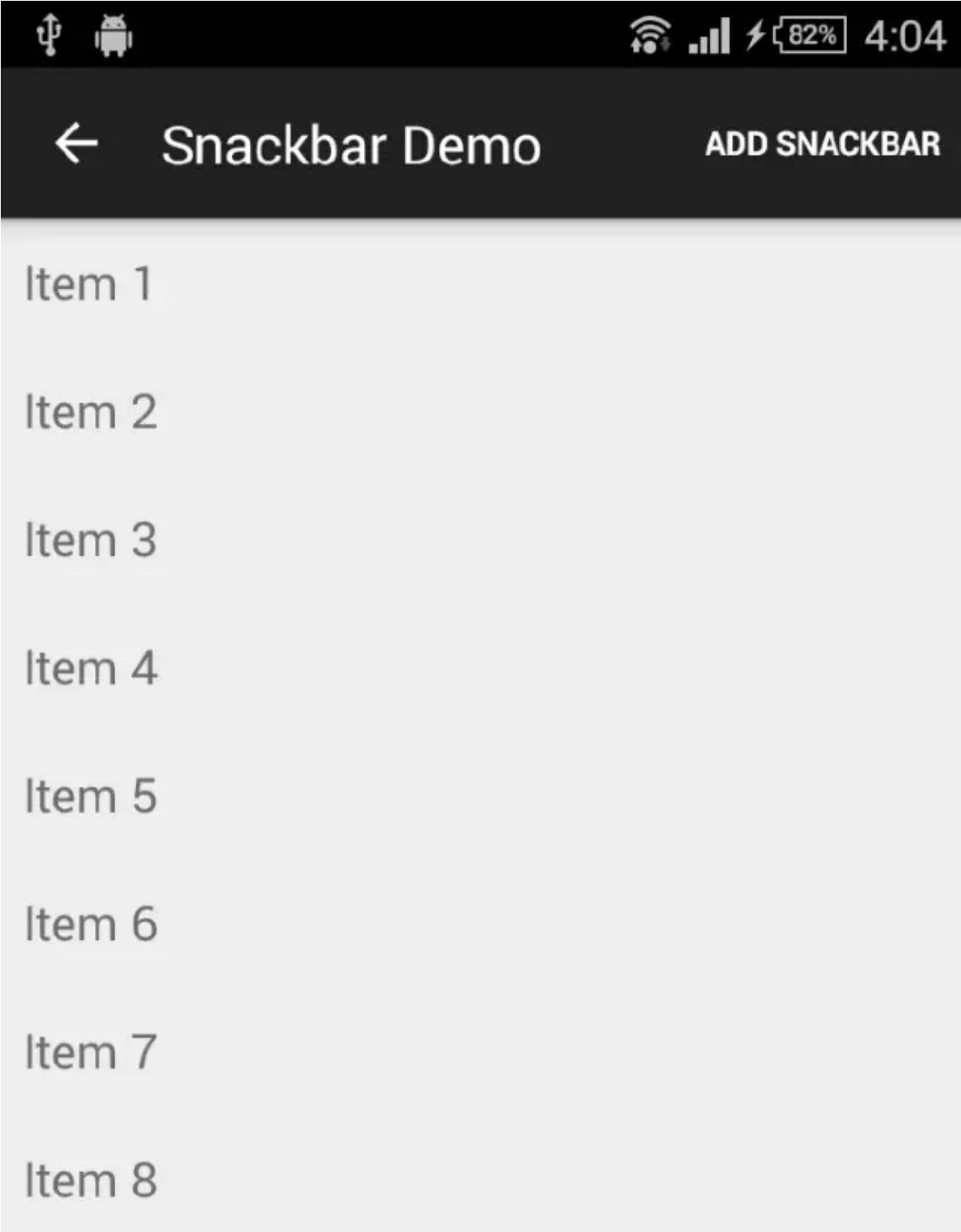
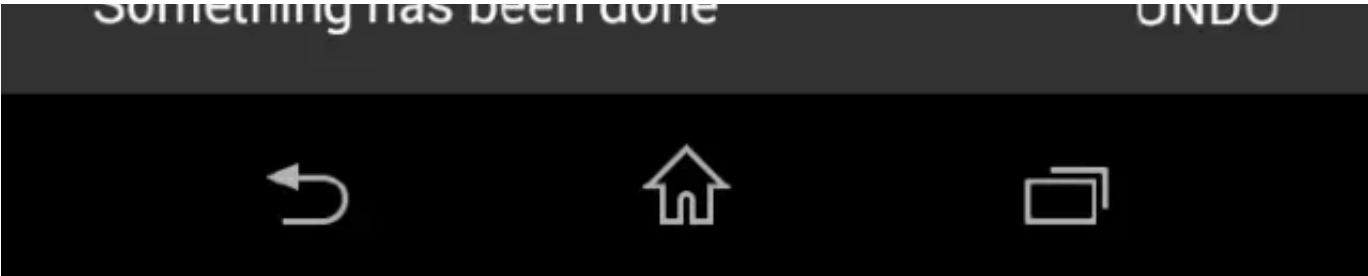


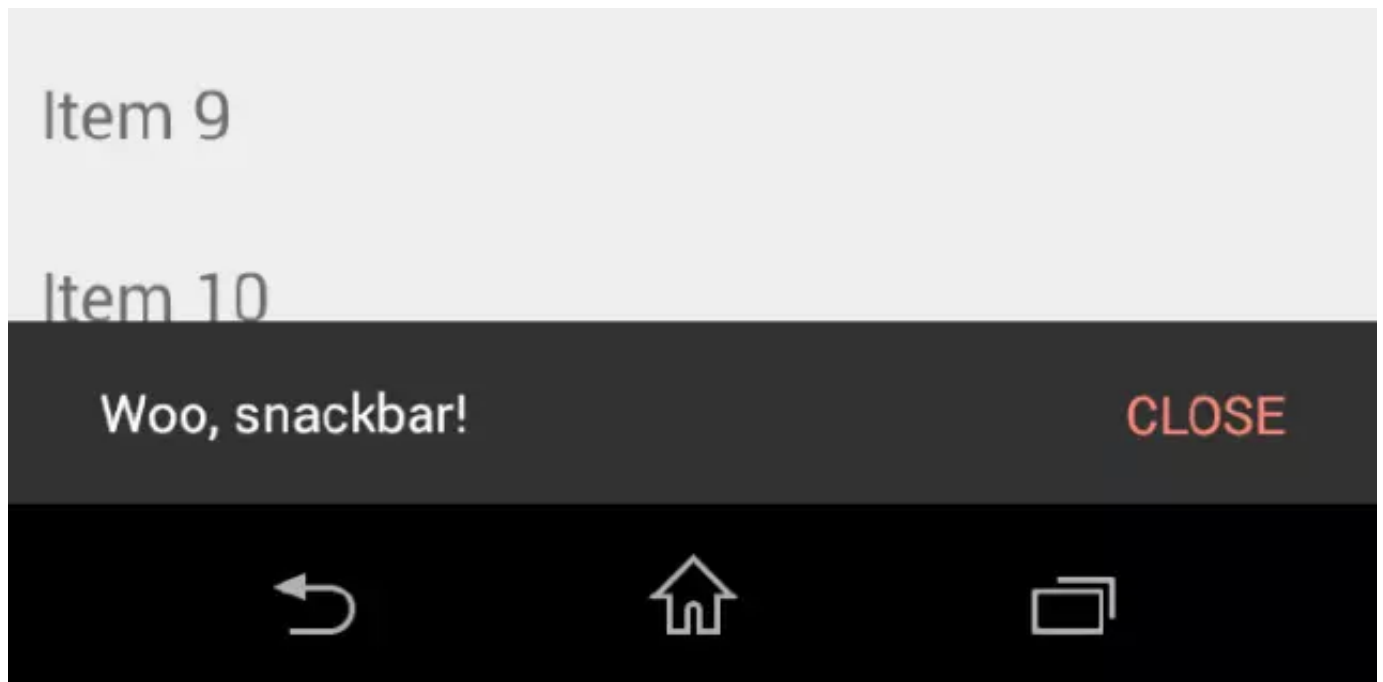
链接: <https://github.com/soarcn/BottomSheet>

SNACKBAR



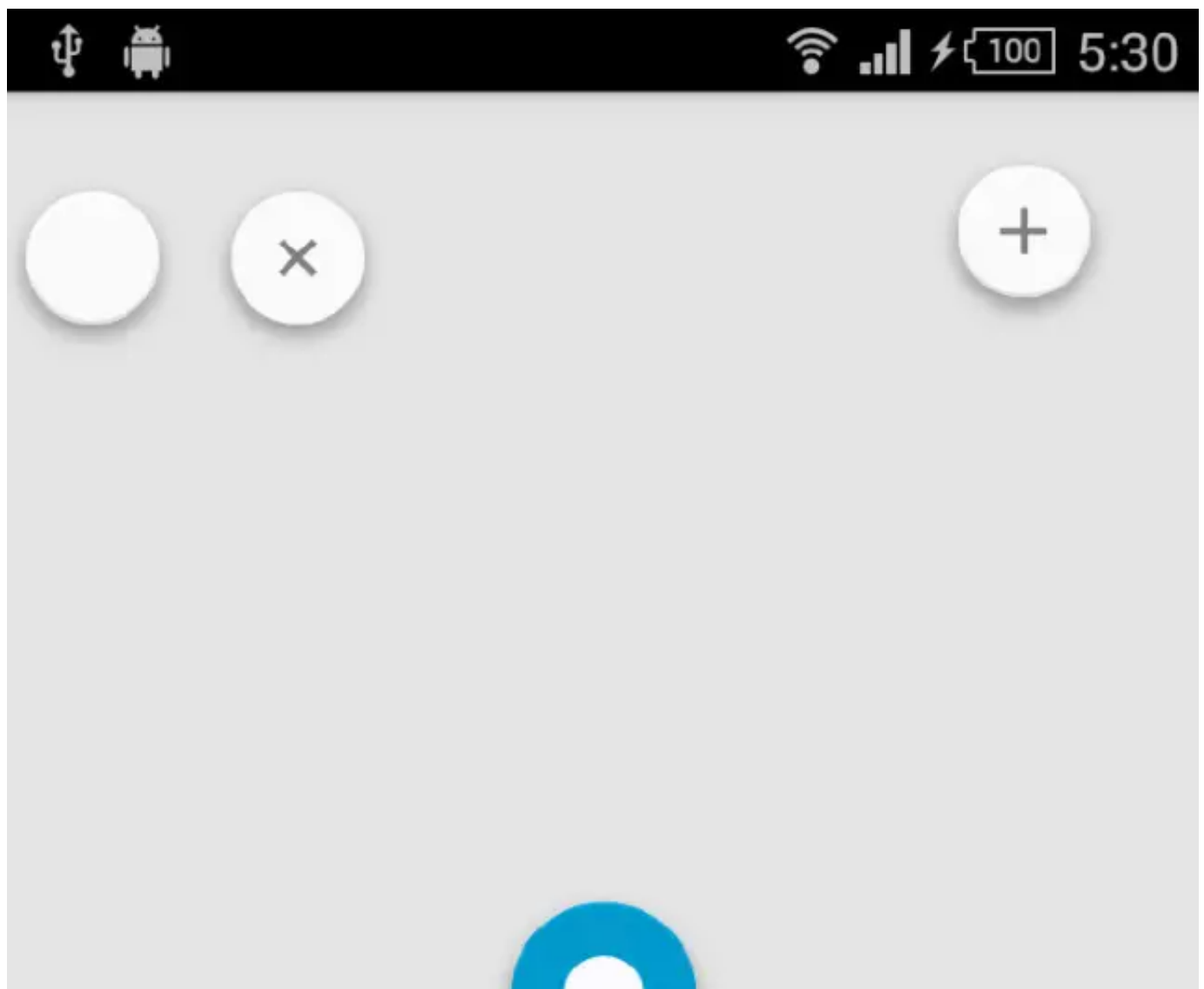


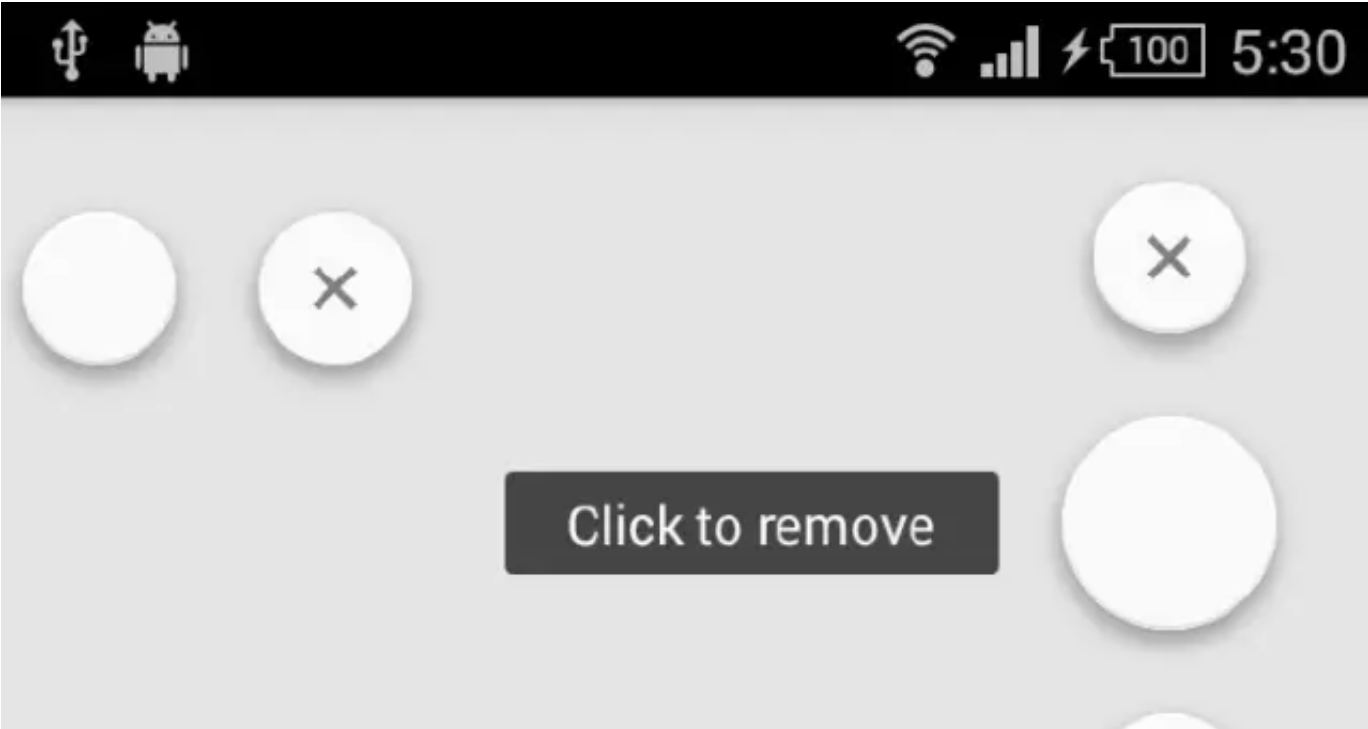
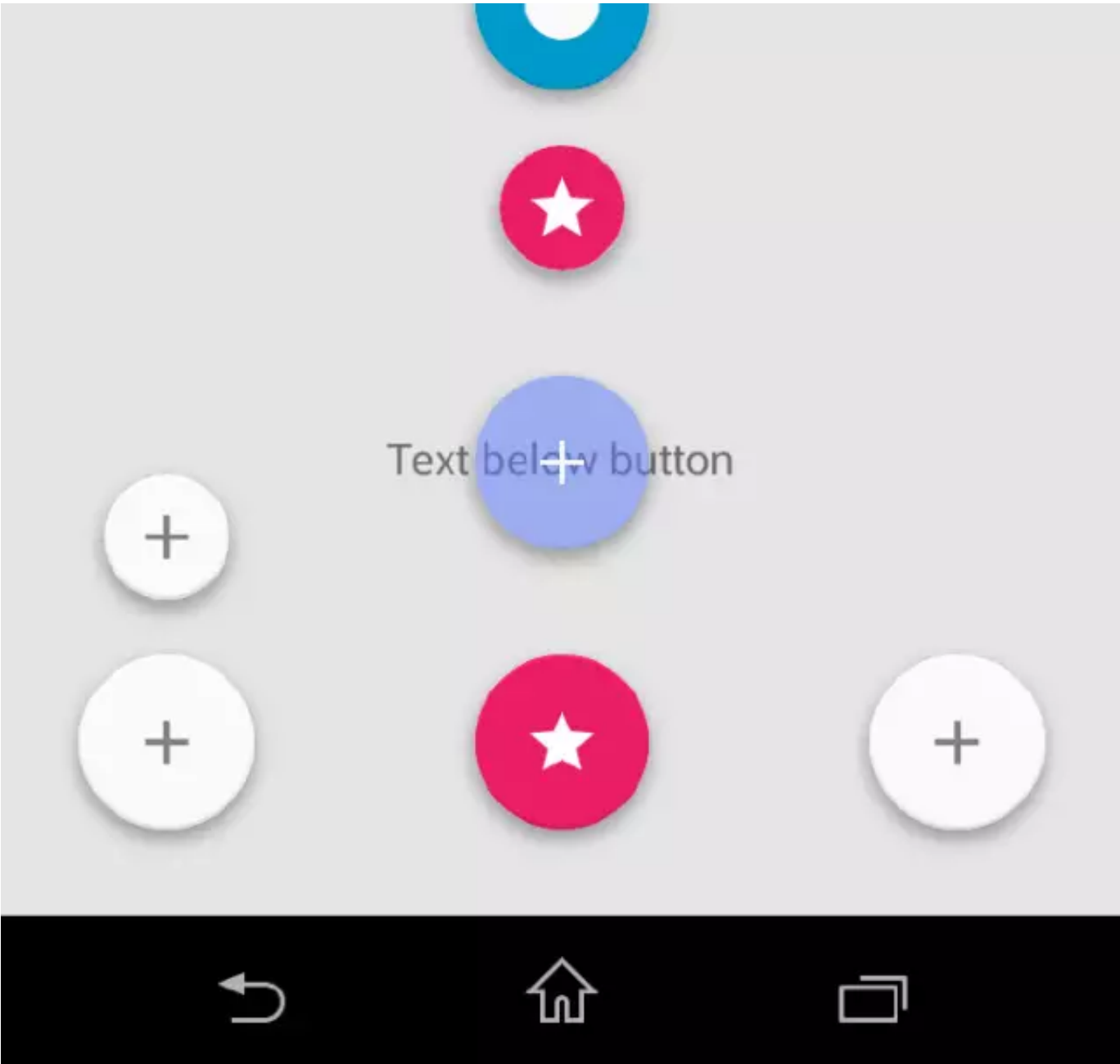


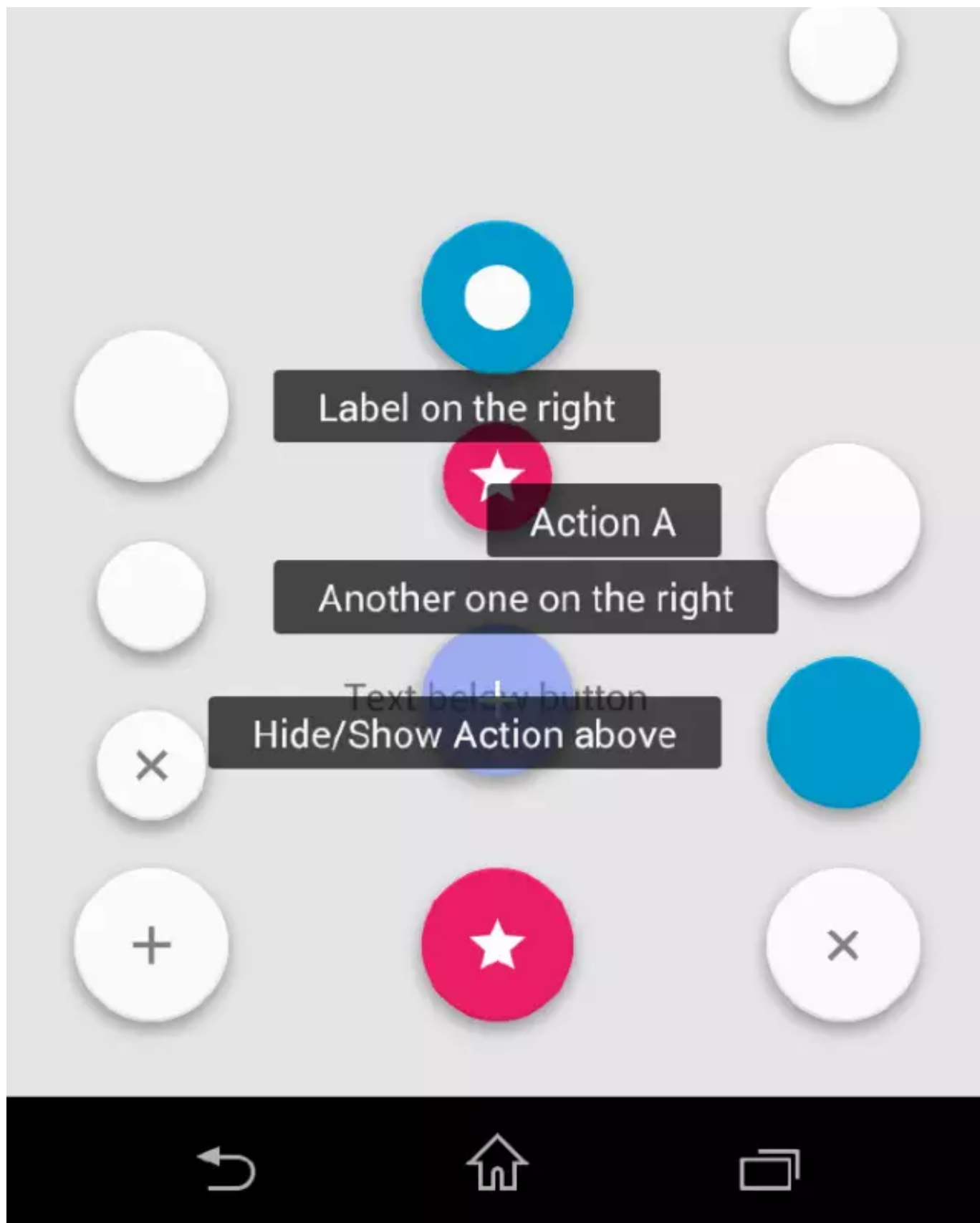


链接：<https://github.com/nispok/snackbar>

FLOATING ACTION BUTTON







链接：<https://github.com/futuresimple/android-floating-action-button>

设计模式：Google Support 自带的下拉刷新（Swipe to refresh）

如果你在你的 App 中用到了 Google 自带的下拉刷新，用 v4 Support 库的第 V21 版便会自动帮你添加一个 Material Design's 的 Spinner。

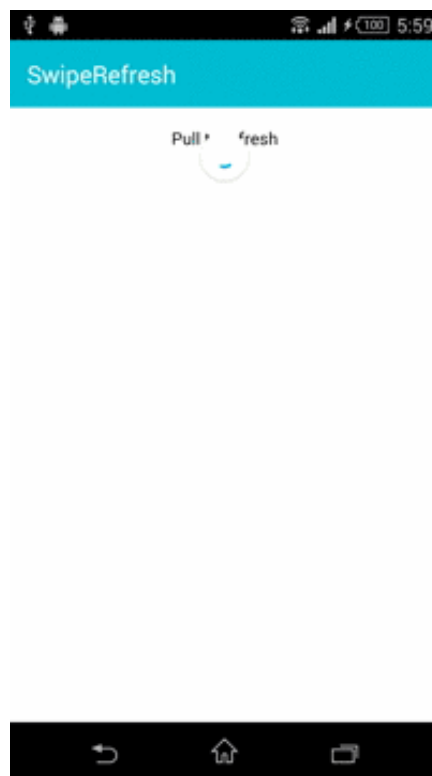
```
SwipeRefreshLayout swipeLayout = (SwipeRefreshLayout) findViewById(R.id.swipe_refresh_layout);  
swipeLayout.setOnRefreshListener(this);  
swipeLayout.setColorSchemeResources(R.color.red, R.color.green, R.color.blue);
```

布局 XML :

```
<android.support.v4.widget.SwipeRefreshLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/swipe_refresh_layout"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
    ...  
  
</android.support.v4.widget.SwipeRefreshLayout>
```

颜色 colors.xml :

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
    <color name="red">#f44336</color>  
    <color name="green">#4caf50</color>  
    <color name="blue">#03a9f4</color>  
</resources>
```



字体：Roboto

MATERIALS SCIENCE

6.626069×10⁻³⁴

One hundred percent cotton bond

Polymer chains

Alloys of aluminium, titanium and magnesium often

PAPERCRAFT

carbon in its graphite form is thought to be ceramic

MATERIALS

Cardstock 80lb ultra-bright orange

STATIONERY

随着 Material Design 一起发布的 Roboto 字体

Roboto 也跟着新的 Material Design 一起被更新了。Google 表示这款字体已经被调整到从手表到桌面、到电视、到汽车，以至于在更多的屏幕尺寸和条件下都能正常表现。要使用它或者任何自定义字体，把你的字体文件拖到 assets 文件夹。

```
Typeface robotoRegular = Typeface.createFromAsset(getAssets(), "Roboto-Regular.ttf");
```

```
//Then set it, for example to a TextView
```

```
textView.setTypeface(robotoRegular);
```

从此处获得 Roboto : <http://www.google.com/design/spec/resources/roboto-noto-fonts.html>

动画和阴影 (elevation) 一览

Lollipop 中最惊艳的应该就是动画了。拥有全新的 activity 动画 (activity transitions) , 触摸时的水波纹动画 (ripple animations) , 还有很多其它的 ! 而且还有阴影 (elevation) 属性 , 使得视图 (View) 终于有了“深度”的概念。动画全部由一个系统级的渲染线程 (RenderThread) 来管理。引用 Android Developers 博客中的原话 : “RippleDrawable 能流畅运行的最大功臣就是 Android 5.0 的渲染线程 (RenderThread) 。直到现在才发布是因为我们一直在优化它在较老 Android 版本上运行时的表现。” 阴影 (elevation) 效果也是如此。但是实现这样的动画有很多方法 , 特别是著名的水波纹效果 (Ripple effect) , 而且有很多开源库也可以帮到你 , 但是我选择在这里不讨论它们 , 因为我觉得如果用开源库的方式来实现就背离我的初衷了。

结论

我的手机即将在接下来的某一天收到 Lollipop 的更新。我常用的一些软件也已经跟进了 Material Design (Walkman, Pushbullet and 所有的 Google 自带应用) 。虽然也不是全部 , 因为它们没有办法自己去实现水波纹效果或者一些 5.0 才有的新功能。对于我们自己开发的应用而言 , 在多版本的系统间保证 UI 的完整性虽然不是必须的 , 但还是很重要的 。因此这篇文章列出了一些更简单的办法来实现这些效果 , 尤其对那些不想在 UI 上花太多时间的开发者。但愿这至少能在 Android 5.0 主导各大设备之前对大家有所帮助。

安卓应用频道

专注分享安卓应用相关内容



微信号: AndroidPD



长按,识别二维码关注

商务合作QQ: 2302462408

[阅读原文](#)



微信扫一扫
关注该公众号