

Android瀑布流StaggeredGridView学习研究

标签: StaggeredGridView 瀑布

2014-12-04

17:075916人阅读评论(6)收藏举报

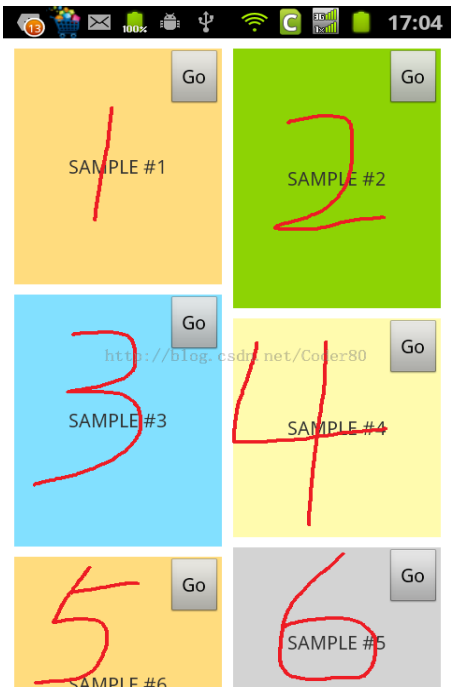
分类: Android进阶操练 (3)

版权声明: 本文为博主原创文章, 未经博主允许不得转载。

关于Android瀑布流控件, 已经在江湖上, 流传已久, 超过两年时间了。网上也有很多相关学习资源, 可以拿来研究学习。github上, 就有两个资源, 可以供学习者膜拜。

- 1.<https://github.com/maurycyw/StaggeredGridView> 此链接有图片加载功能, 但功能相对简单些。
- 2.<https://github.com/etsy/AndroidStaggeredGrid> 提供的瀑布流功能强大, 可以自定义瀑布流列数。

本篇博客, 就讲解etsy的源码为主了。首先看效果图:



首先明确StaggeredGridView中几个变量的定义:

```
[html]
01. private int mColumnCount; /*程序默认瀑布流的列数, 默认情况, 通过资源文件中的integers.xml 中
    grid_column_count定义*/
02. private int mItemMargin; /*程序默认瀑布流的margin, 通过layout文件activity_sgv.xml中的
    app:item_margin="8dp"定义*/
03. private int mColumnWidth; /*程序瀑布流的列宽变量*/
04. private boolean mNeedSync;
05. private int mColumnCountPortrait = DEFAULT_COLUMNS_PORTRAIT; /*程序瀑布流竖屏列数*/
06. private int mColumnCountLandscape = DEFAULT_COLUMNS_LANDSCAPE; /*程序瀑布流横屏列数*/
```

针对瀑布流, 搞清楚如下几个问题, 也算是吃透其中的原理了。

- 1.瀑布流的列数定义好了后, 如何计算每列的宽度?
- 2.瀑布流的列数定义好了后, 如何计算每列的高度?
- 3.瀑布流的HeaderView是如何添加的,其高度宽度如何确定?
- 4.瀑布流的FooterView是如何添加的,其高度宽度如何确定?
- 5.点击瀑布流的item时, 高亮和默认背景的selector如何实现?

一.瀑布流的列数定义好了后，如何计算每列的宽度？

进入StaggeredGridActivity界面，瀑布流的UI效果已经出来了，主要看com.etsy.android.grid.StaggeredGridView的onMeasure方法，确定每一个view的尺寸。

```
[java]
01. @Override
02. protected void onMeasure(final int widthMeasureSpec, final int heightMeasureSpec) {
03.     super.onMeasure(widthMeasureSpec, heightMeasureSpec);
04.     if (mColumnCount <= 0) {
05.         boolean isLandscape = isLandscape();
06.         mColumnCount = isLandscape ? mColumnCountLandscape : mColumnCountPortrait;
07.     }
08.     // our column width is the width of the listview
09.     // minus it's padding
10.     // minus the total items margin
11.     // divided by the number of columns
12.     mColumnWidth = calculateColumnWidth(getMeasuredWidth());
13.     ... ..
```

变量mColumnCount是从资源文件integers.xml 中grid_column_count获取，默认是2，函数calculateColumnWidth用于计算列宽，其定义也比较简单，容易理解。

```
[java]
01. private int calculateColumnWidth(final int gridWidth) {
02.     final int listPadding = getRowPaddingLeft() + getRowPaddingRight();
03.     return (gridWidth - listPadding - mItemMargin * (mColumnCount + 1)) / mColumnCount;
04. }
```

即屏幕宽度 - listPadding - mItemMargin 除以 列数即item的宽度.此时此刻，列宽即可以得到。继续往下debug代码，由于是使用SampleAdapter，瀑布流中的每一个item均要调用getView方法，此方法跟所有的Adapter一样，要从layout文件中，导入用户自定义的布局文件。代码为：convertView = mLayoutInflater.inflate(R.layout.list_item_sample, parent, false);

资源文件list_item_sample.xml其定义如下：

```
[html]
01. <?xml version="1.0" encoding="utf-8"?>
02. <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
03.     android:id="@+id/panel_content"
04.     android:layout_width="match_parent"
05.     android:layout_height="match_parent"
06.     android:orientation="horizontal" >
07.     <com.etsy.android.grid.util.DynamicHeightTextView
08.         android:id="@+id/txt_line1"
09.         android:layout_width="wrap_content"
10.         android:layout_height="wrap_content"
11.         android:clickable="true"
12.         android:background="@drawable/list_item_selector"
13.         android:gravity="center" />
14.     <Button
15.         android:id="@+id/btn_go"
16.         android:layout_width="60dp"
17.         android:layout_height="60dp"
18.         android:layout_gravity="top|right"
19.         android:text="Go" />
20. </FrameLayout>
```

此处需要了解DynamicHeightTextView 的定义了。整个Adapter中，子item的尺寸，均由DynamicHeightTextView 来确定。其

类中的onMeasure如下：

```
[html]
01. @Override
02. protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
03.     if (mHeightRatio > 0.0) {
04.         // set the image views size
05.         int width = MeasureSpec.getSize(widthMeasureSpec);
06.         int height = (int) (width * mHeightRatio);
07.         setMeasuredDimension(width, height);
08.     }
09.     else {
10.         super.onMeasure(widthMeasureSpec, heightMeasureSpec);
11.     }
12. }
```

遍历mHeightRatio宽高比，从setHeightRatio中获取，经过层层代码跟踪，瀑布流中子item的高度已经剖析出来了。子view的宽高确定，首先要根据屏幕尺寸，确定宽度

然后根据SampleAdapter中的getRandomHeightRatio函数，确定高度，高度是宽度的1~1.5倍。

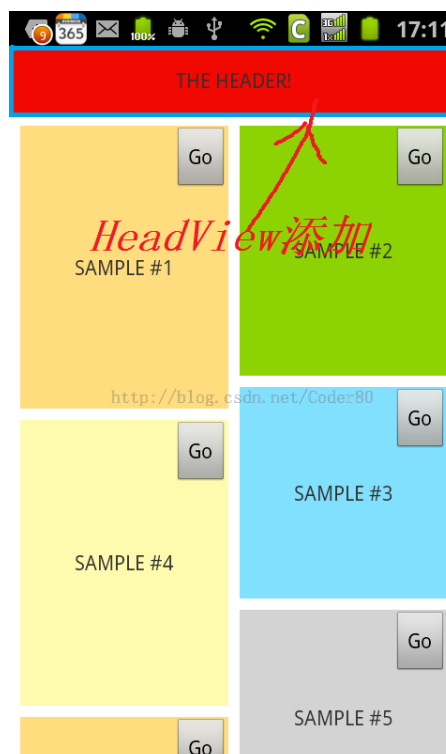
| 载:

```
[html]
01. private double getRandomHeightRatio() {
02.     return (mRandom.nextDouble() / 2.0) + 1.0; // height will be 1.0 - 1.5 the width
03. }
```

剖析至此，可以回答上述的问题1和2了。

二.瀑布流的HeaderView和FooterView是如何添加的,其高度宽度如何确定？

首先看效果图：



| 载:

从StaggeredGridActivity来看，就只有简单的一行代码，mGridView.addView(header)，便可以给list增加head了。瀑布流的SampleAdapter如何跟headview结合在一起呢？

在mGridView.setAdapter(mAdapter)时，调用ExtendableListView的setAdapter方法，ExtendableListView继承于AbsListView。ExtendableListView.java 的setAdapter方法如下：

[html] C { }

```

01. @Override
02. public void setAdapter(final ListAdapter adapter) {
03.     if (mAdapter != null) {
04.         mAdapter.unregisterDataSetObserver(mObserver);
05.     }
06.     // use a wrapper list adapter if we have a header or footer
07.     if (mHeaderViewInfos.size() > 0 || mFooterViewInfos.size() > 0) {
08.         mAdapter = new HeaderViewListAdapter(mHeaderViewInfos, mFooterViewInfos, adapter);
09.     }
10.     else {
11.         mAdapter = adapter;
12.     }
13.     ... ..

```

在mHeaderViewInfos或者mFooterViewInfos的size不为0时，构造一个HeaderViewListAdapter。其定义如下：

| 载:

[html] C { }

```

01. public class HeaderViewListAdapter implements WrapperListAdapter, Filterable

```

从定义可以看出，该对象属于一个ListAdapter，既然是Adapter，当然逃不脱几个重要的函数了。getCount(), getItemViewType().getView()。

[html] C { }

```

01. public int getCount() {
02.     if (mAdapter != null) {
03.         return getFootersCount() + getHeadersCount() + mAdapter.getCount();
04.     } else {
05.         return getFootersCount() + getHeadersCount();
06.     }
07. }

```

| 载:

getCount即adapter的item数量，当mAdapter!=null时，返回head和footer加上mAdapter.getCount.这样。head和footer也就与普通的瀑布流item一起，作为adapter的元素了。

[html] C { }

```

01. public int getItemViewType(int position) {
02.     int numHeaders = getHeadersCount();
03.     if (mAdapter != null && position >= numHeaders) {
04.         int adjPosition = position - numHeaders;
05.         int adapterCount = mAdapter.getCount();
06.         if (adjPosition < adapterCount) {
07.             return mAdapter.getItemViewType(adjPosition);
08.         }
09.     }
10.     return AdapterView.ITEM_VIEW_TYPE_HEADER_OR_FOOTER;
11. }

```

| 载:

getItemViewType函数即更加参数position，来确定view type id，view是从getView函数中创建的。

该函数的意思是：head和footer的位置，返回AdapterView.ITEM_VIEW_TYPE_HEADER_OR_FOOTER，普通的瀑布流item，将返回mAdapter.getItemViewType(adjPosition)。

接着就分析getView函数了。任何一个Adapter，都要重写getView函数了，这是常识。

[html] C { }

```

01. public View getView(int position, View convertView, ViewGroup parent) {
02.     // Header (negative positions will throw an ArrayIndexOutOfBoundsException)
03.     int numHeaders = getHeadersCount();

```

```

04.         if (position < numHeaders) {
05.             return mHeaderViewInfos.get(position).view;
06.         }
07.         // Adapter
08.         final int adjPosition = position - numHeaders;
09.         int adapterCount = 0;
10.         if (mAdapter != null) {
11.             adapterCount = mAdapter.getCount();
12.             if (adjPosition < adapterCount) {
13.                 return mAdapter.getView(adjPosition, convertView, parent);
14.             }
15.         }
16.         // Footer (off-limits positions will throw an ArrayIndexOutOfBoundsException)
17.         return mFooterViewInfos.get(adjPosition - adapterCount).view;
18.     }

```

| 载:

函数的意思理解也很容易，header位置，将返回mHeaderViewInfos.get(position).view；footer位置，返回mFooterViewInfos.get(adjPosition - adapterCount).view。其他的位置，也就是不规则GridView中的item view，返回mAdapter.getView(adjPosition, convertView, parent)，mAdapter对象，将会调用到SampleAdapter.java中的getView方法了。

至此，就明白了Header和footer是如何被添加到瀑布流界面了。接下来，就来确定head和footer的高度宽度问题了。瀑布流中，从UI效果来看，有三种类型的type view，一个是head，一个是StaggeredGridView，另外一个为footer了。计算子view的尺寸，当然关注onMeasureChild函数了。StaggeredGridView.java中的onMeasureChild函数定义如下：

```

[html] C 8
01.     @Override
02.     protected void onMeasureChild(final View child, final LayoutParams layoutParams) {
03.         final int viewType = layoutParams.viewType;
04.         final int position = layoutParams.position;
05.
06.         if (viewType == ITEM_VIEW_TYPE_HEADER_OR_FOOTER ||
07.             viewType == ITEM_VIEW_TYPE_IGNORE) {
08.             // for headers and weird ignored views
09.             super.onMeasureChild(child, layoutParams);
10.         }
11.         else {
12.             ... ..

```

可以看出，当viewType == ITEM_VIEW_TYPE_HEADER_OR_FOOTER时，调用父类ExtendableListView的onMeasureChild方法，计算head的尺度；当viewType != ITEM_VIEW_TYPE_HEADER_OR_FOOTER时，走else流程，根据int childWidthSpec = MeasureSpec.makeMeasureSpec(mColumnWidth, MeasureSpec.EXACTLY) 来计算尺寸。

| 载:

三.点击瀑布流的item时，高亮和默认背景的selector如何实现？



先上效果图：

从github上下载的源代码，经过稍微的修改，便可以实现点击item高亮效果，还是在资源布局文件list_item_sample.xml中看出端倪：

```
[html]
01. <com.etsy.android.grid.util.DynamicHeightTextView
02.     android:id="@+id/txt_line1"
03.     android:layout_width="wrap_content"
04.     android:layout_height="wrap_content"
05.     android:clickable="true"
06.     android:background="@drawable/list_item_selector"
07.     android:gravity="center" />
```

代码：android:background="@drawable/list_item_selector"中，list_item_selector的定义如下：

```
[html]
01. <?xml version="1.0" encoding="utf-8"?>
02. <selector xmlns:android="http://schemas.android.com/apk/res/android">
03.     <!-- shapes defined for Android 2.3 drawable issues -->
04.     <item android:state_pressed="true" >
05.         <shape android:shape="rectangle">
06.             <solid android:color="@color/list_item_pressed" />
07.         </shape>
08.     </item>
09.     <!-- default -->
10.     <item>
11.         <shape android:shape="rectangle">
12.             <solid android:color="@android:color/transparent" />
13.         </shape>
14.     </item>
15. </selector>
```

这个selector的意思是，默认情况下，使用全透明效果

```
[html]
01. <solid android:color="@android:color/transparent" />
```

点中状态时，呈现<solid android:color="@color/list_item_pressed" />的效果：

使用这种机制的原因，也很容易理解,在SampleAdapter中的getView函数中，代码：

```
[html] 01. convertView.setBackgroundResource(mBackgroundColors.get(backgroundIndex));
```

载：

表示不同position的item，显示不同的颜色，那么selector的定义，默认的情况时，使用全透明效果，UI上，就可以看到item的背景色，不会被selector的默认颜色遮挡。

由于在<https://github.com/etsy/AndroidStaggeredGrid>中的项目工程，是在Android Studio中构建的，本人也使用过AS一段时间，发现有很多坑，不习惯，故将代码pull下来后，重新创建工程，在Eclipse中build通过，项目地址在：<https://github.com/hero-peng/My-StaggeredGridView>，需要的读者，请自行下载。