

Hongyang

生命不息，奋斗不止，万事起于忽微，量变引起质变

目录视图

个人资料



鸿洋_

关注

发私信

访问：9689046次

积分：40449

等级：BLOG > 8

排名：第64名

原创：188篇

转载：0篇

译文：6篇

评论：11485条

我的微信公众号

点击直达推送文章汇总

长期为您推荐优秀博文、开源项目、视频等，进入还有好玩的等着你，欢迎扫一扫。



联系方式

新动态

给我留言

QQ群：

429757068

264950424

463081660

请勿重复加群，Thx

文章分类

【Android 5.x】

(11)

【公告】博客专栏旧貌换新颜

【hot】直播技术精选

主流编程语言图谱之二

Android FoldingLayout 折叠布局 原理及实现（二）

标签：Android FoldingLayout

2015-03-19 10:36 20500人阅读 评论

分类：【Android 精彩案例】（36） 【Android 自定义控件实战】（28）

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

转载请标明出处：<http://blog.csdn.net/Imj623565791/article/details/44283093>，本文出自：[【张鸿洋的博客】](#)

1、概述

在上一篇Android FoldingLayout 折叠布局 原理及实现（一）我们实现了下面的1，2，3。

- 1、Matrix的setPolyToPoly使用
- 2、在图片上使用渐变和阴影
- 3、初步的FoldingLayout的实现，完成图片的折叠显示（可控制折叠次数、包含阴影的绘制）
- 4、引入手势，手指可以可以FoldingLayout的折叠
- 5、结合DrawerLayout实现折叠式侧滑
- 6、结合SlidingPaneLayout实现折叠式侧滑

本篇博客将继续完成4，5，6的内容。

上一篇博客中，我们实现了在一个自定义View种绘制我们的折叠图片，使用的是一个固定的图片，当然我们在实际使用布局而不是一张固定的图片。

所以首先我们把上一篇的博客进行修改，改为一个自定义的ViewGroup，我们叫做FoldLayout。

2、FoldLayout的实现

1、实现

我们的想法是这样的，我们的FoldLayout只能有一个直接子元素，当然这个子元素可以是RelativeLayout什么的，可以了我们的FoldLayout，就能实现折叠效果。

那么也就是说，我们的FoldLayout折叠效果展示的是它的子元素的“样子”，那么如何或者这个“样子”呢？

大家都知道，我们的ViewGroup有个方法叫做：dispatchDraw(Canvas)主要用来绘制子元素，我们可以对这个canvas复调用dispatchDraw(Canvas)来实现类似上篇博客最后的效果，这样就完成了我们的可行性的分析。

[java] C P

01. package com.zhy.view;

02.

03. import android.content.Context;

04. import android.graphics.Bitmap;

05. import android.graphics.Bitmap.Config;

06. import android.graphics.Canvas;

07. import android.graphics.Color;

08. import android.graphics.LinearGradient;

09. import android.graphics.Matrix;

10. import android.graphics.Paint;

11. import android.graphics.Paint.Style;

- 【Android 精彩案例】

(37)
- 【Android 源码解析】

(29)
- 【Android 自定义控件实战】

(29)
- 【Android 自定义控件之起步】

(7)
- 【Android 快速开发】

(12)
- 【Android 原生开发游戏】

(3)
- 【Java 并发专题】

(15)
- 【android 进阶之路】

(65)
- 【Java 设计模式】

(10)
- 【Android 百度地图】

(4)
- 【html5 css3精彩案例】

(14)
- 【Android github 控件】

(10)
- 【Android 基础】

(16)
- 【Javascript】

(9)
- 【rabbitMQ 用法】

(5)
- 【Android微知识点】

(4)

友情链接

- [郭霖的博客](#)
- [夏安明的博客](#)
- [任玉刚的博客](#)
- [元斌的博客](#)
- [敬佩的孔老师](#)
- [foruok的订阅号程序视界](#)
- [OpenCV大神shiter](#)
- [专为Android程序员的导航](#)
- [泡在网上的日子](#)

博客专栏

- 

HTML5 & CSS3 实战

文章：11篇

阅读：135335
- 

设计模式融入生活

文章：10篇

阅读：82523
- 

Android 精彩案例

文章：67篇

阅读：4088814

阅读排行

- Android Https相关完全解析 ...

(1524068)
- Android Fragment 真正的完...

(473032)
- Android RecyclerView 使用完...

(353564)
- Android OkHttp完全解析 是...

(268891)
- Android 自定义View (一)

(197237)
- Android 属性动画 (Property...

(190922)
- Android Fragment 真正的完...

(184061)
- Android 屏幕适配方案

(141709)
- Android 手把手教您自定义Vi...

(140712)
- Android 自定义RecyclerView...

(125035)

最新评论

- Android 属性动画 (Property Animation...

Super-B :好东西

```
12. import android.graphics.Shader.TileMode;
13. import android.util.AttributeSet;
14. import android.view.View;
15. import android.view.ViewGroup;
16.
17. public class FoldLayout extends ViewGroup
18. {
19.
20.     private static final int NUM_OF_POINT = 8;
21.     /**
22.      * 图片的折叠后的总宽度
23.      */
24.     private float mTranslateDis;
25.
26.     protected float mFactor = 0.6f;
27.
28.     private int mNumOfFolds = 8;
29.
30.     private Matrix[] mMatrices = new Matrix[mNumOfFolds];
31.
32.     private Paint mSolidPaint;
33.
34.     private Paint mShadowPaint;
35.     private Matrix mShadowGradientMatrix;
36.     private LinearGradient mShadowGradientShader;
37.
38.     private float mFlodWidth;
39.     private float mTranslateDisPerFlod;
40.
41.     public FoldLayout(Context context)
42.     {
43.         this(context, null);
44.     }
45.
46.     public FoldLayout(Context context, AttributeSet attrs)
47.     {
48.         super(context, attrs);
49.
50.         for (int i = 0; i < mNumOfFolds; i++)
51.         {
52.             mMatrices[i] = new Matrix();
53.         }
54.
55.         mSolidPaint = new Paint();
56.         mShadowPaint = new Paint();
57.         mShadowPaint.setStyle(Style.FILL);
58.         mShadowGradientShader = new LinearGradient(0, 0, 0.5f, 0, Color.BLACK,
59.             Color.TRANSPARENT, TileMode.CLAMP);
60.         mShadowPaint.setShader(mShadowGradientShader);
61.         mShadowGradientMatrix = new Matrix();
62.         this.setWillNotDraw(false);
63.
64.
65.
66.     @Override
67.     protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec)
68.     {
69.         View child = getChildAt(0);
70.         measureChild(child, widthMeasureSpec, heightMeasureSpec);
71.         setMeasuredDimension(child.getMeasuredWidth(),
72.             child.getMeasuredHeight());
73.
74.     }
75.
76.     @Override
77.     protected void onLayout(boolean changed, int l, int t, int r, int b)
78.     {
79.         View child = getChildAt(0);
80.         child.layout(0, 0, child.getMeasuredWidth(), child.getMeasuredHeight());
81.
82.         mBitmap = Bitmap.createBitmap(getMeasuredWidth(), getMeasuredHeight(),
83.             Config.ARGB_8888);
84.         mCanvas.setBitmap(mBitmap);
85.         updateFold();
86.
87.     }
88.
89.     private void updateFold()
90.     {
91.         int w = getMeasuredWidth();
92.         int h = getMeasuredHeight();
93.
94.         mTranslateDis = w * mFactor;
95.         mFlodWidth = w / mNumOfFolds;
96.         mTranslateDisPerFlod = mTranslateDis / mNumOfFolds;
97.
98.         int alpha = (int) (255 * (1 - mFactor));
```

FloatingActionButton 完全解析[Design ...
Scott163425 :大神，我学习你的博文很久了，收益很多。现在我遇到一个小问题：CollapsingToolbarLayout...
Android 屏幕适配方案
nuttertools798 :@b87936260:请问value s-sw000dp适配方案是怎样的呢？哪里可以找到资料？
Android 自定义View (二) 进阶
Ruffian-痞子 :楼主这个场景的开发中经常用到。但是我们一般使用textView的drawablesLeft即可实现可惜...
如何利用github打造博客专属域名
sunshine2050_csdn :为啥采用这种方法后，打开网站很慢
Android NestedScrolling机制完全解析 ...
im哆来咪 :@IT_peng:嗯 判断一下第一次上移的时候增加父容器高度。
Android 异步消息处理机制 让你深入理解...
坏蛋lk :@liqianwei1230:可以参考慕课网的nate老师讲的http://www.imooc.co...
Android 自定义View (三) 圆环交替 等待...
demon_die :@u013007459:大神发一个demo学习一下，谢谢！292374728@qq.com
Android NestedScrolling机制完全解析 ...
niaonao :顶
Android NestedScrolling机制完全解析 ...
IT_peng :@duolaimila:你的意思是动态增加父容器高度？

2015博客之星评选

统计

```
300. mSolidPaint.setColor(Color.argb((int) (alpha * 0.8F), 0, 0, 0));
301.
302. mShadowGradientMatrix.setScale(mFlodWidth, 1);
303. mShadowGradientShader.setLocalMatrix(mShadowGradientMatrix);
304. mShadowPaint.setAlpha(alpha);
305.
306. float depth = (float) (Math.sqrt(mFlodWidth * mFlodWidth
307.     - mTranslatedDisPerFlod * mTranslatedDisPerFlod) / 2);
308.
309. float[] src = new float[NUM_OF_POINT];
310. float[] dst = new float[NUM_OF_POINT];
311.
312. for (int i = 0; i < mNumOfFolds; i++)
313. {
314.     mMatrices[i].reset();
315.     src[0] = i * mFlodWidth;
316.     src[1] = 0;
317.     src[2] = src[0] + mFlodWidth;
318.     src[3] = 0;
319.     src[4] = src[2];
320.     src[5] = h;
321.     src[6] = src[0];
322.     src[7] = src[5];
323.
324.     boolean isEven = i % 2 == 0;
325.
326.     dst[0] = i * mTranslatedDisPerFlod;
327.     dst[1] = isEven ? 0 : depth;
328.
329.     dst[2] = dst[0] + mTranslatedDisPerFlod;
330.     dst[3] = isEven ? depth : 0;
331.     dst[4] = dst[2];
332.     dst[5] = isEven ? h - depth : h;
333.     dst[6] = dst[0];
334.     dst[7] = isEven ? h : h - depth;
335.
336.     for (int y = 0; y < 8; y++)
337.     {
338.         dst[y] = Math.round(dst[y]);
339.     }
340.
341.     mMatrices[i].setPolyToPoly(src, 0, dst, 0, src.length >> 1);
342. }
343.
344. private Canvas mCanvas = new Canvas();
345. private Bitmap mBitmap;
346. private boolean isReady;
347.
348. @Override
349. protected void dispatchDraw(Canvas canvas)
350. {
351.
352.     if (mFactor == 0)
353.         return;
354.     if (mFactor == 1)
355.     {
356.         super.dispatchDraw(canvas);
357.         return;
358.     }
359.     for (int i = 0; i < mNumOfFolds; i++)
360.     {
361.         canvas.save();
362.
363.         canvas.concat(mMatrices[i]);
364.         canvas.clipRect(mFlodWidth * i, 0, mFlodWidth * i + mFlodWidth,
365.             getHeight());
366.         if (isReady)
367.         {
368.             canvas.drawBitmap(mBitmap, 0, 0, null);
369.         } else
370.         {
371.             // super.dispatchDraw(canvas);
372.             super.dispatchDraw(mCanvas);
373.             canvas.drawBitmap(mBitmap, 0, 0, null);
374.             isReady = true;
375.         }
376.         canvas.translate(mFlodWidth * i, 0);
377.         if (i % 2 == 0)
378.         {
379.             canvas.drawRect(0, 0, mFlodWidth, getHeight(), mSolidPaint);
380.         } else
381.         {
382.             canvas.drawRect(0, 0, mFlodWidth, getHeight(), mShadowPaint);
383.         }
384.         canvas.restore();
385.     }
```

```

186.     }
187.     //...dispatchDraw
188.
189.     public void setFactor(float factor)
190.     {
191.         this.mFactor = factor;
192.         updateFold();
193.         invalidate();
194.     }
195.
196.     public float getFactor()
197.     {
198.         return mFactor;
199.     }
200.
201. }

```

上述代码大家应该不陌生，只是把从View对单个图片进行绘制的修改为了ViewGroup。

既然是ViewGroup少不了onMeasure,onLayout等。测量和布局完全依赖于它的子View。

然后将需要初始化的一些东西，不依赖于宽度的，比如画笔什么的都放在构造中；依赖宽高的，都在onLayout之后，谓初始化相关代码。

updateFold中的代码，我们也不陌生，因为和上篇博客基本一致。主要就是计算mFlodWidth，mTranslateDisPerFlo
mNumOfFolds去循环初始化我们的matrix.

matrix完成setPolyToPoly以后，我们就可以去绘制了：

```

[java]
01. private Canvas mCanvas = new Canvas();
02. private Bitmap mBitmap;
03. private boolean isReady;
04.
05. @Override
06. protected void dispatchDraw(Canvas canvas)
07. {
08.
09.     if (mFactor == 0)
10.         return;
11.     if (mFactor == 1)
12.     {
13.         super.dispatchDraw(canvas);
14.         return;
15.     }
16.     for (int i = 0; i < mNumOfFolds; i++)
17.     {
18.         canvas.save();
19.
20.         canvas.concat(mMatrices[i]);
21.         canvas.clipRect(mFlodWidth * i, 0, mFlodWidth * i + mFlodWidth,
22.             getHeight());
23.         if (isReady)
24.         {
25.             canvas.drawBitmap(mBitmap, 0, 0, null);
26.         } else
27.         {
28.             // super.dispatchDraw(canvas);
29.             super.dispatchDraw(mCanvas);
30.             canvas.drawBitmap(mBitmap, 0, 0, null);
31.             isReady = true;
32.         }
33.         canvas.translate(mFlodWidth * i, 0);
34.         if (i % 2 == 0)
35.         {
36.             canvas.drawRect(0, 0, mFlodWidth, getHeight(), mSolidPaint);
37.         } else
38.         {
39.             canvas.drawRect(0, 0, mFlodWidth, getHeight(), mShadowPaint);
40.         }
41.         canvas.restore();
42.     }
43. }

```

mFactor主要代表折叠后的总宽度与原宽度的比值，默认不折叠时为1，所以直接调用super.dispatchDraw(canvas);

那么如果为0，说明全部折起来了，我们直接if (mFactor == 0)return;就不要绘制了。

如果(0,1)之间就是正常情况了，如果还记得上一篇博客内容，无非就是根据mNumOfFolds循环绘制多次，每次绘制的

clipRect就可以实现我们的折叠。

这里大家注意看，我在第一次绘制的时候，调用了：

```
[java] C P
01. super.dispatchDraw(mCanvas);
02. canvas.drawBitmap(mBitmap, 0, 0, null);
03. isReady = true;
```

在我们自己new的mBitmap中也绘制了一份图片，因为我不希望每次都是调用super.dispatchDraw，所以只要isReady用绘制mBitmap而避免调用super.dispatchDraw()。

绘制完成图片，就是绘制黑色的遮盖和阴影了~~，就是两个Rect的绘制。

完成这些以后，我们可以简单的坐下测试，使用我们的布局。

2、测试

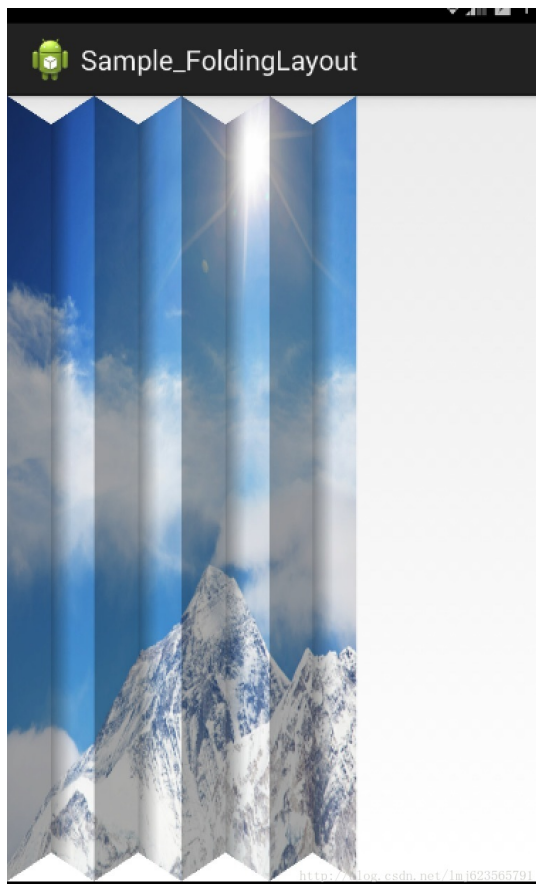
布局文件:

```
[html] C P
01. <com.zhy.view.FoldLayout xmlns:android="http://schemas.android.com/apk/res/android"
02.     xmlns:tools="http://schemas.android.com/tools"
03.     android:id="@+id/id_fold_layout"
04.     android:layout_width="match_parent"
05.     android:layout_height="match_parent" >
06.
07.     <ImageView
08.         android:layout_width="match_parent"
09.         android:layout_height="match_parent"
10.         android:scaleType="fitXY"
11.         android:src="@drawable/xueshan" />
12.
13. </com.zhy.view.FoldLayout>
```

Activity

```
[java] C P
01. package com.zhy.sample.folderlayout;
02.
03. import com.zhy.view.FoldLayout;
04.
05. import android.animation.ObjectAnimator;
06. import android.annotation.SuppressLint;
07. import android.app.Activity;
08. import android.os.Bundle;
09.
10. public class FoldLayoutActivity extends Activity
11. {
12.     private FoldLayout mFoldLayout;
13.
14.     @Override
15.     protected void onCreate(Bundle savedInstanceState)
16.     {
17.         super.onCreate(savedInstanceState);
18.         setContentView(R.layout.activity_fold);
19.
20.         mFoldLayout = (FoldLayout) findViewById(R.id.id_fold_layout);
21.
22.         /*mFoldLayout.post(new Runnable()
23.         {
24.
25.             @SuppressLint("NewApi")
26.             @Override
27.             public void run()
28.             {
29.                 ObjectAnimator.ofFloat(mFoldLayout, "factor", 1, 0, 1)
30.                     .setDuration(5000).start();
31.             }
32.         });*/
33.     }
34. }
35. }
```

现在的效果是，我们把mFactor改为0.6f:



当然了，此时只是静态的，但是我们成功的完成了绘制一个静态图到FlodLayout。

接下来我们为其增加手指的触摸折叠功能。

3、TouchFoldLayout

1、实现

增加触摸功能其实很简单，我们的绘制依赖mFactor这个值，我们只要在onTouchEvent里面去累加手指移动距离，然后了。

```
[java] C P
01. package com.zhy.view;
02.
03. import android.content.Context;
04. import android.graphics.Canvas;
05. import android.util.AttributeSet;
06. import android.view.GestureDetector;
07. import android.view.MotionEvent;
08.
09. public class TouchFoldLayout extends FoldLayout
10. {
11.
12.     private GestureDetector mScrollGestureDetector;
13.
14.     public TouchFoldLayout(Context context, AttributeSet attrs)
15.     {
16.         super(context, attrs);
17.         init(context, attrs);
18.     }
19.
20.     public void init(Context context, AttributeSet attrs)
21.     {
22.         mScrollGestureDetector = new GestureDetector(context,
23.             new ScrollGestureDetector());
24.
25.     }
26.
27.     @Override
28.     public boolean onTouchEvent(MotionEvent event)
29.     {
30.         return mScrollGestureDetector.onTouchEvent(event);
31.     }
32.
33.     private int mTranslation = -1;
34.
```

```

35.     @Override
36.     protected void dispatchDraw(Canvas canvas)
37.     {
38.         if (mTranslation == -1)
39.             mTranslation = getWidth();
40.         super.dispatchDraw(canvas);
41.     }
42.
43.     class ScrollGestureDetector extends GestureDetector.SimpleOnGestureListener
44.     {
45.         @Override
46.         public boolean onDown(MotionEvent e)
47.         {
48.             return true;
49.         }
50.
51.         @Override
52.         public boolean onScroll(MotionEvent e1, MotionEvent e2,
53.             float distanceX, float distanceY)
54.         {
55.             mTranslation -= distanceX;
56.
57.             if (mTranslation < 0)
58.             {
59.                 mTranslation = 0;
60.             }
61.             if (mTranslation > getWidth())
62.             {
63.                 mTranslation = getWidth();
64.             }
65.
66.             float factor = Math.abs(((float) mTranslation)
67.                 / (((float) getWidth()));
68.
69.             setFactor(factor);
70.
71.             return true;
72.         }
73.     }
74.
75. }

```

我们选择继承FoldLayout，重写其onTouchEvent，然后通过mScrollGestureDetector获取移动的距离，最终和width factor，然后调用setFactor进行改变。

```

[java] C P
01. public void setFactor(float factor)
02. {
03.     this.mFactor = factor;
04.     updateFold();
05.     invalidate();
06. }

```

ok,这样就完成了引入手指的控制。

2、测试

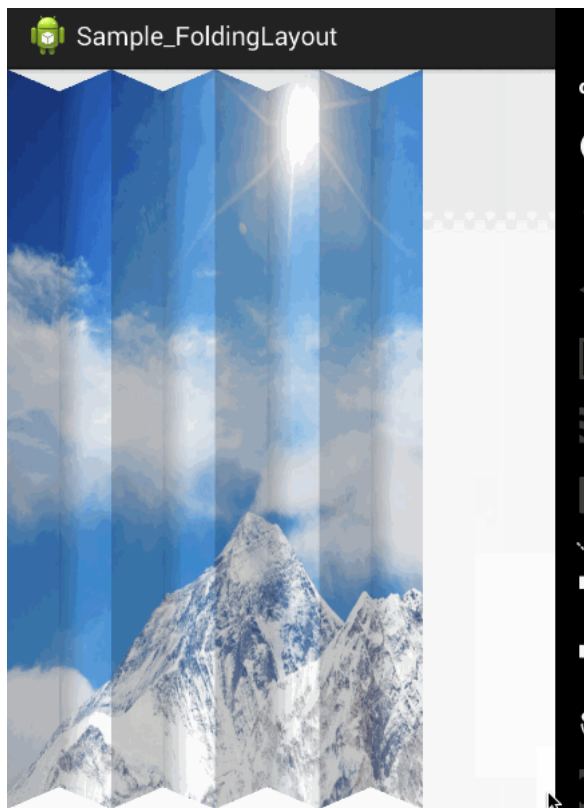
现在改变下布局文件里面的类：

```

[java] C P
01. <com.zhy.view.TouchFoldLayout xmlns:android="http://schemas.android.com/apk/res/android"
02.     xmlns:tools="http://schemas.android.com/tools"
03.     android:id="@+id/id_fold_layout"
04.     android:layout_width="match_parent"
05.     android:layout_height="match_parent" >
06.
07.     <ImageView
08.         android:layout_width="match_parent"
09.         android:layout_height="match_parent"
10.         android:scaleType="fitXY"
11.         android:src="@drawable/xueshan" />
12.
13. </com.zhy.view.TouchFoldLayout>

```

Activity不变，看一下测试效果：（测试前记得把mFactor改为默认值1.0f）



至此我们完成了炫酷的效果，但是我们还需要应用到具体的案例上，否则就是特技，有必要duang一下。

于是我们首先考虑增加到侧滑菜单中去，侧滑菜单有很多选择，google也提供了两个，一个是DrawerLayout，另一个：下面分别展示如何整合入这两个布局。

首先看SlidingPaneLayout，因为DrawerLayout还有些地方需要修改。

4、FoldSlidingPanelLayout

1、实现

对于SlidingPaneLayout的使用，应该没什么问题吧，就是把布局文件的根布局设置为SlidingPaneLayout，然后里面放侧滑菜单，一个代表内容区域。

那么，我们怎么整合到SlidingPaneLayout种去呢？大致两种方法：

- 1、把我们的折叠菜单作为侧滑布局的根布局，然后在Activity种去监听setPanelSlideListener做出改变。
- 2、直接继承SlidingPaneLayout，再其内部将child(0)用FoldLayout包起来，然后监听setPanelSlideListener。

这里我们选择后者，因为后者封装好了，就能直接按照SlidingPaneLayout原本的方式去使用了，不需要做多余的操作。

下面看代码：

```
[java]
01. package com.zhy.view;
02.
03. import android.content.Context;
04. import android.support.v4.widget.SlidingPaneLayout;
05. import android.util.AttributeSet;
06. import android.view.View;
07. import android.view.ViewGroup;
08.
09. public class FoldSlidingPanelLayout extends SlidingPaneLayout
10. {
11.     public FoldSlidingPanelLayout(Context context, AttributeSet attrs)
12.     {
13.         super(context, attrs);
14.     }
15.     @Override
16.     protected void onAttachedToWindow()
17.     {
18.         super.onAttachedToWindow();
19.
20.         View child = getChildAt(0);
21.         if (child != null) {
```



```

22.
23.         removeView(child);
24.         final FoldLayout foldLayout = new FoldLayout(getContext());
25.         //foldLayout.setAnchor(0);
26.         foldLayout.addView(child);
27.         ViewGroup.LayoutParams layPar = child.getLayoutParams();
28.         addView(foldLayout, 0, layPar);
29.
30.         setPanelSlideListener(new PanelSlideListener()
31.         {
32.
33.             @Override
34.             public void onPanelSlide(View arg0, float arg1)
35.             {
36.                 foldLayout.setFactor(arg1);
37.             }
38.
39.             @Override
40.             public void onPanelOpened(View arg0)
41.             {
42.                 // TODO Auto-generated method stub
43.
44.             }
45.
46.             @Override
47.             public void onPanelClosed(View arg0)
48.             {
49.
50.             }
51.         });
52.
53.     }
54. }
55. }

```

我们继承了SlidingPanelLayout，然后在onAttachedToWindow中，取出侧滑的布局，在外层包上一个FoldLayout；并setPanelSlideListener，在onPanelSlide中根据参数，去动态设置FoldLayout的factor。

2、测试

1、布局文件

```

[html] C P
01. <com.zhy.view.FoldSlidingPanelLayout xmlns:android="http://schemas.android.com/apk/res/android"
02.     xmlns:tools="http://schemas.android.com/tools"
03.     android:id="@+id/id_drawerLayout"
04.     android:layout_width="match_parent"
05.     android:layout_height="match_parent" >
06.
07.     <fragment
08.         android:id="@+id/id_left_menu"
09.         android:name="com.zhy.sample.folderlayout.LeftMenuFragment"
10.         android:layout_width="240dp"
11.         android:layout_height="match_parent" />
12.
13.     <RelativeLayout
14.         android:layout_width="match_parent"
15.         android:layout_height="match_parent" >
16.
17.         <ImageView
18.             android:layout_width="match_parent"
19.             android:layout_height="match_parent"
20.             android:scaleType="fitCenter"
21.             android:src="@drawable/xueshan" />
22.     </RelativeLayout>
23.
24. </com.zhy.view.FoldSlidingPanelLayout>

```

我们的菜单使用的是一个fragment。

2、菜单布局

```

[java] C P
01. package com.zhy.sample.folderlayout;
02.
03. import android.os.Bundle;
04. import android.support.v4.app.Fragment;
05. import android.view.LayoutInflater;
06. import android.view.View;
07. import android.view.ViewGroup;

```

```

08. import android.widget.AdapterView;
09. import android.widget.ListView;
10.
11. public class LeftMenuFragment extends Fragment
12. {
13.     private ListView mMenus;
14.     private String[] mMenuItemStr = { "Bear", "Bird", "Cat", "Tigers", "Panda" };
15.
16.     @Override
17.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
18.         Bundle savedInstanceState)
19.     {
20.
21.         View view = inflater.inflate(R.layout.fragment_left_menu, container,
22.             false);
23.         mMenus = (ListView) view.findViewById(R.id.id_left_menu_lv);
24.         mMenus.setAdapter(new ArrayAdapter<String>(getActivity(),
25.             R.layout.item_left_menu, mMenuItemStr));
26.         return view;
27.     }
28. }

```

```

[html]
01. <ListView xmlns:android="http://schemas.android.com/apk/res/android"
02.     xmlns:tools="http://schemas.android.com/tools"
03.     android:id="@+id/id_left_menu_lv"
04.     android:layout_width="240dp"
05.     android:layout_height="match_parent"
06.     android:layout_gravity="start"
07.     android:background="#FFF"
08.     android:choiceMode="singleChoice" />

```

Item就是一个TextView，就不贴了~~

3、Activity

```

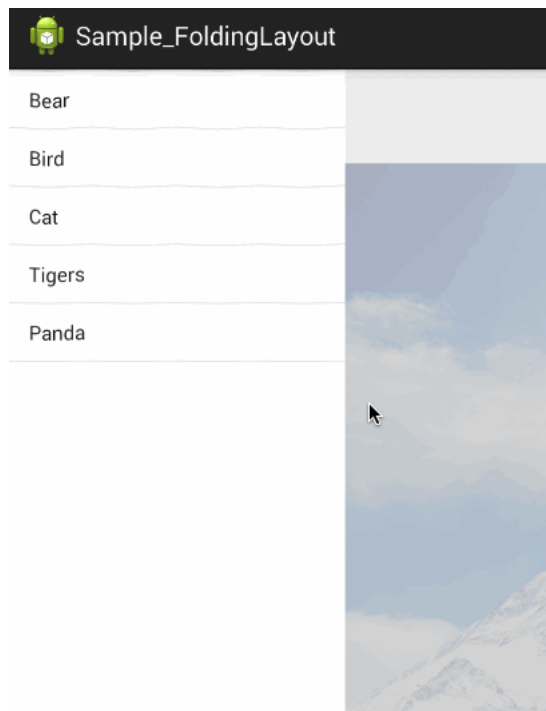
[java]
01. package com.zhy.sample.folderlayout;
02.
03. import android.os.Bundle;
04. import android.support.v4.app.FragmentActivity;
05.
06. public class SlidingPanellayoutSampleActivity extends FragmentActivity
07. {
08.     @Override
09.     protected void onCreate(Bundle arg0)
10.     {
11.         super.onCreate(arg0);
12.         setContentView(R.layout.activity_slidingpanellayout);
13.     }
14. }

```

恩，Activity里面什么都不用做，引入布局文件就行了。

最好看看效果图。

3、效果图



这里菜单块数比较多，大家可以自行修改运行。

ok，到此我们将FoldLayout与SlidingPanelLayout进行了整合，构造了这么个个性的侧滑。

最好还剩下与Drawerlayout的整合。

5、FoldDrawerLayout

1、实现

关于DrawerLayout的使用，与上面的SlidingPanelLayout类似，写写布局文件，引入Activity就好了。我们依然使用上DrawerLayout的子类。

```
[java] C P
01. package com.zhy.view;
02.
03. import android.content.Context;
04. import android.support.v4.view.GravityCompat;
05. import android.support.v4.view.ViewCompat;
06. import android.support.v4.widget.DrawerLayout;
07. import android.util.AttributeSet;
08. import android.util.Log;
09. import android.view.Gravity;
10. import android.view.View;
11. import android.view.ViewGroup;
12.
13. public class FoldDrawerLayout extends DrawerLayout
14. {
15.     private static final String TAG = "DrawerFoldLayout";
16.
17.     public FoldDrawerLayout(Context context, AttributeSet attrs)
18.     {
19.         super(context, attrs);
20.     }
21.
22.     @Override
23.     protected void onAttachedToWindow()
24.     {
25.         super.onAttachedToWindow();
26.
27.         final int childCount = getChildCount();
28.         for (int i = 0; i < childCount; i++)
29.         {
30.             final View child = getChildAt(i);
31.             if (isDrawerView2(child))
32.             {
33.                 Log.e(TAG, "at" + i);
34.                 FoldLayout foldlayout = new FoldLayout(
35.                     getContext());
36.                 //<span style="font-family: Arial, Helvetica, sans-serif;">foldlayout</span><span style="font-
37.                 family: Arial, Helvetica, sans-serif;">.setAnchor(1);</span>
                 removeView(child);
```

```

38.         foldlayout.addView(child);
39.         ViewGroup.LayoutParams layPar = child.getLayoutParams();
40.         addView(foldlayout, i, layPar);
41.     }
42.
43. }
44. setDrawerListener(new DrawerListener()
45. {
46.
47.     @Override
48.     public void onDrawerStateChanged(int arg0)
49.     {
50.         // TODO Auto-generated method stub
51.
52.     }
53.
54.     @Override
55.     public void onDrawerSlide(View drawerView, float slideOffset)
56.     {
57.
58.         if (drawerView instanceof FoldLayout)
59.         {
60.             FoldLayout foldLayout = ((FoldLayout) drawerView);
61.             Log.e(TAG, "slideOffset = " + slideOffset);
62.             foldLayout.setFactor(slideOffset);
63.         }
64.
65.     }
66.
67.     @Override
68.     public void onDrawerOpened(View arg0)
69.     {
70.
71.     }
72.
73.     @Override
74.     public void onDrawerClosed(View arg0)
75.     {
76.
77.     }
78. });
79.
80. }
81.
82. boolean isDrawerView2(View child)
83. {
84.     final int gravity = ((LayoutParams) child.getLayoutParams()).gravity;
85.     final int absGravity = GravityCompat.getAbsoluteGravity(gravity,
86.         ViewCompat.getLayoutDirection(child));
87.     return (absGravity & (Gravity.LEFT | Gravity.RIGHT)) != 0;
88. }
89.
90. }

```

看到这，大家可能会想，然后就和SlidingPanelLayout一样，写写布局文件就好了？其实不是的，如果你这么做了，你因为是这样的：

DraweLayout的侧滑菜单，比如我们拉出来50%，那么正常来说显示的时侧滑布局右侧的50%，但是这个0.5如果设置给布局缩小到50%且在左边。

导致，你拉了50%其实还是上面都看不到，因为折叠到左侧的50%去了。这里依然有两种解决方案：

- 1、结合属性动画，做偏移，具体可参考：[Android DrawerLayout 高仿QQ5.2双向侧滑菜单](#)
- 2、让我们的折叠，收缩到最终的位置可以控制，我们现在统统往最坐标收缩，如果可以设置为最右边，那么本例就没有

2、引入anchor

我们引入一个mAnchor变量，值范围[0, 1]，控制FoldLayout最终折叠到的位置。其实修改的代码比较少，我贴一下修

```

[java]  C  P
01. private void updateFold()
02.     {
03.
04.         //...
05.         float anchorPoint = mAnchor * w;
06.         float midFold = (anchorPoint / mFlodWidth);
07.
08.         for (int i = 0; i < mNumOfFolds; i++)

```

```
09.         {
10.             //引入anchor
11.             dst[0] = (anchorPoint > i * mFoldWidth) ? anchorPoint
12.                 + (i - midFold) * mTranslateDisPerFold : anchorPoint
13.                 - (midFold - i) * mTranslateDisPerFold;
14.             dst[2] = (anchorPoint > (i + 1) * mFoldWidth) ? anchorPoint
15.                 + (i + 1 - midFold) * mTranslateDisPerFold : anchorPoint
16.                 - (midFold - i - 1) * mTranslateDisPerFold;
17.         }
```

唯一改变的就是dst[0]和dst[2]的坐标，当然了，anchor引入以后，你需要判断原始的坐标是否小于anchorPoint，如身量，大于则反之。

记得：

```
[java] C {
01. public void setAnchor(float anchor)
02. {
03.     this.mAnchor = anchor;
04.     updateFold();
05.     invalidate();
06. }
```



上述的FoldDrawerLayout的这行代码：foldlayout.setAnchor(1);让其最后合并位置为右侧。
方式，现在就是写好布局文件，大家直接使用SlidingPanelLayout那个布局文件，改一个根布局类就行。

效果图



到此，我们完成了折叠式菜单从原理到实战的整个过程。当然了，可以完善还有优化的地方很多，大家可以自己去find

源码：[下载](#)

微信公众号请扫描（第一时间通知博客、视频等通知）：



顶

32

踩

2

微信关注我的公众号



上一篇

Android FoldingLayout 折叠布局 原理及实现（一）

下一篇

Android 你应该知道的学习资源 进阶之路贵在坚持

同类文章

【Android 精彩案例】（36）

【Android 自定义控件实战】（28）

- | | | | | |
|-----------------------------|------------|----------|-----------------------------|----------|
| • 巧用ViewPager 打造不一样的广告轮播... | 2016-05-09 | 阅读 32441 | • 如何利用github打造博客专属域名 | 2016-05- |
| • Android 高清加载巨图方案 拒绝压缩图片 | 2015-10-21 | 阅读 50393 | • Android 不规则封闭区域填充 手指秒变... | 2015-05- |
| • Android 不规则图像填充 小玩着色游戏 | 2015-05-18 | 阅读 14992 | • Android FoldingLayout 折 | 15-03- |
| • Android 自定义控件玩转字体变色 打造... | 2015-03-09 | 阅读 30908 | • Android 自定义控件 轻松实 | 15-02- |
| • Android 打造形形色色的进度条 实现可... | 2015-02-02 | 阅读 45553 | • Android 打造炫目的圆形菜单 秒秒钟高... | 2015-01- |

更多文章

猜你在找

Android自动化测试第二季（提高...

Android无障碍测试环境搭建和使...

Android自动化测试第三季

Android开发—SlidingMenu 侧滑...

【Android APP开发】Android高级...

Android FoldingLayout 折叠布局··

Android FoldingLayout 折叠布局··

Android FoldingLayout 折叠布局··

Android FoldingLayout 折叠布局··

Android FoldingLayout 折叠布局··

查看评论



烽火尖

31楼 20

FoldDrawerLayout中的dst[0]以及 dst[2] 是怎么算的？想了一天了，没搞明白，求指导




zhangrh528

30楼 20

只能膜拜

qq_33805633

29楼 20



请收下我的膝盖！！！！



大西北的风

28楼 2016-09-05 14:00


不错，学到了，支持。



qq_27761325


27楼 2016-09-05 13:59

请问 **drawlist**没打开 在主界面点击 获取到的焦点是**drawlist**的。 怎么处理掉这个BUG呢



Dreamer820978076

为什么只能显示八块中的一块的？求解释！谢谢



zhou314655435

回复Dreamer820978076：第一次绘制的时候要调用：`super.dispatchDraw(mCanvas)`；注意这里用的是**mCanvas**，如果写的是**canvas**那就只有一块了

微信关注我的公众号







ANDRNDR

25楼 2016-09-05 13:58

这个太666



一支pds

24楼 2016-09-05 13:57

大神



linxingqianglai

23楼 2016-09-05 13:56


DrawerLayoutSampleActivity为什么没出来效果啊



Faith_hb

22楼 2016-09-05 13:55

vivo xshot 上没有折叠效果



护士leo

21楼 2016-09-05 13:54

大神，膜拜



开发者_android

20楼 2016-09-05 13:53

原因应该是所有值计算过程中都是整除中产生的误差导致。



开发者_android

19楼 2016-09-05 13:52


FoldLayoutActivity向da滑动时，当TouchFoldLayout中**factor**值大小在小数点后第三位范围时会出现bug。既：mTranslation的值在1-8的范围时。



开发者_android

18楼 2016-09-05 13:51

洋哥，越来越不吐鸟了啊。
这效果给力呀哈～～



花-开-花-谢

17楼 2016-09-05 13:50

亲，关注你很久了，。能不能讲讲抽象，接口在项目中的实践，及优点，怎么编写优化的代码框架，自己在项目中几乎没用到接口，很少用到抽象，莫非自己是对他们呢不了解?呵呵



shaziever123

16楼 2016-09-05 13:49

大神慕课网的视频讲的太好了 对Android理解更深了 非常感谢 大神能否抽时间讲讲Android studio的gradle



MJ_csdn


15楼 2016-09-05 13:48

洋哥 v587 就是那么喜欢你的博客

cq522506604

14楼 2016-09-05 14:00

请问 效果图怎么制作的 啊

怀柔老纪


13楼 2016-09-05 13:00

在红米手机上，发现右侧的图不显示，不知道是不是系统兼容性问题


Qiujuer

12楼 2016-09-05 12:00

非常OK 的效果，要是再优化优化基本就比较完善了。

韩曙亮

好炫的效果 顶了

hondyyu

效果看着挺好的。。



Susan8888

9楼 2016-09-05 09:00

已经 不错了

wubp


8楼 2016-09-05 08:00

顶

Ssshawnnn


7楼 2016-09-05 07:00

源码下载链接没有么？

birogul

6楼 2016-09-05 06:00

鸿洋_ i want change folding layout from right side to left side
it it possible?

鸿洋_

Re: 2016-09-05 05:00

回复birogul: just use method setAnchor(int value) , value's range between [0 , maxWidth];

志在愚乐

5楼 2016-09-05 05:00

大神~~~~~

一子非鱼一

4楼 2016-09-05 04:00

前排占座、

征_小站

3楼 2016-09-05 03:00

前排，先顶再看

Hi-Daily

2楼 2016-09-05 02:00

正在扇耳朵。。。

eclipse_xu


1楼 2016-09-05 01:00

没曲线，差评！赶紧搞定7哥的特效

鸿洋_

Re: 2016-09-05 00:00

回复eclipse_xu: 要不要这么快~~~我要发扇耳朵的表情~~

 **MrSimp1e**

回复鸿洋_: 给个好评！5分，下次包邮噢亲！

Re: 20

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题	Hadoop	AWS	移动游戏	Java	Android	iOS	Swift	智能硬件	
IE10	Eclipse	CRM	JavaScript	数据库	Ubuntu	NFC	WAP	jQuery	BI
HTML	SDK	IIS	Fedora	XML	LBS	Unity	Splashtop	UML	component
Cassandra	CloudStack	FTC	coremail	OPhone	CouchBase	云计算	iOS6		
Compuware	大数据	aptech	Perl	Tornado	Ruby	Hibernate	ThinkPHP		
Foundry	Redis	Scala	Django	Bootstrap					

微信关注我的公众号



公司简介 |

网站客服
京 ICP 证 09



| [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
CSDN.NET, All Rights Reserved 