

上一篇[Android 属性动画（Property Animation）完全解析（上）](#)已经基本展示了属性动画的核心用法：

ObjectAnimator实现动画，ValueAnimator实现动画，AnimatorSet的使用等~

当然了属性动画还有一部分的知识点，也能做出很不错的效果，将在本篇博客为您展示~

1、如何使用xml文件来创建属性动画

大家肯定都清楚，View Animator、Drawable Animator都可以在anim文件夹下创建动画，然后在程序中使用，甚至在Theme中设置为属性值。当然了，属性动画其实也可以在文件中声明：

首先在res下建立animator文件夹，然后建立res/animator/scalex.xml

[html] view plaincopy

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <objectAnimator xmlns:android="http://
   schemas.android.com/apk/res/android"
3.     android:duration="1000"
4.     android:propertyName="scaleX"
5.     android:valueFrom="1.0"
6.     android:valueTo="2.0"
7.     android:valueType="floatType" >
8. </objectAnimator>
```

代码：

[java] view plaincopy

```
1. public void scaleX(View view)
2.     {
3.         // 加载动画
4.
5.         Animator anim = AnimatorInflater.loadAnimator(this, R.animator.scalex);
6.         anim.setTarget(mMv);
7.         anim.start();
8.     }
```

使用AnimatorInflater加载动画的资源文件，然后设置目标，就ok~~是不是很简单，这只是单纯横向的放大一倍~

如果我希望纵向与横向同时缩放呢？则可以怎么定义属性文件：

[html] view plaincopy

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <set xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:ordering="together" >
4.
5.     <objectAnimator
6.         android:duration="1000"
7.         android:propertyName="scaleX"
8.         android:valueFrom="1"
9.         android:valueTo="0.5" >
```

```

10.     </objectAnimator>
11.     <objectAnimator
12.         android:duration="1000"
13.         android:propertyName="scaleY"
14.         android:valueFrom="1"
15.         android:valueTo="0.5" >
16.     </objectAnimator>
17.
18. </set>

```

使用set标签，有一个orderring属性设置为together, 【还有另一个值：sequentially（表示一个接一个执行）】。

上篇博客中忽略了一个效果，就是缩放、反转等都有中心点或者轴，默认中心缩放，和中间对称线为反转线，所以我决定这个横向，纵向缩小以左上角为中心点：

代码：

[java] view plaincopy

```

1.  // 加载动画
2.
           Animator anim = AnimatorInflater.loadAnimator(thi
s, R.animator.scale);
3.         mMv.setPivotX(0);
4.         mMv.setPivotY(0);
5.         //显示的调用invalidate
6.         mMv.invalidate();
7.         anim.setTarget(mMv);
8.         anim.start();

```

很简单，直接给View设置pivotX和pivotY，然后调用一下invalidate，就ok了。

下面看效果图：

好了，通过写xml声明动画，使用set嵌套set，结合orderring属性，也基本可以实现任何动画~~上面也演示了pivot的设置。

2、布局动画（Layout Animations）

主要使用LayoutTransition为布局的容器设置动画，当容器中的视图层次发生变化时存在过渡的动画效果。

基本代码为：

[java] view plaincopy

```
1. LayoutTransition transition = new LayoutTransition();
2.
   transition.setAnimator(LayoutTransition.CHANGE_APPEAR
   ING,
3.
   transition.getAnimator(LayoutTransition.CHANG
   E_APPEARING));
4.   transition.setAnimator(LayoutTransition.APPEARING,
5.   null);
6.
   transition.setAnimator(LayoutTransition.DISAPPEARING,
```

```
7.         null);  
8.  
           transition.setAnimator(LayoutTransition.CHANGE_DISAPPEARING,  
9.         null);  
10.    mGridLayout.setLayoutTransition(transition);
```

过渡的类型一共有四种：

`LayoutTransition.APPEARING` 当一个View在ViewGroup中出现时，对此**View**设置的动画

`LayoutTransition.CHANGE_APPEARING` 当一个View在ViewGroup中出现时，对此View对其他View位置造成影响，对其他**View**设置的动画

`LayoutTransition.DISAPPEARING` 当一个View在ViewGroup中消失时，对此**View**设置的动画

`LayoutTransition.CHANGE_DISAPPEARING` 当一个View在ViewGroup中消失时，对此View对其他View位置造成影响，对其他**View**设置的动画

`LayoutTransition.CHANGE` 不是由于View出现或消失造成对其他View位置造成影响，然后对其他View设置的动画。

注意动画到底设置在谁身上，此View还是其他View。

好了下面看一个综合的例子：

布局文件：

[html] view plaincopy

```
1. <LinearLayout xmlns:android="http://schemas.android.com/
   apk/res/android"
2.     xmlns:tools="http://schemas.android.com/tools"
3.     android:id="@+id/id_container"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:orientation="vertical" >
7.
8.     <Button
9.         android:layout_width="wrap_content"
10.        android:layout_height="wrap_content"
11.        android:onClick="addBtn"
12.        android:text="addBtns" />
13.
14.    <CheckBox
15.        android:id="@+id/id_appear"
16.        android:layout_width="wrap_content"
17.        android:layout_height="wrap_content"
18.        android:checked="true"
19.        android:text="APPEARING" />
20.
21.    <CheckBox
22.        android:id="@+id/id_change_appear"
23.        android:layout_width="wrap_content"
24.        android:layout_height="wrap_content"
25.        android:checked="true"
26.        android:text="CHANGE_APPEARING" />
27.
28.    <CheckBox
29.        android:id="@+id/id_disappear"
```

```

30.         android:layout_width="wrap_content"
31.         android:layout_height="wrap_content"
32.         android:checked="true"
33.         android:text="DISAPPEARING" />
34.
35.     <CheckBox
36.         android:id="@+id/id_change_disappear"
37.         android:layout_width="wrap_content"
38.         android:layout_height="wrap_content"
39.         android:checked="true"
40.         android:text="CHANGE_DISAPPEARING " />
41.
42. </LinearLayout>

```

代码:

[java] view plaincopy

```

1. package com.example.zhy_property_animation;
2.
3. import android.animation.LayoutTransition;
4. import android.app.Activity;
5. import android.os.Bundle;
6. import android.view.View;
7. import android.view.View.OnClickListener;
8. import android.view.ViewGroup;
9. import android.widget.Button;
10. import android.widget.CheckBox;
11. import android.widget.CompoundButton;
12.
13.     import android.widget.CompoundButton.OnCheckedChangeListener;
14.
15. import android.widget.GridLayout;
16.

```

15.

```
public class LayoutAnimaActivity extends Activity implements
```

```
    OnCheckedChangeListener
```

```
{
```

```
    private ViewGroup viewGroup;
```

```
    private GridLayout mGridLayout;
```

```
    private int mVal;
```

```
    private LayoutTransition mTransition;
```

```
    private CheckBox mAppear, mChangeAppear, mDisAppear,  
        mChangeDisAppear;
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState)
```

```
    {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.layout_animator);
```

```
        viewGroup = (ViewGroup) findViewById(R.id.id_cont  
ainer);
```

```
        mAppear = (CheckBox) findViewById(R.id.id_appear)  
        ;
```

```
        mChangeAppear = (CheckBox) findViewById(R.id.id_c  
hange_appear);
```

```
        mDisAppear = (CheckBox) findViewById(R.id.id_disa  
pppear);
```

```
        mChangeDisAppear = (CheckBox) findViewById(R.id.i  
d_change_disappear);
```

```
        mAppear.setOnCheckedChangeListener(this);
```

```
        mChangeAppear.setOnCheckedChangeListener(this);
```

```
        mDisAppear.setOnCheckedChangeListener(this);
```



```

40.         mChangeDisAppear.setOnCheckedChangeListener(this)
41.     ;
42.         // 创建一个GridLayout
43.         mGridLayout = new GridLayout(this);
44.         // 设置每列5个按钮
45.         mGridLayout.setColumnCount(5);
46.         // 添加到布局中
47.         viewGroup.addView(mGridLayout);
48.         //默认动画全部开启
49.         mTransition = new LayoutTransition();
50.         mGridLayout.setLayoutTransition(mTransition);
51.
52.     }
53.
54.     /**
55.      * 添加按钮
56.      *
57.      * @param view
58.      */
59.     public void addBtn(View view)
60.     {
61.         final Button button = new Button(this);
62.         button.setText(++mVal + "");
63.
64.         mGridLayout.addView(button, Math.min(1, mGridLayout.getChildCount()));
65.         button.setOnClickListener(new OnClickListener()
66.         {
67.             @Override
68.             public void onClick(View v)
69.             {
70.                 mGridLayout.removeView(button);
71.             }
72.         });
73.     }
74.

```

```

75.     @Override
76.
77.         public void onCheckedChanged(CompoundButton buttonVie
w, boolean isChecked)
78.         {
79.             mTransition = new LayoutTransition();
80.             mTransition.setAnimator(
81.                 LayoutTransition.APPEARING,
82.                 (mAppear.isChecked() ? mTransition
83.
84.                     .getAnimator(LayoutTransition.APP
85.
86.                         EARING) : null));
87.             mTransition
88.                 .setAnimator(
89.
90.                     LayoutTransition.CHANGE_APPEARING
91.
92.
93.
94.
95.
96.
97.
98.
99.
100.
101.
102.
103.
104.
105.
106.
107.
108.
109.
110.
111.
112.
113.
114.
115.
116.
117.
118.
119.
120.
121.
122.
123.
124.
125.
126.
127.
128.
129.
130.
131.
132.
133.
134.
135.
136.
137.
138.
139.
140.
141.
142.
143.
144.
145.
146.
147.
148.
149.
150.
151.
152.
153.
154.
155.
156.
157.
158.
159.
160.
161.
162.
163.
164.
165.
166.
167.
168.
169.
170.
171.
172.
173.
174.
175.
176.
177.
178.
179.
180.
181.
182.
183.
184.
185.
186.
187.
188.
189.
190.
191.
192.
193.
194.
195.
196.
197.
198.
199.
200.
201.
202.
203.
204.
205.
206.
207.
208.
209.
210.
211.
212.
213.
214.
215.
216.
217.
218.
219.
220.
221.
222.
223.
224.
225.
226.
227.
228.
229.
230.
231.
232.
233.
234.
235.
236.
237.
238.
239.
240.
241.
242.
243.
244.
245.
246.
247.
248.
249.
250.
251.
252.
253.
254.
255.
256.
257.
258.
259.
260.
261.
262.
263.
264.
265.
266.
267.
268.
269.
270.
271.
272.
273.
274.
275.
276.
277.
278.
279.
280.
281.
282.
283.
284.
285.
286.
287.
288.
289.
290.
291.
292.
293.
294.
295.
296.
297.
298.
299.
300.
301.
302.
303.
304.
305.
306.
307.
308.
309.
310.
311.
312.
313.
314.
315.
316.
317.
318.
319.
320.
321.
322.
323.
324.
325.
326.
327.
328.
329.
330.
331.
332.
333.
334.
335.
336.
337.
338.
339.
340.
341.
342.
343.
344.
345.
346.
347.
348.
349.
350.
351.
352.
353.
354.
355.
356.
357.
358.
359.
360.
361.
362.
363.
364.
365.
366.
367.
368.
369.
370.
371.
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.
389.
390.
391.
392.
393.
394.
395.
396.
397.
398.
399.
400.
401.
402.
403.
404.
405.
406.
407.
408.
409.
410.
411.
412.
413.
414.
415.
416.
417.
418.
419.
420.
421.
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489.
490.
491.
492.
493.
494.
495.
496.
497.
498.
499.
500.
501.
502.
503.
504.
505.
506.
507.
508.
509.
510.
511.
512.
513.
514.
515.
516.
517.
518.
519.
520.
521.
522.
523.
524.
525.
526.
527.
528.
529.
530.
531.
532.
533.
534.
535.
536.
537.
538.
539.
540.
541.
542.
543.
544.
545.
546.
547.
548.
549.
550.
551.
552.
553.
554.
555.
556.
557.
558.
559.
560.
561.
562.
563.
564.
565.
566.
567.
568.
569.
570.
571.
572.
573.
574.
575.
576.
577.
578.
579.
580.
581.
582.
583.
584.
585.
586.
587.
588.
589.
590.
591.
592.
593.
594.
595.
596.
597.
598.
599.
600.
601.
602.
603.
604.
605.
606.
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.
618.
619.
620.
621.
622.
623.
624.
625.
626.
627.
628.
629.
630.
631.
632.
633.
634.
635.
636.
637.
638.
639.
640.
641.
642.
643.
644.
645.
646.
647.
648.
649.
650.
651.
652.
653.
654.
655.
656.
657.
658.
659.
660.
661.
662.
663.
664.
665.
666.
667.
668.
669.
670.
671.
672.
673.
674.
675.
676.
677.
678.
679.
680.
681.
682.
683.
684.
685.
686.
687.
688.
689.
690.
691.
692.
693.
694.
695.
696.
697.
698.
699.
700.
701.
702.
703.
704.
705.
706.
707.
708.
709.
710.
711.
712.
713.
714.
715.
716.
717.
718.
719.
720.
721.
722.
723.
724.
725.
726.
727.
728.
729.
730.
731.
732.
733.
734.
735.
736.
737.
738.
739.
740.
741.
742.
743.
744.
745.
746.
747.
748.
749.
750.
751.
752.
753.
754.
755.
756.
757.
758.
759.
760.
761.
762.
763.
764.
765.
766.
767.
768.
769.
770.
771.
772.
773.
774.
775.
776.
777.
778.
779.
780.
781.
782.
783.
784.
785.
786.
787.
788.
789.
790.
791.
792.
793.
794.
795.
796.
797.
798.
799.
800.
801.
802.
803.
804.
805.
806.
807.
808.
809.
810.
811.
812.
813.
814.
815.
816.
817.
818.
819.
820.
821.
822.
823.
824.
825.
826.
827.
828.
829.
830.
831.
832.
833.
834.
835.
836.
837.
838.
839.
840.
841.
842.
843.
844.
845.
846.
847.
848.
849.
850.
851.
852.
853.
854.
855.
856.
857.
858.
859.
860.
861.
862.
863.
864.
865.
866.
867.
868.
869.
870.
871.
872.
873.
874.
875.
876.
877.
878.
879.
880.
881.
882.
883.
884.
885.
886.
887.
888.
889.
890.
891.
892.
893.
894.
895.
896.
897.
898.
899.
900.
901.
902.
903.
904.
905.
906.
907.
908.
909.
910.
911.
912.
913.
914.
915.
916.
917.
918.
919.
920.
921.
922.
923.
924.
925.
926.
927.
928.
929.
930.
931.
932.
933.
934.
935.
936.
937.
938.
939.
940.
941.
942.
943.
944.
945.
946.
947.
948.
949.
950.
951.
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963.
964.
965.
966.
967.
968.
969.
970.
971.
972.
973.
974.
975.
976.
977.
978.
979.
980.
981.
982.
983.
984.
985.
986.
987.
988.
989.
990.
991.
992.
993.
994.
995.
996.
997.
998.
999.

```

```
98.         mGridLayout.setLayoutTransition(mTransition);
99.     }
100. }
```

效果图：

动画有点长，耐心点看，一定要注意，是对当前View还是其他Views设置的动画。

当然了动画支持自定义，还支持设置时间，比如我们修改下，添加的动画为：

[java] view plaincopy

```
1.
    mTransition.setAnimator(LayoutTransition.APPEARING, (mApp
ear
2.
                                .isChecked() ? ObjectAnimator.ofFloat(thi
s, "scaleX", 0, 1)
3.
                                : null));
```

则效果为：

原本的淡入，变成了宽度从中间放大的效果~~是不是还不错~~

3、View的anim方法

在SDK11的时候，给View添加了animate方法，更加方便的实现动画效果。

布局文件：

[html] view plaincopy

```
1. <RelativeLayout xmlns:android="http://
   schemas.android.com/apk/res/android"
2.     xmlns:tools="http://schemas.android.com/tools"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     >
6.
7.     <ImageView
8.         android:id="@+id/id_ball"
9.         android:layout_width="wrap_content"
10.        android:layout_height="wrap_content"
11.        android:src="@drawable/bol_blue" />
12.
13.    <LinearLayout
14.        android:layout_width="fill_parent"
15.        android:layout_height="wrap_content"
16.        android:layout_alignParentBottom="true"
17.        android:orientation="horizontal" >
18.
19.        <Button
20.            android:layout_width="wrap_content"
```

```

21.         android:layout_height="wrap_content"
22.         android:onClick="viewAnim"
23.         android:text="View Anim" />
24.
25.     <Button
26.         android:layout_width="wrap_content"
27.         android:layout_height="wrap_content"
28.         android:onClick="propertyValuesHolder"
29.         android:text="PropertyValuesHolder " />
30.
31.
32. </LinearLayout>
33.
34. </RelativeLayout>

```

代码:

[java] view plaincopy

```

1. package com.example.zhy_property_animation;
2.
3. import android.animation.ObjectAnimator;
4. import android.animation.PropertyValuesHolder;
5. import android.app.Activity;
6. import android.os.Bundle;
7. import android.util.DisplayMetrics;
8. import android.util.Log;
9. import android.view.View;
10. import android.widget.ImageView;
11.
12. public class ViewAnimateActivity extends Activity
13. {
14.
15.     protected static final String TAG = "ViewAnimateActiv
16.     ity";

```

```

15.
16.     private ImageView mBlueBall;
17.     private float mScreenHeight;
18.
19.     @Override
20.     protected void onCreate(Bundle savedInstanceState)
21.     {
22.         super.onCreate(savedInstanceState);
23.         setContentView(R.layout.view_animator);
24.
25.         DisplayMetrics outMetrics = new DisplayMetrics();

26.         getWindowManager().getDefaultDisplay().getMetrics
            (outMetrics);
27.         mScreenHeight = outMetrics.heightPixels;
28.         mBlueBall = (ImageView) findViewById(R.id.id_ball
            );
29.
30.     }
31.
32.     public void viewAnim(View view)
33.     {
34.         // need API12
35.         mBlueBall.animate()//
36.             .alpha(0)//
37.             .y(mScreenHeight / 2).setDuration(1000)
38.             // need API 12
39.             .withStartAction(new Runnable()
40.             {
41.                 @Override
42.                 public void run()
43.                 {
44.                     Log.e(TAG, "START");
45.                 }
46.                 // need API 16
47.             }).withEndAction(new Runnable()

```

```

48.         {
49.
50.             @Override
51.             public void run()
52.             {
53.                 Log.e(TAG, "END");
54.                 runOnUiThread(new Runnable()
55.                 {
56.                     @Override
57.                     public void run()
58.                     {
59.                         mBlueBall.setY(0);
60.                         mBlueBall.setAlpha(1.0f);
61.                     }
62.                 });
63.             }
64.         }).start();
65.     }
}

```

简单的使用mBlueBall.animate().alpha(0).y(mScreenHeight / 2).setDuration(1000).start()就能实现动画~~不过需要SDK11，此后在SDK12，SDK16又分别添加了withStartAction和withEndAction用于在动画前，和动画后执行一些操作。当然也可以.setListener(listener)等操作。

使用ObjectAnimator实现上面的变化，我们可以使用：
PropertyViewHolder

[java] view plaincopy

```
1.
    PropertyValuesHolder pvhX = PropertyValuesHolder.ofFloat(
        "alpha", 1f,
2.
        0f, 1f);
3.
    PropertyValuesHolder pvhY = PropertyValuesHolder.ofFloat("y", 0,
4.
        mScreenHeight / 2, 0);
5.
    ObjectAnimator.ofPropertyValuesHolder(mBlueBall, pvhX
        , pvhY).setDuration(1000).start();
```

效果与上面一样。