# **十分钟搞定酷炫动画，Android自定义 View 入门**

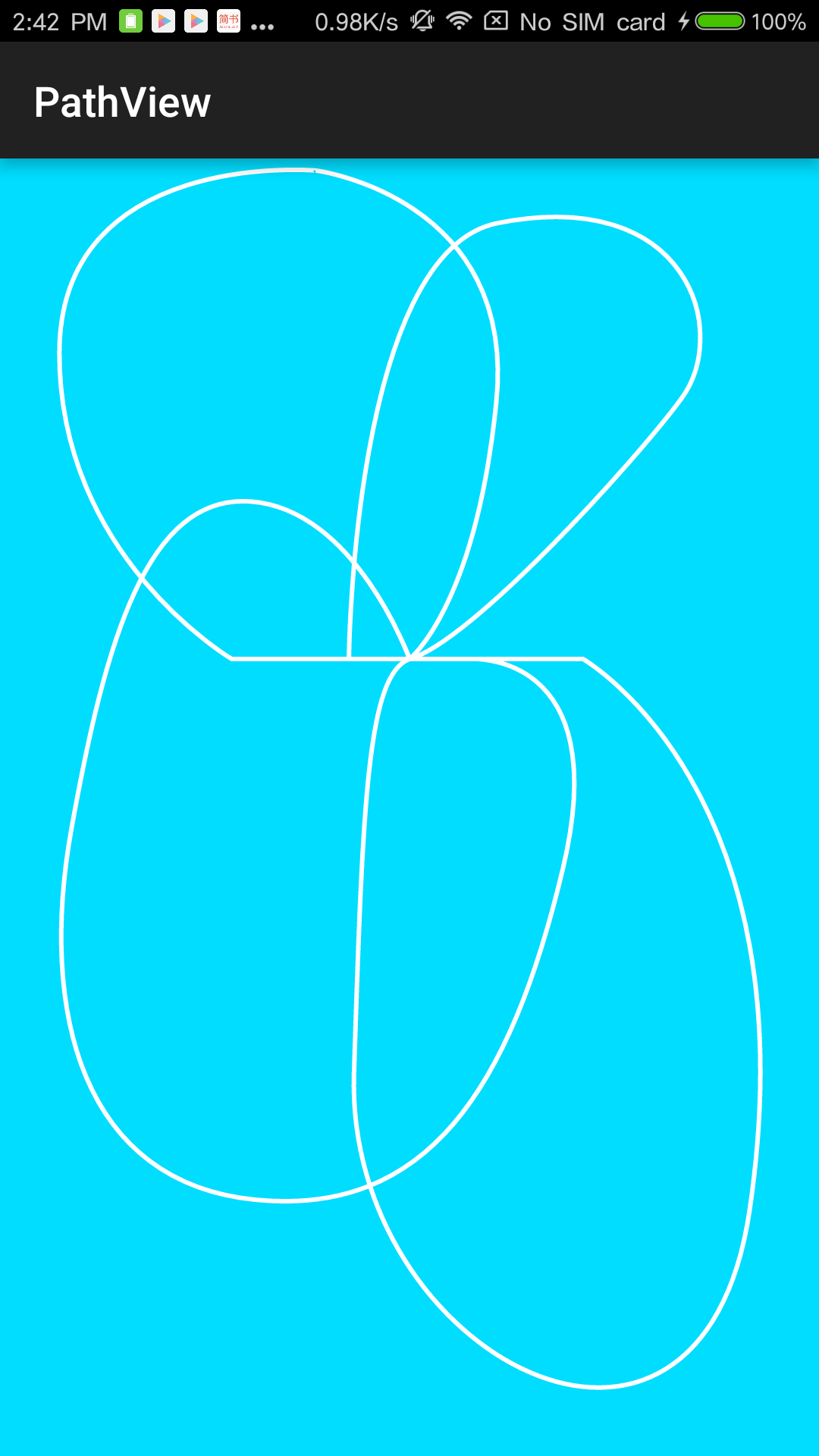
首先对着 gif 图一帧一帧观察了一遍，分析动画的过程。把动画拆解成两部分。

1、四个颜色的圆运动。

2、Logo的出现

logo 的出现就是简单的alpha 动画，难点就在四个圆运动。

找 Ui 拿到了四个圆的运动轨迹，如下图所示：



根据轨迹，我把运动轨迹拆分成平移和半圆旋转，创建出Path路径，再让圆沿着 Path 运动，在运动的时候加上 alpha 和缩放的属性，结束的时候把圆移除掉并显示 logo 就好。

分析结束，接下来就上代码：

第一步：创建 LauncherView 继承 RelativeLayout，在构造方法里面 init（）添加四个颜色的圆

private void init(){

LayoutParamslp=newLayoutParams(ViewGroup.LayoutParams.WRAP\_CONTENT,ViewGroup.LayoutParams.WRAP\_CONTENT);

lp.addRule(CENTER\_HORIZONTAL,TRUE);//这里的TRUE 要注意 不是true

lp.addRule(CENTER\_VERTICAL,TRUE);

lp.setMargins(0,0,0,dp80);

purple=newImageView(getContext());

purple.setLayoutParams(lp);

purple.setImageDrawable(getResources().getDrawable(R.drawable.shape\_circle\_purple));

addView(purple);

yellow=newImageView(getContext());

yellow.setLayoutParams(lp);

yellow.setImageDrawable(getResources().getDrawable(R.drawable.shape\_circle\_yellow));

addView(yellow);

blue=newImageView(getContext());

blue.setLayoutParams(lp);

blue.setImageDrawable(getResources().getDrawable(R.drawable.shape\_circle\_blue));

addView(blue);

red=newImageView(getContext());

red.setLayoutParams(lp);

red.setImageDrawable(getResources().getDrawable(R.drawable.shape\_circle\_red));

addView(red);

}

ObjectAnimator动画原理

ObjectAnimator.ofObject(….,”xxx”,估值值,区间数组); 【定义动画属性xxx和区间】

插值器/加速器（Interpolator）【返回当前数字进度t】

估值值（Evaluator）【根当前数字进度计算并返回当前值】

调用setXxx函数 【根据封装好的setXxx函数并反射调用，将第三步返回当前值以参数传入】

*[自定义ObjectAnimator属性请戳这里](http://blog.csdn.net/cxmscb/article/details/52485878" \t "_blank)*

绘制四个圆圈的 Path，这里以红色圆圈为例，三阶贝塞尔曲线描点不会的话，可以参考 Ui 的设计路径描点，Ps 的钢笔工具就是贝塞尔曲线。

ViewPathredPath1=newViewPath();//偏移坐标

redPath1.moveTo(0,0);

redPath1.lineTo(mWidth/5-mWidth/2,0);

ViewPathredPath2=newViewPath();

redPath2.moveTo(mWidth/5-mWidth/2,0);

redPath2.curveTo(-700,-mHeight/2,mWidth/3\*2,-mHeight/3\*2,0,-dp80);

setAnimation(red,redPath1,redPath2);

接下来将 Path 转换成 ObjectAnimation

private void setAnimation(final ImageView target,ViewPath path1,ViewPath path2){

//左右平移

ObjectAnimator redAnim1=ObjectAnimator.ofObject(newViewObj(target),"fabLoc",newViewPathEvaluator(),path1.getPoints().toArray());

redAnim1.setInterpolator(newAccelerateDecelerateInterpolator());

redAnim1.setDuration(800);

//贝塞尔曲线

ObjectAnimator redAnim2=ObjectAnimator.ofObject(newViewObj(target),"fabLoc",newViewPathEvaluator(),path2.getPoints().toArray());

redAnim2.setInterpolator(newAccelerateDecelerateInterpolator());

//组合动画

addAnimation(redAnim1,redAnim2,target);

}

然后组合动画得到一个圆的完整运行轨迹。

private void addAnimation(ObjectAnimatoranimator1,ObjectAnimatoranimator2,ImageViewtarget){

ObjectAnimator alpha=ObjectAnimator.ofFloat(target,View.ALPHA,1f,0.5f);

ObjectAnimator scaleX=ObjectAnimator.ofFloat(target,View.SCALE\_X,1,getScale(target),1.0f);

ObjectAnimator scaleY=ObjectAnimator.ofFloat(target,View.SCALE\_Y,1,getScale(target),1.0f);

AnimatorSet all2=newAnimatorSet();

all2.setDuration(1800);

all2.playTogether(alpha,scaleX,scaleY,animator2);

all2.addListener(newAnimEndListener(target));

AnimatorSetall=newAnimatorSet();

all.playSequentially(animator1,all2);

all.start();

}

最后，显示 logo 动画

private void showLogo(){

Viewview=View.inflate(getContext(),R.layout.widget\_load\_view,this);

Viewlogo=view.findViewById(R.id.iv\_logo);

finalViewslogo=view.findViewById(R.id.iv\_slogo);

ObjectAnimatoralpha=ObjectAnimator.ofFloat(logo,View.ALPHA,0f,1f);

alpha.setDuration(800);

alpha.start();

newHandler().postDelayed(newRunnable(){

@Override

public voidrun(){

ObjectAnimator alpha=ObjectAnimator.ofFloat(slogo,View.ALPHA,0f,1f);

alpha.setDuration(200);

alpha.start();

}

},400);

}

好，到这里，炫酷的启动页动画已经撸出来了，就是一个简单的ObjectAnimator使用，大家不要被自定义 view 这个纸老虎吓到。