**Canvas 缩放（Scale）**

**前言：**前几天用到Canvas.scale(flostsx, float sy, float px, float py)函数，研究源码后没有看懂，就去网上找资料，发现关于Canvas.scale(flost sx, float sy, float px, float py)的分析很少。经过一天的研究，在此分享一下个人对此的理解，欢迎与各位交流。

本文只介绍Canvas的缩放（scale），关于平移（translate）和旋转（rotate）推荐看以下文章：<http://blog.csdn.net/harvic880925/article/details/39080931>

**一. 缩放（Scale）**

Canvas缩放有以下两个方法：

public void scale (float sx, floatsy) ;

public final void scale (float sx,float sy, float px, float py);

1.    publicvoid scale (float sx, float sy) ;

Paint mPaint= new Paint();  
canvas.drawColor(Color.*BLUE*);  
mPaint.setColor(Color.*GRAY*);  
canvas.drawRect(new Rect(0, 0, 400, 400), mPaint);  
  
// 保存画布状态  
canvas.save();  
canvas.scale(0.5f, 0.5f);  
mPaint.setColor(Color.*YELLOW*);  
canvas.drawRect(new Rect(0, 0, 400, 400), mPaint);  
// 画布状态回滚  
canvas.restore();

效果就相当于用个钉子钉在(0,0)处，然后把矩形的x，y缩放为一半，如下图所示:



2.    publicfinal void scale (float sx, float sy, float px, float py);

Paint mPaint = new Paint();

canvas.drawColor(Color.*BLUE*);

mPaint.setColor(Color.*GRAY*);

canvas.drawRect(new Rect(0, 0, 400, 400), mPaint);

// 保存画布状态

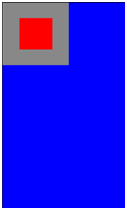
canvas.save();

canvas.scale(0.5f, 0.5f, 200, 200);

mPaint.setColor(Color.*RED*);

canvas.drawRect(new Rect(0, 0, 400, 400), mPaint);

前两个参数为将画布在x、y方向上缩放的倍数，而px和py 分别为缩放的基准点，如下图所示：



**二. 重点来了**

Canvas.scale (float sx, float sy, float px, float py) 源码如下：

*/\*\**

*\* Preconcat the current matrix with the specified scale.*

*\**

*\** ***@param*** *sx The amount to scale in X*

*\** ***@param*** *sy The amount to scale in Y*

*\** ***@param*** *px The x-coord for the pivot point (unchanged by the scale)*

*\** ***@param*** *py The y-coord for the pivot point (unchanged by the scale)*

*\*/*

public final void scale(float sx, float sy, float px, float py) {

    translate(px, py);

    scale(sx, sy);

    translate(-px, -py);

}

translate(px,py);

scale(sx,sy);

translate(-px,-py);   …………..I

和

scale(sx,sy);        …………..II

不是一样的吗？为什么显示的效果不同？

原因是translate(px, py)移动的物理距离分别是px和py，经过scale(sx, sy)缩放后再通过translate(-px, -py)位移，移动的物理距离就是-px\*sx和-py\*sy。

我个人的理解是scale(sx, sy)缩放类似于px转换成dp的过程，才会出现I和II的效果不同。

**三. 验证**

1. 先translate(float fx, float fy)位移，后scale(float sx, float sy)缩放

Paint mPaint = new Paint();

canvas.drawColor(Color.*BLUE*);

mPaint.setColor(Color.*GRAY*);

canvas.drawRect(new Rect(0, 0, 400, 400), mPaint);

// 保存画布状态

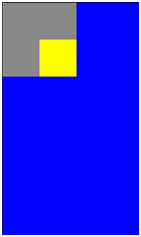
canvas.save();

canvas.translate(200, 200);

canvas.scale(0.5f, 0.5f);

mPaint.setColor(Color.*YELLOW*);

canvas.drawRect(new Rect(0, 0, 400, 400), mPaint);



2. 先scale(float sx, float sy)缩放，后translate(float fx, float fy)位移

Paint mPaint = new Paint();

canvas.drawColor(Color.*BLUE*);

mPaint.setColor(Color.*GRAY*);

canvas.drawRect(new Rect(0, 0, 400, 400), mPaint);

// 保存画布状态

canvas.save();

canvas.scale(0.5f, 0.5f);

canvas.translate(200, 200);

mPaint.setColor(Color.*YELLOW*);

canvas.drawRect(new Rect(0, 0, 400, 400), mPaint);

