

## DrawView.java

```
package com.macdougherty;

//Mainline code provided by
http://bestsiteinthemultiverse.com/2008/11/android-graphics-example/ with additions made
by Mac

import android.app.Activity;

public class DrawView extends Activity
{
    DemoView demoview;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        demoview = new DemoView(this);
        setContentView(demoview);
    }

    private class DemoView extends View
    {
        public DemoView(Context context)
        {
            super(context);
        }

        @Override protected void onDraw(Canvas canvas)
        {
            super.onDraw(canvas);

            // custom drawing code here
            // remember: y increases from top to bottom
            // x increases from left to right
            int x = 0;
            int y = 0;
            Paint paint = new Paint();
            paint.setStyle(Paint.Style.FILL);

            // make the entire canvas white
            paint.setColor(Color.WHITE);
            canvas.drawPaint(paint);
            // another way to do this is to use:
            // canvas.drawColor(Color.WHITE);

            // draw a solid blue circle
            paint.setColor(Color.BLUE);
            canvas.drawCircle(20, 20, 15, paint);

            // draw blue circle with antialiasing turned on
            paint.setAntiAlias(true); //Smooths the edges of the circle
            paint.setColor(Color.RED);
            canvas.drawCircle(60, 20, 15, paint);

            // compare the above circles once drawn
            // the first circle has a jagged perimeter
            // the second circle has a smooth perimeter
        }
    }
}
```

## DrawView.java

```
// draw a solid red rectangle
paint.setAntiAlias(false);
paint.setColor(Color.RED);
canvas.drawRect(100, 100, 200, 200, paint);

// create and draw triangles
// use a Path object to store the 3 line segments
// use .offset to draw in many locations
// note: this triangle is not centered at 0,0
paint.setStyle(Paint.Style.STROKE);
paint.setStrokeWidth(2);
paint.setColor(Color.RED);
Path path = new Path();
path.moveTo(0, -10);
path.lineTo(5, 0);
path.lineTo(-5, 0);
path.close();
path.offset(10, 40);
canvas.drawPath(path, paint);
path.offset(50, 100);
canvas.drawPath(path, paint);
// offset is cumulative
// next draw displaces 50,100 from previous
path.offset(50, 100);
canvas.drawPath(path, paint);

// draw some text using STROKE style
paint.setStyle(Paint.Style.STROKE);
paint.setStrokeWidth(1);
paint.setColor(Color.MAGENTA);
paint.setTextSize(30);
canvas.drawText("Style.STROKE", 75, 75, paint);

// draw some text using FILL style
paint.setStyle(Paint.Style.FILL);
//turn antialiasing on
paint.setAntiAlias(true);
paint.setTextSize(30);
canvas.drawText("Style.FILL", 75, 110, paint);

// draw some rotated text
// get text width and height
// set desired drawing location
x = 75;
y = 185;
paint.setColor(Color.GRAY);
paint.setTextSize(25);
String str2rotate = "Rotated!";

// draw bounding rect before rotating text
Rect rect = new Rect();
paint.getTextBounds(str2rotate, 0, str2rotate.length(), rect);
canvas.translate(x, y);
paint.setStyle(Paint.Style.FILL);
// draw unrotated text
canvas.drawText("!Rotated", 0, 0, paint);
paint.setStyle(Paint.Style.STROKE);
canvas.drawRect(rect, paint);
```

DrawView.java

```
// undo the translate
canvas.translate(-x, -y);

// rotate the canvas on center of the text to draw
canvas.rotate(-45, x + rect.exactCenterX(),
              y + rect.exactCenterY());

// draw the rotated text
paint.setStyle(Paint.Style.FILL);
canvas.drawText(str2rotate, x, y, paint);

//undo the rotate
canvas.restore();
canvas.drawText("After canvas.restore()", 50, 250, paint);

// draw a thick dashed line
DashPathEffect dashPath =
    new DashPathEffect(new float[]{20,5}, 1);
paint.setPathEffect(dashPath);
paint.setStrokeWidth(8);
canvas.drawLine(0, 300 , 320, 300, paint);
}
}
}
```