转载请注明链接: https://blog.csdn.net/feather_wch/article/details/79991218

Path文本动画实战

版本:2018/4/18-1

1、Path获取文本路径

> 1. 参数1：字符串
> 2. 参数2/3: 字符串中的一部分(这里是全部字符串0~length)
> 3. 参数4，参数5. 本文开始的坐标(0, 文本高度)
> 4. mSrcPath为存储文本路径的path

```
paint.getTextPath(text, 0, text.length(), 0, paint.getTextSize(),
               mSrcPath);
```

2、通过PathMeasure获取文本路径以及总长度

```
//1. 设置mPathMeasure
        mPathMeasure.setPath(mSrcPath, false);

//2. 通过PathMeasure获取文字路径的总长度（需要循环遍历）
        mLengthSum = mPathMeasure.getLength();
        while (mPathMeasure.nextContour()) {
            mLengthSum += mPathMeasure.getLength();
        }
```

3、PathMeasure的 `getSegment` 方法

> 1. `PathMeasure` 中将文字分为一个个 `segment`片段 ，调用第一次 `getSegment`仅仅是获得第一个文字
> 2. `mPathMeasure.nextContour()` 能进行遍历

```
//1-设置Src路径
mPathMeasure.setPath(mSrcPath, false);
//2-拼接出目标路径
while (stopDistance > mPathMeasure.getLength()) {
    stopDistance -= mPathMeasure.getLength();
    mPathMeasure.getSegment(0, mPathMeasure.getLength(), mDstPath, true);
    if (!mPathMeasure.nextContour()) {
        break;
    }
}
//3-最后一段
mPathMeasure.getSegment(0, stopDistance, mDstPath, true);
```

TextPathView

```java
public class TextPathView extends AppCompatTextView {
    /**
     * 最终绘制目标的路径
     */
    Path mDstPath = new Path();
    /**
     * Path辅助进行截断
     */
    PathMeasure mPathMeasure = new PathMeasure();
    /**
     * 绘制目标的画笔
     */
    Paint mPaint = new Paint();
    /**
     * 源文字路径
     */
    private Path mSrcPath = new Path();
    /**
     * 动画进度(0~1f)
     */
    private float mProgress = 0f;
    /**
     * 源文本路径的总长度
     */
    private float mLengthSum = 0;

    /**
     * 文字线条的宽度
     */
    private float mStrokeWidth = 1f;

    /**
     * 动画时长
     */
    private int mDuration = 2000;

    public int getDuration() {
        return mDuration;
    }

    public void setDuration(int duration) {
        mDuration = duration;
    }

    public float getStrokeWidth() {
        return mStrokeWidth;
    }

    public void setStrokeWidth(float strokeWidth) {
        mStrokeWidth = strokeWidth;
    }

    public TextPathView(Context context) {
        super(context);
        initPath(getText().toString());
    }

    private void initPath(String text) {

        Paint paint = new Paint();
        paint.setTextSize(getTextSize());
        //1. 获取到文字路径,保存到path中
        paint.getTextPath(text, 0, text.length(), 0, paint.getTextSize(),
                mSrcPath);

        //2. 设置mPathMeasure
        mPathMeasure.setPath(mSrcPath, false);

        //3. 通过PathMeasure获取文字路径的总长度
        mLengthSum = mPathMeasure.getLength();
        while (mPathMeasure.nextContour()) {
            mLengthSum += mPathMeasure.getLength();
        }
    }
    public TextPathView(Context context, @Nullable AttributeSet attrs) {
        super(context, attrs);
        initPath(getText().toString());
    }

    public TextPathView(Context context, @Nullable AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        initPath(getText().toString());
    }

    public float getProgress() {
        return mProgress;
    }

    public void setProgress(float progress) {
        mProgress = progress;
        postInvalidate();
    }

    /**
     * 设置字符串(内部会进行动画的准备工作)
     * @param text
     */
    public void setText(String text) {
        setText(text, BufferType.NORMAL);
        initPath(text);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        //1. 结束的长度
        float stopDistance = mLengthSum * mProgress;
        //2. 配置好pathmeasure
        mPathMeasure.setPath(mSrcPath, false);
        //3.
        while (stopDistance > mPathMeasure.getLength()) {
            stopDistance -= mPathMeasure.getLength();
            mPathMeasure.getSegment(0, mPathMeasure.getLength(), mDstPath, true);
            if (!mPathMeasure.nextContour()) {
                break;
            }
        }
        mPathMeasure.getSegment(0, stopDistance, mDstPath, true);
        //4. 绘制
        mPaint.setStyle(Paint.Style.STROKE);
        mPaint.setStrokeWidth(mStrokeWidth);
        mPaint.setColor(getCurrentTextColor());
        mPaint.setAntiAlias(true);
        canvas.drawPath(mDstPath, mPaint);
    }

    public void startAnim() {
        ObjectAnimator objectAnimator
                = ObjectAnimator.ofFloat(TextPathView.this, "Progress",
                0, 1f);
        objectAnimator.setDuration(mDuration);
        objectAnimator.start();
    }
}
```