

- [实例：波浪进度球](#)
- [2、爱心](#)
- [参考资料](#)

## 实例：波浪进度球

1、进度球控件-BezierProgressBar.java

```

public class BezierProgressBar extends View{

    private Paint mPaint;

    private int mScreenWidth;
    private int mScreenHeight;
    private float mCenterX;
    private float mCenterY;
    private float mRadius;

    private int mWaveLength = 250; //px
    private int mWaveCount = 0;

    private float mProgress = 0f; //0~1f

    private Path mPath;

    private int mStrokeWidth = 10;
    private int mColor = Color.parseColor("#29b6f6");
    private int mBackgroundColor = Color.parseColor("#ffffffff");

    public float getProgress() {
        return mProgress;
    }

    Path mTextOutlineSrcPath;
    Path mTextOutlineDstPath;

    PathMeasure mPathMeasure;

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

    public BezierProgressBar(Context context) {
        super(context);
    }

    public BezierProgressBar(Context context, @Nullable AttributeSet attrs) {
        super(context, attrs);
    }

    public BezierProgressBar(Context context, @Nullable AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
    }

    public BezierProgressBar(Context context, @Nullable AttributeSet attrs, int defStyleAttr, int defStyleRes) {
        super(context, attrs, defStyleAttr, defStyleRes);
    }

    @Override
    protected void onSizeChanged(int w, int h, int oldw, int oldh) {

```

```

super.onSizeChanged(w, h, oldw, oldh);
//1-获取屏幕宽高
mScreenWidth = w;
mScreenHeight = h;

mWaveLength = (int) (mScreenWidth / 1.5f);
mWaveCount = (mScreenWidth + mWaveLength) / mWaveLength + 1;

//2-初始化path
mPath = new Path();
mPaint = new Paint();
mTextOutlineSrcPath = new Path();
mTextOutlineDstPath = new Path();
mPathMeasure = new PathMeasure(); //获取到文字外框的path

//3-
mCenterX = mScreenWidth / 2;
mCenterY = mScreenHeight / 2;
mRadius = Math.min(mScreenWidth, mScreenHeight) / 2 - mStrokeWidth;
}

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    //波浪path和画笔
    mPath.reset();
    mPaint.reset();
    //外框路线
    mTextOutlineSrcPath.reset();
    mTextOutlineDstPath.reset();

    /**
     * 1-获取波浪的Path
     */
    int accelerateRate = (mDuration + 1000) / 1000;

    // 波浪偏移
    float waveOffsetY = mWaveLength / 9;
    // x轴偏移
    float offsetX = (accelerateRate * mProgress * mWaveLength);
    // y轴偏移(总长度 + 海浪的高度 提升动画体验)
    //float offsetY = mProgress * (mScreenHeight + waveOffsetY); //本质应该加上该值
    float offsetY = mProgress * (mScreenHeight); //无法完全覆盖圆形，但是可以保证填满

    float startX = 0 - mWaveLength + offsetX % mWaveLength;
    float startY = mScreenHeight - offsetY;
    mPath.moveTo(startX, startY);
    for (int i = 0; i < mWaveCount; i++) {
        mPath.quadTo(startX + (i * mWaveLength) + 1/4f * mWaveLength, startY - waveOffsetY,
            startX + (i * mWaveLength) + 1/2f * mWaveLength, startY);
        mPath.quadTo(startX + (i * mWaveLength) + 3/4f * mWaveLength, startY + waveOffsetY,
            startX + (i * mWaveLength) + 1f * mWaveLength, startY);
    }
    mPath.lineTo(mScreenWidth, mScreenHeight);
    mPath.lineTo(0, mScreenHeight);
}

```

```

mPath.close();
mPaint.setAntiAlias(true);

/**
 * 2-通过将波浪限制在圆形之内
 */
//离屏缓冲(View的setLayerType会直接把整个View都绘制在离屏缓冲中)
int saved = canvas.saveLayer(null, null, Canvas.ALL_SAVE_FLAG);

//1. 绘制圆形(将波浪所在圆形的内部)-第一个为目标图形
mPaint.setStyle(Paint.Style.FILL);
mPaint.setColor(mBackgroundColor);
canvas.drawCircle(mCenterX, mCenterY, mRadius, mPaint);

Xfermode xfermode = new PorterDuffXfermode(PorterDuff.Mode.SRC_ATOP);
mPaint.setXfermode(xfermode);
//2. 绘制波浪的路线-第二个为src图形
mPaint.setColor(mColor);
mPaint.setStyle(Paint.Style.FILL);
canvas.drawPath(mPath, mPaint);

mPaint.setXfermode(null);

//恢复
canvas.restoreToCount(saved);

/**
 * 3-绘制真正的圆形
 */
mPaint.setColor(mColor);
mPaint.setStrokeWidth(10);
if(mProgress >= 1f){ //>=100%的时候进行填满
    mPaint.setStyle(Paint.Style.FILL);
    canvas.drawCircle(mCenterX, mCenterY, mRadius, mPaint);
}else{
    mPaint.setStyle(Paint.Style.STROKE);
    canvas.drawCircle(mCenterX, mCenterY, mRadius, mPaint);
}

/**
 * 4-绘制进度
 */
String text = (int)(mProgress * 100) + "%";
mPaint.setStyle(Paint.Style.FILL);
mPaint.setTextSize(mTextSize);
mPaint.setColor(mBackgroundColor);
//可见
mPaint.setAlpha(255);
mPaint.setFakeBoldText(true);
float textWidth = mPaint.measureText(text);
canvas.drawText(text, mCenterX - textWidth/2, mCenterY + mTextSize / 2, mPaint);
/**
 * 绘制外框
 */
//2-获取到文本Path

```

```

mPaint.getTextPath(text, 0, text.length(),
    mCenterX - textWidth/2, mCenterY + mTextSize / 2,
    mTextOutlineSrcPath);

mPathMeasure.setPath(mTextOutlineSrcPath, false);
do{
    mPathMeasure.getSegment(0, mPathMeasure.getLength(), mTextOutlineDstPath, true);
}while(mPathMeasure.nextContour());

mPaint.setStyle(Paint.Style.STROKE);
mPaint.setStrokeWidth(mStrokeWidth / 5);
mPaint.setColor(mColor);
mPaint.setAntiAlias(true);
canvas.drawPath(mTextOutlineDstPath, mPaint);
}

private int mTextSize = 100; //px

public void setTextSizeDp(int textSizeDp) {
    final float scale = getContext().getResources().getDisplayMetrics().density;
    mTextSize = (int) (textSizeDp * scale + 0.5f);
}

private int mDuration = 10000;

public void startAnimation(){
    ObjectAnimator objectAnimator = ObjectAnimator.ofFloat(this, "progress", 0f, 1f);
    objectAnimator.setDuration(mDuration);
    objectAnimator.setInterpolator(new LinearInterpolator());
    objectAnimator.start();
}

@Override
public boolean onTouchEvent(MotionEvent event) {
    if(event.getAction() == MotionEvent.ACTION_DOWN){
        startAnimation();
    }
    return super.onTouchEvent(event);
}

public int getWaveLength() {
    return mWaveLength;
}

public void setWaveLength(int waveLength) {
    mWaveLength = waveLength;
    mWaveCount = (mScreenWidth + mWaveLength) / mWaveLength + 1;
}

public int getStrokeWidth() {
    return mStrokeWidth;
}

public void setStrokeWidth(int strokeWidth) {

```

```

        mStrokeWidth = strokeWidth;
        mRadius = Math.min(mScreenWidth, mScreenHeight) / 2 - mStrokeWidth;
    }

    public int getColor() {
        return mColor;
    }

    public void setColor(int color) {
        mColor = color;
    }

    public int getBackgroundColor() {
        return mBackgroundColor;
    }

    public void setBackgroundColor(int backgroundColor) {
        mBackgroundColor = backgroundColor;
    }

    public int getDuration() {
        return mDuration;
    }

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

```

## 2、布局进行设置

```

<com.feather.imageview.Widget.BezierProgressBar
    android:id="@+id/md_bezier_progressbar"
    android:layout_width="200dp"
    android:layout_height="200dp"/>

```

## 3、使用

```

BezierProgressBar progressBar = findViewById(R.id.md_bezier_progressbar);
progressBar.startAnimation();

```

# 2、爱心

## 1、爱心控件-BezierHeart.java

```

public class BezierHeart extends View{

    private float mProgress = 0f; //0~1f

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

    public BezierHeart(Context context) {
        super(context);
    }

    public BezierHeart(Context context, @Nullable AttributeSet attrs) {
        super(context, attrs);
    }

    public BezierHeart(Context context, @Nullable AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
    }

    public BezierHeart(Context context, @Nullable AttributeSet attrs, int defStyleAttr, int defStyleRes) {
        super(context, attrs, defStyleAttr, defStyleRes);
    }

    private int mScreenWidth;
    private int mScreenHeight;
    private float mCenterX;
    private float mCenterY;
    private float mRadius;

    private int mStrokeWidth = 10;

    @Override
    protected void onSizeChanged(int w, int h, int oldw, int oldh) {
        super.onSizeChanged(w, h, oldw, oldh);
        //1-获取屏幕宽高
        mScreenWidth = w;
        mScreenHeight = h;

        //2-圆心
        mCenterX = mScreenWidth / 2;
        mCenterY = mScreenHeight / 2;
        mRadius = Math.min(mScreenWidth, mScreenHeight) / 2 - mStrokeWidth;

        float offset = mRadius * FLOAT_C;

        //3-初始化所有点
        mPointF0 = new PointF(mCenterX + 0, mCenterY + mRadius);
        mPointF1 = new PointF(mCenterX + offset, mCenterY + mRadius);
        mPointF2 = new PointF(mCenterX + mRadius, mCenterY + offset);
    }
}

```

```

        mPointF3 = new PointF(mCenterX + mRadius, mCenterY + 0);
        mPointF4 = new PointF(mCenterX + mRadius, mCenterY - offset);
        mPointF5 = new PointF(mCenterX + offset, mCenterY - mRadius);
        mPointF6 = new PointF(mCenterX, mCenterY - mRadius);
        mPointF7 = new PointF(mCenterX - offset, mCenterY - mRadius);
        mPointF8 = new PointF(mCenterX - mRadius, mCenterY - offset);
        mPointF9 = new PointF(mCenterX - mRadius, mCenterY);
        mPointF10 = new PointF(mCenterX - mRadius, mCenterY + offset);
        mPointF11 = new PointF(mCenterX - offset, mCenterY + mRadius);
        //4-
        mPath = new Path();
        //5-
        mPaint = new Paint();
    }

    public int getStrokeWidth() {
        return mStrokeWidth;
    }

    public void setStrokeWidth(int strokeWidth) {
        mStrokeWidth = strokeWidth;
        mRadius = Math.min(mScreenWidth, mScreenHeight) / 2 - mStrokeWidth;
    }

    private static final float FLOAT_C = 0.551915024494f;

    PointF mPointF0;
    PointF mPointF1;
    PointF mPointF2;
    PointF mPointF3;
    PointF mPointF4;
    PointF mPointF5;
    PointF mPointF6;
    PointF mPointF7;
    PointF mPointF8;
    PointF mPointF9;
    PointF mPointF10;
    PointF mPointF11;

    Path mPath;
    Paint mPaint;

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

        float p60offset = (mRadius / 5 * 3) * mProgress;
        float p110offset = (mRadius / 5 * 2) * mProgress;
        float pAll0offset = (mRadius / 5) * mProgress;

        float offset = mRadius * FLOAT_C;

        mPointF0.y = mCenterY + (mRadius - pAll0offset);

        mPointF1.x = mCenterX + offset - pAll0offset;

```



```

mPointF1.y = mCenterY + (mRadius - p1110ffset);

mPointF2.x = mCenterX + mRadius - pAlloffset ;
mPointF2.y = mCenterY + offset - pAlloffset;

mPointF3.x = mCenterX + mRadius - pAlloffset;

mPointF4.x = mCenterX + mRadius - pAlloffset;
mPointF4.y = mCenterY - offset + pAlloffset;

mPointF5.x = mCenterX + offset - pAlloffset;
mPointF5.y = mCenterY - mRadius + pAlloffset;

mPointF6.y = mCenterY - (mRadius - p60ffset);

mPointF7.x = mCenterX - offset + pAlloffset;
mPointF7.y = mCenterY - mRadius + pAlloffset;

mPointF8.x = mCenterX - mRadius + pAlloffset;
mPointF8.y = mCenterY - offset + pAlloffset;

mPointF9.x = mCenterX - mRadius + pAlloffset;

mPointF10.x = mCenterX - mRadius + pAlloffset;
mPointF10.y = mCenterY + offset - pAlloffset;

mPointF11.x = mCenterX - offset + pAlloffset;
mPointF11.y = mCenterY + (mRadius - p1110ffset);


mPath.reset();
mPath.moveTo(mPointF0.x, mPointF0.y);
mPath.cubicTo(mPointF1.x, mPointF1.y, mPointF2.x, mPointF2.y, mPointF3.x, mPointF3.y);
mPath.cubicTo(mPointF4.x, mPointF4.y, mPointF5.x, mPointF5.y, mPointF6.x, mPointF6.y);
mPath.cubicTo(mPointF7.x, mPointF7.y, mPointF8.x, mPointF8.y, mPointF9.x, mPointF9.y);
mPath.cubicTo(mPointF10.x, mPointF10.y, mPointF11.x, mPointF11.y, mPointF0.x, mPointF0.y);
mPath.close();

mPaint.setStyle(Paint.Style.FILL_AND_STROKE);
mPaint.setColor(Color.RED);
mPaint.setAntiAlias(true);
mPaint.setStrokeCap(Paint.Cap.ROUND);
mPaint.setStrokeJoin(Paint.Join.ROUND);
mPaint.setStrokeWidth(mStrokeWidth);

canvas.drawPath(mPath, mPaint);
}

private int mDuration = 500;

public void startAnimation(){
    ObjectAnimator objectAnimator = ObjectAnimator.ofFloat(this, "progress", 0f, 1f);
    objectAnimator.setDuration(mDuration);
    objectAnimator.setInterpolator(new OvershootInterpolator());
    objectAnimator.start();
}

```

```

}

@Override
public boolean onTouchEvent(MotionEvent event) {
    if(event.getAction() == MotionEvent.ACTION_DOWN){
        startAnimation();
    }
    return super.onTouchEvent(event);
}
}

```

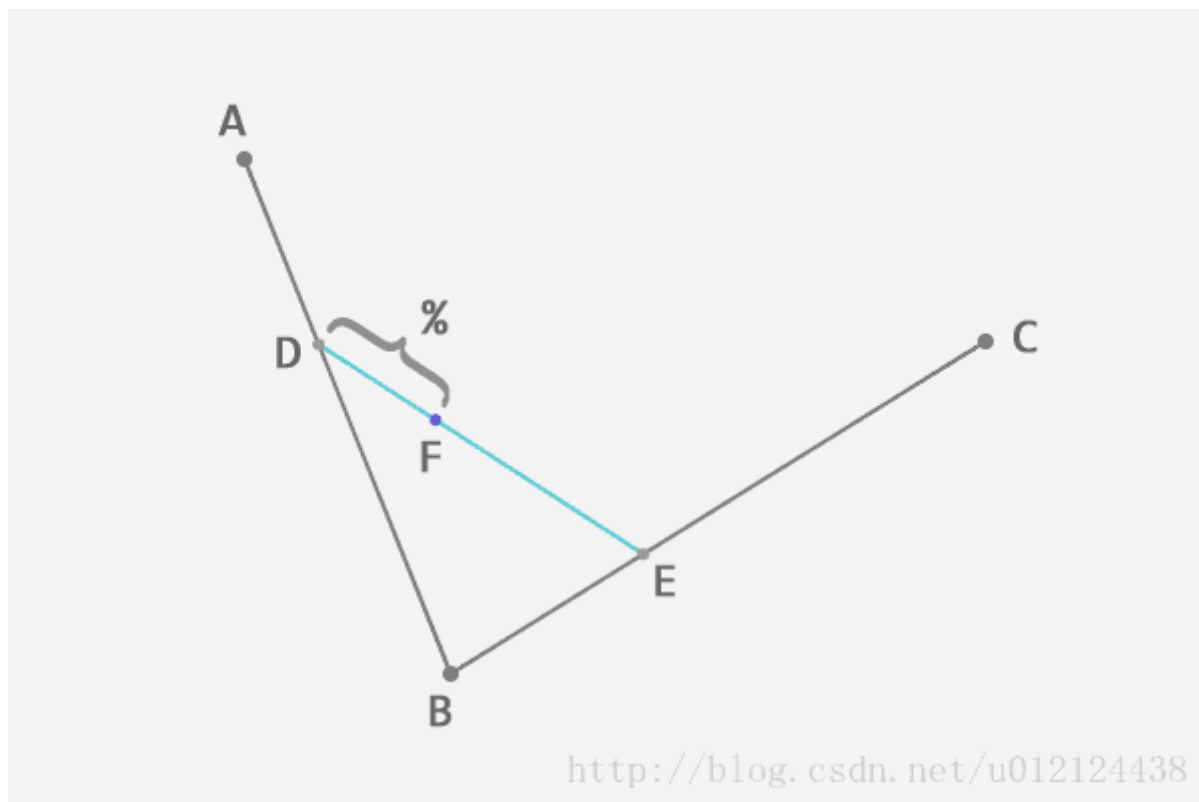
## 2、开启动画

```

BezierHeart bezierHeart = findViewById(R.id.md_bezier_heart);
bezierHeart.startAnimation();

```

## 参考资料



$$AD:DB = BE:EC = DF:FE$$