

关于贝塞尔曲线的简单教程

在写android的应用的过程中，如果涉及到animation的话，有极高的可能性需要用到贝塞尔曲线。

什么是贝塞尔曲线？

贝塞尔曲线奠定了计算机绘图的基础，任何一个曲线都可以用贝塞尔曲线绘制方法进行绘制。

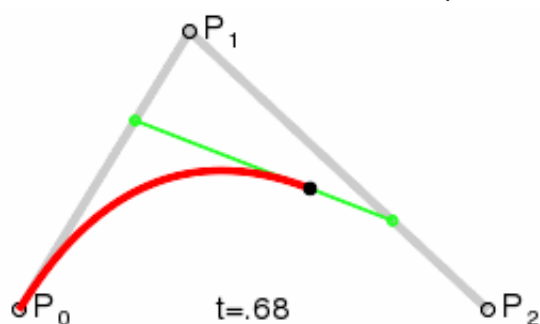
连接AB BC，并在AB上取点D，BC上取点E，使其满足条件：

$$\frac{AD}{AB} = \frac{BE}{BC}$$

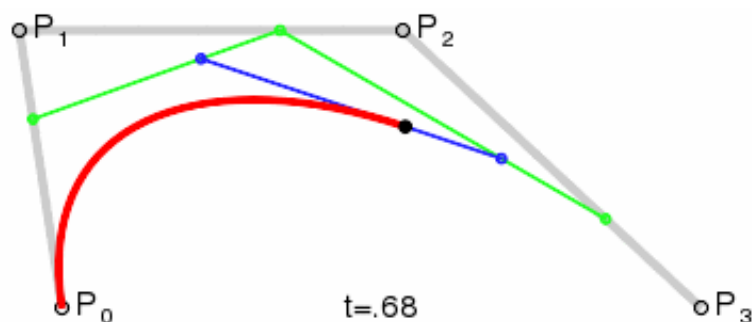
连接DE，取点F，使得：

$$\frac{AD}{AB} = \frac{BE}{BC} = \frac{DF}{DE}$$

所以二阶的贝塞尔曲线如下图：（方法名：quadTo）



同理，三阶的贝塞尔曲线如下图：(方法名：cubicTo)



在绘制贝塞尔曲线的时候需要用到canvas，paint，path等等的知识，如果有看不懂的地方去补一下相关知识。

直接上代码：

```
class BezierView : View {

    private var path: Path = Path()

    private lateinit var paint: Paint

    private var h: Int = 0
    private var w: Int = 0

    private var controlPoint: PointF = PointF()

    constructor(context: Context) : this(context, null)
    constructor(context: Context, attributeSet: AttributeSet?) : super(context, attributeSet)

    override fun onSizeChanged(w: Int, h: Int, oldw: Int, oldh: Int) {
        super.onSizeChanged(w, h, oldw, oldh)

        // 设置当前view的高和宽
        this.h = h
        this.w = w
    }

    @SuppressWarnings("ClickableViewAccessibility")
    override fun onTouchEvent(event: MotionEvent?): Boolean {
        // 初始化控制点, 以及有TouchEvent的时候更新控制点坐标
        controlPoint = PointF(event?.x ?: 0F, event?.y ?: 0F)

        // 更新View
        invalidate()
        return true
    }

    @SuppressWarnings("DrawAllocation")
    override fun onDraw(canvas: Canvas?) {
        super.onDraw(canvas)

        paint = Paint()

        // 重置path, 为的是防止重复绘制贝塞尔曲线, 使画布上残留多条曲线
        path.reset()
    }
}
```

```

// 配置画笔paint
paint.color = context.getColor(R.color.colorAccent)
paint.strokeWidth = 2F
paint.style = Paint.Style.STROKE

// 设置左右两个基准点
val pointLeft = PointF(0F, h / 2.toFloat())
val pointRight = PointF(w.toFloat(), h / 2.toFloat())

// 绘制左右基准点
canvas?.drawPoint(pointLeft.x, pointLeft.y, paint)
canvas?.drawPoint(pointRight.x, pointRight.y, paint)

// 绘制关于贝塞尔曲线的辅助线
canvas?.drawLine(pointLeft.x, pointLeft.y, controlPoint.x, controlPoint.y, paint)
canvas?.drawLine(pointRight.x, pointRight.y, controlPoint.x, controlPoint.y, paint)

paint.color = context.getColor(R.color.colorPrimaryDark)

// 为了绘制贝塞尔曲线，需要移动到其中一个基准点
path.moveTo(pointLeft.x, pointLeft.y)

// 根据基准点和控制点，绘制贝塞尔曲线
path.quadTo(controlPoint.x, controlPoint.y, pointRight.x, pointRight.y)

// 在画布上画path
canvas?.drawPath(path, paint)
}

}

```

效果如下图：



