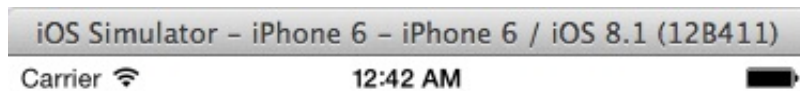


一、饼状图

/**目标

- *掌握饼状图的绘制原理
 - *熟悉quartz2d 的画弧方法
- */



步骤:

1. 自定义一个饼状View(PieView), 添加到控制器View上
2. 添加PieView的一个类型为数据的sections属性, 存储所有分类的个数, 并添加一个颜色数组, 用于存储颜色
3. 在drawRect方法中遍历sections的大小
4. 遍历sections中的个数, 进行总数绘总
5. 定义一个 "扇形的起始位置"
6. 设置路径中心点
7. 遍历sections, 计算数组中每一个元素占用总数的比例
8. 根据比例计算饼状的结束位置, 并设置 "弧" 路径
9. 渲染扇形在UIView上, 实现实心的扇形
8. 给 "扇形的起始位置" 重新赋值, 进入下一个循环

二、图形上下文栈

"什么是图形上下文栈?"

- (1) 将当前的图形上下文状态copy一份到栈, 这就是 '图形上下文栈'
- (2) 上下文的什么状态呢? 比如 颜色、线宽, 这些都是上下文的状态

"图形上下文有什么用?"

- (1) 恢复最初的绘图状态

"图形上下文API"

1. 保存图形上下文状态使用CGContextSaveGState方法
2. 恢复图形上下文状态使用CGContextRestoreGState方法
3. CGContextRestoreGState不能调用多次, 要看图形上下文栈有多少个上下文状态可恢复

三、矩阵操作

/**目标

- *掌握在图层上下文中的平稳、缩放、旋转
- */

// 平移

```
CGContextTranslateCTM(ctx, 0, -80);
```

// 缩放 - xy方向缩放到原来的几倍

```
CGContextScaleCTM(ctx, 1.0, 1.0);
```

// 旋转 沿左上角旋转

```
CGContextRotateCTM(ctx, -M_PI * 0.1);
```

四、裁剪圆形图片

/**目标

- *掌握CGContextClip方法的作用, 这个方法是裁剪 "路径" 之外多余的部分

```
*/
```

"裁剪圆形图片"

步骤

1. 自定义一个CircleImageView控件，在drawRect中，获取上下文，往上下文中添加一个圆的路径
2. 调用CGContextClip的方法，裁剪路径之外多余的部分
3. 自定义的控制中，添加一个imageName属性，然后回到drawrect方法画图
4. 画圆的边框，使用CGContextAddEllipseInRect添加圆的路径，并使用CGContextStrokePath画空心圆

五、动画(气球)

```
/**
```

- *1. 掌握定时器NSTimer和CADisplayLink的使用
 - *2. CADisplayLink多用于UI的刷新，1/60秒调用一次
 - *3. CADisplayLink要添加到主运行循环才能使用
- ```
*/
```

## "一个气球下降"

步骤

1. 自定义一个BalloonView，添加一个position属性，用于记录气球的位置
2. 在drawRect方法中，创建一张图片，并画出来
3. 在awakeFromNib中，使用定时器NSTimer实现气球下降功能效果  
[NSTimer scheduledTimerWithTimeInterval:1 target:self  
selector:@selector(setNeedsDisplay) userInfo:nil repeats:YES]
4. 在drawRect如果气球的位置到达底部，从顶部重新开始下降
5. 在awakeFromNib中，使用定时器CADisplayLink实现气球下降功能效果  
(1) 创建CADisplayLink对象，调用addToRunLoop, 定时器就会执行  
(2) CADisplayLine有个duration属性，代理定时器调用的间隔时间，不能改

## "多个气球上升"