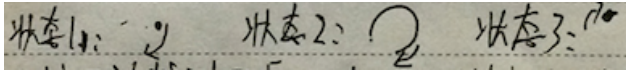


绘制进度由 0-1 交替的旋转动画

1. 简述:



2. 实现:

```
#import <UIKit/UIKit.h>

@interface MYImageScanProgressView : UIView

- (instancetype)initWithFrame:(CGRect)frame andLineWidth:(CGFloat)lineWidth andLineColor:(UIColor *)lineColor;

@end
```

```
@interface MYImageScanProgressView()

@property (nonatomic, assign) CGFloat lineWidth; /**<边框宽度*/
@property (nonatomic, strong) UIColor *lineColor; /**<边框颜色*/

@property (nonatomic, strong) CAShapeLayer *progressLayer; /**<进度条*/

@end

@implementation MYImageScanProgressView

- (instancetype)initWithFrame:(CGRect)frame {
    return [self initWithFrame:frame
                    andLineWidth:3.0
                    andLineColor:[UIColor whiteColor]];
}

- (instancetype)initWithFrame:(CGRect)frame andLineWidth:(CGFloat)lineWidth andLineColor:(UIColor *)lineColor {
    self = [super initWithFrame:frame];
    if(self) {
        self.lineWidth = lineWidth;
        self.lineColor = lineColor;
        [self.layer setAllowsEdgeAntialiasing:YES];
        [self setup];
    }

    return self;
}
```

```

-(void)setup {
    NSAssert(self.lineWidth > 0.0, @"lineWidth must great than zero");
    //NSAssert(self.lineColor.count > 0, @"lineColor must set");

    //外层旋转动画
    CABasicAnimation *rotationAnimation = [CABasicAnimation animationWithKeyPath:@"transform.rotation.z"];
    rotationAnimation.fromValue = @0.0;
    rotationAnimation.toValue = @(2*M_PI);
    rotationAnimation.repeatCount = HUGE_VALF;
    rotationAnimation.duration = 3.0;
    rotationAnimation.beginTime = 0.0;
    //rotationAnimation.fillMode = kCAFillModeForwards;
    rotationAnimation.removedOnCompletion = NO;
    [self.layer addAnimation:rotationAnimation forKey:@"rotationAnimation"];

    //内层进度条动画
    CABasicAnimation *strokeAnim1 = [CABasicAnimation animationWithKeyPath:@"strokeEnd"];
    strokeAnim1.fromValue = @0.0;
    strokeAnim1.toValue = @1.0;
    strokeAnim1.duration = 1.0;
    strokeAnim1.beginTime = 0.0;
    strokeAnim1.timingFunction = [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseIn];

    //内层进度条动画
    CABasicAnimation *strokeAnim2 = [CABasicAnimation animationWithKeyPath:@"strokeStart"];
    strokeAnim2.fromValue = @0.0;
    strokeAnim2.toValue = @1.0;
    strokeAnim2.duration = 1.0;
    strokeAnim2.beginTime = 1.0;
    strokeAnim2.timingFunction = [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseOut];

    CAAAnimationGroup *animGroup = [CAAnimationGroup animation];
    animGroup.duration = 2.0;
    animGroup.repeatCount = HUGE_VALF;
    //animGroup.fillMode = kCAFillModeForwards;
    animGroup.animations = @[strokeAnim1, strokeAnim2];
    animGroup.removedOnCompletion = NO;
    UIBezierPath *path = [UIBezierPath bezierPathWithOvalInRect:CGRectMake(self.lineWidth, self.lineWidth, CGRectGetWidth(self.frame)-self.lineWidth*2,
        CGRectGetHeight(self.frame)-self.lineWidth*2)];

    self.progressLayer = [CAShapeLayer layer];
    self.progressLayer.lineWidth = self.lineWidth;
    self.progressLayer.lineCap = kCALineCapRound;
    self.progressLayer.strokeColor = self.lineColor.CGColor;
    self.progressLayer.fillColor = [UIColor clearColor].CGColor;
    self.progressLayer.strokeStart = 0.0;
    self.progressLayer.strokeEnd = 1.0;
    self.progressLayer.allowsEdgeAntialiasing = YES;
    self.progressLayer.path = path.CGPath;
    [self.progressLayer addAnimation:animGroup forKey:@"strokeAnim"];

    [self.layer addSublayer:self.progressLayer];
}

-(void)dealloc
{
    [self.layer removeAllAnimations];
    [self removeFromSuperview];
}

```