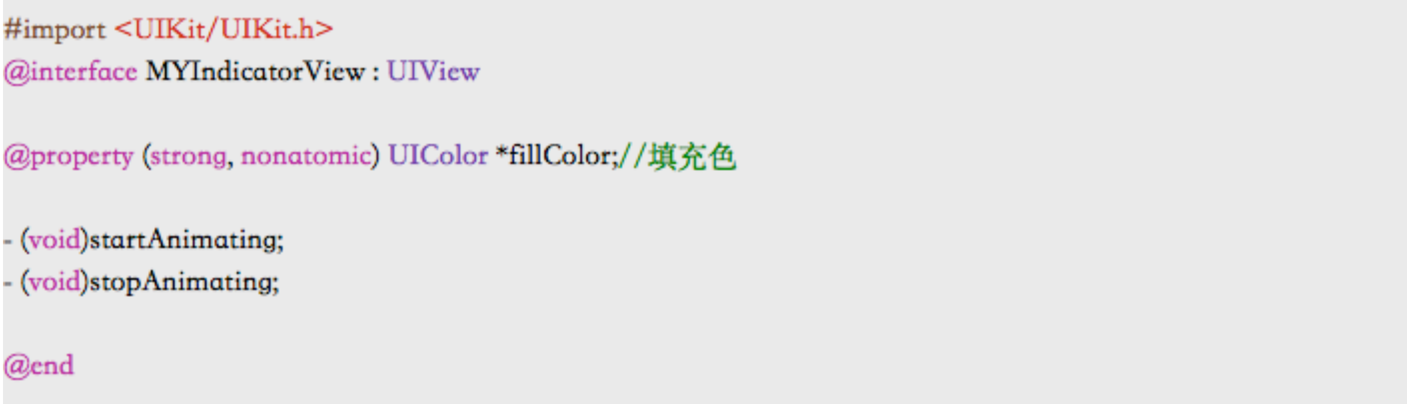
绘制正弦波形

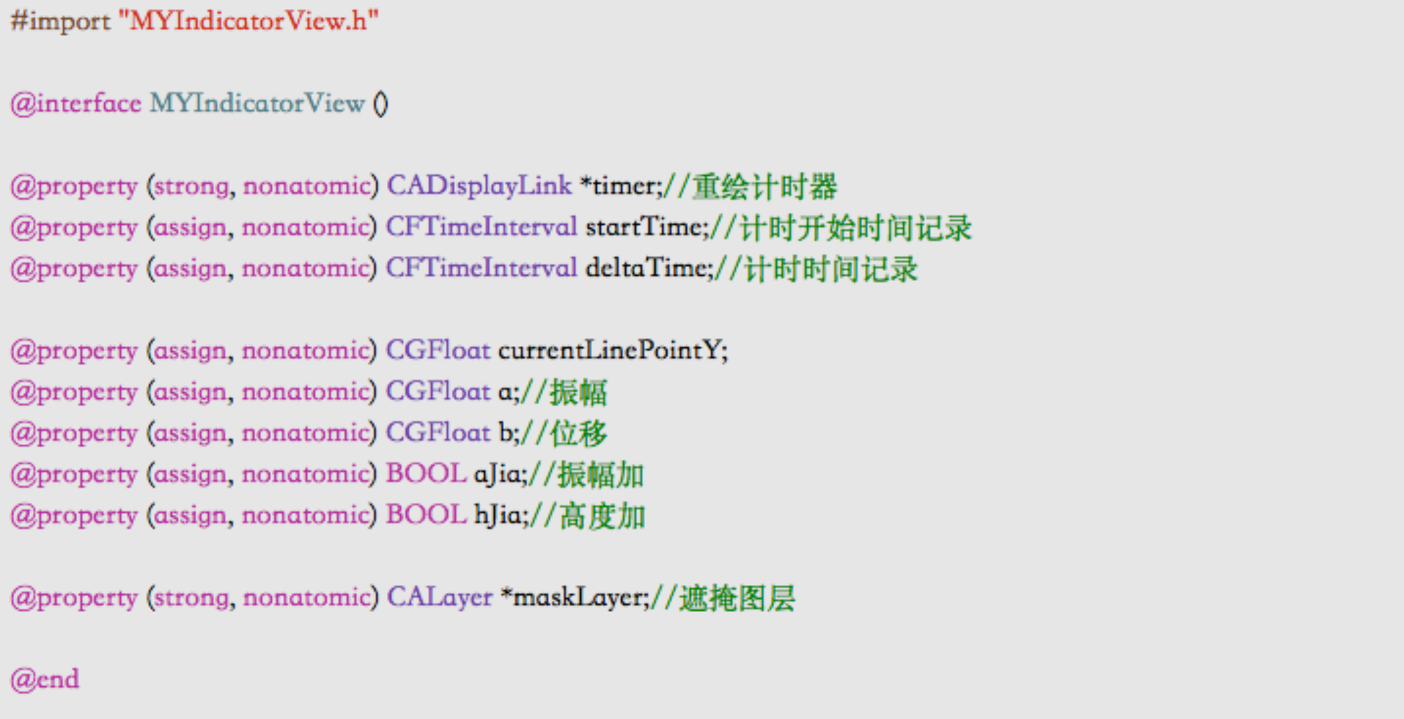
1. 简述：

UI 设计页面加载动画以产品 logo 做遮罩，实现波浪式加载动画效果。

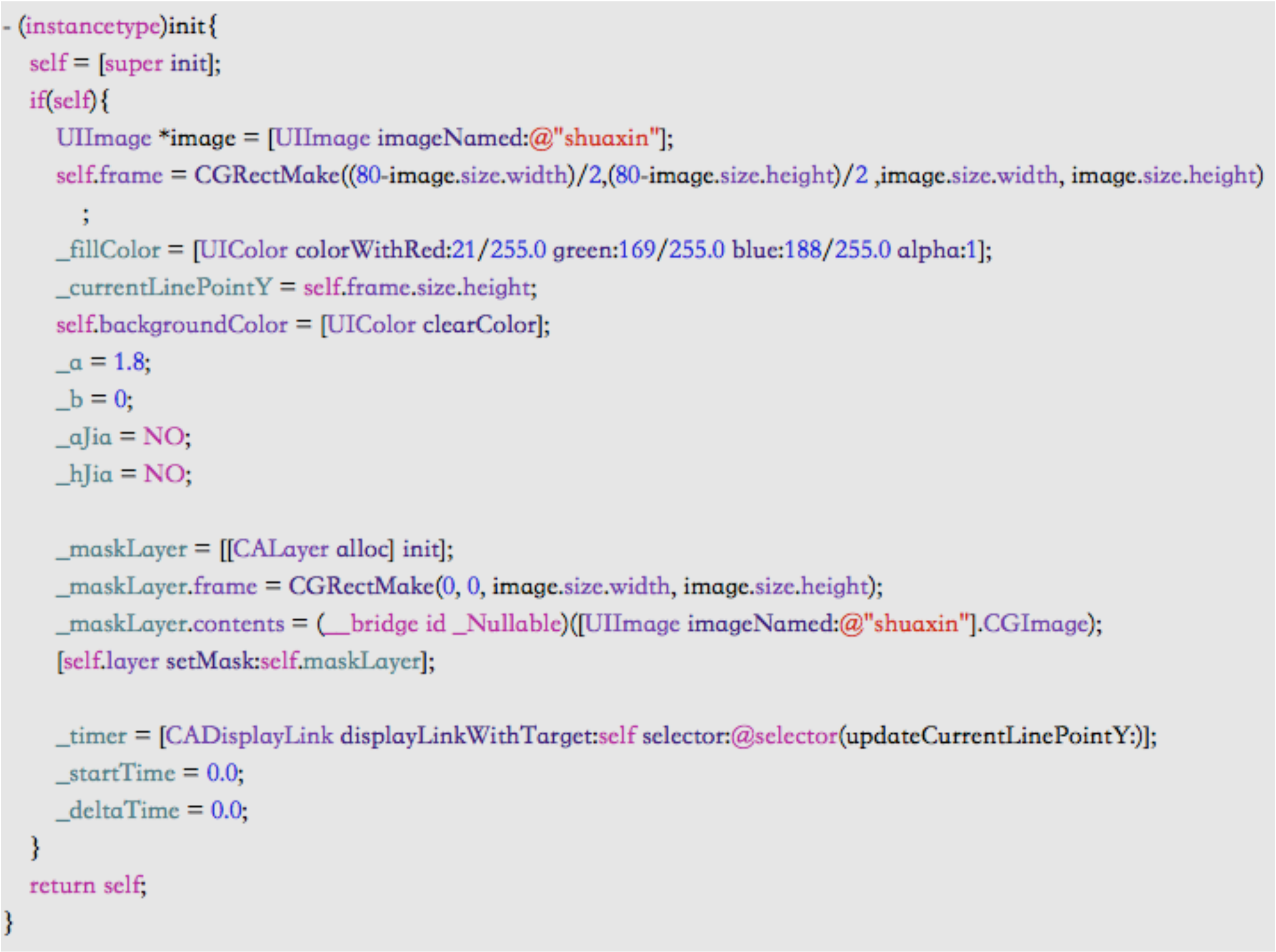
2. 实现：



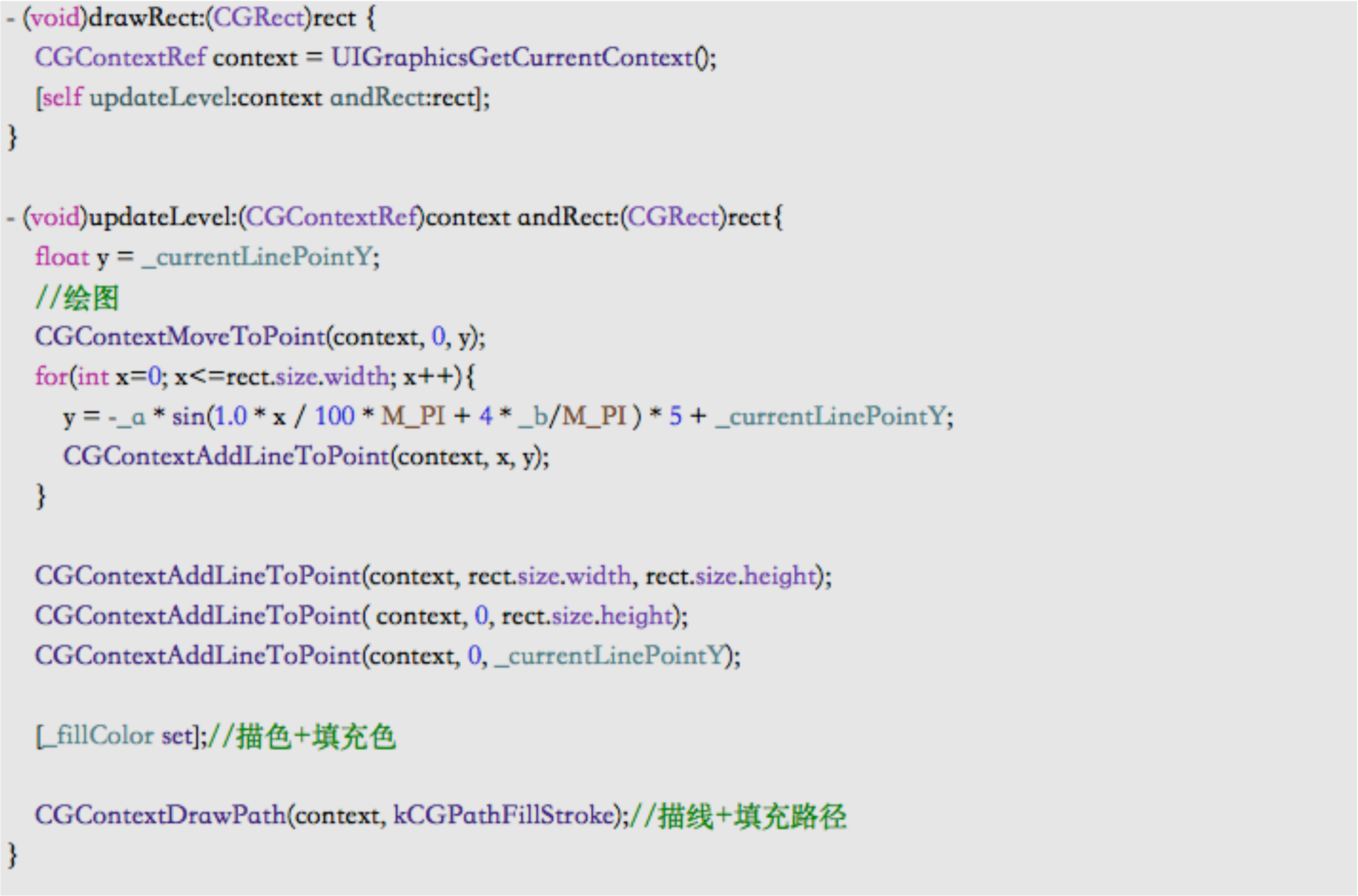
```
#import <UIKit/UIKit.h>
@interface MYIndicatorView : UIView

@property (strong, nonatomic) UIColor *fillColor;//填充色

- (void)startAnimating;
- (void)stopAnimating;

@end
```



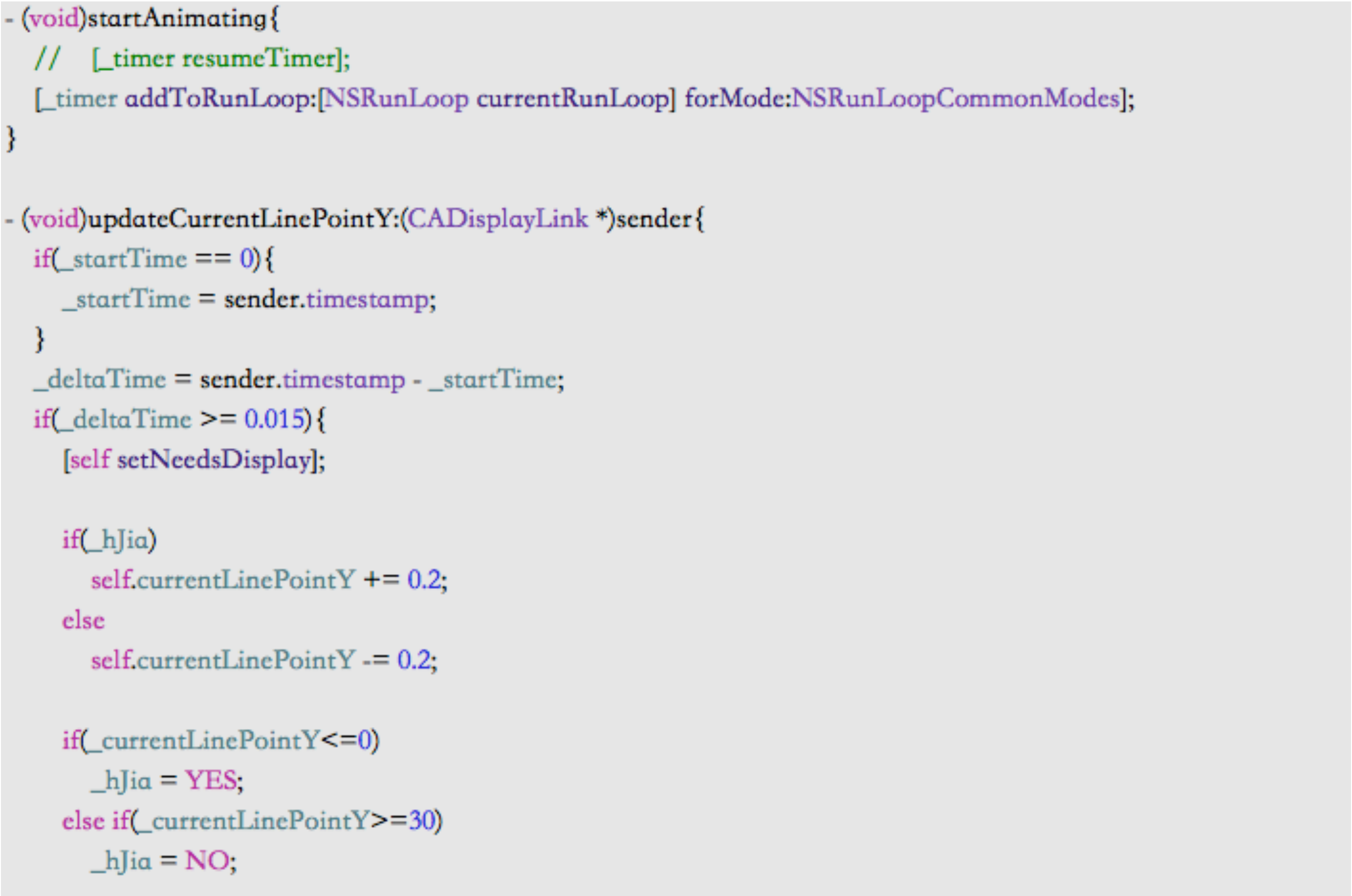
```
#import "MYIndicatorView.h"

@interface MYIndicatorView ()

@property (strong, nonatomic) CADisplayLink *timer;//重绘计时器
@property (assign, nonatomic) CFTimeInterval startTime;//计时开始时间记录
@property (assign, nonatomic) CFTimeInterval deltaTime;//计时时间记录

@property (assign, nonatomic) CGFloat currentLinePointY;
@property (assign, nonatomic) CGFloat a;//振幅
@property (assign, nonatomic) CGFloat b;//位移
@property (assign, nonatomic) BOOL aJia;//振幅加
@property (assign, nonatomic) BOOL hJia;//高度加

@property (strong, nonatomic) CALayer *maskLayer;//遮掩图层

@end
```



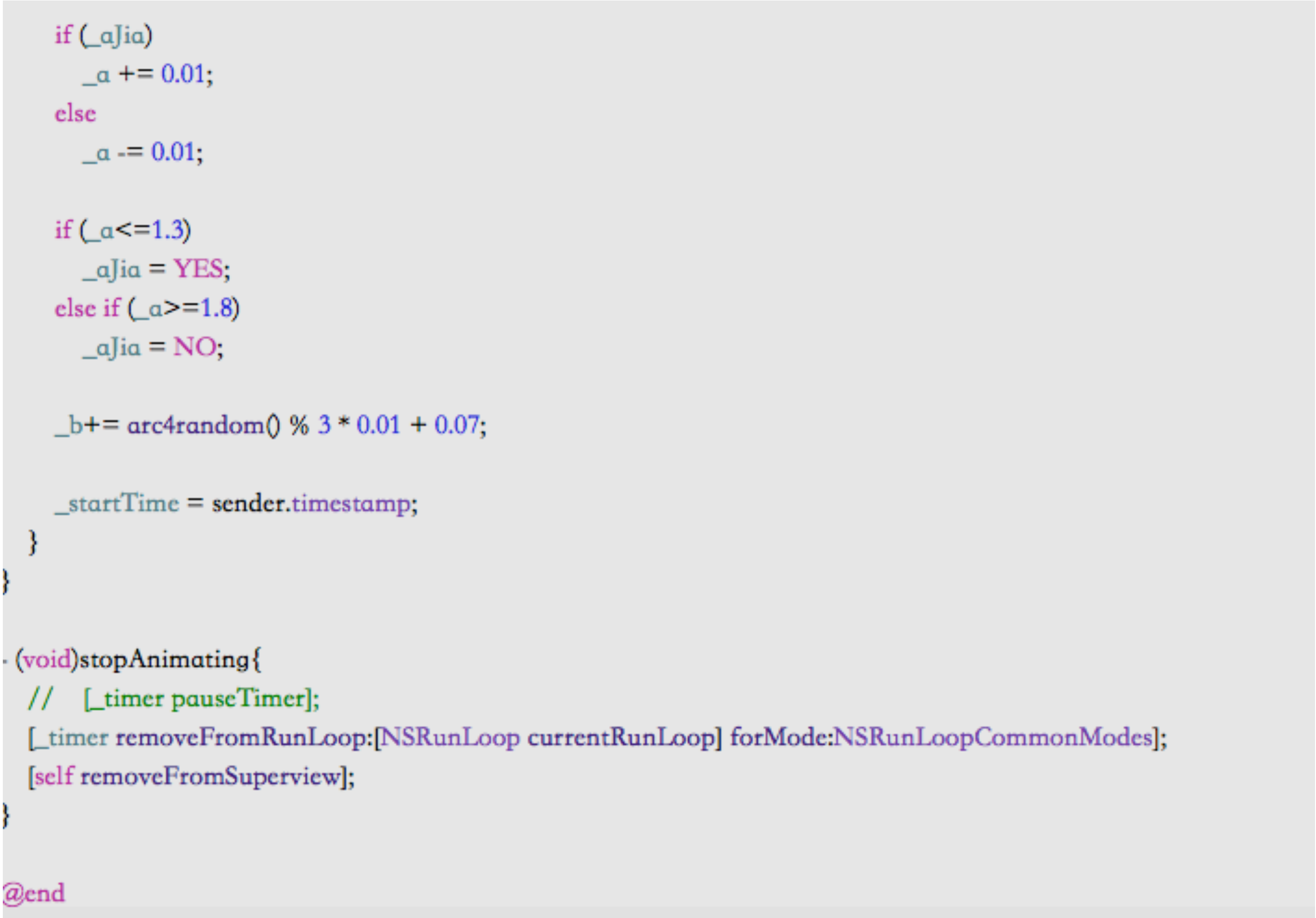
```
- (instancetype)init{
   self = [super init];
   if(self){
      UIImage *image = [UIImage imageNamed:@"shuaxin"];
      self.frame = CGRectMake((80-image.size.width)/2,(80-image.size.height)/2 ,image.size.width, image.size.height)
         ;
      _fillColor = [UIColor colorWithRed:21/255.0 green:169/255.0 blue:188/255.0 alpha:1];
      _currentLinePointY = self.frame.size.height;
      self.backgroundColor = [UIColor clearColor];
      _a = 1.8;
      _b = 0;
      _aJia = NO;
      _hJia = NO;

      _maskLayer = [[CALayer alloc] init];
      _maskLayer.frame = CGRectMake(0, 0, image.size.width, image.size.height);
      _maskLayer.contents = (__bridge id _Nullable)([UIImage imageNamed:@"shuaxin"].CGImage);
      [self.layer setMask:self.maskLayer];

      _timer = [CADisplayLink displayLinkWithTarget:self selector:@selector(updateCurrentLinePointY:)];
      _startTime = 0.0;
      _deltaTime = 0.0;
   }
   return self;
}
```



```
- (void)drawRect:(CGRect)rect {
   CGContextRef context = UIGraphicsGetCurrentContext();
   [self updateLevel:context andRect:rect];
}

- (void)updateLevel:(CGContextRef)context andRect:(CGRect)rect{
   float y = _currentLinePointY;
   //绘图
   CGContextMoveToPoint(context, 0, y);
   for(int x=0; x<=rect.size.width; x++){
      y = -_a * sin(1.0 * x / 100 * M_PI + 4 * _b/M_PI ) * 5 + _currentLinePointY;
      CGContextAddLineToPoint(context, x, y);
   }

   CGContextAddLineToPoint(context, rect.size.width, rect.size.height);
   CGContextAddLineToPoint( context, 0, rect.size.height);
   CGContextAddLineToPoint(context, 0, _currentLinePointY);

   [_fillColor set];//描色+填充色

   CGContextDrawPath(context, kCGPathFillStroke);//描线+填充路径
}
```



```
- (void)startAnimating{
    //    [_timer resumeTimer];
    [_timer addToRunLoop:[NSRunLoop currentRunLoop] forMode:NSRunLoopCommonModes];
}

- (void)updateCurrentLinePointY:(CADisplayLink *)sender{
    if(_startTime == 0){
        _startTime = sender.timestamp;
    }
    _deltaTime = sender.timestamp - _startTime;
    if(_deltaTime >= 0.015){
        [self setNeedsDisplay];

        if(_hJia)
            self.currentLinePointY += 0.2;
        else
            self.currentLinePointY -= 0.2;

        if(_currentLinePointY<=0)
            _hJia = YES;
        else if(_currentLinePointY>=30)
            _hJia = NO;
```



```
        if (_aJia)
            _a += 0.01;
        else
            _a -= 0.01;

        if (_a<=1.3)
            _aJia = YES;
        else if (_a>=1.8)
            _aJia = NO;

        _b+= arc4random() % 3 * 0.01 + 0.07;

        _startTime = sender.timestamp;
    }
}

- (void)stopAnimating{
    //    [_timer pauseTimer];
    [_timer removeFromRunLoop:[NSRunLoop currentRunLoop] forMode:NSRunLoopCommonModes];
    [self removeFromSuperview];
}

@end
```