图片轮播

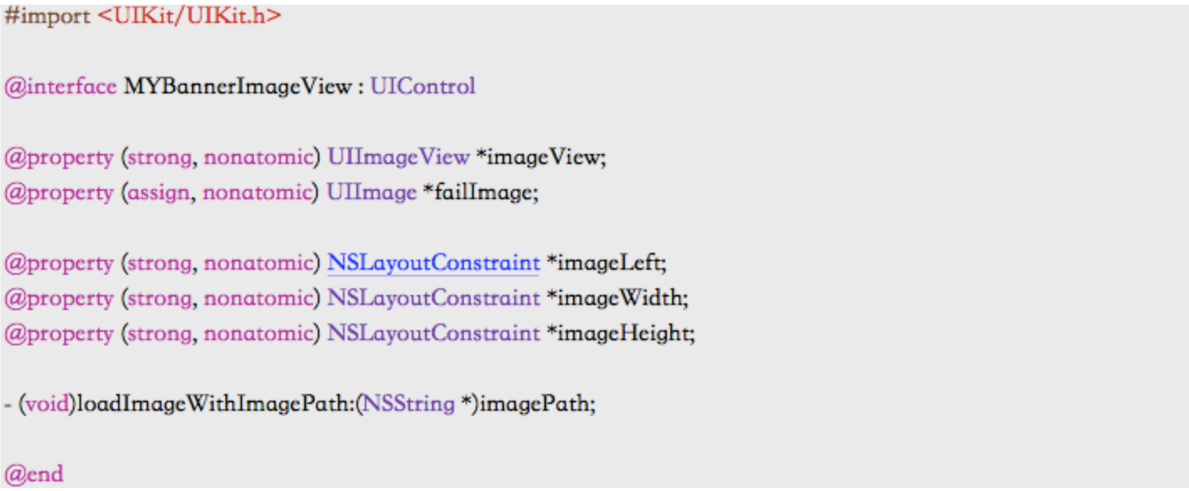
1. 简述：

三张图片实现图片轮播（带视差效果、下拉放大效果）

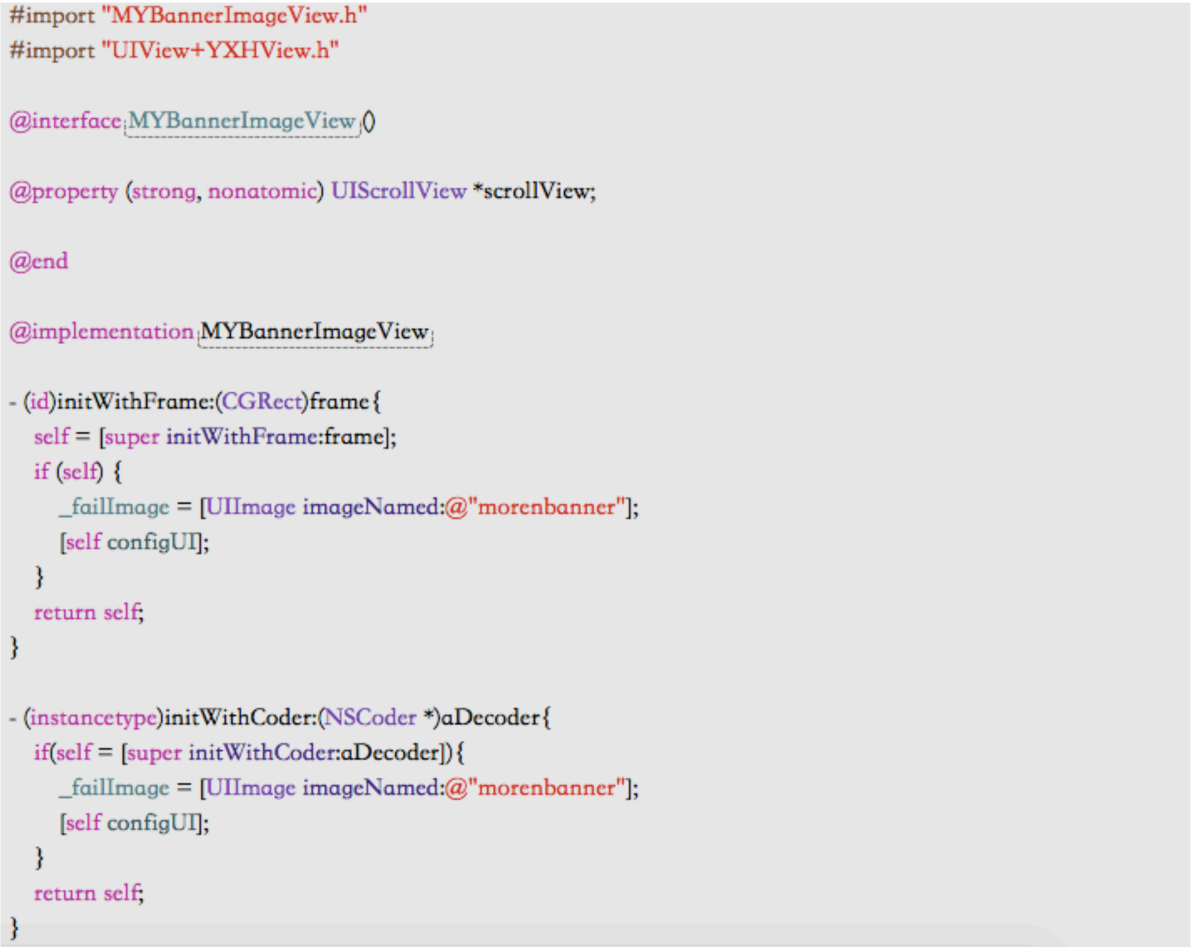
2. 原理：

始终中间图片为当前图片，在 scrollView 滚动结束时，重新加载三张图片 url，将下一个图片重新置于中间显示。

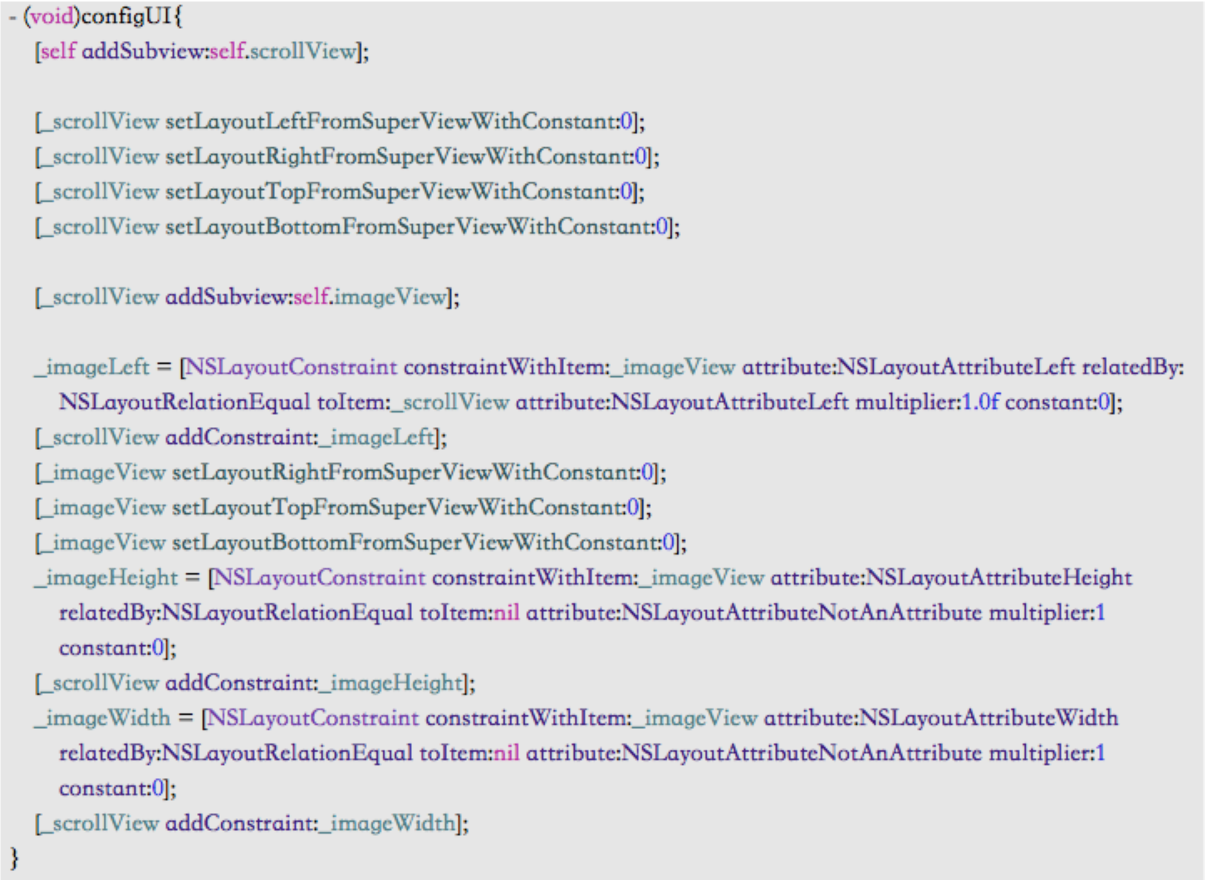
3. 实现：



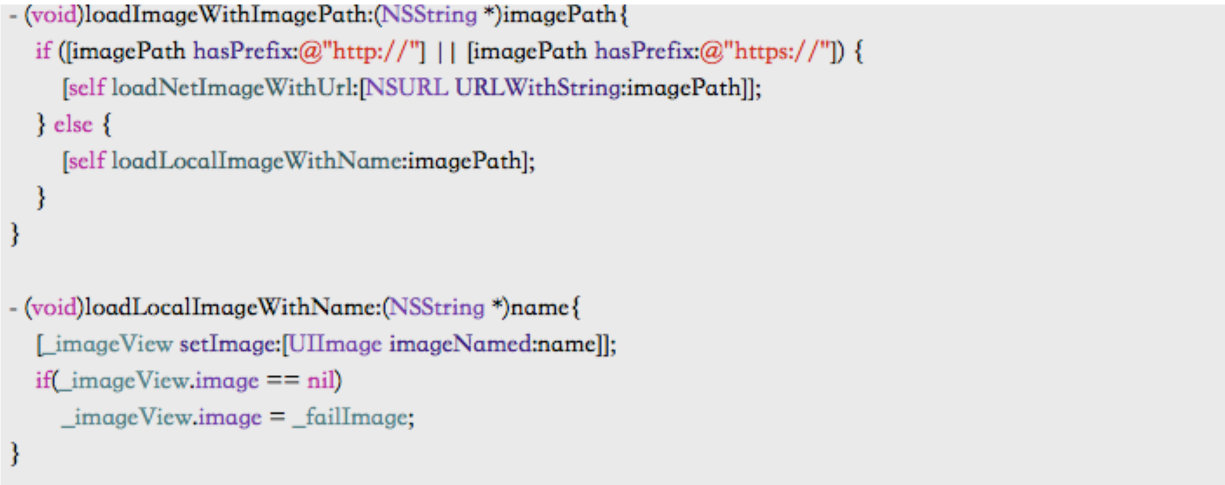
```
#import <UIKit/UIKit.h>

@interface MYBannerImageView : UIControl

@property (strong, nonatomic) UIImageView *imageView;
@property (assign, nonatomic) UIImage *failImage;

@property (strong, nonatomic) NSLayoutConstraint *imageLeft;
@property (strong, nonatomic) NSLayoutConstraint *imageWidth;
@property (strong, nonatomic) NSLayoutConstraint *imageHeight;

- (void)loadImageWithImagePath:(NSString *)imagePath;

@end
```



```
#import "MYBannerImageView.h"
#import "UIView+YXHView.h"

@interface MYBannerImageView ()

@property (strong, nonatomic) UIScrollView *scrollView;

@end

@implementation MYBannerImageView

- (id)initWithFrame:(CGRect)frame{
    self = [super initWithFrame:frame];
    if (self) {
        _failImage = [UIImage imageNamed:@"morenbanner"];
        [self configUI];
    }
    return self;
}

- (instancetype)initWithCoder:(NSCoder *)aDecoder{
    if(self = [super initWithCoder:aDecoder]){
        _failImage = [UIImage imageNamed:@"morenbanner"];
        [self configUI];
    }
    return self;
}
```



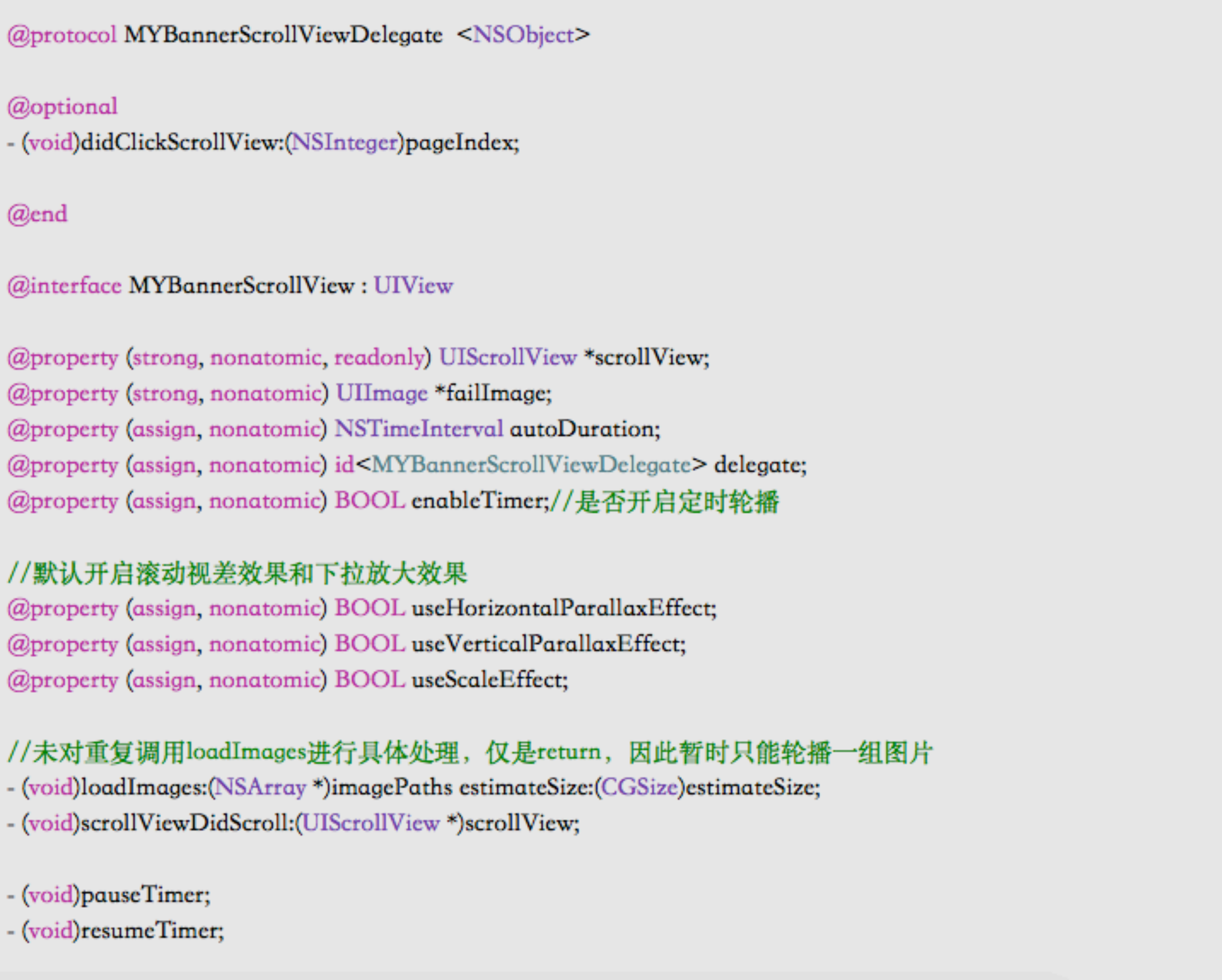
```
- (void)configUI{
    [self addSubview:self.scrollView];

    [_scrollView setLayoutLeftFromSuperViewWithConstant:0];
    [_scrollView setLayoutRightFromSuperViewWithConstant:0];
    [_scrollView setLayoutTopFromSuperViewWithConstant:0];
    [_scrollView setLayoutBottomFromSuperViewWithConstant:0];

    [_scrollView addSubview:self.imageView];

    _imageLeft = [NSLayoutConstraint constraintWithItem:_imageView attribute:NSLayoutAttributeLeft relatedBy:
        NSLayoutRelationEqual toItem:_scrollView attribute:NSLayoutAttributeLeft multiplier:1.0f constant:0];
    [_scrollView addConstraint:_imageLeft];
    [_imageView setLayoutRightFromSuperViewWithConstant:0];
    [_imageView setLayoutTopFromSuperViewWithConstant:0];
    [_imageView setLayoutBottomFromSuperViewWithConstant:0];
    _imageHeight = [NSLayoutConstraint constraintWithItem:_imageView attribute:NSLayoutAttributeHeight
        relatedBy:NSLayoutRelationEqual toItem:nil attribute:NSLayoutAttributeNotAnAttribute multiplier:1
        constant:0];
    [_scrollView addConstraint:_imageHeight];
    _imageWidth = [NSLayoutConstraint constraintWithItem:_imageView attribute:NSLayoutAttributeWidth
        relatedBy:NSLayoutRelationEqual toItem:nil attribute:NSLayoutAttributeNotAnAttribute multiplier:1
        constant:0];
    [_scrollView addConstraint:_imageWidth];
}
```



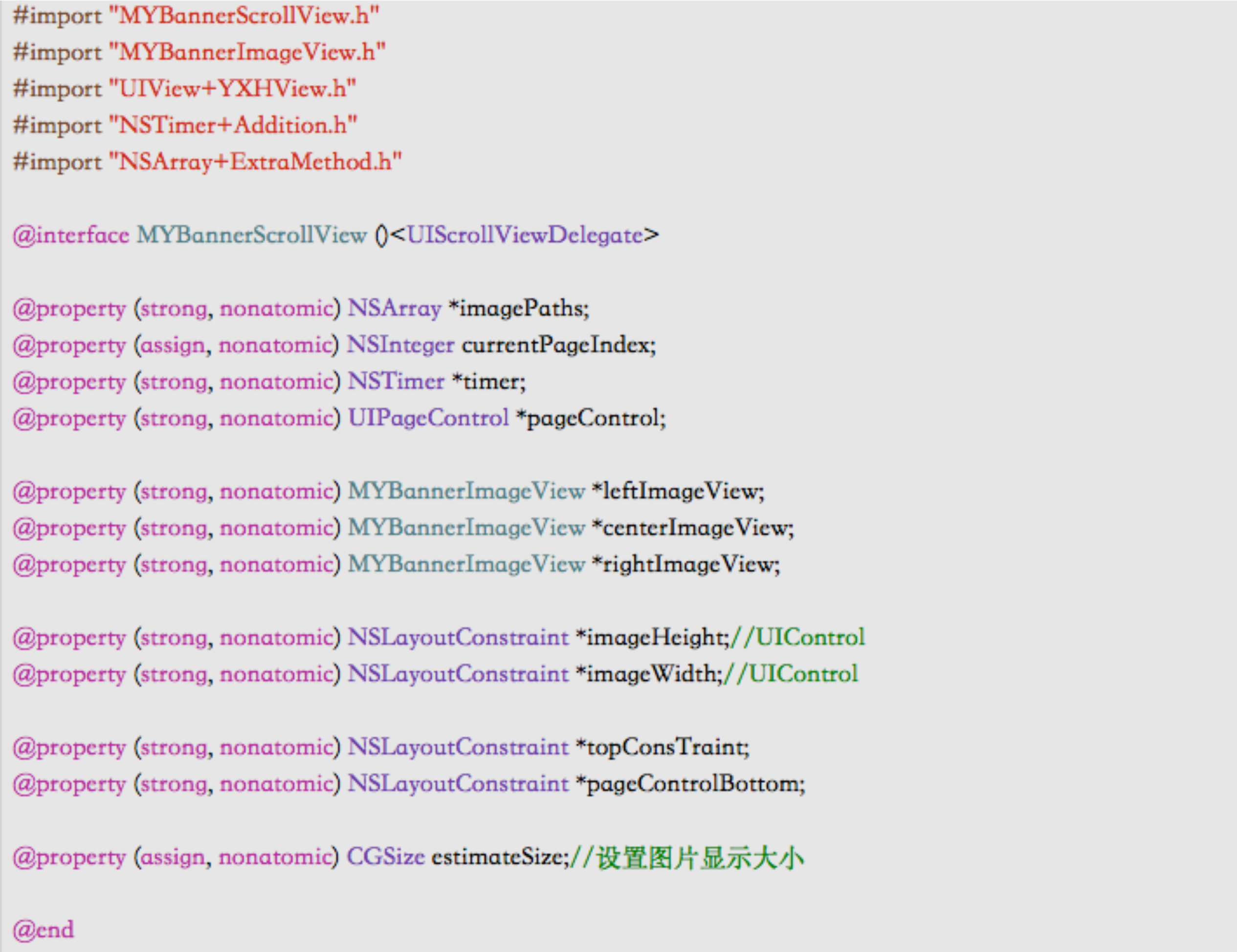
```
- (void)loadImageWithImagePath:(NSString *)imagePath{
    if ([imagePath hasPrefix:@"http://"] || [imagePath hasPrefix:@"https://"]) {
        [self loadNetImageWithUrl:[NSURL URLWithString:imagePath]];
    } else {
        [self loadLocalImageWithName:imagePath];
    }
}

- (void)loadLocalImageWithName:(NSString *)name{
    [_imageView setImage:[UIImage imageNamed:name]];
    if(_imageView.image == nil)
        _imageView.image = _failImage;
}
```

```
- (void)loadNetImageWithUrl:(NSURL *)url{
   if(_imageView.image == nil)
      _imageView.image = _failImage;

   UIImage *image = [[MYNetworkProxy defaultProxy] getImageIfExisted:url];

   if (image) {
      [_imageView setImage:image];
      return;
   }
      __weak typeof(self) weakSelf = self;
      [[MYNetworkProxy defaultProxy] fetchImageWithURL:url withFetchResultBlock:^(UIImage *fetchImage,
         BOOL isCache) {
       if(fetchImage != nil){
            [weakSelf.imageView setImage:fetchImage];
         }
      }];
}
```

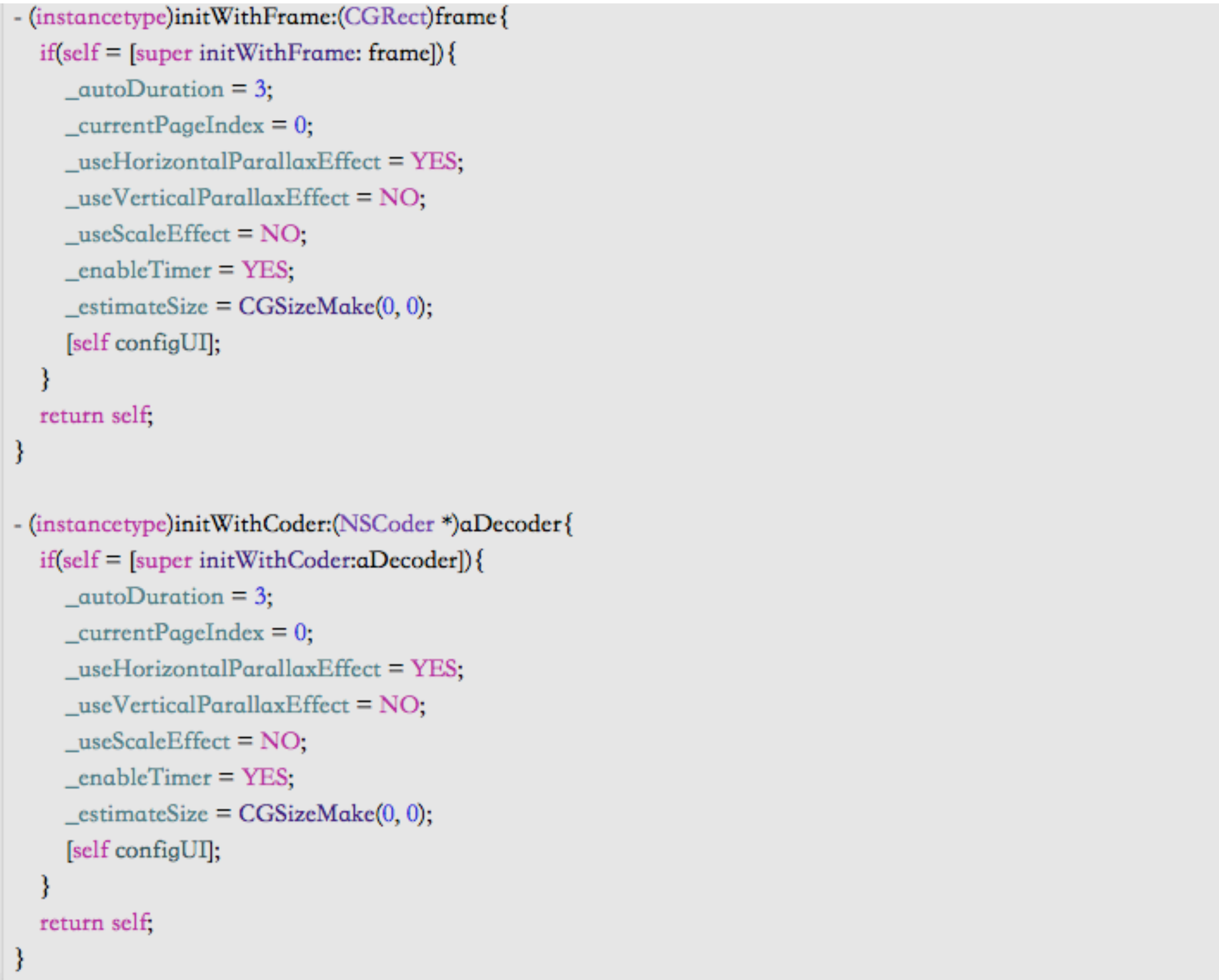
```
#pragma mark -Subviews

- (UIImageView *)imageView{
   if(!_imageView){
      _imageView = [[UIImageView alloc] init];
      [_imageView setTranslatesAutoresizingMaskIntoConstraints:NO];
      [_imageView setUserInteractionEnabled:NO];
      _imageView.contentMode = UIViewContentModeScaleAspectFill;
   }
   return _imageView;
}

- (UIScrollView *)scrollView{
   if(!_scrollView){
      _scrollView = [[UIScrollView alloc] init];
      [_scrollView setBounces:NO];
      [_scrollView setScrollEnabled:NO];
      [_scrollView setUserInteractionEnabled:NO];
      [_scrollView setTranslatesAutoresizingMaskIntoConstraints:NO];
   }
   return _scrollView;
}
```



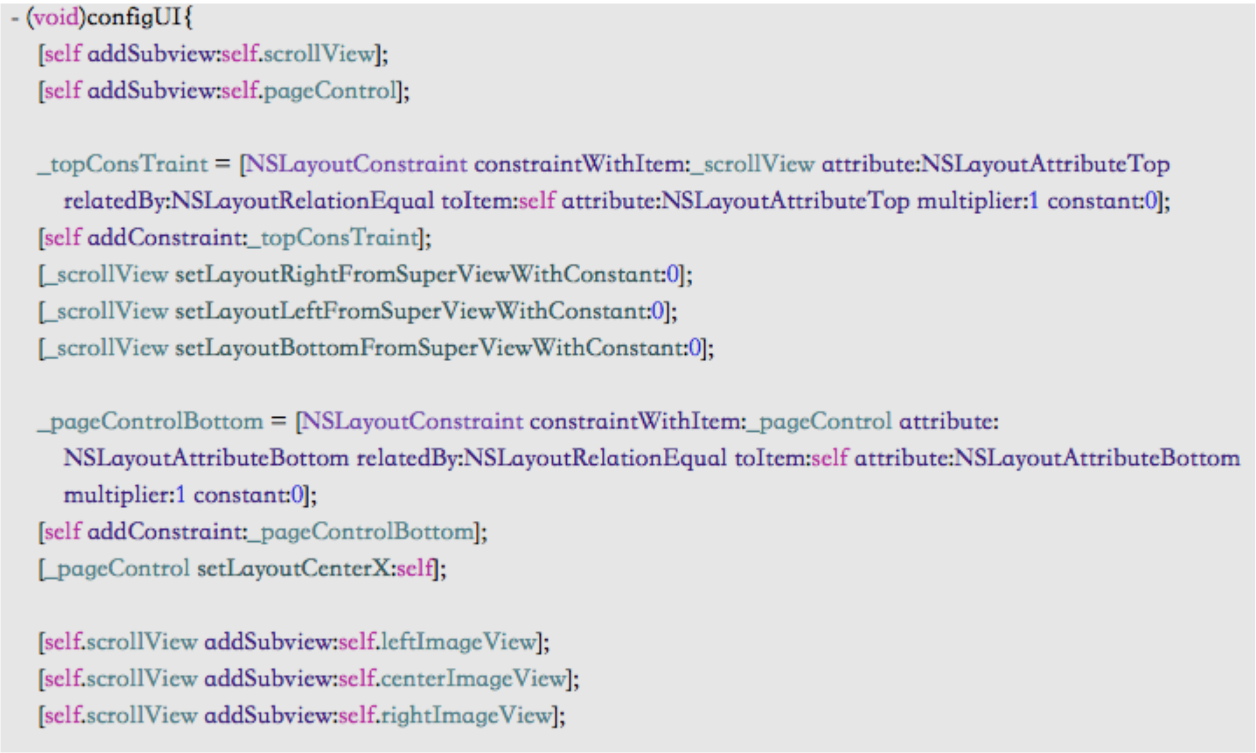
```
@protocol MYBannerScrollViewDelegate  <NSObject>

@optional
- (void)didClickScrollView:(NSInteger)pageIndex;

@end

@interface MYBannerScrollView : UIView

@property (strong, nonatomic, readonly) UIScrollView *scrollView;
@property (strong, nonatomic) UIImage *failImage;
@property (assign, nonatomic) NSTimeInterval autoDuration;
@property (assign, nonatomic) id<MYBannerScrollViewDelegate> delegate;
@property (assign, nonatomic) BOOL enableTimer;//是否开启定时轮播

//默认开启滚动视差效果和下拉放大效果
@property (assign, nonatomic) BOOL useHorizontalParallaxEffect;
@property (assign, nonatomic) BOOL useVerticalParallaxEffect;
@property (assign, nonatomic) BOOL useScaleEffect;

//未对重复调用loadImages进行具体处理，仅是return，因此暂时只能轮播一组图片
- (void)loadImages:(NSArray *)imagePaths estimateSize:(CGSize)estimateSize;
- (void)scrollViewDidScroll:(UIScrollView *)scrollView;

- (void)pauseTimer;
- (void)resumeTimer;
```



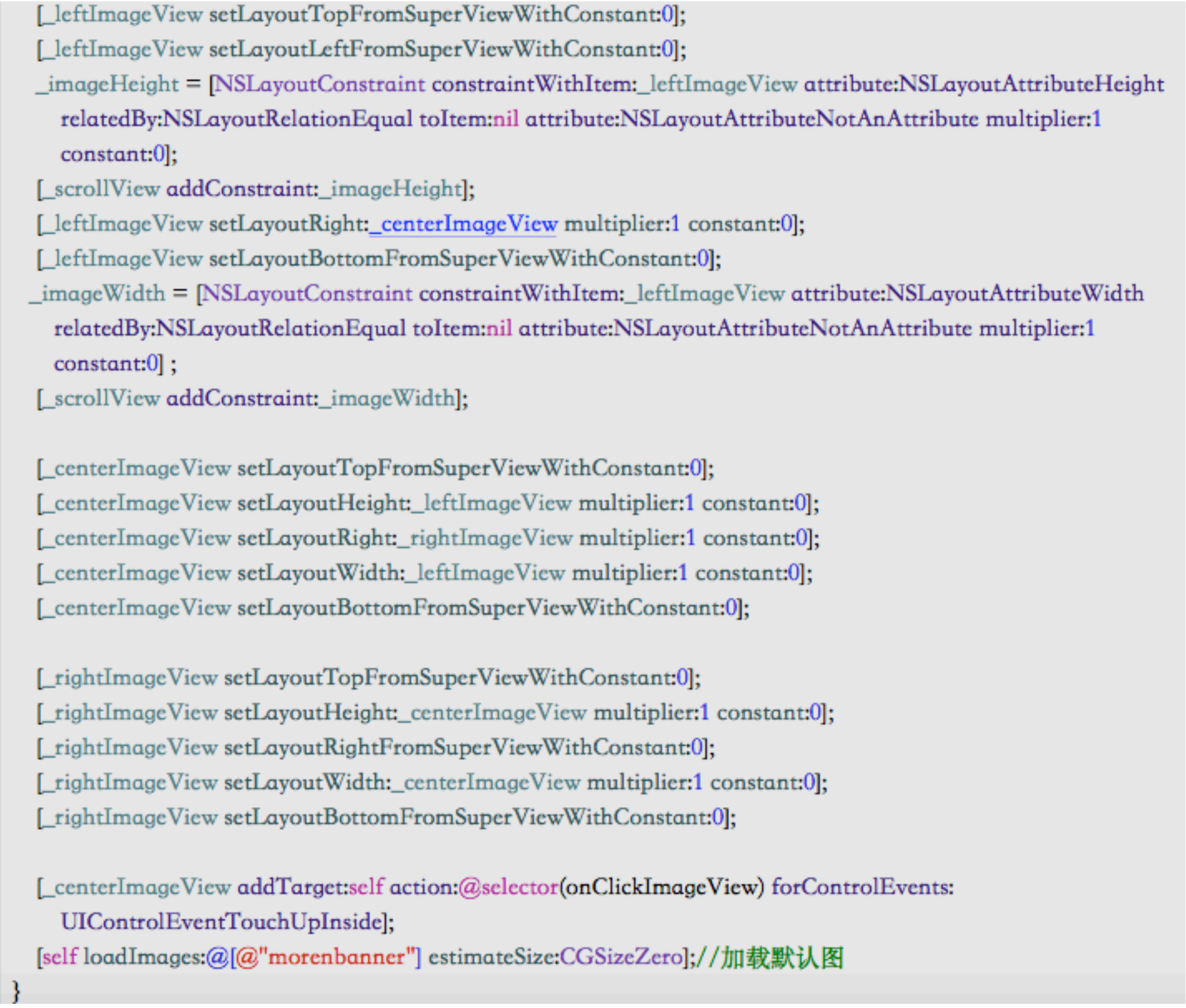
```
#import "MYBannerScrollView.h"
#import "MYBannerImageView.h"
#import "UIView+YXHView.h"
#import "NSTimer+Addition.h"
#import "NSArray+ExtraMethod.h"

@interface MYBannerScrollView ()<UIScrollViewDelegate>

@property (strong, nonatomic) NSArray *imagePaths;
@property (assign, nonatomic) NSInteger currentPageIndex;
@property (strong, nonatomic) NSTimer *timer;
@property (strong, nonatomic) UIPageControl *pageControl;

@property (strong, nonatomic) MYBannerImageView *leftImageView;
@property (strong, nonatomic) MYBannerImageView *centerImageView;
@property (strong, nonatomic) MYBannerImageView *rightImageView;

@property (strong, nonatomic) NSLayoutConstraint *imageHeight;//UIControl
@property (strong, nonatomic) NSLayoutConstraint *imageWidth;//UIControl

@property (strong, nonatomic) NSLayoutConstraint *topConsTraint;
@property (strong, nonatomic) NSLayoutConstraint *pageControlBottom;

@property (assign, nonatomic) CGSize estimateSize;//设置图片显示大小

@end
```



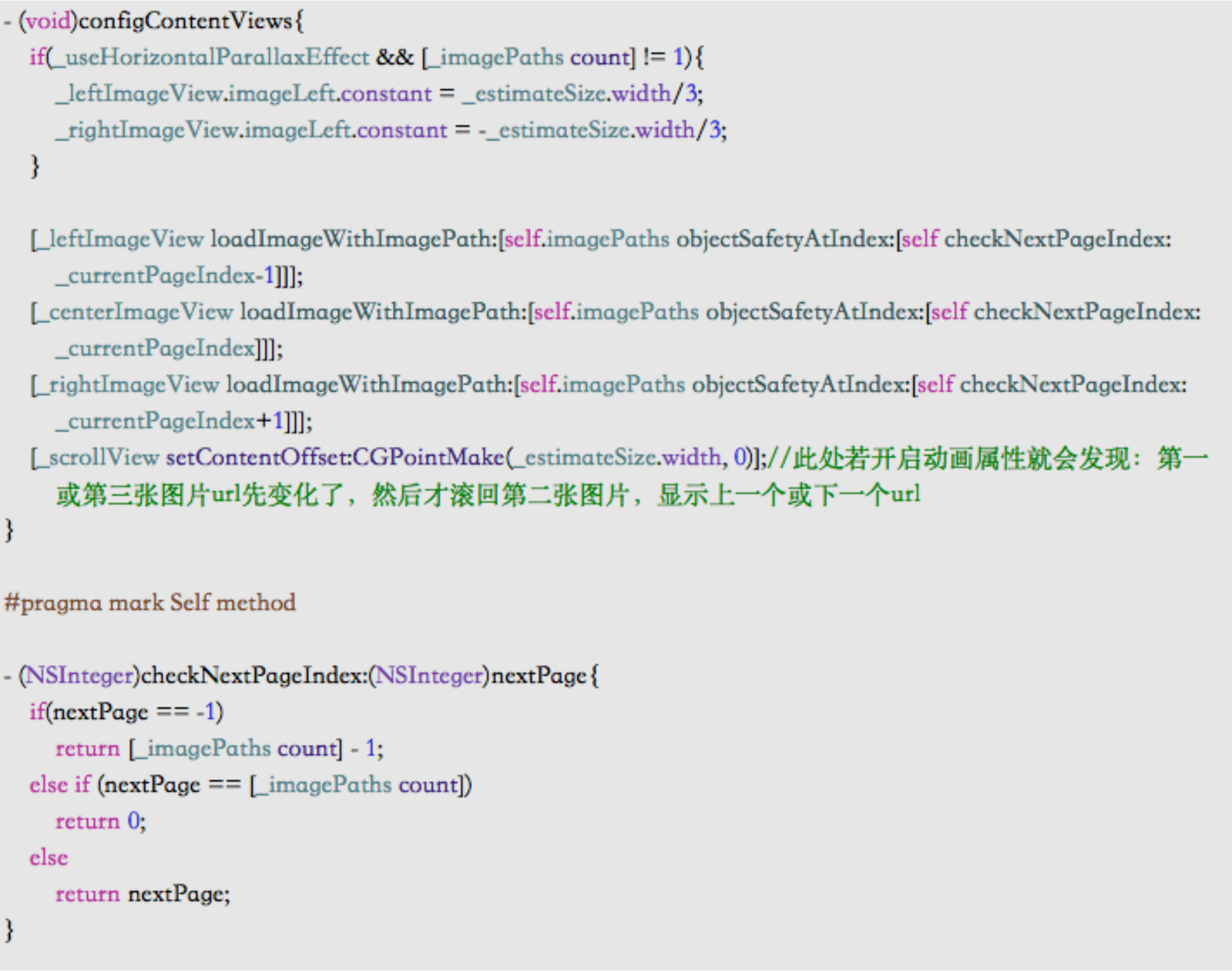
```
- (instancetype)initWithFrame:(CGRect)frame{
   if(self = [super initWithFrame: frame]){
      _autoDuration = 3;
      _currentPageIndex = 0;
      _useHorizontalParallaxEffect = YES;
      _useVerticalParallaxEffect = NO;
      _useScaleEffect = NO;
      _enableTimer = YES;
      _estimateSize = CGSizeMake(0, 0);
      [self configUI];
   }
   return self;
}

- (instancetype)initWithCoder:(NSCoder *)aDecoder{
   if(self = [super initWithCoder:aDecoder]){
      _autoDuration = 3;
      _currentPageIndex = 0;
      _useHorizontalParallaxEffect = YES;
      _useVerticalParallaxEffect = NO;
      _useScaleEffect = NO;
      _enableTimer = YES;
      _estimateSize = CGSizeMake(0, 0);
      [self configUI];
   }
   return self;
}
```



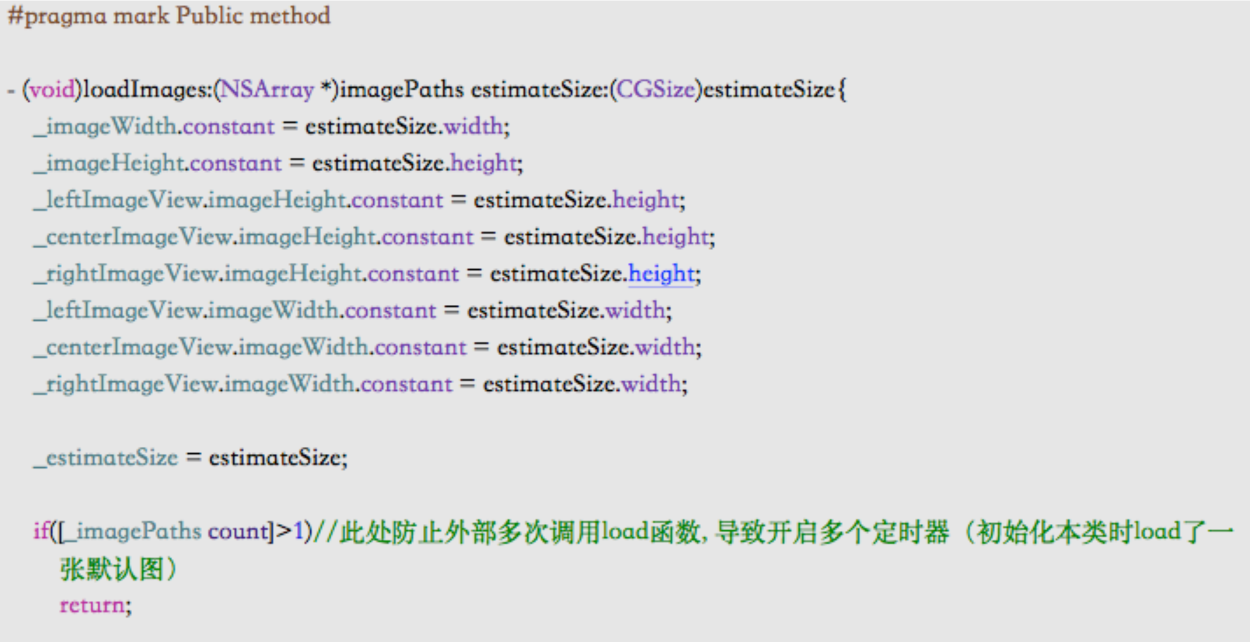
```
- (void)configUI{
   [self addSubview:self.scrollView];
   [self addSubview:self.pageControl];

   _topConsTraint = [NSLayoutConstraint constraintWithItem:_scrollView attribute:NSLayoutAttributeTop
      relatedBy:NSLayoutRelationEqual toItem:self attribute:NSLayoutAttributeTop multiplier:1 constant:0];
   [self addConstraint:_topConsTraint];
   [_scrollView setLayoutRightFromSuperViewWithConstant:0];
   [_scrollView setLayoutLeftFromSuperViewWithConstant:0];
   [_scrollView setLayoutBottomFromSuperViewWithConstant:0];

   _pageControlBottom = [NSLayoutConstraint constraintWithItem:_pageControl attribute:
      NSLayoutAttributeBottom relatedBy:NSLayoutRelationEqual toItem:self attribute:NSLayoutAttributeBottom
      multiplier:1 constant:0];
   [self addConstraint:_pageControlBottom];
   [_pageControl setLayoutCenterX:self];

   [self.scrollView addSubview:self.leftImageView];
   [self.scrollView addSubview:self.centerImageView];
   [self.scrollView addSubview:self.rightImageView];
```



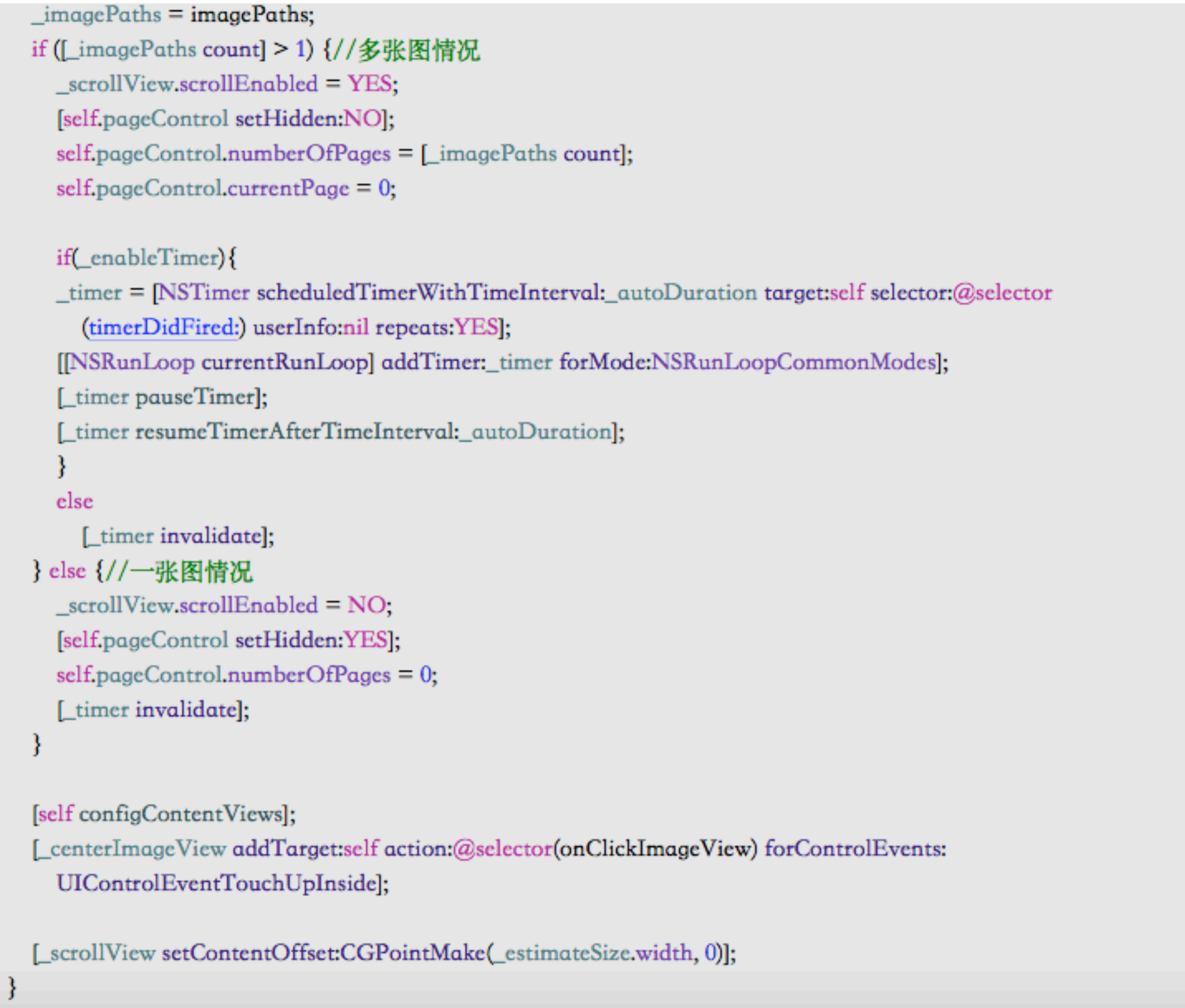
```
    [_leftImageView setLayoutTopFromSuperViewWithConstant:0];
    [_leftImageView setLayoutLeftFromSuperViewWithConstant:0];
    _imageHeight = [NSLayoutConstraint constraintWithItem:_leftImageView attribute:NSLayoutAttributeHeight
       relatedBy:NSLayoutRelationEqual toItem:nil attribute:NSLayoutAttributeNotAnAttribute multiplier:1
       constant:0];
    [_scrollView addConstraint:_imageHeight];
    [_leftImageView setLayoutRight:_centerImageView multiplier:1 constant:0];
    [_leftImageView setLayoutBottomFromSuperViewWithConstant:0];
   _imageWidth = [NSLayoutConstraint constraintWithItem:_leftImageView attribute:NSLayoutAttributeWidth
      relatedBy:NSLayoutRelationEqual toItem:nil attribute:NSLayoutAttributeNotAnAttribute multiplier:1
      constant:0] ;
    [_scrollView addConstraint:_imageWidth];

    [_centerImageView setLayoutTopFromSuperViewWithConstant:0];
    [_centerImageView setLayoutHeight:_leftImageView multiplier:1 constant:0];
    [_centerImageView setLayoutRight:_rightImageView multiplier:1 constant:0];
    [_centerImageView setLayoutWidth:_leftImageView multiplier:1 constant:0];
    [_centerImageView setLayoutBottomFromSuperViewWithConstant:0];

    [_rightImageView setLayoutTopFromSuperViewWithConstant:0];
    [_rightImageView setLayoutHeight:_centerImageView multiplier:1 constant:0];
    [_rightImageView setLayoutRightFromSuperViewWithConstant:0];
    [_rightImageView setLayoutWidth:_centerImageView multiplier:1 constant:0];
    [_rightImageView setLayoutBottomFromSuperViewWithConstant:0];

    [_centerImageView addTarget:self action:@selector(onClickImageView) forControlEvents:
       UIControlEventTouchUpInside];
    [self loadImages:@[@"morenbanner"] estimateSize:CGSizeZero];//加载默认图
}
```



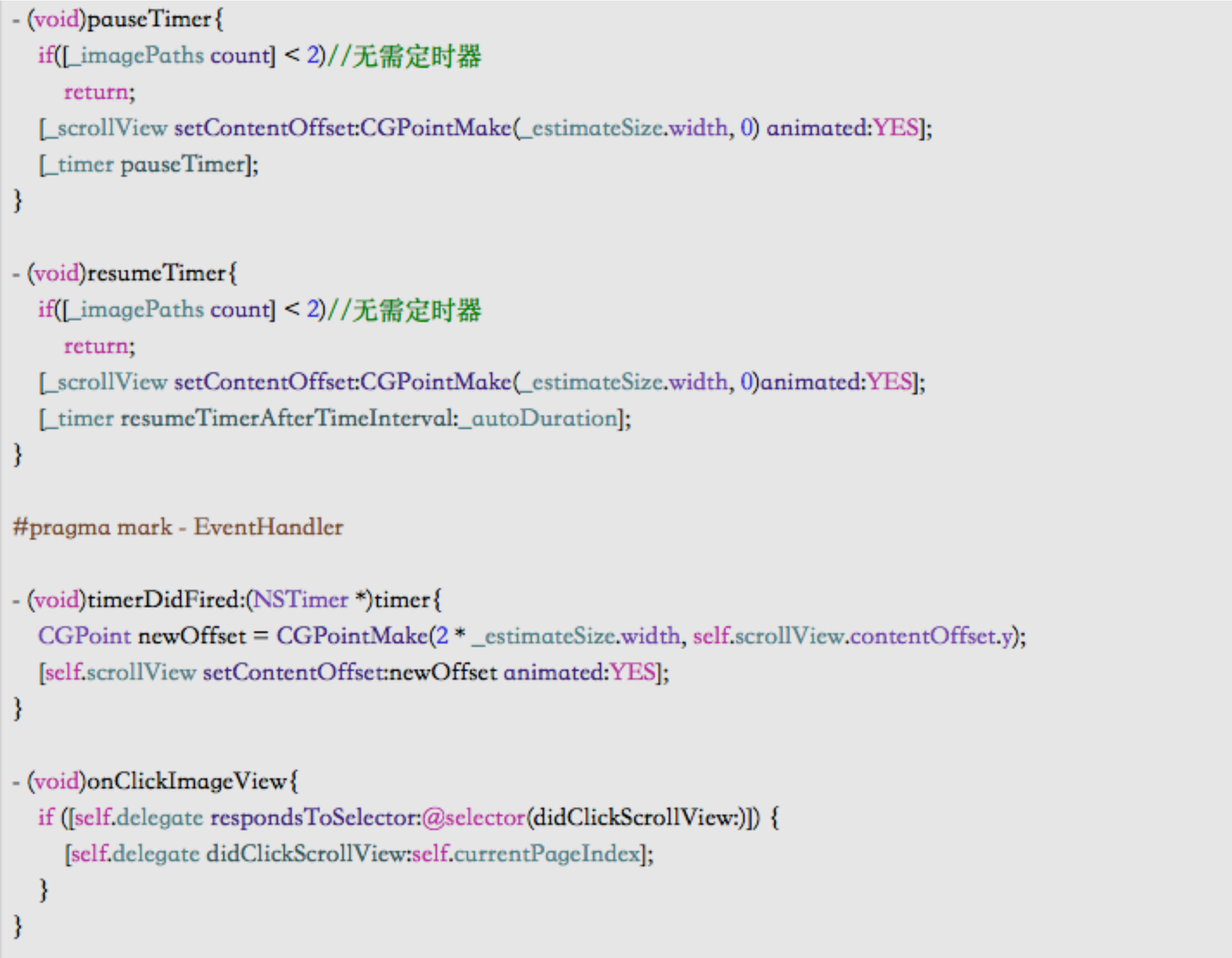
```
- (void)configContentViews{
    if(_useHorizontalParallaxEffect && [_imagePaths count] != 1){
        _leftImageView.imageLeft.constant = _estimateSize.width/3;
        _rightImageView.imageLeft.constant = -_estimateSize.width/3;
    }

    [_leftImageView loadImageWithImagePath:[self.imagePaths objectSafetyAtIndex:[self checkNextPageIndex:
        _currentPageIndex-1]]];
    [_centerImageView loadImageWithImagePath:[self.imagePaths objectSafetyAtIndex:[self checkNextPageIndex:
        _currentPageIndex]]];
    [_rightImageView loadImageWithImagePath:[self.imagePaths objectSafetyAtIndex:[self checkNextPageIndex:
        _currentPageIndex+1]]];
    [_scrollView setContentOffset:CGPointMake(_estimateSize.width, 0)];//此处若开启动画属性就会发现：第一
        或第三张图片url先变化了，然后才滚回第二张图片，显示上一个或下一个url
}

#pragma mark Self method

- (NSInteger)checkNextPageIndex:(NSInteger)nextPage{
    if(nextPage == -1)
        return [_imagePaths count] - 1;
    else if (nextPage == [_imagePaths count])
        return 0;
    else
        return nextPage;
}
```



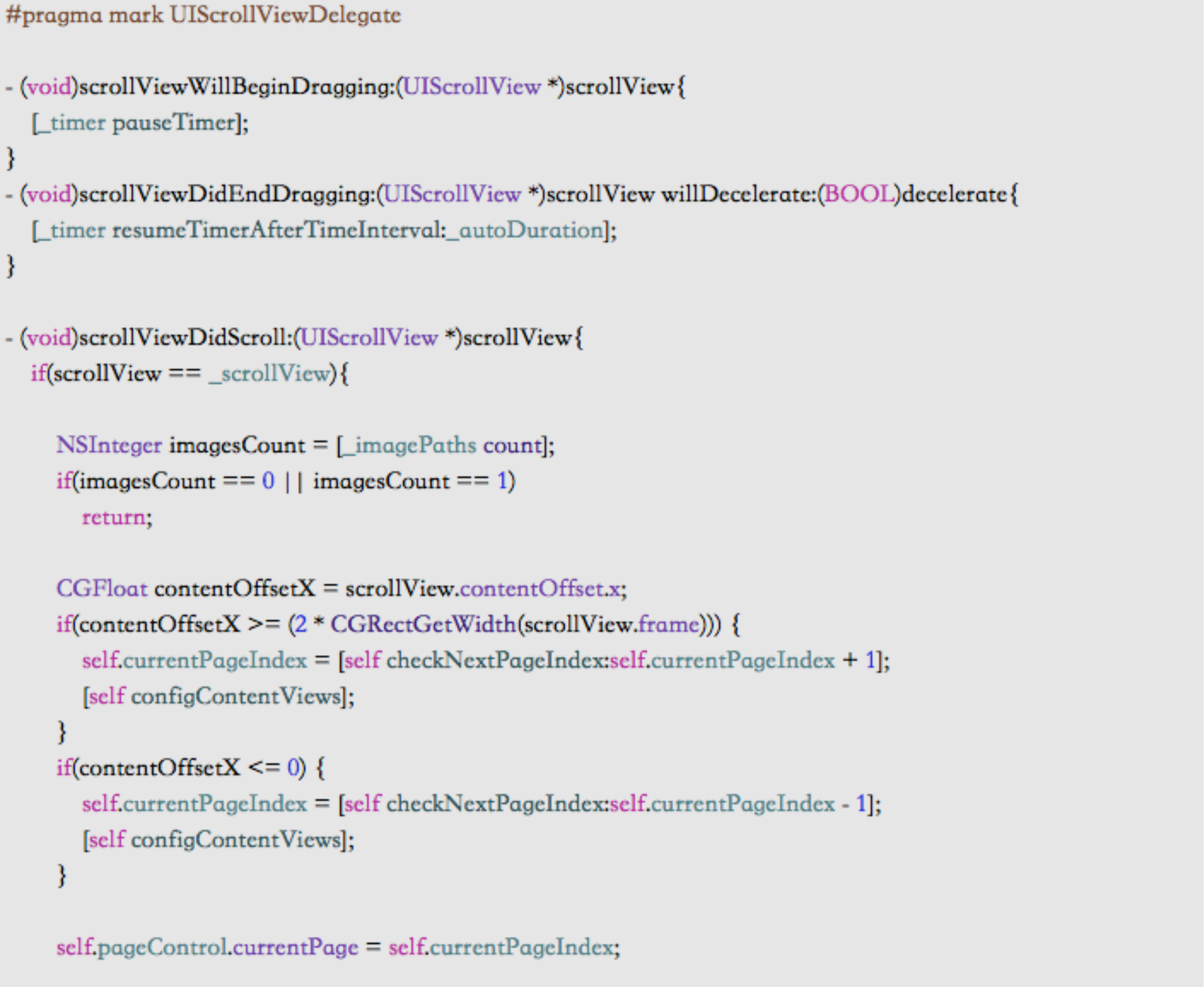
```
#pragma mark Public method

- (void)loadImages:(NSArray *)imagePaths estimateSize:(CGSize)estimateSize{
    _imageWidth.constant = estimateSize.width;
    _imageHeight.constant = estimateSize.height;
    _leftImageView.imageHeight.constant = estimateSize.height;
    _centerImageView.imageHeight.constant = estimateSize.height;
    _rightImageView.imageHeight.constant = estimateSize.height;
    _leftImageView.imageWidth.constant = estimateSize.width;
    _centerImageView.imageWidth.constant = estimateSize.width;
    _rightImageView.imageWidth.constant = estimateSize.width;

    _estimateSize = estimateSize;

    if([_imagePaths count]>1)//此处防止外部多次调用load函数,导致开启多个定时器（初始化本类时load了一
        张默认图）
        return;
```



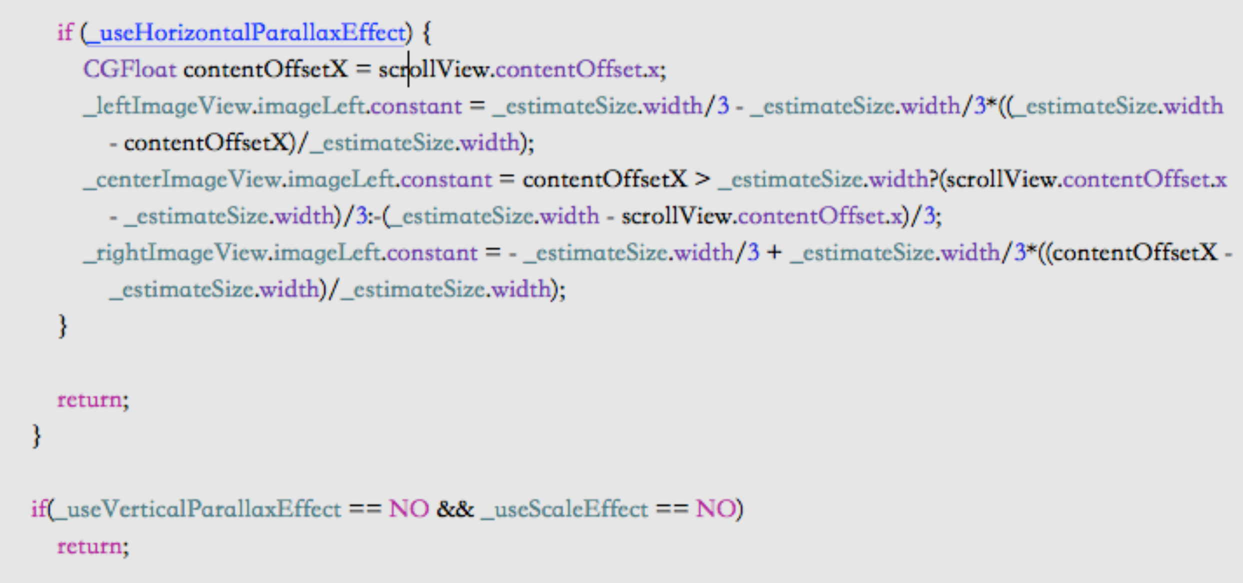
```
    _imagePaths = imagePaths;
    if ([_imagePaths count] > 1) {//多张图情况
        _scrollView.scrollEnabled = YES;
        [self.pageControl setHidden:NO];
        self.pageControl.numberOfPages = [_imagePaths count];
        self.pageControl.currentPage = 0;

        if(_enableTimer){
        _timer = [NSTimer scheduledTimerWithTimeInterval:_autoDuration target:self selector:@selector
            (timerDidFired:) userInfo:nil repeats:YES];
        [[NSRunLoop currentRunLoop] addTimer:_timer forMode:NSRunLoopCommonModes];
        [_timer pauseTimer];
        [_timer resumeTimerAfterTimeInterval:_autoDuration];
        }
        else
            [_timer invalidate];
    } else {//一张图情况
        _scrollView.scrollEnabled = NO;
        [self.pageControl setHidden:YES];
        self.pageControl.numberOfPages = 0;
        [_timer invalidate];
    }

    [self configContentViews];
    [_centerImageView addTarget:self action:@selector(onClickImageView) forControlEvents:
        UIControlEventTouchUpInside];

    [_scrollView setContentOffset:CGPointMake(_estimateSize.width, 0)];
}
```



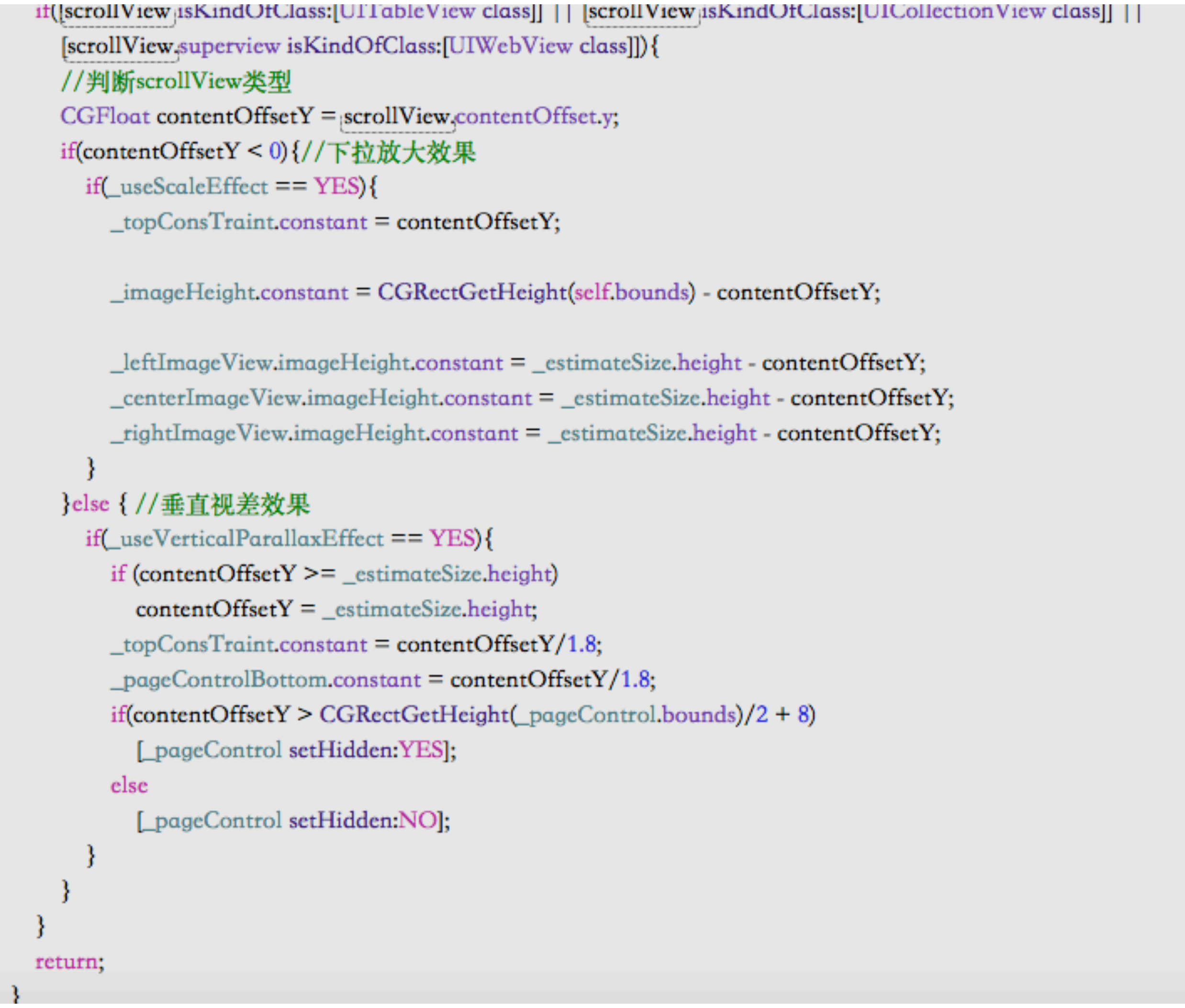
```
- (void)pauseTimer{
    if([_imagePaths count] < 2)//无需定时器
        return;
    [_scrollView setContentOffset:CGPointMake(_estimateSize.width, 0) animated:YES];
    [_timer pauseTimer];
}

- (void)resumeTimer{
    if([_imagePaths count] < 2)//无需定时器
        return;
    [_scrollView setContentOffset:CGPointMake(_estimateSize.width, 0)animated:YES];
    [_timer resumeTimerAfterTimeInterval:_autoDuration];
}

#pragma mark - EventHandler

- (void)timerDidFired:(NSTimer *)timer{
    CGPoint newOffset = CGPointMake(2 * _estimateSize.width, self.scrollView.contentOffset.y);
    [self.scrollView setContentOffset:newOffset animated:YES];
}

- (void)onClickImageView{
    if ([self.delegate respondsToSelector:@selector(didClickScrollView:)]) {
        [self.delegate didClickScrollView:self.currentPageIndex];
    }
}
```



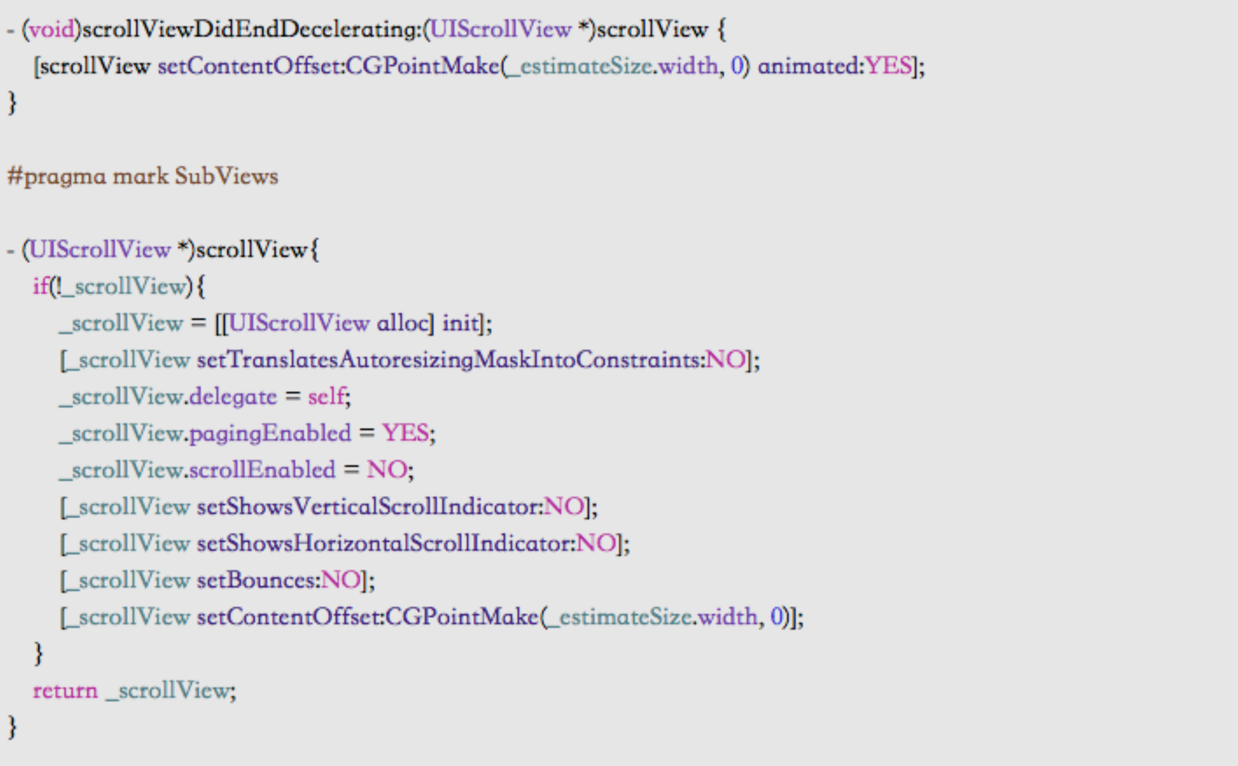
```
#pragma mark UIScrollViewDelegate

- (void)scrollViewWillBeginDragging:(UIScrollView *)scrollView{
   [_timer pauseTimer];
}
- (void)scrollViewDidEndDragging:(UIScrollView *)scrollView willDecelerate:(BOOL)decelerate{
   [_timer resumeTimerAfterTimeInterval:_autoDuration];
}

- (void)scrollViewDidScroll:(UIScrollView *)scrollView{
   if(scrollView == _scrollView){

      NSInteger imagesCount = [_imagePaths count];
      if(imagesCount == 0 || imagesCount == 1)
         return;

      CGFloat contentOffsetX = scrollView.contentOffset.x;
      if(contentOffsetX >= (2 * CGRectGetWidth(scrollView.frame))) {
         self.currentPageIndex = [self checkNextPageIndex:self.currentPageIndex + 1];
         [self configContentViews];
      }
      if(contentOffsetX <= 0) {
         self.currentPageIndex = [self checkNextPageIndex:self.currentPageIndex - 1];
         [self configContentViews];
      }

      self.pageControl.currentPage = self.currentPageIndex;
```



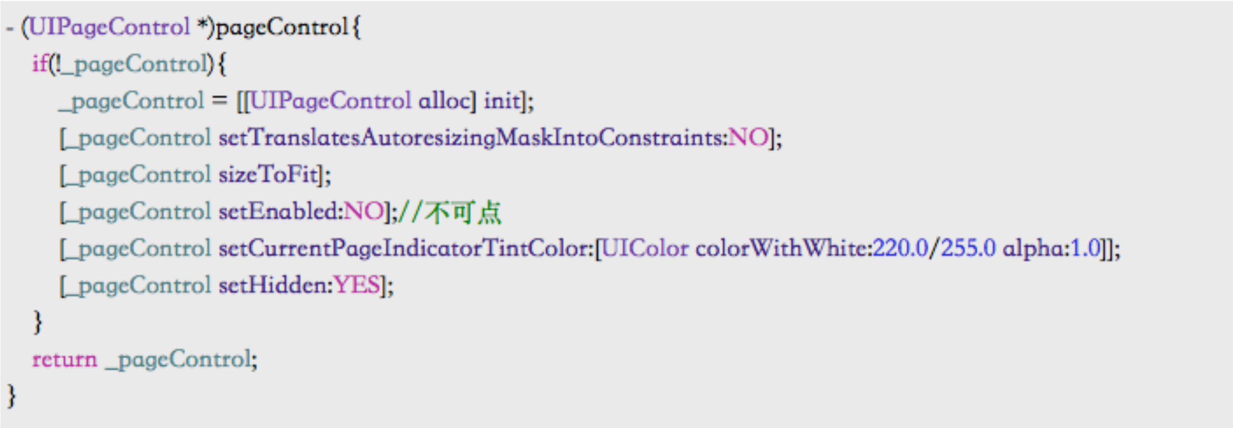
```
      if (_useHorizontalParallaxEffect) {
         CGFloat contentOffsetX = scrollView.contentOffset.x;
         _leftImageView.imageLeft.constant = _estimateSize.width/3 - _estimateSize.width/3*((_estimateSize.width
            - contentOffsetX)/_estimateSize.width);
         _centerImageView.imageLeft.constant = contentOffsetX > _estimateSize.width?(scrollView.contentOffset.x
            - _estimateSize.width)/3:-(_estimateSize.width - scrollView.contentOffset.x)/3;
         _rightImageView.imageLeft.constant = - _estimateSize.width/3 + _estimateSize.width/3*((contentOffsetX -
            _estimateSize.width)/_estimateSize.width);
      }

      return;
   }

   if(_useVerticalParallaxEffect == NO && _useScaleEffect == NO)
      return;
```



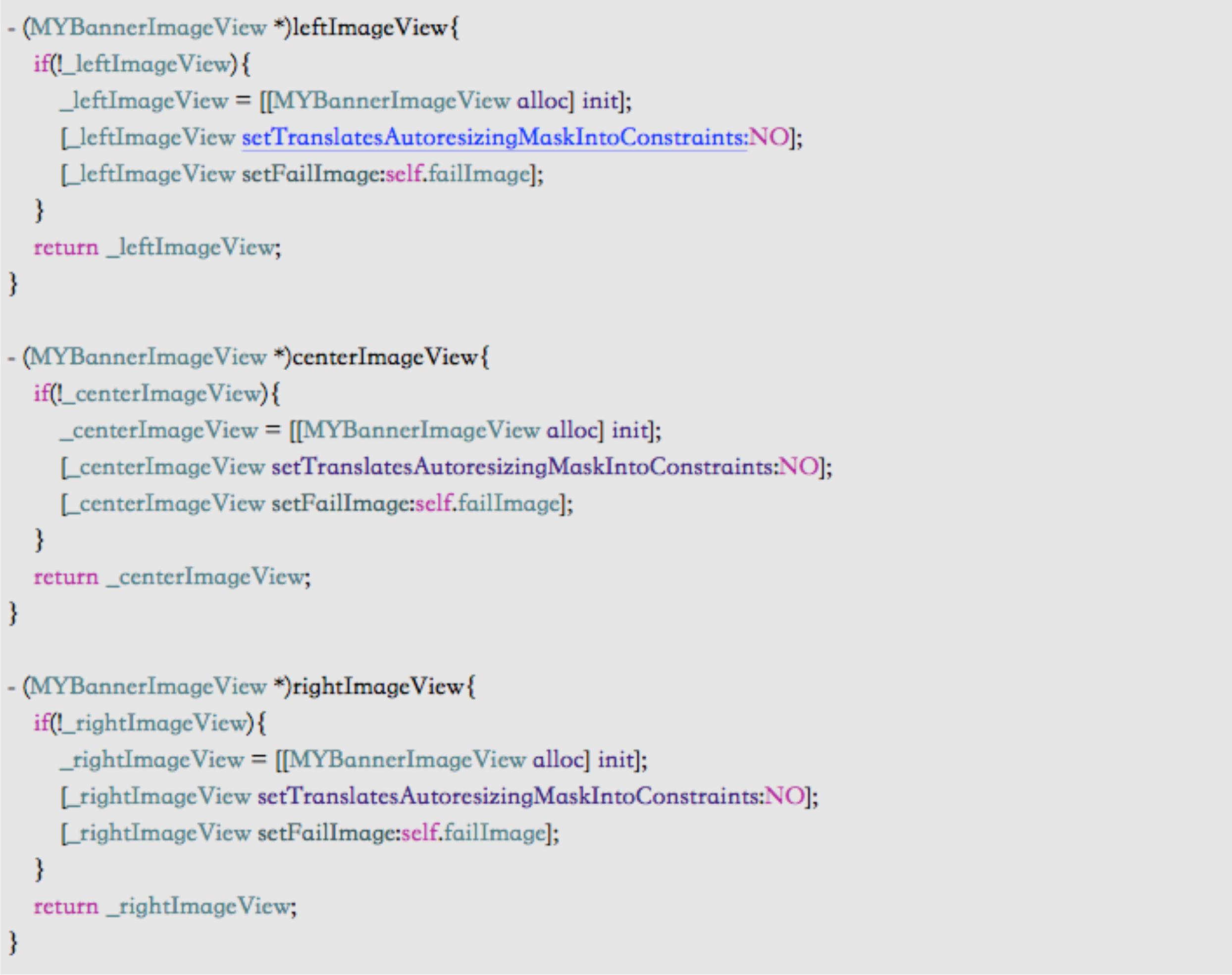
```
    if([scrollView isKindOfClass:[UITableView class]] || [scrollView isKindOfClass:[UICollectionView class]] ||
        [scrollView.superview isKindOfClass:[UIWebView class]]){
        //判断scrollView类型
        CGFloat contentOffsetY = scrollView.contentOffset.y;
        if(contentOffsetY < 0){//下拉放大效果
            if(_useScaleEffect == YES){
                _topConsTraint.constant = contentOffsetY;

                _imageHeight.constant = CGRectGetHeight(self.bounds) - contentOffsetY;

                _leftImageView.imageHeight.constant = _estimateSize.height - contentOffsetY;
                _centerImageView.imageHeight.constant = _estimateSize.height - contentOffsetY;
                _rightImageView.imageHeight.constant = _estimateSize.height - contentOffsetY;
            }
        }else { //垂直视差效果
            if(_useVerticalParallaxEffect == YES){
                if (contentOffsetY >= _estimateSize.height)
                    contentOffsetY = _estimateSize.height;
                _topConsTraint.constant = contentOffsetY/1.8;
                _pageControlBottom.constant = contentOffsetY/1.8;
                if(contentOffsetY > CGRectGetHeight(_pageControl.bounds)/2 + 8)
                    [_pageControl setHidden:YES];
                else
                    [_pageControl setHidden:NO];
            }
        }
    }
    return;
}
```



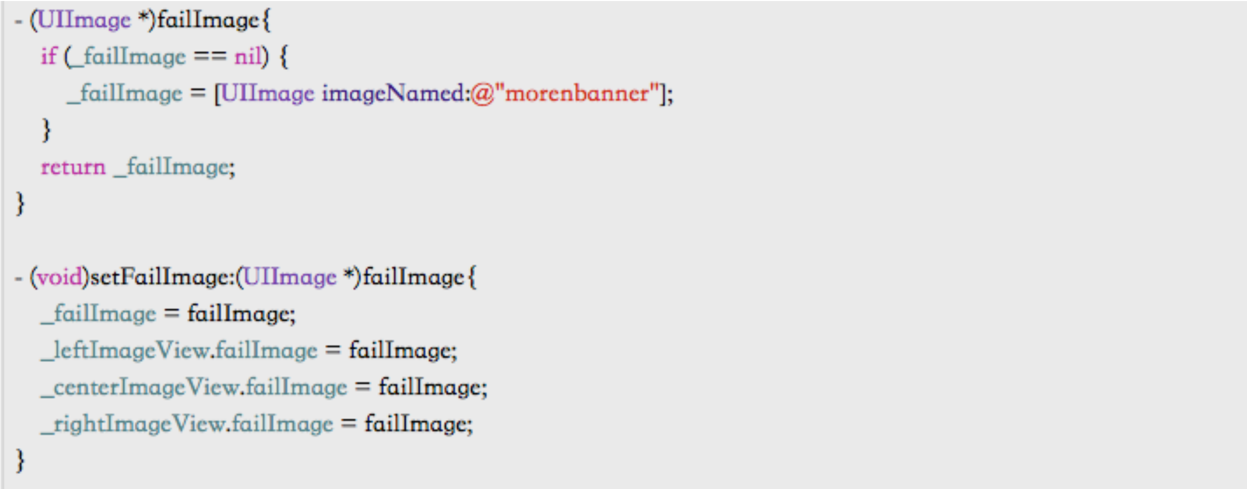
```
- (void)scrollViewDidEndDecelerating:(UIScrollView *)scrollView {
    [scrollView setContentOffset:CGPointMake(_estimateSize.width, 0) animated:YES];
}

#pragma mark SubViews

- (UIScrollView *)scrollView{
    if(!_scrollView){
        _scrollView = [[UIScrollView alloc] init];
        [_scrollView setTranslatesAutoresizingMaskIntoConstraints:NO];
        _scrollView.delegate = self;
        _scrollView.pagingEnabled = YES;
        _scrollView.scrollEnabled = NO;
        [_scrollView setShowsVerticalScrollIndicator:NO];
        [_scrollView setShowsHorizontalScrollIndicator:NO];
        [_scrollView setBounces:NO];
        [_scrollView setContentOffset:CGPointMake(_estimateSize.width, 0)];
    }
    return _scrollView;
}
```



```
- (UIPageControl *)pageControl{
    if(!_pageControl){
        _pageControl = [[UIPageControl alloc] init];
        [_pageControl setTranslatesAutoresizingMaskIntoConstraints:NO];
        [_pageControl sizeToFit];
        [_pageControl setEnabled:NO];//不可点
        [_pageControl setCurrentPageIndicatorTintColor:[UIColor colorWithWhite:220.0/255.0 alpha:1.0]];
        [_pageControl setHidden:YES];
    }
    return _pageControl;
}
```



```
- (MYBannerImageView *)leftImageView{
    if(!_leftImageView){
        _leftImageView = [[MYBannerImageView alloc] init];
        [_leftImageView setTranslatesAutoresizingMaskIntoConstraints:NO];
        [_leftImageView setFailImage:self.failImage];
    }
    return _leftImageView;
}

- (MYBannerImageView *)centerImageView{
    if(!_centerImageView){
        _centerImageView = [[MYBannerImageView alloc] init];
        [_centerImageView setTranslatesAutoresizingMaskIntoConstraints:NO];
        [_centerImageView setFailImage:self.failImage];
    }
    return _centerImageView;
}

- (MYBannerImageView *)rightImageView{
    if(!_rightImageView){
        _rightImageView = [[MYBannerImageView alloc] init];
        [_rightImageView setTranslatesAutoresizingMaskIntoConstraints:NO];
        [_rightImageView setFailImage:self.failImage];
    }
    return _rightImageView;
}
```



```
- (UIImage *)failImage{
    if (_failImage == nil) {
        _failImage = [UIImage imageNamed:@"morenbanner"];
    }
    return _failImage;
}

- (void)setFailImage:(UIImage *)failImage{
    _failImage = failImage;
    _leftImageView.failImage = failImage;
    _centerImageView.failImage = failImage;
    _rightImageView.failImage = failImage;
}
```