

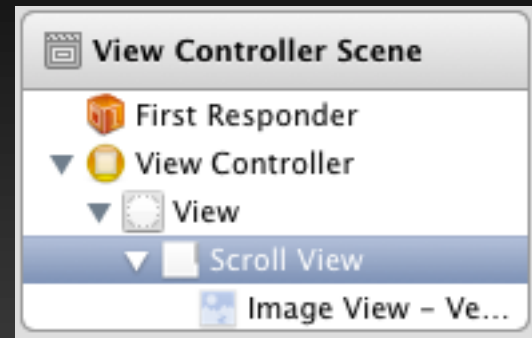
# UIScrollView



Trinh Minh Cuong

# Tạo một zooming scroll view

1. Đặt UIImageView nằm trong UIScrollView
2. UIScrollView.delegate trở tới UIViewController
3. UIViewController tuân theo UIScrollViewDelegate

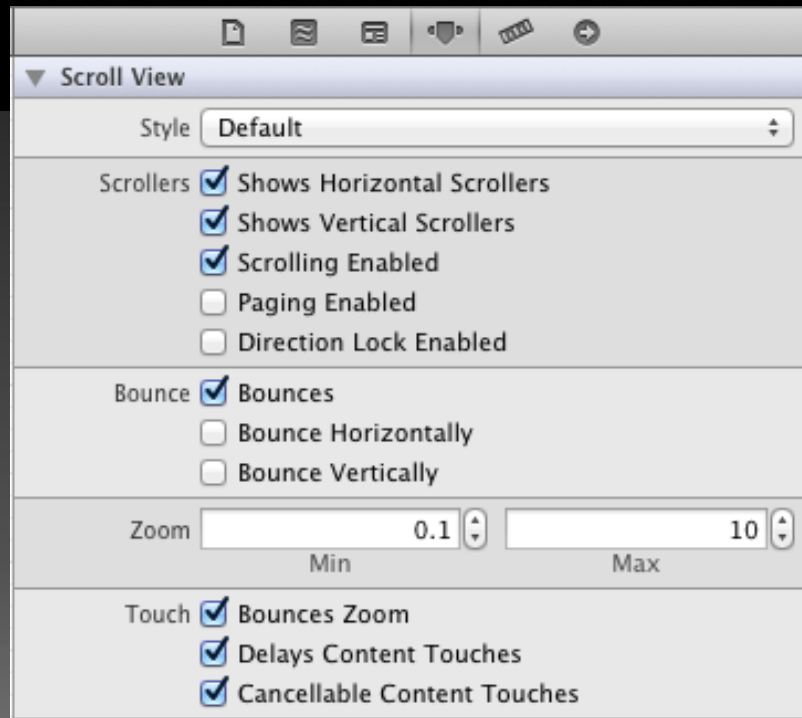


```
@interface ViewController : UIViewController <UIScrollViewDelegate>
```

- Viết hàm này, trả về UIImageView chứa ảnh còn phóng to, thu nhỏ

```
- (UIView *)viewForZoomingInScrollView:(UIScrollView *)scrollView
{
    return self.photo;
}
```

- Điều chỉnh hệ số Zoom



# Không sử dụng cách kéo, thả, cấu hình trên StoryBoard

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.imageView = [[UIImageView alloc] initWithImage:[UIImage
imageName:@"Ventura.jpg"]];

    [self.scrollView
setContentSize:CGSizeMake(self.imageView.frame.size.width,
self.imageView.frame.size.height)];

    self.scrollView.minimumZoomScale = 0.2;
    self.scrollView.maximumZoomScale = 4.0;
    self.scrollView.delegate = self;
    self.scrollView.clipsToBounds = YES;
    [self.scrollView addSubview:self.photo];
}
```



# Mở ví dụ SimpleZoom ra chạy thử

# Để làm việc tốt với UIScrollView cần làm 2 thứ

---

1. Cấu hình UIScrollView

2. Lập trình UIViewController thực hiện các hàm trong UIScrollViewDelegate



# Cấu hình cho UIScrollView

# Managing the Display of Content

- setContentOffset:animated:  
contentOffset *property*  
contentSize *property*  
contentInset *property*

## Managing Scrolling

- scrollEnabled *property*  
directionalLockEnabled *property*  
scrollsToTop *property*
- scrollRectToVisible:animated:  
pagingEnabled *property*  
bounces *property*  
alwaysBounceVertical *property*  
alwaysBounceHorizontal *property*
- touchesShouldBegin:withEvent:inContentView:
- touchesShouldCancelInContentView:  
canCancelContentTouches *property*  
delaysContentTouches *property*  
decelerationRate *property*  
dragging *property*  
tracking *property*  
decelerating *property*



## Managing the Scroll Indicator

- `indicatorStyle` *property*
- `scrollIndicatorInsets` *property*
- `showsHorizontalScrollIndicator` *property*
- `showsVerticalScrollIndicator` *property*
- `flashScrollIndicators`

## Zooming and Panning

- `panGestureRecognizer` *property*
- `pinchGestureRecognizer` *property*
- `zoomToRect:animated:`
  - `zoomScale` *property*
- `setZoomScale:animated:`
  - `maximumZoomScale` *property*
  - `minimumZoomScale` *property*
  - `zoomBouncing` *property*
  - `zooming` *property*
  - `bouncesZoom` *property*

## Managing the Delegate

- `delegate` *property*

# <UIScrollViewDelegate>

## Zooming and Paging



- Để điều khiển, hứng sự kiện của UIScrollView, ta gán UIScrollView.delegate tới UIViewController.
- Còn UIViewController thì tuân thủ UIScrollViewDelegate

## Responding to Scrolling and Dragging

- `scrollViewDidScroll:`
- `scrollViewWillBeginDragging:`
- `scrollViewWillEndDragging:withVelocity:targetContentOffset:`
- `scrollViewDidEndDragging:willDecelerate:`
- `scrollViewShouldScrollToTop:`
- `scrollViewDidScrollToTop:`
- `scrollViewWillBeginDecelerating:`
- `scrollViewDidEndDecelerating:`

## Managing Zooming

- `viewForZoomingInScrollView:`
- `scrollViewWillBeginZooming:withView:`
- `scrollViewDidEndZooming:withView:atScale:`
- `scrollViewDidZoom:`

## Responding to Scrolling Animations

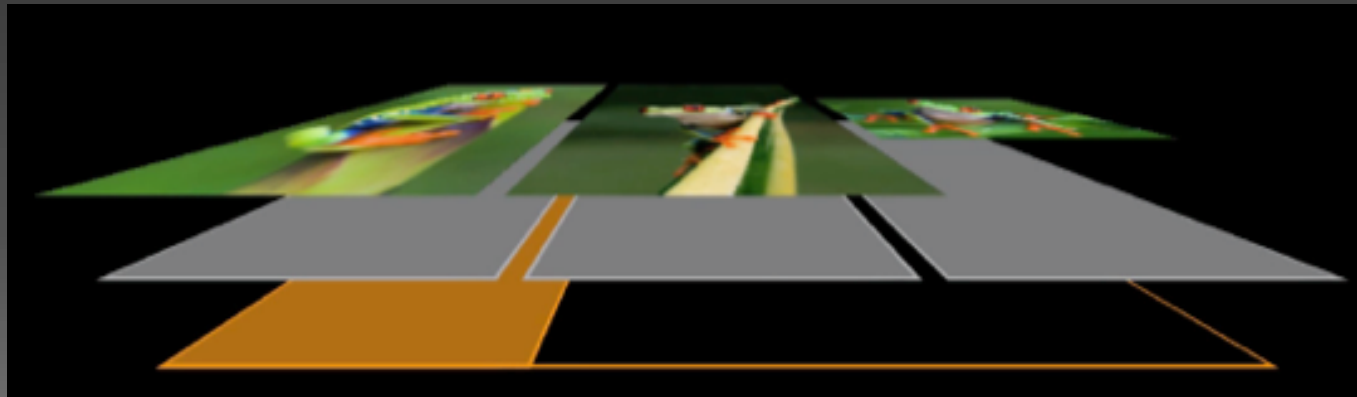
- `scrollViewDidEndScrollingAnimation:`



# Demo

## Ví dụ “Zoom in Paging Scroll View”

- Tạo Paging trong UIScrollView
- Lồng ScrollView trong ScrollView để tạo hiệu ứng mong muốn.





# Thuộc tính pagingEnabled và ContentSize

```
self.myScrollView = [[UIScrollView alloc] initWithFrame:scrollViewRect];  
self.myScrollView.pagingEnabled = YES;  
self.myScrollView.contentSize = CGSizeMake(scrollViewRect.size.width *  
3.0f, scrollViewRect.size.height);
```

# View Configuration

setContentSize

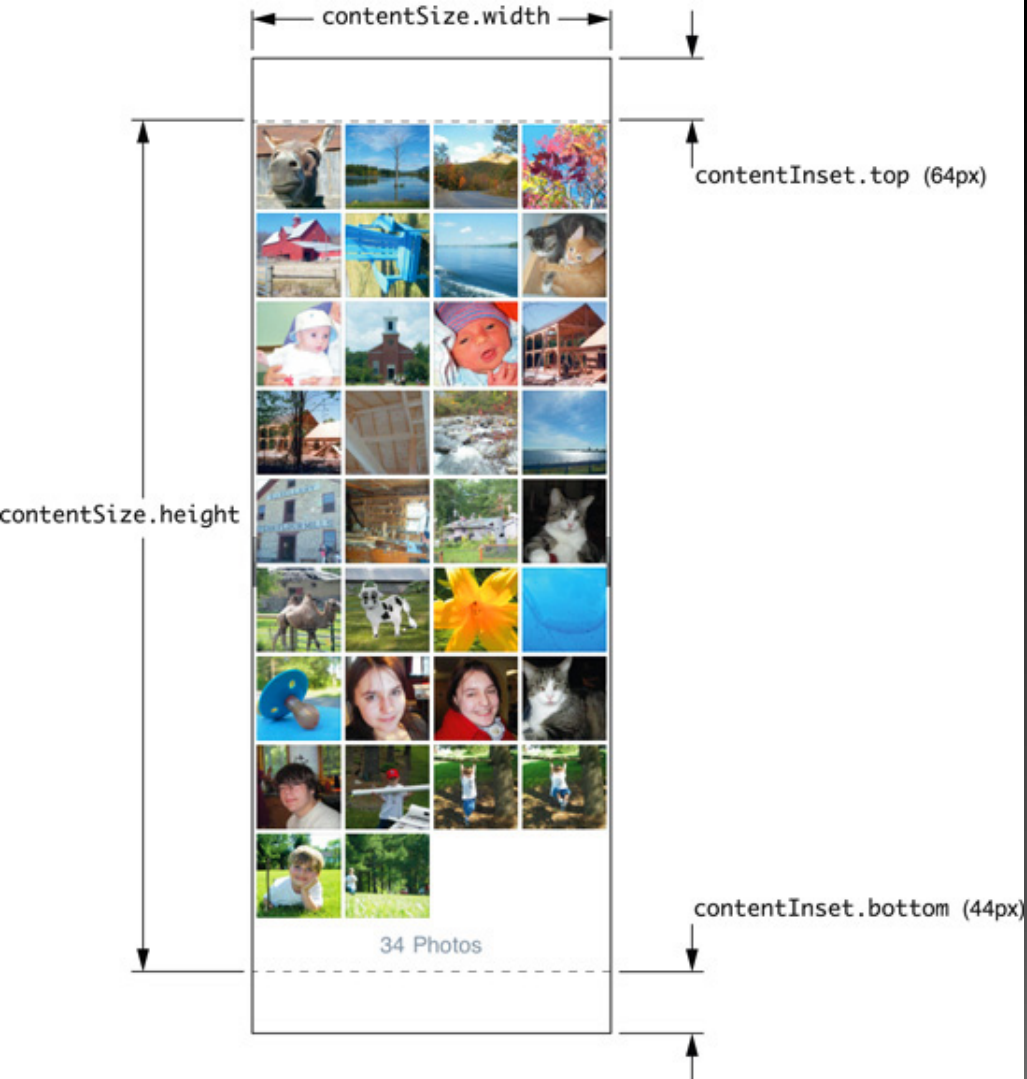




# Coordination of content in UIScrollView

- `contentInset`
- `contentOffset`
- `contentSize`





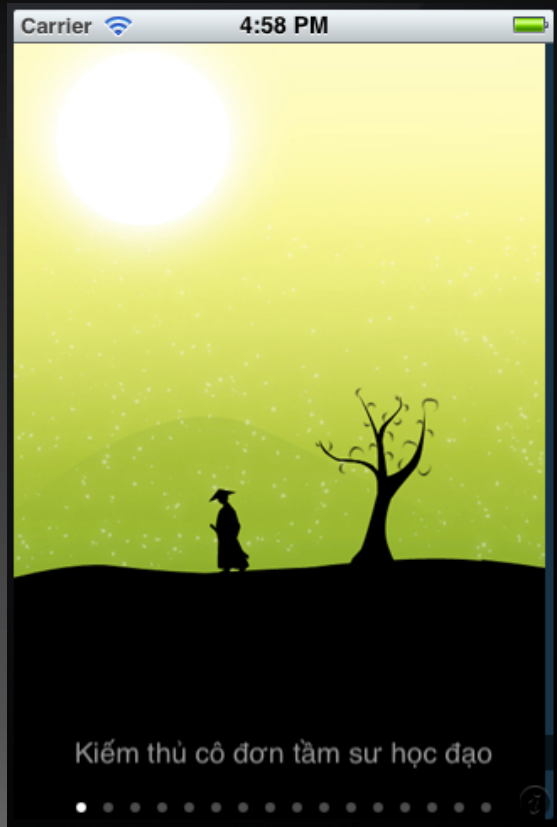
# UIScrollView property explain

---

- Content size: kích thước nội dung hiển thị bên trong scroll view.
- ContentInset: phần thừa ra không hiển thị nội dung scroll view



# Background Chooser



- Model:
  - Photo, DataSource
- ViewController

```
- (void)scrollViewDidScroll:(UIScrollView *)scrollView {  
    // Load the pages which are now on screen  
    [self loadVisiblePages];  
}
```

```
- (void)loadVisiblePages {  
- // First, determine which page is currently visible  
CGFloat pageWidth = self.scrollView.frame.size.width;  
NSInteger page = (NSInteger)floor(self.scrollView.contentOffset.x/  
pageWidth + 0.5f);  
  
    if (self.pageControl.currentPage != page || isFirstTimeLoading)  
{  
        isFirstTimeLoading = NO;  
        // Update the page control  
        self.pageControl.currentPage = page;  
  
        //[self loadPage:page-1];  
        [self loadPage:page];  
        [self loadPage:page+1];  
  
        self.labelPopup.text = [self.datasource  
descriptionAtIndex:page];  
    }  
}
```



# ScrollViewSuite : TapToZoom

- Fitting the image to the screen on launch
- Detecting single, double, and two-finger taps using `UIGestureRecognizer`
- Zooming in response to taps



# ScrollViewSuite : AutoScroll

- Single tap: toggle display thumb view
- Double tap – single touch
- Double tap – two touches:
- Use of the `canCancelContentTouches` property to track moving touches in a subview of a scroll view
- How to implement autoscrolling in response to a subview being dragged





# ScrollViewSuite - Titling

- How to subclass UIScrollView to add content tiling
- Reusing tiles to optimize performance and memory use
- Changing the resolution of the content in response to zooming

# UITextView

---

- Multilines
- No multi colors
- No bold / italic formatting

# 3<sup>rd</sup> parties RichText

- <http://www.cocoacontrols.com/platforms/ios/controls/egotextview>
- <https://github.com/SquaredTiki/EditableCoreTextOverlay>
- <http://codaset.com/jer/jtextview>
- <http://www.cocoacontrols.com/platforms/ios/controls/dtcoretext>
- <http://www.cocoacontrols.com/platforms/ios/controls/rtlabel>

# UIProgressView

---

- See demo “OpenPDF ProgressView”

```
- (void)connection:(NSURLConnection *)connection didReceiveData:
(NSData *)data
{

    filesizereceived += [data length];
    filepercentage = (float)filesizereceived/(float)totalfilesize;
    progressView.progress = filepercentage;

    int p = 100 *filepercentage;
    NSString *percent = [NSString stringWithFormat: @"%d %%",p];

    progressLabel.text = percent;
    [receivedData appendData:data];

}
```