

# UIScrollView新手教程 | AppCoda翻译系列



**作者** 我是乔忘记疯狂 (/u/c9c816980018) [+ 关注](#)

2015.06.29 19:32\* 字数 4114 阅读 5088 评论 2 喜欢 39 阅读 5088 评论 2 喜欢 39 (/u/c9c816980018)

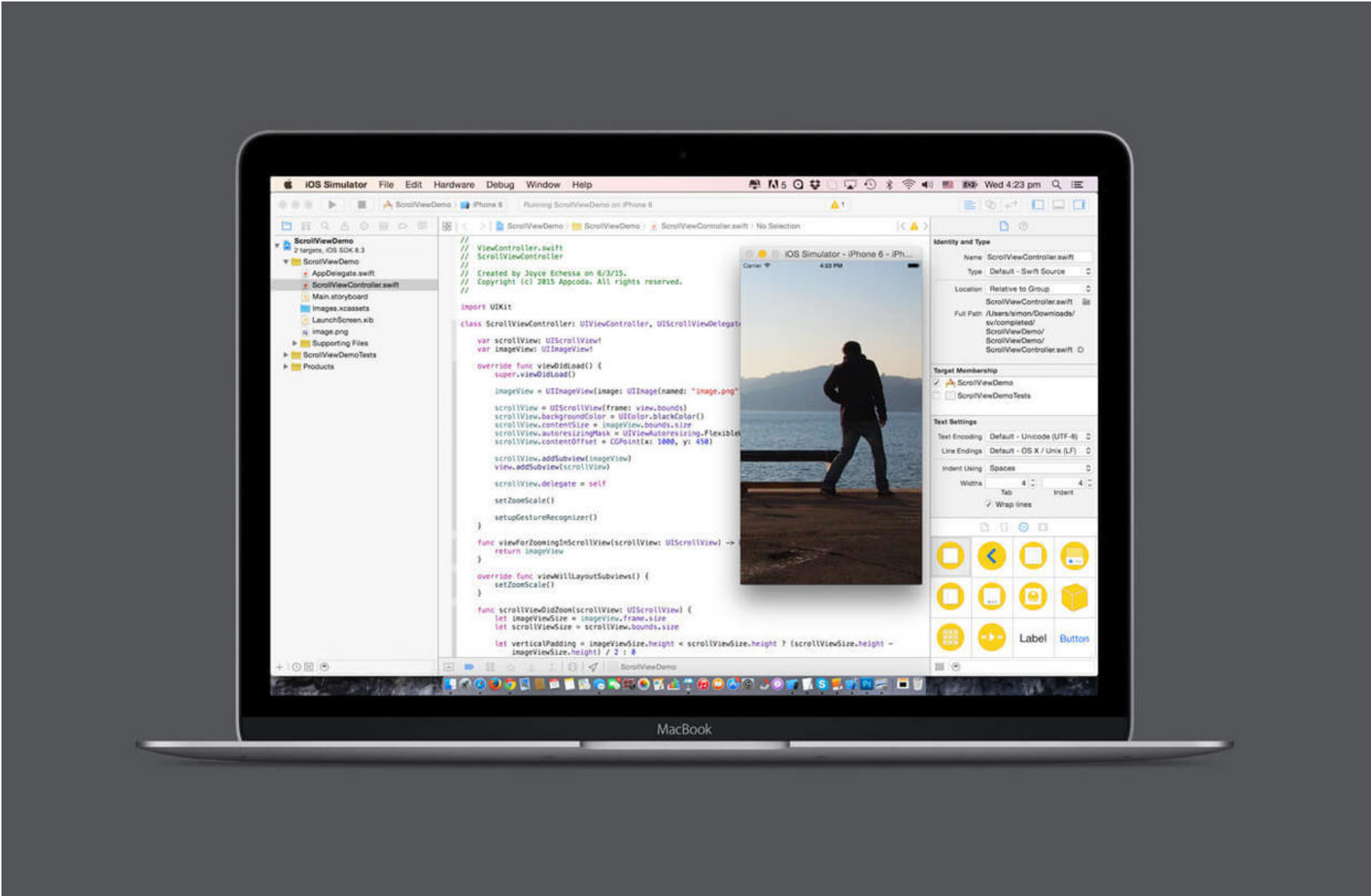
(本文翻译自AppCoda，详见：原文链接：A Beginner’s Guide to UIScrollView (http://www.appcoda.com/uiscrollview-introduction/))

在iOS开发中，滚动视图（UIScrollView）通常用于显示内容尺寸大于屏幕尺寸的视图。滚动视图有以下两个主要作用：

- 让用户可以通过拖拽手势来观看想看到的内容
- 让用户可以通过捏合手势来放大或缩小观看的内容

在iOS应用中常见的表格视图（UITableView）就继承自滚动视图，并因此可以通过上下滚动来显示更多的内容。

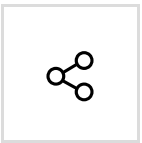
在本篇教程中，我们将讨论滚动视图的诸多方面内容，主要包括：使用纯代码和可视化编程两种方式来创建一个滚动视图、实现滚动和缩放功能，以及如何嵌套使用滚动视图。



继续阅读之前，请先下载本文示例代码所需的资源文件，详见：资源文件地址 (https://www.dropbox.com/s/qhdja7ixcfixu8e/UIScrollViewStarter.zip?dl=0)。（译注：原文资源文件地址需要FQ访问，本人已转存到GitHub上，详见这里 (https://github.com/WXGBridgeQ/UIScrollViewStarter)）

## 使用纯代码方式创建UIScrollView

UIScrollView 同其他视图一样，可以通过纯代码和可视化编程两种方式来创建。在创建之后，只需要少量额外设置就可以让 UIScrollView 获得基本的滚动功能。



UIScrollView 也和其他视图一样，应该被一个控制器管理或者添加到某个视图层级中。想要完成滚动功能还需要对 UIScrollView 进行以下两步设置：

- 必须设置 UIScrollView 的 `contentSize` 属性，它提供了 UIScrollView 的内容的大小，也就是可以滚动的区域的大小。
- 必须为 UIScrollView 添加一个或多个用于显示和滚动的子视图，这些视图提供了 UIScrollView 显示的内容。

你还可以根据应用的具体需求设置 UIScrollView 的一些显示效果，比如：是否显示水平和竖直方向的滚动条、滚动的弹性效果、缩放的弹性效果，以及允许的滚动方向等。

接下来我们将在代码中创建一个 UIScrollView。在下载的资源文件中打开 ScrollViewDemo 工程。它就是一个简单的 Single View Application 工程，只不过将 storyboard 中根控制器的类型绑定为自己新建的叫做 ScrollViewController 的控制器，还在项目中添加了一张我们要用到的图片，图片名称为 `image.png`。

接下来打开 ScrollViewController.swift 文件，添加如下代码。

```
var scrollView: UIScrollView!  
var imageView: UIImageView!
```

按如下代码所示修改 `viewDidLoad()` 方法。

```
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    imageView = UIImageView(image: UIImage(named: "image.png"))  
  
    scrollView = UIScrollView(frame: view.bounds)  
    scrollView.backgroundColor = UIColor.blackColor()  
    scrollView.contentSize = imageView.bounds.size  
    scrollView.autoresizingMask = UIViewAutoresizing.FlexibleWidth | UIViewAutoresizing.FlexibleHeight  
  
    scrollView.addSubview(imageView)  
    view.addSubview(scrollView)  
}
```

上述代码创建了一个 UIScrollView 和 UIImageView，UIImageView 被设置为 UIScrollView 的子视图。`contentSize` 属性控制滚动区域的大小，我们将它设置为跟图片的尺寸一样大（2000×1500）。我们将滚动视图的背景色设置为黑色，这样图片就像在一块黑色幕布上滚动一样。我们将滚动视图的 `autoresizingMask` 属性设置为 `.FlexibleWidth` 和 `.FlexibleHeight`，使它能够在设备旋转之后自动适应新的宽度和高度。运行当前应用，你已经可以通过拖拽手势来滚动显示图片了。



当你启动应用后，你会发现图片初始显示区域是它左上角的部分。





这是因为滚动视图的 `bounds` 的起点默认为(0, 0)，代表了左上角。如果你想改变启动后显示的位置，你需要更改滚动视图的 `bounds` 的起点。因为这种需求经常被提起，所以 `UIScrollView` 专门提供了一个属性 `contentOffset` 用来实现这种需求。

在代码中添加如下语句，注意添加在设置`autoresizingMask`语句之后。

```
scrollView.contentOffset = CGPointMake(x: 1000, y: 450)
```

重新运行应用，你会发现一开始就会显示图片的另一部分而不是左上角。你可以通过这种方式来决定程序启动后将要显示的内容。



## 缩放

我们已经添加了一个 `UIScrollView`，并且能够让用户通过拖拽来观看尺寸大于屏幕尺寸的内容。相当棒，但如果视图能够缩放的话会带来更好的体验。

要支持缩放功能，你必须为 `UIScrollView` 设置一个代理，而且代理必须遵守 `UIScrollViewDelegate` 协议，代理还需要实现 `viewForZoomingInScrollView()` 方法，该方法返回想要被缩放的视图。

你还应该为缩放设置一个比例，可以通过 `UIScrollView` 的 `minimumZoomScale` 和 `maximumZoomScale` 这两个属性来实现，它们的默认值都是1.0。

按照如下代码更改 `ScrollViewController` 的定义：

```
class ScrollViewController: UIViewController, UIScrollViewDelegate {
```

然后添加如下代码：

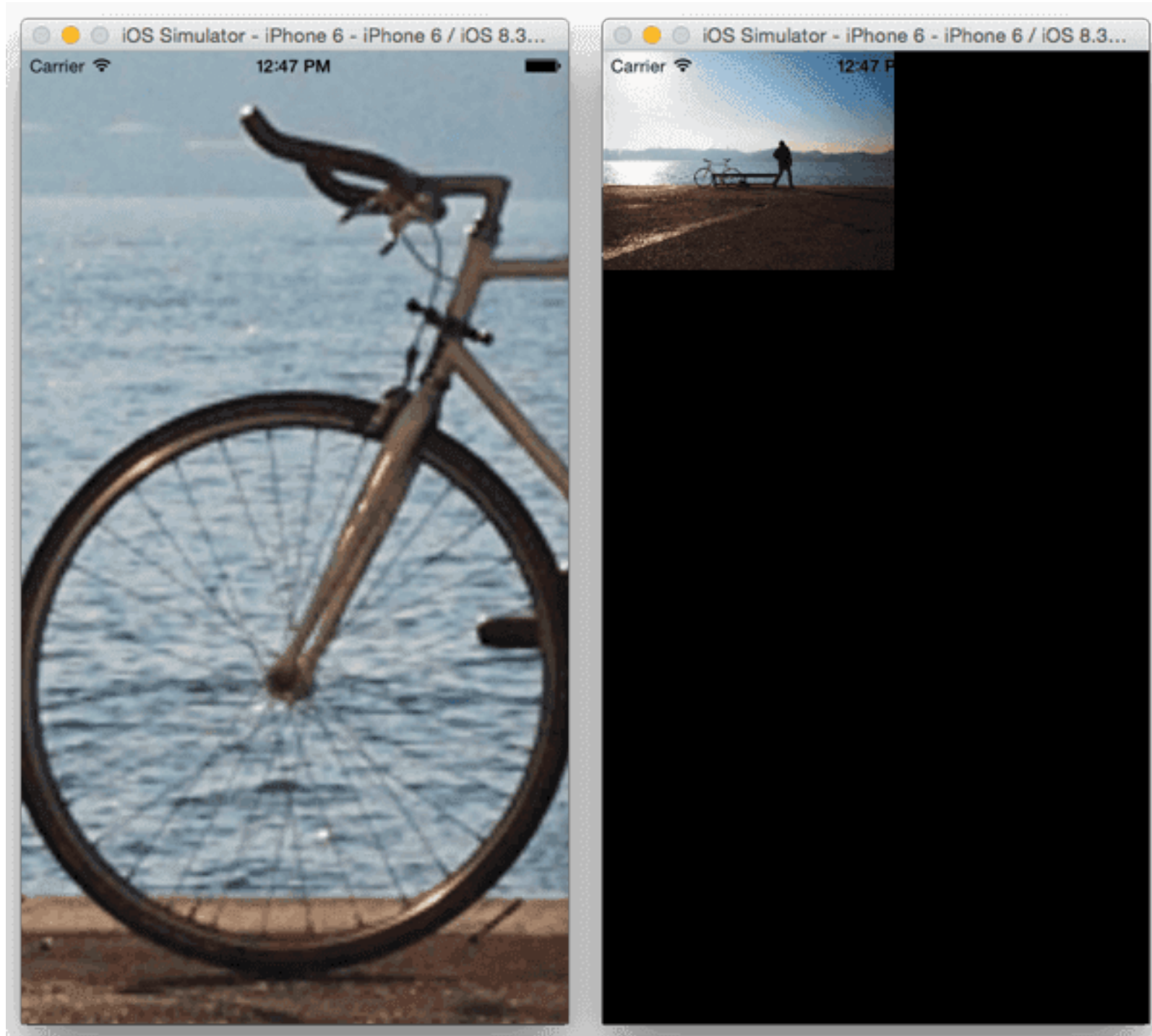
```
func viewForZoomingInScrollView(scrollView: UIScrollView) -> UIView? {  
    return imageView  
}
```

接下来在 `viewDidLoad()` 方法的最后添加如下代码：

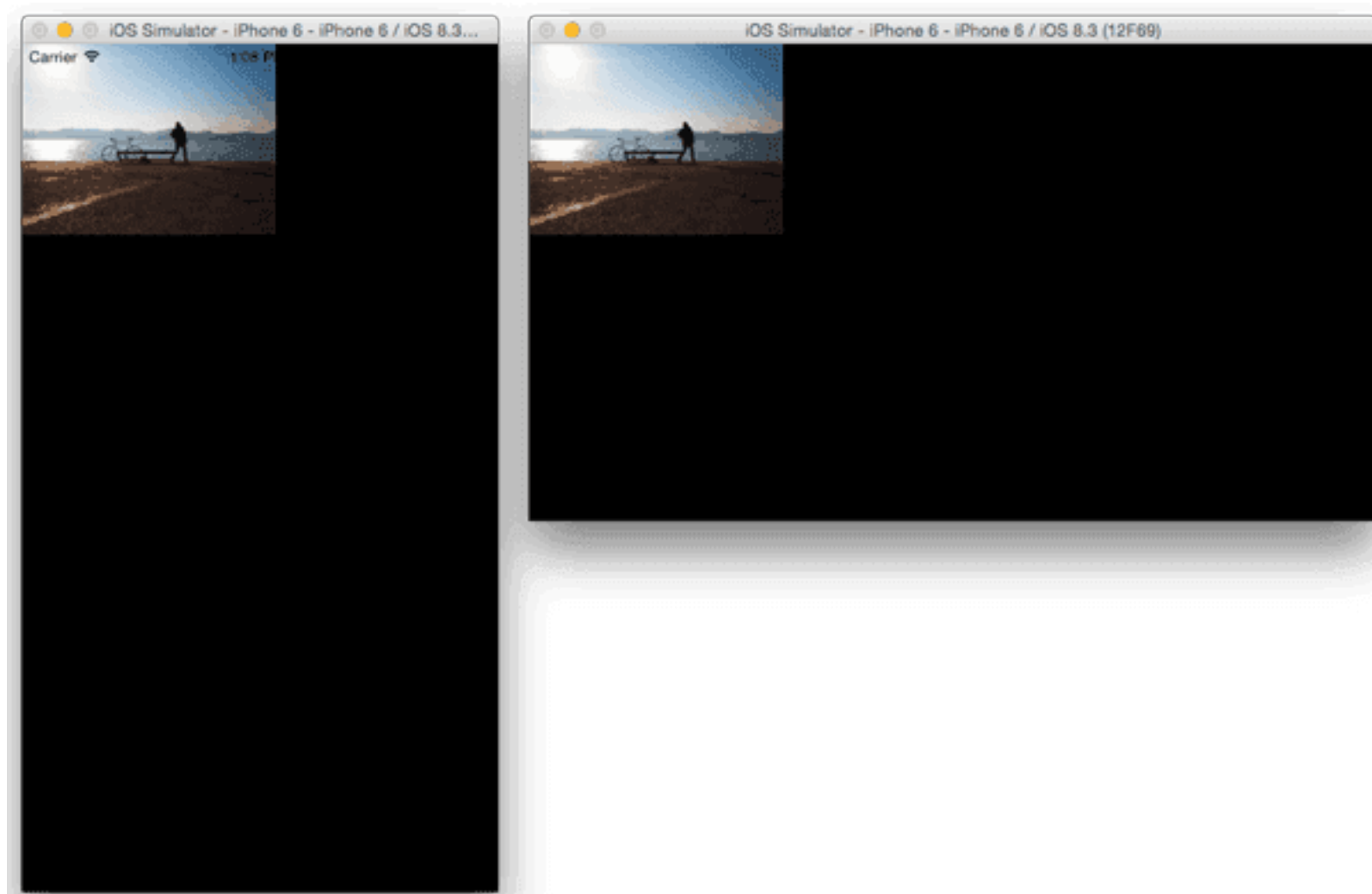


```
scrollView.delegate = self
scrollView.minimumZoomScale = 0.1
scrollView.maximumZoomScale = 4.0
scrollView.zoomScale = 1.0
```

在上述代码中，我们设置了 zoomScale 为1.0，然后设置了缩放的最大和最小比例。在程序运行后，会按照图片的原始尺寸显示（因为 zomScale 为1.0），当你使用捏合手势来操作图片时，你会发现图片可以被缩放了。我们设置了 maximumZoomScale 为4.0，所以图片最大只能放大到4倍。你也会发现，图片放大4倍后会变得很模糊，所以接下来我们会把它的缩放比例重新设置为1.0。



从上面的图片中我们可以发现，我们之前将 minimumZoomScale 设置为0.1实在是太小了，屏幕空出了很多空闲的地方。在横屏模式下，空闲的区域看上去更大。我们希望图片能在某一方向上能与屏幕相匹配，让图片既能完全显示，又能尽量减少屏幕的空闲空间。



要达到这样的效果，你必须通过图片尺寸和 UIScrollView 的尺寸来计算最小的缩放比例。

首先在 viewDidLoad() 方法中删除以下三行代码：

```
scrollView.minimumZoomScale = 0.1
scrollView.maximumZoomScale = 4.0
scrollView.zoomScale = 1.0
```

在控制器类中添加如下方法。在方法中，我们算出图片同 UIScrollView 的高度和宽度的比值，并将最小缩放比例设置为两者中更小的那个。注意，我们已经删除了 maximumZoomScale 的设置，所以它的默认值为1.0。

```
func setZoomScale() {
    let imageViewSize = imageView.bounds.size
    let scrollViewSize = scrollView.bounds.size
    let widthScale = scrollViewSize.width / imageViewSize.width
    let heightScale = scrollViewSize.height / imageViewSize.height

    scrollView.minimumZoomScale = min(widthScale, heightScale)
    scrollView.zoomScale = 1.0
}
```

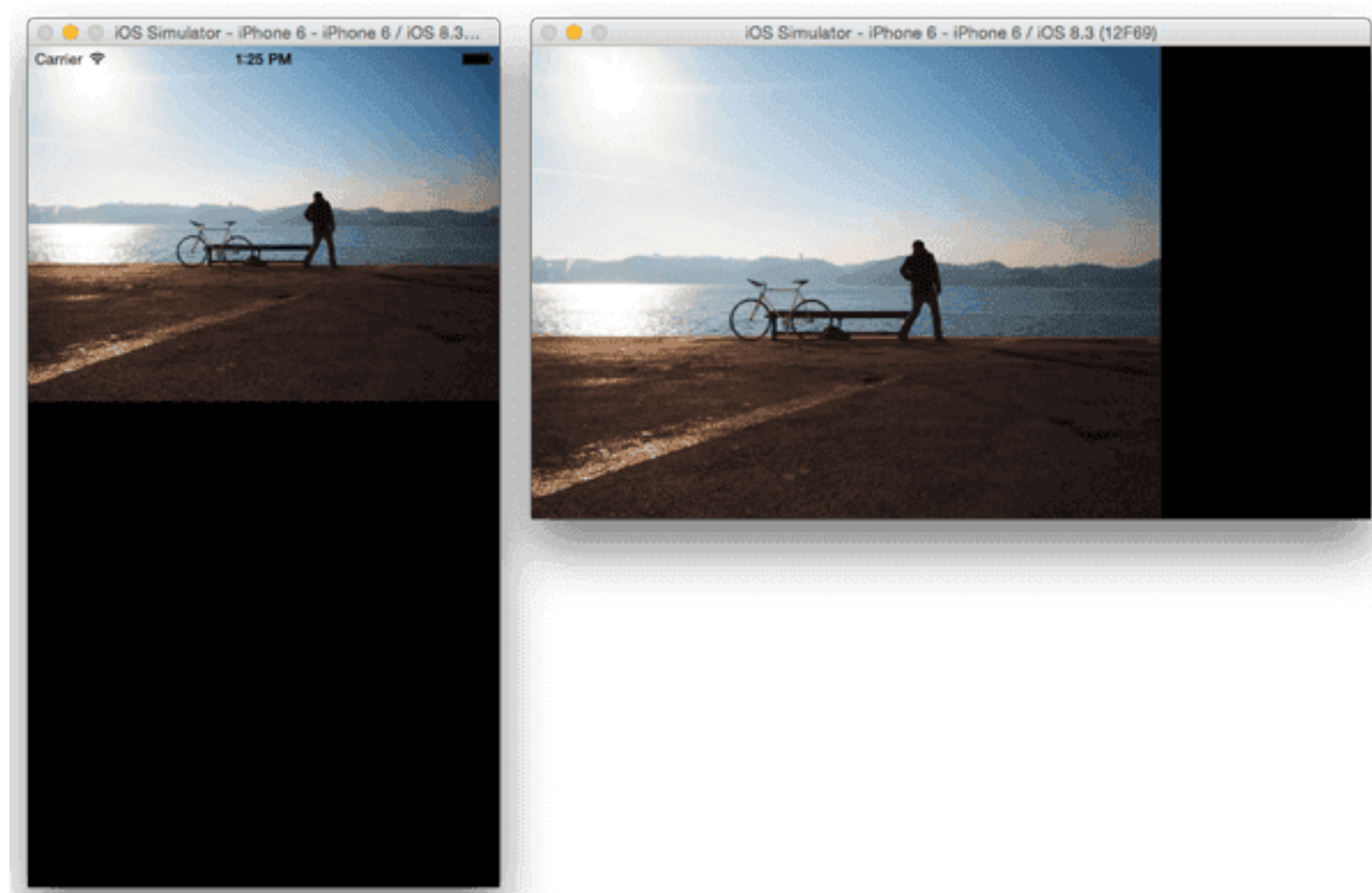
在 viewDidLoad() 方法最后调用这个方法：

```
setZoomScale()
```

在 viewWillLayoutSubviews() 方法中也需要调用该方法，这样当用户改变屏幕方向后，图片的尺寸仍然是正确的。

```
override func viewWillLayoutSubviews() {
    setZoomScale()
}
```

运行程序，现在你会发现无论你缩放到多小，图片都会完整显示并且尽量占满剩余的空间。



我们可以发现，图片是被定位在屏幕左上角的，我们希望将它放在屏幕中间。

在代码中添加如下方法。



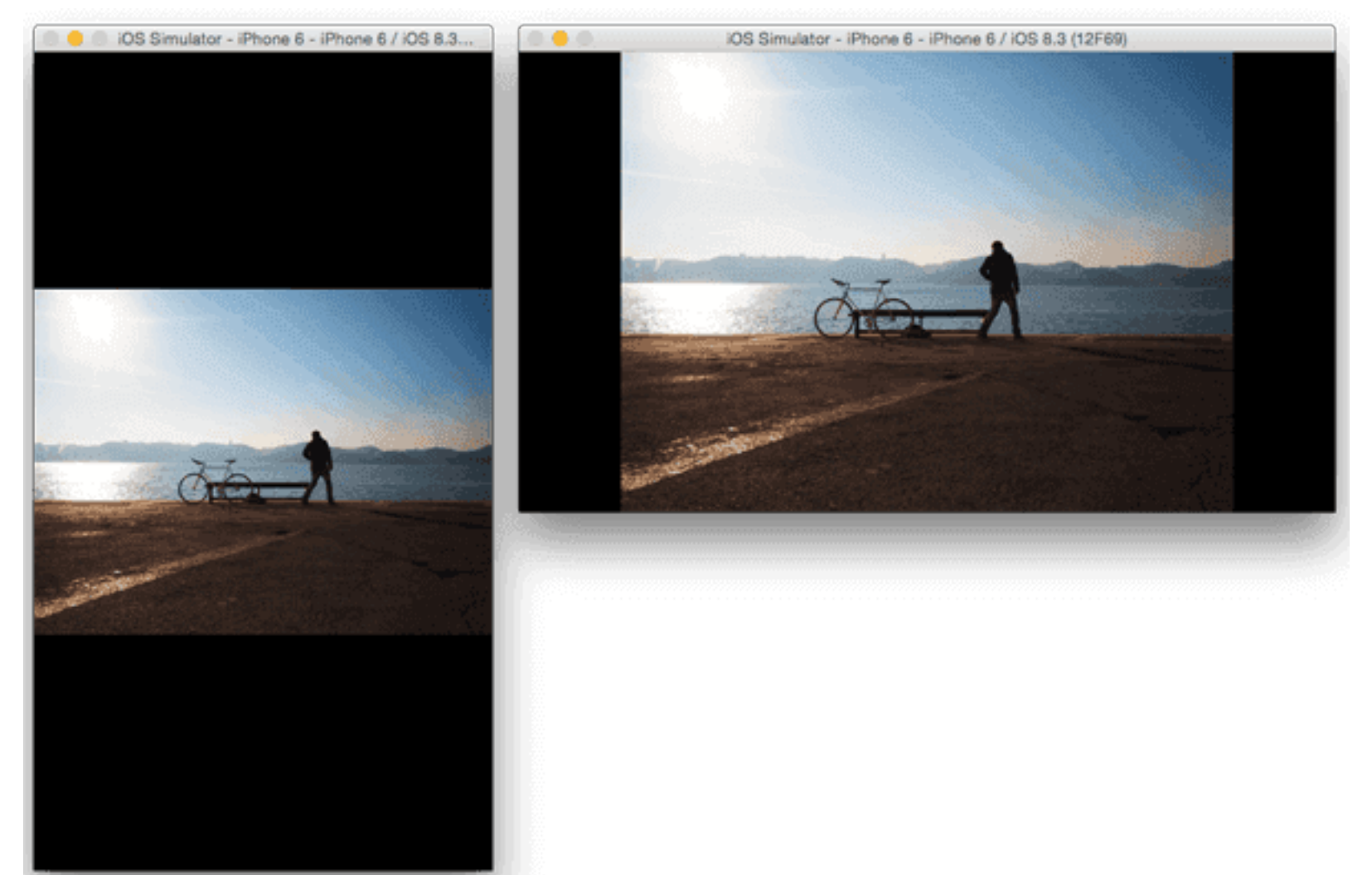
```
func scrollViewDidZoom(scrollView: UIScrollView) {
    let imageViewSize = imageView.frame.size
    let scrollViewSize = scrollView.bounds.size

    let verticalPadding = imageViewSize.height < scrollViewSize.height ? (scrollViewSize.height - imageViewSize.height) / 2 : 0
    let horizontalPadding = imageViewSize.width < scrollViewSize.width ? (scrollViewSize.width - imageViewSize.width) / 2 : 0

    scrollView.contentInset = UIEdgeInsets(top: verticalPadding, left: horizontalPadding, bottom: verticalPadding, right: horizontalPadding)
}
```

这个方法在缩放的时候就会被调用，它会通知代理 `UIScrollView` 的缩放比例发生改变。在上面的方法中，我们计算了图片在滚动视图中的内间距，从而使图片始终在屏幕的中间。对于上、下方向的内边距，我们首先判断图片视图的高度是否小于滚动视图的高度，如果是就将边距设为两者的差值的一半，否则设为0。水平间距我们采用同样的方式计算。然后通过 `contentInset` 属性设置所有方向的内边距，这个属性代表了 `UIScrollView` 的内容距离 `UIScrollView` 本身四周的距离。

运行程序，你会发现当你缩小图片时，图片始终保持在屏幕的中间。



## 通过双击来缩放

`UIScrollView` 默认只支持通过捏合手势来实现缩放效果，如果想实现通过双击来缩放，则需要自己做些额外的设置。

`iOS人机界面指南` 中介绍了可以通过双击手势来达到缩放的效果。使用双击手势进行缩放需要一定的前提：要缩放的视图只能在最大和最小比例两个固定值之间来回缩放，就像苹果官方的相册应用一样，当你双击图片时，图片放大至最大，当你再次双击时，图片缩小至最小，或者可以通过连续的双击使视图一点点达到最大，然后再次双击的时候，将视图恢复为全屏显示。但是大多数应用需要实现更灵活的双击缩放效果，例如地图应用，当你双击时会使其放大，继续双击会继续放大，想要缩小则可以使用双指捏合手势来实现。

要想在你的程序中实现双击缩放功能，你需要监听 `UIScrollView` 的手势并进行处理。在我们的程序中，我们将模仿苹果官方的相册应用的效果，当你双击时放大到最大值，再次双击时则缩小到最小值。

在代码中添加如下两个方法。



```
func setGestureRecognizer() {
    let doubleTap = UITapGestureRecognizer(target: self, action: "handleDoubleTap:")
    doubleTap.numberOfTapsRequired = 2
    scrollView.addGestureRecognizer(doubleTap)
}

func handleDoubleTap(recognizer: UITapGestureRecognizer) {
    if (scrollView.zoomScale > scrollView.minimumZoomScale) {
        scrollView.setZoomScale(scrollView.minimumZoomScale, animated: true)
    } else {
        scrollView.setZoomScale(scrollView.maximumZoomScale, animated: true)
    }
}
```

然后在 `viewDidLoad()` 方法最后调用上面的方法。

```
setGestureRecognizer()
```

在上面的代码中，我们为 `UIScrollView` 添加了一个双击手势的监听，然后根据图片当前的缩放比例，来判断是将图片放大或者缩小。

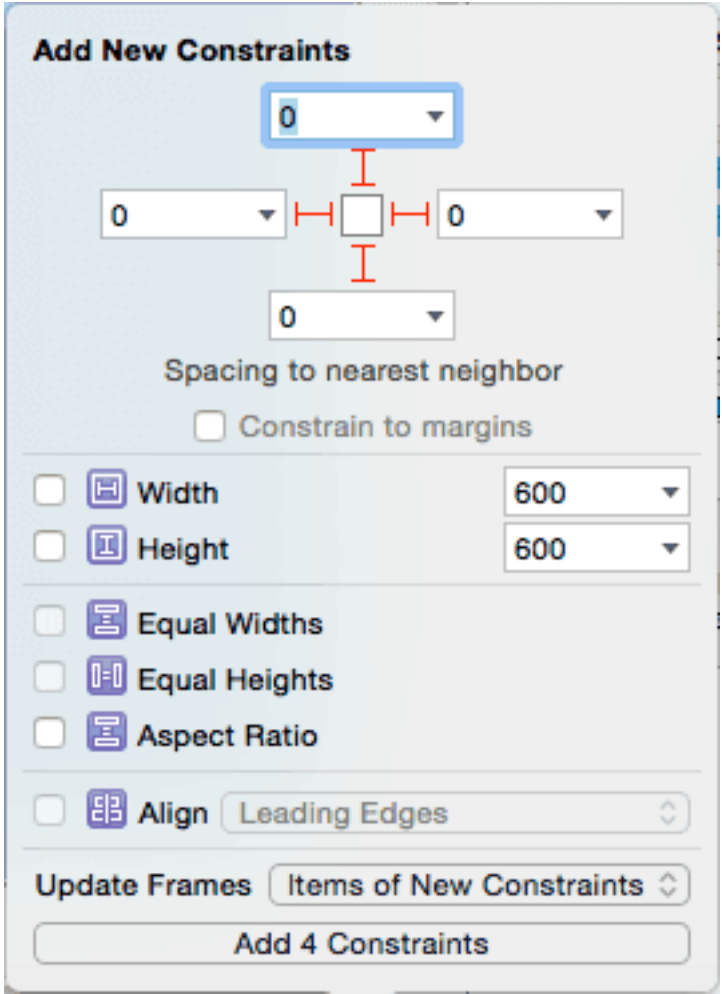
运行程序，你会发现已经能通过双击手势来缩放图片了。

## 用可视化编程方式创建UIScrollView

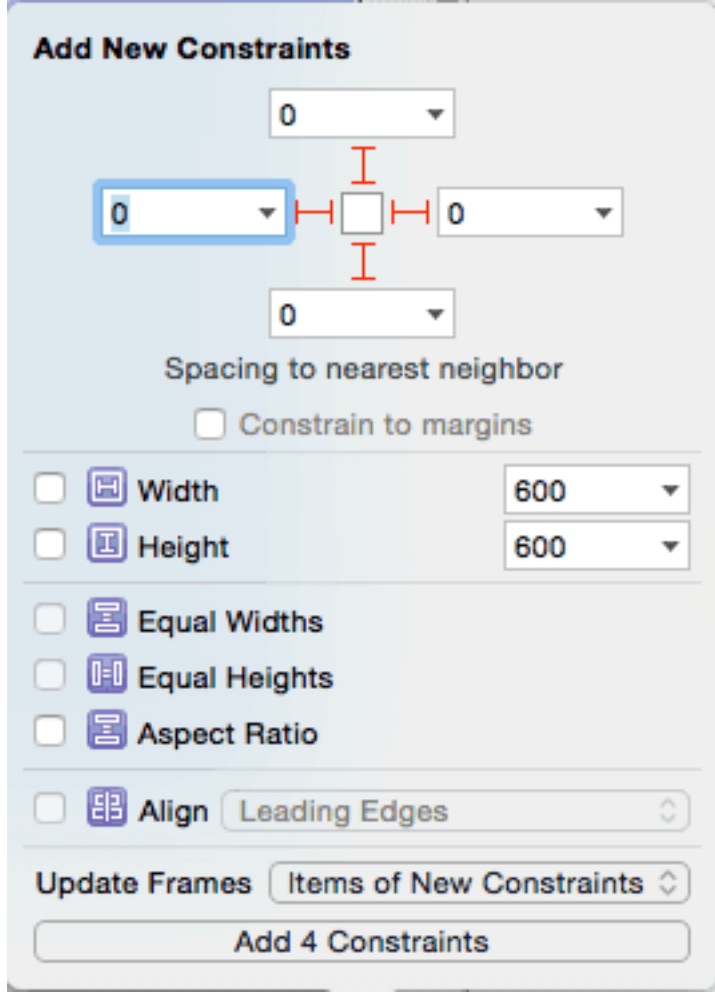
使用 `storyboard` 可以实现和我们上面使用代码方式实现的同样的功能，而且更为简单，代码量更少。

在 `Main.storyboard` 文件中，拖一个新的视图控制器，并将其设置为初始控制器（既可以将箭头拖到新控制器上，也可以在属性选项卡中选中 `Is Initial View Controller` 复选框）。

拖一个 `UIScrollView` 到新的控制器中，然后设置其边缘始终粘着屏幕。



然后拖一个 `UIImageView` 到刚才的 `UIScrollView` 中，将它的边缘设置为粘着 `UIScrollView` 。



要记住 UIScrollView 需要知道它的内容的大小，才可以实现滚动。当你为 UIImageView 设置图片时， UIScrollView 的内容大小就会被自动设置为图片的大小。

在 UIImageView 的属性选项卡中，将 Image 属性设置为 image.png，然后通过 updating the frames 解决自动布局的问题。运行程序，你会发现已经实现图片的滚动显示了，并且没有敲一行代码。你还可以在 UIScrollView 的属性选项卡中查看还有哪些属性可以设置，比如可以设置最大和最小的缩放比例。

如果想要实现缩放功能，你仍然需要通过代码，设置代理并实现 viewForZoomingInScrollView() 方法，同我们之前做过的一样，就不再重复一遍了。

## UIScrollView的嵌套使用

可以在一个 UIScrollView 中嵌套另一个 UIScrollView，两个 UIScrollView 既可以是相同方向滚动的，也可以是不同方向的。这部分内容的示例代码请使用 NestedScrollViews 项目。

### 相同方向的UIScrollView嵌套

相同方向的 UIScrollView 嵌套是指一个 UIScrollView，它有另一个 UIScrollView 作为子控件，并且它们的滚动方向一致。你可以用相同方向的嵌套来实现这样的效果，比如在 UIScrollView 中添加多组要区分开的数据，你还可以通过它来实现两个 UIScrollView 同时滚动时的视差效果。在我们的示例中，我们将两个相同方向的 UIScrollView 设置不同的滚动速度，从而实现滚动时的视差效果。

打开 NestedScrollViews 项目中的 storyboard 文件，你将看到两个 UIScrollView，分别叫做 foreground 和 background。background 里面添加了一个 UIImageView，并将图片设置为 image.png，foreground 里面添加了一些标签和一个作为容器用的 UIView，这些标签只是为了方便我们观看视图的滚动，容器视图我们将在下一节内容中才用到。

我们的界面这样就算搭建完成了，现在运行程序的话，你会发现只有 foreground 视图在滚动，而 background 视图保持不动。接下来我们将要实现 background 的滚动，并且实现滚动的视差效果。

首先将 foreground 和 background 两个 UIScrollView 连线到控制器，之后代码会如下所示：

```
@IBOutlet weak var background: UIScrollView!  
@IBOutlet weak var foreground: UIScrollView!
```

我们需要知道 foreground 视图滚动了多长的距离，用来计算 background 视图需要滚动多长的距离。所以我们需要为 foreground 视图设置一个代理，用来监听它的滚动。



```
class ViewController: UIViewController, UIScrollViewDelegate {
```

在 `viewDidLoad()` 方法中设置foreground视图的代理。

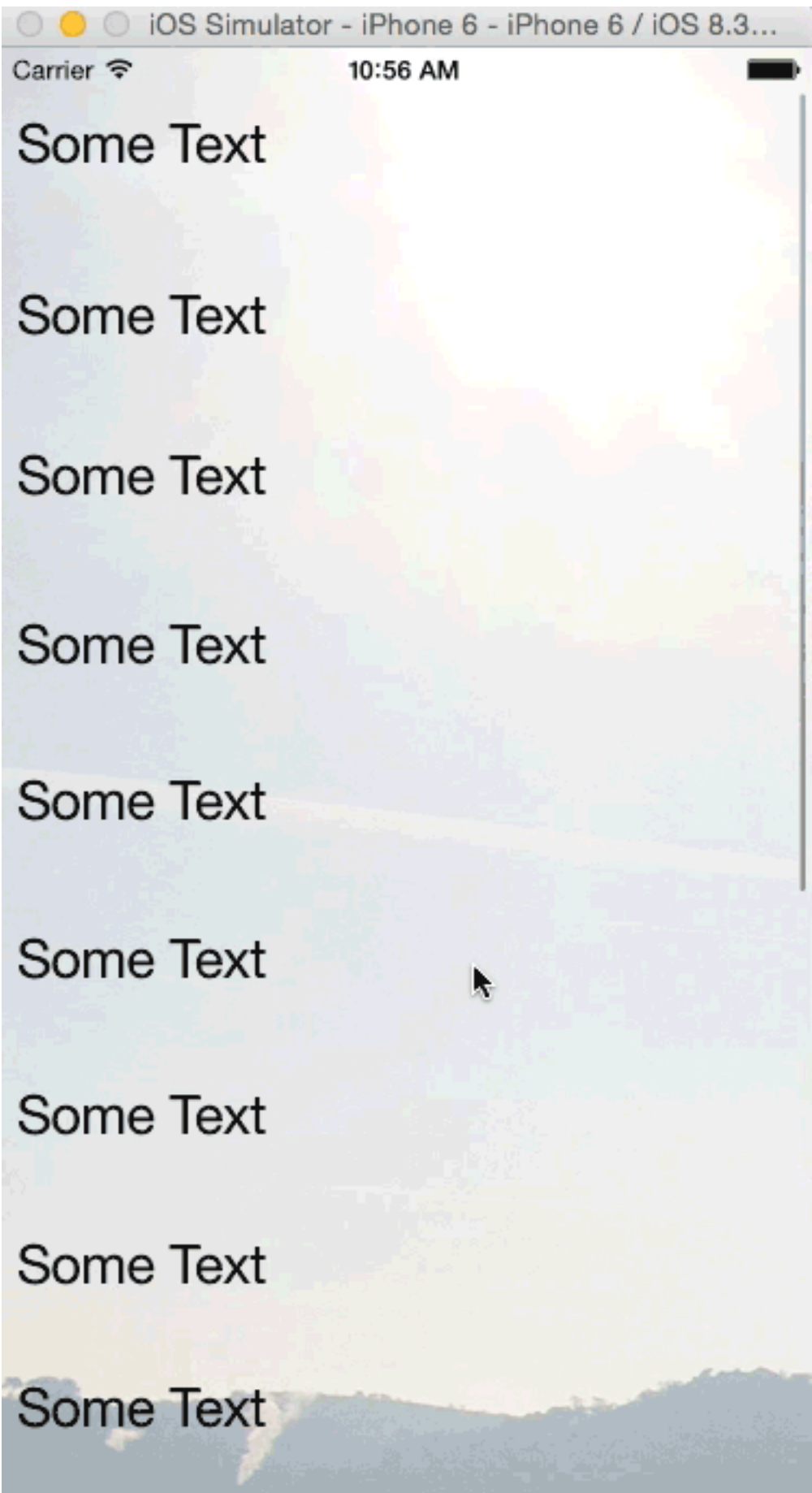
```
foreground.delegate = self
```

然后实现如下代理方法。

```
func scrollViewDidScroll(scrollView: UIScrollView) {
    let foregroundHeight = foreground.contentSize.height - CGRectGetHeight(foreground.bounds)
    let percentageScroll = foreground.contentOffset.y / foregroundHeight
    let backgroundHeight = background.contentSize.height - CGRectGetHeight(background.bounds)

    background.contentOffset = CGPoint(x: 0, y: backgroundHeight * percentageScroll)
}
```

在上面的代码中，我们获取了foreground视图可以滚动的最大高度，然后用当前滚动的距离除以它以获取滚动的比例，然后获取background视图可以滚动的最大高度，将其乘以滚动比例，就可以得到background应该滚动的距离。运行你的程序，在进行滚动时你会发现foreground和background两个视图都在滚动，并且background视图滚动的更快，从而有一种视差效果。



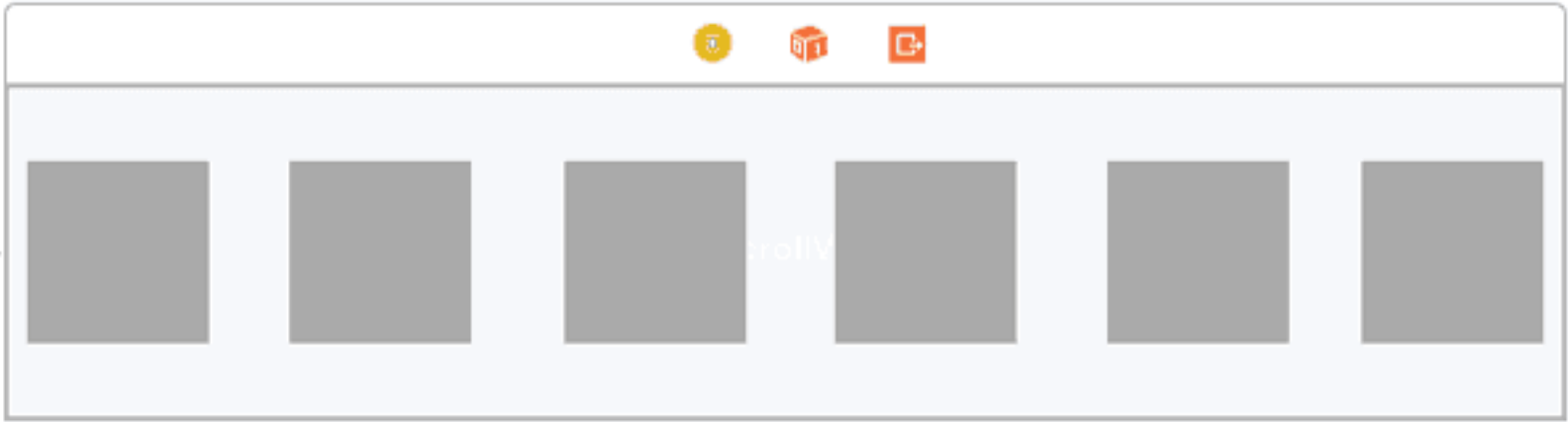
## 交叉方向的UIScrollView嵌套

交叉方向的 UIScrollView 嵌套是指一个 UIScrollView，它有另一个 UIScrollView 作为子控件，并且它们的滚动方向正好相差90°，接下来我们就演示一下这种情况。

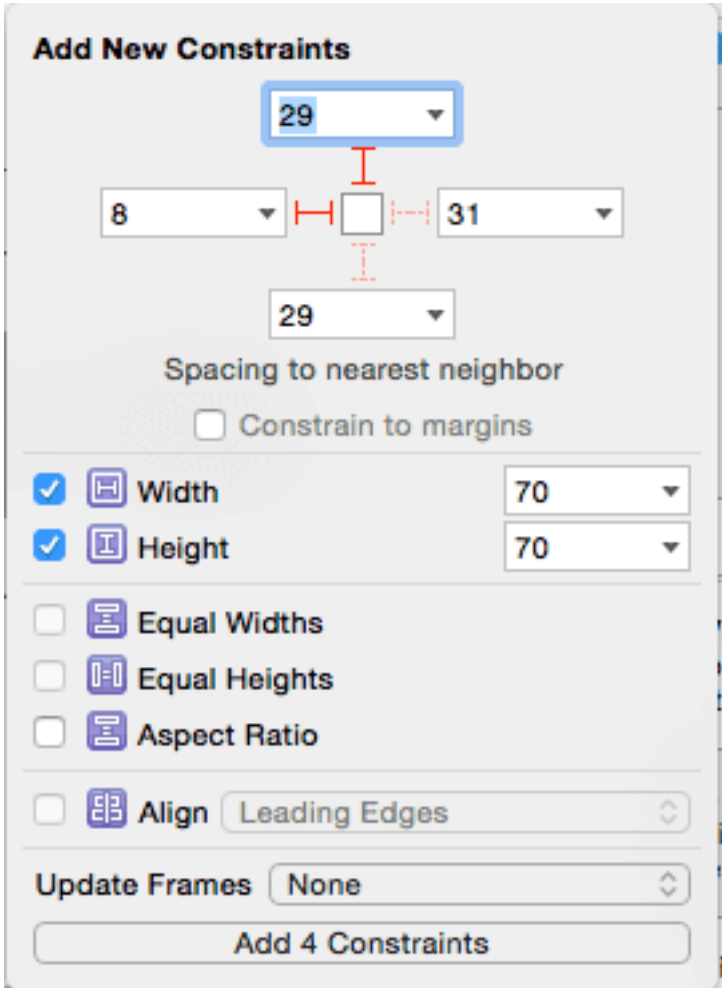
在 NestedScrollViews 项目中，你会发现在foreground里面有一个 Container View，我们将用它来设置我们水平滚动的 UIScrollView。

在 storyboard 中新拖入一个控制器，按住Control键从 Container View 拖到新的控制器，选择`embed`方式。然后选中这个控制器，将它的Size选项设为`Freeform`并将其高度设为128，因为 Container View 的高度就是128。

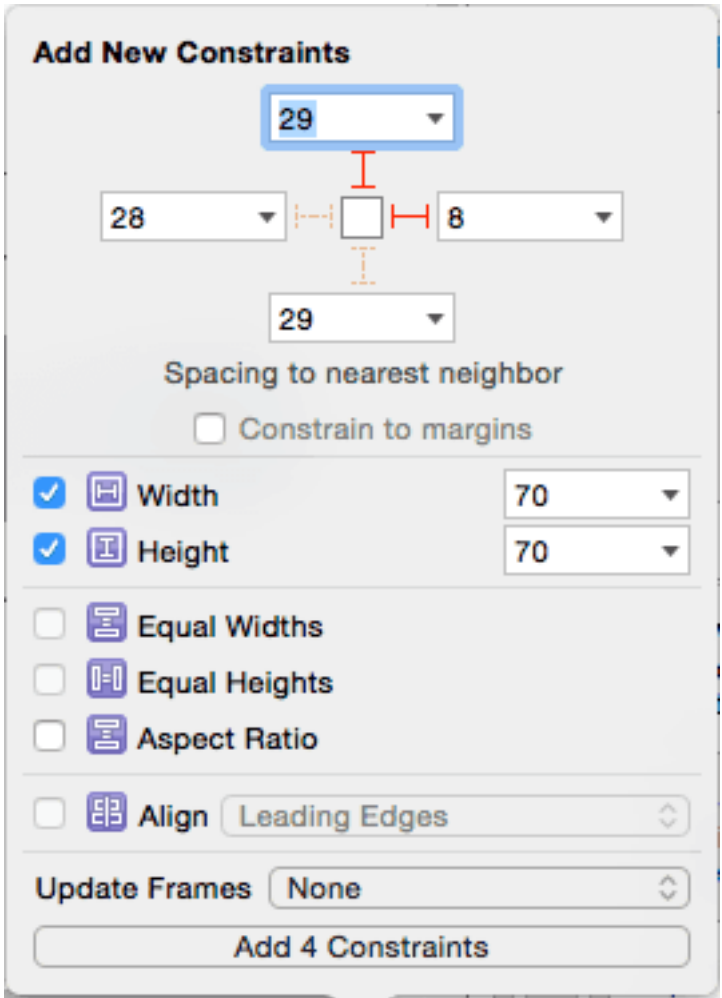
往新控制器中拖入一个 UIScrollView，设置其边缘始终粘着父控件。然后在 UIScrollView 中拖入一个70×70的 UIView，将其背景色设为灰色方便我们观看，然后复制多个，从左到右依次摆放在 UIScrollView 中。你不需要精确地去设置每一个 UIView 的位置，接下来我会教你们怎么去做。现在我们的控制器界面应该是这个样子。



选择最左边的 UIView，添加它上边和左边的约束，再添加宽度和高度约束。

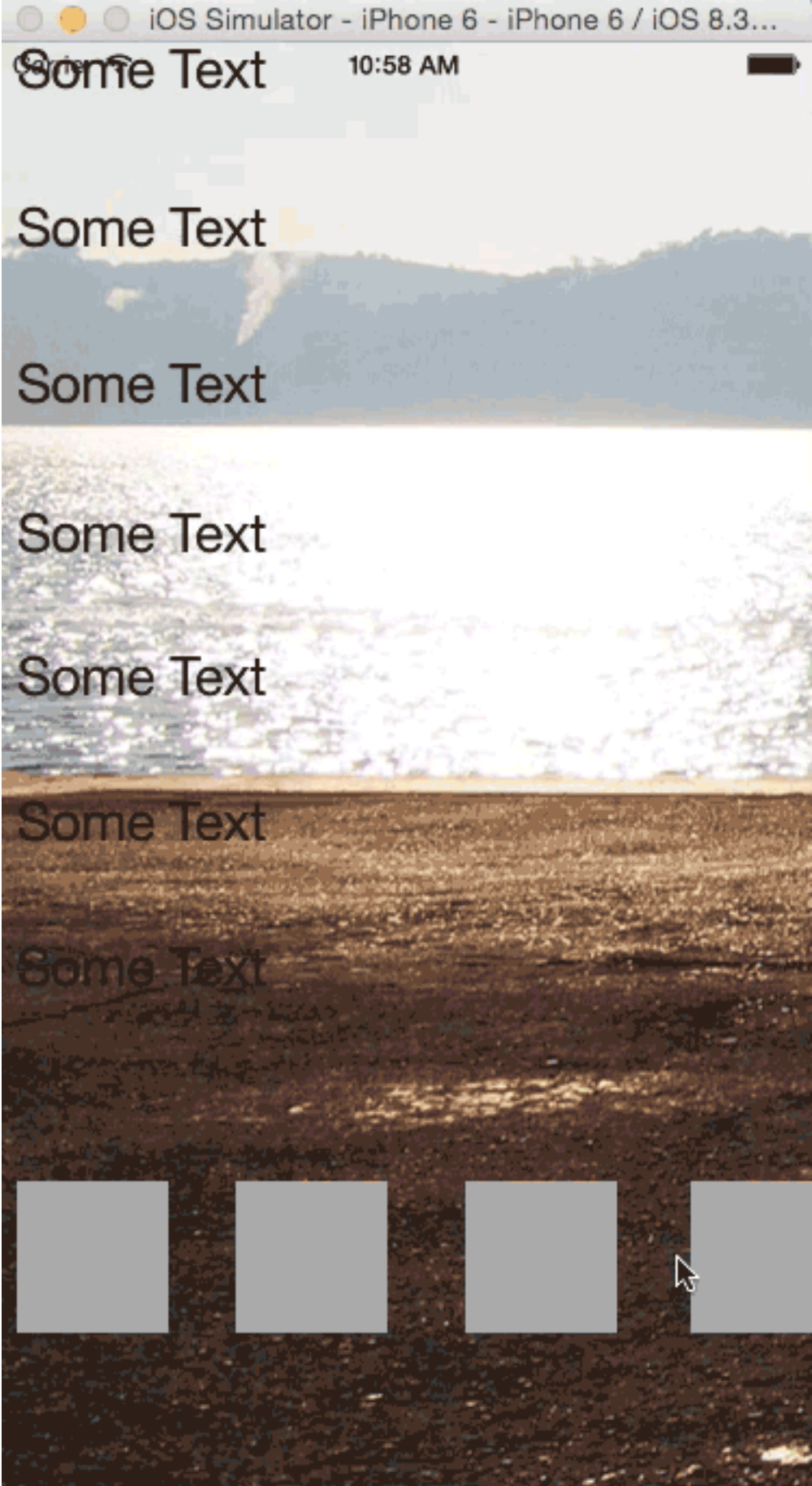


再选择最右边的 UIView，添加它的上边、右边、宽度和高度约束。



接下来，选中我们的 UIScrollView，然后点击上面菜单栏中的 Editor > Resolve Auto Layout Issues > All Views > Add Missing Constraints。这样我们所有的 UIView 就都添加好了约束。运行你的程序，竖直滚动到底部，你会看见我们的 Container View，你可以水平滚动它里面的内容。下图中，我将控制器自身视图的背景色设置为透明，所以你看到的效果就是这样的。





我们的教程这就结束了，并没有包含 UIScrollView 所有的方方面面，但我希望通过这篇教程可以让你对 UIScrollView 有初步的了解，更多 UIScrollView 的知识，你可以查看苹果的官方文档：Scroll View Programming Guide ([https://developer.apple.com/library/prerelease/ios/documentation/WindowsViews/Conceptual/UIScrollView\\_pg/Introduction/Introduction.html#//apple\\_ref/doc/uid/TP40008179-CH1-SW1](https://developer.apple.com/library/prerelease/ios/documentation/WindowsViews/Conceptual/UIScrollView_pg/Introduction/Introduction.html#//apple_ref/doc/uid/TP40008179-CH1-SW1))。

你可以在这里 (<https://www.dropbox.com/s/91buj9d9gmphoen/UIScrollViewFinal.zip?dl=0>)下载完整的示例程序作为学习参考。（译注：原文资源文件地址需要FQ访问，本人已转存到GitHub上，详见这里 (<https://github.com/WXGBridgeQ/UIScrollViewFinal>)）

iOS开发入门 (/nb/883094)

举报文章

© 著作权归作者所有



我是乔忘记疯狂 (/u/c9c816980018) ♂

写了 24653 字，被 563 人关注，获得了 244 个喜欢  
(/u/c9c816980018)写了 24653 字，被 563 人关注，获得了 244 个喜欢

+ 关注

85后程序员，SwiftGG 翻译组成员。

如果觉得我的文章对您有用，请随意赞赏。您的支持将鼓励我继续创作！

赞赏支持



后发表评论 (/sign-in?utm\_source=desktop&utm\_medium=not-signed-in-comment-form)

2条评论

只看作者

按喜欢排序 按时间正序 按时间倒序



布袋的世界 (/u/d2587156c4c8)

2楼 · 2017.01.16 12:43

(/u/d2587156c4c8)

厉害了! thanks for translating this article ...



赞



回复



布袋的世界 (/u/d2587156c4c8)

3楼 · 2017.01.16 19:02

(/u/d2587156c4c8)

在NestedScrollViews项目中，你会发现现在foreground里面有一个Container View，我们将用它来设置我们水平滚动的UIScrollView。

水平的时候没有办法Scroll，请问要怎么办？！



赞



回复

被以下专题收入，发现更多相似内容



iOS Dev... (/c/3233d1a249ca?utm\_source=desktop&utm\_medium=notes-included-collection)



iOS Dev... (/c/ee25d429d275?utm\_source=desktop&utm\_medium=notes-included-collection)



iOS移动开发 (/c/db63a204c1a3?utm\_source=desktop&utm\_medium=notes-included-collection)



快速入门iOS (/c/11adc632c888?utm\_source=desktop&utm\_medium=notes-included-collection)



寒哥管理的技术专题 (/c/5be41e88940c?utm\_source=desktop&utm\_medium=notes-included-collection)



编程 (/c/2736ca570b23?utm\_source=desktop&utm\_medium=notes-included-collection)



iOS干货 (/c/cb5e312aaa16?utm\_source=desktop&utm\_medium=notes-included-collection)

展开更多

推荐阅读

更多精彩内容 > (/)

Effective Objective-C 读书笔记(二) | 初步理解OC运...

(/p/3cdd05a9427b?

Class 类型对象 OC本身是一种强类型语言，但其运行时功能让它又有了动态语言的特点。OC中对象的类型和对象所执行的方法都是在运行时阶段进行查找并

utm\_campaign=maleskine&utm\_content=note&utm\_

我是乔忘记疯狂 (/u/c9c816980018?

utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

苹果官方类库中NSDateFormatter的小bug (/p/b16113f02bb0?utm\_cam...

不知道有多少人和我一样是今天才知道这个问题的

letformatter=NSDateFormatter()formatter.dateFormat="yyyy-MM-



我是乔忘记疯狂 (/u/c9c816980018?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

打过几炮又如何 (/p/2db9f35cd213?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

此文纯属虚构，不喜勿喷 这是我休假的第一天，也是我决意要和男友苏城分手的第一天。起因是：他偷偷背着我炒股，并赔光了我们共同的二十万元钱。正

我爱你三小姐 (/u/f1b9d7095c46?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

如何做到高度自律？ (/p/6d4e2d14234b?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

一 我们身边有很多这样的人。他们一边抱怨命运不公，自己运气太差，一边刷微博，玩游戏。他们一边愤愤不平，一边荒废光阴。他们讨厌自己现在的工作

安梳颜 (/u/d90828191ace?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

我出身寒门，想说说寒门贵子的路 (/p/f37393d6ce53?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

文|时青言 — 1 — 我是土生土长的农村娃，儿时家境十分贫穷，父母小学文凭，是名词中所谓的寒门。写下此文时，我已大学本科毕业，母校是一所211一本院

时青言 (/u/9787ddc08cad?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

 登录/注册

为你个性化推荐内容

(/sign\_in?utm\_source=desktop&utm\_medium=note-bottom-bind)

 下载简书App

随时随地发现和创作内容

(/apps/download?utm\_source=desktop&utm\_medium=click-note-bottom-bind)