

# iOS 视图控制器编程指南—— 支持无障碍

中国信息无障碍产品联盟&信息无障碍研究会 译制

刘辉 刘彪 沈广荣 20160722

[原文地址](#)

## 目录

1.简介.....	1
2.移动 VoiceOver 光标到特定元素 .....	1
3.响应特殊 VoiceOver 手势 .....	2
3.1 双指搓擦.....	3
3.2 魔法轻拍.....	3
3.3 三指滚动.....	3
3.4 增加和减少.....	3
4.观察无障碍通知.....	4

# 1.简介

一个可以被所有人使用的无障碍应用——包含能力障碍和肢体障碍的用户——保持应用的功能性和可用性是一个有用的工具。为了实现无障碍，一个 iOS 应用应该提供其用户界面元素的信息给 VoiceOver，视觉障碍用户就可以与这些元素进行交互了。UIKit 对象默认是无障碍的，但是开发者仍可以从视图控制器角度做某些事情来实现无障碍，这些事情包含：

- 保证界面中的每个用户元素可访问，包含控件和静态元素，例如 labels；
- 保证无障碍元素提供精确有用的信息。

通过程式设置 VoiceOver 焦点环的位置、响应特殊 VoiceOver 手势、观察无障碍通知，可以增强 VoiceOver 用户对应用的体验。

## 2.移动 VoiceOver 光标到特定元素

当应用在屏幕上呈现新的视图，思考设置 VoiceOver 光标的位置。当屏幕布局改变时，VoiceOver 焦点环，也被称为 VoiceOver 光标，重置其位置到屏幕上显示的从左到右从上到下的第一个元素。将光标放置在一个更加合适的元素上，可以加快界面的用户导航。例如，当新增一个新的视图控制器到导航控制器堆栈，VoiceOver 光标落在导航条的返回按钮上。开发者可能想要将光标移动到导航条的标题区域或者新弹出窗口的某个元素上。

为了改变光标的位置，使用 `UIAccessibilityPostNotification` 功能发送一个 `UIAccessibilityScreenChangedNotification` 通知。该通知告知 VoiceOver 屏幕内容改变了。当发送该通知时，指定想要哪个元素接收焦点，如表 6-1 所示。

表 6-1 发送一个无障碍通知改变第一个朗读的元素

```
@implementation MyViewController
- (void)viewDidAppear:(BOOL)animated {
    [super viewDidAppear:animated];
    //第二个参数是新的焦点元素。
```

```
UIAccessibilityPostNotification(UIAccessibilityScreenChangedNotification, self.myFirstElement);  
}  
@end
```

布局改变，包含旋转引起的改变，重置 VoiceOver 光标的位置。当视图控制器的布局改变时，发送 `UIAccessibilityLayoutChangedNotification` 通知。和 `UIAccessibilityScreenChangedNotification` 通知一样，可以指定想要 VoiceOver 第一个聚焦的新元素对象。

### 3. 响应特殊 VoiceOver 手势

为了触发应用特定操作，VoiceOver 定义了 5 个特殊手势。

**双指搓擦：** 一个双指 Z 字形手势，关闭模态对话框，或者返回导航层次中的上一级；

**魔法轻拍：** 一个双指双击手势，执行最期望的操作；

**三指滚动：** 一个三指扫动手势，垂直或水平滚动内容；

**增加：** 一个单指向上扫动手势，增加元素的值；

**减少：** 一个单指向下扫动手势，减少元素的值；

使用这些手势执行视图或视图控制器的特定任务。UIKit 会寻找一个方法实现与手势对应的方法。UIKit 使用 `responder` 链搜索该方法，从有 VoiceOver 焦点的元素开始。如果没有对象实现合适的方法，UIKit 为该手势执行系统默认操作。例如，如果在一个音乐应用中，从当前视图到应用门户都没有找到魔法轻拍的实现，魔法轻拍手势实现播放和暂停音乐。

尽管开发者可以在处理器中执行任何想要的操作，但是选择 VoiceOver 用户期望的应用操作应该遵循某些准则。跟其他手势一样，VoiceOver 手势的实现应该遵循一个模式，这样无障碍应用的交互就能符合直觉。

**注意：** 所有 VoiceOver 特殊手势方法返回一个布尔值，该值决定是否通过 `responder` 链传递。暂停传递，返回 YES，否则，返回 NO。

## 3.1 双指搓擦

使用 `accessibilityPerformEscape` 方法处理双指搓擦手势。对一个覆盖内容的视图——例如，一个模态对话框或一个警告框——使用该方法关闭该覆盖。双指搓擦手势的功能像电脑键盘 Esc 键的功能；它关闭一个临时对话框或浮层来恢复主要内容。开发者也可以使用双指搓擦手势在自定义导航层级中回到上一级。如果已经使用了 `UINavigationController` 对象，不需要实现该手势，因为 `UINavigationController` 对象已经处理了该手势。

## 3.2 魔法轻拍

使用 `accessibilityPerformMagicTap` 方法来处理魔法轻拍手势。魔法轻拍快速执行用户常用的或最想要的操作。例如，在电话应用中，一个魔法轻拍接听或挂断一个呼叫；在一个钟表应用中，魔法轻拍启动和停止秒表。也可能会使用该手势去触发与 VoiceOver 光标高亮元素没有必然关系的操作。在应用中任何位置处理魔法轻拍手势，都会应用门户中实现 `accessibilityPerformMagicTap` 方法。

## 3.3 三指滚动

使用 `accessibilityScroll` 方法滚动自定义视图的内容，当 VoiceOver 用户执行三指滚动手势。一个展示书页的自定义视图可能使用该手势来换页。传递到该方法的参数标识滚动的方法。

## 3.4 增加和减少

使用 `accessibilityIncrement` 和 `accessibilityDecrement` 方法来增加和减少元素的值。具有 `UIAccessibilityTraitAdjustable` 特性的元素必须实现该方法。

## 4. 观察无障碍通知

UIKit 发送无障碍通知来告知应用相关事件。应用对象可以观察任何相关通知，并使用他们实现合适的任务。例如，当 VoiceOver 完成页面上最后一行的阅读，iBook 应用使用 `UIAccessibilityAnnouncementDidFinishNotification` 通知来换页并继续阅读。该行为提供一个无缝的、无中断的阅读体验。

使用默认通知中心为无障碍通知注册一个观察器。表 6-2 展示了一个记录视图样例是否成功阅读通知或者是否被用户打断。

表 6-2 为无障碍通知注册一个观察器

```
@implementation MyViewController
- (void)viewDidLoad
{
    [super viewDidLoad];

    [[NSNotificationCenter defaultCenter]
     addObserver:self
     selector:@selector(didFinishAnnouncement:)
     name:UIAccessibilityAnnouncementDidFinishNotification
     object:nil];
}

- (void)didFinishAnnouncement:(NSNotification *)dict
{
    NSString *valueSpoken = [[dict userInfo]
objectForKey:UIAccessibilityAnnouncementKeyStringValue];
    NSString *wasSuccessful = [[dict userInfo]
objectForKey:UIAccessibilityAnnouncementKeyWasSuccessful];
    // ...
}
@end
```

订阅的另一个帮助通知是 [UIAccessibilityVoiceOverStatusChanged](#) 通知。当 VoiceOver 打开或关闭时，可以使用该通知进行删除。如果当应用挂起时，该通知出现，当应用回到前景时将会收到通知。

可以观察的无障碍通知列表，详见 [UIAccessibility 协议参考](#)。