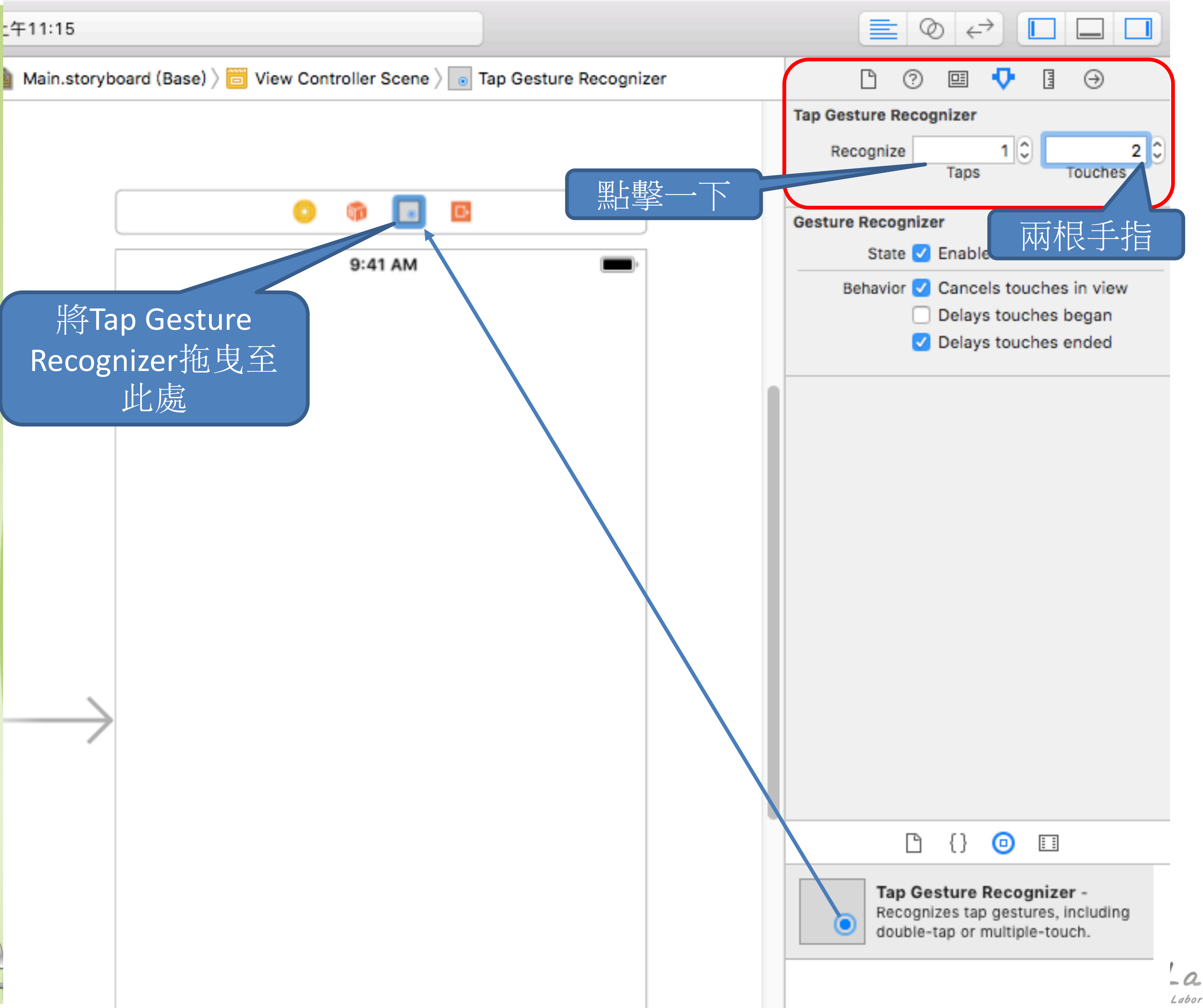
Class 08

TAIWAN TECH National Taiwan University of Science and Technology

HPCLab High-Performance Computing Laboratory

Gesture - Tap



Tap Gesture Recognizer

Recognize 1 Taps 2 Touches

Gesture Recognizer

State ☑ Enabled

Behavior ☑ Cancels touches in view

□ Delays touches began

☑ Delays touches ended

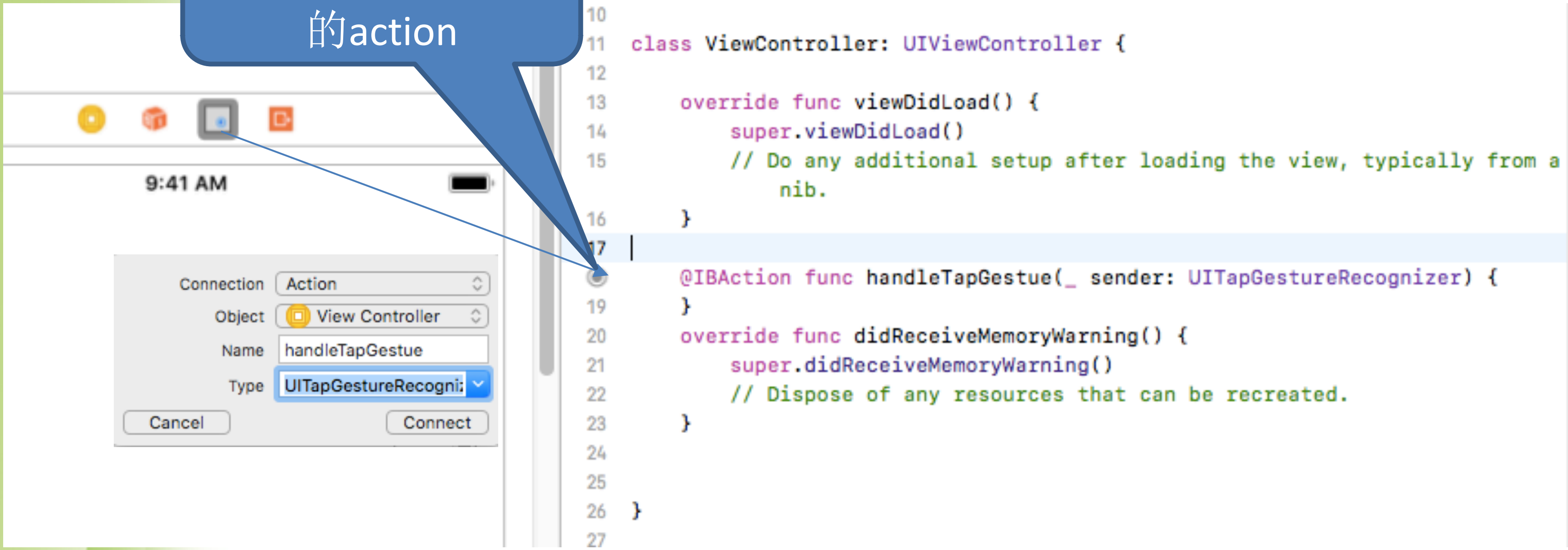
Tap Gesture Recognizer - Recognizes tap gestures, including double-tap or multiple-touch.

將Tap Gesture Recognizer拖曳至此處

點擊一下

兩根手指

Control+drag 建立 handleTapGesture 的action



9:41 AM

Connection: Action

Object: View Controller

Name: handleTapGestue

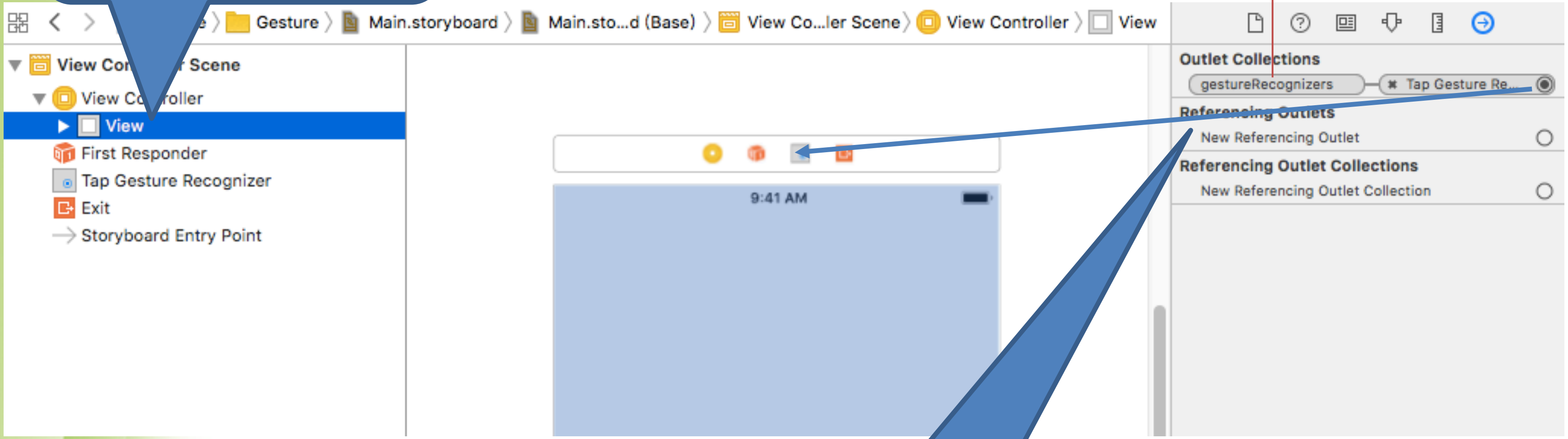
Type: UITapGestureRecogni:

```
class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    @IBAction func handleTapGestue(_ sender: UITapGestureRecognizer) {
    }
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


}
```

```
@IBAction func handleTapGesture(_ sender: UITapGestureRecognizer) {
    for i in 0 ..< sender.numberOfTouches {
        let point = sender.location(ofTouch: i, in: sender.view)
        print("第 \(i) 根手指座標為 (\(point.x), \(point.y))")
    }
}
```

Step1
點選View

gestureRecognizers



Gesture 〉 Main.storyboard 〉 Main.sto...d (Base) 〉 View Co...ler Scene 〉 View Controller 〉 View

- View Con... Scene
  - View Co...roller
    - View
  - First Responder
  - Tap Gesture Recognizer
  - Exit
  - Storyboard Entry Point

9:41 AM

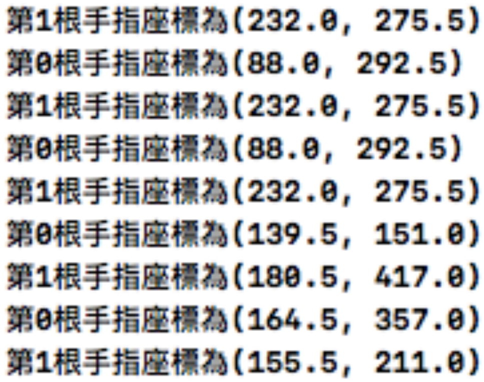
Outlet Collections
gestureRecognizers — Tap Gesture Re... ●

Referencing Outlets
New Referencing Outlet ○

Referencing Outlet Collections
New Referencing Outlet Collection ○

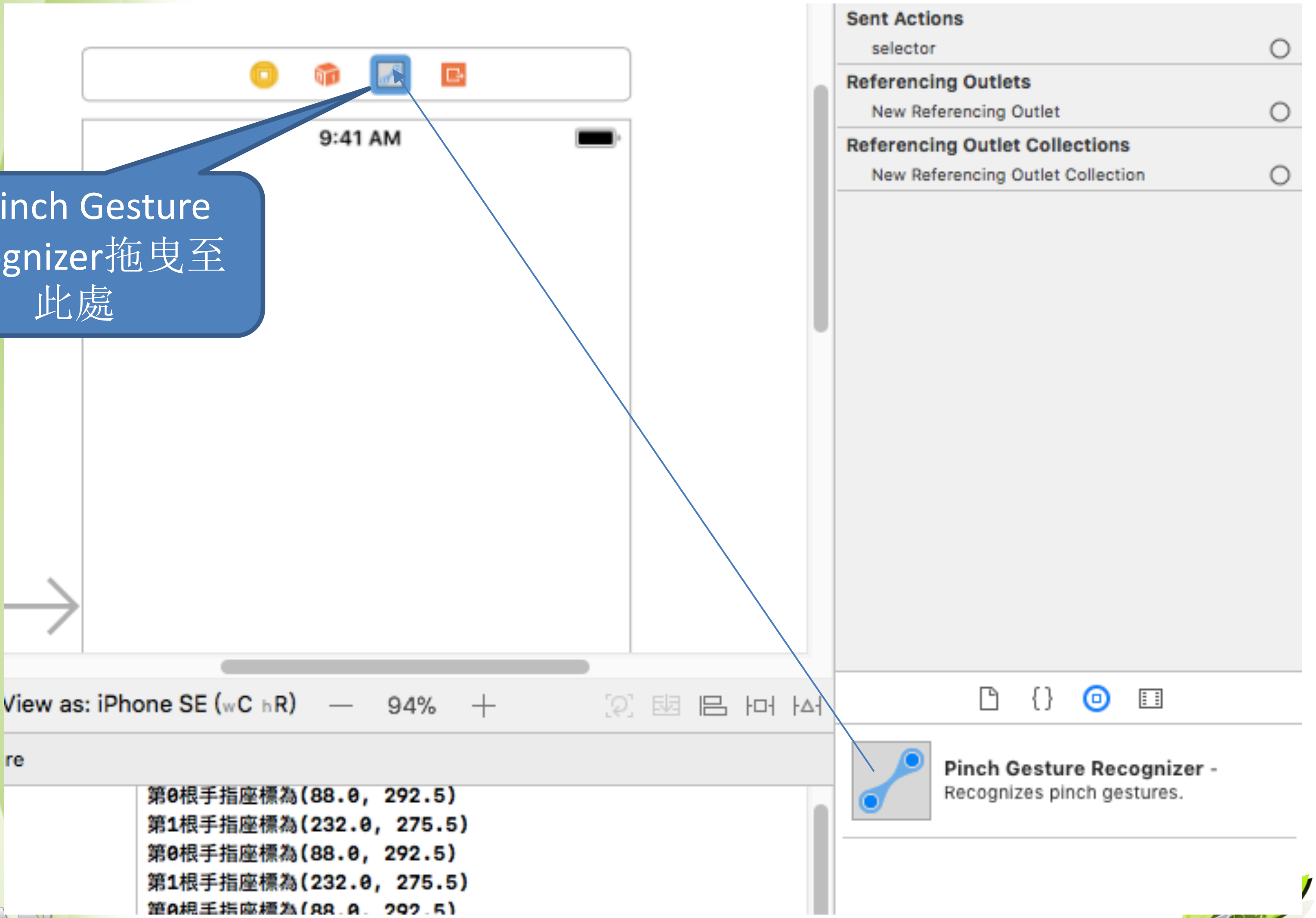
Step2
將tapGesture加入View
的gestureRecognizer中

Option + click



```
第1根手指座標為(232.0, 275.5)
第0根手指座標為(88.0, 292.5)
第1根手指座標為(232.0, 275.5)
第0根手指座標為(88.0, 292.5)
第1根手指座標為(232.0, 275.5)
第0根手指座標為(139.5, 151.0)
第1根手指座標為(180.5, 417.0)
第0根手指座標為(164.5, 357.0)
第1根手指座標為(155.5, 211.0)
```

Gesture - Pinch

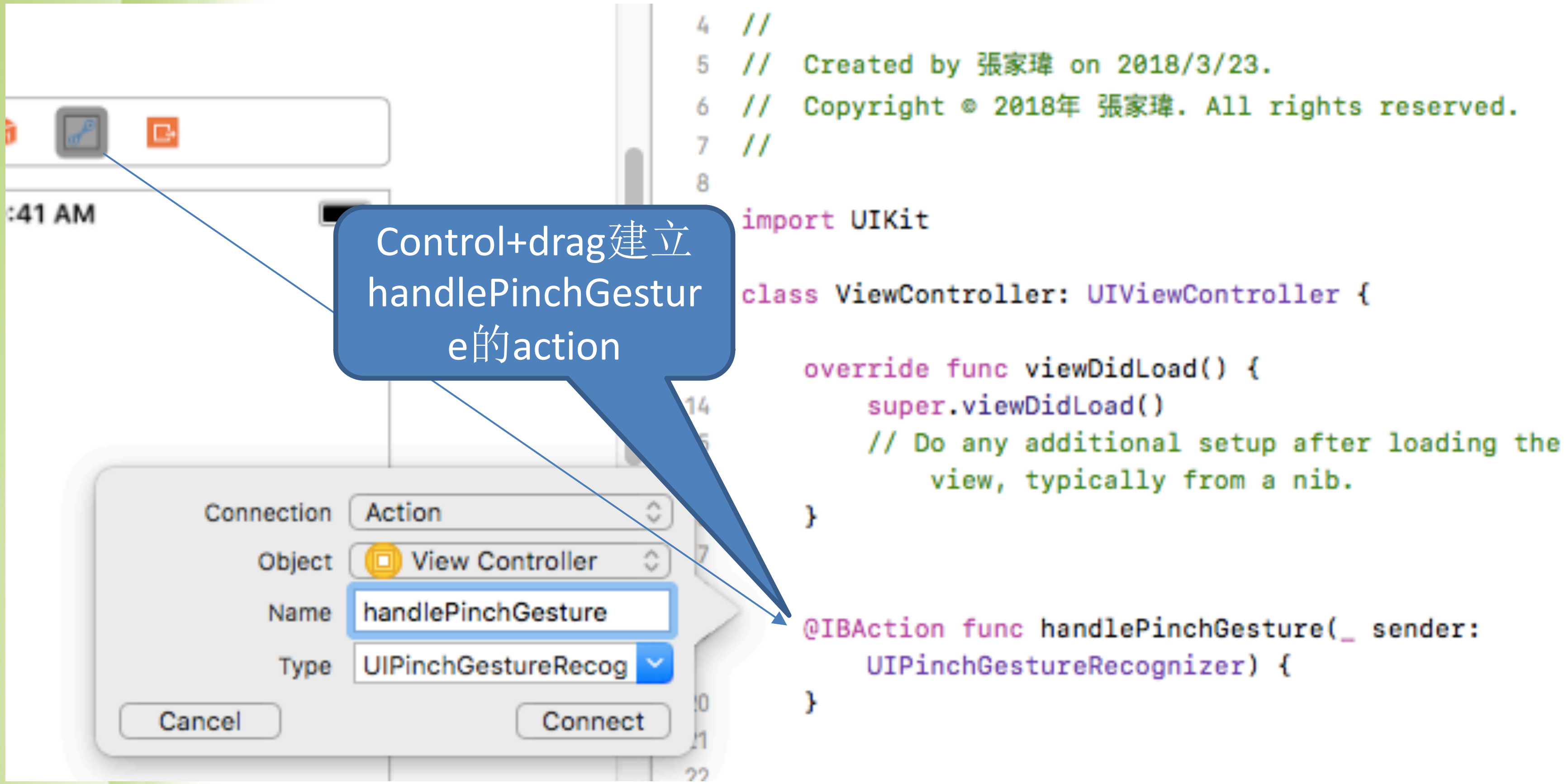


將Pinch Gesture Recognizer拖曳至此處

9:41 AM
View as: iPhone SE (wC hR)
—
94%
+
re
Sent Actions
selector ○
Referencing Outlets
New Referencing Outlet ○
Referencing Outlet Collections
New Referencing Outlet Collection ○

```
第0根手指座標為(88.0, 292.5)
第1根手指座標為(232.0, 275.5)
第0根手指座標為(88.0, 292.5)
第1根手指座標為(232.0, 275.5)
第0根手指座標為(88.0, 292.5)
```

Pinch Gesture Recognizer - Recognizes pinch gestures.



Control+drag建立handlePinchGesture的action

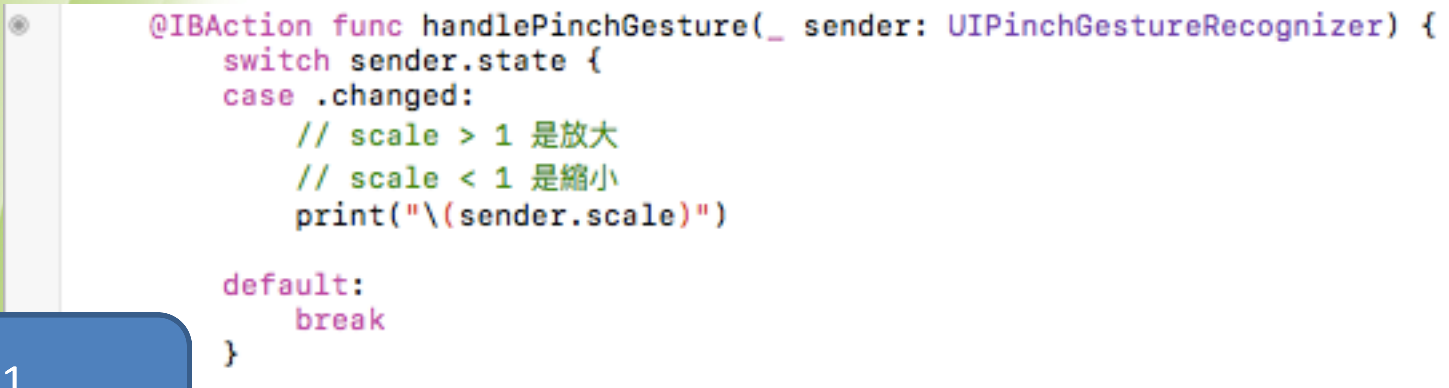
Connection: Action

Object: View Controller

Name: handlePinchGesture

Type: UIPinchGestureRecog

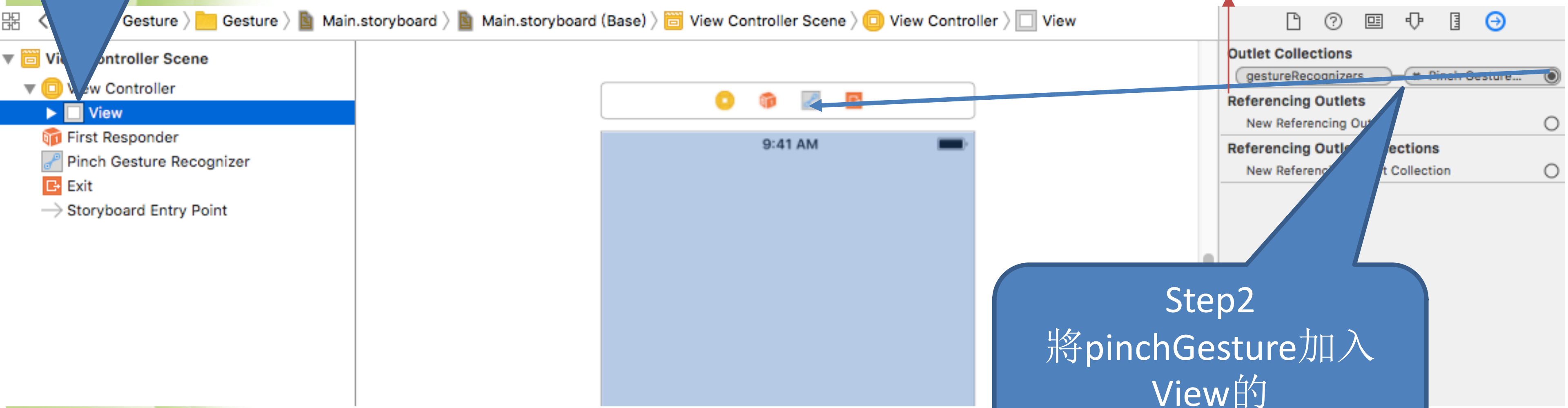
```
//
//  Created by 張家瑋 on 2018/3/23.
//  Copyright © 2018年 張家瑋. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the
            view, typically from a nib.
    }

    @IBAction func handlePinchGesture(_ sender:
        UIPinchGestureRecognizer) {
    }
```



```
@IBAction func handlePinchGesture(_ sender: UIPinchGestureRecognizer) {
    switch sender.state {
    case .changed:
        // scale > 1 是放大
        // scale < 1 是縮小
        print("\(sender.scale)")

    default:
        break
    }
```

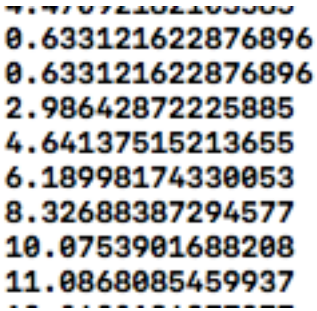
Step1
點選View

gestureRecognizers



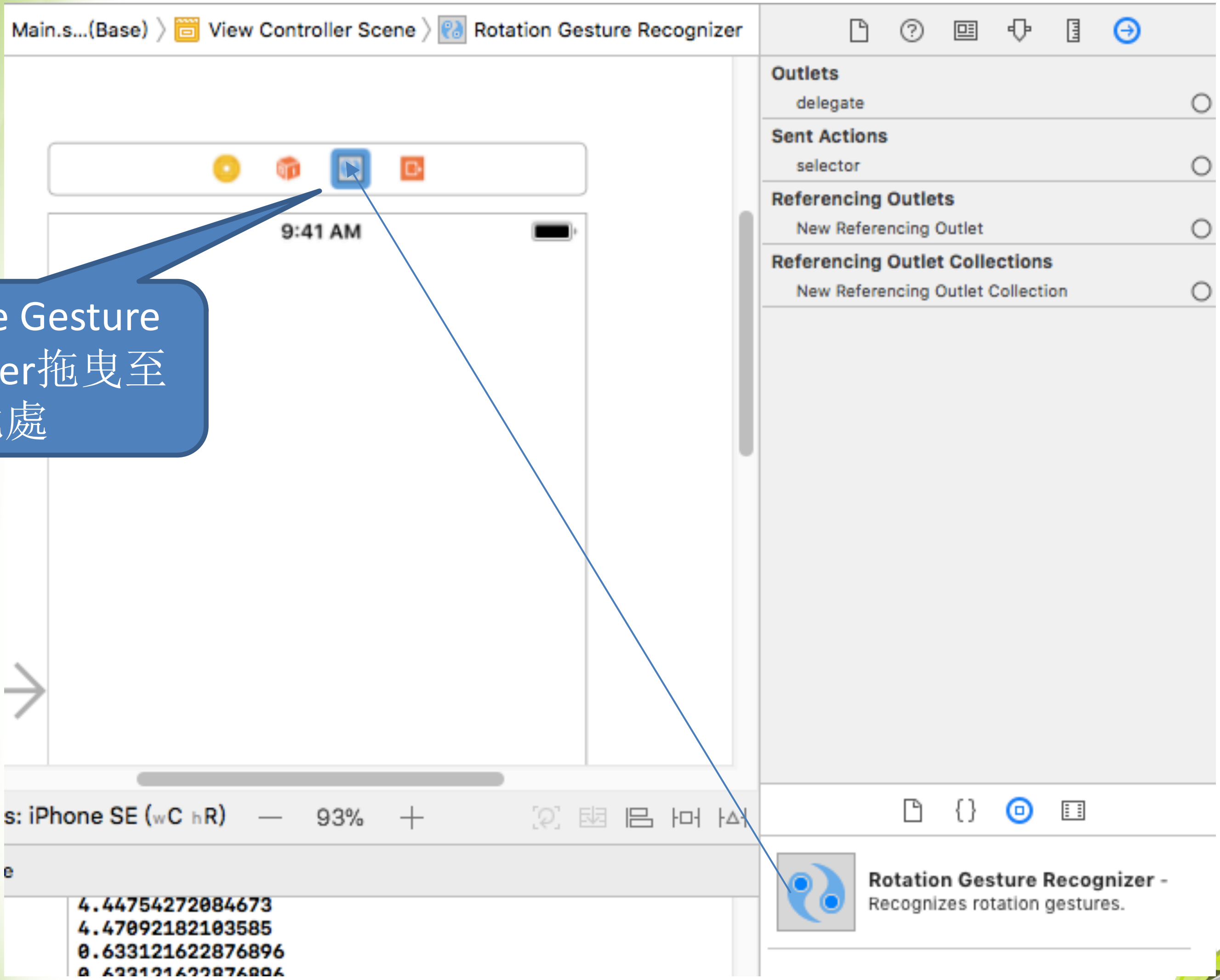
Step2
將pinchGesture加入View的gestureRecognizer中

option+click+drag



```
0.633121622876896
0.633121622876896
2.98642872225885
4.64137515213655
6.18998174330053
8.32688387294577
10.0753901688208
11.0868085459937
```

Gesture - Rotation



Main.s...(Base) 〉 View Controller Scene 〉 Rotation Gesture Recognizer

Outlets
- delegate ○

Sent Actions
- selector ○

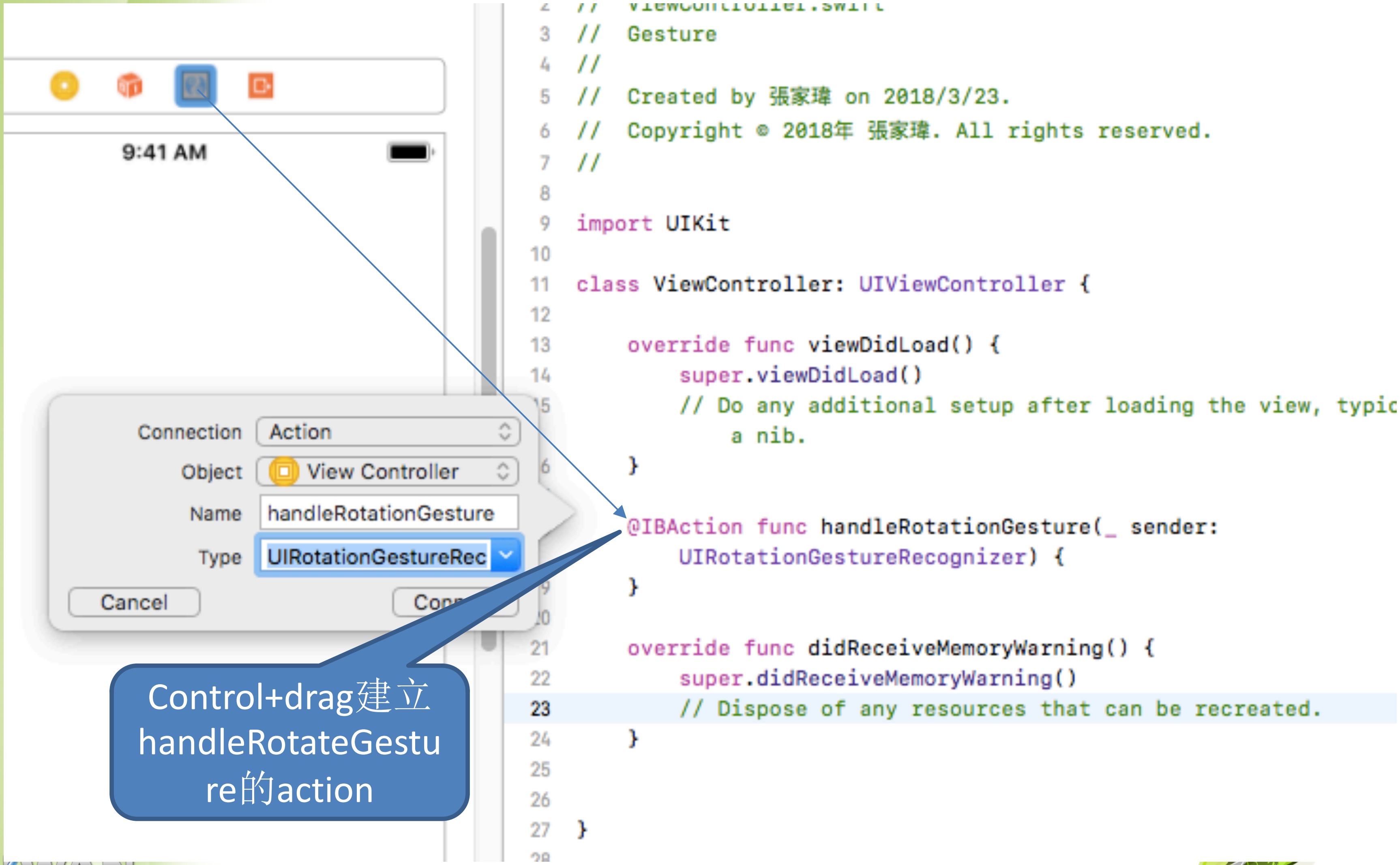
Referencing Outlets
- New Referencing Outlet ○

Referencing Outlet Collections
- New Referencing Outlet Collection ○

Rotation Gesture Recognizer - Recognizes rotation gestures.

```
4.44754272084673
4.47092182103585
0.633121622876896
0.633121622876896
```

將Rotate Gesture Recognizer拖曳至此處



Connection: Action

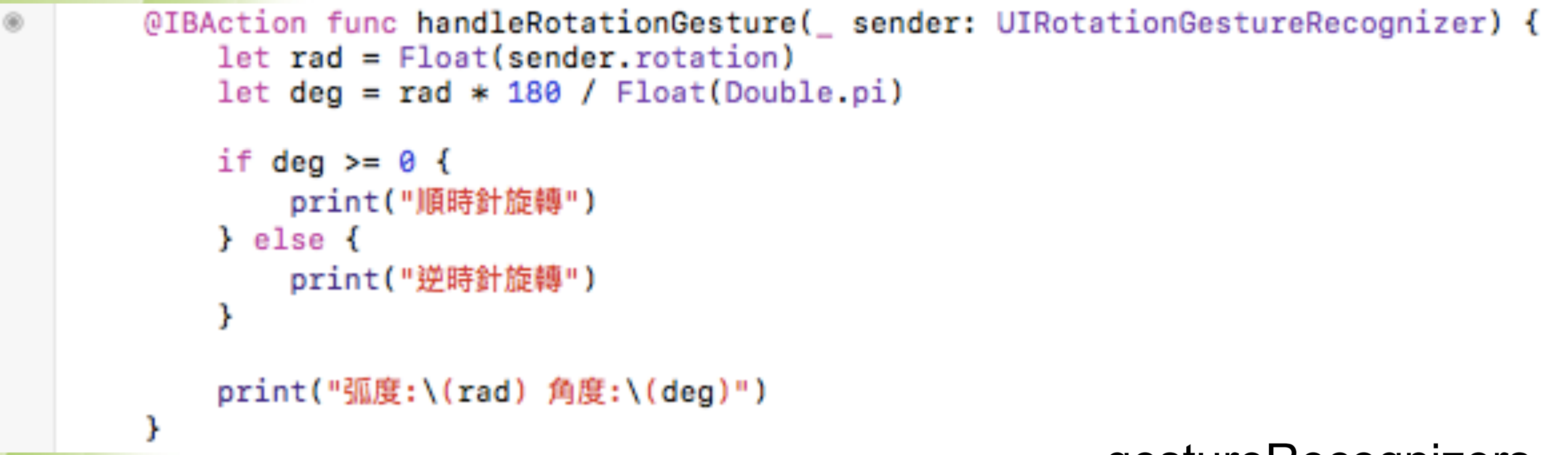
Object: View Controller

Name: handleRotationGesture

Type: UIRotationGestureRec

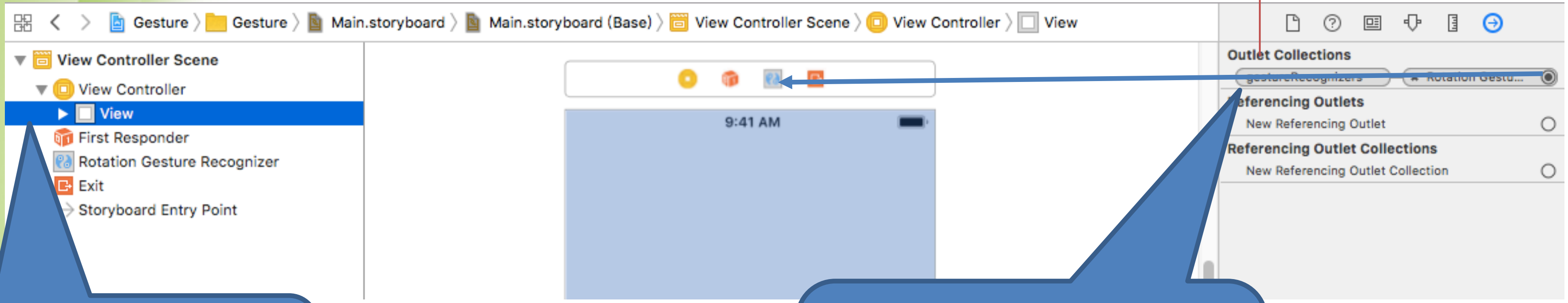
```
//  ViewController.swift
//  Gesture
//
//  Created by 張家瑋 on 2018/3/23.
//  Copyright © 2018年 張家瑋. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typic
            a nib.
    }

    @IBAction func handleRotationGesture(_ sender:
        UIRotationGestureRecognizer) {
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


}
```

Control+drag建立handleRotateGesture的action



```
@IBAction func handleRotationGesture(_ sender: UIRotationGestureRecognizer) {
    let rad = Float(sender.rotation)
    let deg = rad * 180 / Float(Double.pi)

    if deg >= 0 {
        print("順時針旋轉")
    } else {
        print("逆時針旋轉")
    }

    print("弧度:\(rad) 角度:\(deg)")
}
```

gestureRecognizers

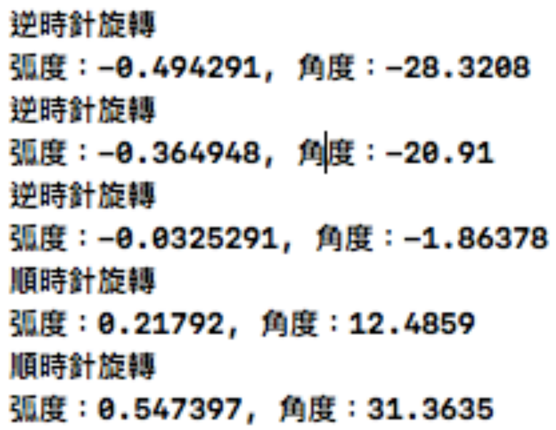


Gesture 〉 Gesture 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 View Controller Scene 〉 View Controller 〉 View
View Controller Scene
View Controller
View
First Responder
Rotation Gesture Recognizer
Exit
Storyboard Entry Point
9:41 AM
Outlet Collections
gestureRecognizers
Rotation Gestu...
Referencing Outlets
New Referencing Outlet ○
Referencing Outlet Collections
New Referencing Outlet Collection ○

Step1
點選View

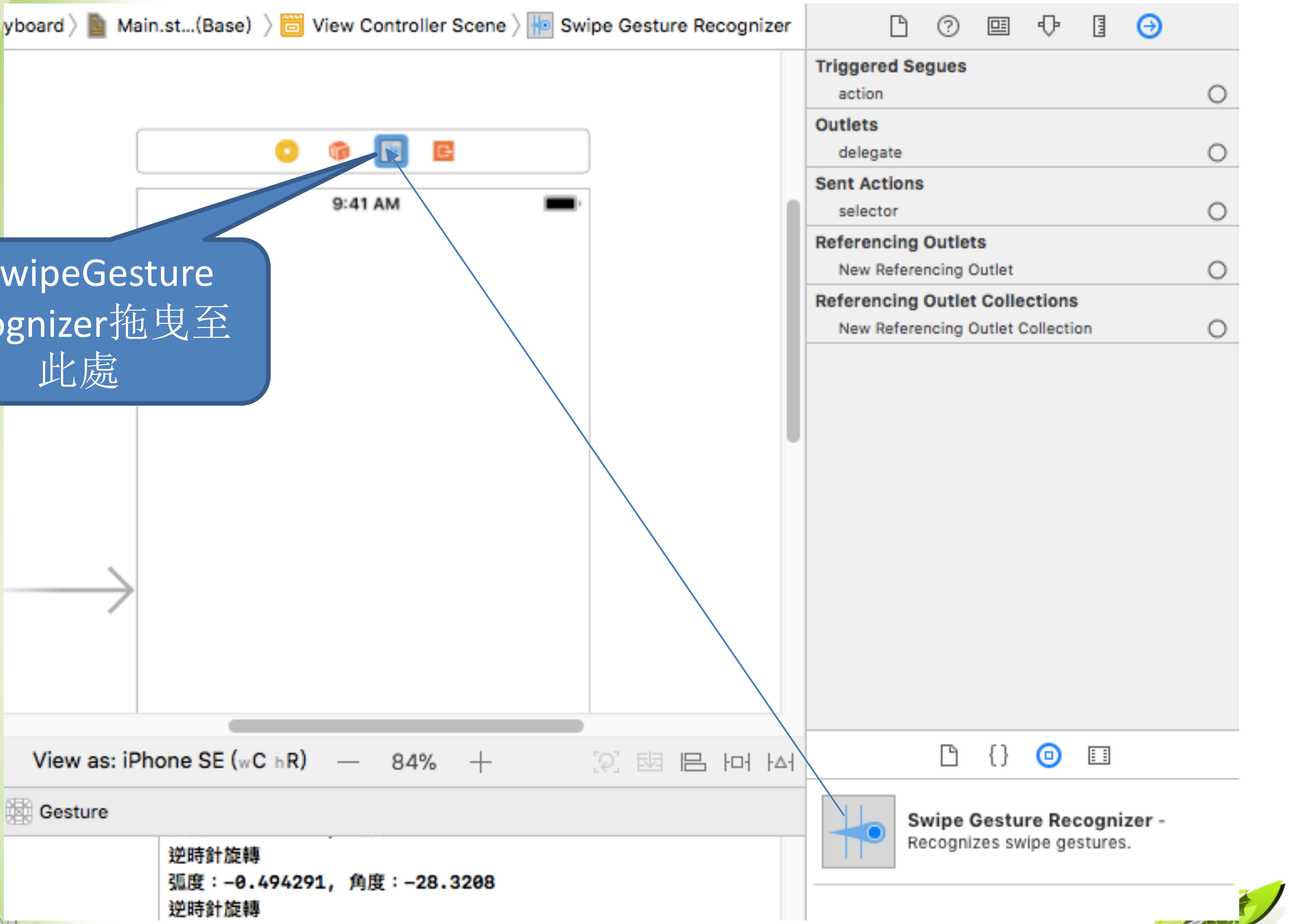
Step2
將rotateGesture加入View的gestureRecognizer中

option+click+drag



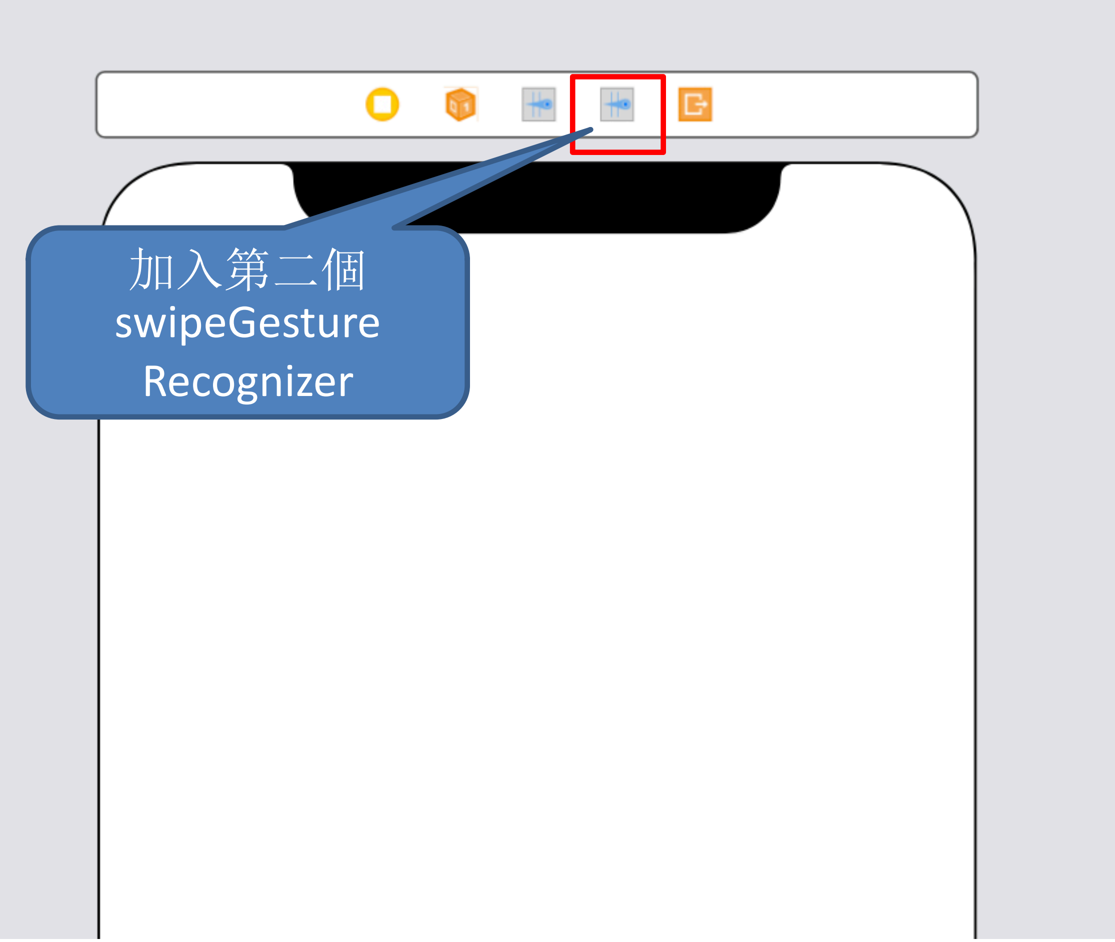
```
逆時針旋轉
弧度：-0.494291, 角度：-28.3208
逆時針旋轉
弧度：-0.364948, 角度：-20.91
逆時針旋轉
弧度：-0.0325291, 角度：-1.86378
順時針旋轉
弧度：0.21792, 角度：12.4859
順時針旋轉
弧度：0.547397, 角度：31.3635
```

Gesture - Swipe

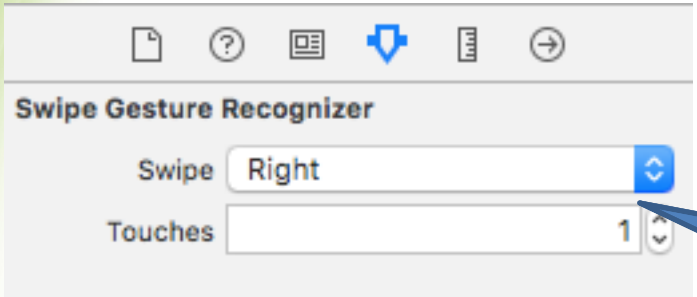

yboard 〉 Main.st...(Base) 〉 View Controller Scene 〉 Swipe Gesture Recognizer
9:41 AM
View as: iPhone SE (wC hR)
—
84%
+
Gesture
Triggered Segues
action
Outlets
delegate
Sent Actions
selector
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Swipe Gesture Recognizer - Recognizes swipe gestures.

將swipeGesture Recognizer拖曳至此處

```
逆時針旋轉
弧度：-0.494291, 角度：-28.3208
逆時針旋轉
```



加入第二個 swipeGesture Recognizer

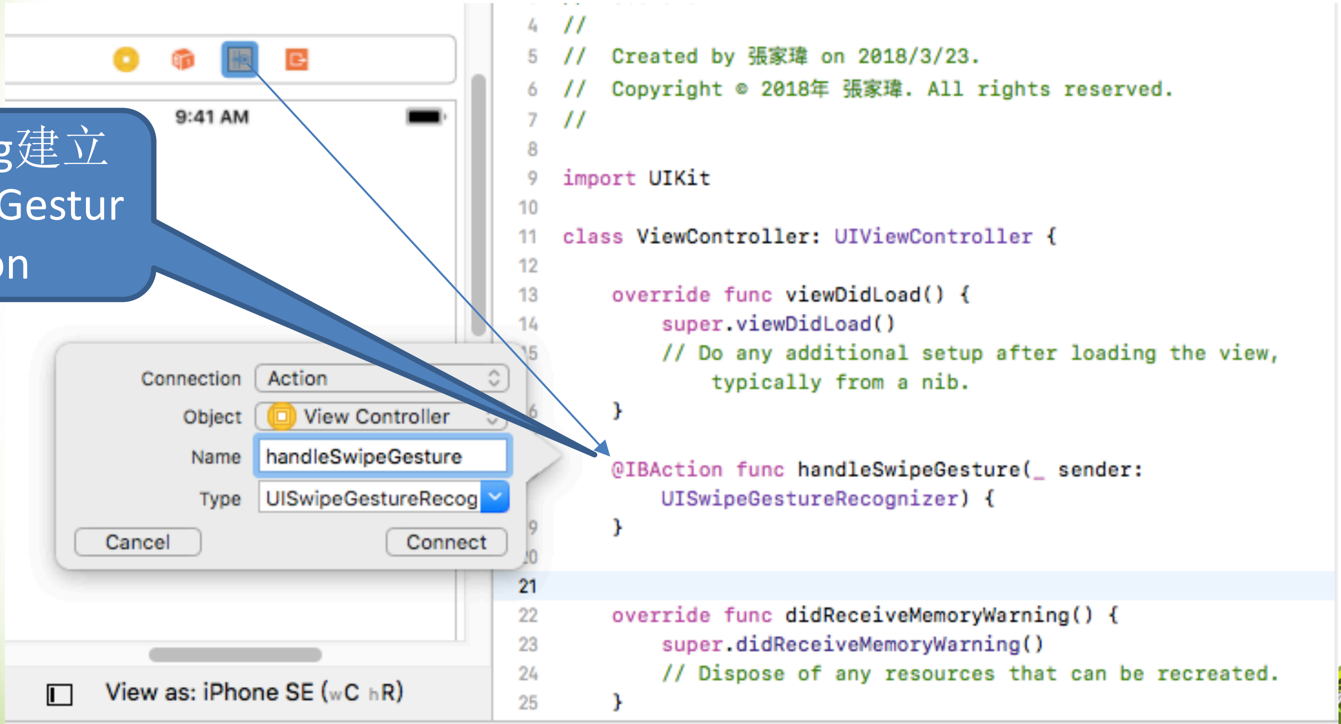


Swipe Gesture Recognizer

Swipe: Right

Touches: 1

一根指頭向右滑



Connection: Action

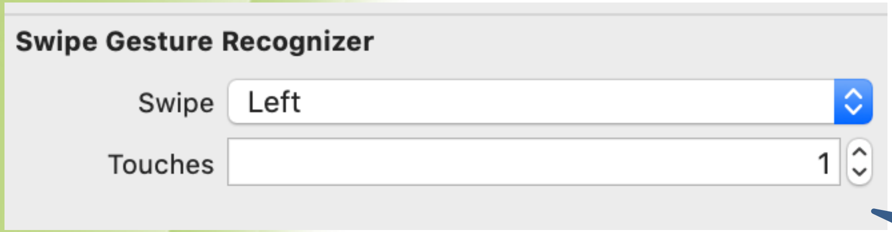
Object: View Controller

Name: handleSwipeGesture

Type: UISwipeGestureRecog

```
//
//  Created by 張家瑋 on 2018/3/23.
//  Copyright © 2018年 張家瑋. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view,
            typically from a nib.
    }

    @IBAction func handleSwipeGesture(_ sender:
        UISwipeGestureRecognizer) {
    }


    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
```

Control+drag建立handleSwipeGesture的action

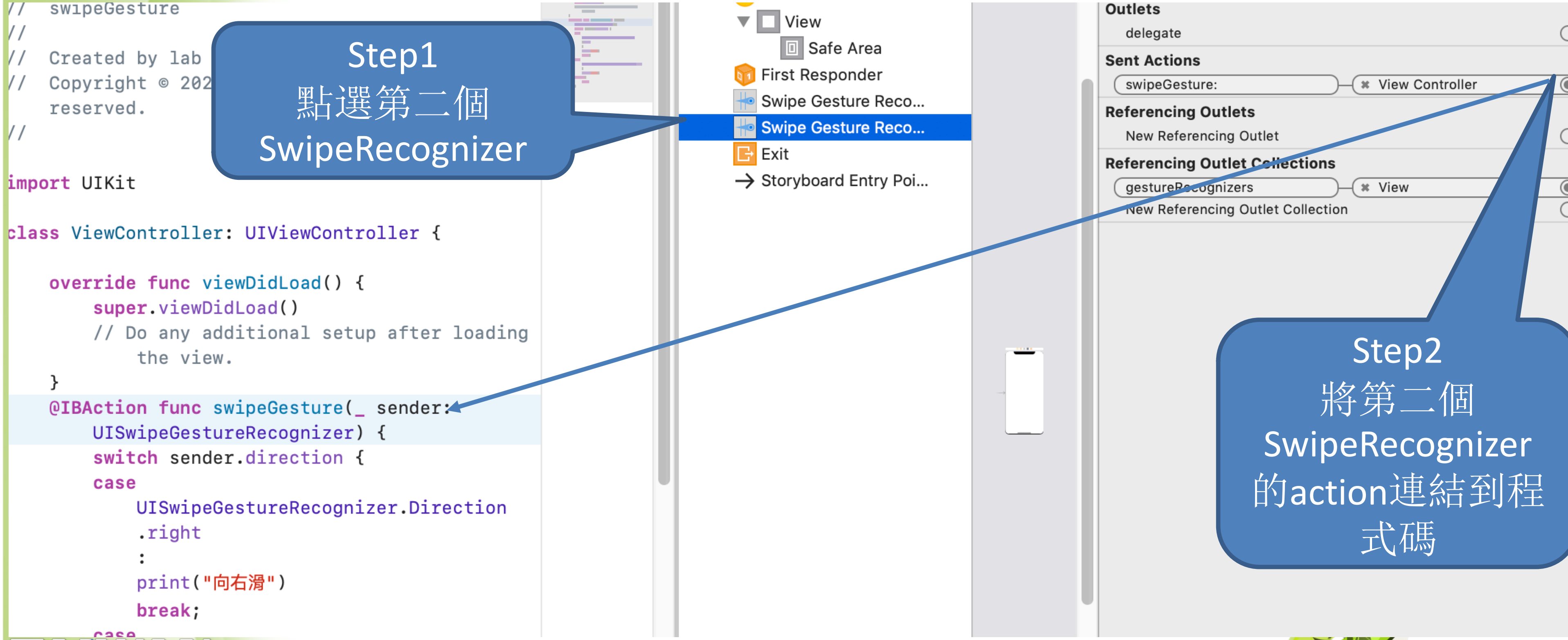


Swipe Gesture Recognizer

Swipe: Left

Touches: 1

一根指頭向左滑



```
//  swipeGesture
//
//  Created by lab
//  Copyright © 202
    reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading
            the view.
    }
    @IBAction func swipeGesture(_ sender:
        UISwipeGestureRecognizer) {
        switch sender.direction {
        case
            UISwipeGestureRecognizer.Direction
            .right
            :
            print("向右滑")
            break;
        case
```

Outlets
delegate

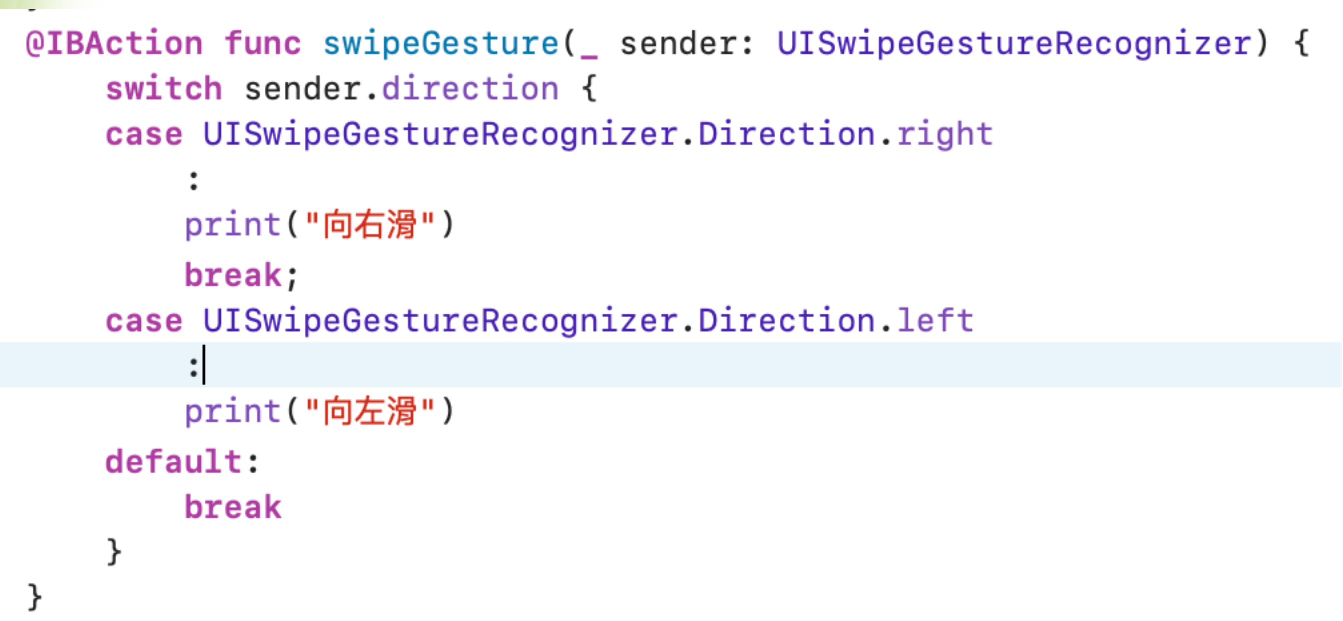
Sent Actions
swipeGesture: — View Controller

Referencing Outlets
New Referencing Outlet

Referencing Outlet Collections
gestureRecognizers — View
New Referencing Outlet Collection

Step1
點選第二個
SwipeRecognizer

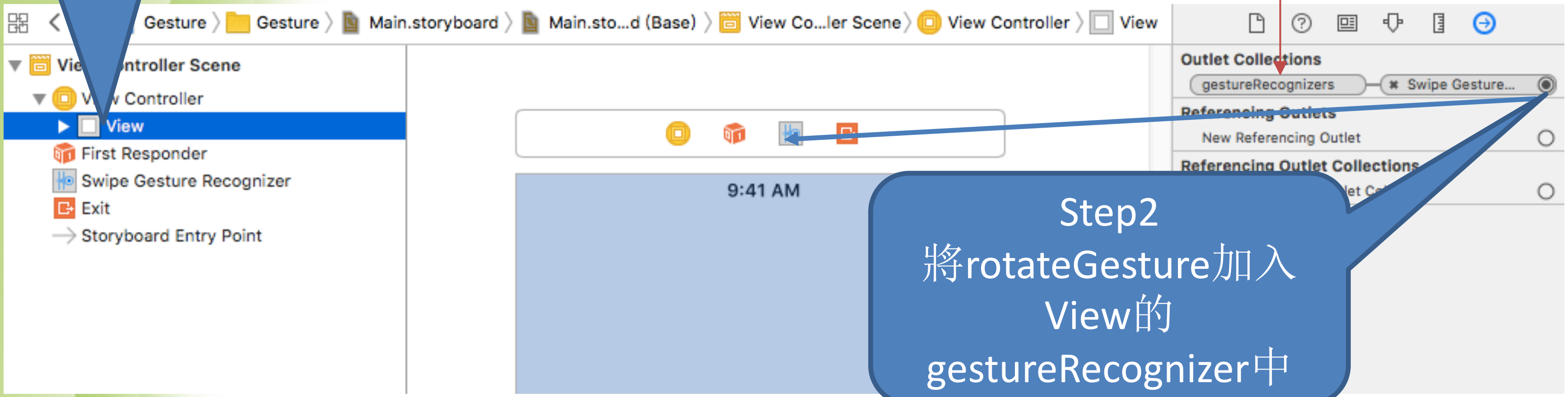
Step2
將第二個
SwipeRecognizer
的action連結到程
式碼



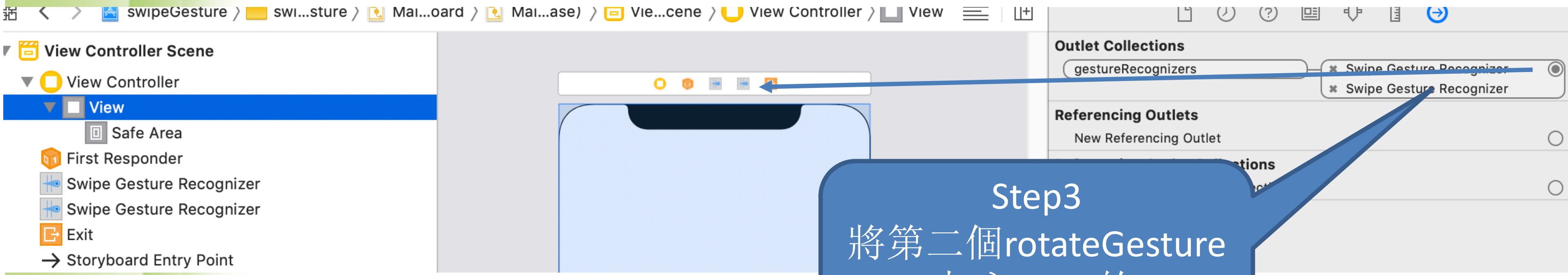
```
@IBAction func swipeGesture(_ sender: UISwipeGestureRecognizer) {
    switch sender.direction {
    case UISwipeGestureRecognizer.Direction.right
        :
        print("向右滑")
        break;
    case UISwipeGestureRecognizer.Direction.left
        :
        print("向左滑")
    default:
        break
    }
}
```

Step1
點選View

gestureRecognizers

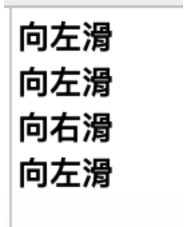


Step2
將rotateGesture加入View的gestureRecognizer中



Step3
將第二個rotateGesture加入View的gestureRecognizer中

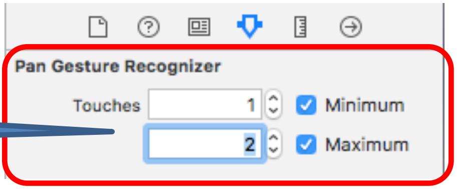
click+drag



向左滑
向左滑
向右滑
向左滑

Gesture - Pan

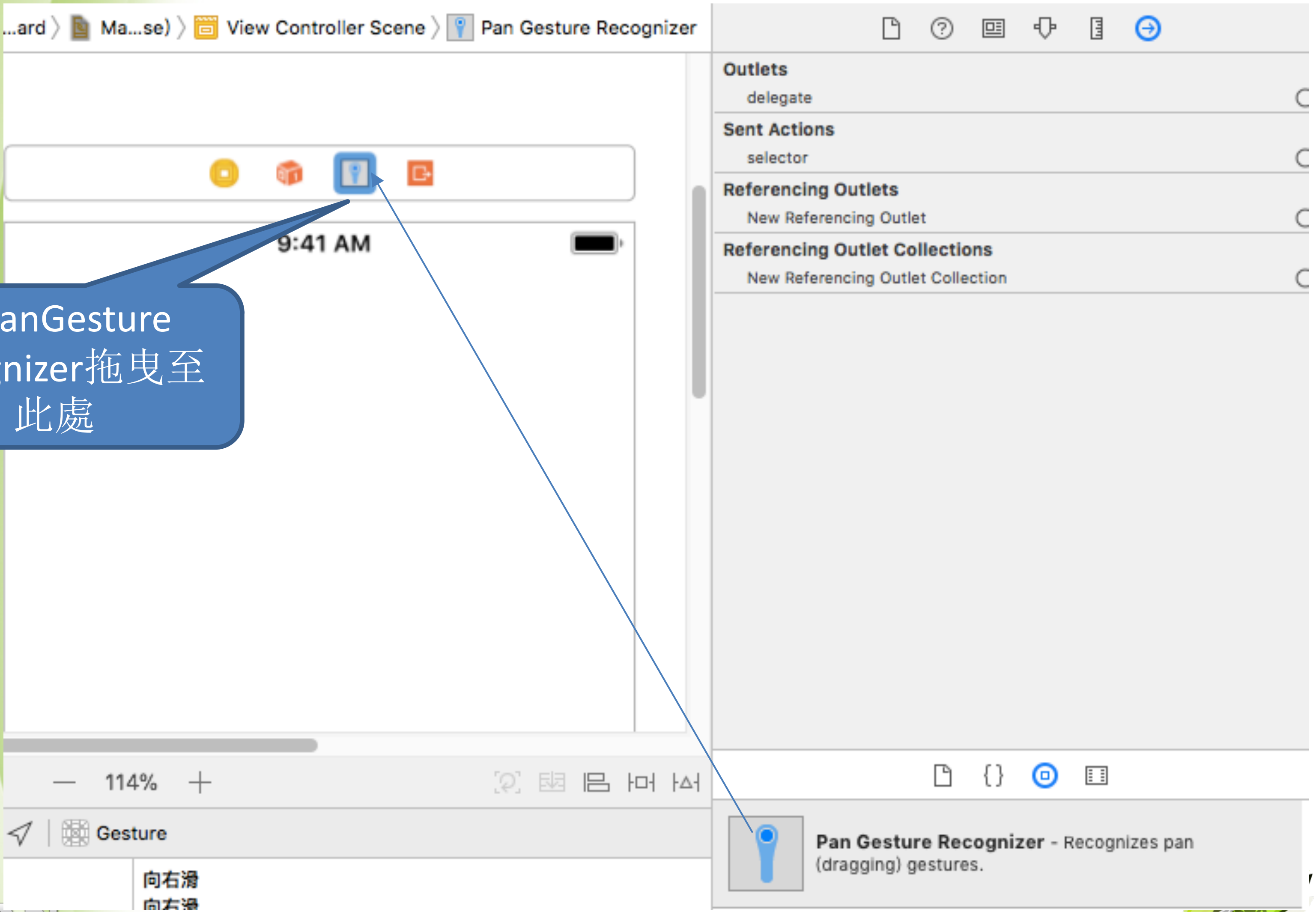
觸發action所需使用的手指數範圍



Pan Gesture Recognizer

Touches 1 ☑ Minimum

2 ☑ Maximum



...ard 〉 Ma...se) 〉 View Controller Scene 〉 Pan Gesture Recognizer

Outlets
delegate

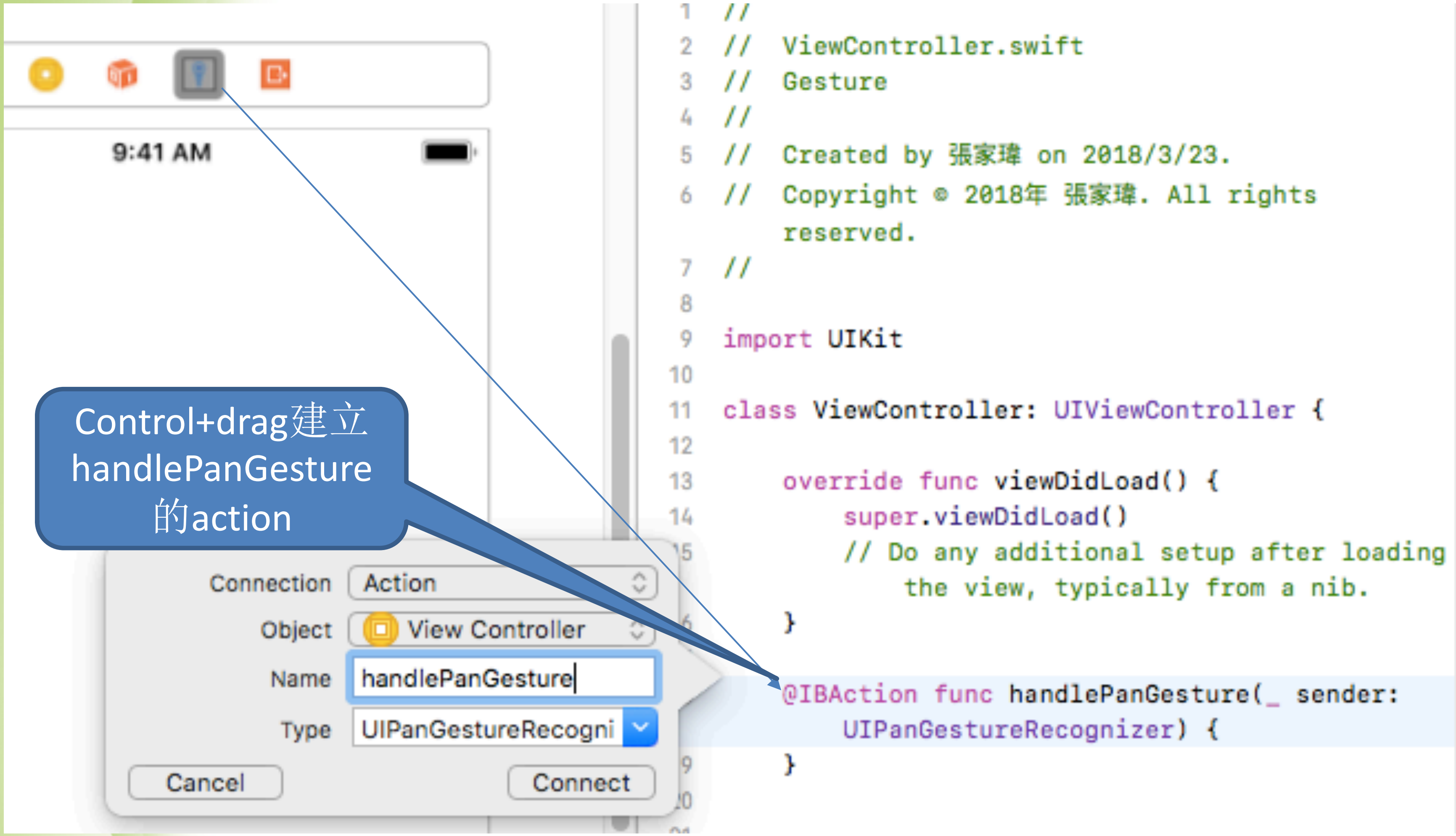
Sent Actions
selector

Referencing Outlets
New Referencing Outlet

Referencing Outlet Collections
New Referencing Outlet Collection

Pan Gesture Recognizer - Recognizes pan (dragging) gestures.

將panGesture Recognizer拖曳至此處



Control+drag建立handlePanGesture的action

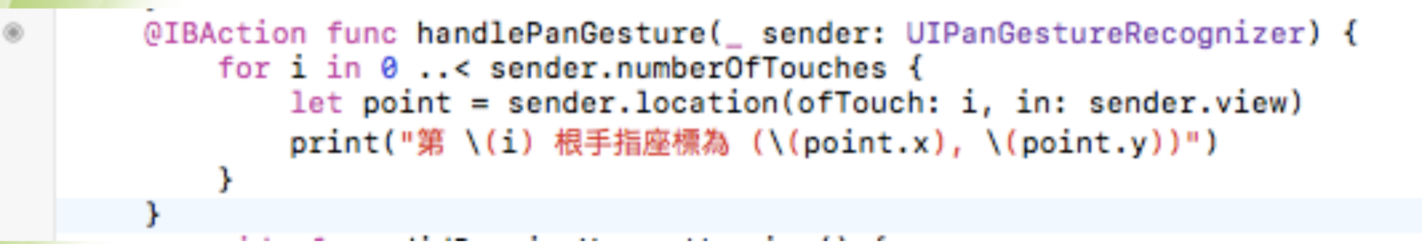
Connection: Action

Object: View Controller

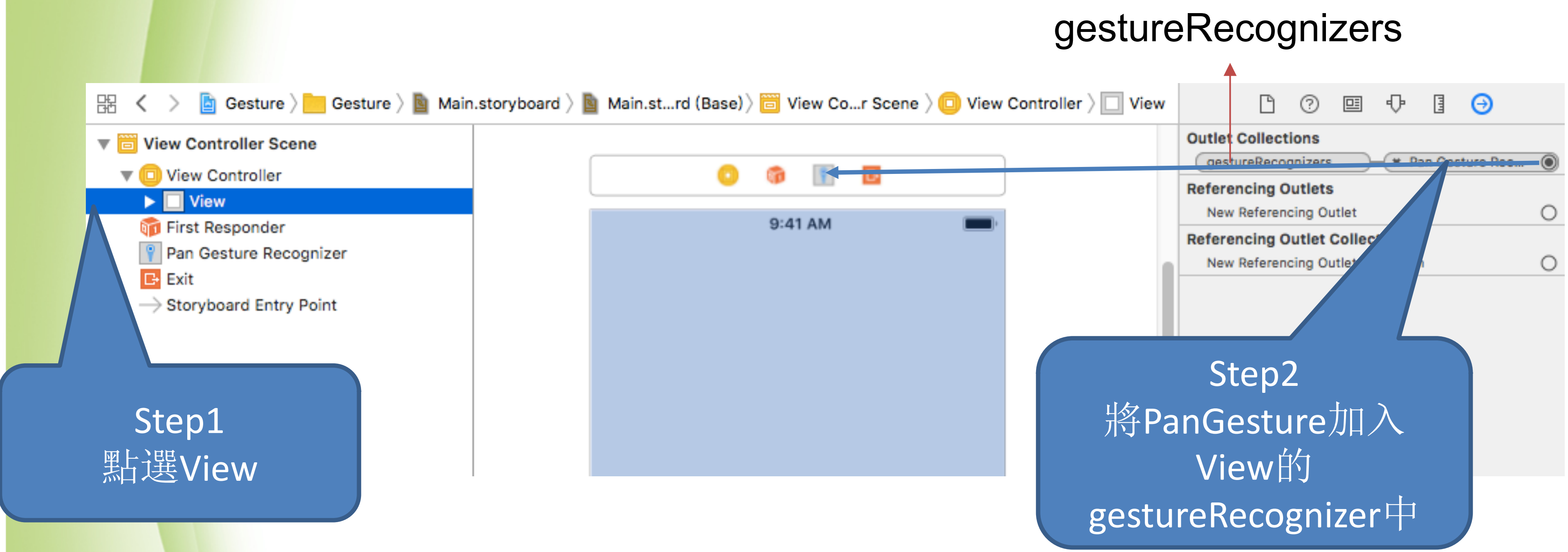
Name: handlePanGesture

Type: UIPanGestureRecogni

```
//
//  ViewController.swift
//  Gesture
//
//  Created by 張家瑋 on 2018/3/23.
//  Copyright © 2018年 張家瑋. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    @IBAction func handlePanGesture(_ sender: UIPanGestureRecognizer) {
    }
```



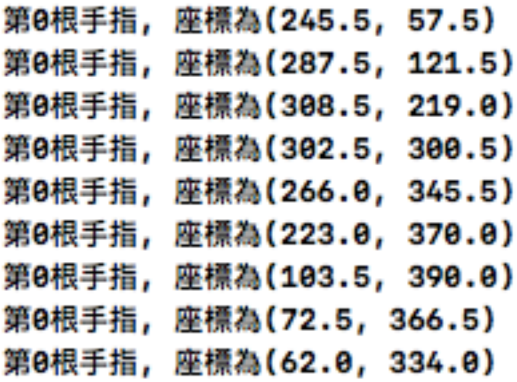
```
@IBAction func handlePanGesture(_ sender: UIPanGestureRecognizer) {
    for i in 0 ..< sender.numberOfTouches {
        let point = sender.location(ofTouch: i, in: sender.view)
        print("第 \(i) 根手指座標為 (\(point.x), \(point.y))")
    }
}
```



Step1
點選View

Step2
將PanGesture加入View的gestureRecognizer中

Click + drag

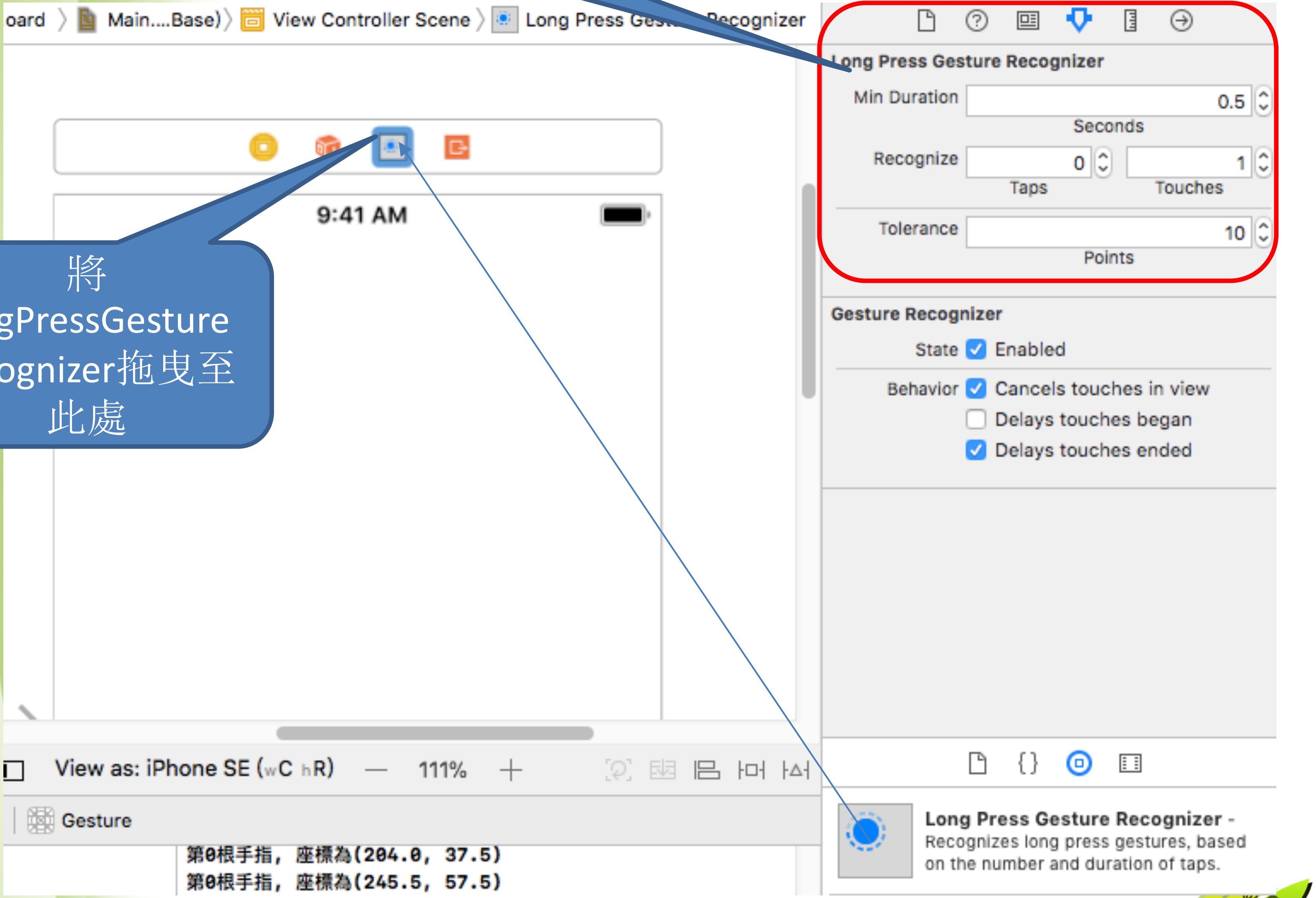


```
第0根手指, 座標為(245.5, 57.5)
第0根手指, 座標為(287.5, 121.5)
第0根手指, 座標為(308.5, 219.0)
第0根手指, 座標為(302.5, 300.5)
第0根手指, 座標為(266.0, 345.5)
第0根手指, 座標為(223.0, 370.0)
第0根手指, 座標為(103.5, 390.0)
第0根手指, 座標為(72.5, 366.5)
第0根手指, 座標為(62.0, 334.0)
```

Gesture – Long Press

長按0.5秒後觸發action

將longPressGestureRecognizer拖曳至此處



Long Press Gesture Recognizer

Min Duration 0.5 Seconds

Recognize 0 Taps 1 Touches

Tolerance 10 Points

Gesture Recognizer

State ☑ Enabled

Behavior ☑ Cancels touches in view

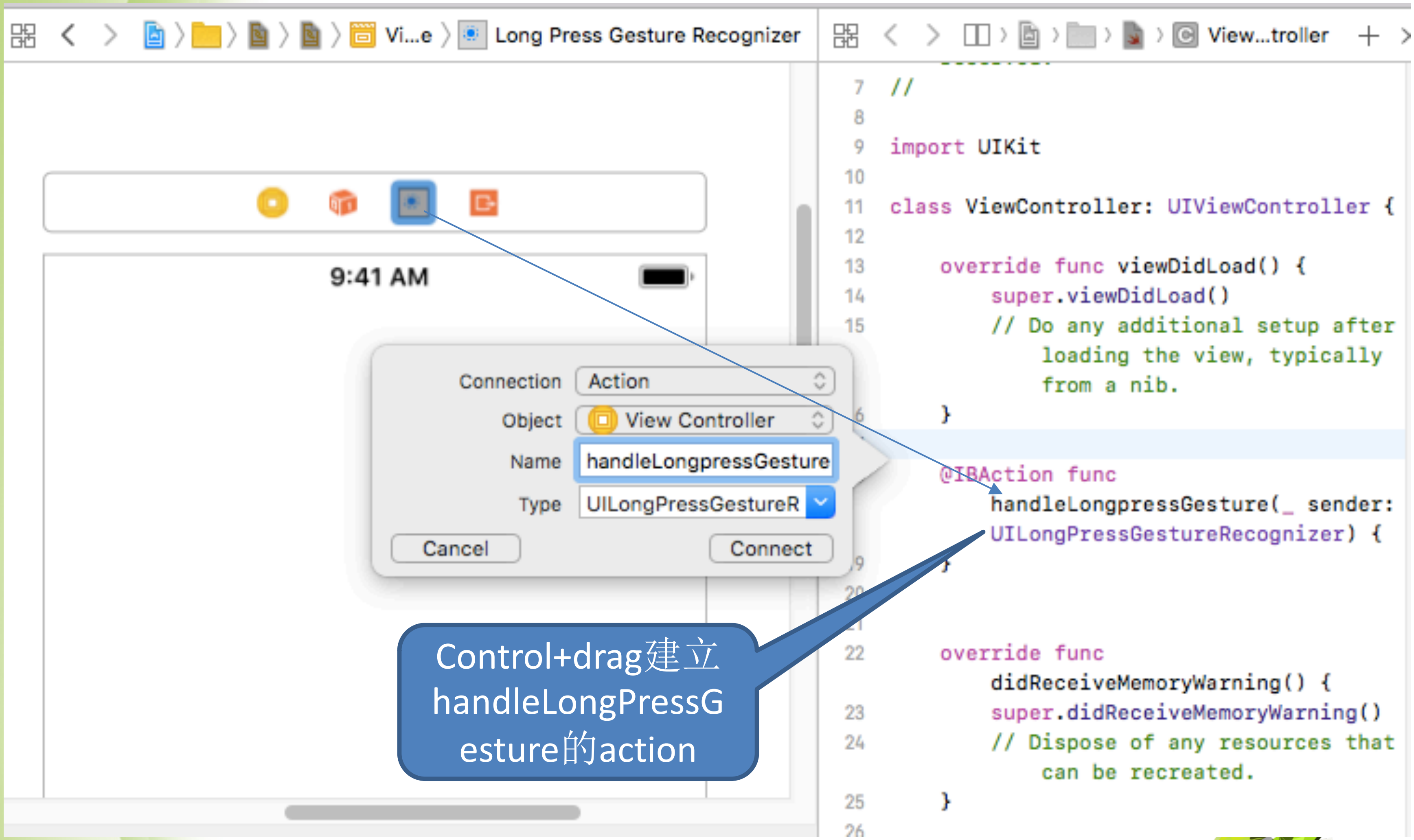
□ Delays touches began

☑ Delays touches ended

Long Press Gesture Recognizer - Recognizes long press gestures, based on the number and duration of taps.

```
第0根手指, 座標為(204.0, 37.5)
第0根手指, 座標為(245.5, 57.5)
```

oard 〉 Main....Base) 〉 View Controller Scene 〉 Long Press Gesture Recognizer
9:41 AM
View as: iPhone SE (wC hR)
111%
Gesture



Connection: Action

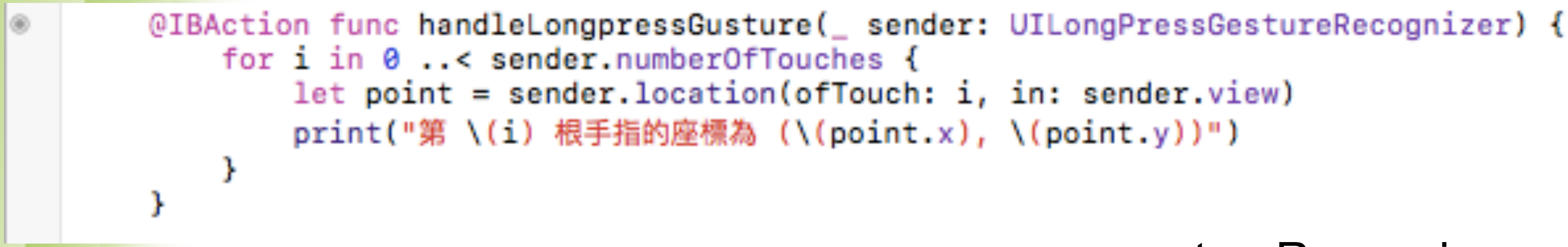
Object: View Controller

Name: handleLongpressGesture

Type: UILongPressGestureR

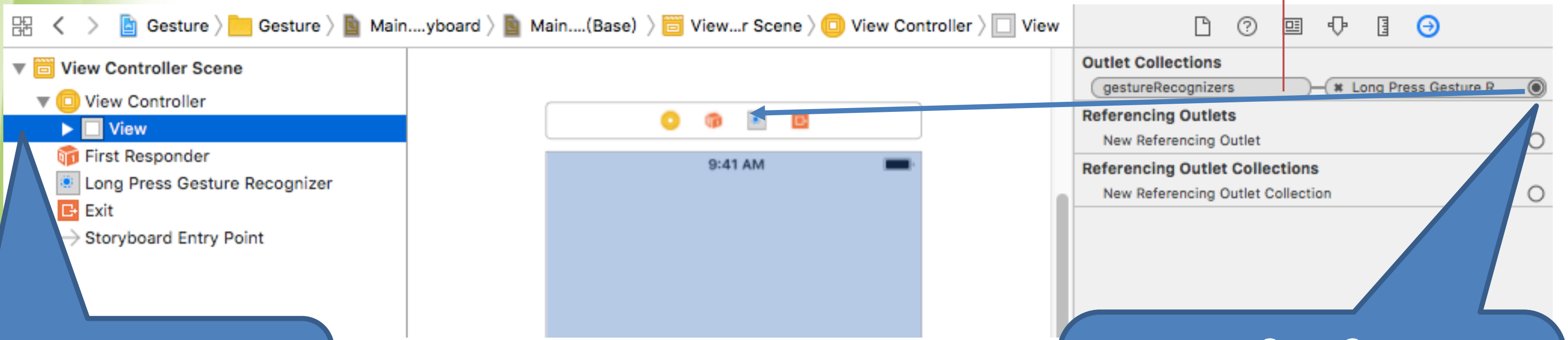
```
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after
            loading the view, typically
            from a nib.
    }

    @IBAction func
        handleLongpressGesture(_ sender:
        UILongPressGestureRecognizer) {
    }


    override func
        didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that
            can be recreated.
    }
```

Control+drag建立handleLongPressGesture的action



```
@IBAction func handleLongpressGusture(_ sender: UILongPressGestureRecognizer) {
    for i in 0 ..< sender.numberOfTouches {
        let point = sender.location(ofTouch: i, in: sender.view)
        print("第 \(i) 根手指的座標為 (\(point.x), \(point.y))")
    }
}
```

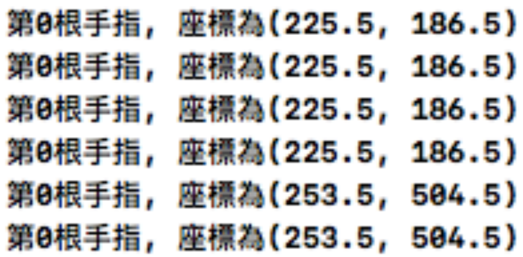
gestureRecognizers


Gesture 〉 Gesture 〉 Main....yboard 〉 Main....(Base) 〉 View...r Scene 〉 View Controller 〉 View
View Controller Scene
View Controller
View
First Responder
Long Press Gesture Recognizer
Exit
Storyboard Entry Point
9:41 AM
Outlet Collections
gestureRecognizers
Long Press Gesture R
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection

Step1
點選View

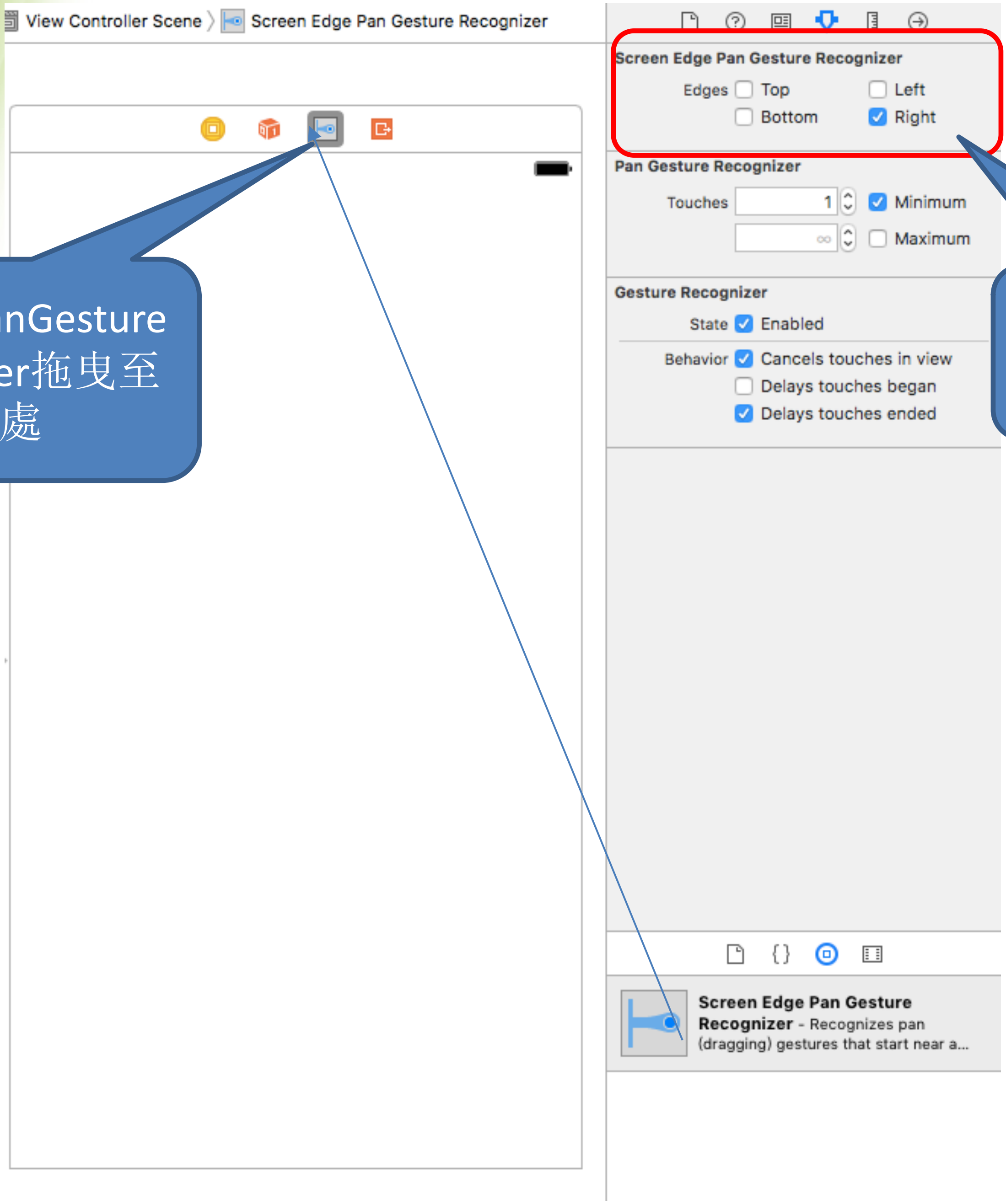
Step2
將longPressGesture加入View的gestureRecognizer中

Long press



```
第0根手指, 座標為(225.5, 186.5)
第0根手指, 座標為(225.5, 186.5)
第0根手指, 座標為(225.5, 186.5)
第0根手指, 座標為(225.5, 186.5)
第0根手指, 座標為(253.5, 504.5)
第0根手指, 座標為(253.5, 504.5)
```

Gesture – Edge Pan



View Controller Scene 〉 Screen Edge Pan Gesture Recognizer

Screen Edge Pan Gesture Recognizer

Edges □ Top □ Left
□ Bottom ☑ Right

Pan Gesture Recognizer

Touches 1 ☑ Minimum
∞ □ Maximum

Gesture Recognizer

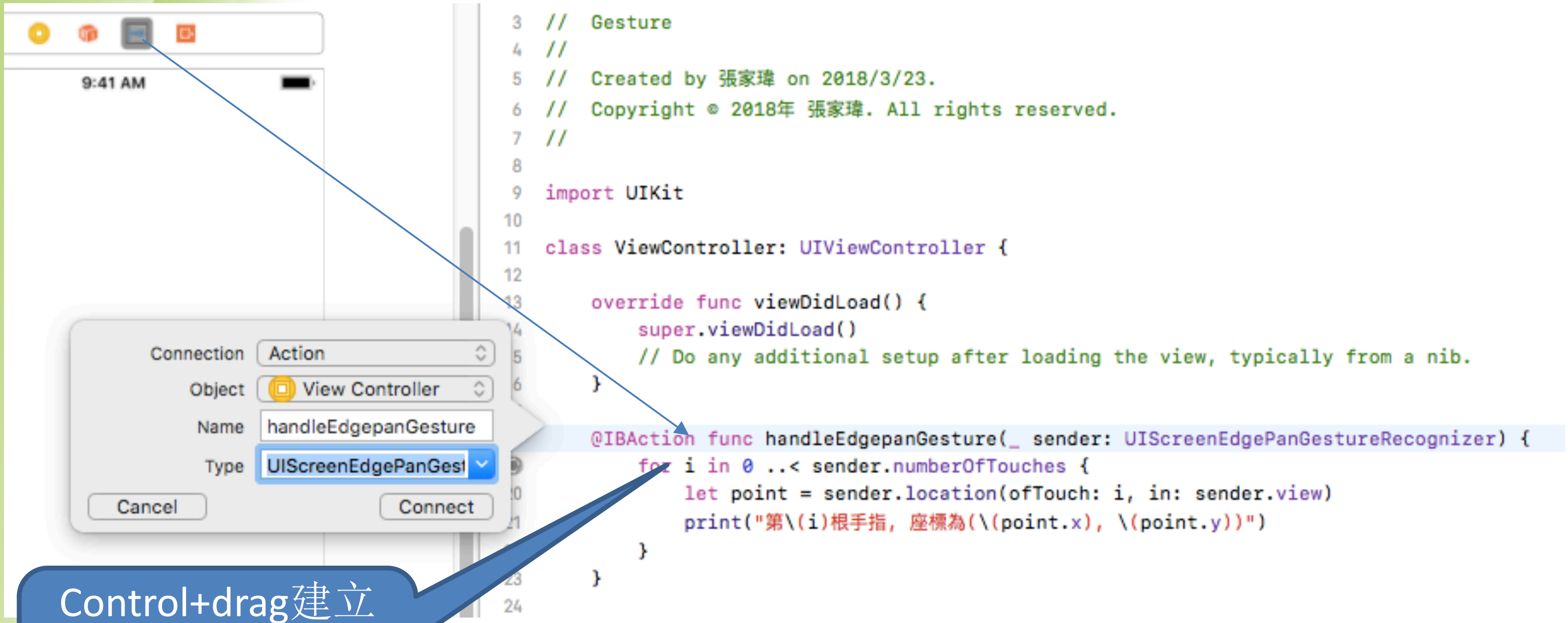
State ☑ Enabled

Behavior ☑ Cancels touches in view
□ Delays touches began
☑ Delays touches ended

Screen Edge Pan Gesture Recognizer - Recognizes pan (dragging) gestures that start near a...

將edgePanGesture Recognizer拖曳至此處

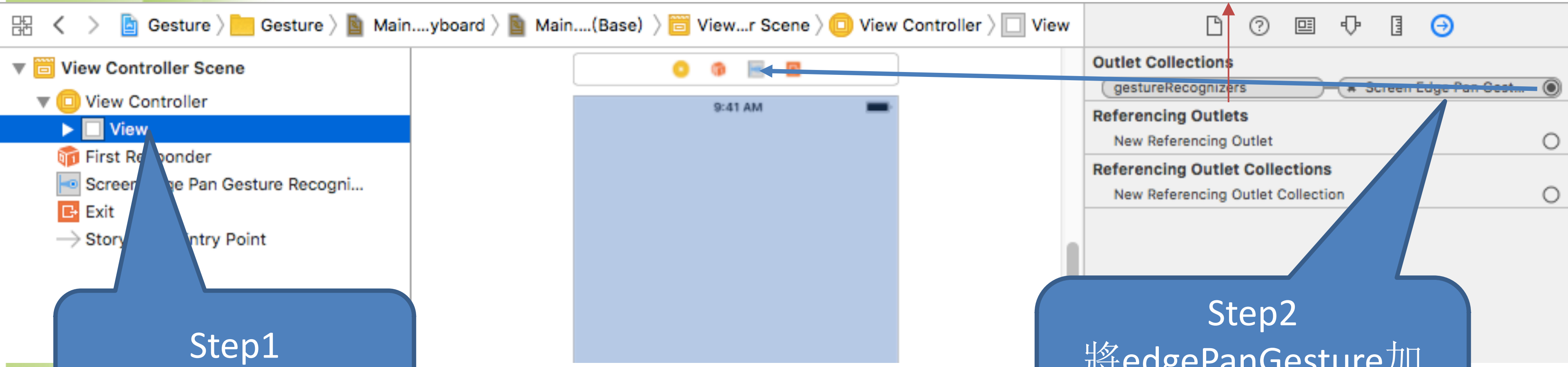
選擇螢幕的邊緣



Connection: Action
Object: View Controller
Name: handleEdgepanGesture
Type: UIScreenEdgePanGest

```
//  Gesture
//
//  Created by 張家瑋 on 2018/3/23.
//  Copyright © 2018年 張家瑋. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    @IBAction func handleEdgepanGesture(_ sender: UIScreenEdgePanGestureRecognizer) {
        for i in 0 ..< sender.numberOfTouches {
            let point = sender.location(ofTouch: i, in: sender.view)
            print("第\(i)根手指, 座標為(\(point.x), \(point.y))")
        }
    }

```

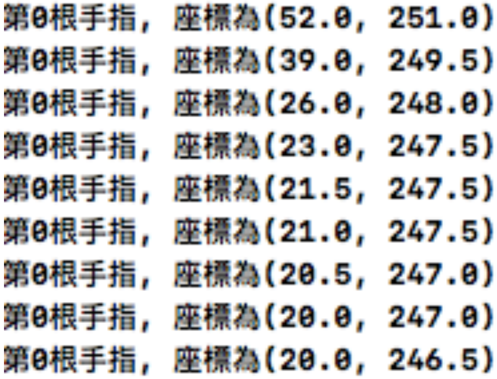
Control+drag建立handleEdgePanGesture的action gestureRecognizers



Step1
點選View

Step2
將edgePanGesture加入View的gestureRecognizer中

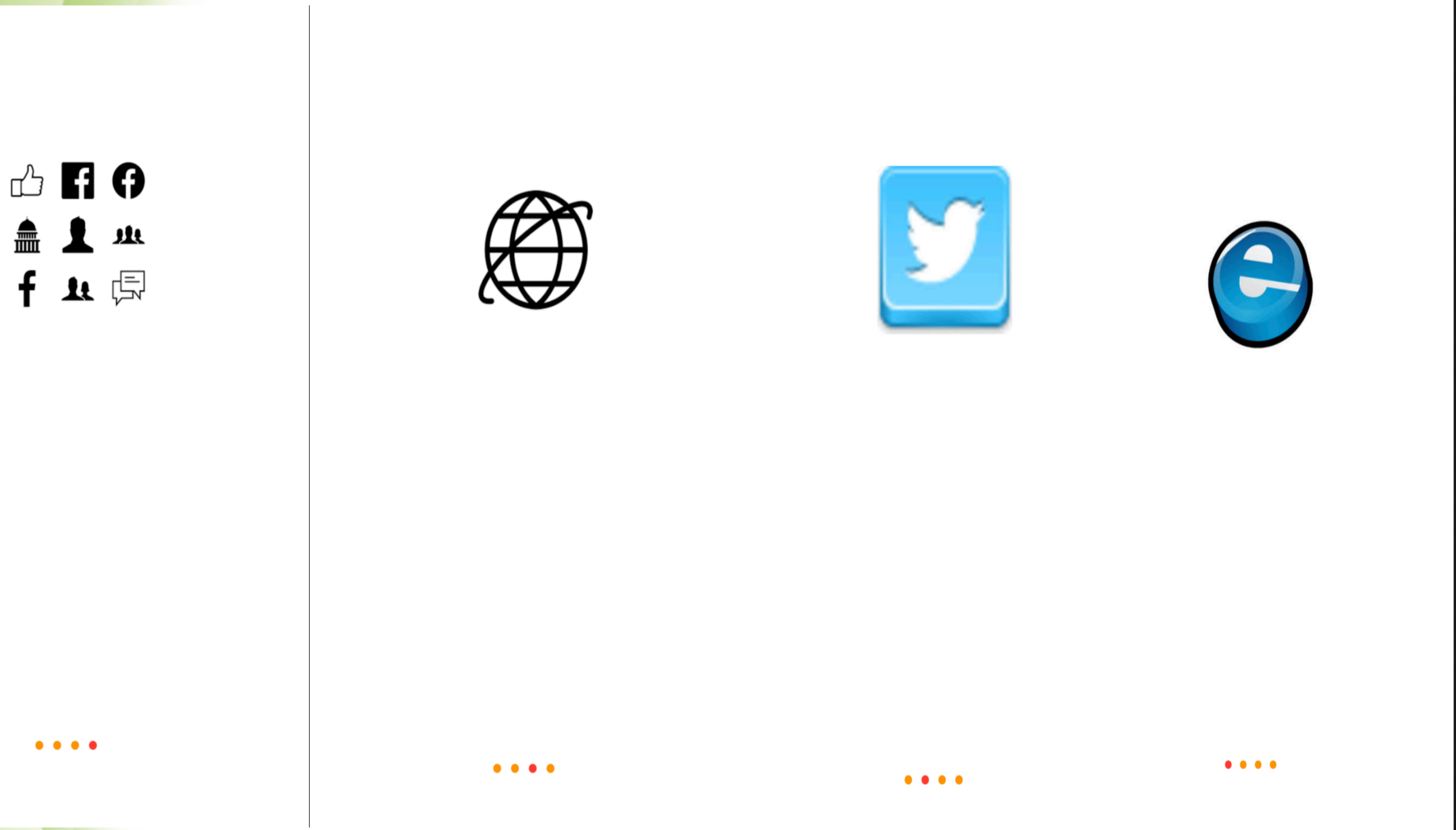
從右側邊往左滑



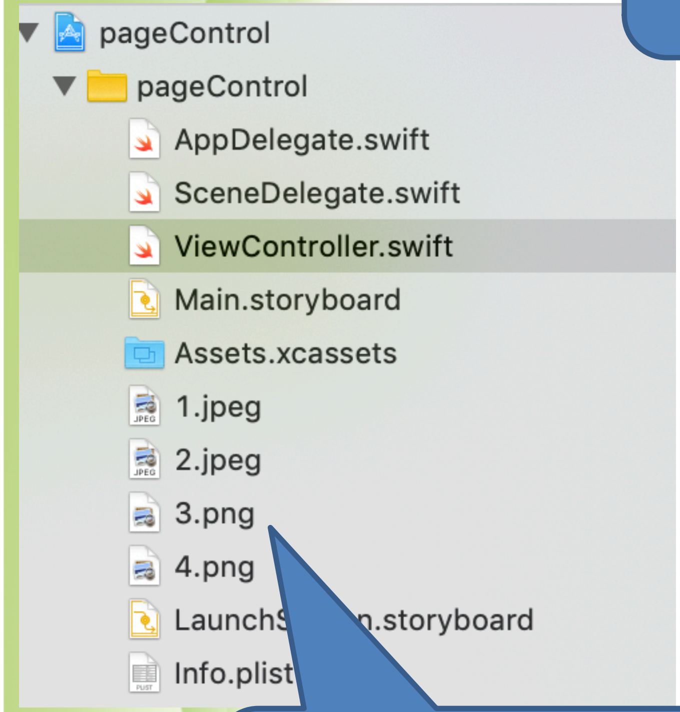
```
第0根手指, 座標為(52.0, 251.0)
第0根手指, 座標為(39.0, 249.5)
第0根手指, 座標為(26.0, 248.0)
第0根手指, 座標為(23.0, 247.5)
第0根手指, 座標為(21.5, 247.5)
第0根手指, 座標為(21.0, 247.5)
第0根手指, 座標為(20.5, 247.0)
第0根手指, 座標為(20.0, 247.0)
第0根手指, 座標為(20.0, 246.5)
```

Page Control

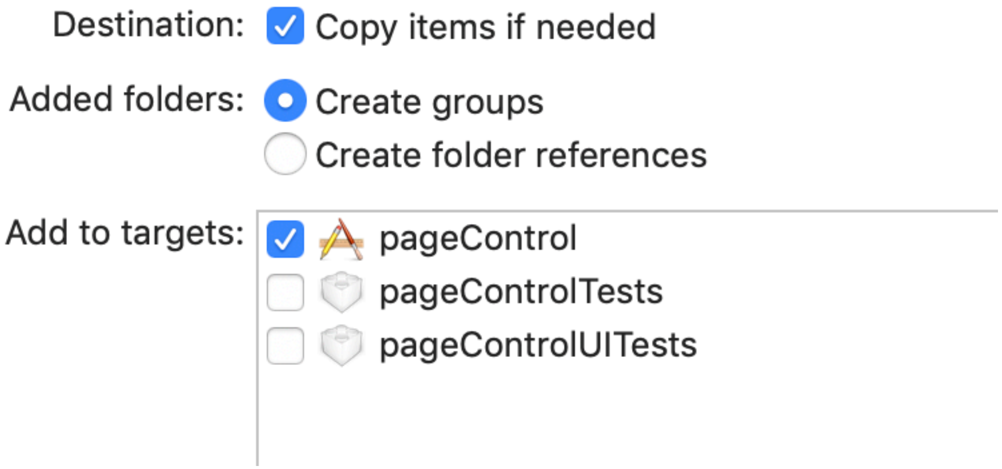
完成後可以切換四種頁面



勾選需要時複製項目



- pageControl
  - pageControl
    - AppDelegate.swift
    - SceneDelegate.swift
    - ViewController.swift
    - Main.storyboard
    - Assets.xcassets
    - 1.jpeg
    - 2.jpeg
    - 3.png
    - 4.png
    - LaunchS n.storyboard
    - Info.plist

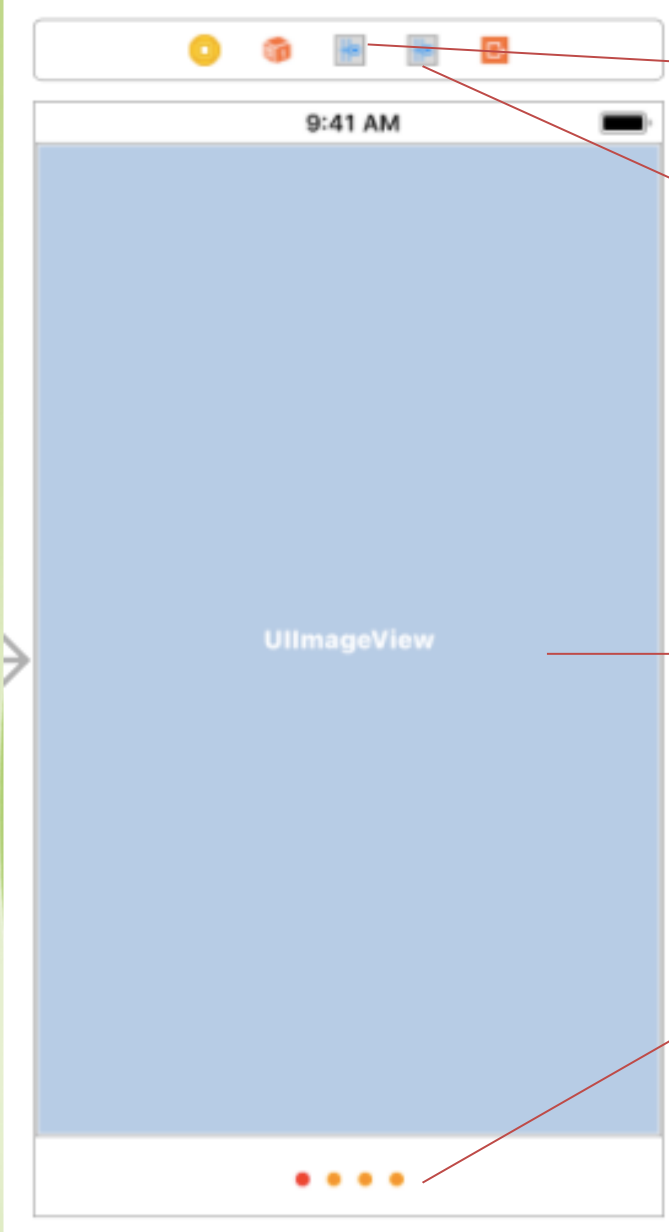


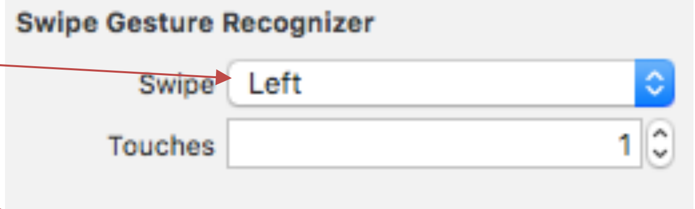
Destination: ☑ Copy items if needed

Added folders: ● Create groups
○ Create folder references

Add to targets: ☑ pageControl
□ pageControlTests
□ pageControlUITests

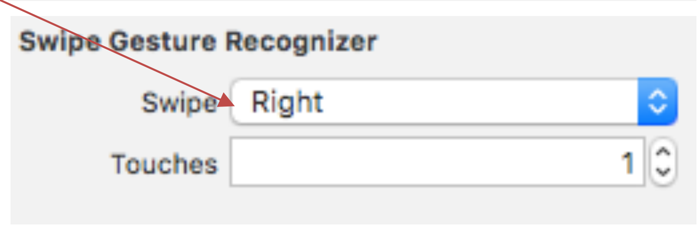
將1.jpeg 2.jpeg 3.png 4.png加入專案中




Swipe Gesture Recognizer

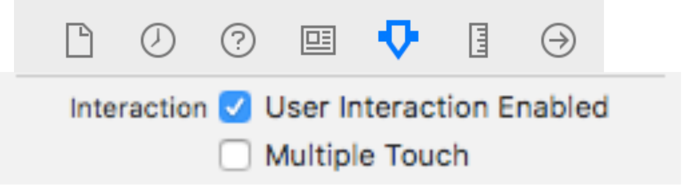
Swipe: Left

Touches: 1

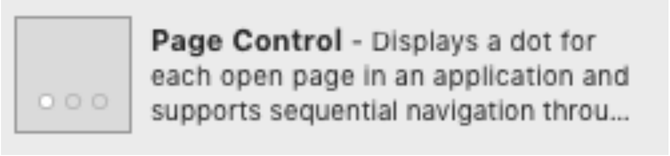

Swipe Gesture Recognizer

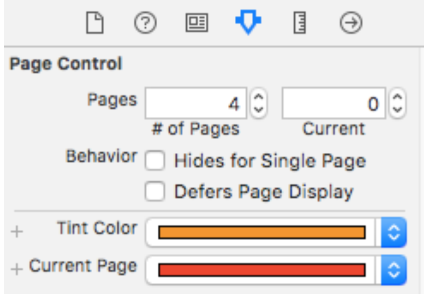
Swipe: Right

Touches: 1


Interaction ☑ User Interaction Enabled

□ Multiple Touch


Page Control - Displays a dot for each open page in an application and supports sequential navigation throu...


Page Control

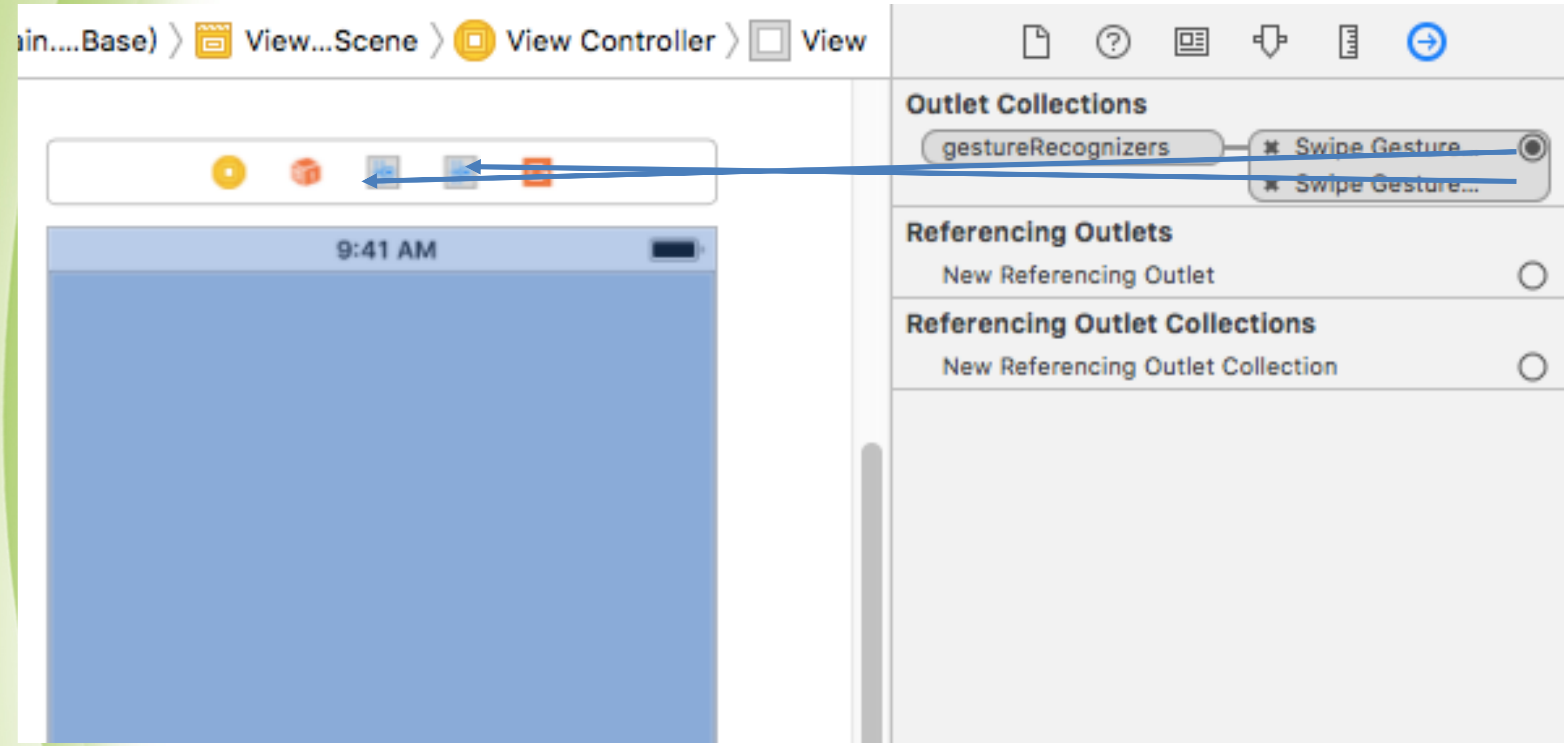
Pages: 4 (# of Pages) 0 (Current)

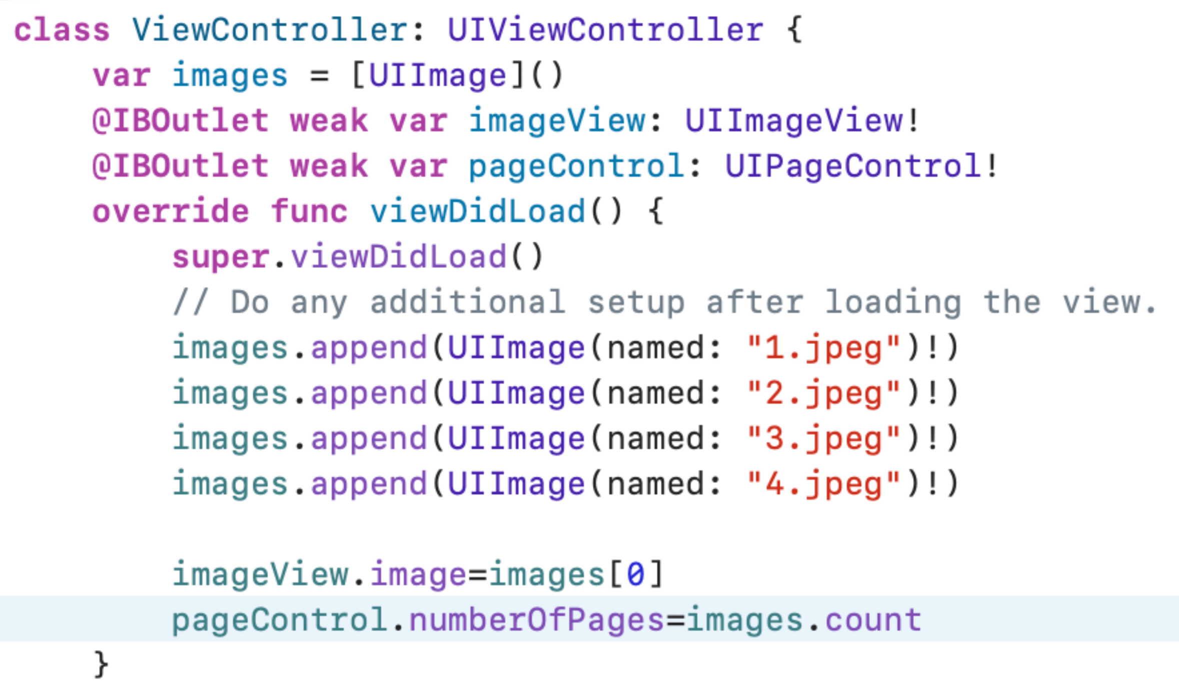
Behavior □ Hides for Single Page

□ Defers Page Display

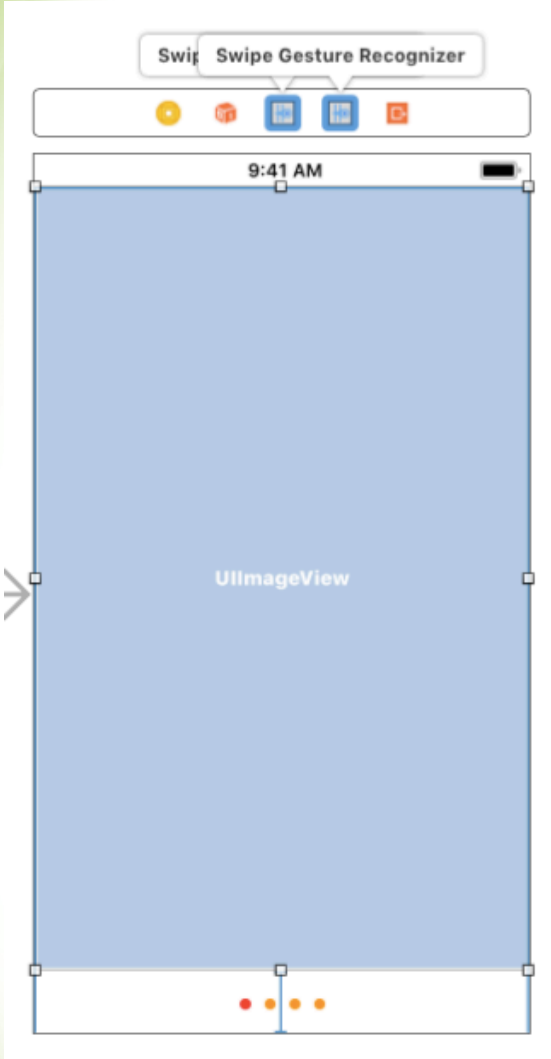
Tint Color 🟧

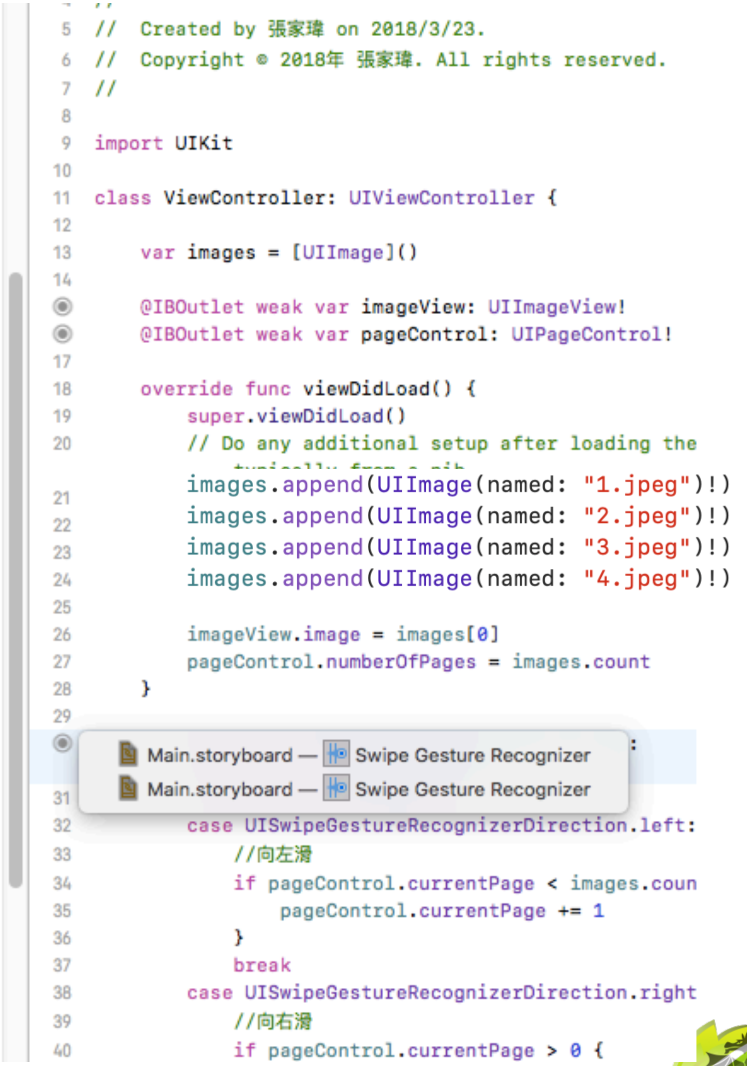
Current Page 🟥


in....Base)
View...Scene
View Controller
View
9:41 AM
Outlet Collections
gestureRecognizers
Swipe Gesture...
Swipe Gesture...
Referencing Outlets
New Referencing Outlet ○
Referencing Outlet Collections
New Referencing Outlet Collection ○

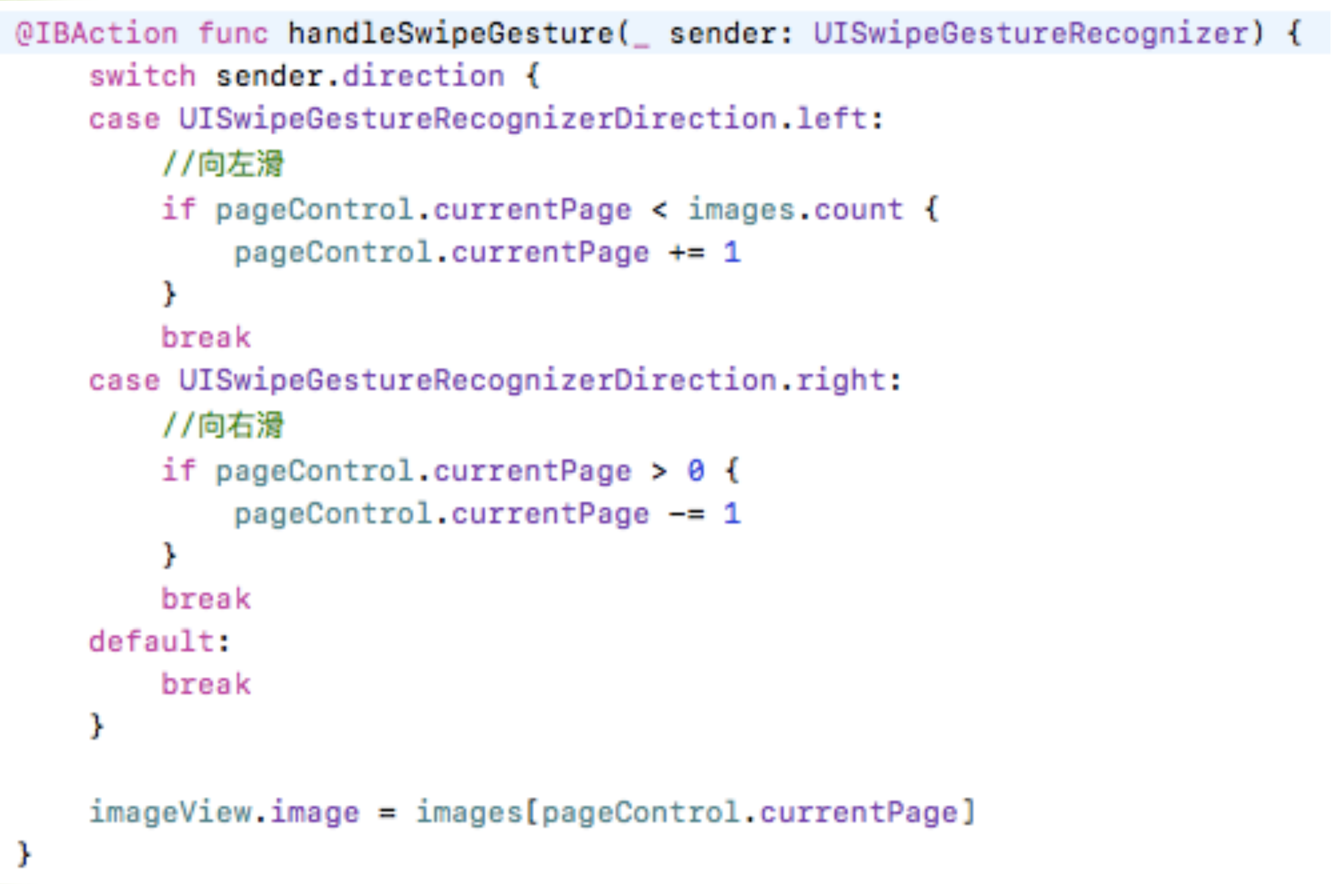


```swift
class ViewController: UIViewController {
    var images = [UIImage]()
    @IBOutlet weak var imageView: UIImageView!
    @IBOutlet weak var pageControl: UIPageControl!
    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
        images.append(UIImage(named: "1.jpeg")!)
        images.append(UIImage(named: "2.jpeg")!)
        images.append(UIImage(named: "3.jpeg")!)
        images.append(UIImage(named: "4.jpeg")!)

        imageView.image=images[0]
        pageControl.numberOfPages=images.count
    }
```




Main.storyboard — Swipe Gesture Recognizer
Main.storyboard — Swipe Gesture Recognizer

```
//
//  Created by 張家瑋 on 2018/3/23.
//  Copyright © 2018年 張家瑋. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    var images = [UIImage]()

    @IBOutlet weak var imageView: UIImageView!
    @IBOutlet weak var pageControl: UIPageControl!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the
          typically from a nib.
        images.append(UIImage(named: "1.jpeg")!)
        images.append(UIImage(named: "2.jpeg")!)
        images.append(UIImage(named: "3.jpeg")!)
        images.append(UIImage(named: "4.jpeg")!)

        imageView.image = images[0]
        pageControl.numberOfPages = images.count
    }

                                                  :

        case UISwipeGestureRecognizerDirection.left:
            //向左滑
            if pageControl.currentPage < images.coun
                pageControl.currentPage += 1
            }
            break
        case UISwipeGestureRecognizerDirection.right
            //向右滑
            if pageControl.currentPage > 0 {
```



```
@IBAction func handleSwipeGesture(_ sender: UISwipeGestureRecognizer) {
    switch sender.direction {
    case UISwipeGestureRecognizerDirection.left:
        //向左滑
        if pageControl.currentPage < images.count {
            pageControl.currentPage += 1
        }
        break
    case UISwipeGestureRecognizerDirection.right:
        //向右滑
        if pageControl.currentPage > 0 {
            pageControl.currentPage -= 1
        }
        break
    default:
        break
    }

    imageView.image = images[pageControl.currentPage]
}
```