

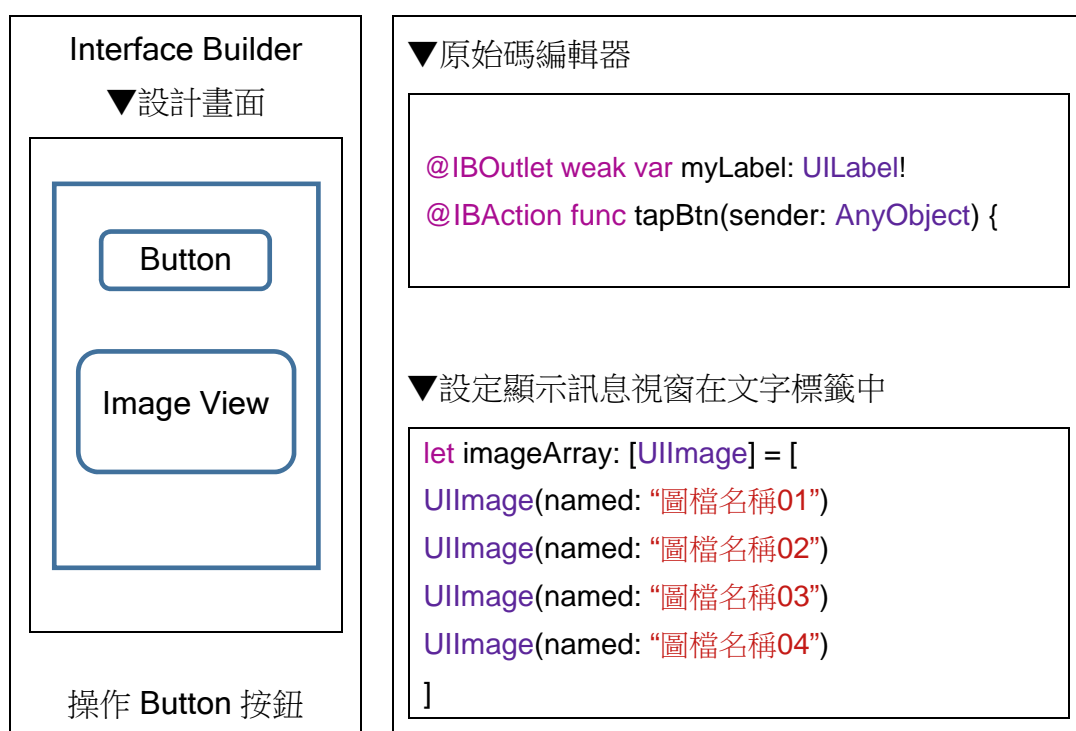
CHAPTER 6-2

UIImage View：圖像方塊顯示動畫效果

利用圖像方塊產生動畫功能

學習概念：

1. 首先用 IB 建立【圖像方塊】。
2. 將【操作按鈕】及【圖像方塊】與【程式碼】連結。
3. 最後在實作檔中相關程式，於處理載入後所觸發的事件，也就是撰寫利用【操作按鈕】結合【圖像方塊】，產生動畫顯示在【設計畫面】上的程式。

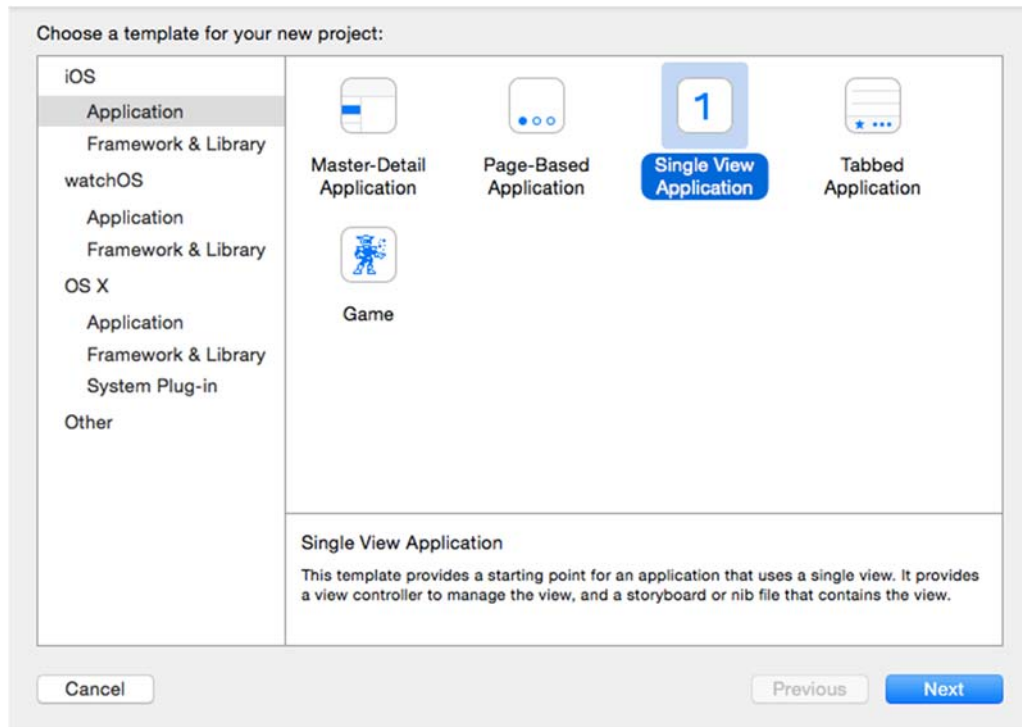


【執行結果】

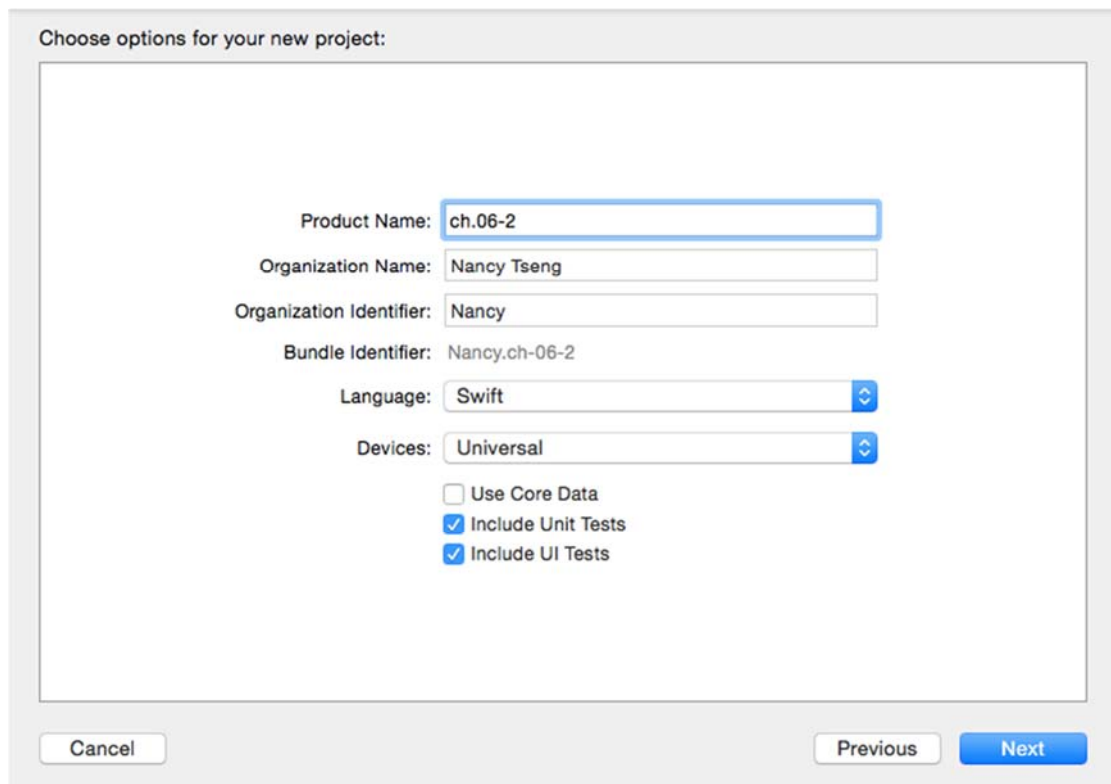
當 App 執行後，點選【操作按鈕】後，讓【圖像方塊】產生動畫的效果，例如：圖檔播放的順序、播放停頓的時間、重複播放的次數、開始播放以及停止播放動畫等，顯示在〈設計畫面〉中。

Step.1

開啟 **xcode** 時會出現的畫面，點選 **iOS** 下的【**Application**】，接著右視窗選擇【**Single View Application**】，點選【**Next**】選項後進入設定的基本視窗。

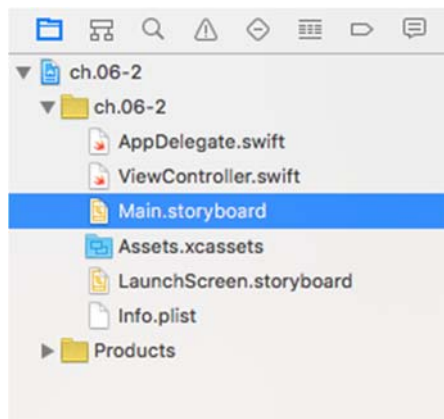


檔名及名稱設定，請將【**Product Name**】設定為 **ch.06-2**

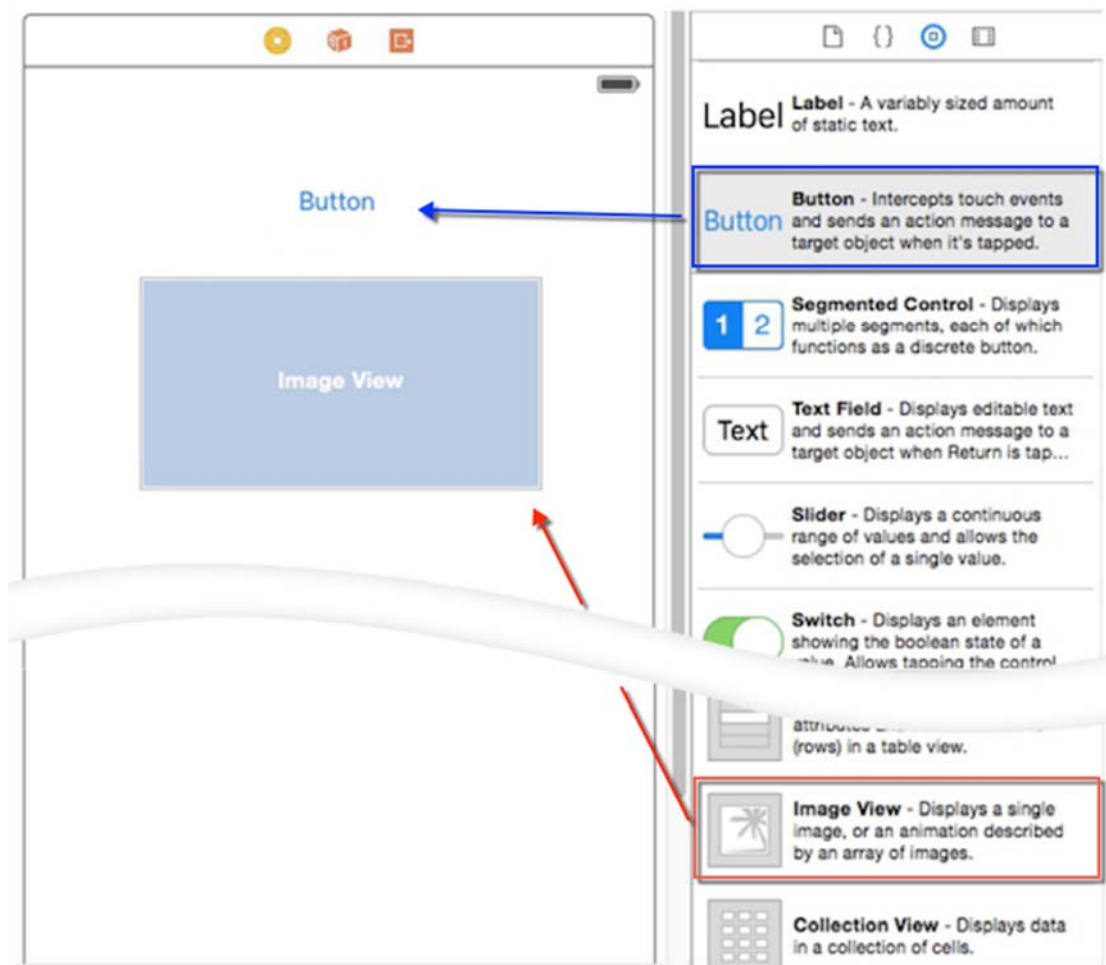


Step.2

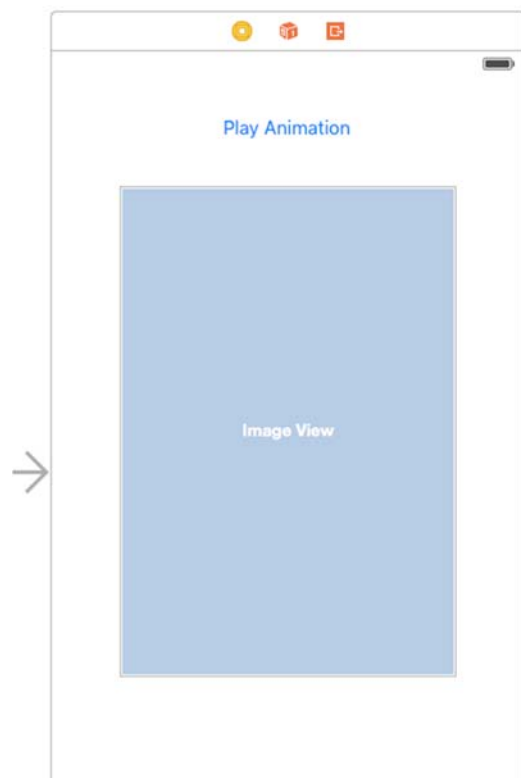
選取【Main.storyboard】



本章操作已選擇【iPhone 4.7-inch】 操作頁面。(詳見 5-1 屬性設定小技巧)
從【物件區】中拖曳【操作按鈕】「Button」及【圖像方塊】「Image View」到 IB 畫面中。



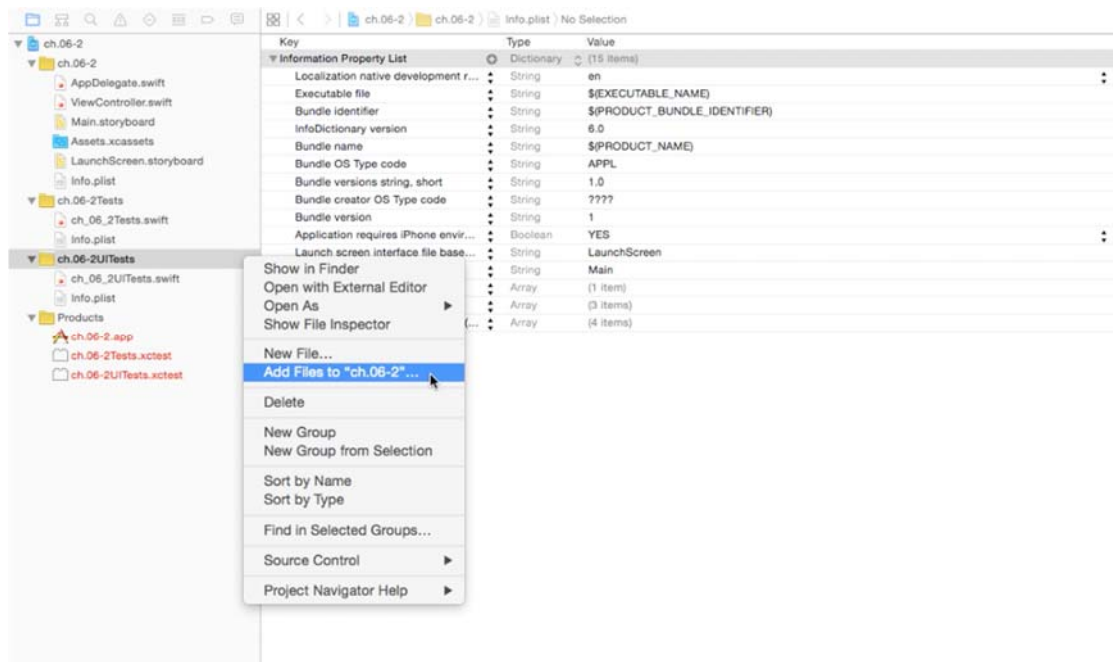
並將【操作按鈕】「Button」更改名稱命名為「Play Animation」



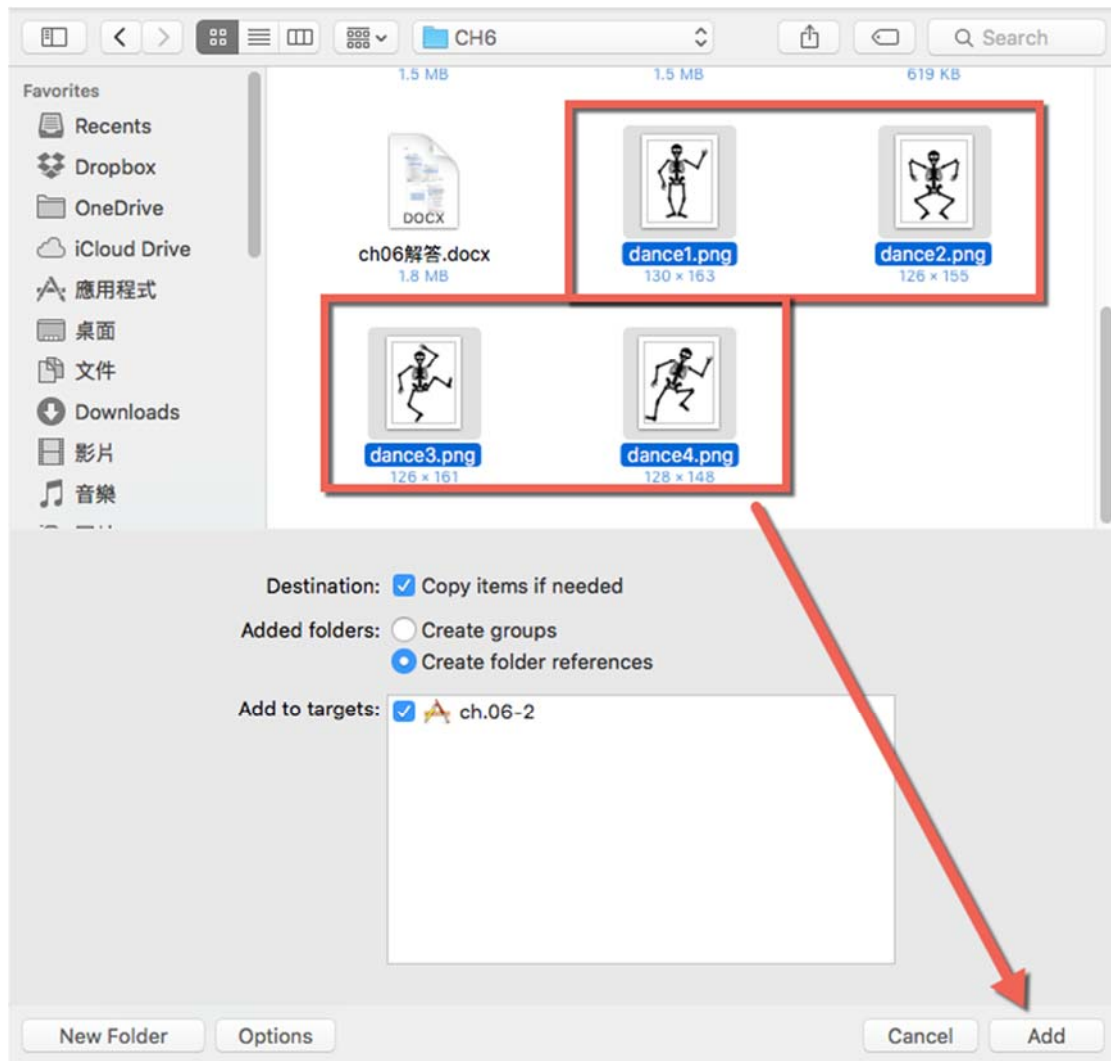
拖曳 Image View 後調整適當的大小

Step.3

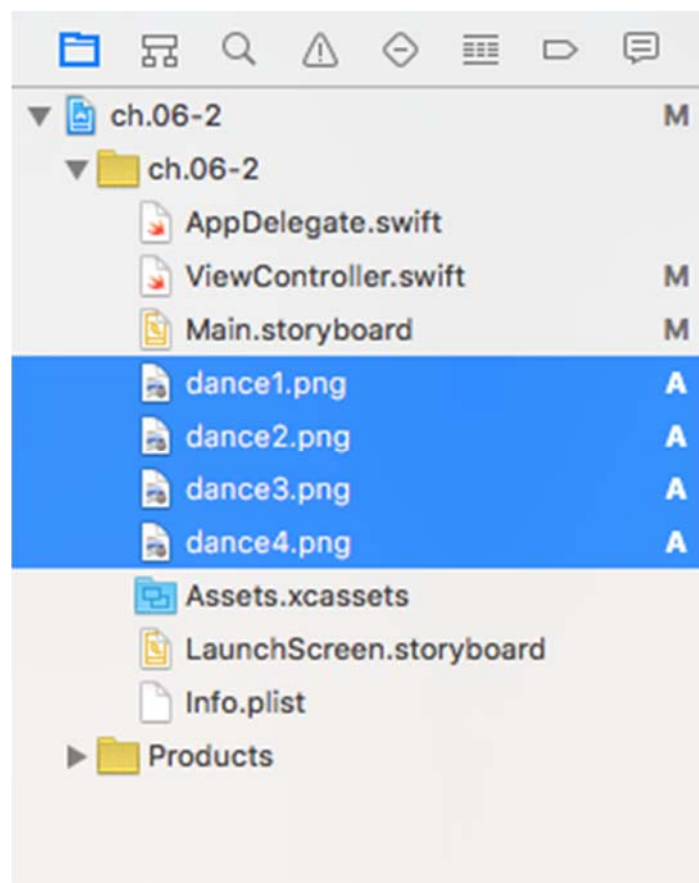
- (1) 在這裡我們選擇一張圖片，展開 ch.06-2 目錄並且在此目錄中任一位置按右鍵→Add Files to “ch.06-2....”。



(2) 在這裡我們選擇選取的檔案，名稱為 <dance01.png> <dance02.png> <dance03.png> <dance04.png>，四張圖片檔案並在選取後按「Add」。



(3) 完成後，接著會看到顯示出四個檔案，名稱分別為＜dance1.png＞＜dance2.png＞＜dance3.png＞＜dance4.png＞，顯示在 ch.06-2 目錄底下。



(4) 或者，讀者們可以利用拖曳的方式，將圖檔加入到專案內。

將要加入的图片檔案，選取後用拖曳方式加入

Step.4.

接著點選右上方工具列視窗【輔助編碼器】，就是雙圈符號【2】，進行程式碼編輯。

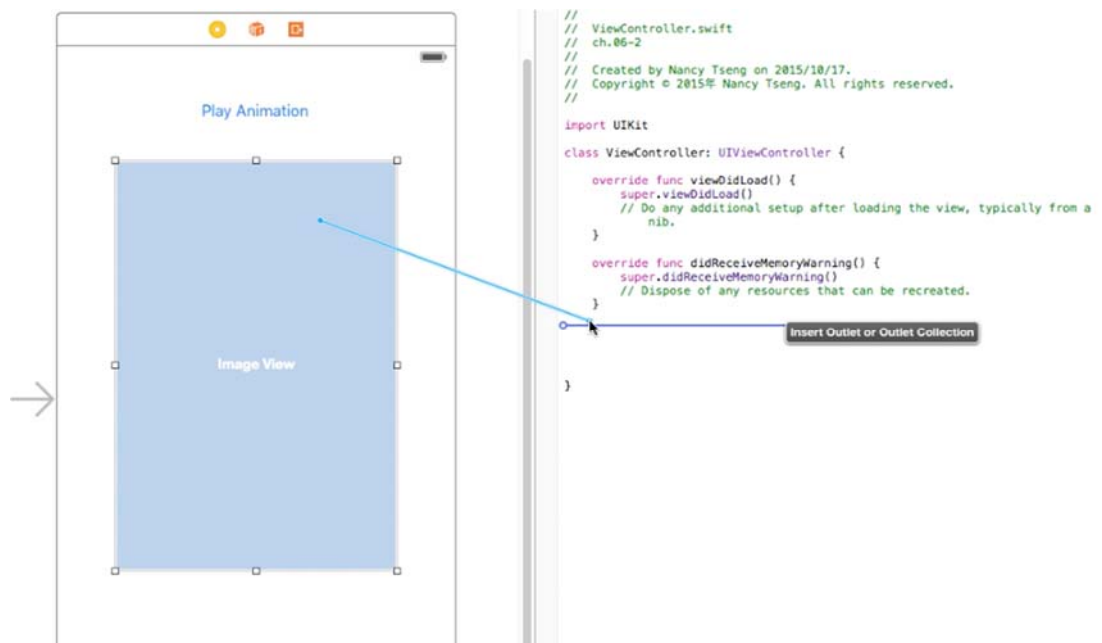


(詳見 5-1 屬性設定小技巧)

Step.5

(1) 透過連結即產生的程式碼，控制 IB 建立的元件和【程式碼】連結。

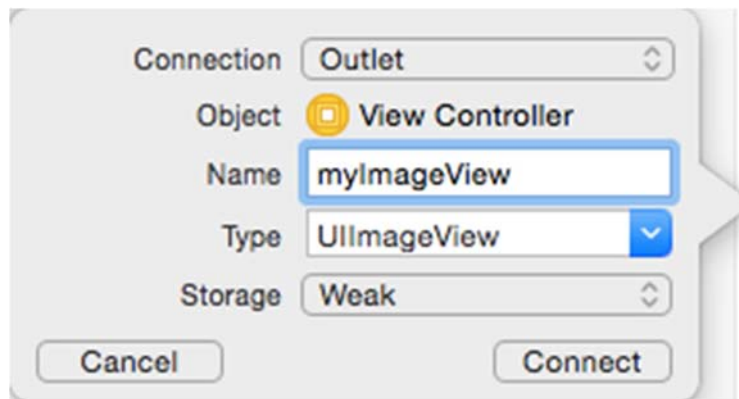
(2) 將【圖像方塊】「Image View」與「變數名稱」連結。



按住【Control】鍵用滑鼠拖曳 Image View 元件。

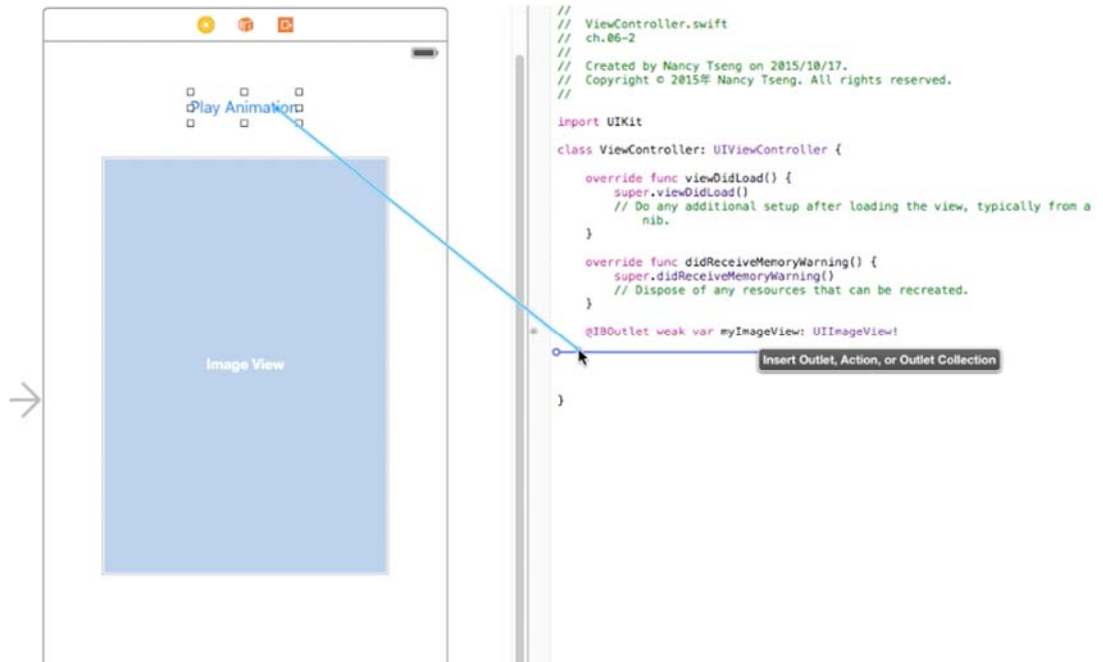
在「Connection」欄位點選【Outlet】、「Name」欄位設定名稱【myImageView】以及在「Type」欄位點選【UIImageView】後，按

【Connect】按鈕。



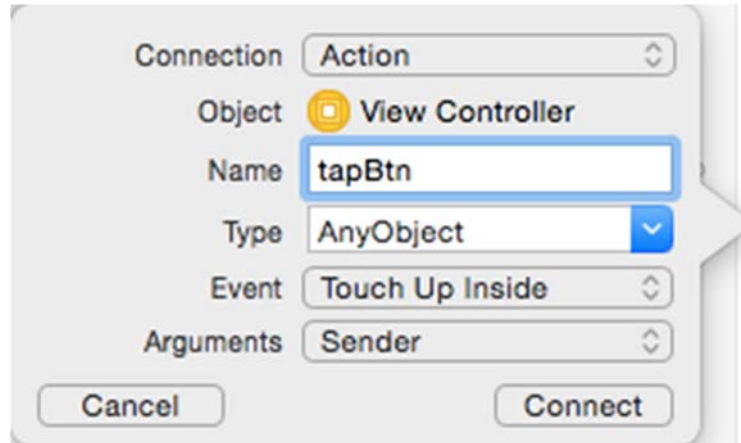
在拖曳後自動彈跳的視窗

(3) 將【操作按鈕】「Button」與「變數名稱」連結。



按住【Control】鍵用滑鼠拖曳 Button 元件。

在「Connection」欄位點選【Action】、「Name」欄位設定名稱【tapBtn】以及在「Type」欄位點選【AnyObject】後，按【Connect】按鈕。



在拖曳後自動彈跳的視窗

(4) 將會自動插入程式碼，作為與 IB 的連結。

```
import UIKit

class ViewController: UIViewController {

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

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

    @IBOutlet weak var myImageView: UIImageView!

    @IBAction func tapBtn(sender: AnyObject) {
    }

}
```

連結後出現的【程式碼】。

Step.6

(1) 接下來，我們在【紅框】中加入設定 **swift** 的程式碼，將透過串接執行程式後，將結果顯示在【圖像方塊】「Image View」中。

宣告一個陣列 **imageArray** 來儲存這四張圖。

```
import UIKit

class ViewController: UIViewController {

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

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

    @IBOutlet weak var myImageView: UIImageView!

    @IBAction func tapBtn(sender: AnyObject) {
        let imageArray: [UIImage] = [
            UIImage(named: "dance1.png")!,
            UIImage(named: "dance2.png")!,
            UIImage(named: "dance3.png")!,
            UIImage(named: "dance4.png")!
        ]

        myImageView.animationImages = imageArray
        myImageView.animationDuration = 1.0
        myImageView.animationRepeatCount = 3
        myImageView.startAnimating()
    }
}
```



屬性設定小技巧

Swift 語言教學：

設定顯示 Image View 呈現效果

1. 設定顯示動畫陣列

每個圖檔皆以 UIImage 的 named 建立屬性加入陣列，意即依照給定的順序來播放圖檔的，產生動畫的效果。在這裡我們給定陣列名稱為 imageArray。

範例：圖像方塊.animationImages = imageArray;

2. 設定顯示動畫停格速度

範例：圖像方塊.animationDuration = 秒數

3. 設定顯示播放動畫次數

範例：圖像方塊.animationRepeatCount = 重覆播放的次數
(備註：如果要不斷重覆的播放，則屬性設定為 0)

4. 設定「啟動」動畫播放

範例：圖像方塊.startAnimating()

5. 設定「停止」動畫播放

範例：圖像方塊.stopAnimating()

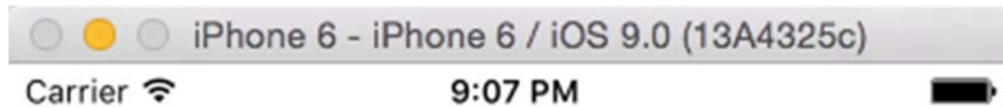
Step.7

在上方工具列按下【執行鍵▶】(Build and then run the current scheme)，啟動模擬器執行程式。



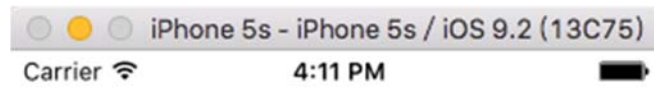
Step.8

當 App 啟動後在顯示畫面時，在點選【操作按鈕】「Button」的同時，將會動畫結果設定的內容，顯示在【圖像方塊】也就是「Image View」中的圖片內容。



Play Animation

試試按一下【操作按鈕】「Button」後會出現的結果～



[Play Animation](#)

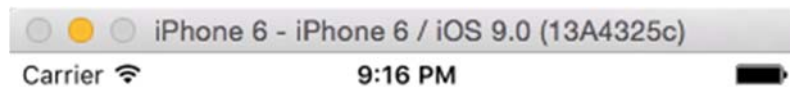


在我們點選【操作按鈕】時，則顯示畫面會將圖像透過所要呈現的樣態來顯示。

自我練習

實作執行後結果：

這次呢，我們要練習的是運用【Button】及【Image View】與程式結合執行，透過【Button】指令選項讓【Image View】根據我們所撰寫的內容；在執行與【程式碼】連結並串接時，當執行【RUN】後，透過點選【操作按鈕】會「啟動」以及「停止」動畫播放圖像的內容視窗。



Button



當點選【Button】啟動播放動畫，再點選一下【Button】停止播放動畫