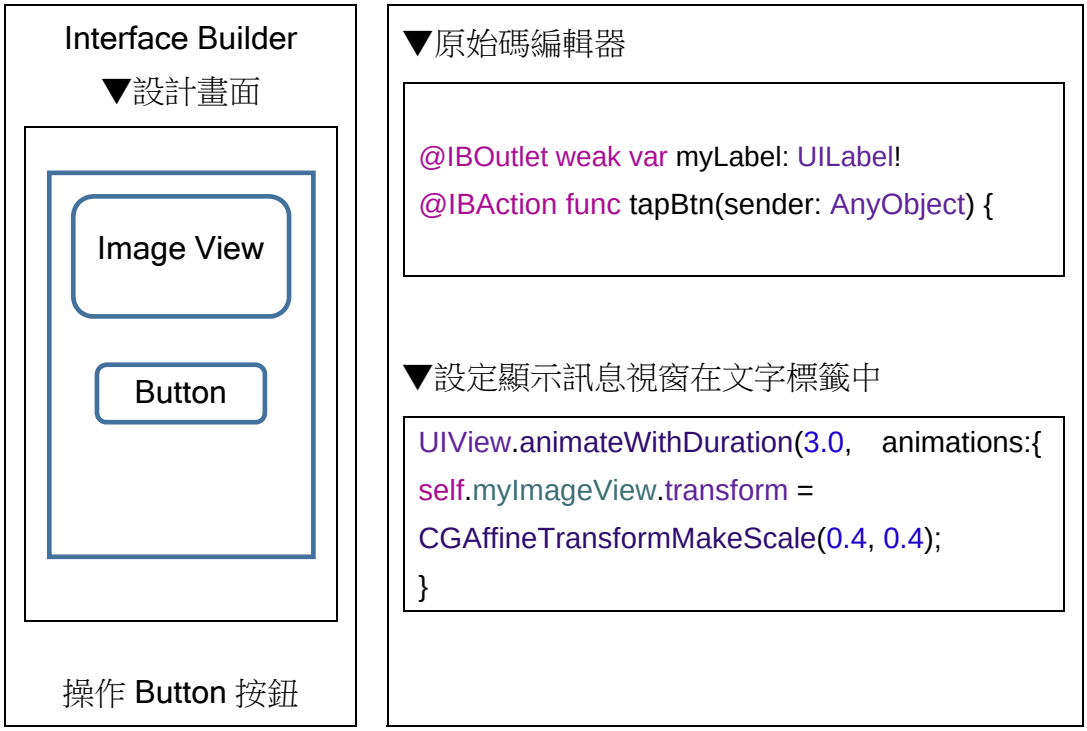
CHAPTER 6-3

UIView：動畫效果之圖像方塊

利用圖像方塊產生縮放、移動功能

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。

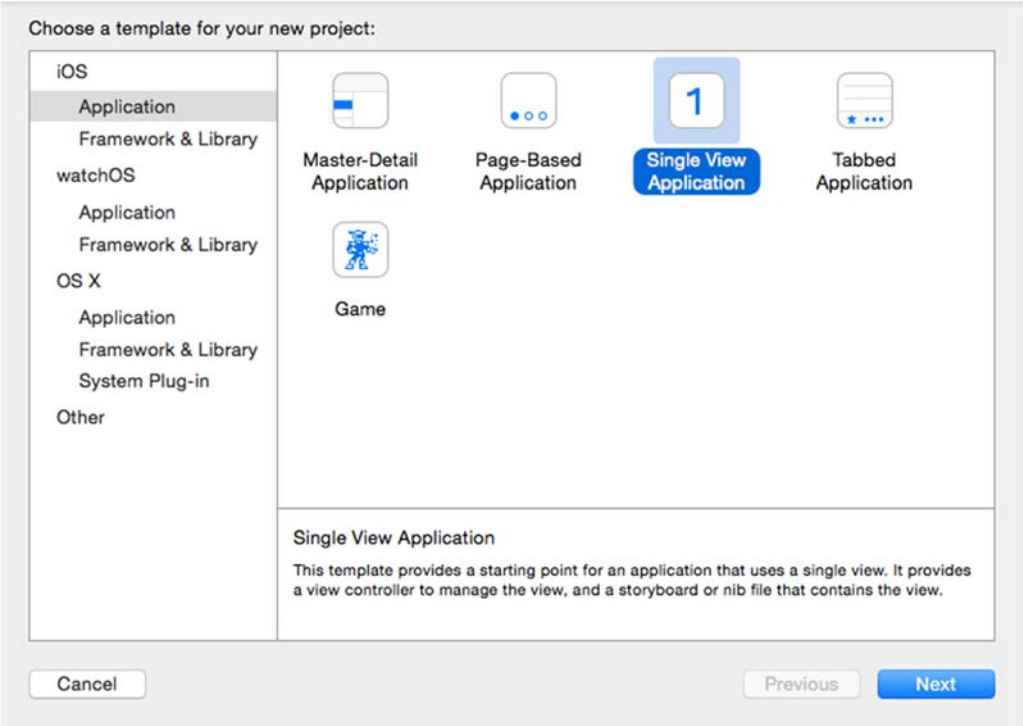


【執行結果】

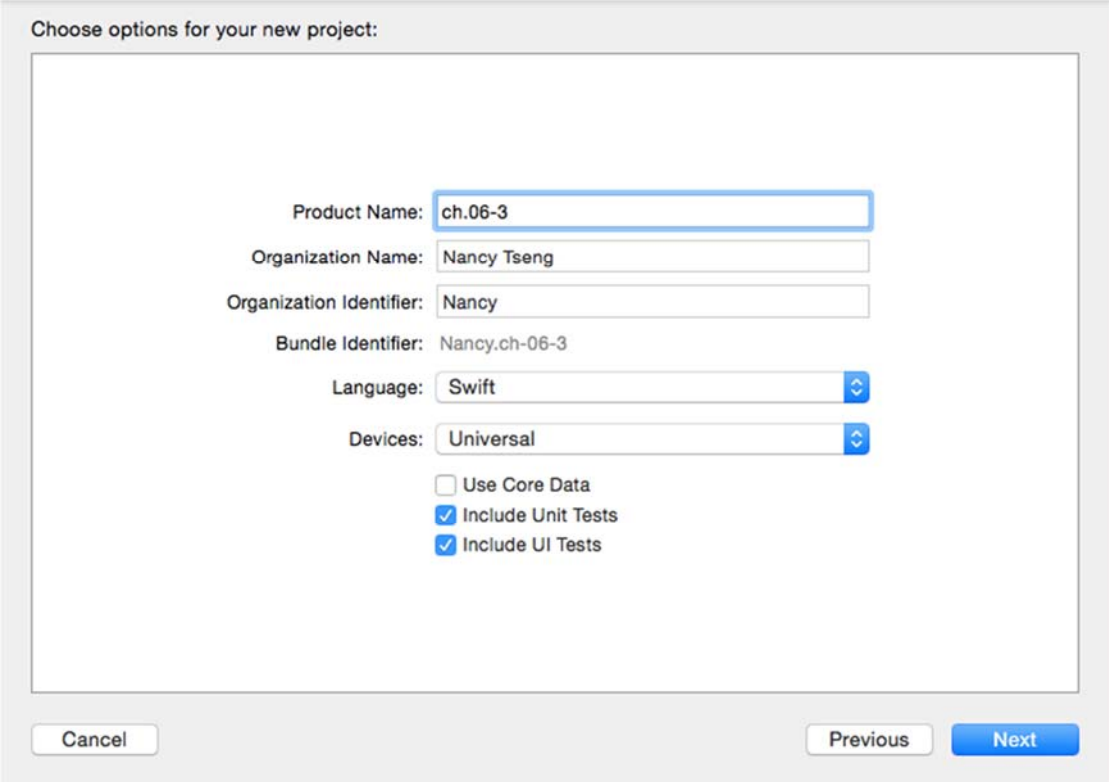
當 App 執行後，點選【操作按鈕】後，讓【圖像方塊】產生縮放的效果，例如：圖檔播放的速度、播放停頓的時間、重複播放的次數、開始播放以及停止播放動畫等，顯示在〈設計畫面〉中。

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。

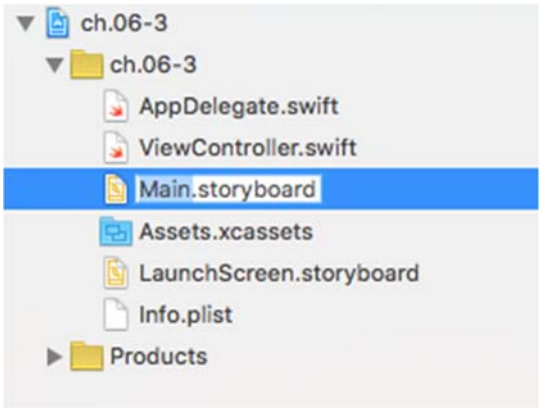


檔名及名稱設定，請將【Product Name】設定為 ch.06-3

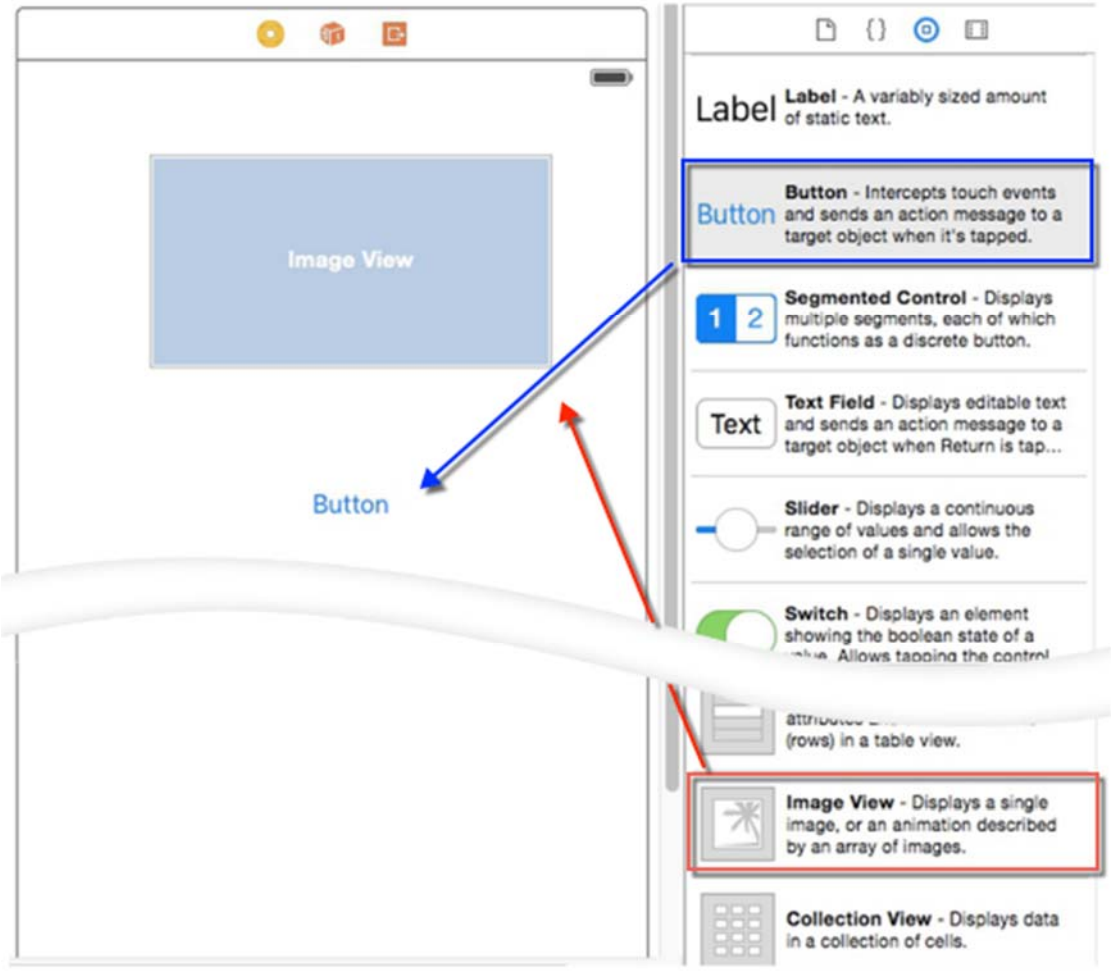


Step.2

選取【Main.storboard】

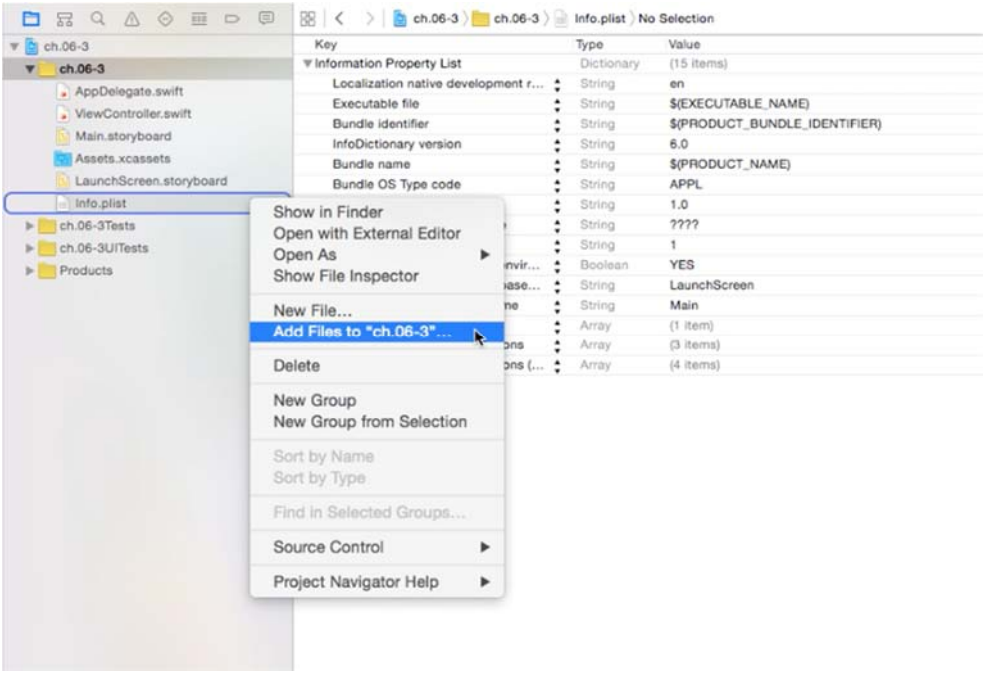


本章操作已選擇【iPhone 4.7-inch】 操作頁面。(詳見 5-1 屬性設定小技巧)
從【物件區】中拖曳【操作按鈕】「Button」及【圖像方塊】「Image View」到 IB 畫面中。

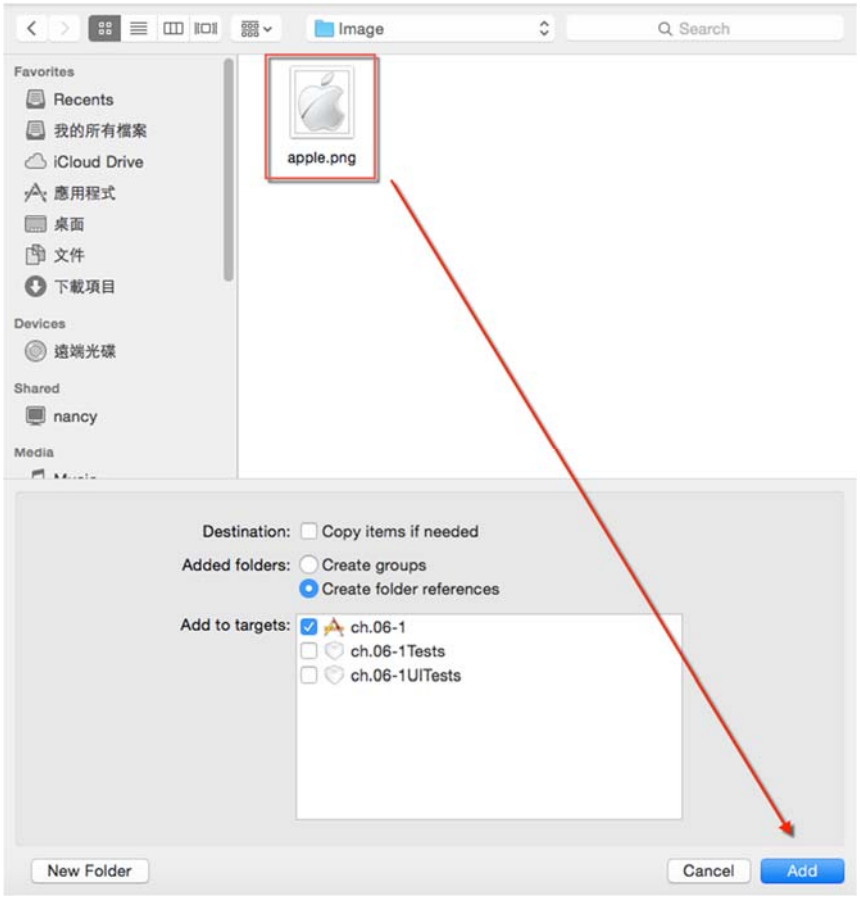


Step.3

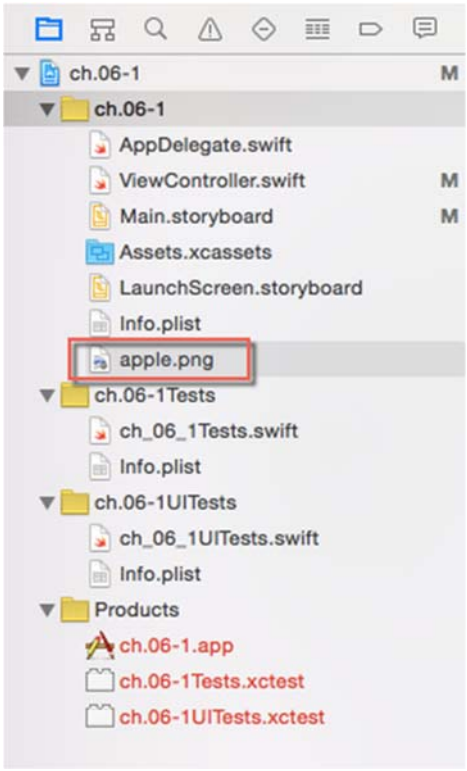
(1) 在這裡我們選擇一張圖片，展開 ch.06-3 目錄並且在 info.plint 檔名按右鍵→Add Files to “ch.06-3….”。



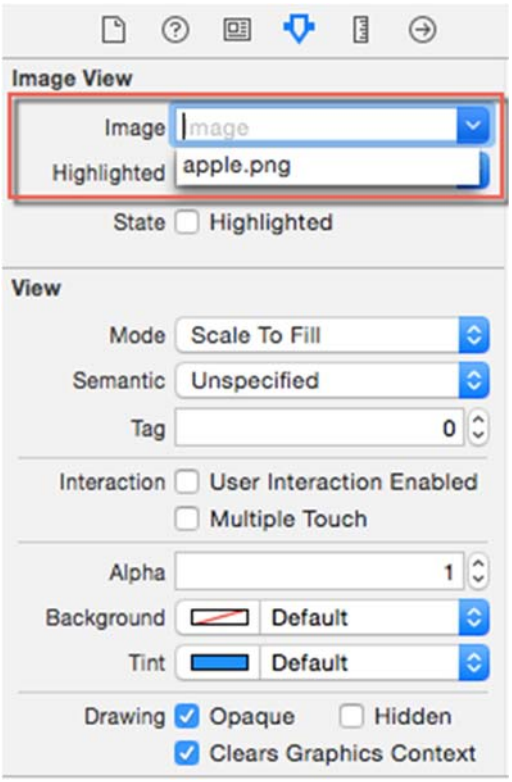
(2) 在這裡我們選擇選取的檔案名稱為＜apple.png＞選取後按「Add」。



(3)　完成後，接著會看到顯示出個檔名為＜apple.png＞。



(4)　我們在右邊「屬性檢視」的視窗，點選我們所要加入點片的檔案名稱＜apple.png＞



(5)　在經由「屬性檢視」視窗加入後，隨即會發現在【設計畫面】的視窗會顯示我們所指定的＜apple.png＞圖像。

Step.4.

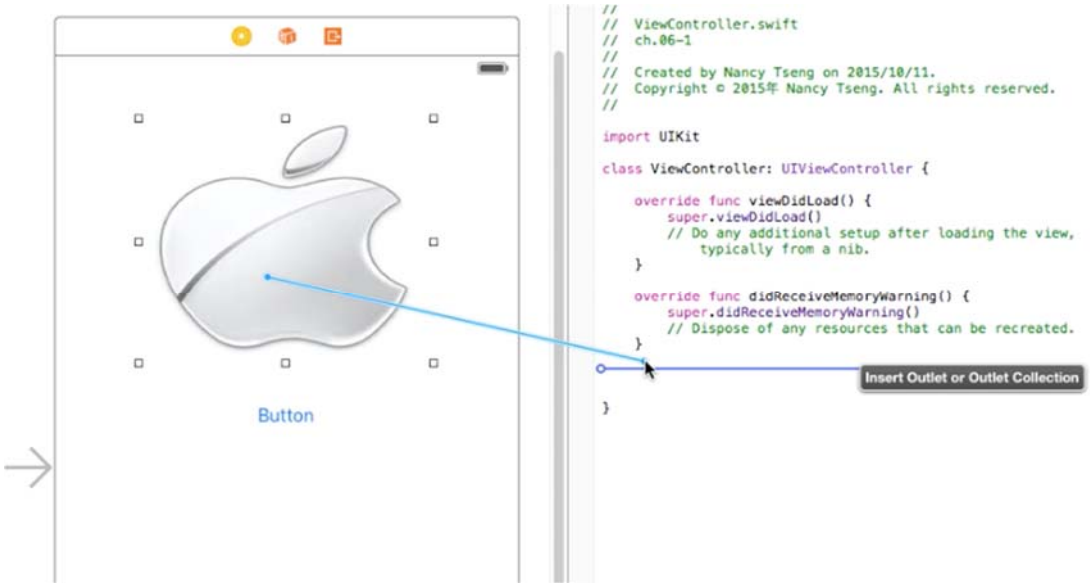
接著點選右上方工具列視窗【輔助編碼器】，就是雙圈符號【2】，進行程式碼編輯。

（詳見 5-1 屬性設定小技巧）

Step.5

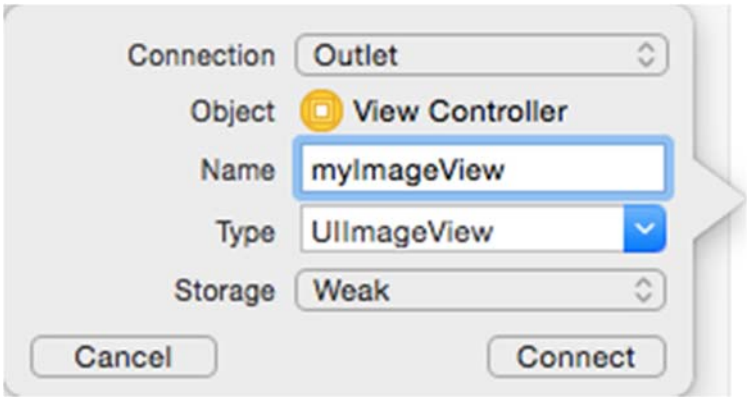
(1) 透過連結即產生的程式碼，控制 IB 建立的元件和【程式碼】連結。

(2) 將【圖像方塊】「Image View」與「變數名稱」連結。



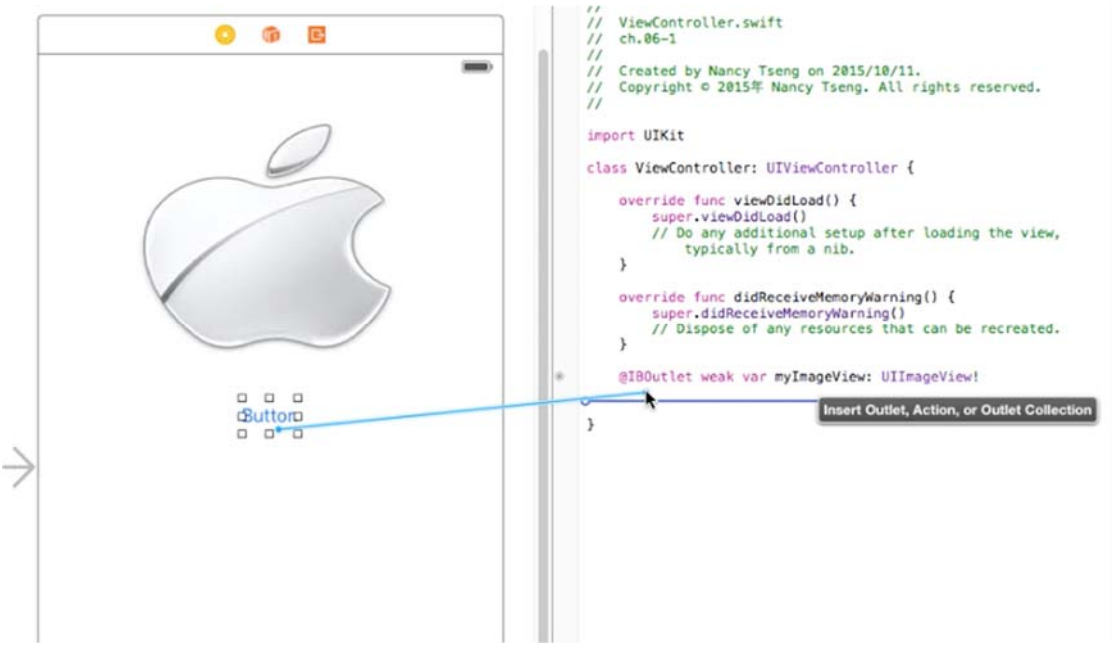
按住【Control】用滑鼠拖曳 Image View 元件。

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。



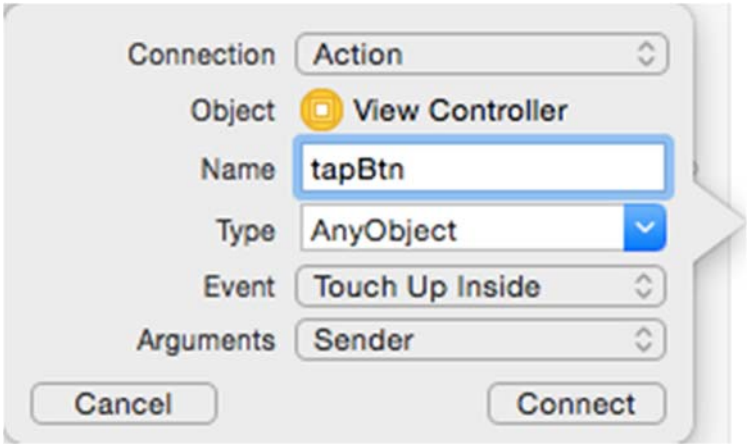
在拖曳後自動彈跳的視窗

(3) 將【操作按鈕】「Button」與「變數名稱」連結。



按住【Control】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。

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        myImageView.center = CGPointMake(160, 150);

    }

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


    @IBOutlet weak var myImageView: UIImageView!

    @IBAction func tapBtn(sender: AnyObject) {

        UIView.animateWithDuration(3.0,  animations:{
            self.myImageView.transform = CGAffineTransformMakeScale(0.4, 0.4);
        })

    }

}
```

屬性設定小技巧

Swift 語言教學：

在我們所設定的「操作按鈕」和「圖像方塊」，顯示畫面上的元件都具備著有 UIView 的特性，可以藉由設定位置、大小、透明度、旋轉的角度等。介紹以下屬性：

1. 設定動畫開始時的圖片狀態（透明度與縮放比例）

(1) **設定中心點，範例：**self.myImageView.center = CGPointMake(x,y);

(2) **設定透明度，範例：**self.myImageView.alpha = **透明度**;

備註：0.0 表示完全透明；1.0 表示完全不透明。

(3) **圖像尺寸變形及縮放**，self.myImageView.transform =

CGAffineTransformMakeScale(x 方向比例,y 方向比例);

2. 設定動畫的參數（動畫名稱、速度、開始及結束的處理）

(1) UIView.beginAnimations("animation", context:nil);
(2) UIView.setAnimationDuration(1.5);

3. 設定動畫結束的圖片狀態（透明度與縮放比例）

(1) myImageView.center = CGPointMake(x,y);

(2) myImageView.alpha =**透明度**;

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。

當在【操作按鈕】「Button」後，會出現的變化～

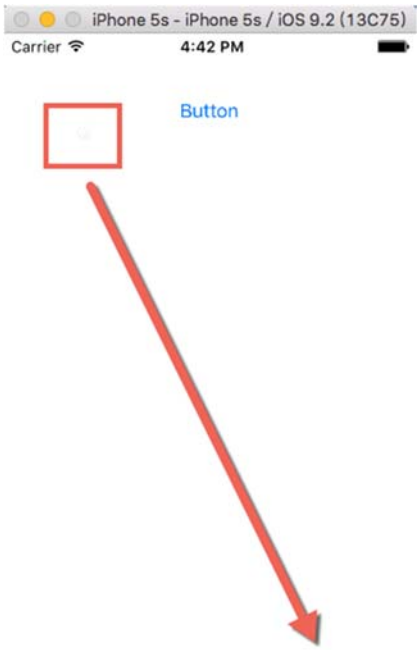
在我們點選【操作按鈕】後，
則顯示畫面會將圖像透過 swift 設定由大變小的樣態來顯示。

（小編在這裡設定為 iPhone6 畫面，供各位讀者們參考）

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「移動」動畫播放圖像的內容視窗。



當點選「Button」啟動播放動畫，畫面中的蘋果會自左上移右下。

畫面中的蘋果經點選「Button」後會自左上移右下。