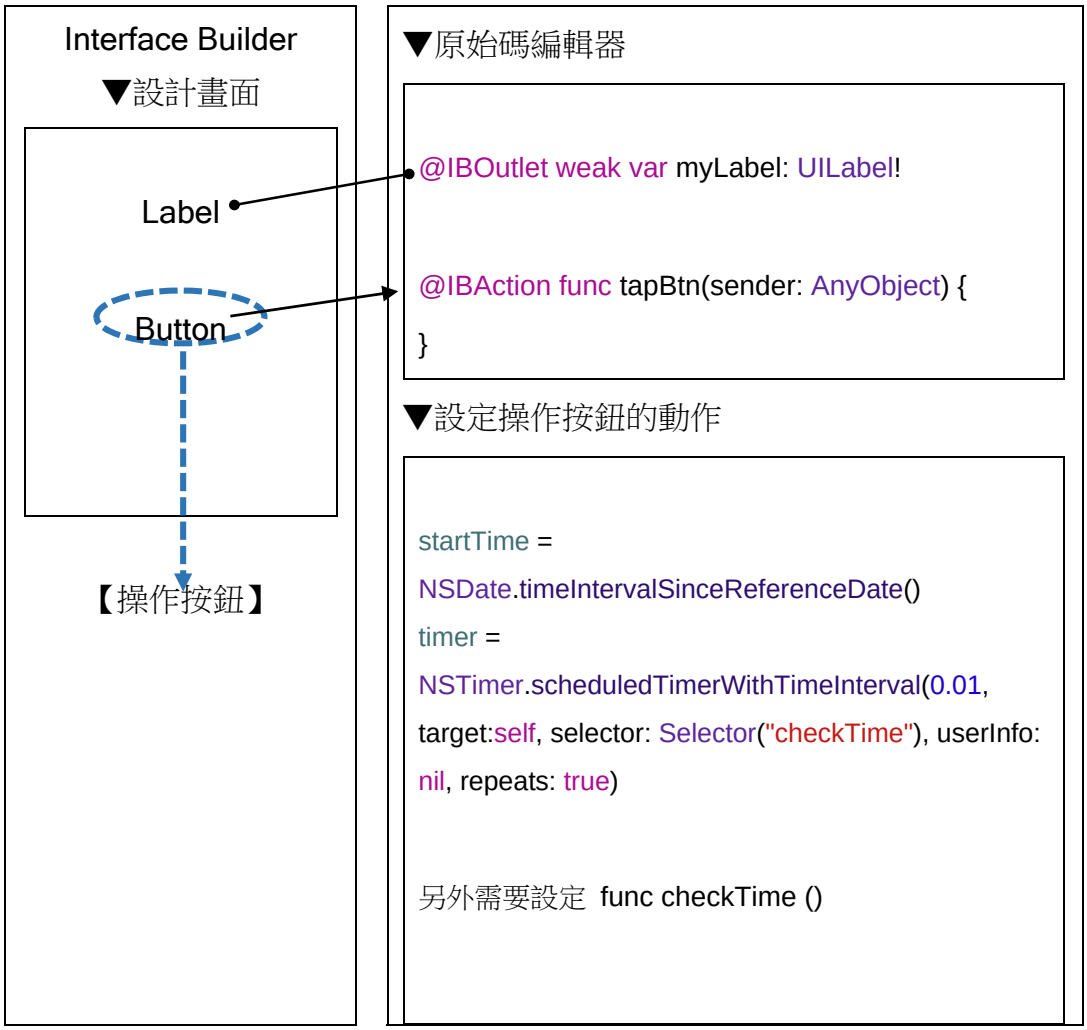
CHAPTER 6-6

NSTimer：計時器進階使用

利用計時器製作倒數計時器

學習概念：

1. 首先用 IB 建立【文字標籤】及【操作按鈕】。
2. 接著將【文字標籤】、【操作按鈕】與【程式碼連結】。
3. 最後在實作檔中實作相關程式，在處理畫面載入後所觸發的事件，也就是在撰寫按下【操作按鈕】後，可以顯示【文字標籤】上內容的程式。

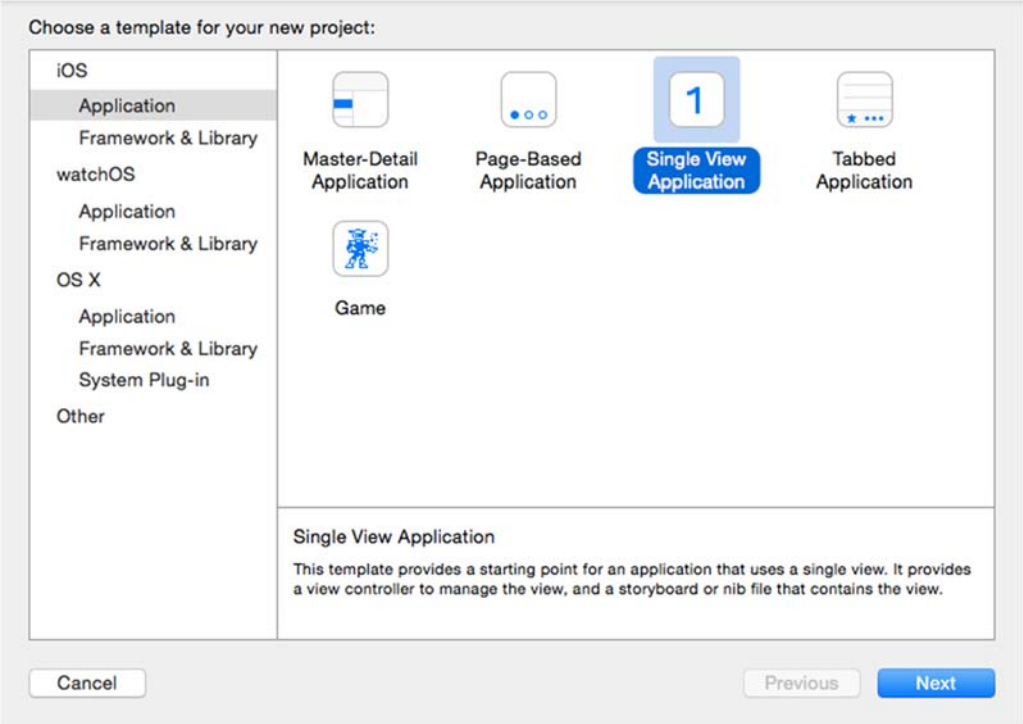


【執行結果】

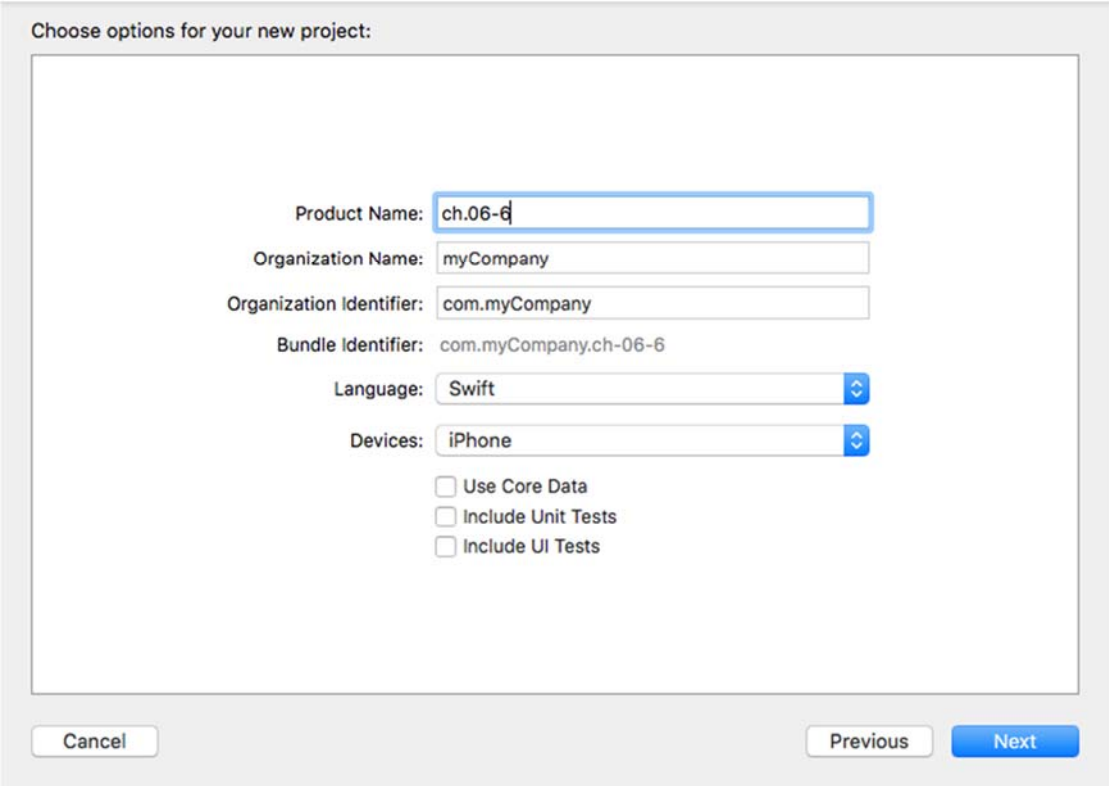
當 App 執行後，點選【操作按鈕】後，讓【文字標籤】會顯示倒數計時的時間，顯示在〈設計畫面〉中。

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。

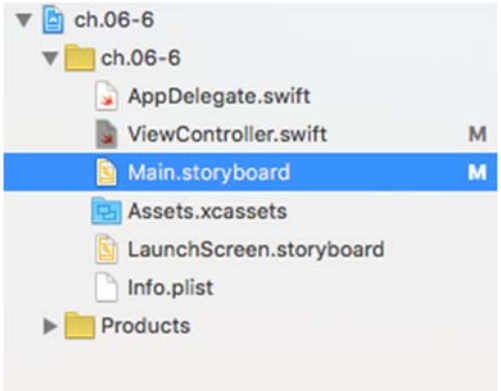


檔名及名稱設定，請將【Product Name】設定為 ch.06-6



Step.2

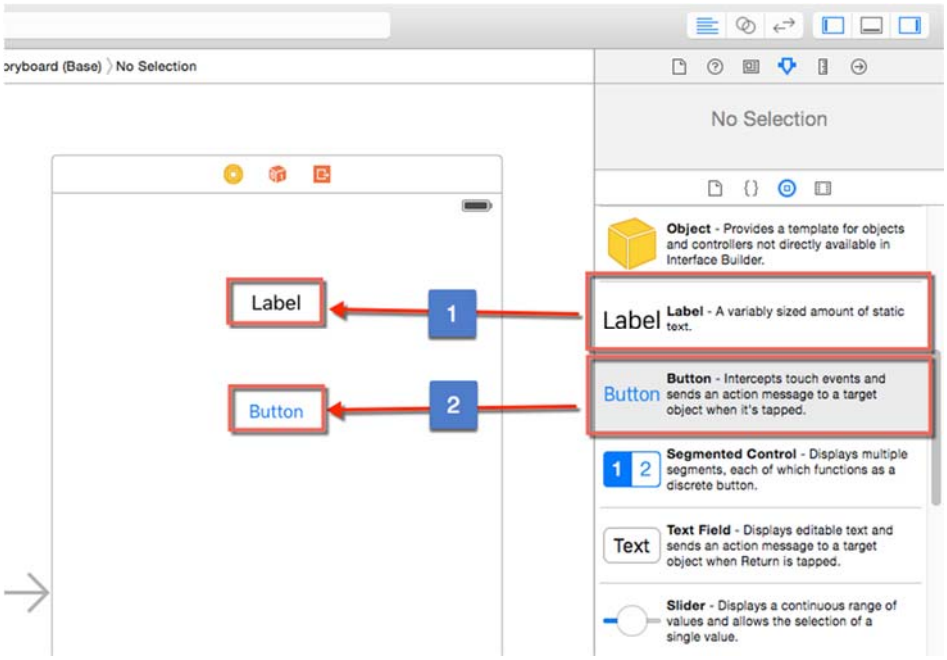
選取【Main.storboard】



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）

Step.3

從【物件庫】中拖曳【Label】【方框 1】及【Button】【方框 2】到畫面中。



Step.4.

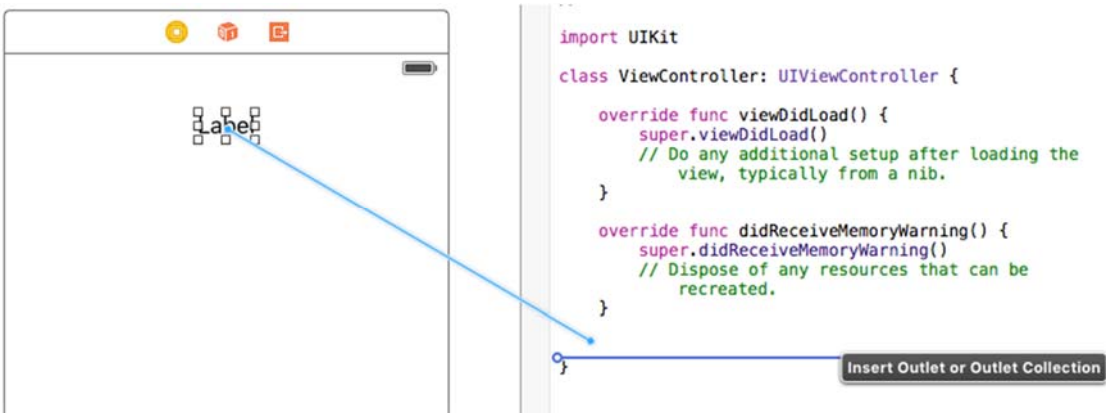
接著點選右上方工具列視窗【輔助編碼器】，就是雙圈符號【2】，進行程式碼編輯。

（詳見 5-1 屬性設定小技巧）

Step.5

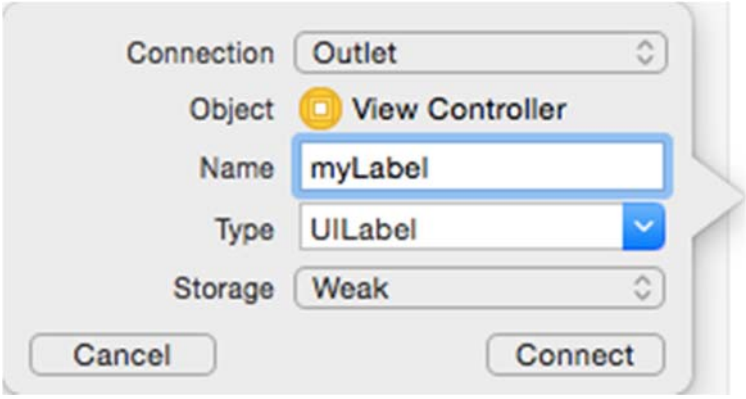
(1) 透過連結即產生的程式碼，控制 IB 建立的元件和【程式碼】連結。

(2) 將【文字標籤】「Label」與「變數名稱」連結。



按住【Control】用滑鼠拖曳 Label 元件。

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



在拖曳後自動彈跳的視窗

(3) 將會自動插入程式碼，作為與 IB 的連結。

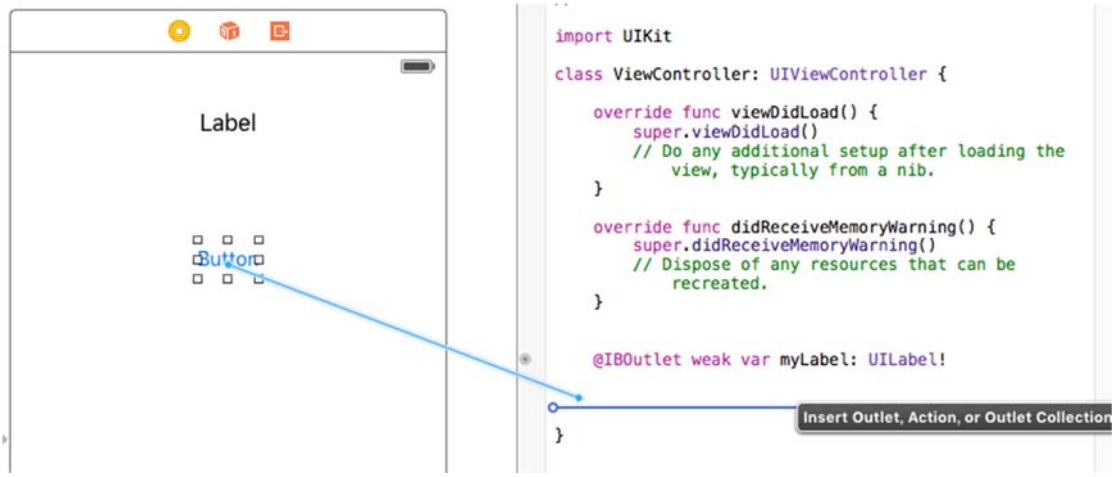
```
import UIKit

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 myLabel: UILabel!

}
```

連結後出現的程式碼。

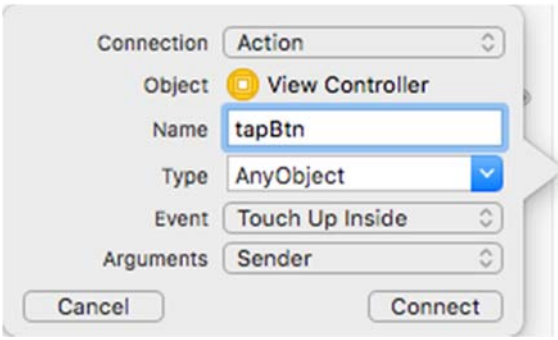
Step.6

(1) 將【操作按鈕】和【程式碼】連結。

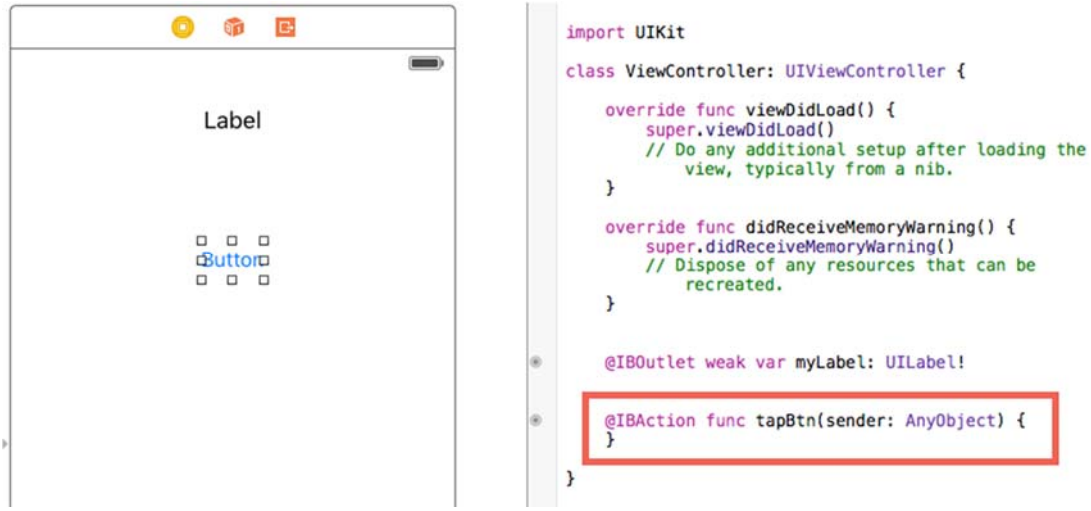
(2) 在 IB 視窗上點選【操作按鈕】，接著按住【contral】鍵，用滑鼠拖曳【操作按鈕】到右邊視窗與【程式碼】連結。



(3) 在「Connection」欄位點選【Action】、「Name」欄位設定名稱【tapBtu】以及在「Event」欄位點選【Touch Up Inside】後，按【Connect】按鈕。

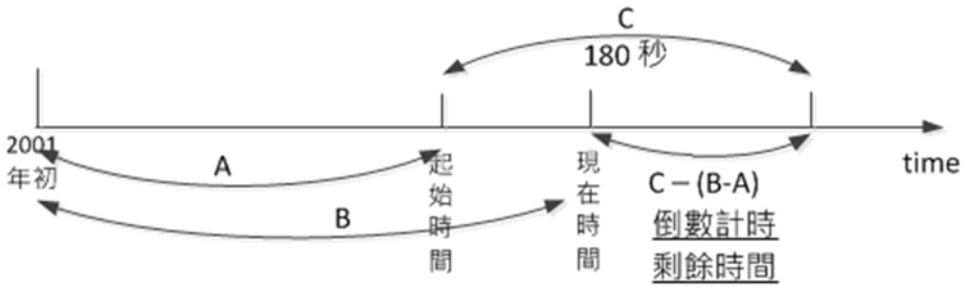


(4) 透過連結即產生紅框處的程式碼，控制 IB 建立的元件。



Step.7

(1) 接下來，我們在【紅框】中加入設定 swift 的程式碼。



傳回從 2001 年年初至現在共經過多少時間

```
startTime = NSDate.timeIntervalSinceReferenceDate()
```

設定計時器每隔 0.01 秒，執行 checkTime() 這個 function。

```
timer = NSTimer.scheduledTimerWithTimeInterval(0.01, target:self, selector: Selector("checkTime"), userInfo: nil, repeats: true)
```

傳回從 2001 年年初至呼叫 checkTime()現在共經過多少時間

```
 let currentTime = NSDate.timeIntervalSinceReferenceDate()
```

因為設定是倒數三分鐘(180秒) 所以要計算剩餘時間

```
 let elapsedTime = 180 - ( currentTime - startTime )
```

將剩餘的秒數換算為 幾分鐘：幾秒鐘，並顯示於【文字標籤】中

```
 let minutes = Int16(elapsedTime / 60.0)
 let seconds = Int16(elapsedTime % 60)
 let strMinutes = String(format: "%02d", minutes)
 let strSeconds = String(format: "%02d", seconds)
 myLabel.text = "\(strMinutes):\(strSeconds)"
```

```
import UIKit

var startTime = NSDate.timeIntervalSinceReferenceDate()
var timer = NSTimer()

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 myLabel: UILabel!


    @IBAction func tapBtn(sender: AnyObject) {
        startTime = NSDate.timeIntervalSinceReferenceDate()

        timer = NSTimer.scheduledTimerWithTimeInterval(0.01, target:self, selector:
            Selector("checkTime"), userInfo: nil, repeats: true)
    }

    func checkTime(){

        let currentTime = NSDate.timeIntervalSinceReferenceDate()
        let elapsedTime = 180 - ( currentTime - startTime )

        let minutes = Int16(elapsedTime / 60.0)
        let seconds = Int16(elapsedTime % 60)
        let strMinutes = String(format: "%02d", minutes)
        let strSeconds = String(format: "%02d", seconds)
        myLabel.text = "\(strMinutes):\(strSeconds)"

        if (elapsedTime <= 0) {
            timer.invalidate()
            // 讓計時器停下來， 就不會再次呼叫 updateTime: 方法
            let alertController = UIAlertController (title: "計時器", message: "倒數計時結束",
                preferredStyle: .Alert)
            alertController.addAction(UIAlertAction(title: "確定", style: .Default,
            handler: {action in self.myLabel.text = "倒數結束!" }   ))
            presentViewController(alertController, animated: true, completion: nil)
        }
    }
}
```

Step.8

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.9

當 App 啟動後在顯示畫面時，按下【開始】後，顯示倒數時間在【文字標籤】中，倒數結束將會顯示視窗。

自 我 練 習

實作執行後結果：

接下呢，我們要練習的是運用計時器來製作碼錶，按下開始就可以開始計時，按下結束則暫停計時。