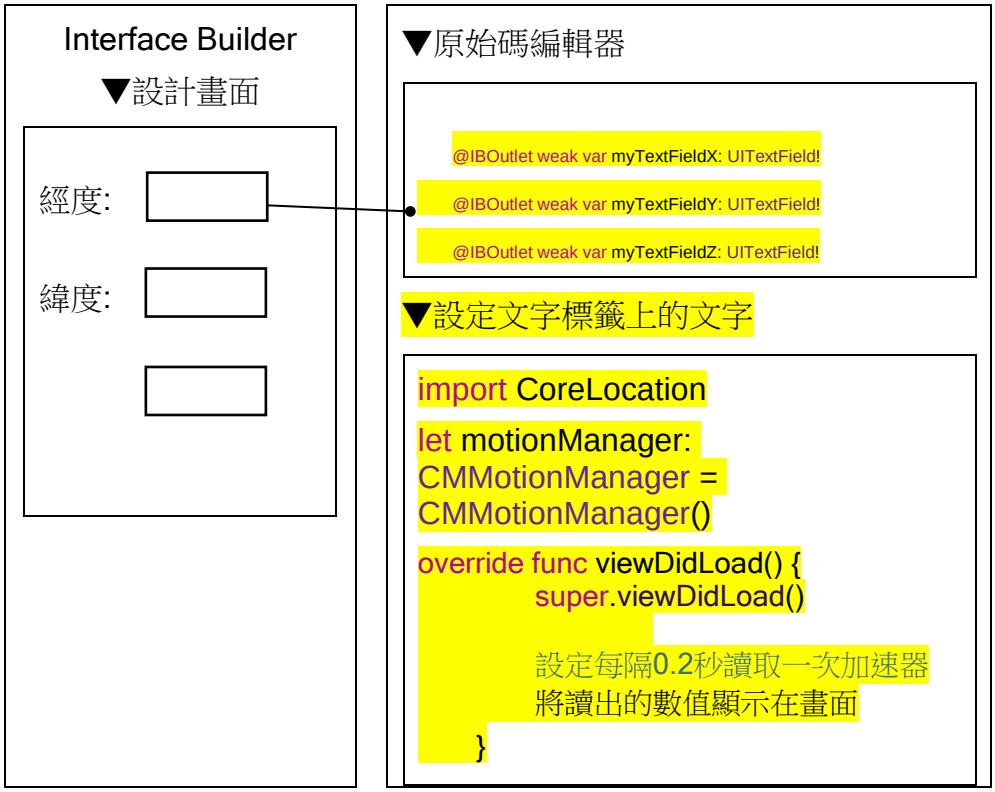
CHAPTER 7-3

CoreLocation: 讀取經緯度與方向

在手機中顯示目前位置的經緯度以及方向

學習概念：

1. 首先用 IB 建立【文字標籤】與【文字欄位】。
2. 接著將【文字欄位】與【程式碼連結】。
3. 最後在實作相關程式，載入 CoreLocation 函式庫
4. 在處理畫面載入後所觸發的事件，也就是【ViewDidLoad】方法。

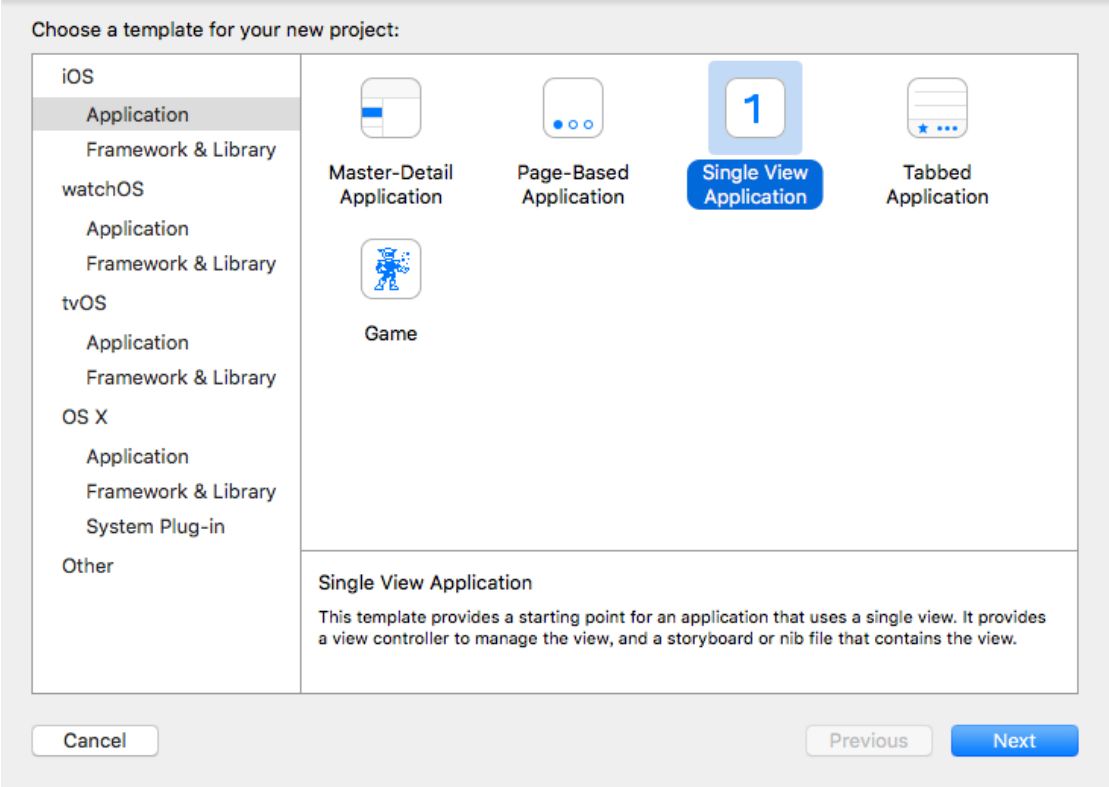


【執行結果】

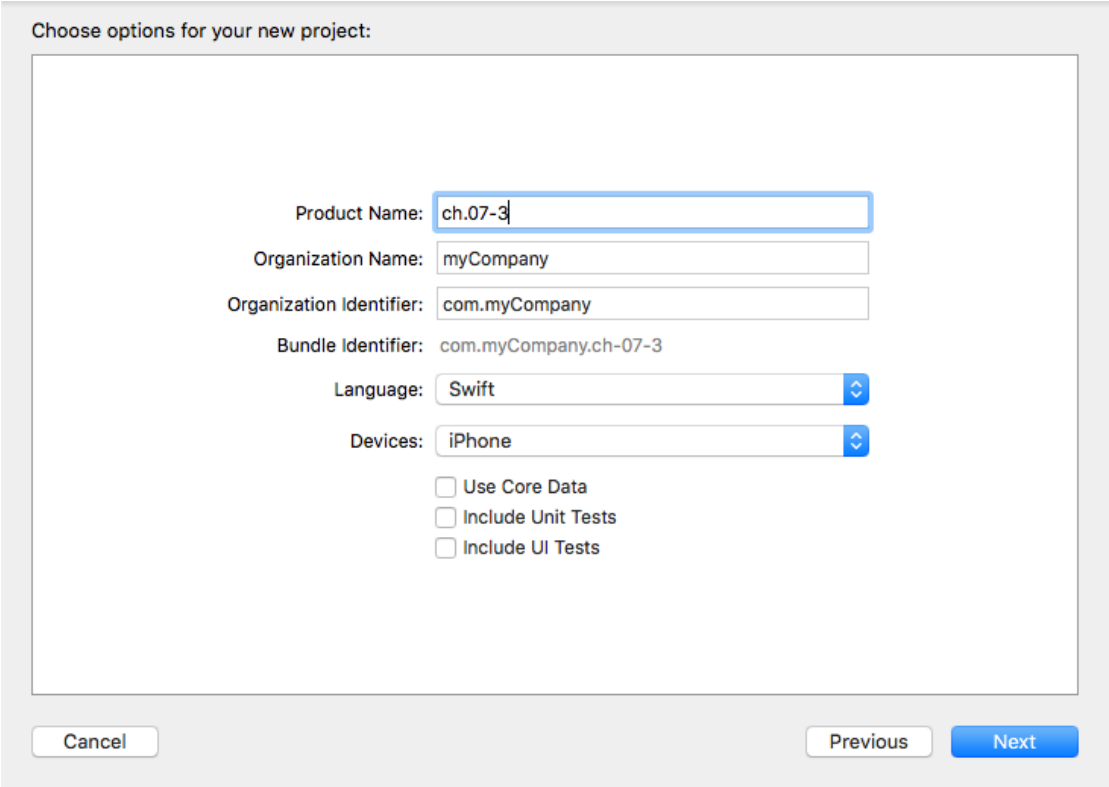
當 App 執行後，會隨著手機的移動【文字欄位】上顯示數值。

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。

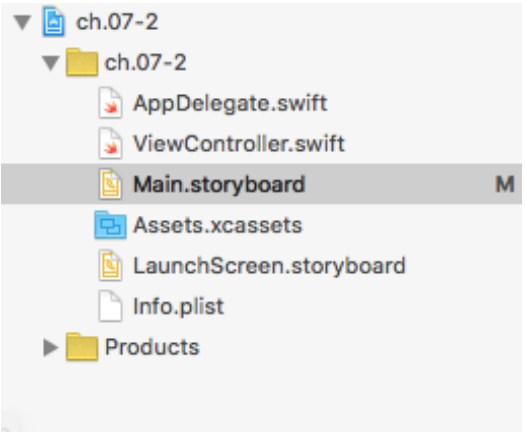


檔名及名稱設定，請將【Product Name】設定為 ch.07-3



Step.2

選取【Main.storboard】



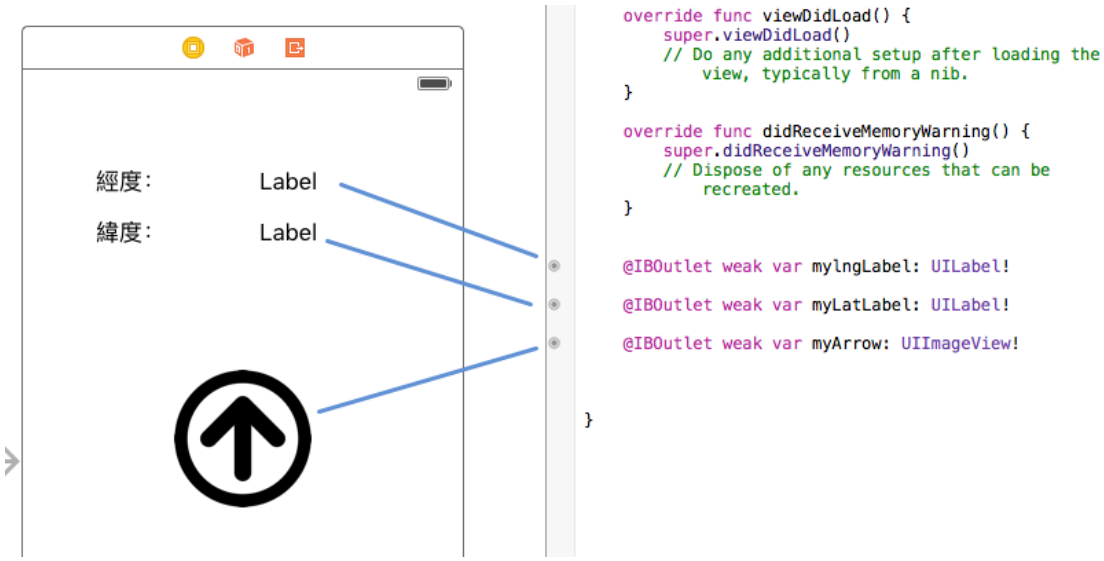
從【物件庫】中拖曳【Label】、【Text Field】【ImageView】到畫面中，加入圖檔 arrow.png，並將圖檔放置於 ImageView 中。

Step.3.

將編輯畫面分割為四部分，以方便連結元件與程式

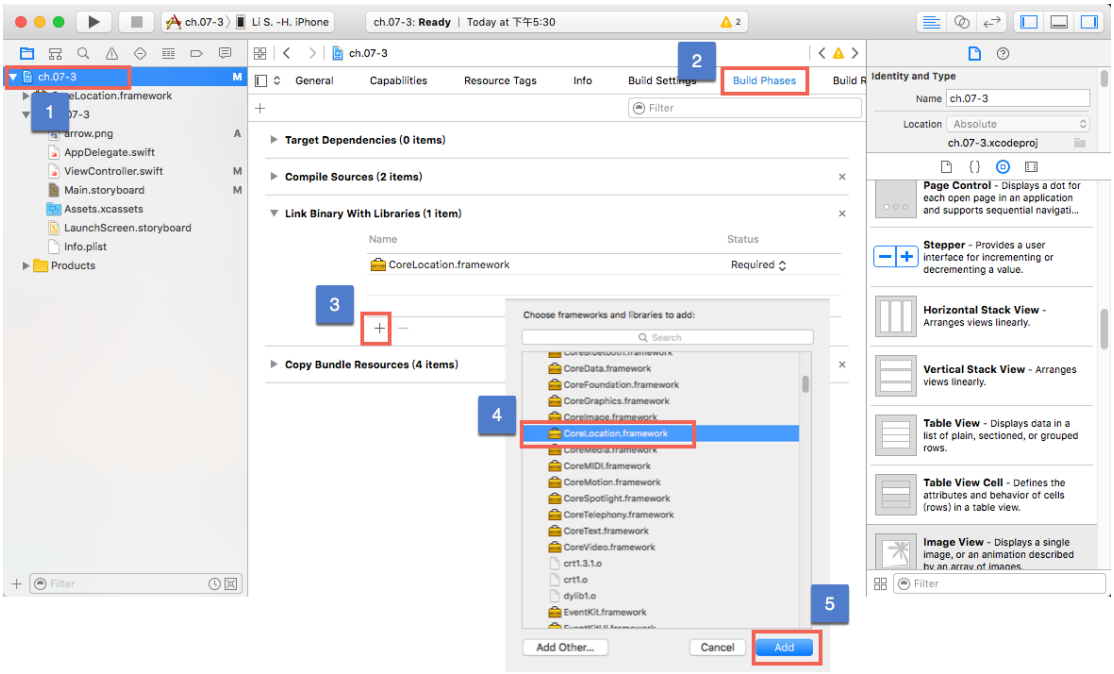
Step.4

(1) 將【文字欄位】和【名稱連結】，分別命名 mylngLabel, myLatLabel, myArrow



Step.5 。

(1) 點選右上方專案名稱，以開啟專案的基本設定。
(2) 切換到 Build Phase
(3) 下拉 Link Binary With Libraries，點選『+』以新增函式。
(4) 選擇 CoreLocation.Framework
(5) 點選『Add』以加入此函式庫



Step.6 。

(1) 於程式中 加入引用 CoreLocation 函式庫。

```
import CoreLocation
```

(2) 於 class ViewController 之後需加上 CLLocationManagerDelegate 變數，

```
class ViewController: UIViewController ,
CLLocationManagerDelegate{
```

(3) 建立位置感應器的物件

```
let locationManager = CLLocationManager()
```

(4) 宣告兩個變數資料型態是 NSString 用來接收傳出來的座標值

```
var LatitudeGPS = NSString()
var LongitudeGPS = NSString()
```

(5) 設定位置感應器物件的屬性。

```
讓位置感應器的作用於主程式上
self.locationManager.delegate = self
設定精細度為最高
self.locationManager.desiredAccuracy = kCLLocationAccuracyBest
設定多遠的距離才更新位置資訊，目前設定為一移動就更新
self.locationManager.distanceFilter = kCLDistanceFilterNone
獲得手機權限
self.locationManager.requestWhenInUseAuthorization()
開始執行取得位置資訊
self.locationManager.startUpdatingLocation()
開始執行取得方向資訊
self.locationManager.startUpdatingHeading()
```

kCLLocationAccuracyBestForNavigation	最高精確度，搭配其他的感應器並及時更新，適用於導航
kCLLocationAccuracyBest	最高精確度
kCLLocationAccuracyNearestTenMeters	精確度 10 公尺
kCLLocationAccuracyHundredMeters	精確度 100 公尺
kCLLocationAccuracyKilometer	精確度 1 公里

kCLLocationAccuracyThreeKilometers	精確度 3 公里

(6) 手動撰寫位置更新時的動作。

讀出經緯度的數值並將數值轉換為文字。

```
func locationManager(manager: CLLocationManager,
didUpdateLocations locations: [CLLocation]){

        LatitudeGPS = String(format: "%.6f",
manager.location!.coordinate.latitude)
        LongitudeGPS = String(format: "%.6f",
manager.location!.coordinate.longitude)
        myLatLabel.text = String( LatitudeGPS )
        myLngLabel.text = String( LongitudeGPS )

    }
```

(7) 手動撰寫方向更新時的動作。

每次要重置圖片的位置，才能保證圖片每次偏轉量正常，而不是疊加上去。

讀出弧度的數值並將其轉為角度，並旋轉圖形。

。

```
func locationManager( manager: CLLocationManager,
didUpdateHeading newHeading: CLHeading) {
        self.myArrow.transform = CGAffineTransformIdentity
        let degree = manager.heading!.magneticHeading * M_PI /
180
        self.myArrow.transform =
CGAffineTransformMakeRotation( CGFloat( degree ) )
    }
```

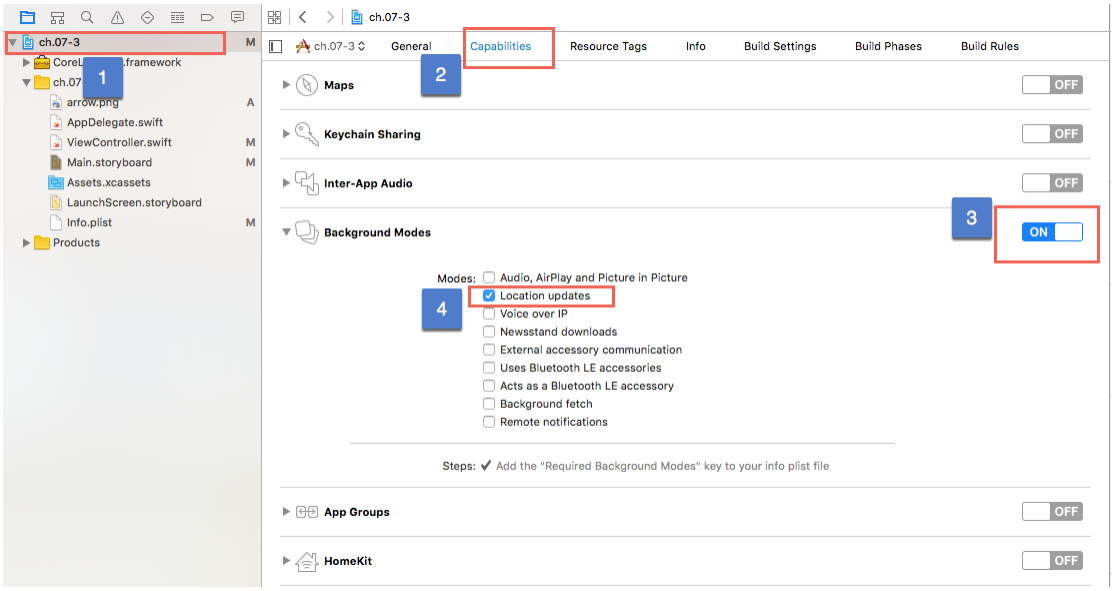
```
import UIKit
import CoreLocation

class ViewController: UIViewController , CLLocationManagerDelegate{

    let locationManager = CLLocationManager()
    var LatitudeGPS = NSString()
    var LongitudeGPS = NSString()

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

        self.locationManager.delegate = self
        self.locationManager.desiredAccuracy = kCLLocationAccuracyBest
        self.locationManager.distanceFilter = kCLDistanceFilterNone
        self.locationManager.requestWhenInUseAuthorization()
        self.locationManager.startUpdatingLocation()
        self.locationManager.startUpdatingHeading()

    }

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

    func locationManager(manager: CLLocationManager, didUpdateLocations locations: [CLLocation]){

        LatitudeGPS = String(format: "%.6f", manager.location!.coordinate.latitude)
        LongitudeGPS = String(format: "%.6f", manager.location!.coordinate.longitude)
        myLatLabel.text = String( LatitudeGPS )
        myLngLabel.text = String( LongitudeGPS )
    }

    func locationManager( manager: CLLocationManager,  didUpdateHeading newHeading: CLHeading) {

        self.myArrow.transform = CGAffineTransformIdentity
        let degree = -1 * manager.heading!.magneticHeading * M_PI / 180
        self.myArrow.transform = CGAffineTransformMakeRotation( CGFloat( degree ) )
    }

    @IBOutlet weak var myLngLabel: UILabel!

    @IBOutlet weak var myLatLabel: UILabel!

    @IBOutlet weak var myArrow: UIImageView!

}
```

Step.7 。

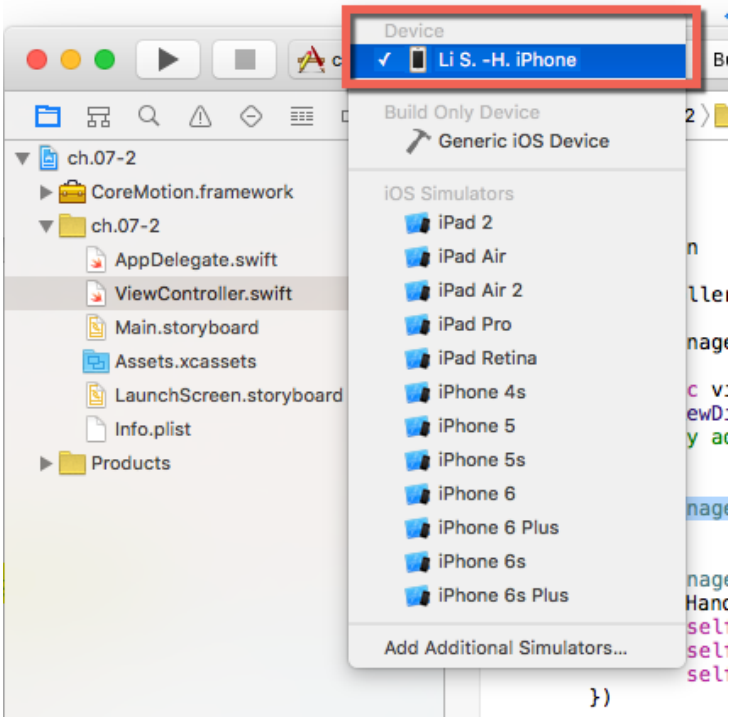
(1) 在執行之前需要先設定 info.plist 以讓手機在執行 APP 時會詢問是否允許使用 App 時取用您的位置嗎？

需要手動增加 下列兩個參數任一，以產生對話窗獲得使用者授權。
NSLocationAlwaysUsageDescription。
NSLocationWhenInUseUsageDescription

(2) 允許手機在背景執行時仍然可以取得位置的更新。



(3) 執行程式前，請先選擇實體手機


Device
Li S. -H. iPhone
Build Only Device
Generic iOS Device
iOS Simulators
iPad 2
iPad Air
iPad Air 2
iPad Pro
iPad Retina
iPhone 4s
iPhone 5
iPhone 5s
iPhone 6
iPhone 6 Plus
iPhone 6s
iPhone 6s Plus
Add Additional Simulators...
ch.07-2
CoreMotion.framework
ch.07-2
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products

自 我 練 習

實作執行後結果：

設計一個球滾動的遊戲，球會隨著重力的位置而移動，因此，當手機改變方向時，球會在視窗內滾動。