

Build an application

Tutor : Michael Pan

Application包涵

- Source codes
 - 主要程式碼
 - Frameworks
- Xib files
 - 圖形介面
 - 物件之間的關係
- Resources - (圖檔，音樂，等等)
- info.plist - 設定檔

UIKit Framework

- 提供基本介面的元件
- UIApplication
 - 每個程式裡只有唯一的一個
 - 分配Event
 - 處理status bar, 程式icon...
 - 利用delegation來完成程式任務而非繼承

```
[UIApplication sharedApplication]
```

Delegation - 委任

- 物件要完成某件行為的時候利用別人定義好的某(幾)項功能來完成
- <UIApplicationDelegate>

Delegation - Animation



Delegation - Code



```
@interface A{
    B* delegate;
}
-(void) passToB;
@end

@implementation A
-(void) passToB{
    [delegate atctionOfB];
}
@end
```

```
A * a = [A new] ;
B * b = [B new];
a.delegate = b;
```

```
[a passToB]
```

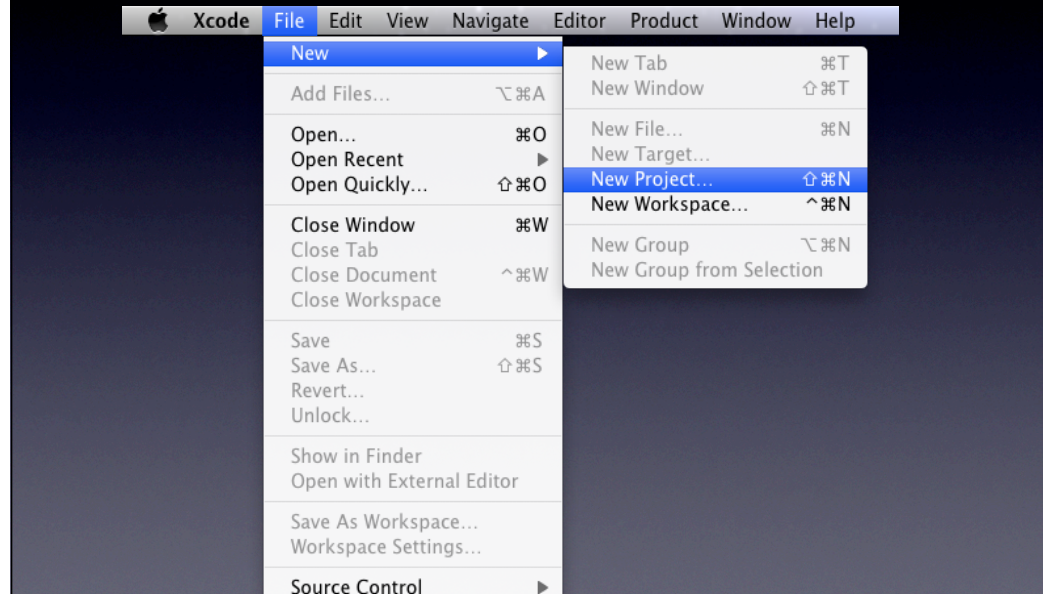
UIApplicationDelegate

```
- (void)applicationDidReceiveMemoryWarning:  
(UIApplication *)application;  
- (void)applicationWillResignActive:  
(UIApplication *)application;  
- (BOOL)application:(UIApplication  
*)application handleOpenURL:(NSURL *)url;  
- (BOOL)application:(UIApplication  
*)application didFinishLaunchingWithOptions:  
(NSDictionary *)launchOptions  
- (void)applicationWillTerminate:  
(UIApplication *)application;
```

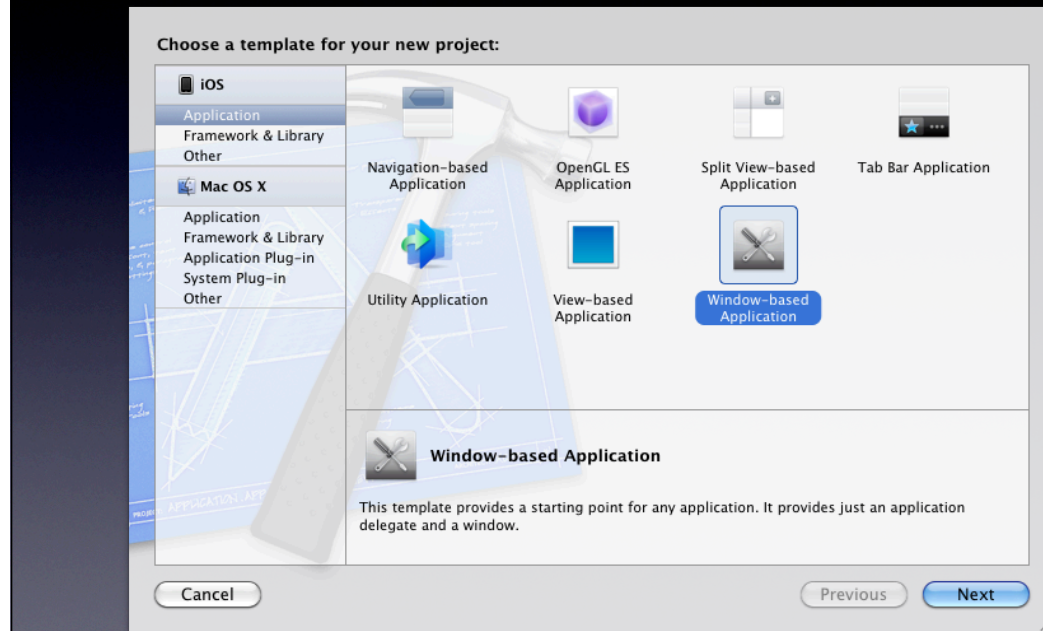
開啟專案



開啟專案 or

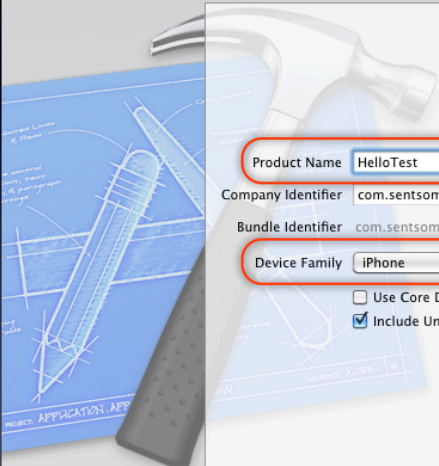


Window-based



命名

Choose options for your new project:



Product Name

Company Identifier

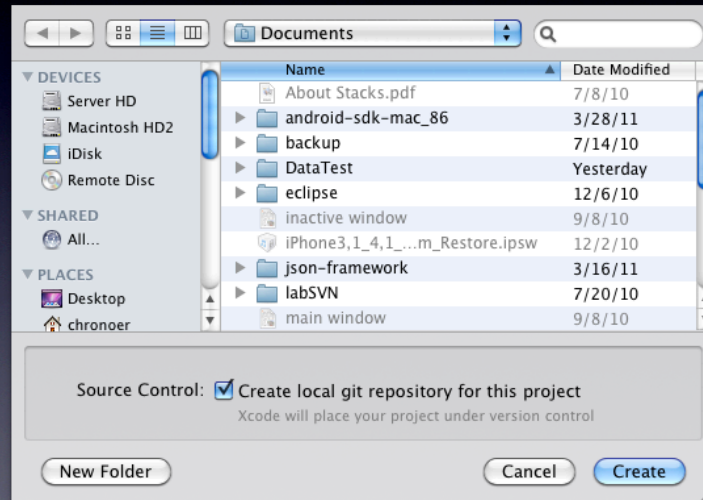
Bundle Identifier

Device Family

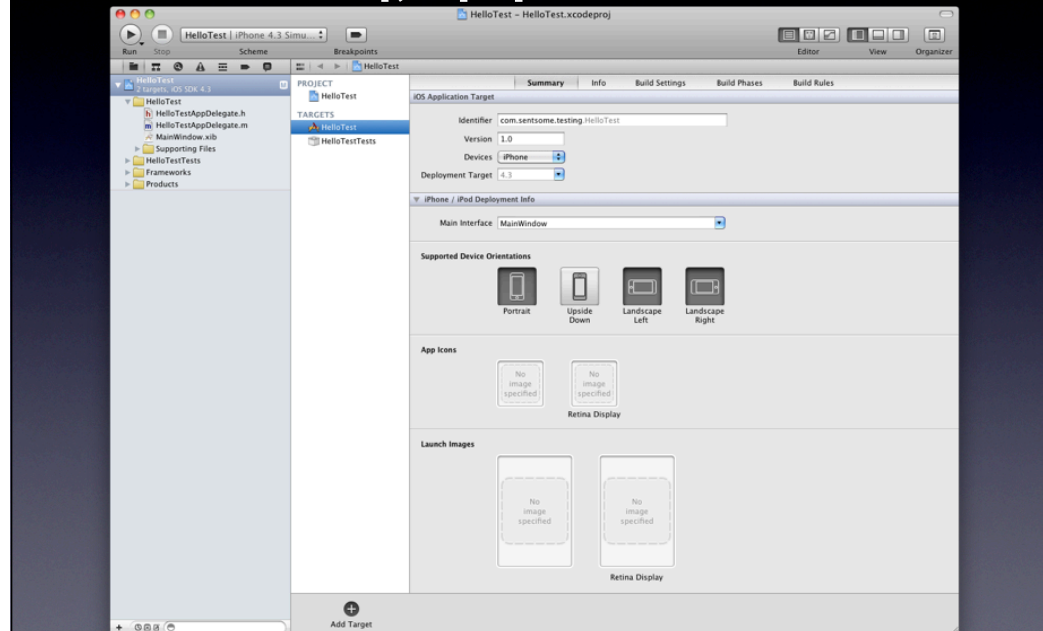
Use Core Data

Include Unit Tests

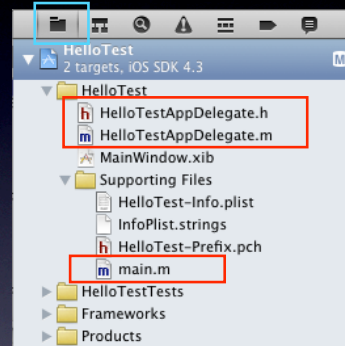
Save



Xcode 視窗



左邊欄

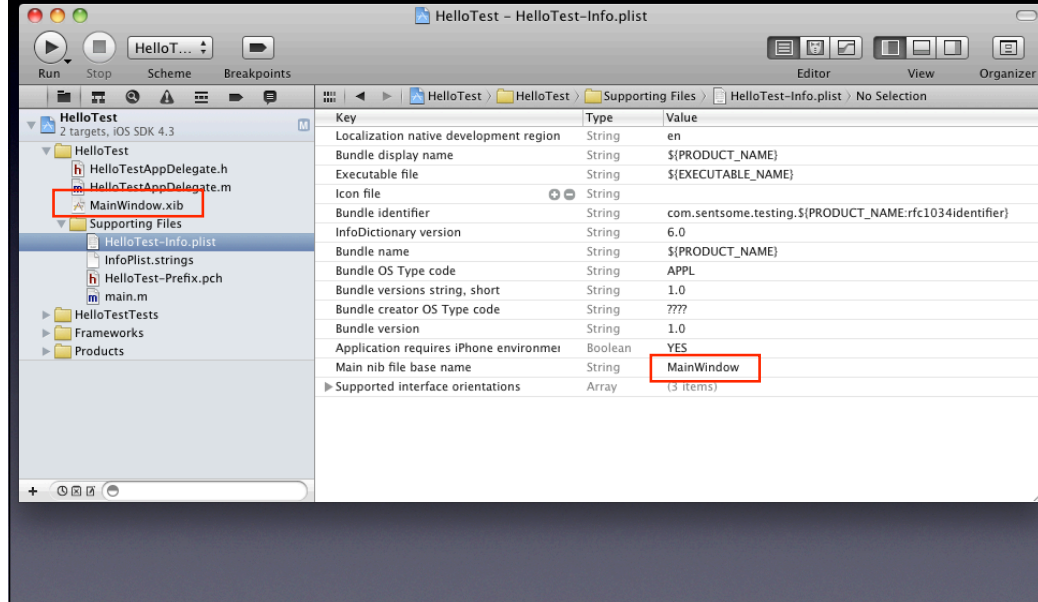


實作 <UIApplicationDelegate>

產生 UIApplication 實體

Project Navigator

Main XIB



main.m

```
#import <UIKit/UIKit.h>

int main(int argc, char *argv[]) {

    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc]
init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```


UIApplicationMain

```
int UIApplicationMain (  
    int argc, char *argv[], NSString *principalClassName,  
    NSString *delegateClassName  
);
```

argc

在 argv 裡元素的個數。通常對應 main 的參數。

argv

一串輸入值。通常對應 main 的參數。

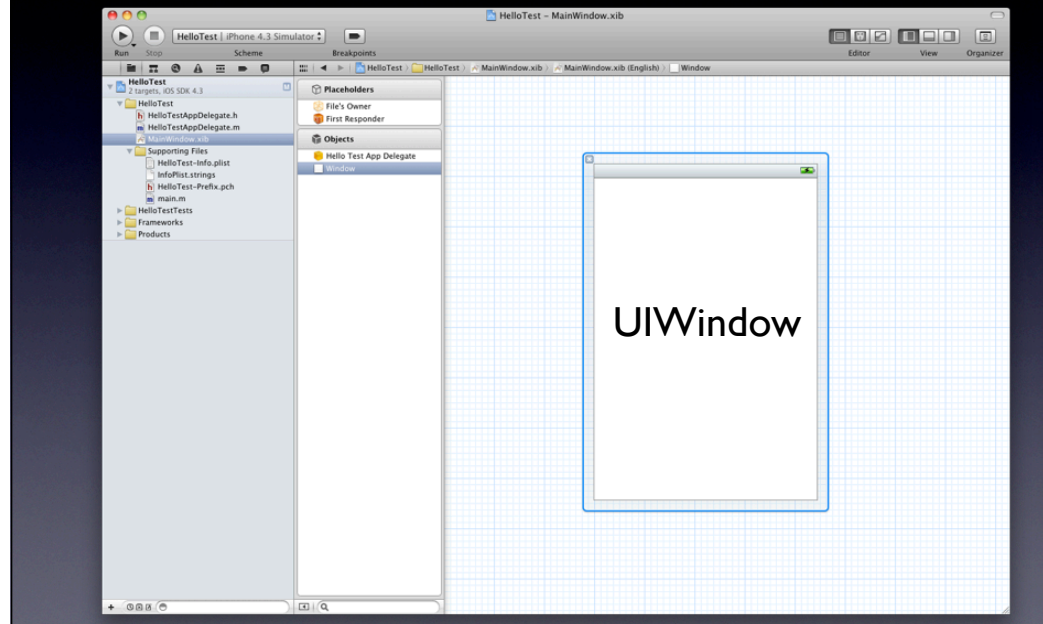
principalClassName

UIApplication 或其子類別實體的 Class name，如果值是 nil 則認為是 UIApplication。

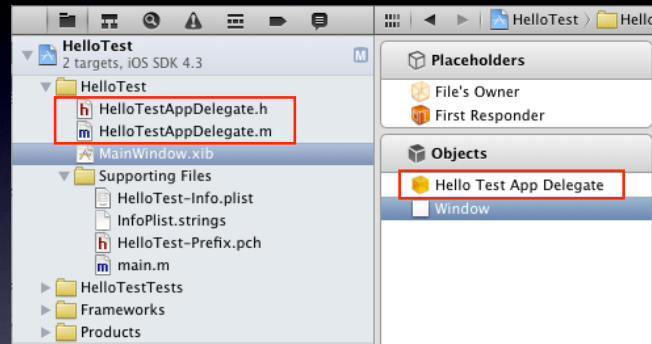
delegateClassName

接受 application 委任實體的 Class name。如果 principalClassName 是 UIApplication 的子類別時，可以指定這個子類別當委任者。其子類別的實體接受 application-delegate 訊息。如果是 nil 代表你要從 main nib 檔(在 iOS 是 xib 檔，且一開始預設為 MainWindow.xib)載入受委任者物件。

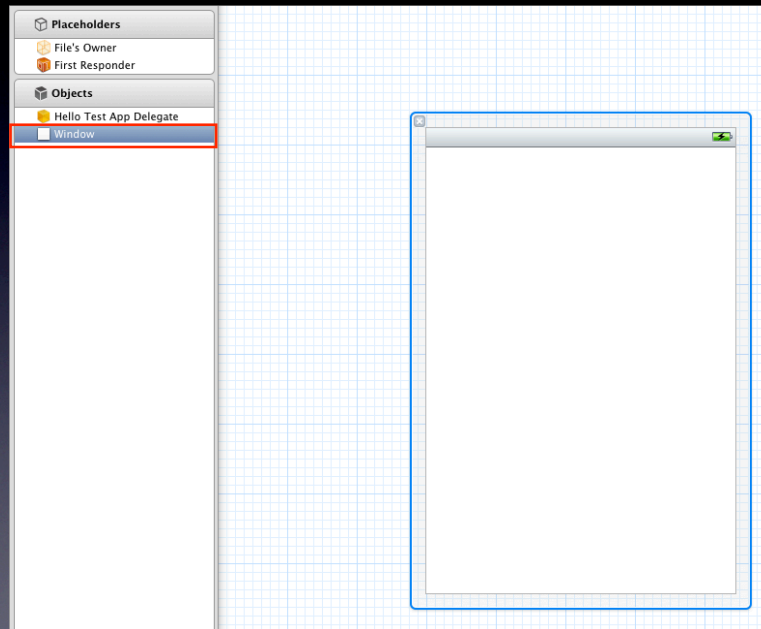
打開 MainWindow.xib



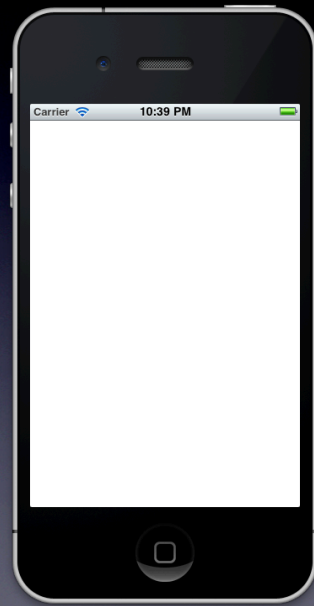
由 IB 產生的物件 - AppDelegate



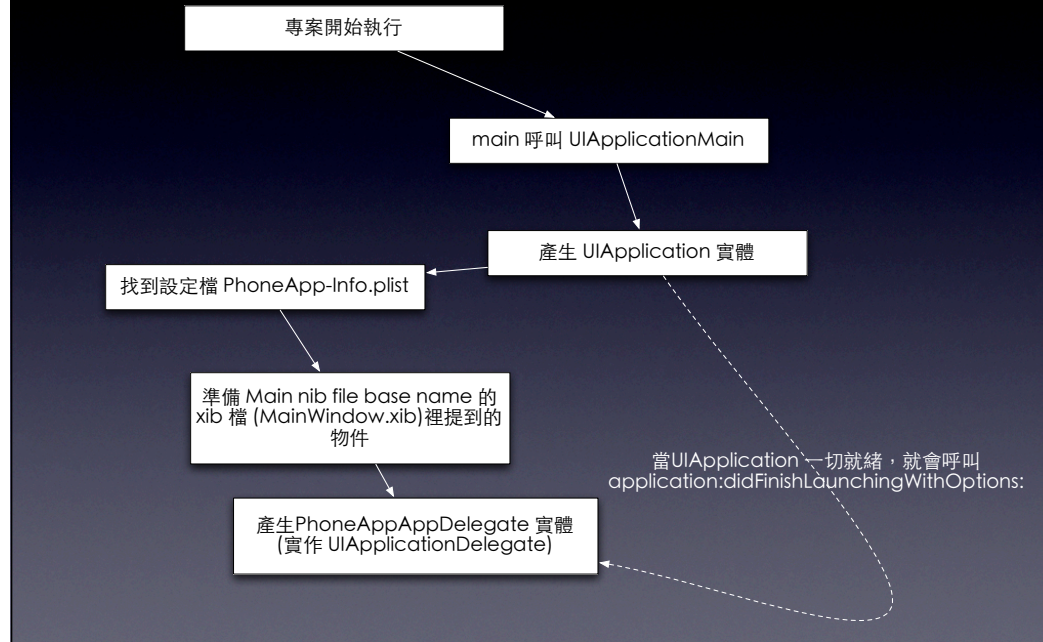
最基本的畫面 - UIWindow



執行結果



流程圖



Model, View, Controller

MVC Design Pattern

Controller

Model

View

View

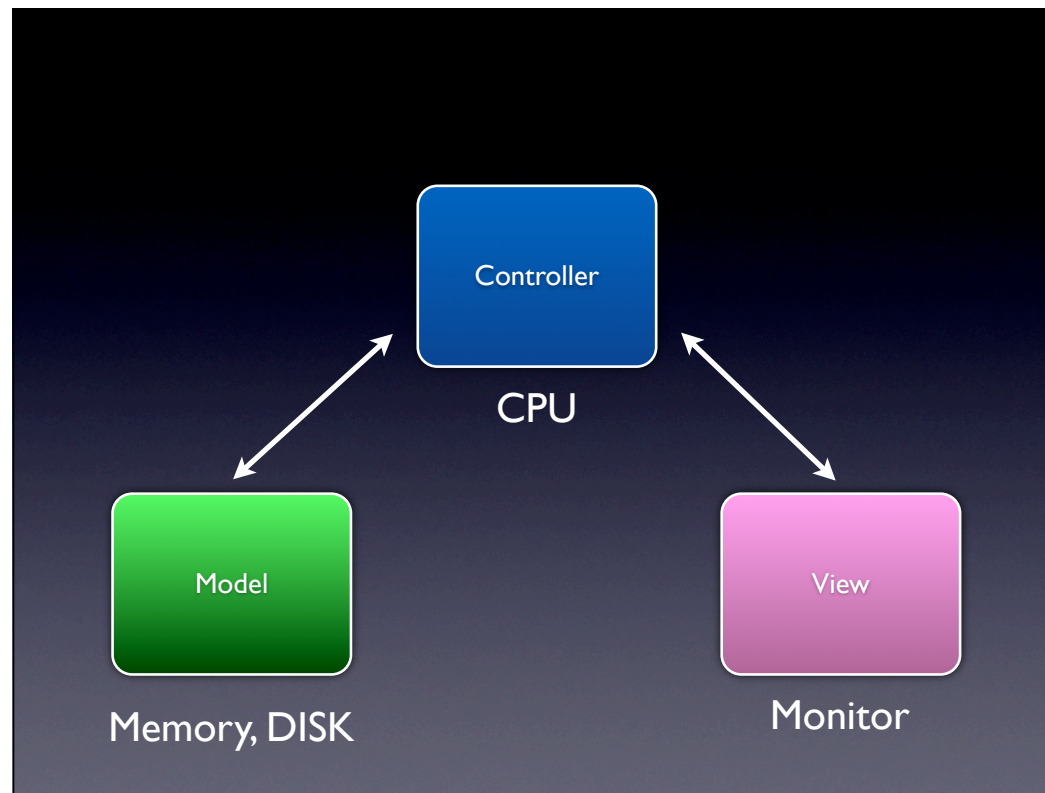
- 資料呈現
- 和使用者接觸的部分
- 不儲存資料

Model

- 儲存，管理資料
- 不和使用者互動
- 通常不和View直接接觸

Controller

- Model 和 View之間溝通的媒介
- 使用者和View接觸，依照使用者的指示，把資料從Model搬到View去呈現
- 程式的邏輯，或是大腦



Interface Builder

Xib or Nib files

- V in MVC model
- 排版所需的使用者介面
- 連結Controller和使用者的介面

Load nib or xib file

- 執行時Object會被解壓縮
- 儲存value / setting
- 確定IBAction和IBOutlet的連結
- 沒有定義解壓縮的順序

當xib檔Loading成功之後

- 在controller 實作 -awakeFromNib
- 當-awakeFromNib被呼叫時表示所有xib上的東西都解壓縮完，connection也都連結好，xib的物件都被建立

Demo - awakeFromNib

Target, Action & Event

Event-Event Handler

- View類型的物件可以把和使用者互動產生的Event傳送給某個target - event handler，再依target所定義的action行為而動作
- Event有許多類型
 - Touch
 - Value change
 - Editing

Event, Target/Action

Type Something

target : `myObject`
action : `@selector(print)`
event : `TouchUpInside`

Event occurs



```
@interface MyController {  
}  
  
-(void) print;  
@end
```

Event Handler

Action methods

(void)actionMethod;

// 不管誰傳送event，不管event是那一個，直接執行

- (void)actionMethod:(id)**sender**;

// 想要知道是誰傳送event，但不理會event是什麼

- (void)actionMethod:(id)**sender** withEvent:(UIEvent *)**event**

// 關心event長什麼樣子，傳送者是誰

Example



```
-(void) valueChanged:(id) sender {  
    value = [sender value];  
}
```

Multiple target-action

- 對於同一個Event可以觸發多個target和多個action
- Desktop 的Cocoa 只有一個target-action
- 可以在IB設定好

手動設定

```
@interface UIControl
```

```
- (void)addTarget:(id)target action:(SEL)action
```

```
    forControlEvents:
```

```
    (UIControlEvents)controlEvents;
```

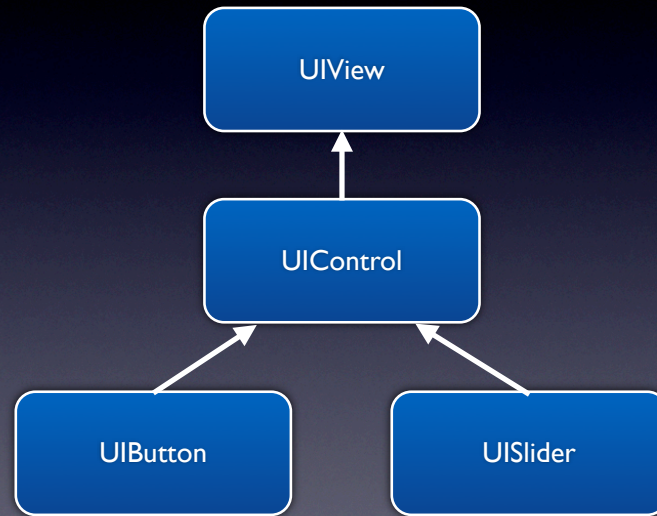
```
- (void)removeTarget:(id)target action:(SEL)action
```

```
    forControlEvents:
```

```
    (UIControlEvents)controlEvents;
```

```
@end
```


UI - Components



Target, Action & Event

Review

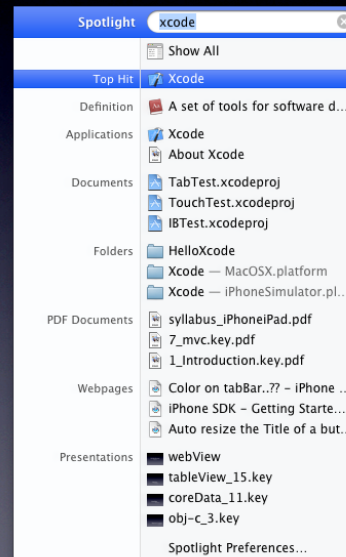
- Xcode & Interface Builder

Oh! Xcode



我的Xcode在那？

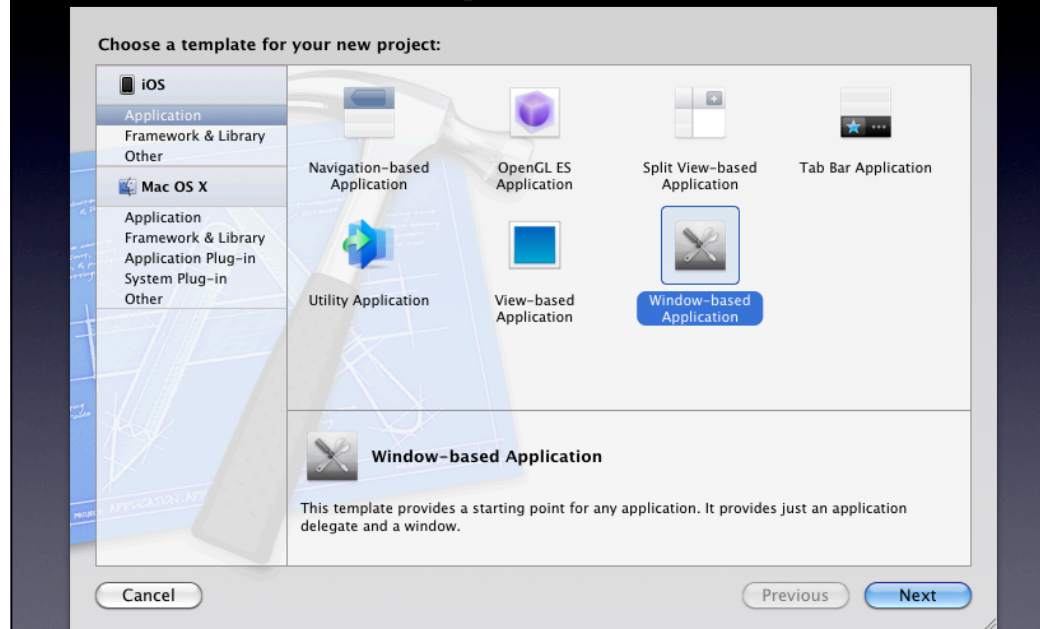
Control + 空白鍵



開啟一個新Project

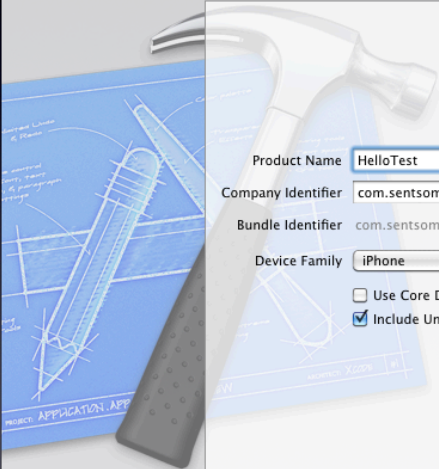


你會看到



給個名字

Choose options for your new project:



Product Name

Company Identifier

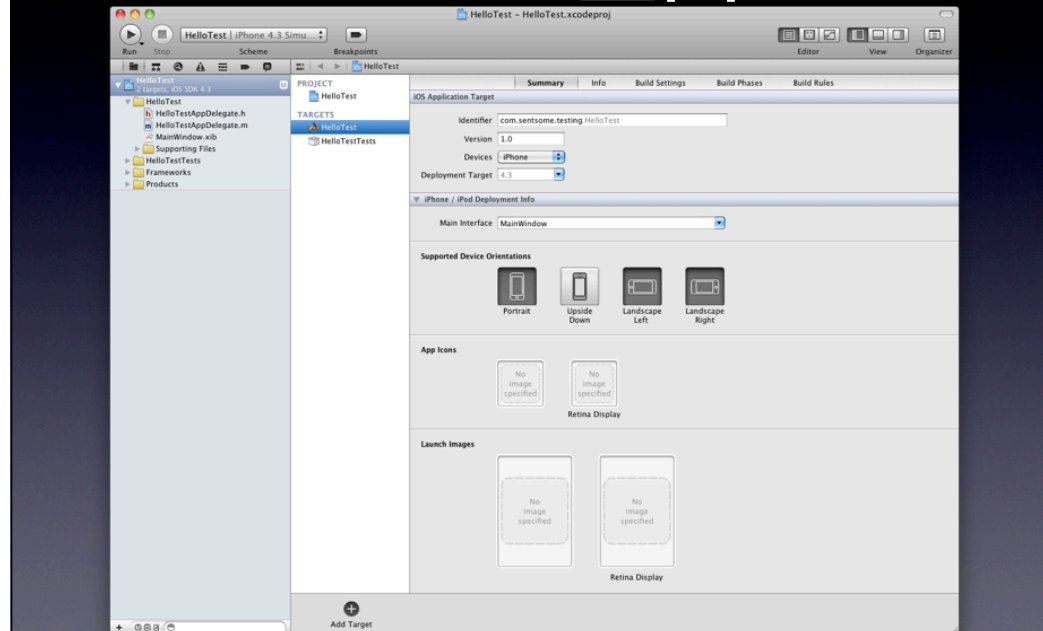
Bundle Identifier

Device Family

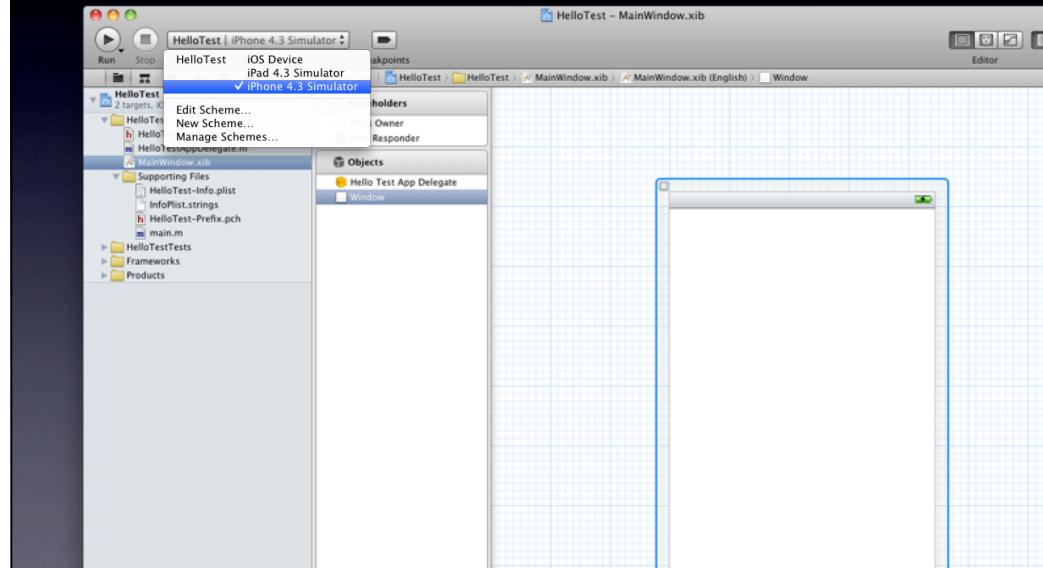
Use Core Data

Include Unit Tests

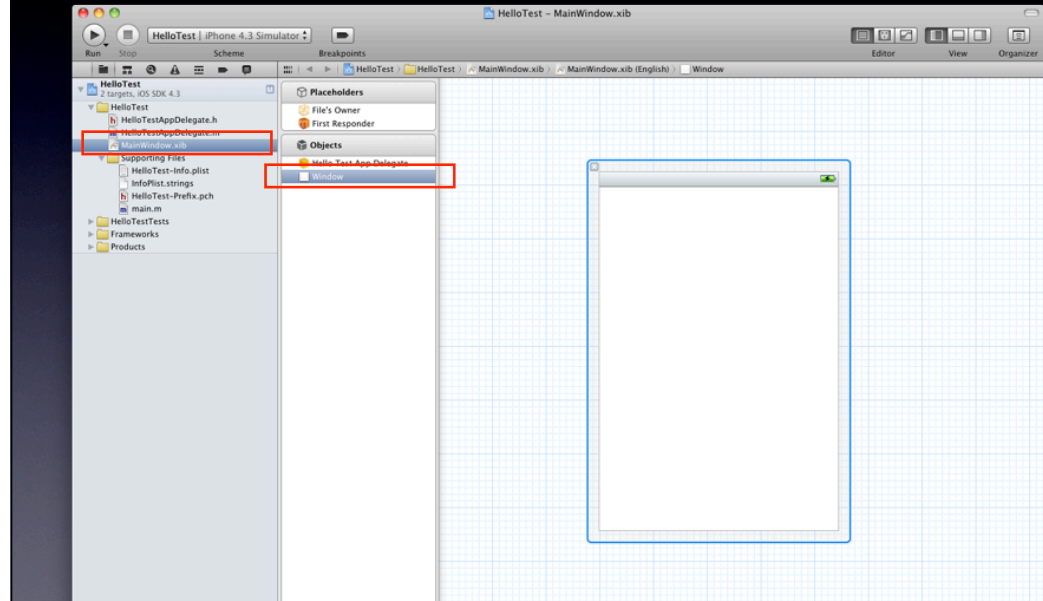
Xcode 畫面



選擇 Simulator



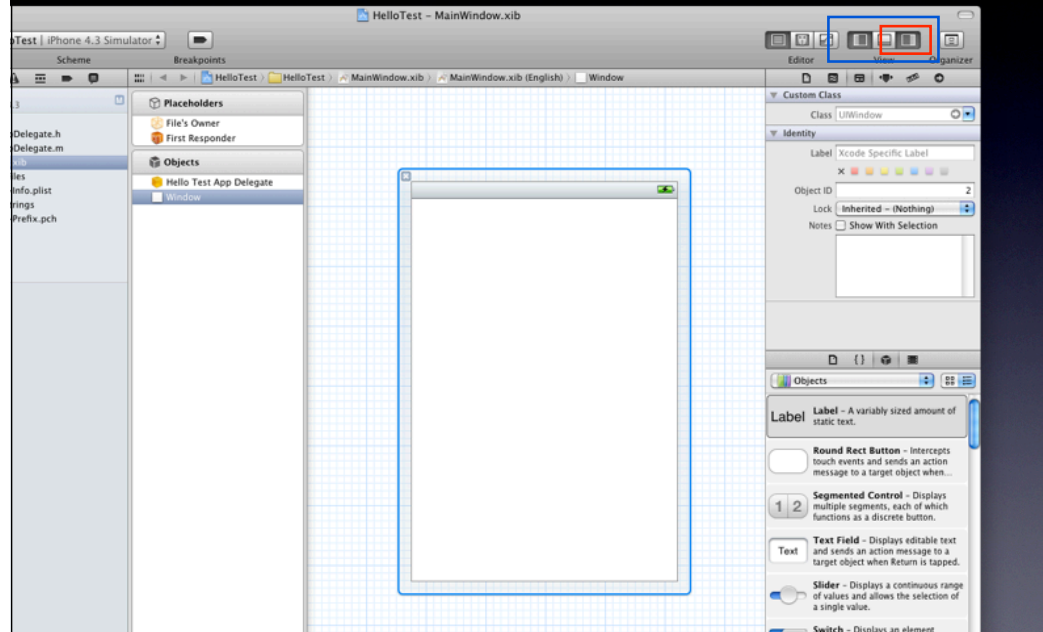
開啟Xib檔



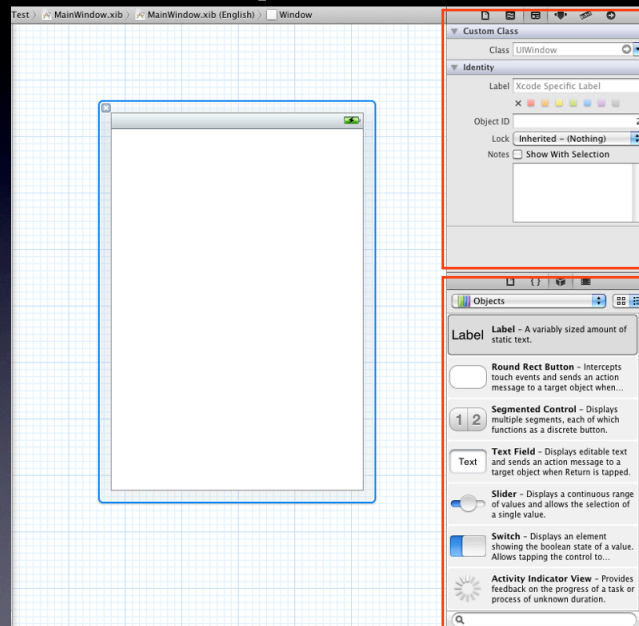
Oh! Interface Builder



有那些東東



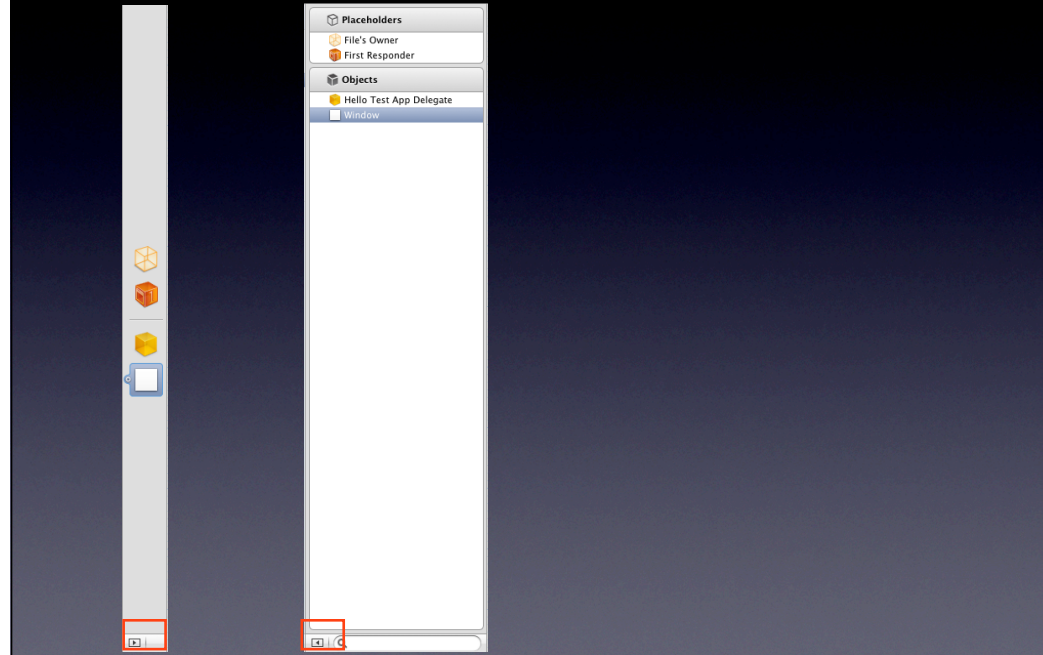
Inspector & Library



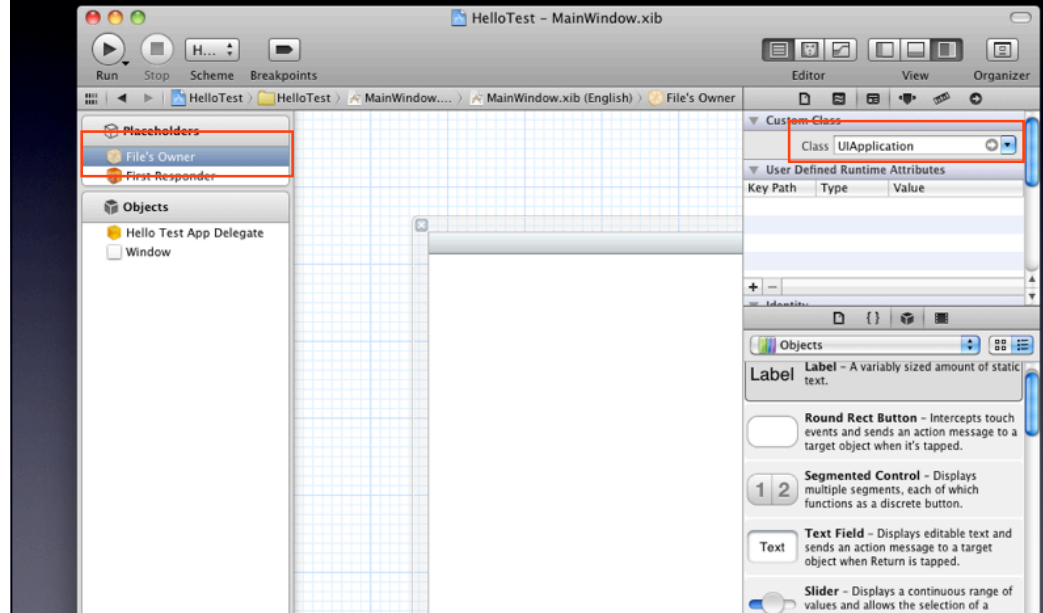
Inspector

Library

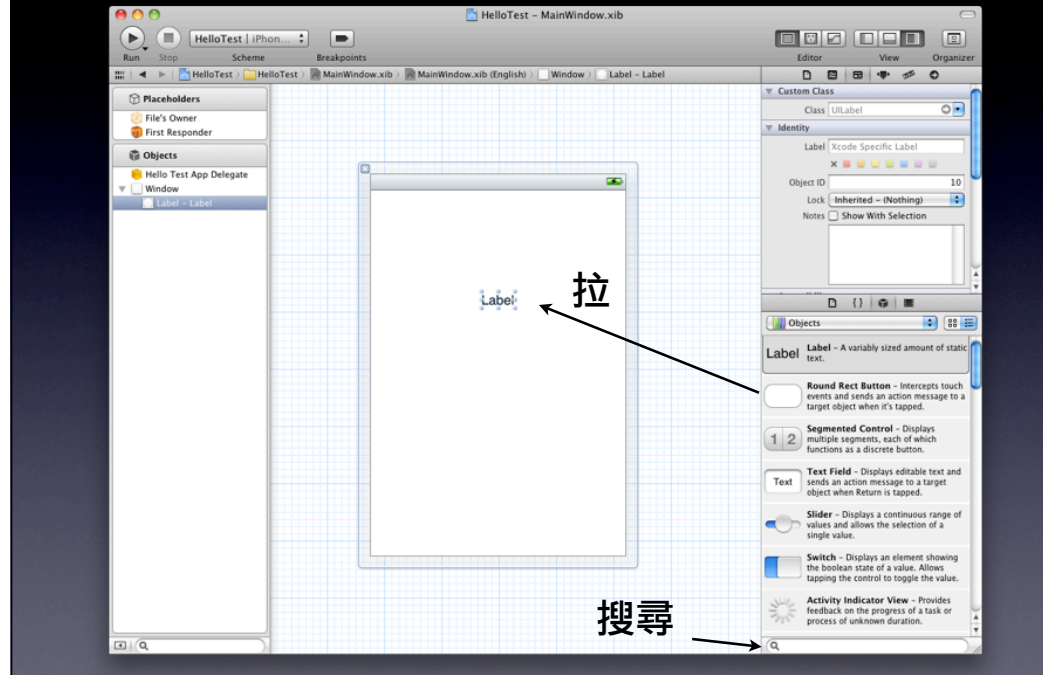
Document Window



File's Owner



拉好視覺元件

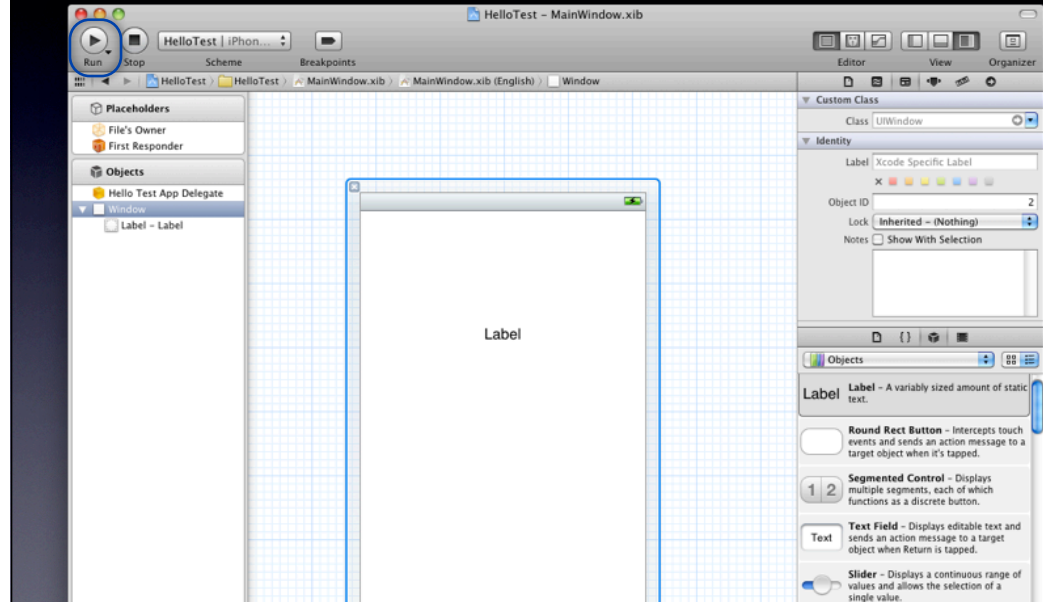


記得存檔

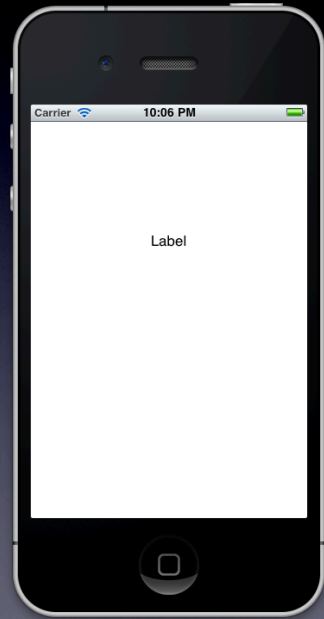
Command + S

⌘: Command

回到Xcode 執行程式



結果



Hello iPhone App

Templates

Choose a template for your new project:

for game 3D Model

Basic

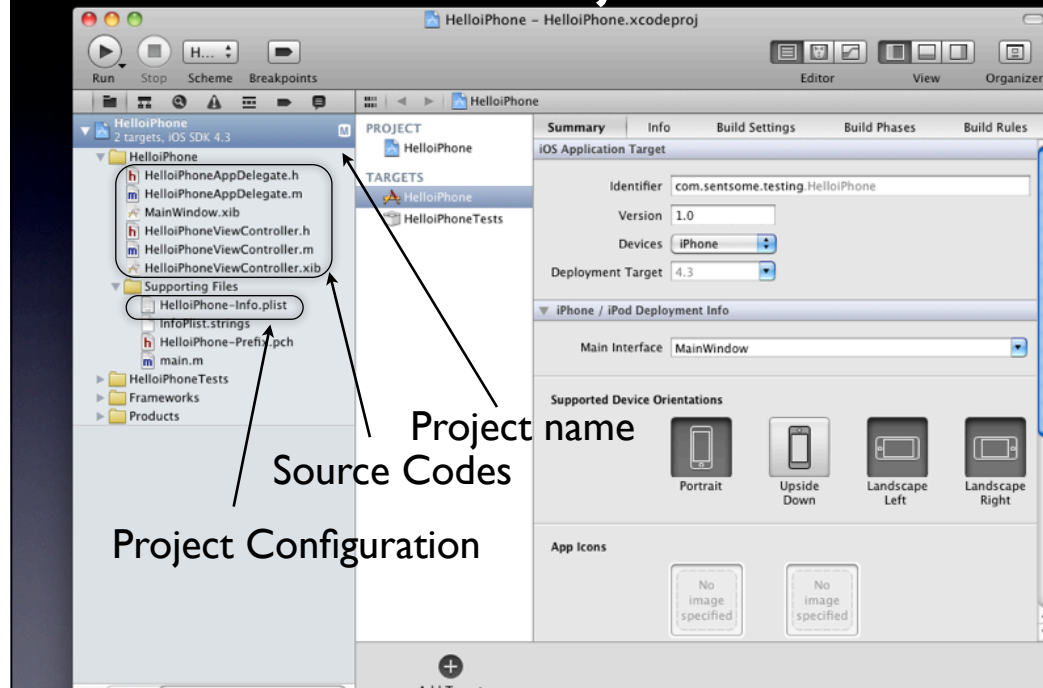
With Controller and View

View-based Application

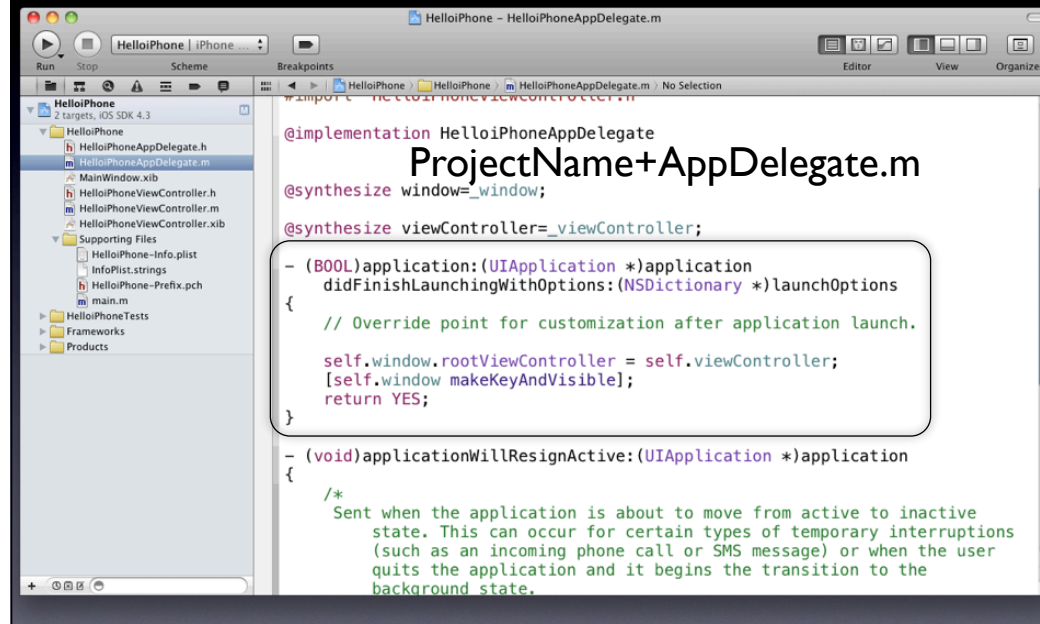
This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a nib file that contains the view.

Cancel Previous Next

View-Based Project



Entry Point



```

@implementation HelloiPhoneAppDelegate
ProjectName+AppDelegate.m
@synthesize window=_window;
@synthesize viewController=_viewController;

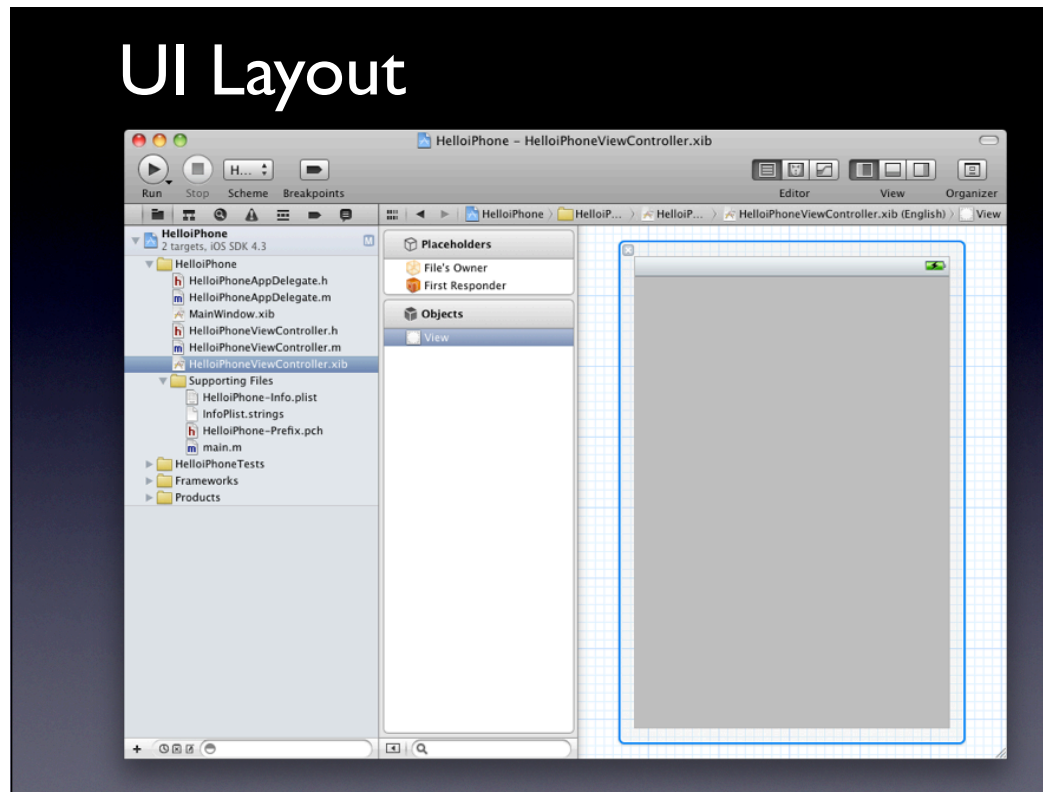
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.

    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}

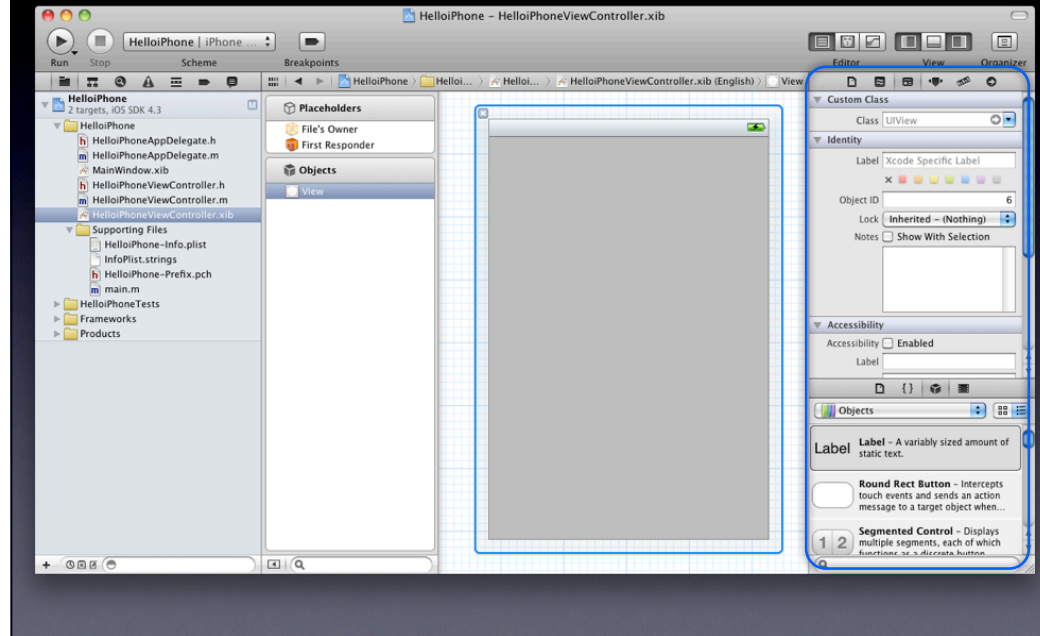
- (void)applicationWillResignActive:(UIApplication *)application
{
    /*
     Sent when the application is about to move from active to inactive
     state. This can occur for certain types of temporary interruptions
     (such as an incoming phone call or SMS message) or when the user
     quits the application and it begins the transition to the
     background state.
    */
}

```

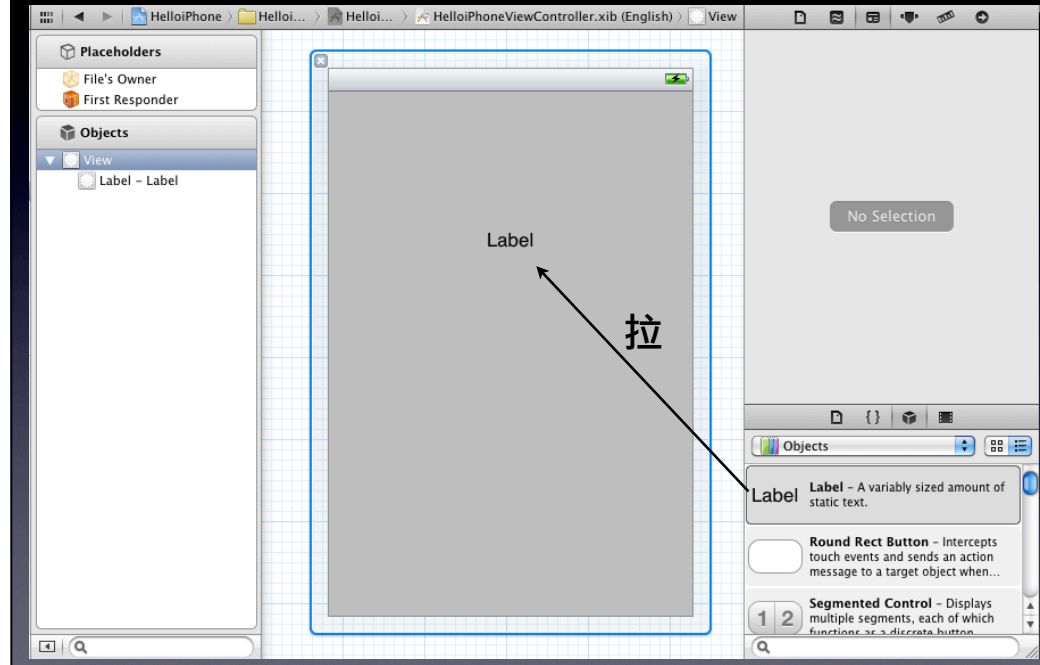

UI Layout



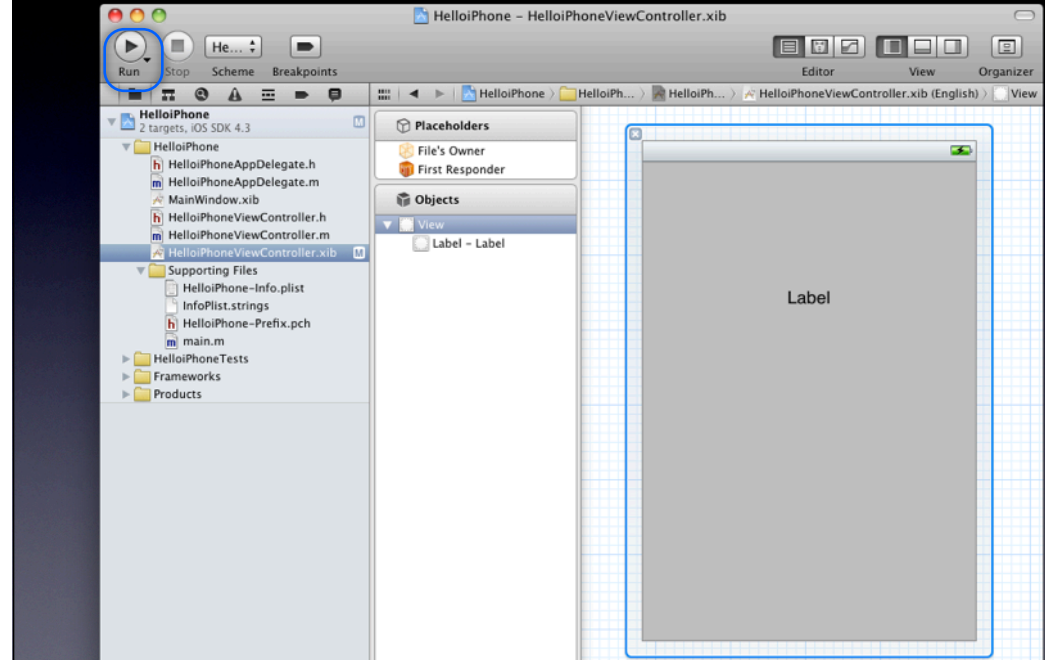
Interface Builder



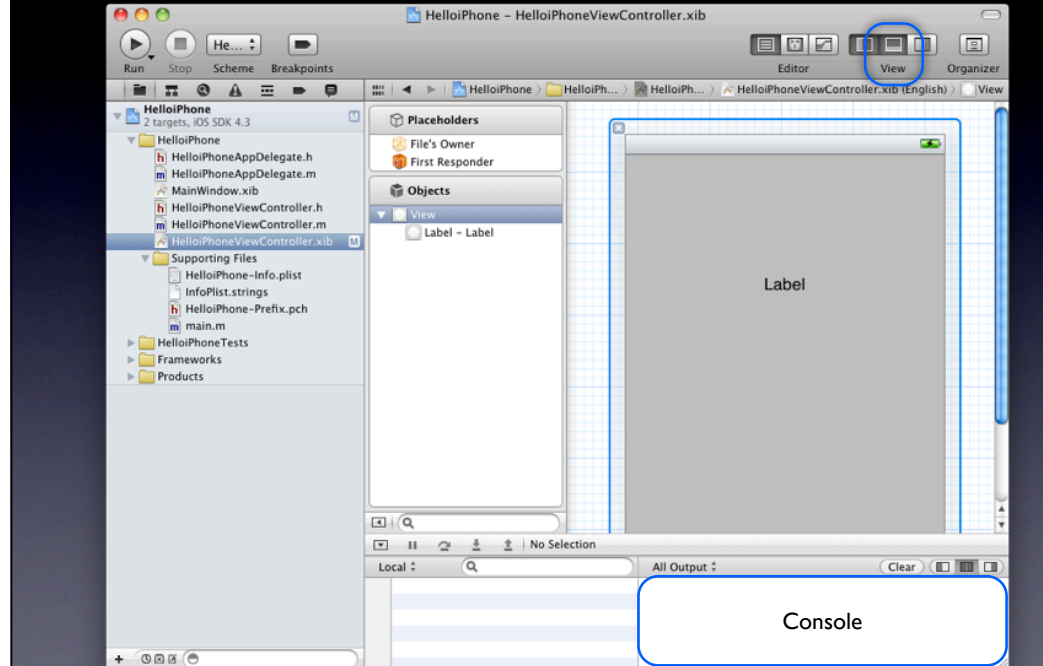
拉元件到View



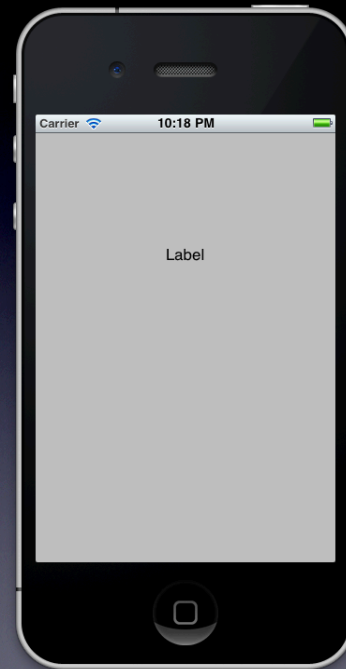
Build and Go - XCode



Console



Recap



String in Objective-C

@” this is a string ” not “string in c or c++”

是一個物件

內容不可更動

NSLog()

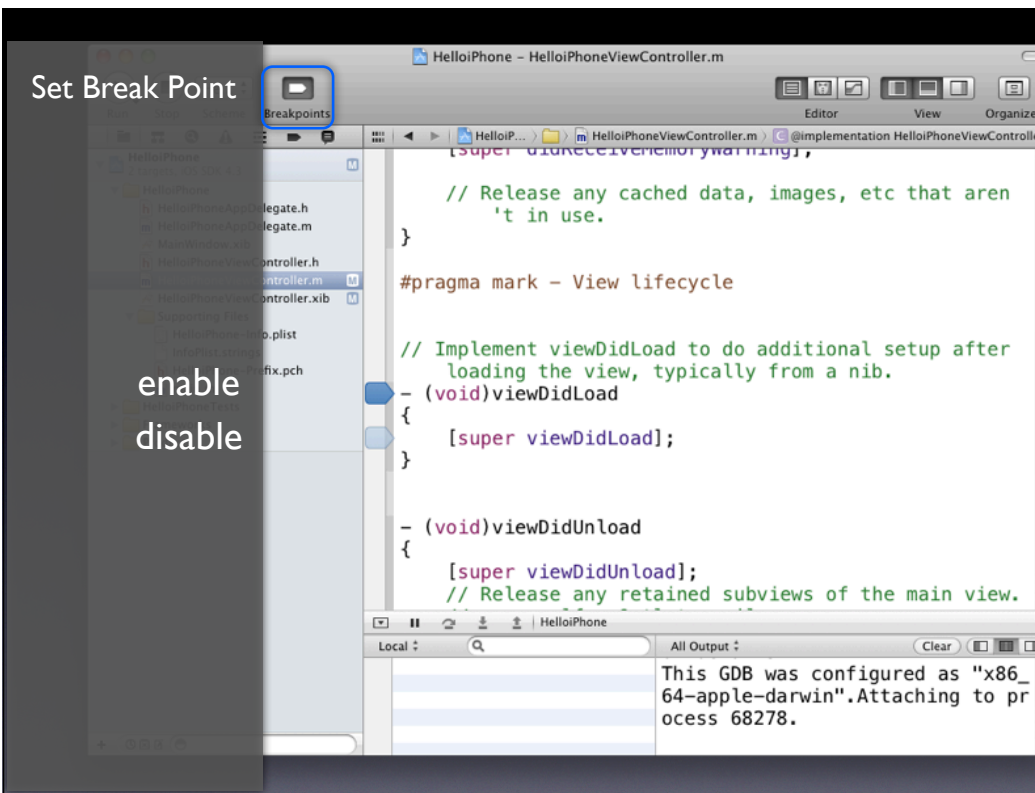
類似printf()

%d : 整數

%f : 浮點數

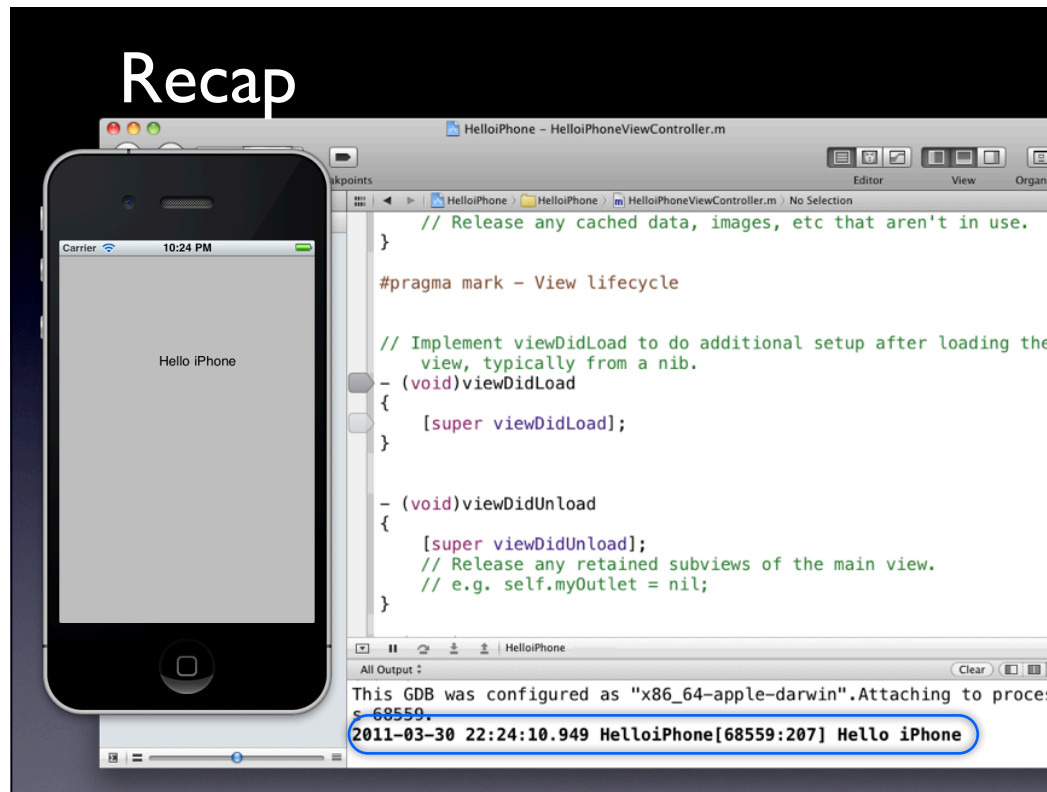
%@ : objectiv-c string

```
NSLog(@" My name is %@, i am %d years old ", @"Michael", 5);
```

Demo Break Point

Recap



The screenshot displays the Xcode IDE with the following components:

- Simulator:** A virtual iPhone device showing the text "Hello iPhone" on its screen. The status bar at the top indicates "Carrier", signal strength, Wi-Fi, and the time "10:24 PM".
- Code Editor:** The file "HelloiPhoneViewController.m" is open, showing Objective-C code for the view lifecycle. The code includes a pragma mark for "View lifecycle" and implements the `viewDidLoad` and `viewDidUnload` methods. The `viewDidLoad` method calls `[super viewDidLoad];`, and `viewDidUnload` calls `[super viewDidUnload];` and releases any retained subviews.
- Console:** The "All Output" window shows the following log message: `2011-03-30 22:24:10.949 HelloiPhone[68559:207] Hello iPhone`. This message is highlighted with a blue circle.

Simulator



Simulate

Single Finger

Tap

Drag

Two Fingers

Zoom in/out (click + alt)

Move (click + alt + shift)

Simulator - Support I

支援功能	解說
操作模擬器	
逆時針轉模擬器	command + 方向鍵 左
順時針轉模擬器	command + 方向鍵 右
搖晃模擬器	control + command + Z
鎖定螢幕	command + L
解鎖螢幕	任意鍵 或 滑動螢幕上解鎖鈕
回到主畫面	shift + command + H
傳送記憶體不足之訊息	因為模擬器的記憶體很多不會因為記憶體不足而收到不足的訊息。透過這個機制可以讓開發者知道收到記憶體不足的訊息時有沒有依開發者的預期而行為
改變手機狀態	切換一般和電話中這兩個狀態。可使開發者知道程式的介面在電話中狀態時變成如何
模擬外接鍵盤	關掉軟體鍵盤，使用使接鍵盤
手勢	
Tap(擊點)	滑鼠click
Touch and Hold (按住)	按住滑鼠鍵
Double Tap (雙擊點)	Double Click
Swipe (刷)	從一點按住滑鼠鍵，移動到另一點並放開
Flick (輕拂)	從一點按住滑鼠鍵，快速地移動到另一點並放開

Simulator - Support 2

支援功能	解說
Drag	從一點按住滑鼠鍵，移動到另一點
Pinch	按下 Option 不動，會看到兩個圓圈 移動兩圓圈代表移動Touch的點 按下shift 可以移動Pinch中心 按下滑鼠鍵，移動兩圓圈到結束的位置再放 開滑鼠鍵和 Option
安裝程式	Xcode 自動把專案安裝在 Simulator
移除程式	長按某個程式的icon，會看到所有程式在扭 動並出現關掉的小icon，按下關掉icon就會 移除程式。按下 Home Button 就會停止扭 動。
重製	回復到原來的設定，資料也會不見。Reset Content and Setting
Core Location	只會回傳一個坐標點。經度122.0307，緯 度 37.3317，就是Apple 公司所在位置。大 概有 100 公尺的誤差。
聲音播放	支援 System Sound Service 除了振動。也 有AVAudioPlayer的聲音播放
多媒體播放	MPMoviePlayerController

Demo Simulator

Questions?