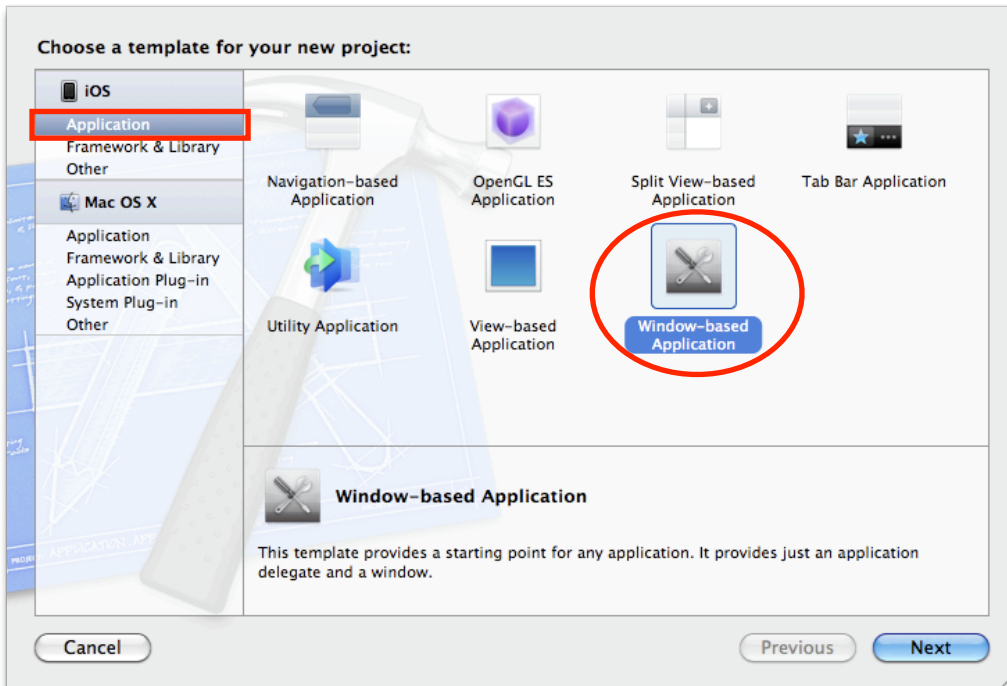


# Lab SimpleNav

Step 1. 開啓一個新的project, 選擇window-based application

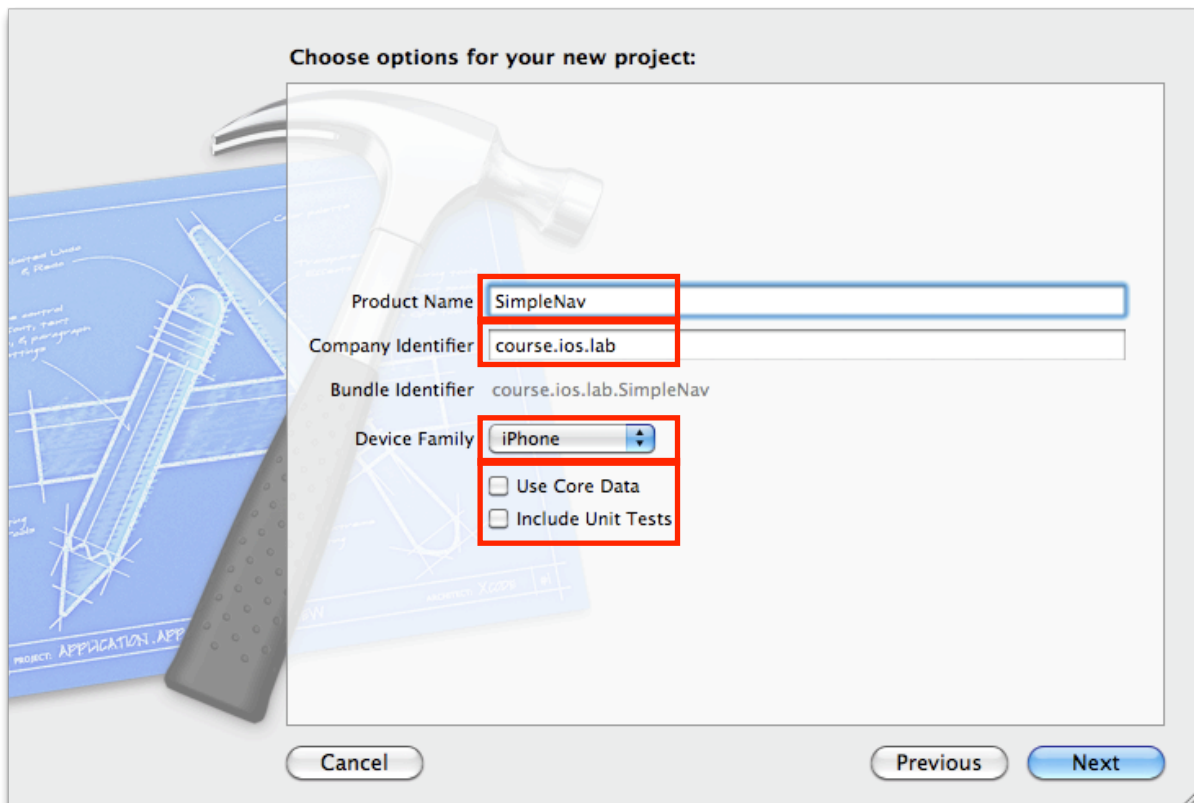


Step 2. 並將此專案命名為 **SimpleNav**

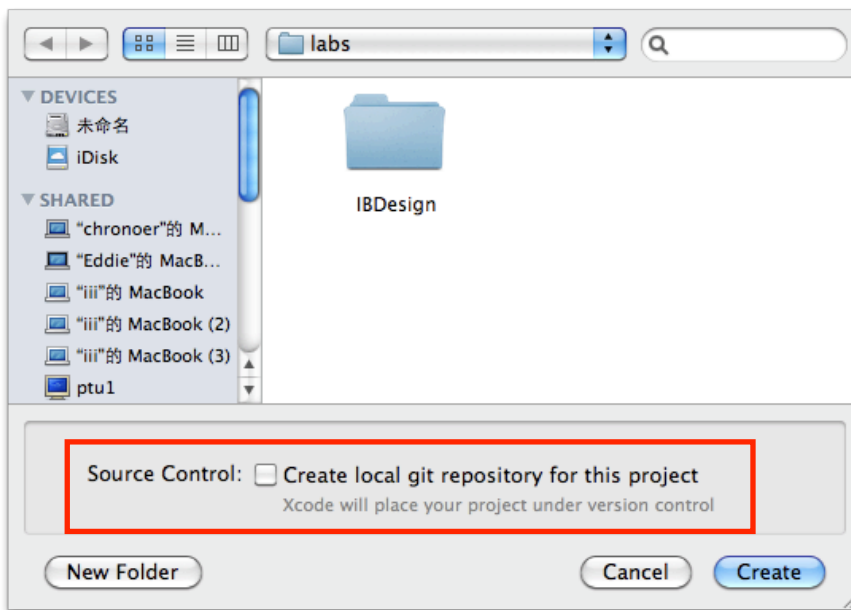
Company Identifier是填入Bundle的名稱,在此統一填入**course.ios.lab** (也可自行填入)

Device Family選擇**iPhone**

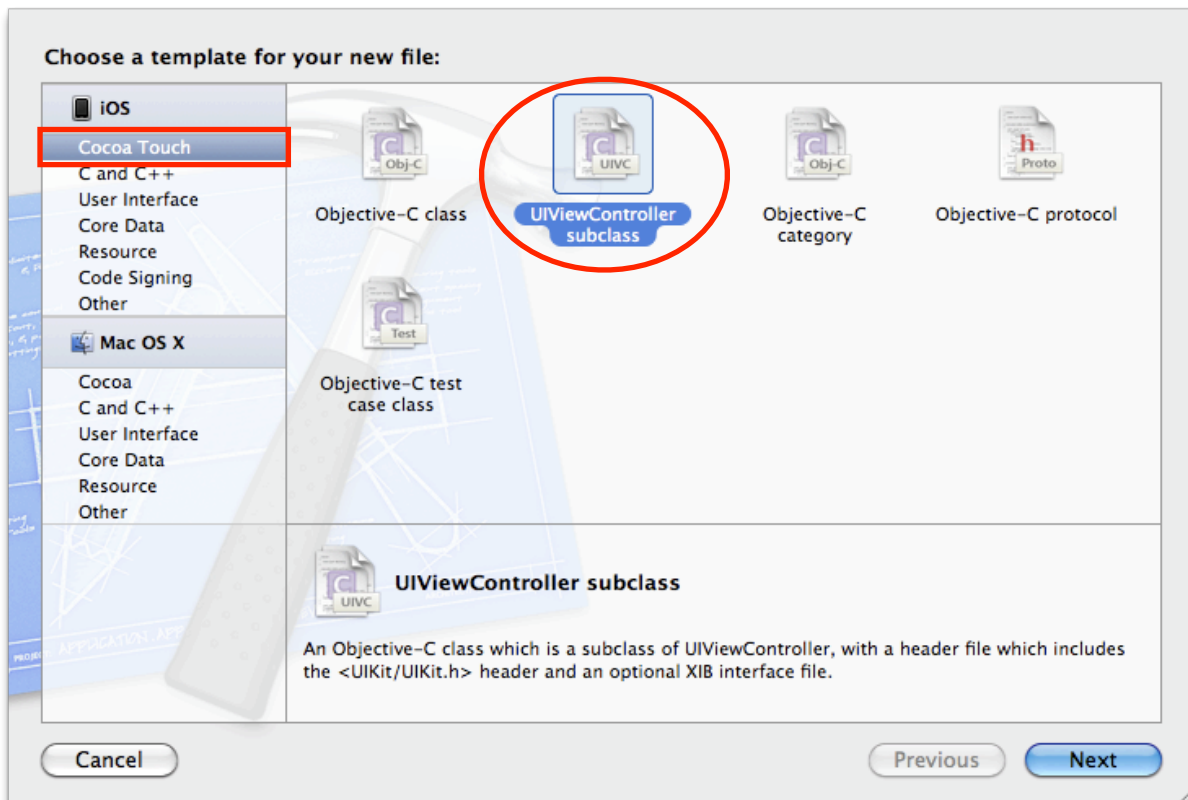
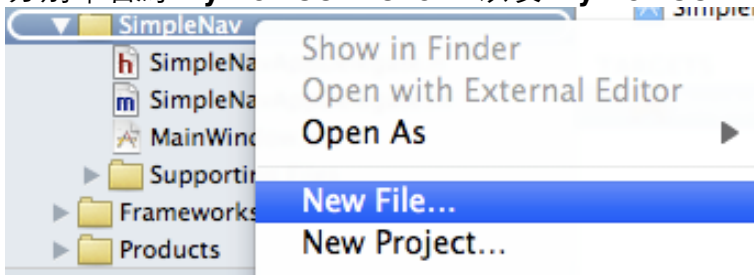
在這專案不勾選Use Core Data,而Include Unit Tests是做語意測試用,在此我們統一不勾選



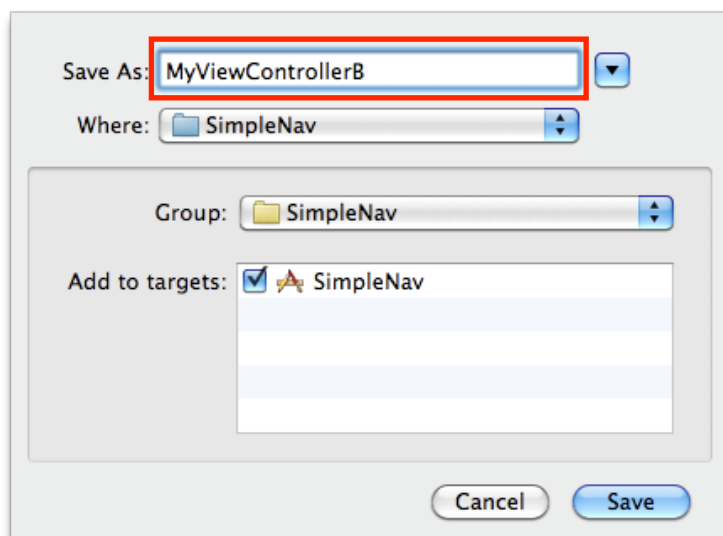
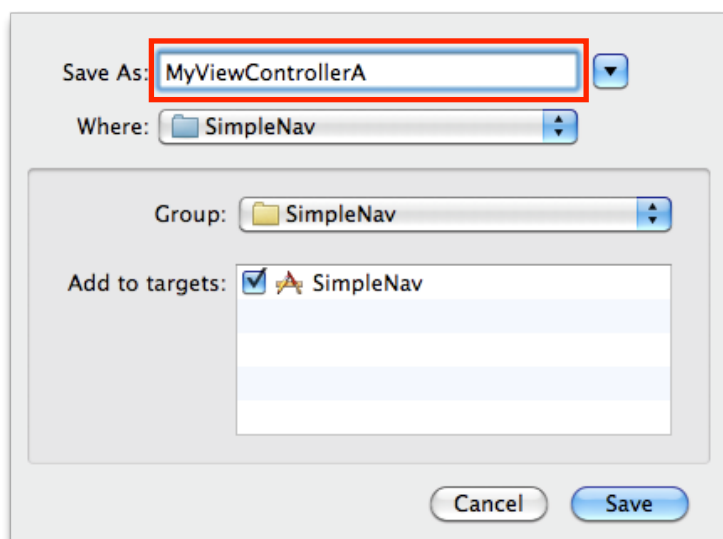
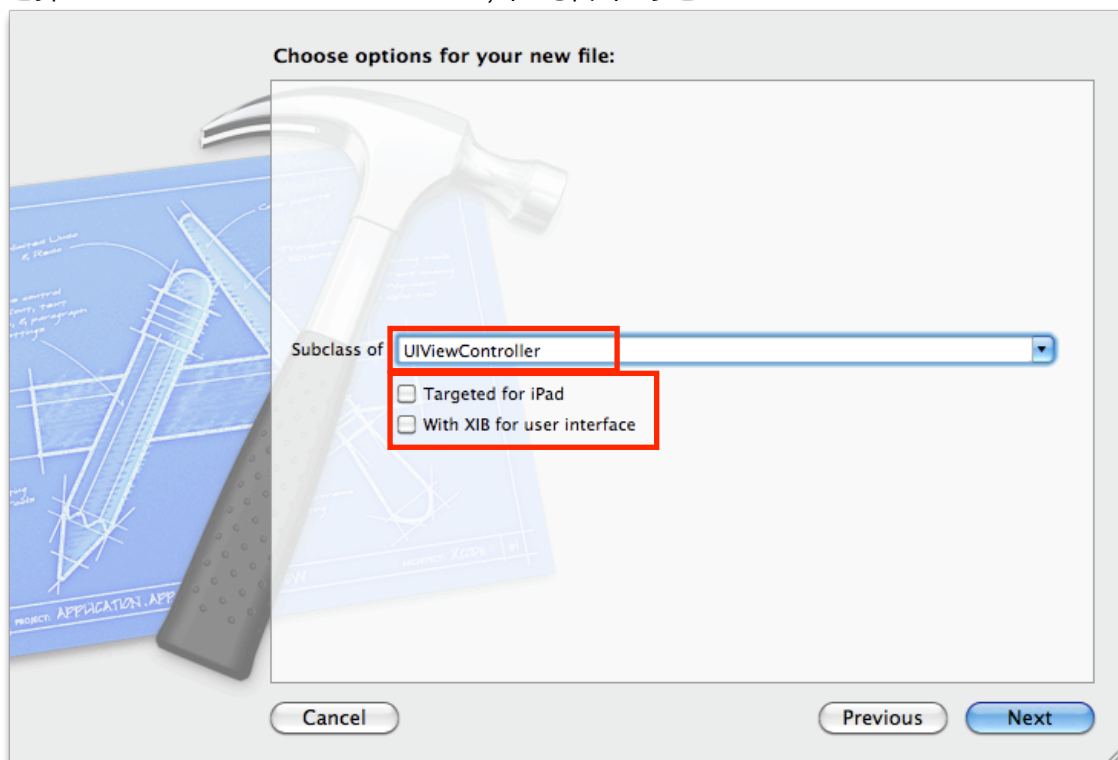
Step 3.選擇存檔的位置, 在此我們不做version control,統一不勾選Create local git repository for this project



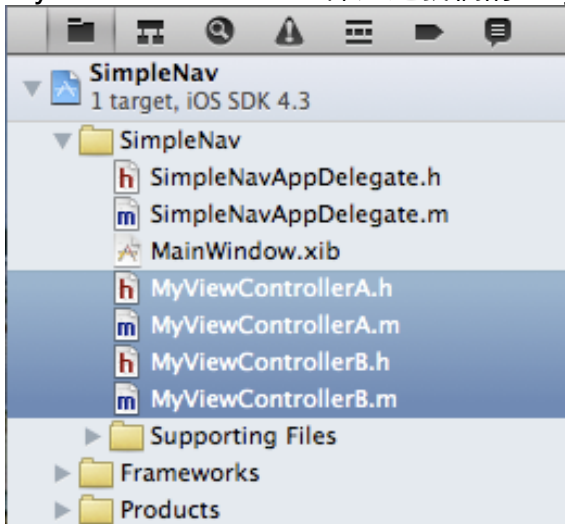
Step 4. 我們新增兩個 UIViewController, 在 SimpleNav 資料夾點右鍵 > new file 選擇iOS 裡Cocoa Touch Class分類的UIViewController subclass  
分別命名為 **MyViewControllerA** 以及 **MyViewControllerB**



選擇 Subclass of UIViewController, 在此皆不勾選 With XIB for user interface



四個檔案包括 MyViewControllerA.h MyViewControllerA.m MyViewControllerB.h MyViewControllerB.m 都加進我們的Project裡了



Step 5. 在 SimpleNavAppDelegate.m , 我們打算將 MyViewControllerA 當作第一張 viewController,所以先import MyViewControllerA.h

```
#import "SimpleNavAppDelegate.h"
#import "MyViewControllerA.h"
```

```
@implementation SimpleNavAppDelegate
```

```
@synthesize window=_window;
```

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    UINavigationController* myNav = [[UINavigationController alloc]
init];
    MyViewControllerA *myViewConA = [[MyViewControllerA alloc] init];
    [myNav pushViewController:myViewConA animated:YES];
    [self.window addSubview:myNav.view];

    [self.window makeKeyAndVisible];
    return YES;
}
```

接著在 - (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions 中, 建立一個 UINavigationController 物件.,接著建立一個 myViewControllerA 物件

```
UINavigationController* myNav = [[UINavigationController alloc] init];
MyViewControllerA* myViewConA = [[MyViewControllerA alloc] init];
```

然後我們把 剛建立好的 myViewConA push到 myNav 這個navigation controller 之中,當作第一個畫面

```
[myNav pushViewController:myViewConA animated:YES];
```

最後我們把 myNav 的 view 加到 window中.

```
[self.window addSubview:myNav.view];
```

如此, window , navigation controller以及 view controller 就設定好了

Step 6. 我們接著要在 MyViewControllerA 使用 CodeUIViewController lab中學過的方法, 不透過 interface builder , 自定上面的view,我們先定義一個 myButton, 以及相對應的動作 nextView, 目的是當我按下 myButton, 會跳至下一個 viewController.

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

@interface MyViewControllerA : UIViewController {
    UIButton *myButton;
}

-(void) nextView;

@end
```

Step 7. 在 MyViewControllerA.m, 我們要把 myViewControllerB 當作接在 myViewControllerA 的下一張 view, 所以我們先 import myViewControllerB.h

```
#import "MyViewControllerA.h"
#import "MyViewControllerB.h"
```

Step 8. 在MyViewControllerA.m 實作constructor, destructor, 其中dealloc已由系統產生好了

```
-(id)init
{
    if(self = [super init])
    {}
    return self;
}

- (void)dealloc
{
    [super dealloc];
}
```

Step 9. 找到 loadView, 將Mark去掉, 我們實作UIView, UILabel, UIButton 物件

```
- (void)loadView
{
    UIView* myView=[[UIView alloc] initWithFrame:[UIScreen
 mainScreen].applicationFrame];
    UILabel* myLabel = [[UILabel alloc] initWithFrame:CGRectMake(80,
 100, 300, 50)];
    [myLabel setText:@"MyViewController A"];

    myButton =[UIButton buttonWithType:UIButtonTypeRoundedRect];
    myButton.frame = CGRectMake(80, 200, 150, 50);
    [myButton setTitle:@"Next View" forState:UIControlStateNormal];
    [myButton addTarget:self action:@selector(nextView)
forControlEvents:UIControlEventTouchUpInside];

    [myView addSubview:myLabel];
    [myView addSubview:myButton];
    self.view = myView;
    [myView release];
}
```

在這段程式中

開一個 UIView 形態的 view, 當作最底層的 view

```
UIView* myView=[[UIView alloc] initWithFrame:[UIScreen mainScreen].applicationFrame];
```

直接將 myLabel 初始化, 而且直接用 initWithFrame 將它的位置大小定義出來

```
UILabel* myLabel = [[UILabel alloc] initWithFrame:CGRectMake(80, 100, 300, 50)];
```

將 myButton 初始化, 並且將它的 type 設成 UIButtonTypeRoundedRect. 並將 myButton 在 myView 上的位置和大小定義出來

```
myButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
```

```
myButton.frame = CGRectMake(80, 200, 150, 50);
```

設定 myButton 以及 myLabel 上的文字, 並 將 myAction 與 myButton 的 touch down 動作連結.

```
[myButton setTitle:@"Next View" forState:UIControlStateNormal];
```

```
[myLabel setText:@"myViewController A"];
```

```
[myButton addTarget:self action:@selector(nextView)  
forControlEvents:UIControlEventTouchUpInside];
```

接著我們把 myLabel 以及 myButton 加到 myView 裡

```
[myView addSubview:myLabel];
```

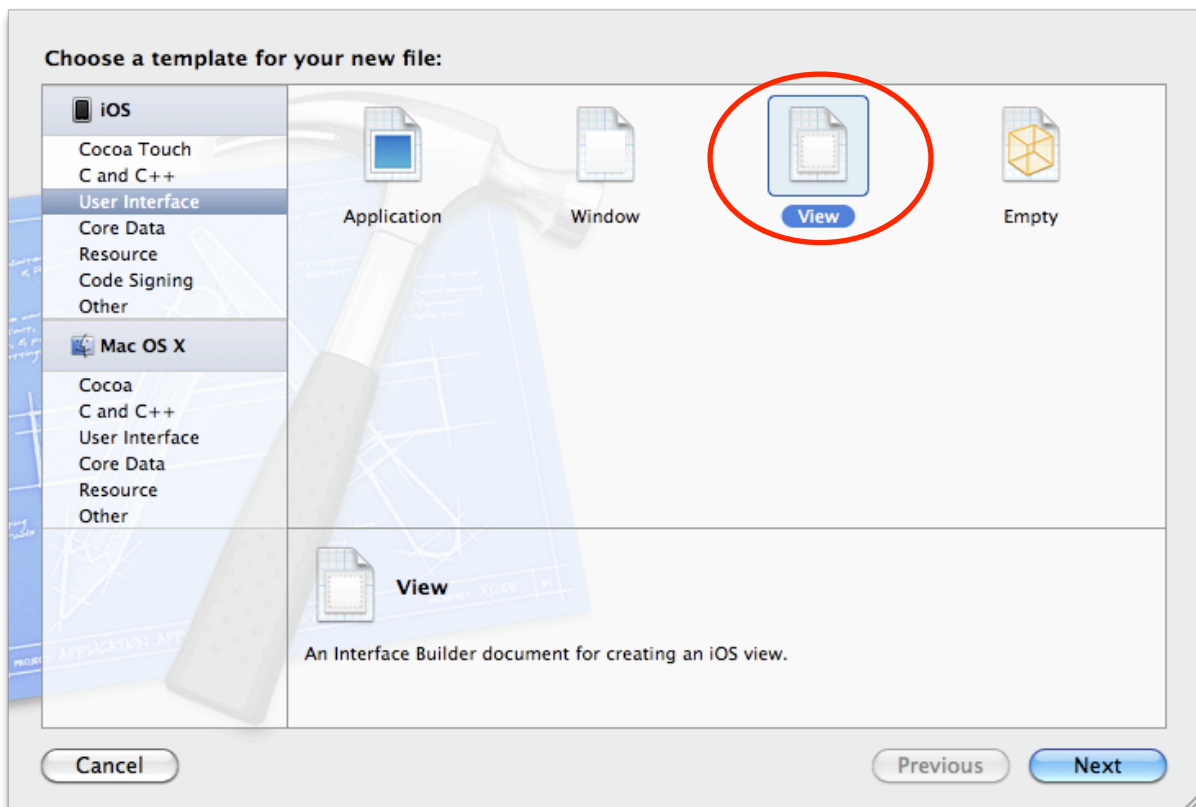
```
[myView addSubview:myButton];
```

我們將 myView 傳給這個 UIViewController 管理, 然後就可以把 myView release 掉了.

```
self.view = myView;
```

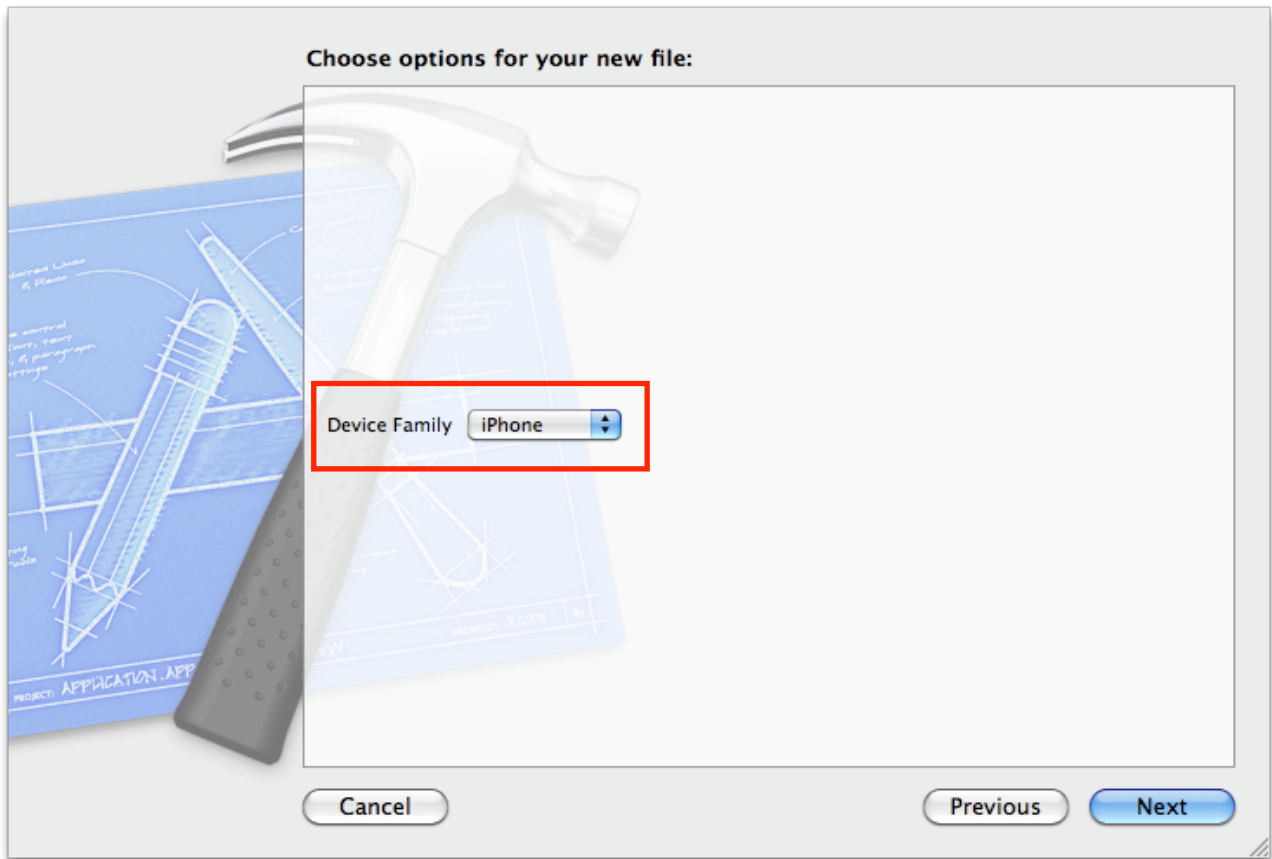
```
[myView release];
```

Step 10. 在實作 nextView 這個 action 之前, 我們先實作 MyViewControllerB, 我們的 MyViewControllerB 使用 IBViewController 學過的方法來實作. 首先在 SimpleNav 資料夾點右鍵選擇 New File 選擇 iOS 的 User Interfaces > View

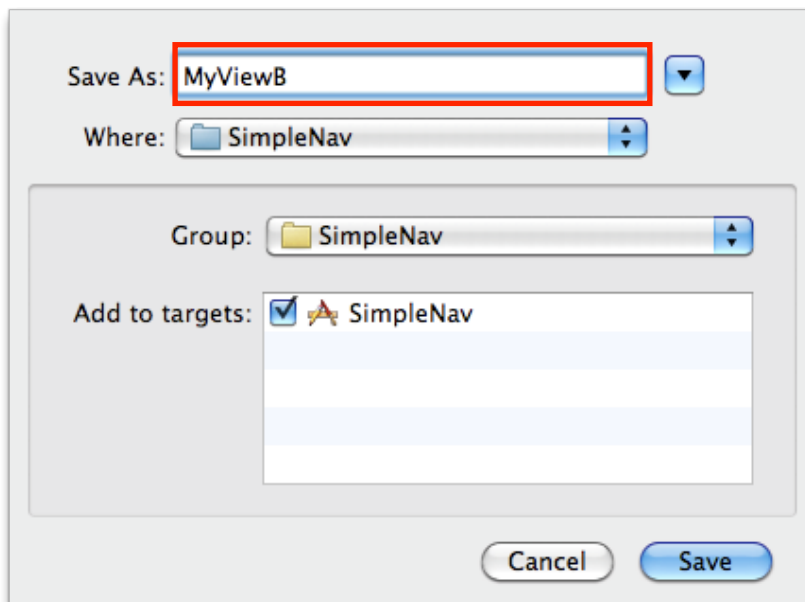




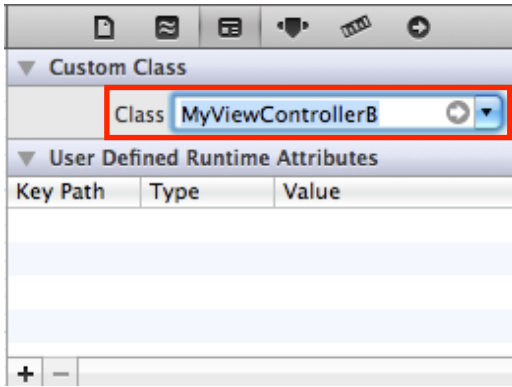
Step 11. Device Family 選擇iPhone



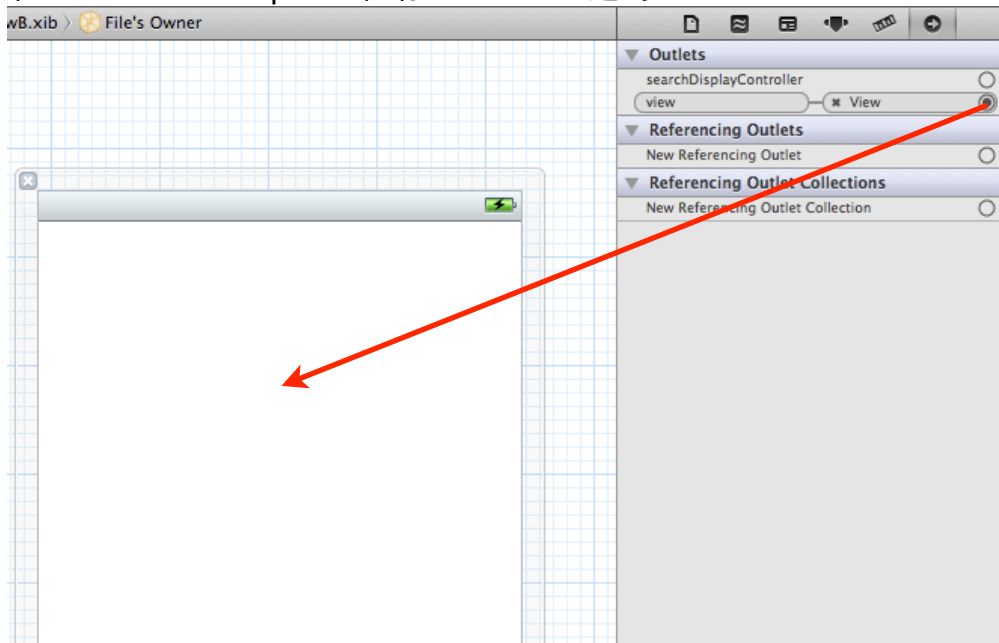
Step 12. 命名為 MyViewB



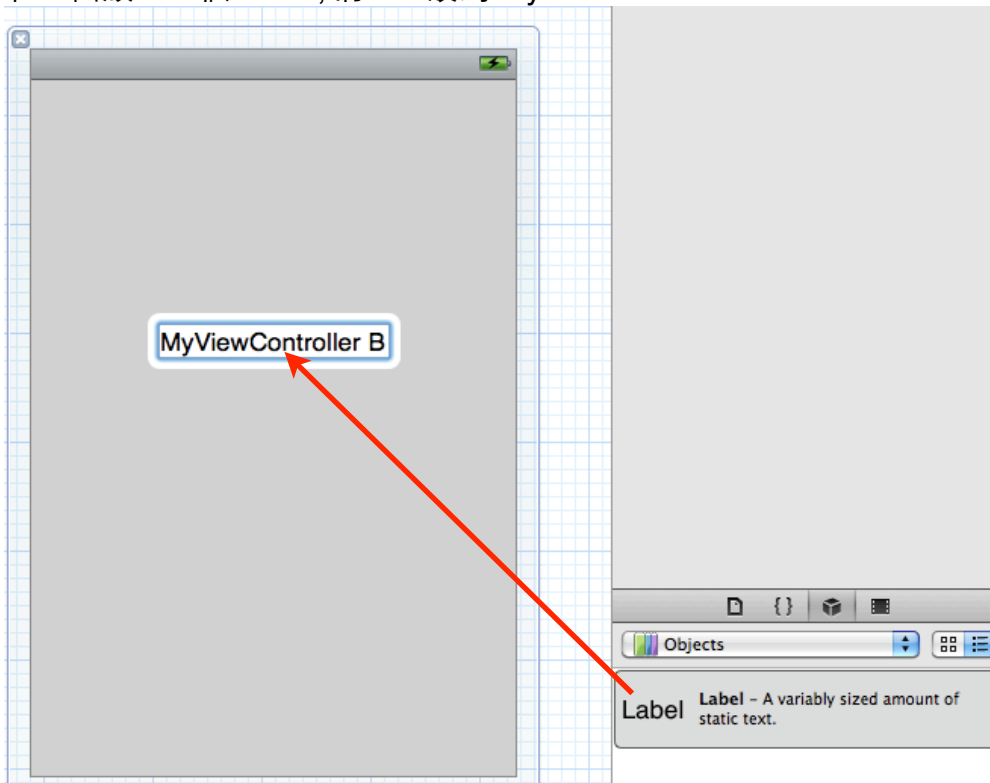
Step 13. 開啓 MyViewB.xib , 將 File's Owner 設定成 MyViewControllerB



在Connections Inspector裡將connection 連到 View



在上面放上一個 Label, 將Title設為 MyViewController B





Step 14. 在 MyViewControllerA.m 裡, 我們要實作 nextView 這個 action

```
#import "MyViewControllerA.h"  
#import "MyViewControllerB.h"
```

```
@implementation MyViewControllerA
```

```
-(void) nextView  
{  
    MyViewControllerB* myViewConB = [[MyViewControllerB alloc]  
initWithNibName:@"MyViewB" bundle:nil];  
    [self.navigationController pushViewController:myViewConB  
animated:YES];  
    [myViewConB release];  
}
```

建立MyViewControllerB,使用initWithNibName :@"MyViewB" 連結 MyViewB.

```
myViewControllerB* myViewConB = [[myViewControllerB alloc]  
initWithNibName:@"MyViewB" bundle:nil];
```

將 myViewConB push到 self 的 navigationController之中, 然後 release myViewConB.

```
[self.navigationController pushViewController:myViewConB  
animated:YES];  
[myViewConB release];
```

Step 15. Run (⌘+R)

當我們按下 Next View Button, 就會切到下一個頁面, 按上方的 Back 可以退回上一頁

