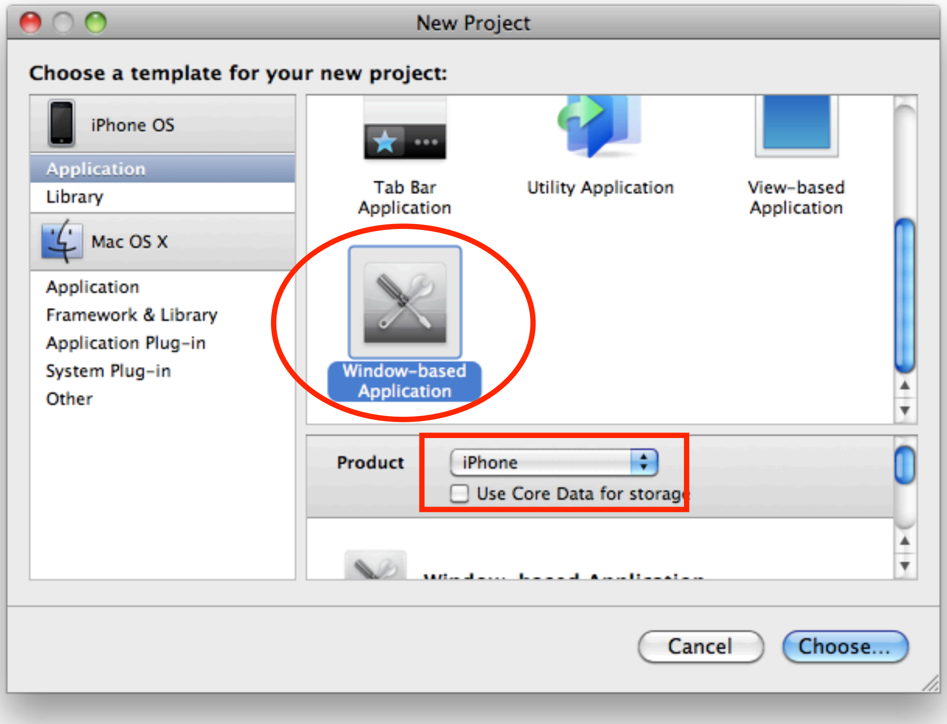
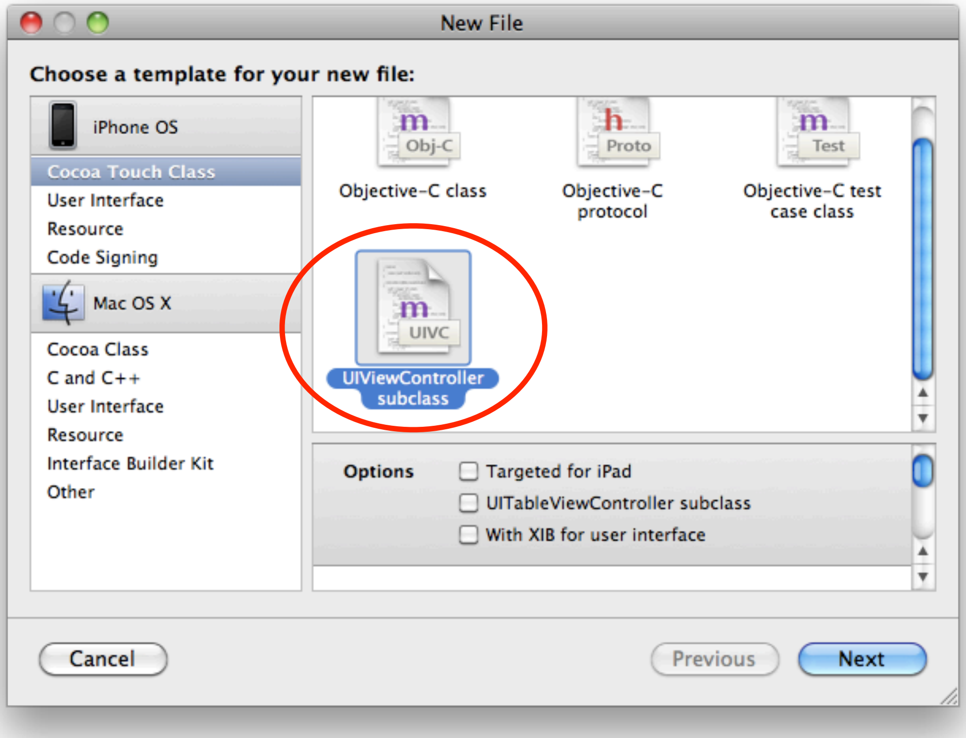
Lab simple_nav - navigation bar

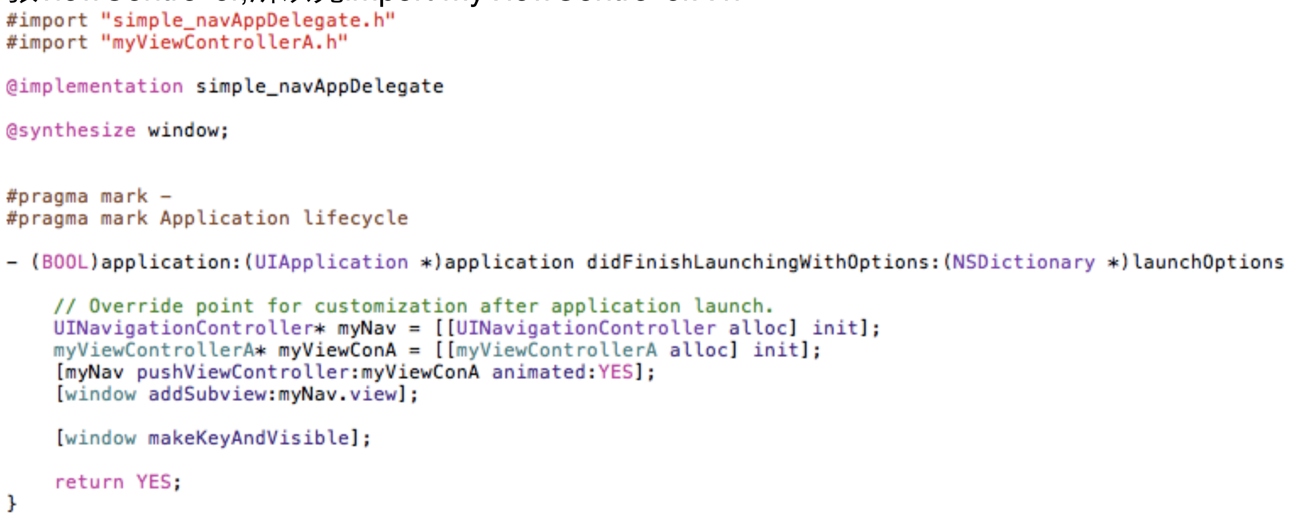
Step1. 在File開啓一個新的project, 選擇 window based application, 將project取名為 simple_nav



Step2. 我們新增兩個 UIViewController , file > new file 選擇UIViewController subclass, 分別命名為 myViewControllerA 以及 myViewControllerB



Step 3. classes > simple_nav_AppDelegate.m , 我們打算將 myViewControllerA 當作第一張viewController,所以先import myViewControllerA.h.



```
#import "simple_navAppDelegate.h"
#import "myViewControllerA.h"

@implementation simple_navAppDelegate

@synthesize window;


#pragma mark -
#pragma mark Application lifecycle

- (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];
    [window addSubview:myNav.view];

    [window makeKeyAndVisible];

    return YES;
}
```

接著在 application didFinishLaunchingWithOptions: 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

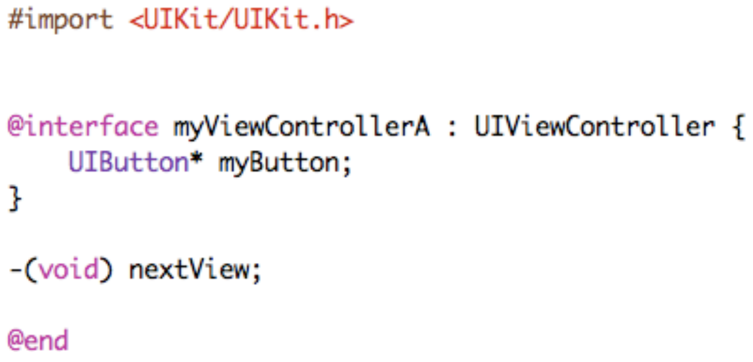
```
[myNav pushViewController:myViewConA animated:YES];
```

最後我們把 myNav 的 view 加到 window中.

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

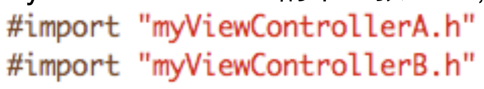
如此, window , navigation controller以及 view controller 就設定好了

Step 4. 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.



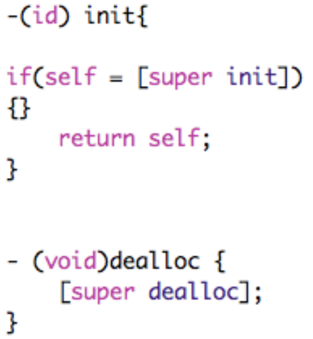
```
#import <UIKit/UIKit.h>


@interface myViewControllerA : UIViewController {
    UIButton* myButton;
}

-(void) nextView;

@end
```

Step 5. classes > 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



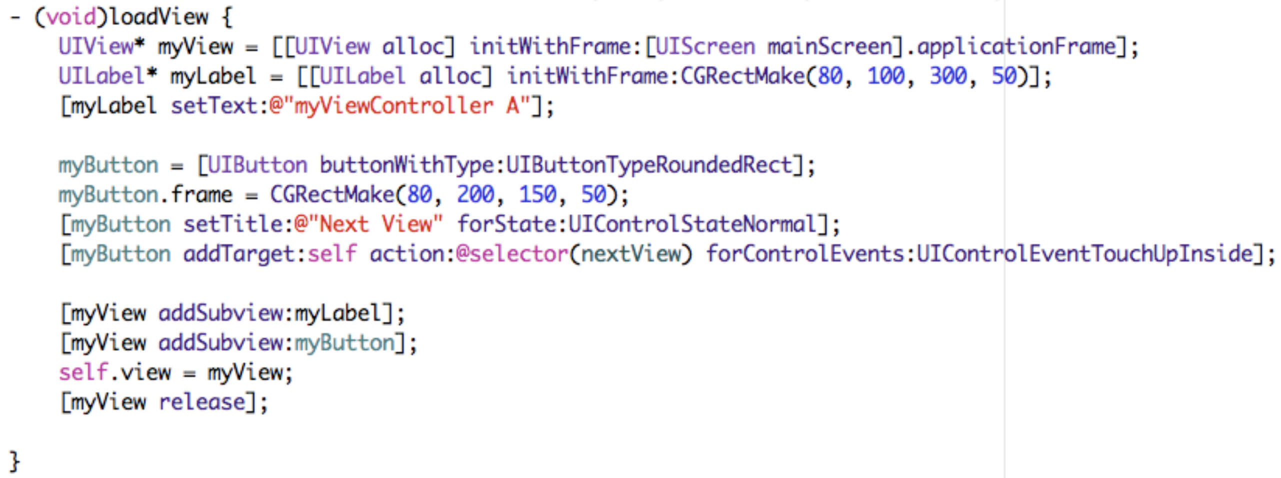
```
#import "myViewControllerA.h"
#import "myViewControllerB.h"
```

Step 6. classes > 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 7. 找到 loadView, 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];

}
```

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)];

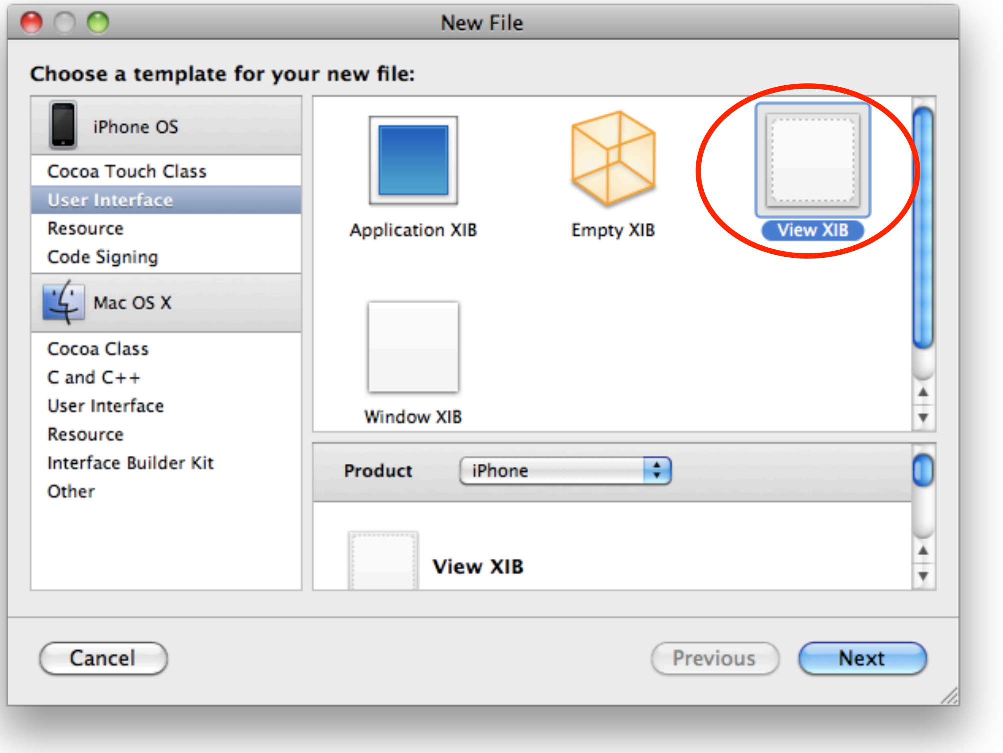
將 myButton 初始化, 並且將它的 type 設成UIButtonTypeRoundedRect.並將myButton在myView上的位置和大小定義出來
myButton =[UIButton buttonWithType:UIButtonTypeRoundedRect];
myButton.frame = CGRectMake(80, 200, 150, 50);

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];

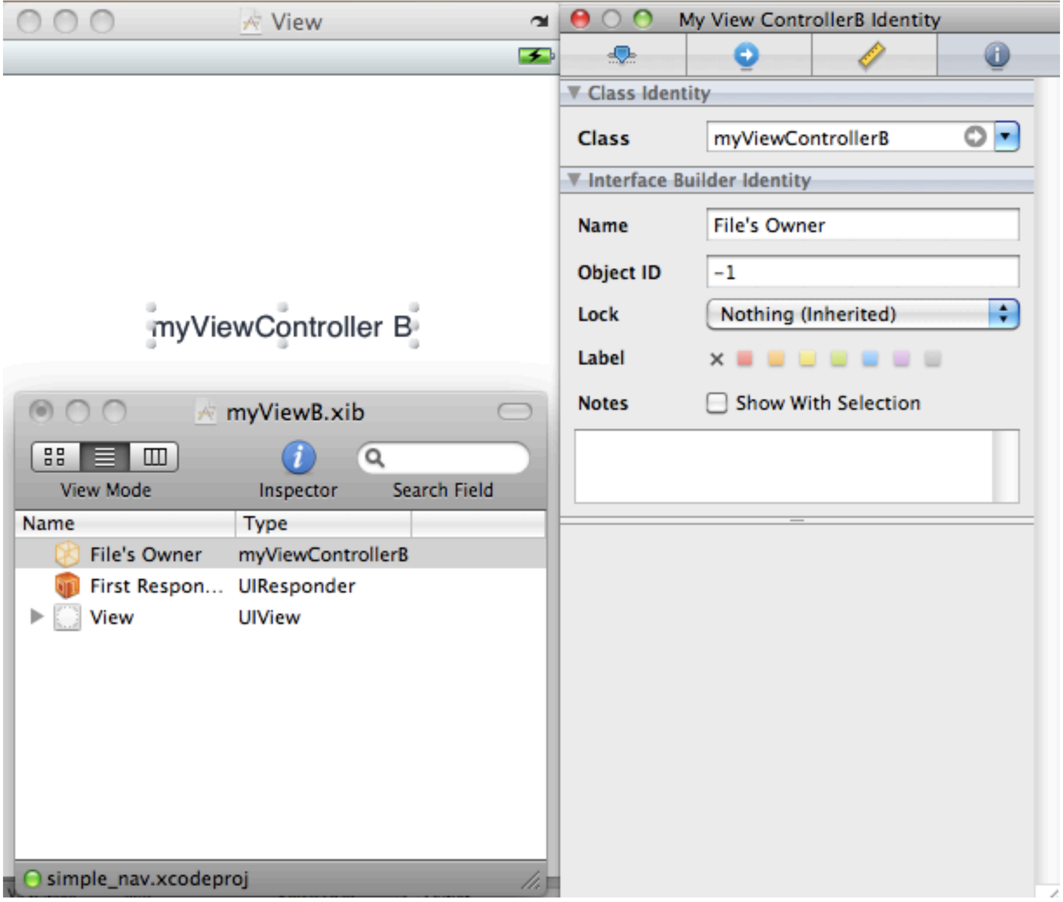
接著我們把myLabel以及myButton加到myView裡
[myView addSubview:myLabel];
[myView addSubview:myButton];

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 8. 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 file > new file 選擇 User Interfaces > view XIB. 命名為 myViewB.



Step 9. Resources > myViewB.xib , 使用 interface builder, 在上面放上一個 label, 文字改成 myViewController B, 記得將 file’s owner 設定成 myViewControllerB



Name	Type
File's Owner	myViewControllerB
First Respon...	UIResponder
View	UIView

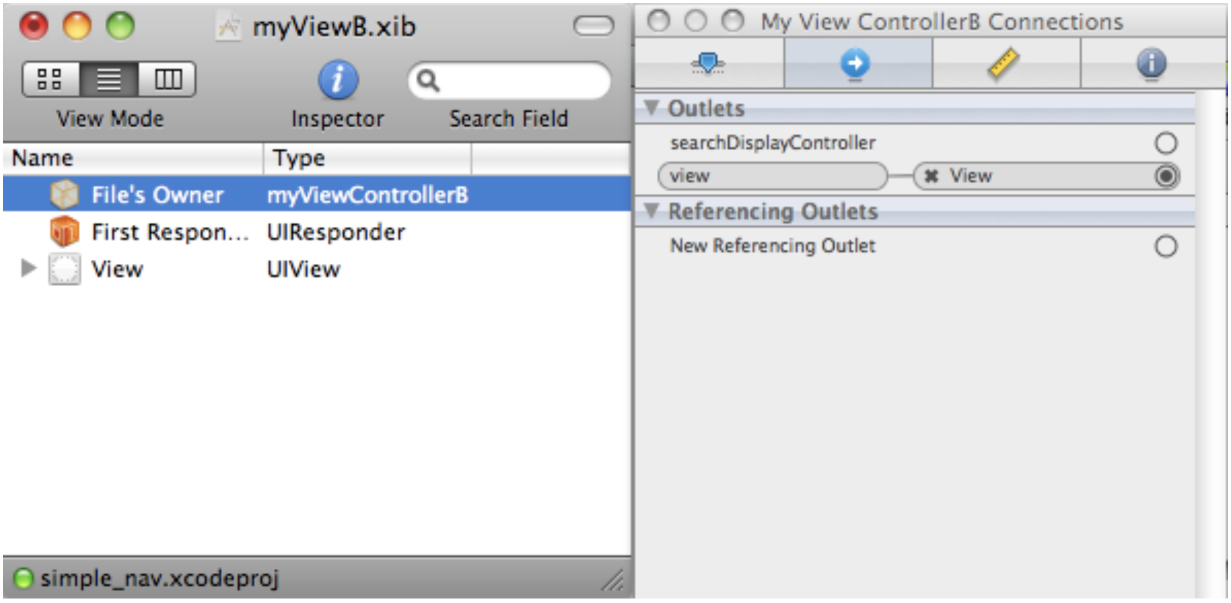
Class: myViewControllerB

Name: File's Owner

Object ID: -1

Lock: Nothing (Inherited)

並且,將connection 連到 view



Name	Type
File's Owner	myViewControllerB
First Respon...	UIResponder
View	UIView

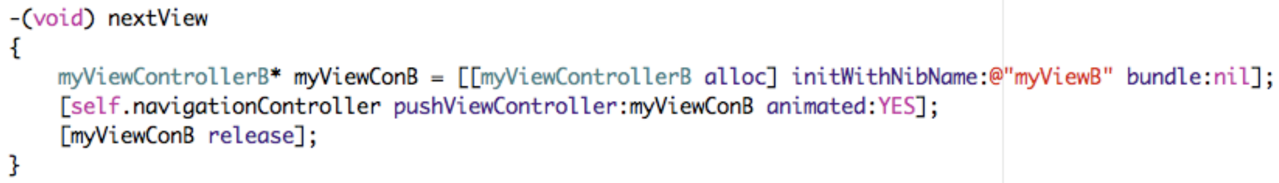
Outlets

- searchDisplayController ○
- view — View ●

Referencing Outlets

- New Referencing Outlet ○

Step 10. classes > myViewControllerA.m, 我們要實作 nextView 這個 action



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

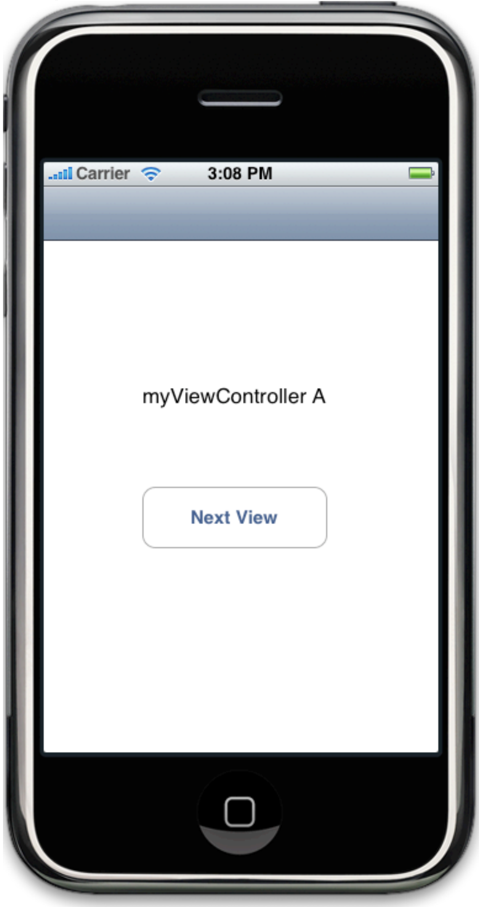
建立myViewControllerB,使用initWithNibName :@”myViewB” 連結 myViewB.

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

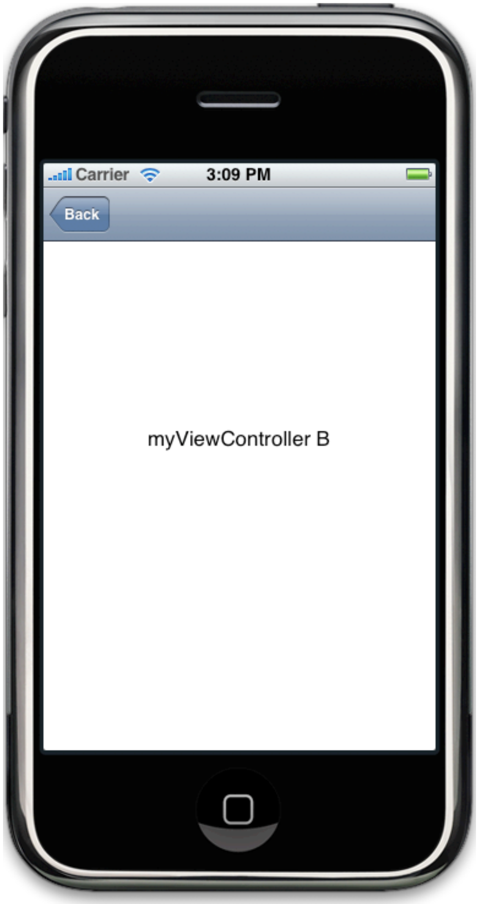
將 myViewController push到 navigationController之中, 然後 release myViewConB.

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

Step 11. build and GO



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


Carrier
3:09 PM
Back
myViewController B