

Exercise 6: Multi-view applications

So far, we have only been concerned with applications that have one view. In this exercise we introduce the navigation controller for creating multi-view applications.

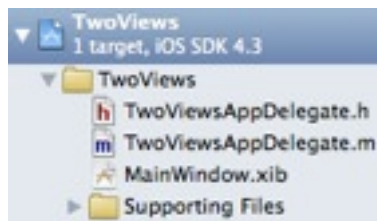
Task 1: Creating a navigation view controller

a) Create a new iOS project, but this time it should be a Windows-based Application:

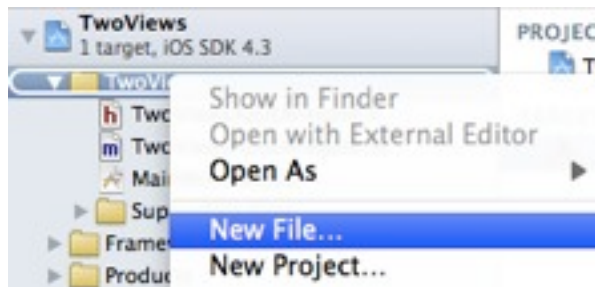


Call the project 'TwoViews'.

Notice that this type of project initially contains no view controllers:



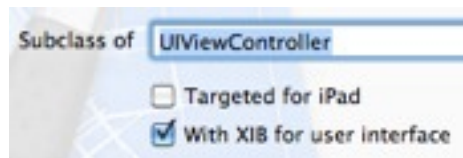
b) Right click on the 'TwoViews' folder and select 'New File...'



Choose the type for the new file:



Make sure it is a subclass of UIViewController, and 'With XIB':



Call the new file 'RootViewController'.

d) In TwoViewsAppDelegate.h, add a UINavigationController:

```
@interface TwoViewsAppDelegate : NSObject <UIApplicationDelegate> {
    UINavigationController *navController;
}

@property (nonatomic, retain) IBOutlet UIWindow *window;
@property (nonatomic, retain) IBOutlet UINavigationController *navController;

@end
```

e) In TwoViewsAppDelegate.m, add a synthesize for the navigation controller, and an extra line to connect up the navigation controller:

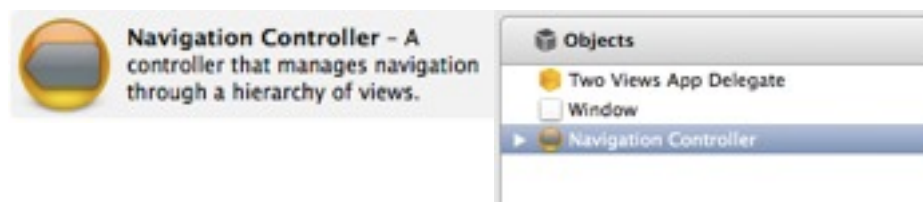
```
@implementation TwoViewsAppDelegate

@synthesize window=_window;
@synthesize navController;

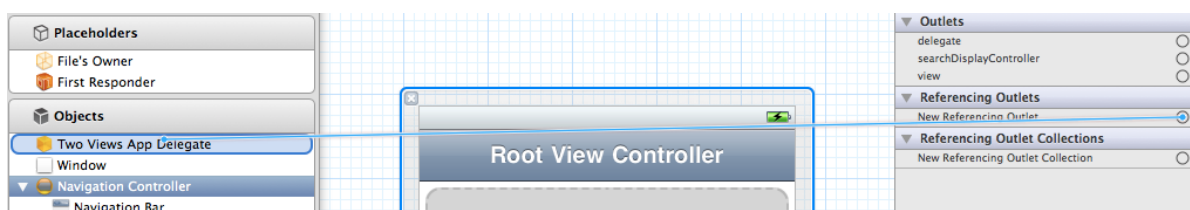
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [self.window addSubview:navController.view];
    [self.window makeKeyAndVisible];
    return YES;
}
```

f) Open MainWindow.xib, then:

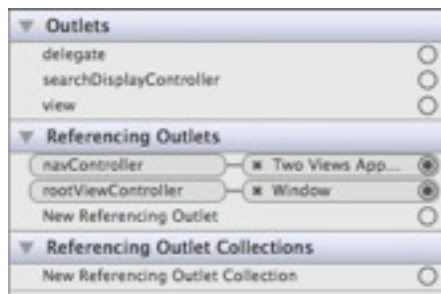
i. Drag a 'Navigation Controller' onto the list of 'Objects'



ii. Click on the Navigation Controller, open the Identify Inspector, then drag a 'Referencing Outlet' to the 'Two Views App Delegate', and select 'navController'.



iii. Drag another 'Referencing Outlet' to 'Window', connecting it to 'rootViewController', so you should have two connections:

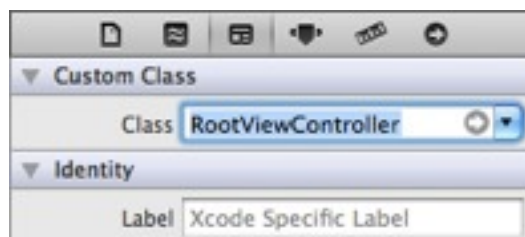


g) Next, also in the MainWindow.xib, click into the Navigation Controller and select the Root View Controller:

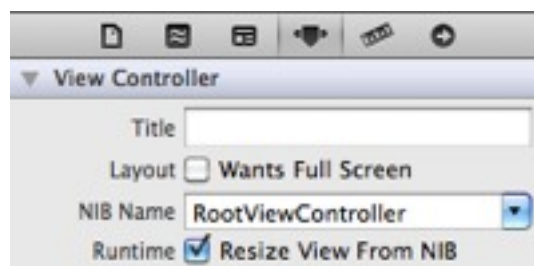


Then:

i. In the Identity Inspector, set the 'Class' as RootViewController:



ii. In the Attributes Inspector, set the 'NIB Name' as RootViewController:



h) Open RootViewController.xib and add a label to center of the view that says 'Root'.

Run the app. You should see a blue navigation bar at the top and your label.

Task 2: Adding a second view controller

- a) Create another new file, also a subclass of UIViewController, and call it 'SecondViewController'.
- b) Open SecondViewController.xib and add a label that says 'We did it!'.
- c) Open RootViewController.xib and remove the label. Add a button, and connect it up to a method in RootViewController:

```
- (IBAction)openSecondView:(id)sender {
    SecondViewController *controller = [[SecondViewController alloc]
initWithNibName:@"SecondViewController" bundle:nil];
    [self.navigationController pushViewController:controller animated:YES];
    [controller release];
}
```

- d) Compile the app and you will find an error. Fix it by adding an import statement for SecondViewController.h.

Run the app, and your button should open the second view.

Task 3: Tables and navigation controllers

Navigation controllers are often used with table view controllers.

- a) Create a new Navigation-based Application called 'TableNav'. This project already contains a UINavigationController and a UITableViewController. You just need to configure the table like in Workshop 7.
- b) In RootViewController.h, declare an NSArray called 'list'.
- c) In RootViewController.m, add the following to viewDidLoad:

```
list = [[NSArray alloc] initWithObjects:@"King Naresuan Monument",
@"Library", @"Textile Museum", nil];
```

- d) RootViewController.m already contains methods for the table view. Update numberOfRowsInSection and cellForRowAtIndexPathIndexPath, such that the table shows the data in the list (same as Workshop 7, Task 3).

Run the app. You should see a table with 3 items.

- e) You can change the title in the navigation bar, by adding the following to viewDidLoad:

```
self.title = @"Mor Nor";
```

Task 4: A detail controller

Let's change the app so that when you click on a row, a new view is created.

- a) Create a new file, a subclass of UIViewController, and call it 'DetailViewController'.

b) Add a UILabel to DetailViewController.xib, and connect it up to a property in the interface file (don't forget @property and @synthesize).

c) Open RootViewController.m, and add an import for DetailViewController.h. Find the didSelectRowAtIndexPath method and replace it with:

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:
(NSIndexPath *)indexPath {
    DetailViewController *detailViewController = [[DetailViewController
alloc] initWithNibName:@"DetailViewController" bundle:nil];
    [self.navigationController pushViewController:detailViewController
animated:YES];
    [detailViewController release];
    detailViewController.label.text = [list objectAtIndex:indexPath.row];
}
```

Run the app. You should be able to click on a row and it will open the new view.

Task 5: UIImageView

The UIImageView is an object that displays an image on the screen.

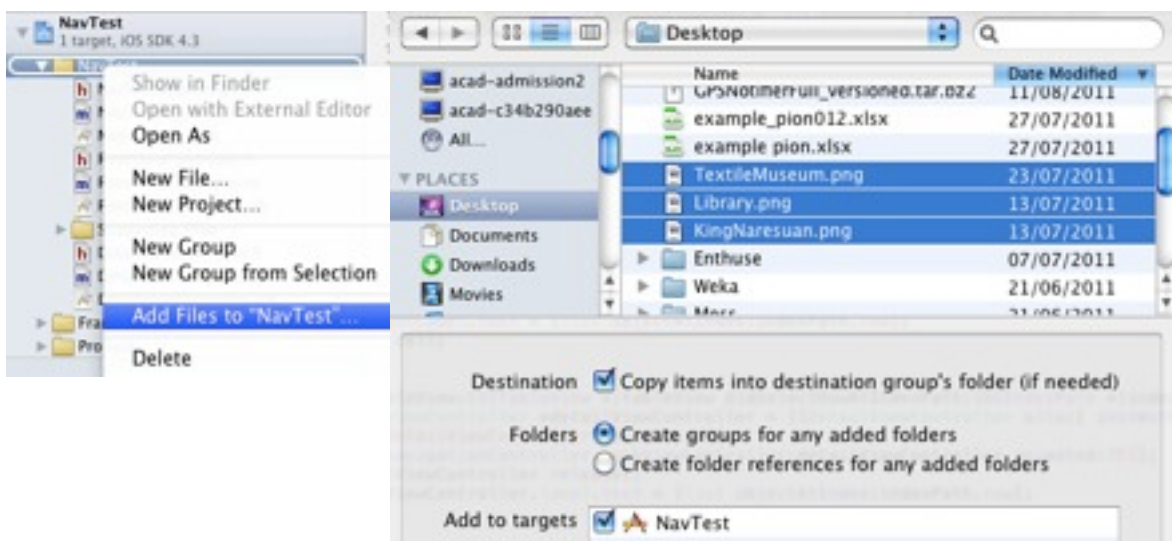
a) Find the 'Image View' in the Objects Library and drag it into the view in DetailViewController.xib.

b) In DetailViewController.h, create a property for the UIImageView. In DetailViewController.m, synthesize the property.

c) Open RootViewController.m and go to the didSelectRowAtIndexPath method. Add the following code at the end of the method:

```
detailViewController.imageView.image =
    [UIImage imageNamed:@"KingNaresuan.png"];
```

d) Download and unzip the images in [workshop8images.zip](#). Add these files to the project:



Run the app and you should see an image in the detail.

e) Modify the code in the `didSelectRowAtIndexPath` method so that it sets the correct image (KingNaresuan.png, Library.png or TextileMuseum.png).

Task 6: Contacts 2

In Exercise 5 you created a contacts application. Create a new Navigation-based Application called 'Contacts2'. The app should be the same as Contacts, except that when you click on a row it will open a `DetailViewController` (not an `UIAlertView`) showing the contact information.