

iPhone:
Once Upon a Time....

Mark Brindle
9 January 2012

Storyboard

Storyboard

- Life before storyboard

Storyboard

- Life before storyboard
- What is a storyboard?

Storyboard

- Life before storyboard
- What is a storyboard?
- How do we use it?

Storyboard

- Life before storyboard
- What is a storyboard?
- How do we use it?
- What does it do for us?

Storyboard

- Life before storyboard
- What is a storyboard?
- How do we use it?
- What does it do for us?
- Things to be aware of

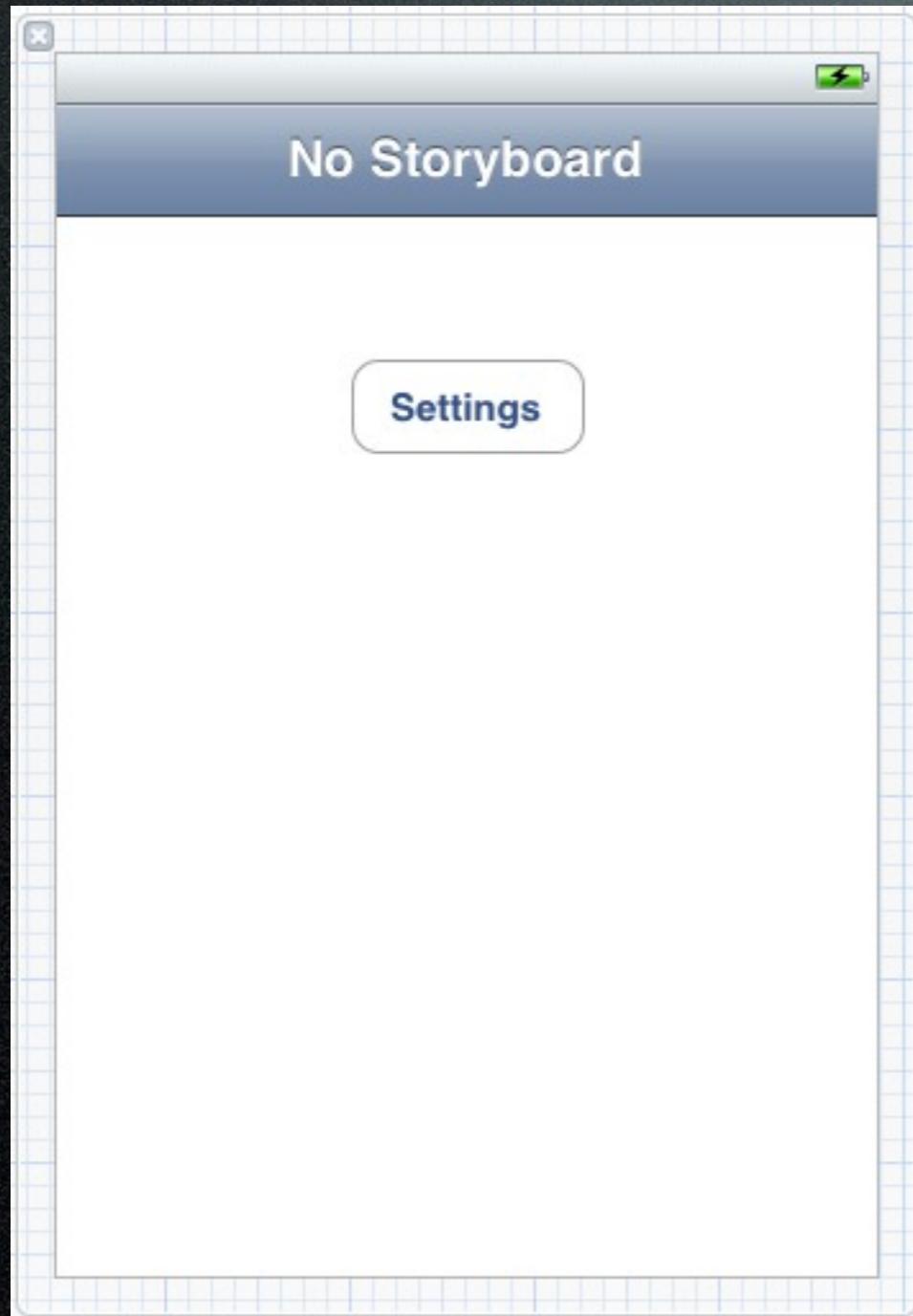
Life before storyboard

Life before storyboard

Separate screens; Independent XIB's

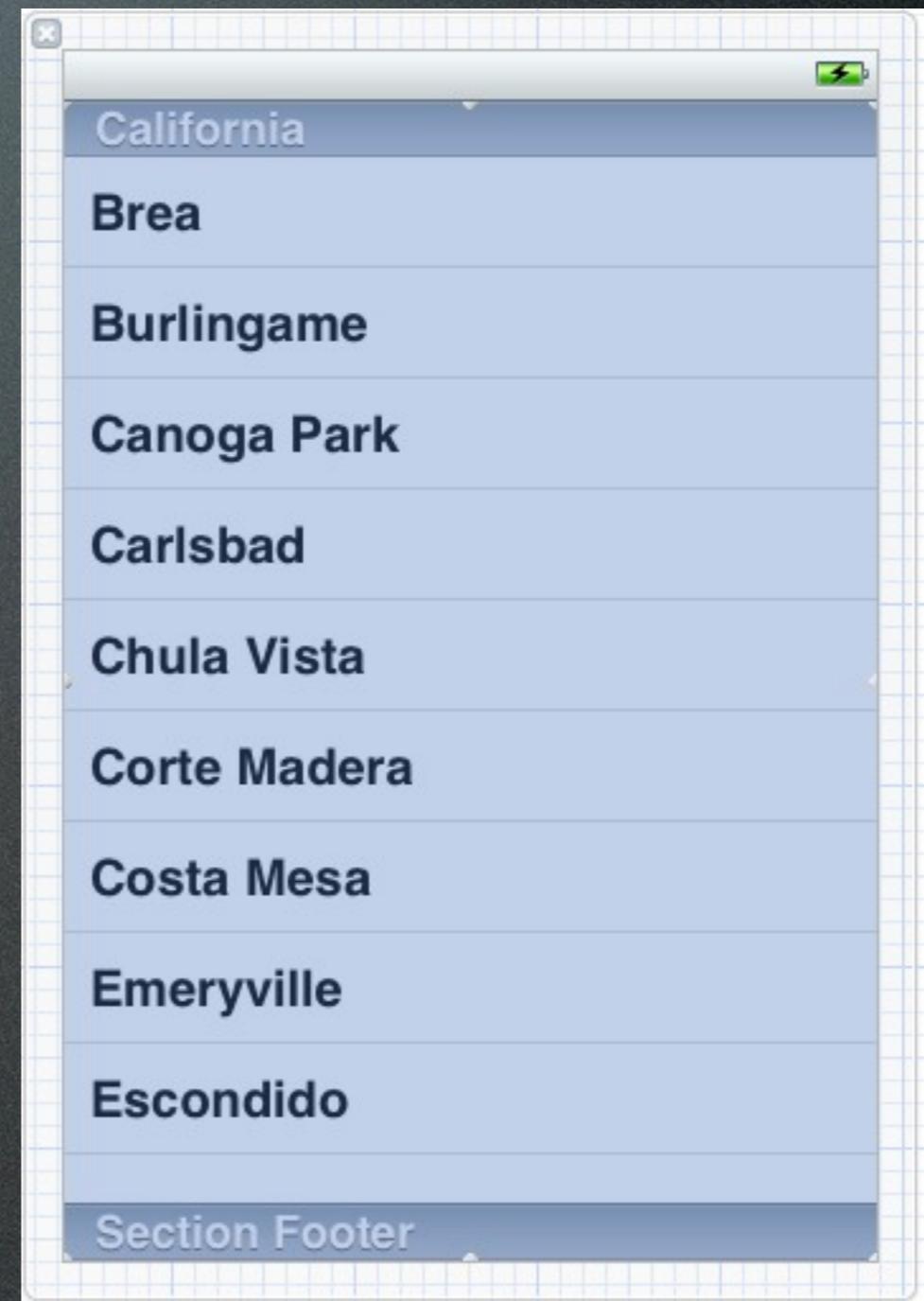
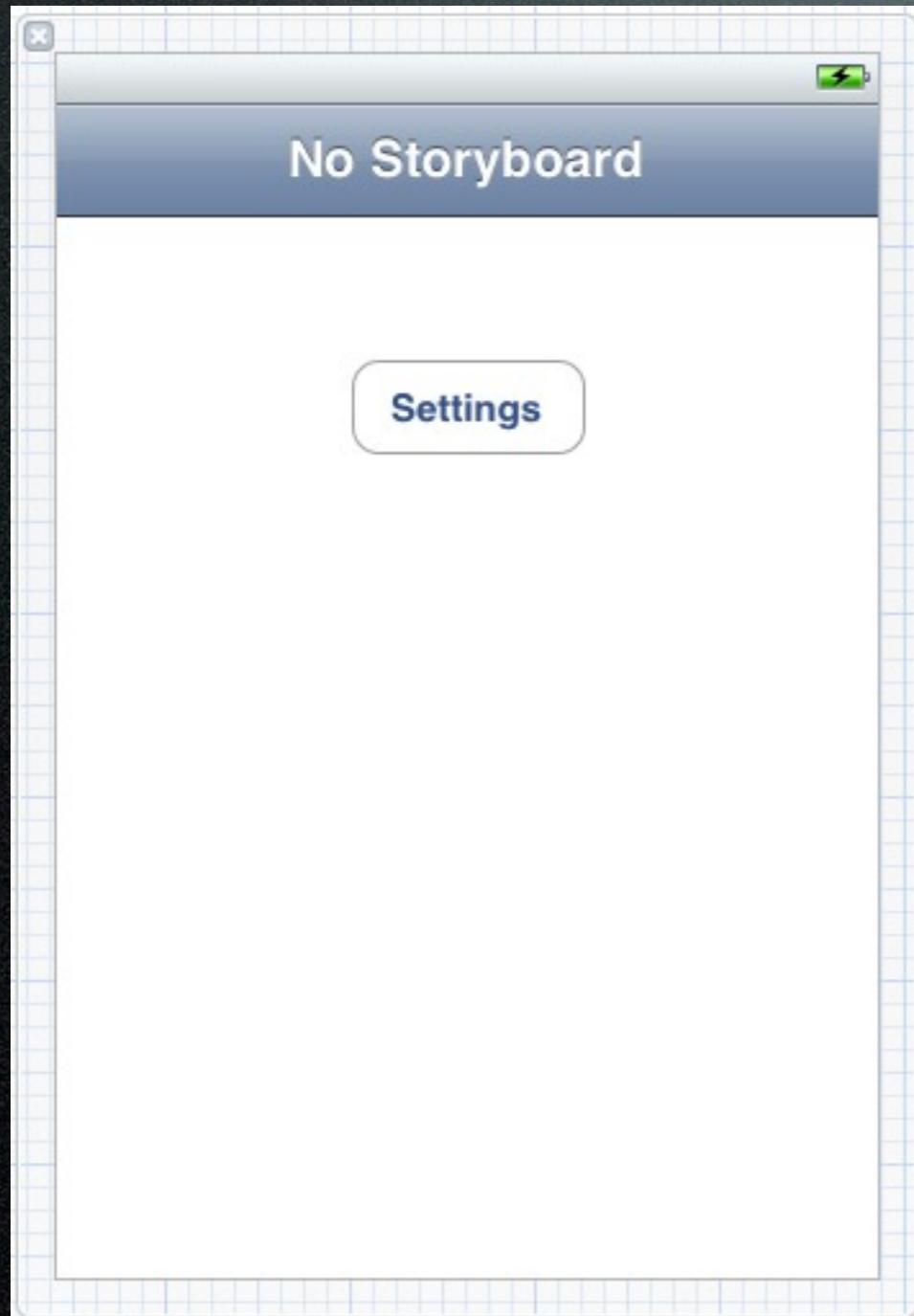
Life before storyboard

Separate screens; Independent XIB's

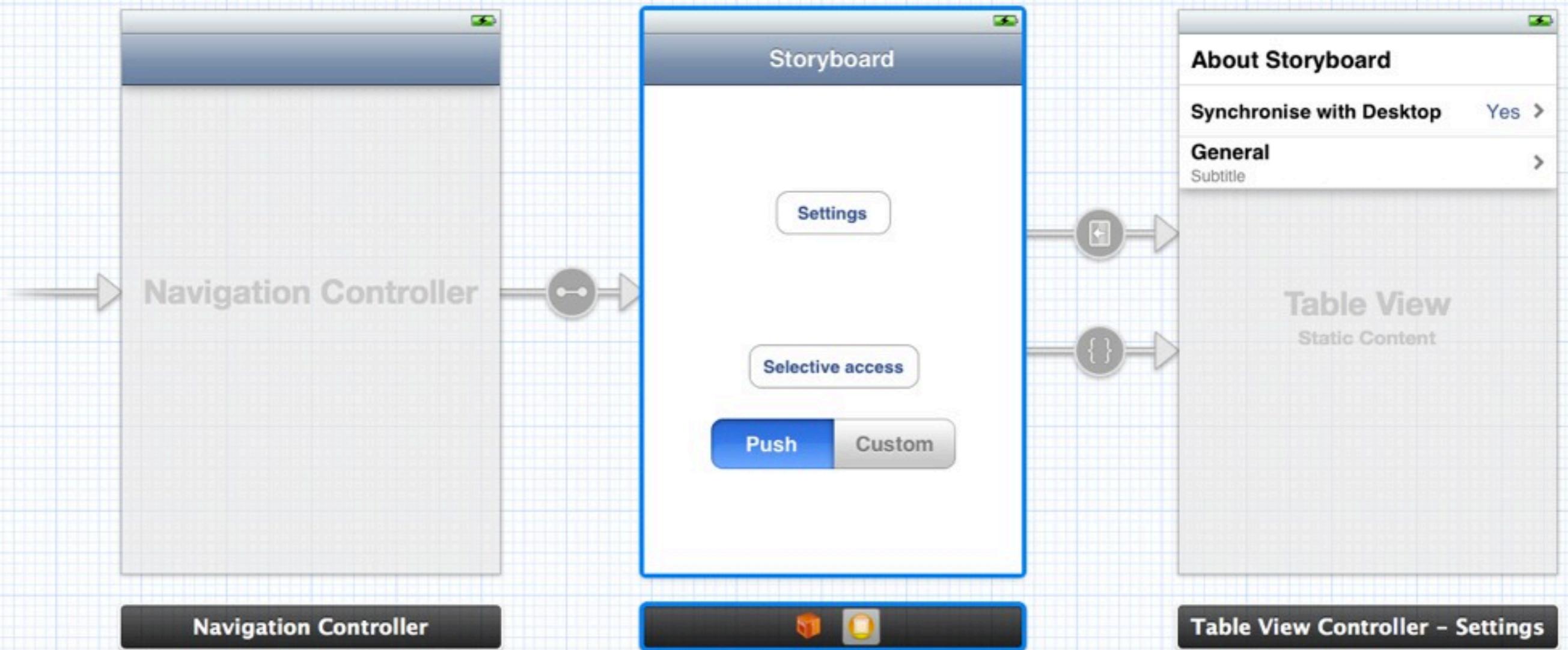


Life before storyboard

Separate screens; Independent XIB's



Storyboard - big picture



What is a storyboard?

What is a storyboard?

- new way to define app user interface

What is a storyboard?

- new way to define app user interface
- introduces:

What is a storyboard?

- new way to define app user interface
- introduces:
 - scene (often \equiv IB screen)

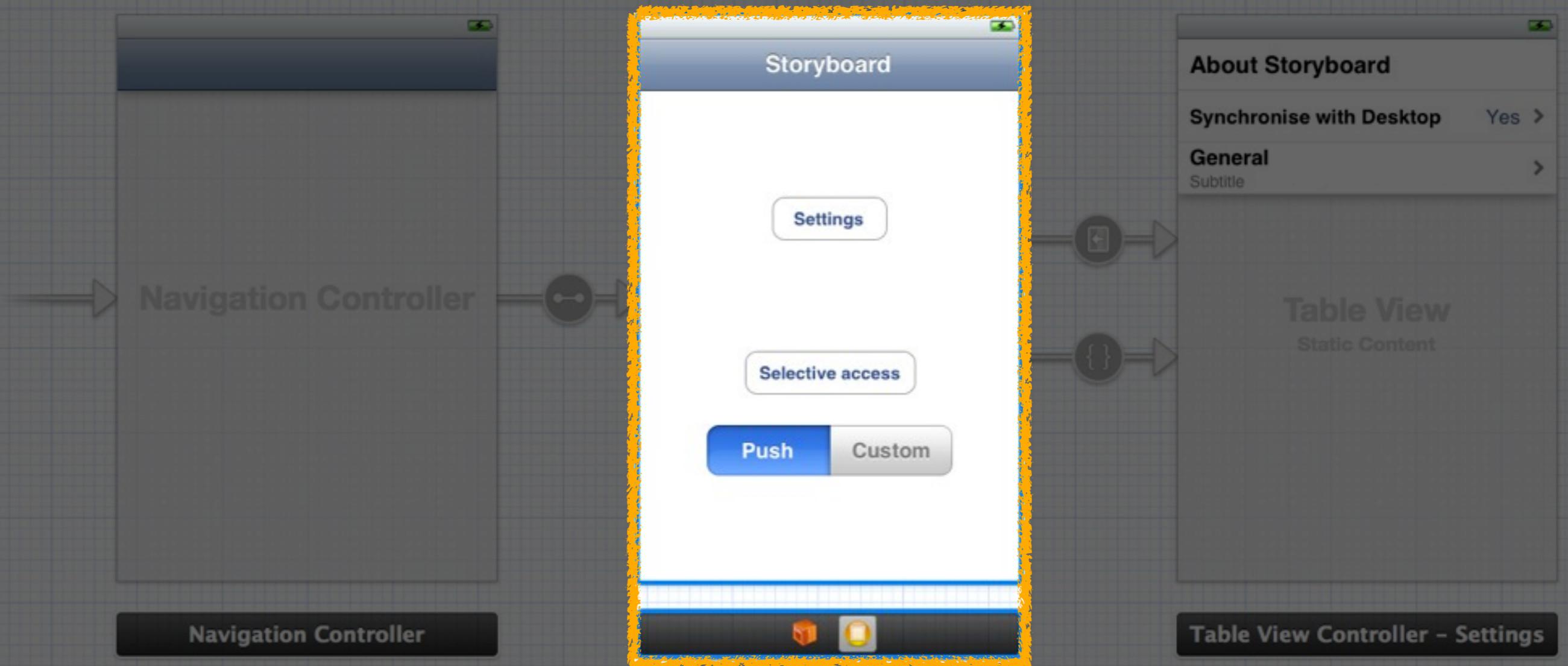
What is a storyboard?

- new way to define app user interface
- introduces:
 - scene (often \equiv IB screen)
 - dock (\equiv IB Document Outline)

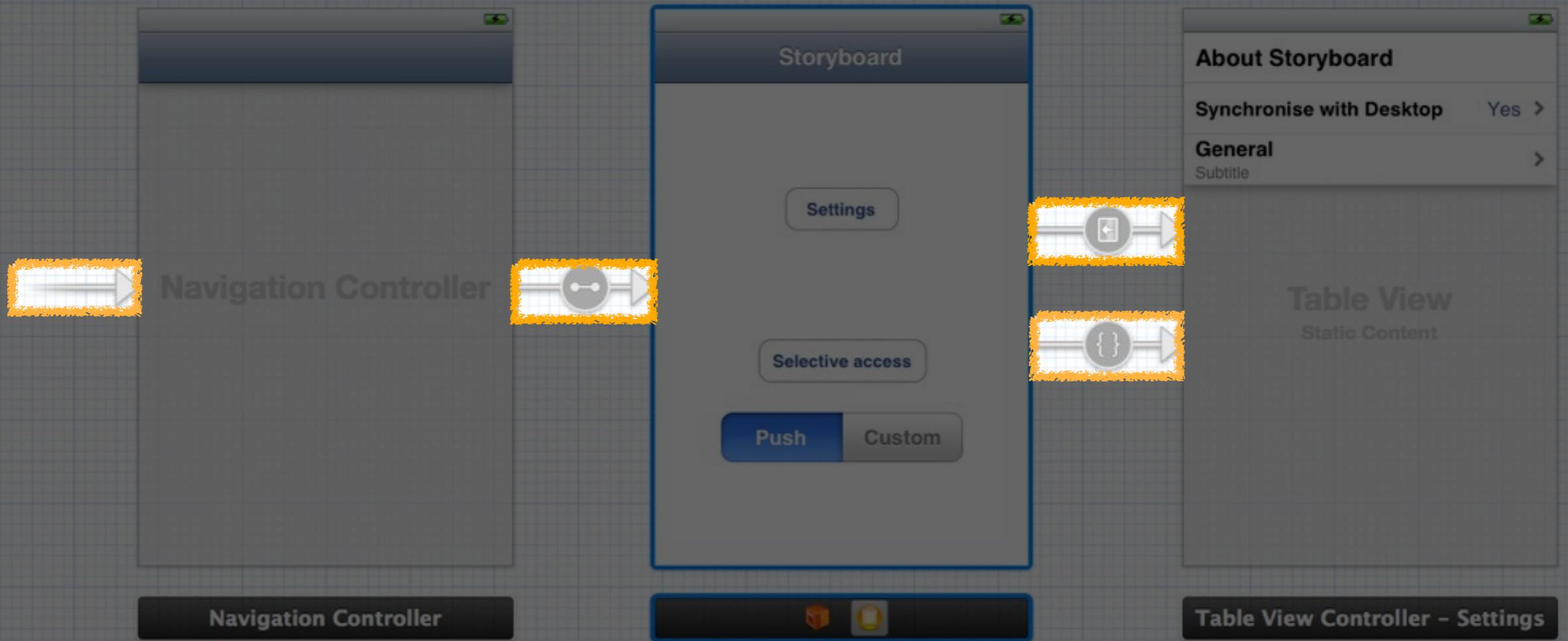
What is a storyboard?

- new way to define app user interface
- introduces:
 - scene (often \equiv IB screen)
 - dock (\equiv IB Document Outline)
 - segue (transition between scenes)

Scene and associated dock



Segues



How do we use it?

How do we use it?

- visual designer (very similar to IB)

How do we use it?

- visual designer (very similar to IB)
 - collection of XIB's

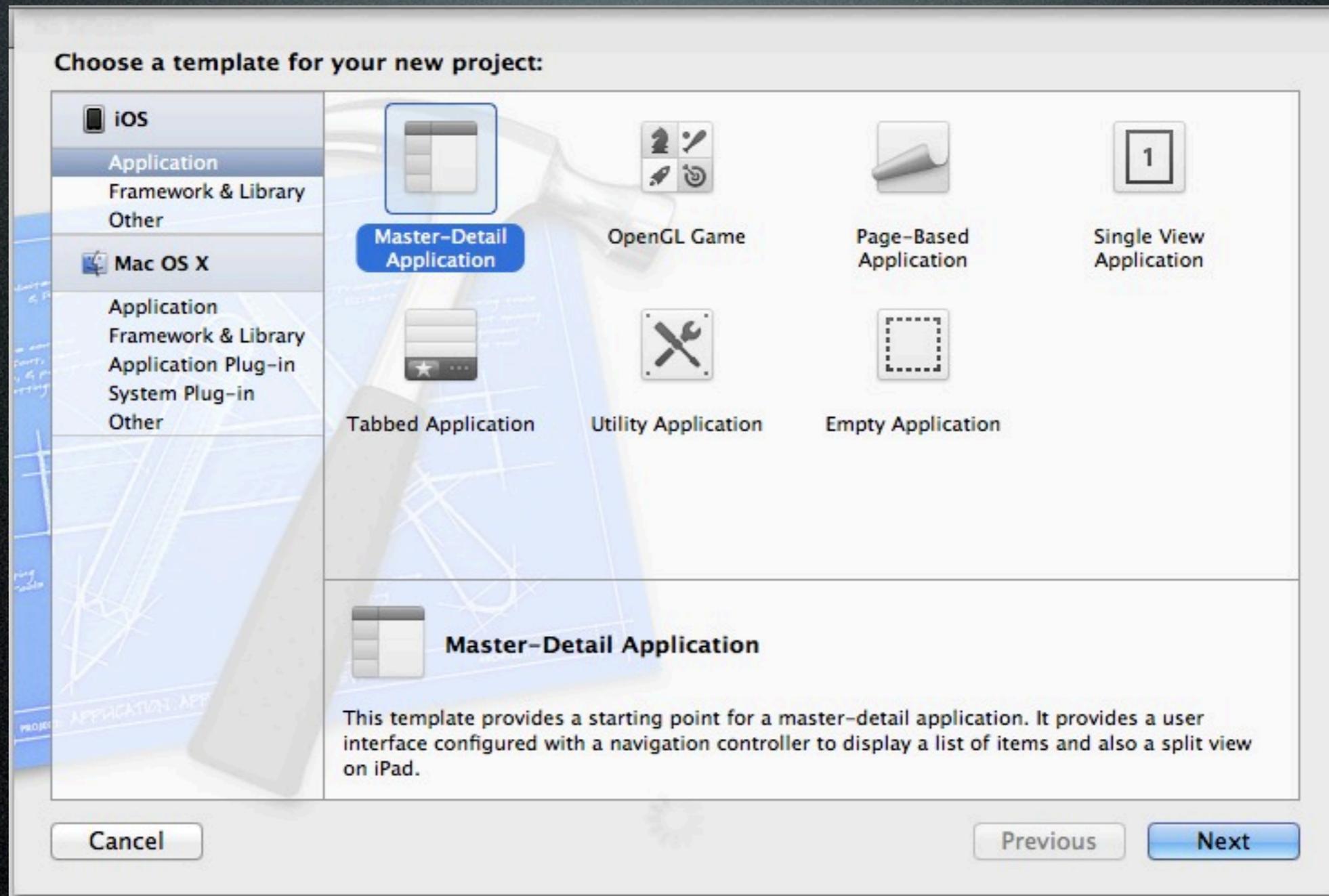
How do we use it?

- visual designer (very similar to IB)
 - collection of XIB's
 - an XML file really!

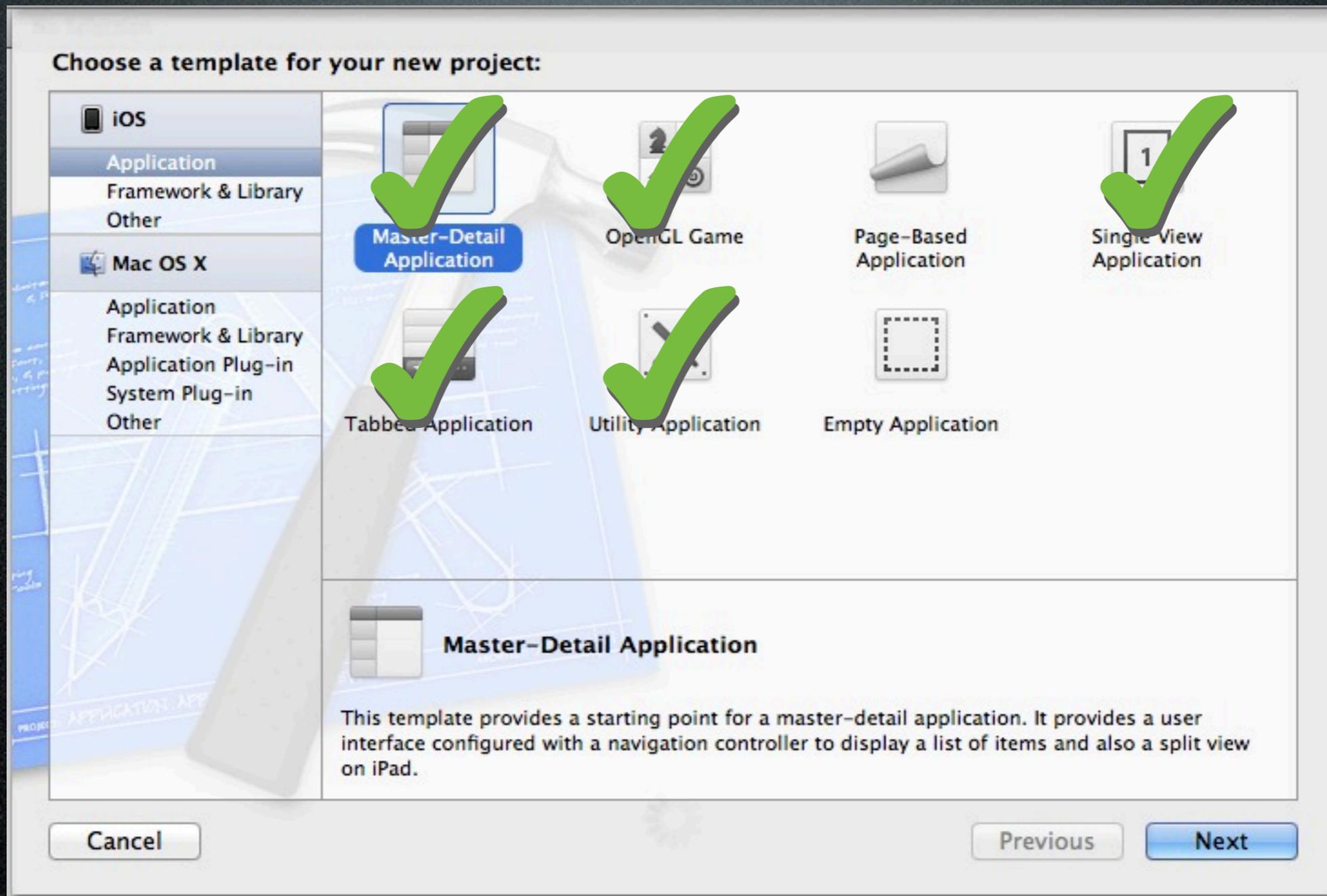
How do we use it?

- visual designer (very similar to IB)
 - collection of XIB's
 - an XML file really!
- and some 'behind-the-scenes' code

Storyboard templates



Storyboard templates



Storyboard template

Choose options for your new project:

Product Name

Company Identifier

Bundle Identifier

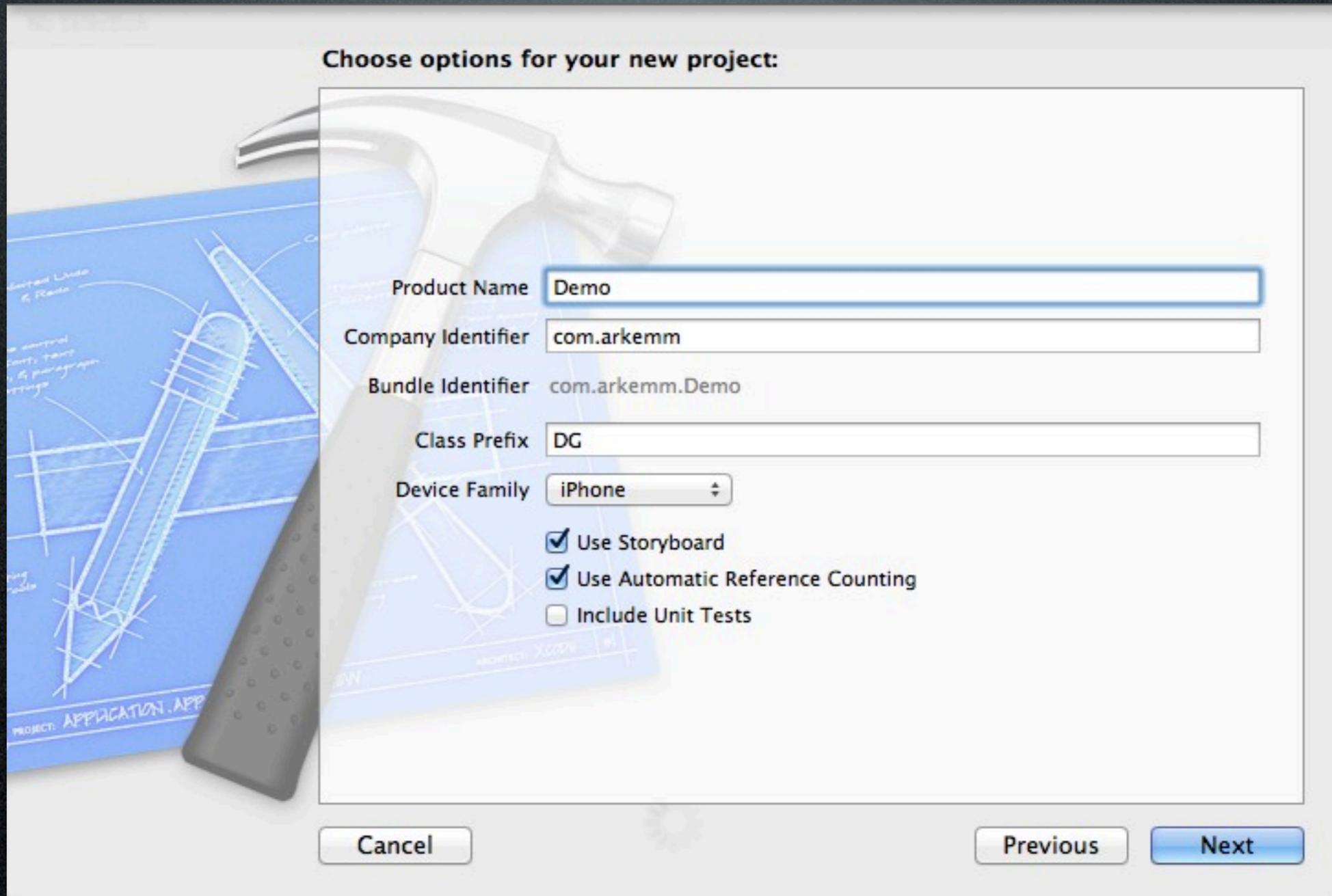
Class Prefix

Device Family

Use Storyboard

Use Automatic Reference Counting

Include Unit Tests

The image shows a dialog box for creating a new project in Xcode. The dialog is titled "Choose options for your new project:" and is overlaid on a background image of a hammer and blueprints. The form contains several input fields: "Product Name" (Demo), "Company Identifier" (com.arkemm), "Bundle Identifier" (com.arkemm.Demo), "Class Prefix" (DG), and "Device Family" (iPhone). There are also three checkboxes: "Use Storyboard" (checked), "Use Automatic Reference Counting" (checked), and "Include Unit Tests" (unchecked). At the bottom, there are three buttons: "Cancel", "Previous", and "Next".

Storyboard template

Choose options for your new project:

Product Name

Company Identifier

Bundle Identifier

Class Prefix

Device Family

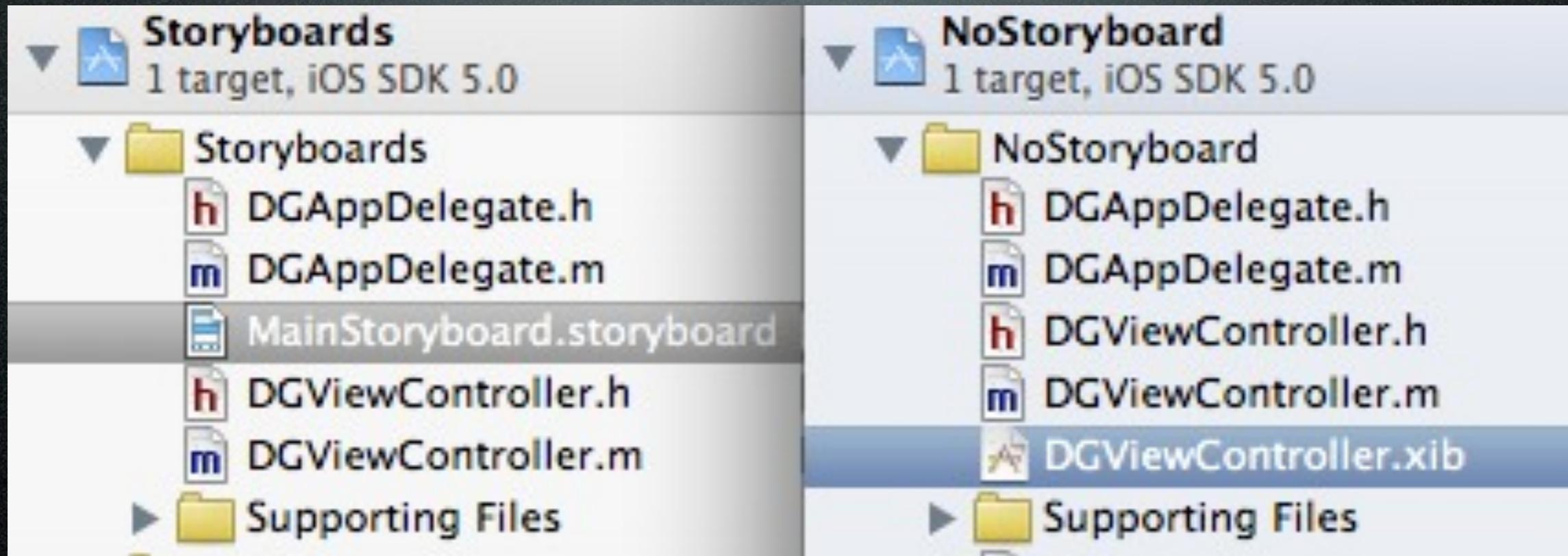
Use Storyboard

Use Automatic Reference Counting

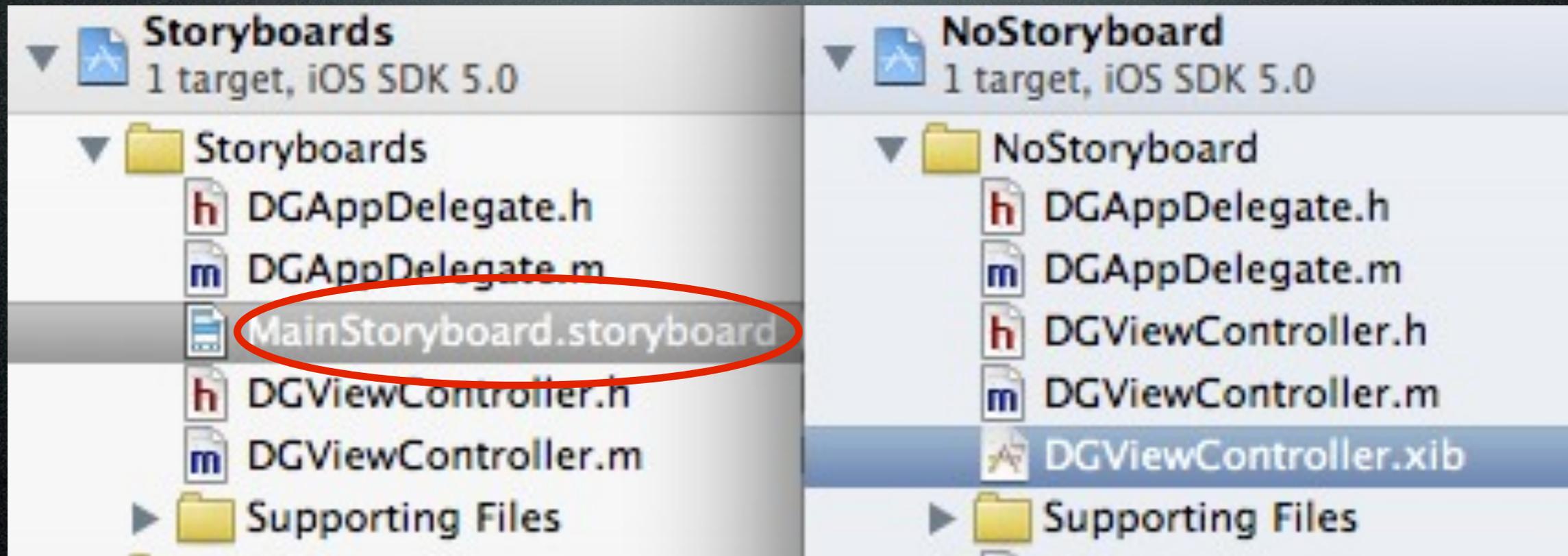
Include Unit Tests

A screenshot of the Xcode project creation dialog box. The dialog is titled "Choose options for your new project:" and contains several input fields and checkboxes. The "Use Storyboard" checkbox is checked and circled in red. The "Use Automatic Reference Counting" checkbox is also checked, while the "Include Unit Tests" checkbox is unchecked. The "Next" button is highlighted in blue, indicating it is the active option.

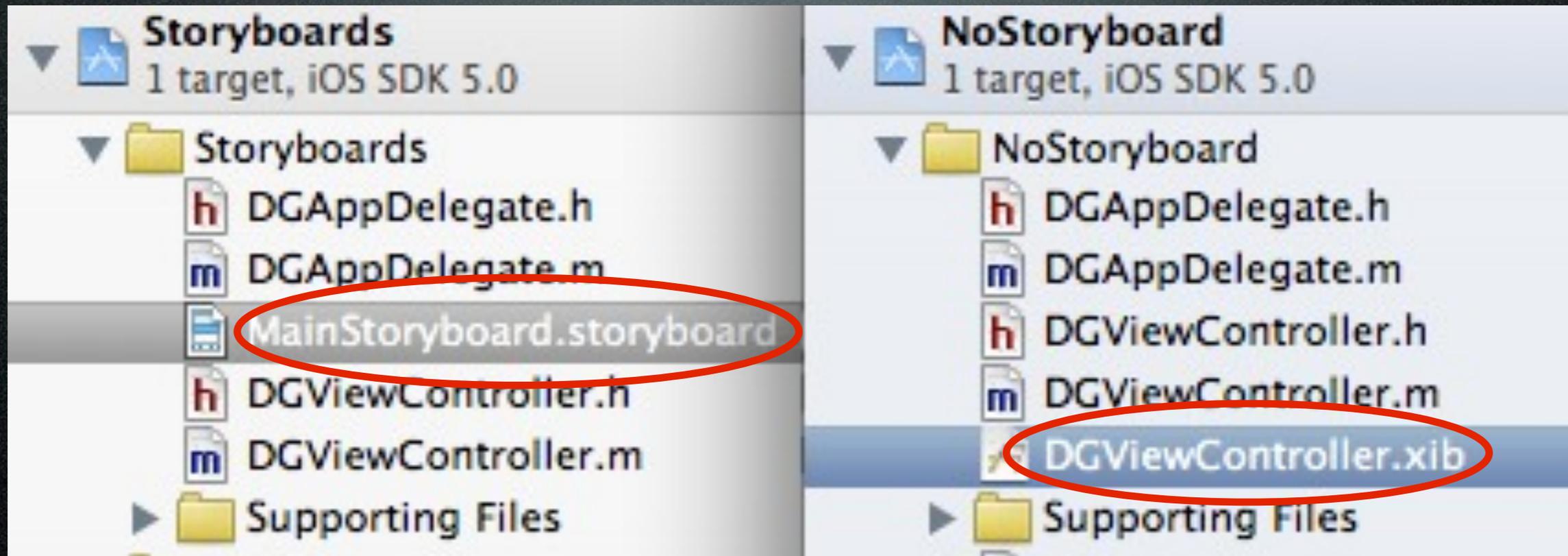
Storyboard files



Storyboard files



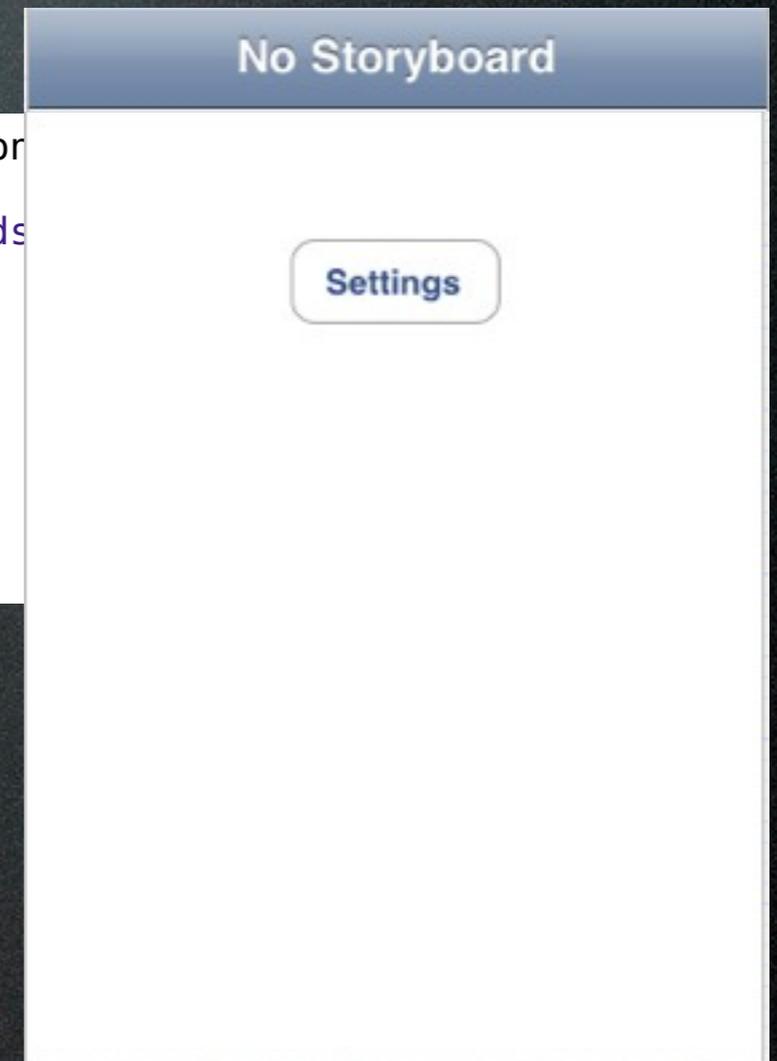
Storyboard files



Application Start-up

AppDelegate launch code...

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:  
{  
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]  
};
```



Application Start-up

AppDelegate launch code...

No Storyboard

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    DGViewController *controller = [[DGViewController alloc] initWithNibName:@"DGViewController" bundle:nil];
}
```

Application Start-up

AppDelegate launch code...

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    DGViewController *controller = [[DGViewController alloc] initWithNibName:@"DGViewController" bundle:nil];
    UINavigationController *aNavigationController = [[UINavigationController alloc]
                                                initWithRootViewController:controller];
}
}
```

Application Start-up

AppDelegate launch code...

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    DGViewController *controller = [[DGViewController alloc] initWithNibName:@"DGViewController" bundle:nil];
    UINavigationController *aNavigationController = [[UINavigationController alloc]
                                                initWithRootViewController:controller];
    self.rootController = aNavigationController;
}
```

Application Start-up

AppDelegate launch code...

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    DGViewController *controller = [[DGViewController alloc] initWithNibName:@"DGViewController" bundle:nil];
    UINavigationController *aNavigationController = [[UINavigationController alloc]
                                                initWithRootViewController:controller];

    self.rootController = aNavigationController;
    [self.window addSubview:[rootController view]];
}
```

Application Start-up

AppDelegate launch code...

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    DGViewController *controller = [[DGViewController alloc] initWithNibName:@"DGViewController" bundle:nil];
    UINavigationController *aNavigationController = [[UINavigationController alloc]
                                                initWithRootViewController:controller];

    self.rootController = aNavigationController;
    [self.window addSubview:[rootController view]];
    [self.window makeKeyAndVisible];
}
```

Application Start-up

AppDelegate launch code...

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    DGViewController *controller = [[DGViewController alloc] initWithNibName:@"DGViewController" bundle:nil];
    UINavigationController *aNavigationController = [[UINavigationController alloc]
                                                initWithRootViewController:controller];

    self.rootController = aNavigationController;
    [self.window addSubview:[rootController view]];
    [self.window makeKeyAndVisible];
    return YES;
}
```

Application Start-up

AppDelegate launch code...

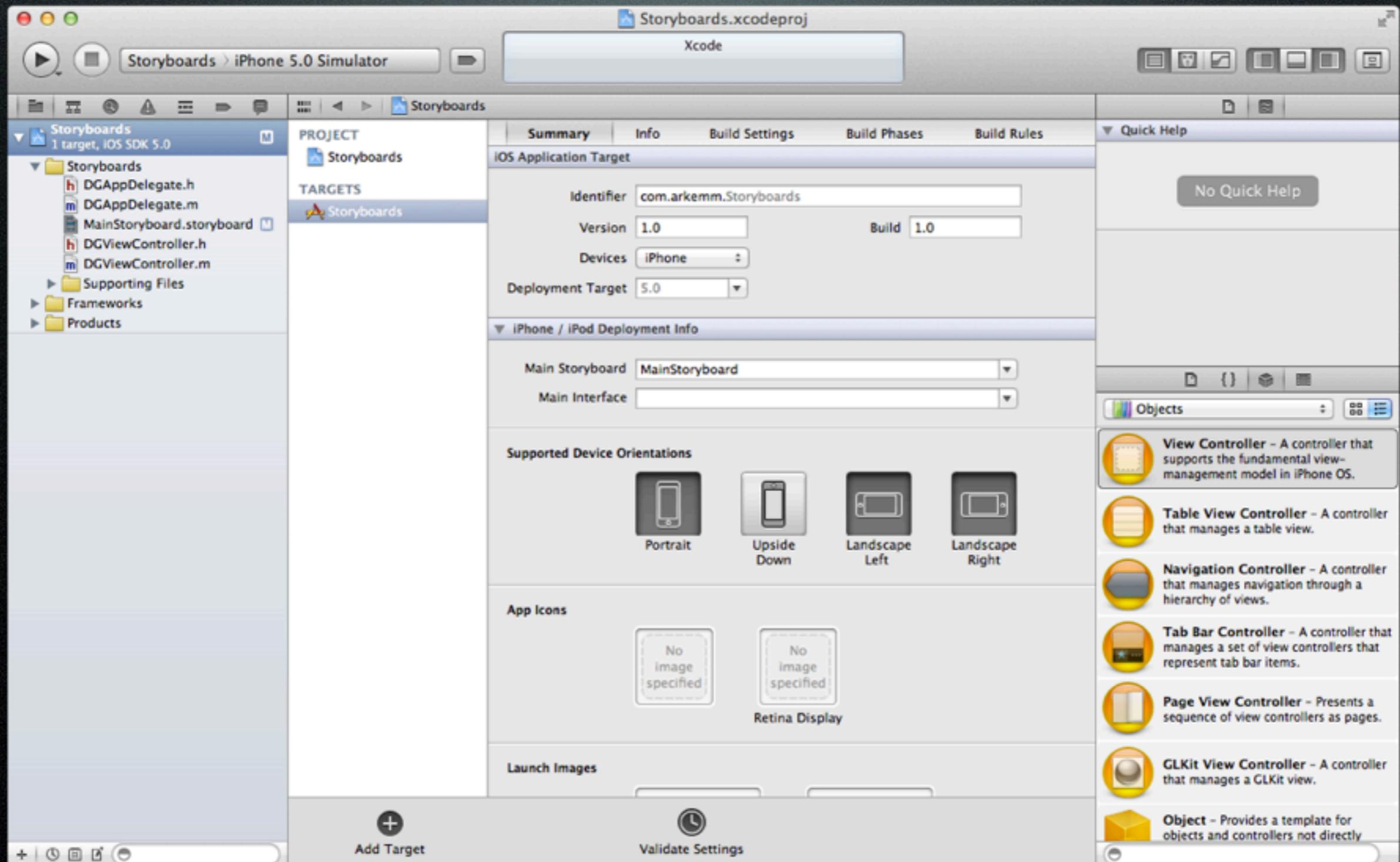
```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    DGViewController *controller = [[DGViewController alloc] initWithNibName:@"DGViewController" bundle:nil];
    UINavigationController *aNavigationController = [[UINavigationController alloc]
                                                    initWithRootViewController:controller];

    self.rootController = aNavigationController;
    [self.window addSubview:[rootController view]];
    [self.window makeKeyAndVisible];
    return YES;
}
```

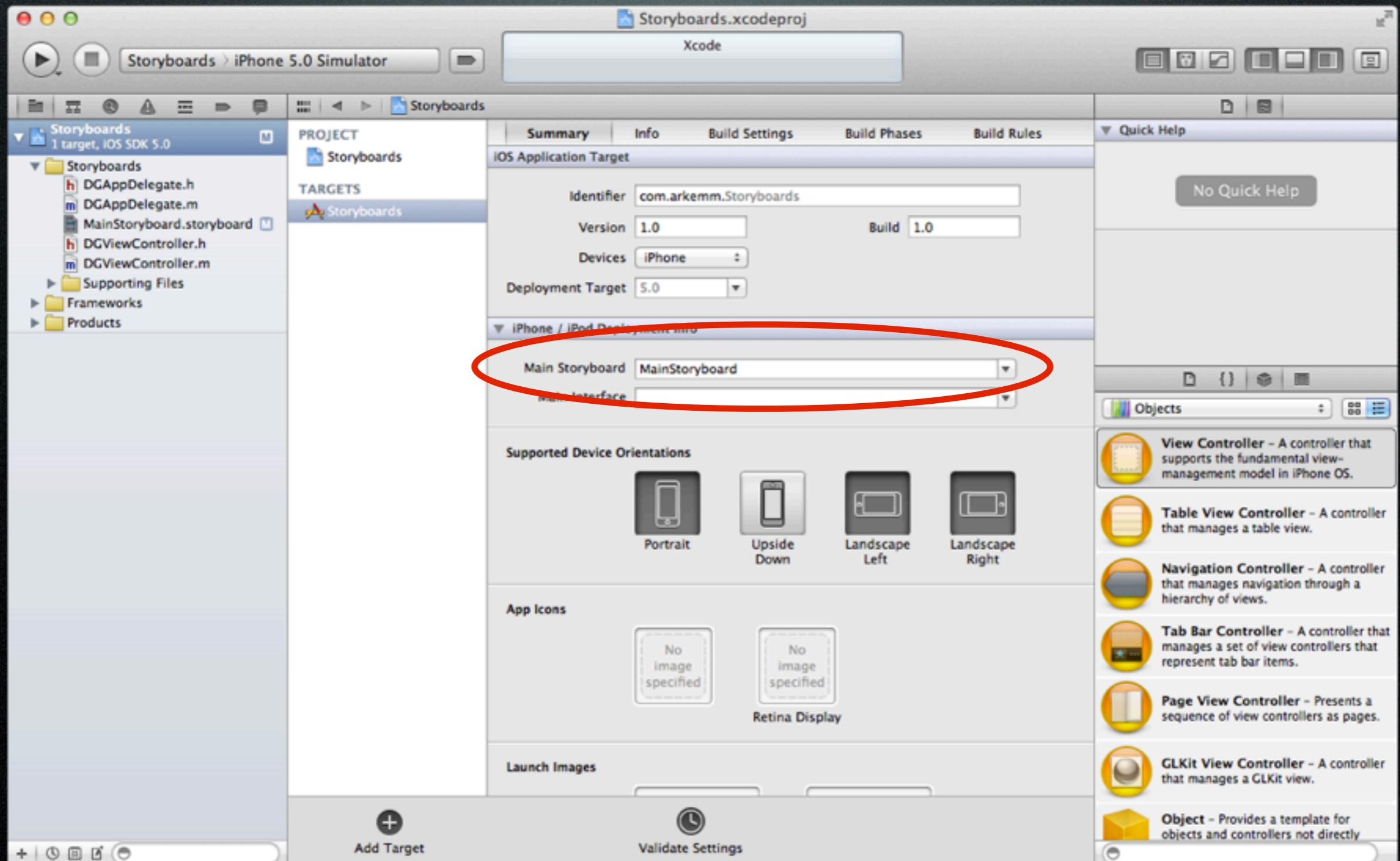
...and with storyboard

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    return YES;
}
```

Deployment info



Deployment info



plist entry

The screenshot shows the Xcode interface with the Storyboards-Info.plist file open. The 'Main storyboard file base name' entry is highlighted in blue. The table below shows the contents of the plist file.

Key	Type	Value
Localization native development region	String	en
Bundle display name	String	\$(PRODUCT_NAME)
Executable file	String	\$(EXECUTABLE_NAME)
Icon files	Array	(0 items)
Bundle identifier	String	com.arkemm.\$(PRODUCT_NAME:rfc1034identifier)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1.0
Application requires iPhone environment	Boolean	YES
Main storyboard file base name	String	MainStoryboard
Required device capabilities	Array	(1 item)
Supported interface orientations	Array	(3 items)

On the right side of the interface, the 'Quick Help' panel is visible, showing the declaration for the selected entry: `UIMainStoryboardFile`. Below it, the 'Objects' panel lists various UI components like View Controller, Table View Controller, Navigation Controller, Tab Bar Controller, Page View Controller, GLKit View Controller, and Object.

plist entry

The screenshot shows the Xcode interface with the Storyboards-Info.plist file open. The 'Main storyboard file base name' entry is highlighted in blue. A red callout box points to this entry with the text `<key>UIMainStoryboardFile</key>`.

Key	Type	Value
Localization native development region	String	en
Bundle display name	String	\$(PRODUCT_NAME)
Executable file	String	\$(EXECUTABLE_NAME)
Icon files	Array	(0 items)
Bundle identifier	String	com.arkemm.\$(PRODUCT_NAME:rfc1034identifier)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1.0
Application requires iPhone environment	Boolean	YES
Main storyboard file base name	String	MainStoryboard
Required device capabilities	Array	(1 item)
Supported interface orientations	Array	(3 items)

`<key>UIMainStoryboardFile</key>`

Let's see that in action

- Demo

Behind-the-scenes code

Behind-the-scenes code

- moving data between scenes

Behind-the-scenes code

- moving data between scenes
- initiating a transition

Behind-the-scenes code

- moving data between scenes
- initiating a transition
- interacting with the storyboard itself

Moving data

Passing data to destination controller

-[UIViewController prepareForSegue:sender]

- pointers to source & destination controllers
- string identifier

Returning data from destination controller

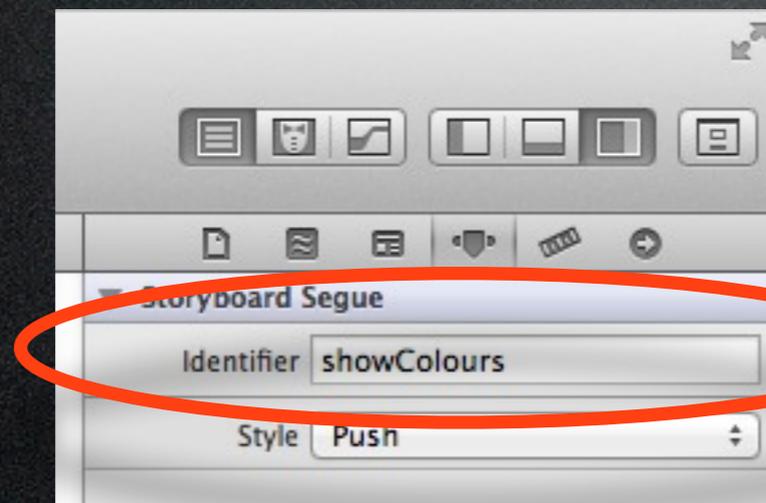
- via delegate

Passing data to controller

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([[segue identifier] isEqualToString:@"selectTags"]) {
        TagsController *controller = [segue destinationViewController];
        [controller setDelegate:self];
        [controller setSelectedTags:[self selectedTags]];
    }
    if ([[segue identifier] isEqualToString:@"showColours"]) {
        ColourController *controller = [segue destinationViewController];
        [controller setDelegate:self];
        [controller setSelectedColour:[self colourCell colour]];
    }
}
```

Passing data to controller

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([[segue identifier] isEqualToString:@"selectTags"]) {
        TagsController *controller = [segue destinationViewController];
        [controller setDelegate:self];
        [controller setSelectedTags:[self selectedTags]];
    }
    if ([[segue identifier] isEqualToString:@"showColours"]) {
        ColourController *controller = [segue destinationViewController];
        [controller setDelegate:self];
        [controller setSelectedColour:[self colourCell colour]];
    }
}
```



Returning results

Returning results

Source conforms to a protocol

```
@protocol ShowDetailsDelegate <NSObject>  
- (void)controller:(UIViewController *)controller didSelect:(NSObject *)sel;  
@end
```

Returning results

Source conforms to a protocol

```
@protocol ShowDetailsDelegate <NSObject>  
- (void)controller:(UIViewController *)controller didSelect:(NSObject *)sel;  
@end
```

Destination controller holds a delegate pointer

```
@property (weak, nonatomic) id <ShowDetailsDelegate> delegate;
```

Returning results

Source conforms to a protocol

```
@protocol ShowDetailsDelegate <NSObject>
- (void)controller:(UIViewController *)controller didSelect:(NSObject *)sel;
@end
```

Destination controller holds a delegate pointer

```
@property (weak, nonatomic) id <ShowDetailsDelegate> delegate;
```

Destination calls protocol method on completion

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)ip {
    NSObject *client = [[self fetchedResultsController] objectAtIndex:indexPath:ip];
    [[self delegate] controller:self didSelect:client];
}
```

Returning results

Source conforms to a protocol

```
@protocol ShowDetailsDelegate <NSObject>
- (void)controller:(UIViewController *)controller didSelect:(NSObject *)sel;
@end
```

Destination controller holds a delegate pointer

```
@property (weak, nonatomic) id <ShowDetailsDelegate> delegate;
```

Destination calls protocol method on completion

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)ip {
    NSManagedObject *client = [[self fetchedResultsController] objectAtIndex:indexPath:ip];
    [[self delegate] controller:self didSelect:client];
}
```

Source implements protocol method

```
-(void)controller:(UIViewController *)controller didSelect:(NSObject *)sel {
    [self setSelectedClient:(NSManagedObject *)sel];
    [[self clientNameLabel] setText:[[self selectedClient] title]];
    [[self navigationController] pushViewControllerAnimated:YES];
}
```

Initiating a transition

Initiating a segue in code

```
-[UIViewController performSegue:sender:]
```

```
[self performSegueWithIdentifier:@"Settings" sender:self];
```

Interacting with storyboard

Explicitly loading a storyboard

- `+[UIStoryboard storyboardWithName:bundle:]`

Starting scene of a storyboard

- `-[UIStoryboard instantiateInitialViewController]`

Specifying view controller by identifier

- `-[UIStoryboard instantiateViewControllerWithIdentifier]`

What does it do for us?

What does it do for us?

- overview of entire application

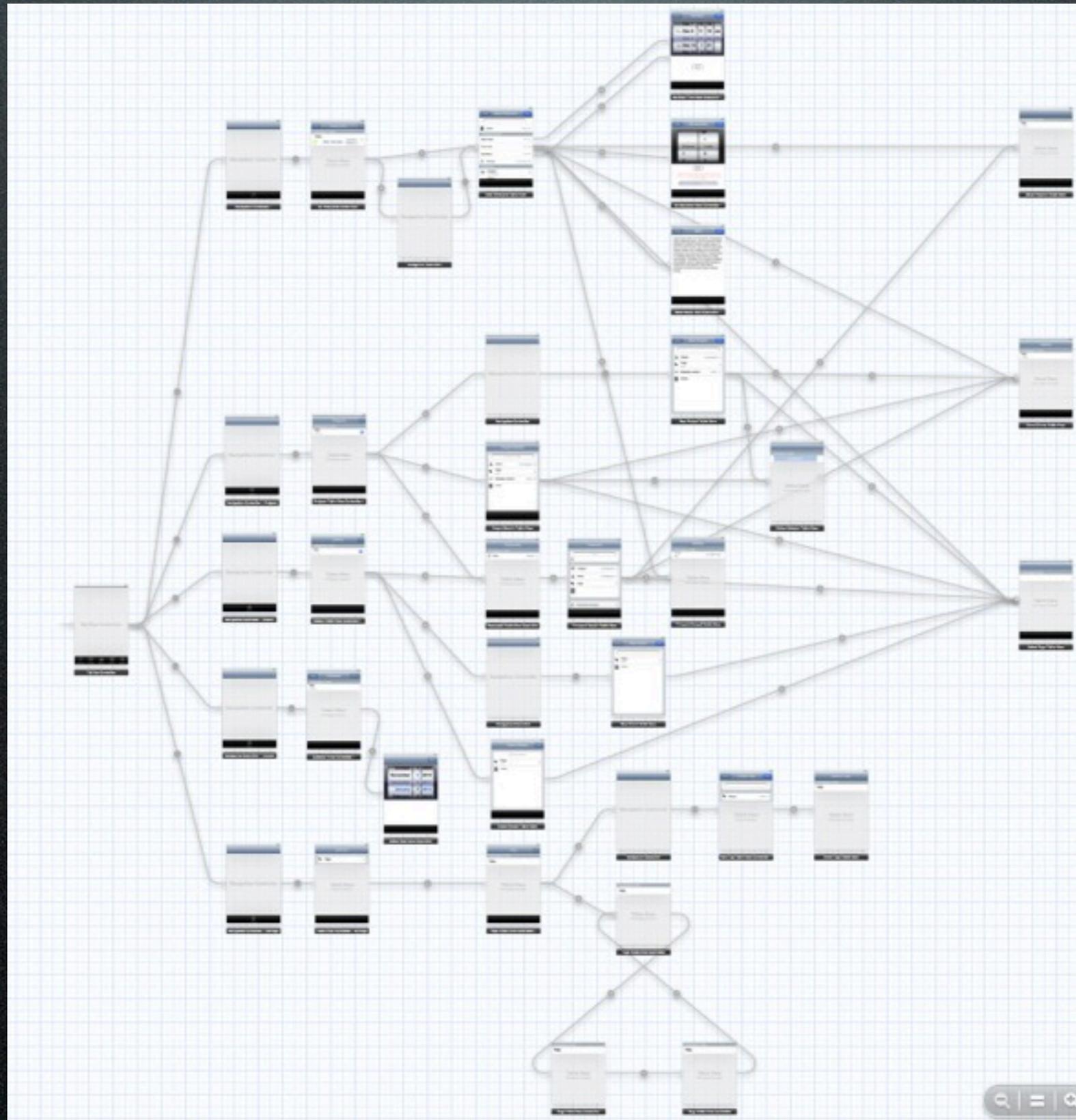
What does it do for us?

- overview of entire application
- visually represent table data

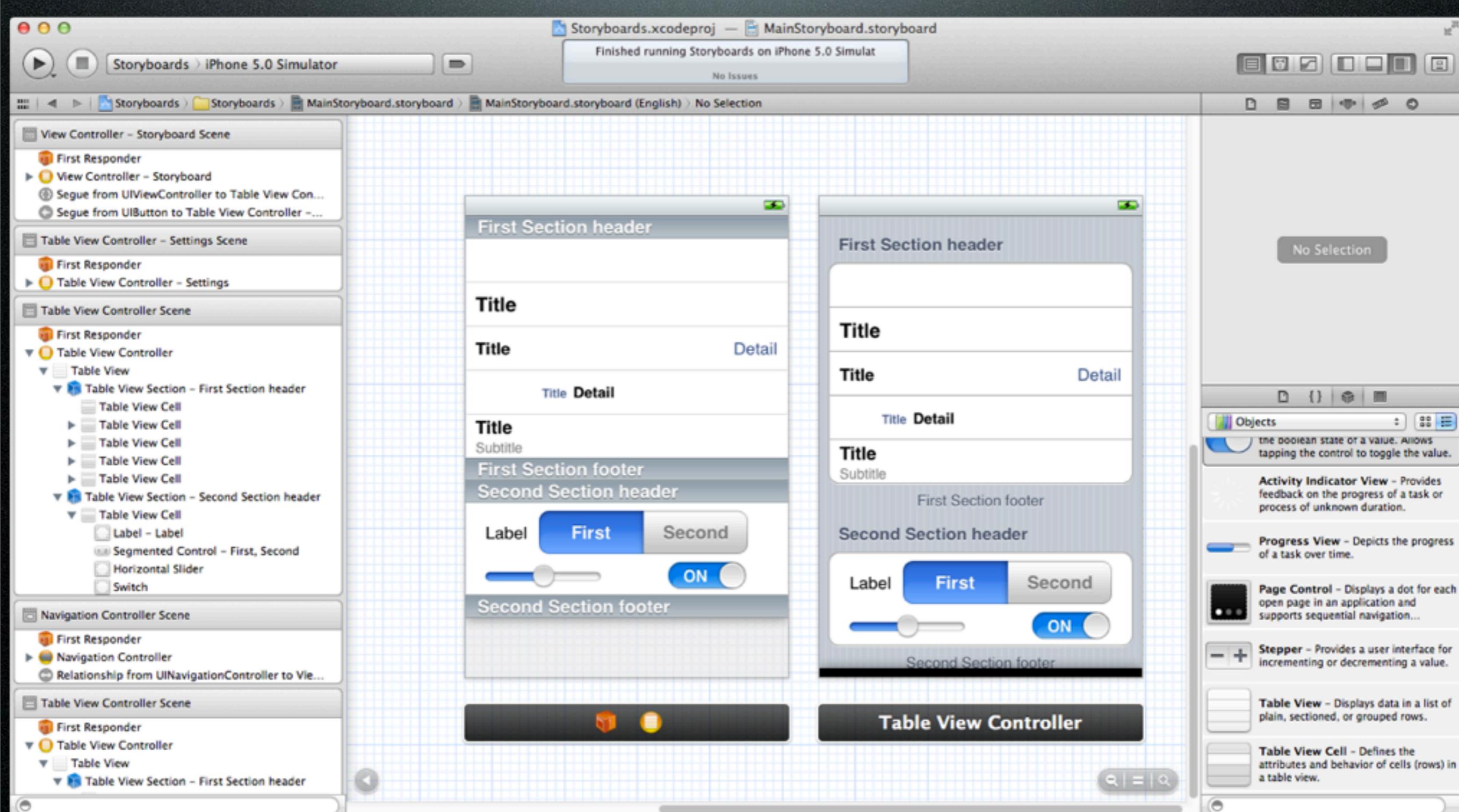
What does it do for us?

- overview of entire application
- visually represent table data
- reduce or even eliminate code

Overview



Static table view



Static table view

Custom

Basic

Right Detail

Left Detail

Subtitle

Custom cell with contents



Code reduction



Code reduction

```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    NSString *CellIdentifier;
    if ([indexPath row] == 0) {
        CellIdentifier = @"Cell";
    } else {
        if ([indexPath row] == 1) {
            CellIdentifier = @"DefaultCell";
        } else {
            CellIdentifier = @"SubtitleCell";
        }
    }
    UISwitch *bluetooth;
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        switch ([indexPath row]) {
            case 1:
                cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleValue1 reuseIdentifier:CellIdentifier];
                [cell setAccessoryType:UITableViewCellAccessoryDisclosureIndicator];
                break;
            case 2:
                cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleValue1 reuseIdentifier:CellIdentifier];
                bluetooth = [[UISwitch alloc] initWithFrame:CGRectMake(200.0f, 8.0f, 79.0f, 27.0f)];
                [bluetooth setOn:YES];
                [[cell contentView] addSubview:bluetooth];
                break;
            default:
                cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];
                [cell setAccessoryType:UITableViewCellAccessoryDisclosureIndicator];
                break;
        }
    }
    // Configure the cell...
    switch ([indexPath row]) {
        case 1:
            [[cell.textLabel] setText:@"General"];
            break;
        case 2:
            [[cell.textLabel] setText:@"Bluetooth"];
            break;
        default:
            [[cell.textLabel] setText:@"About"];
            [[cell.detailTextLabel] setText:@"DG Demo"];
            break;
    }
    return cell;
}
```

Code reduction

```
-(UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    NSString *CellIdentifier;
    if ([indexPath row] == 0) {
        CellIdentifier = @"Cell";
    } else {
        if ([indexPath row] == 1) {
            CellIdentifier = @"DefaultCell";
        } else {
            CellIdentifier = @"SubtitleCell";
        }
    }
    UISwitch *bluetooth;
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        switch ([indexPath row]) {
            case 1:
                cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];
                [cell setAccessoryType:UITableViewCellAccessoryDisclosureIndicator];
                break;
            case 2:
                cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleValue1 reuseIdentifier:CellIdentifier];
                bluetooth = [[UISwitch alloc] initWithFrame:CGRectMake(100.0f, 8.0f, 79.0f, 27.0f)];
                [bluetooth setOn:YES];
                [[cell contentView] addSubview:bluetooth];
                break;
            default:
                cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];
                [cell setAccessoryType:UITableViewCellAccessoryDisclosureIndicator];
                break;
        }
    }
    // Configure the cell...
    switch ([indexPath row]) {
        case 1:
            [[cell.textLabel] setText:@"General"];
            break;
        case 2:
            [[cell.textLabel] setText:@"Bluetooth"];
            break;
        default:
            [[cell.textLabel] setText:@"About"];
            [[cell.detailTextLabel] setText:@"DG Demo"];
            break;
    }
    return cell;
}
```

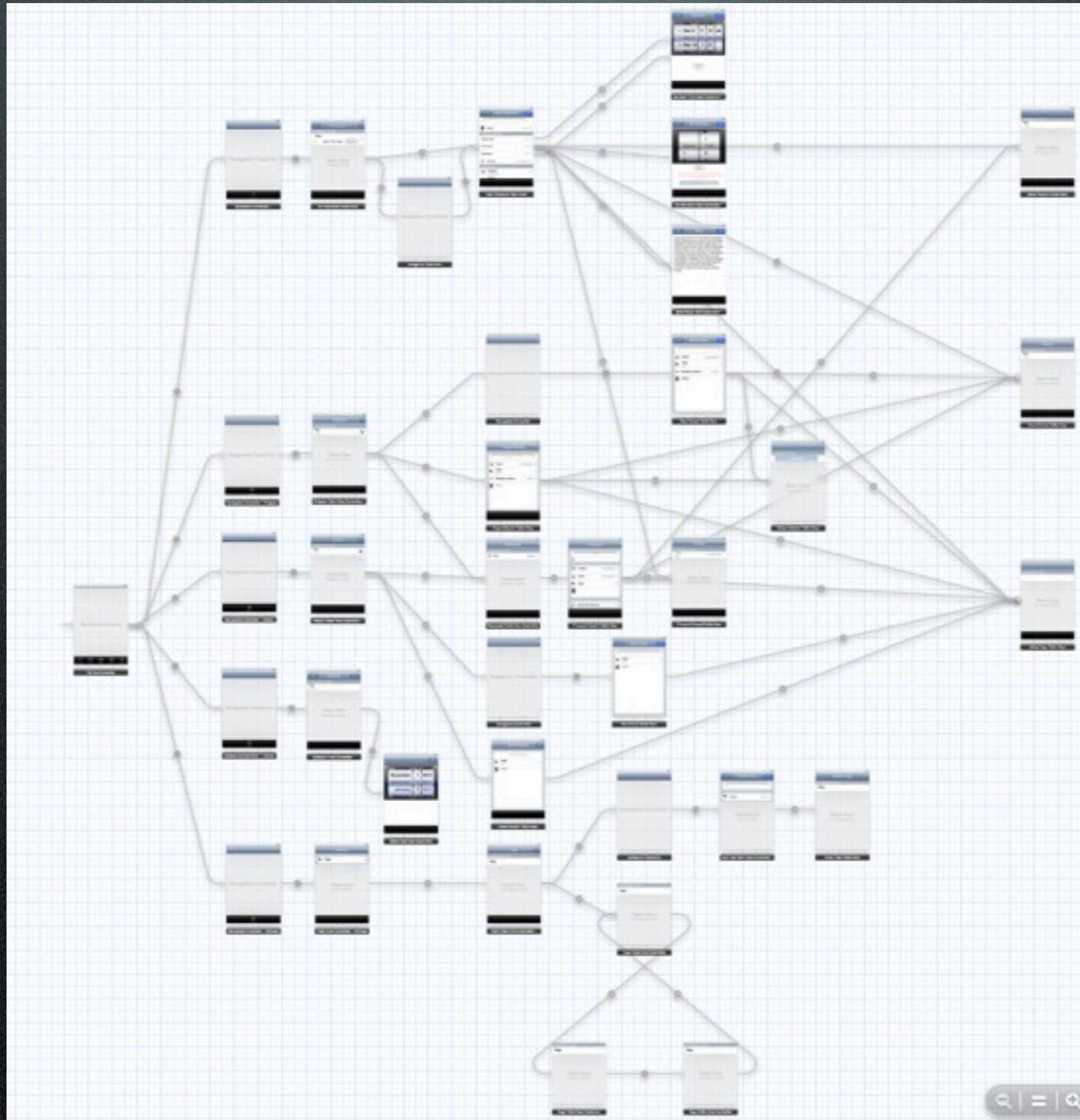
Things to be aware of

- attaching segues
- remember to disconnect connections before deleting objects
- initWithCoder / initWithNibName

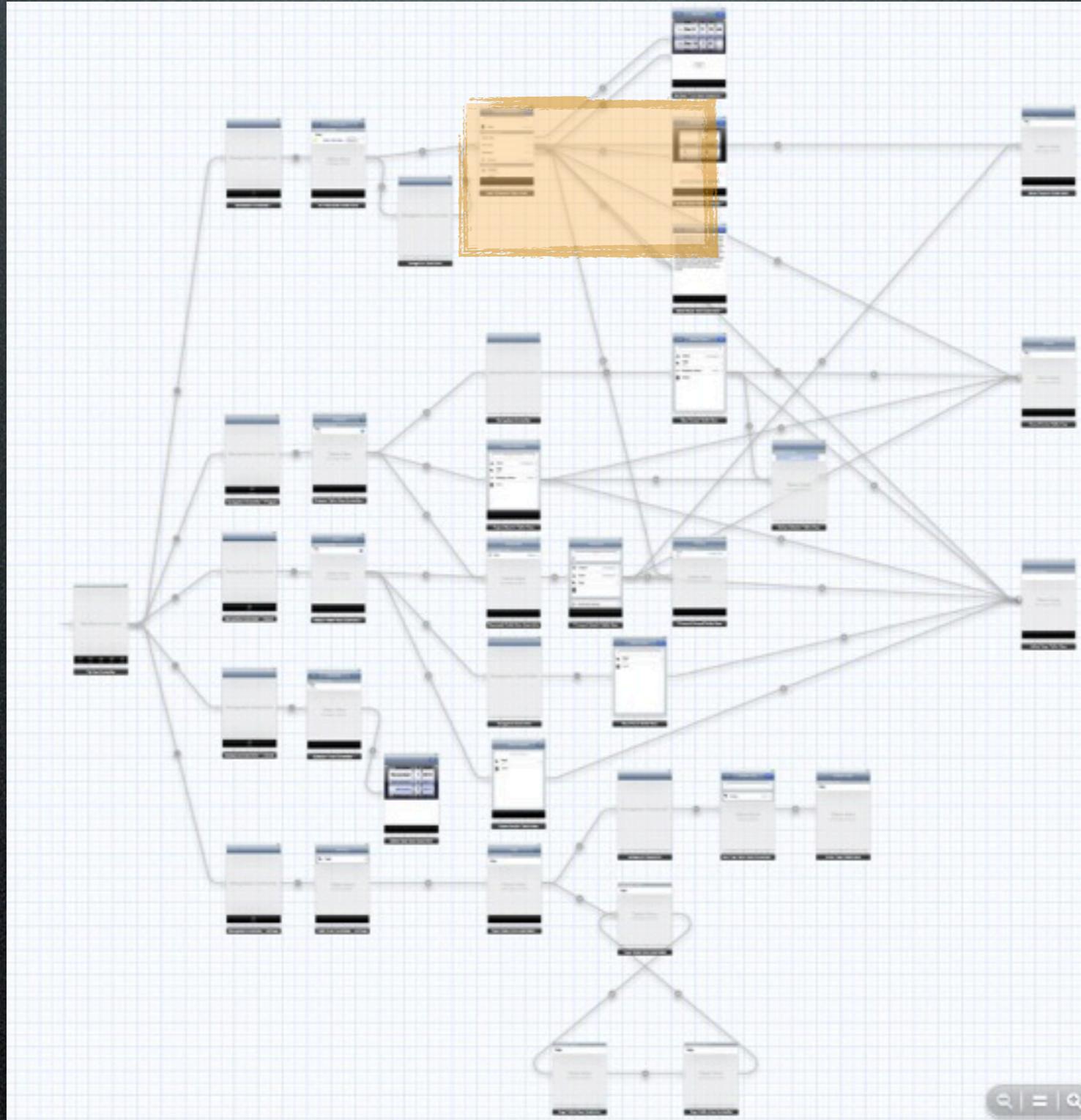
Things to be aware of

- cannot easily reuse table view cell designs
- cannot use custom parent view controllers

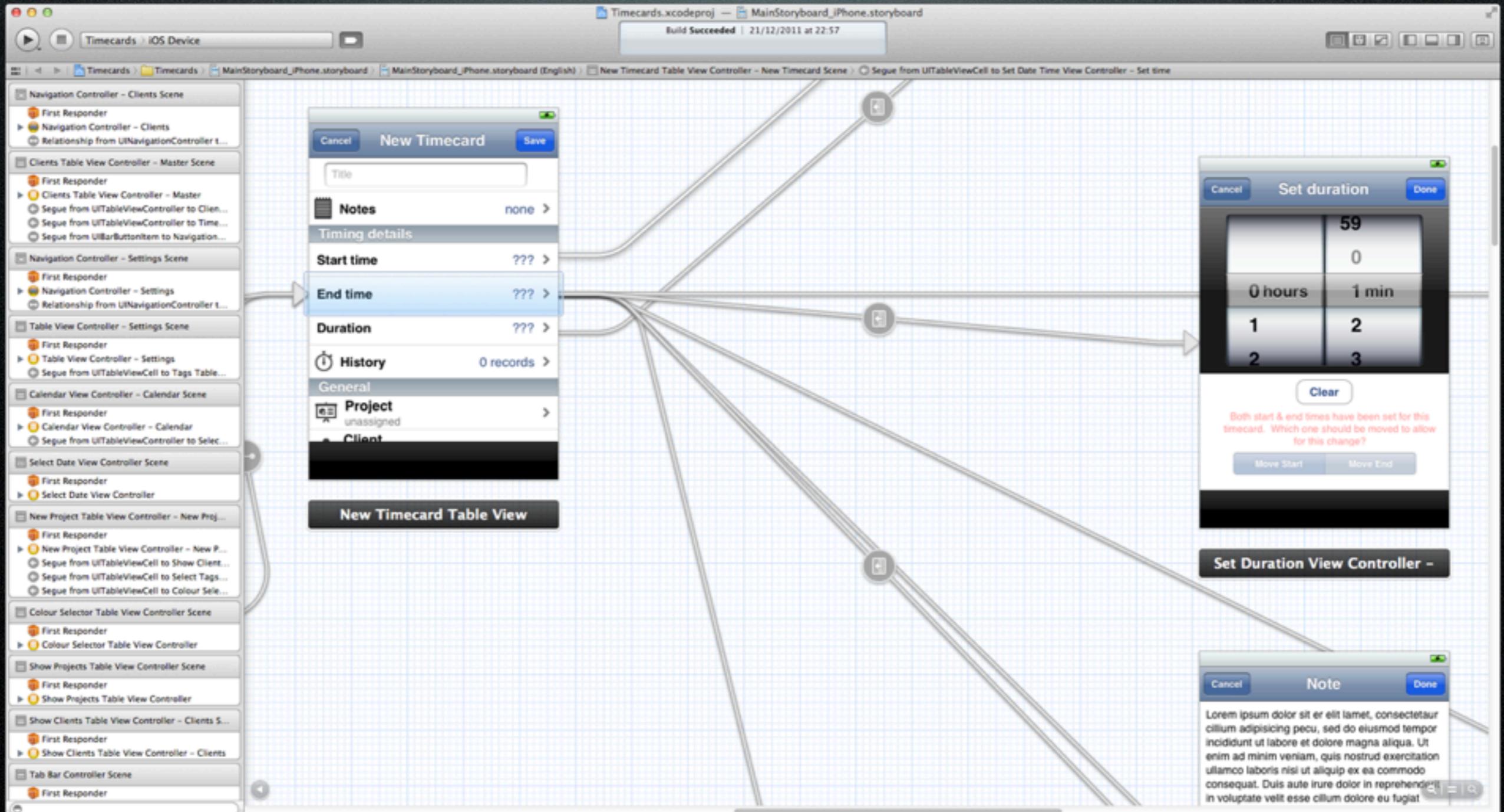
Not all easy though!



Not all easy though!



Edit mode



Thank you

- Questions?