

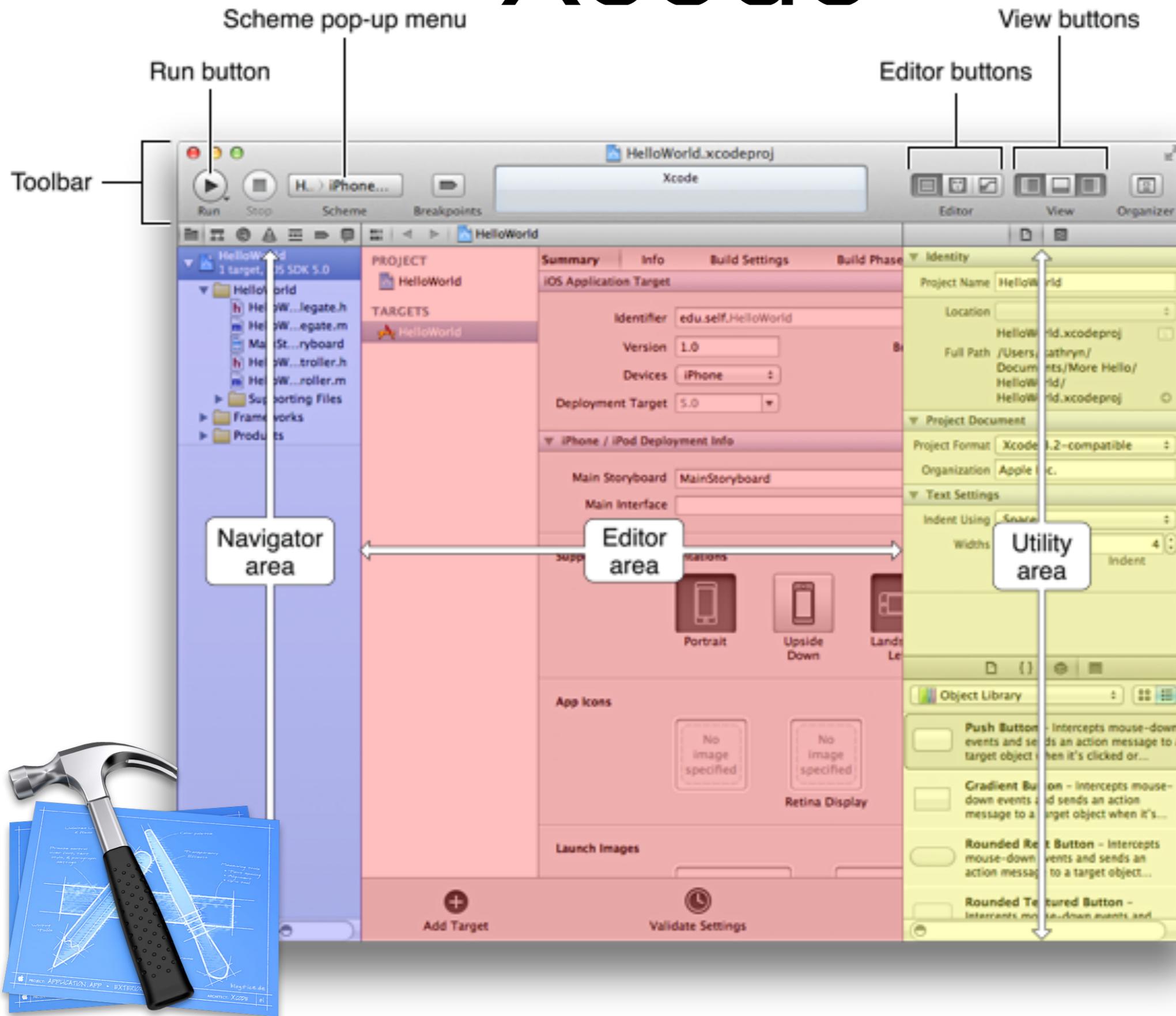
# iOS5 Workshop

Xcode, and stuff.

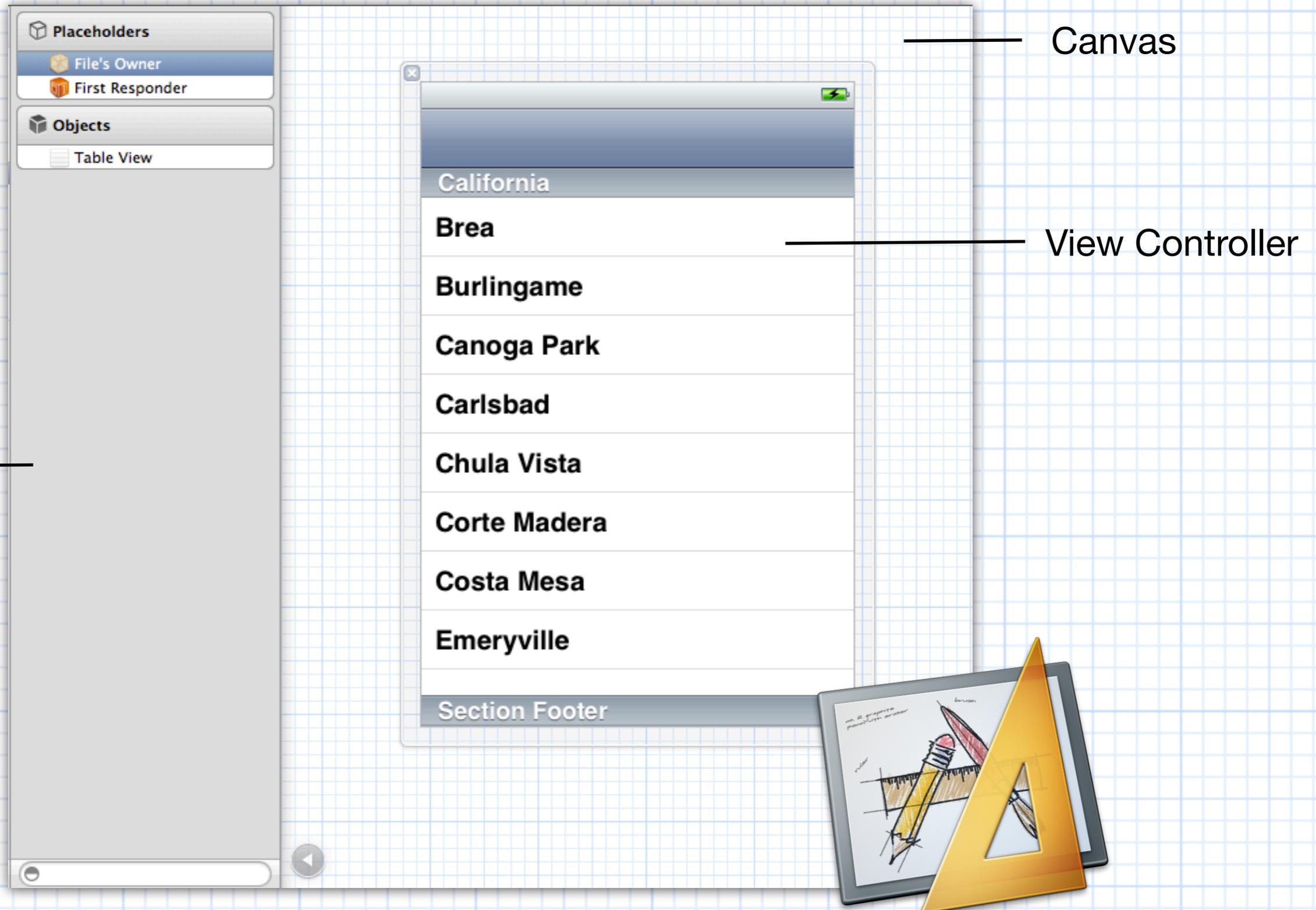
# What You'll See Today

- ▶ The iOS SDK & ToolKit
- ▶ Your 1st Xcode project
- ▶ The application lifecycle
- ▶ Table views & navigation controllers galore
- ▶ Interface Builder's *Assistant*
- ▶ Web views
- ▶ Secrets??

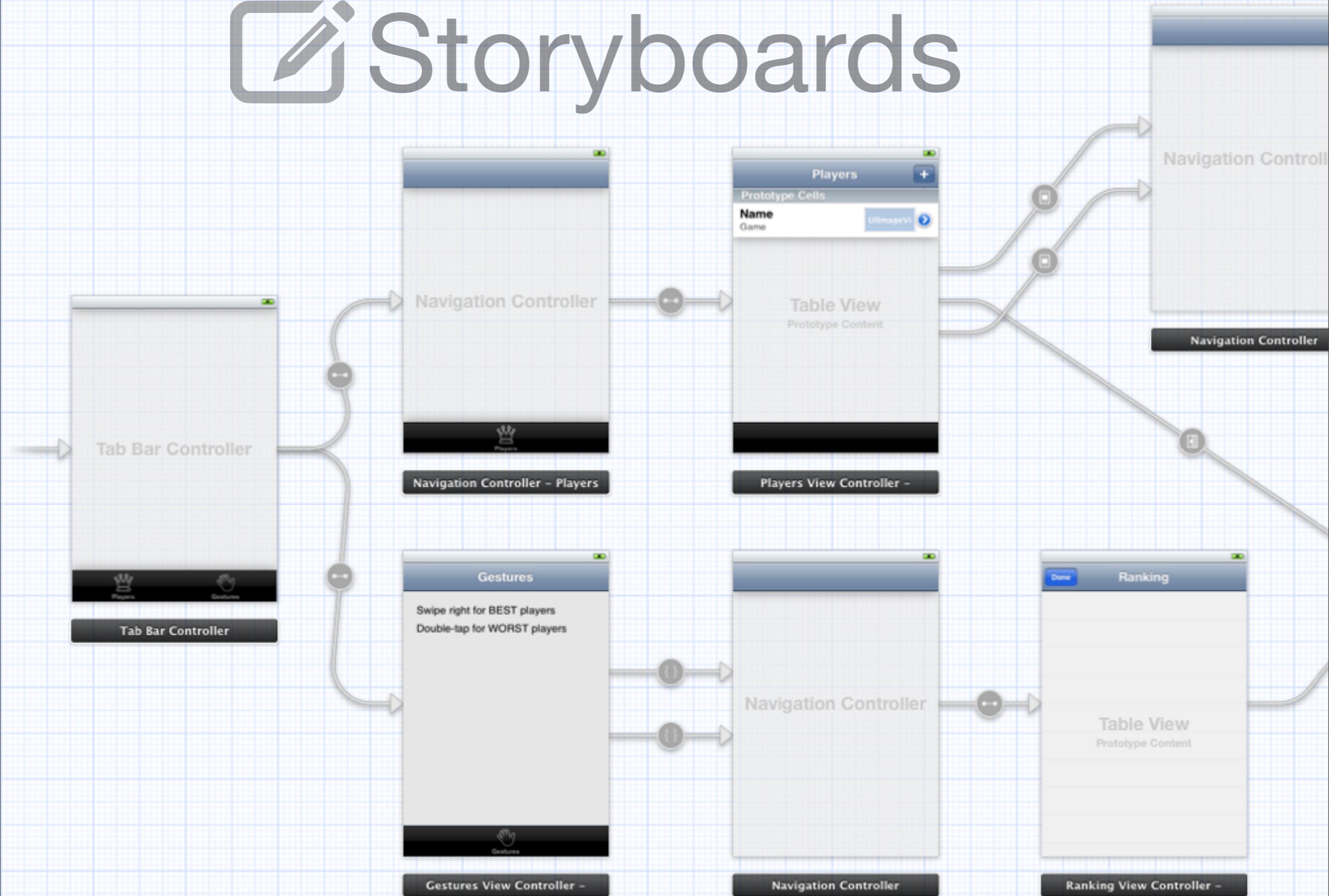
# Xcode



# Interface Builder



# Storyboards





# Getting Your Hands Dirty



Checkout *ios-workshop* from github:  
[github.com/kyleturner/ios-workshop](https://github.com/kyleturner/ios-workshop)



Open *iOS Workshop/iOS Workshop.xcodeproj*



Let's go!

# What's Inside

- ▶ **App Delegate**
- ▶ **Domain:** *WSLesson*
- ▶ **ViewControllers:**
  - ▶ *WSMasterViewController*: a list of available workshop lessons.
  - ▶ *WSDetailViewController*: details about a particular lesson.
  - ▶ *WSSecretsViewController*: ??????

# Objective-C

- ▶ Dynamic, OO Language
- ▶ Behavior determined @ runtime vs. compile time
- ▶ Classes defined as interface & implementation files
- ▶ Header (.h) files define public declarations
- ▶ Implementation (.m) files define the details

# App Delegate

.h

```
@interface WAppDelegate : UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) UIWindow *window;
@property (strong, nonatomic) UINavigationController *navigationController;
|
@end
```

---

.m

```
#import "WAppDelegate.h"
#import "WSMasterViewController.h"

@implementation WAppDelegate

@synthesize window = _window;
@synthesize navigationController = _navigationController;

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

    // YOUR override point for customization after application launch.  You'll want to initialize a view controller, set him as the window
    // 's root view controller, and call makeKeyAndVisible to display that view controller's view on screen.

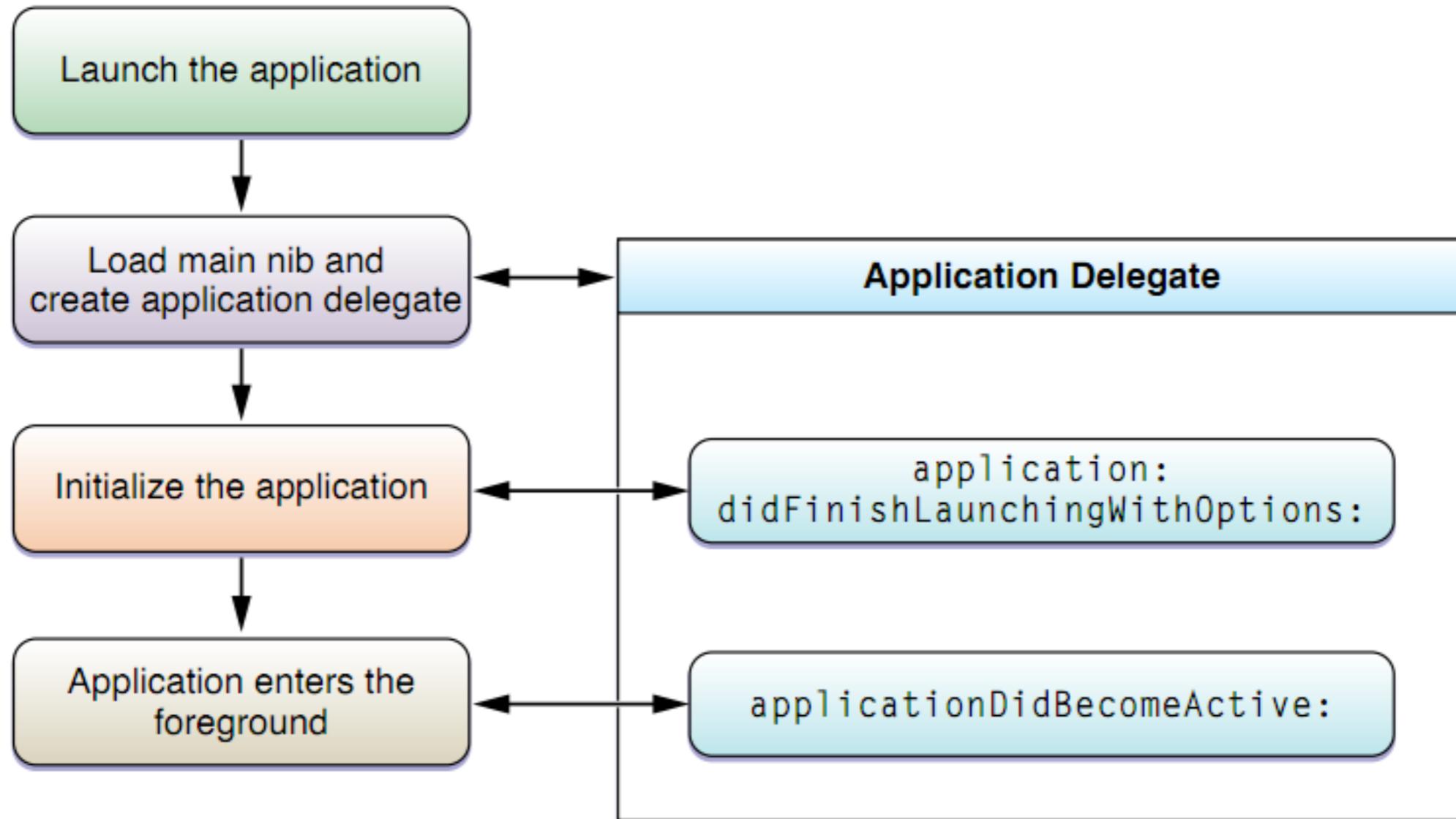
    WSMasterViewController *masterViewController = [[WSMasterViewController alloc] initWithNibName:@"WSMasterViewController" bundle:nil];
    self.navigationController = [[UINavigationController alloc] initWithRootViewController:masterViewController];

    self.window.rootViewController = self.navigationController;
    [self.window makeKeyAndVisible];

    return YES;
}
```

# Application Lifecycle

Who's This App Delegate Guy?



Can we write some  
code already?

# WSLesson

```
@interface WSLesson : NSObject

@property (copy, nonatomic) NSString *title;
@property (strong, nonatomic) NSURL *url;

+ (NSArray *)workshopLessons;

@end

@implementation WSLesson

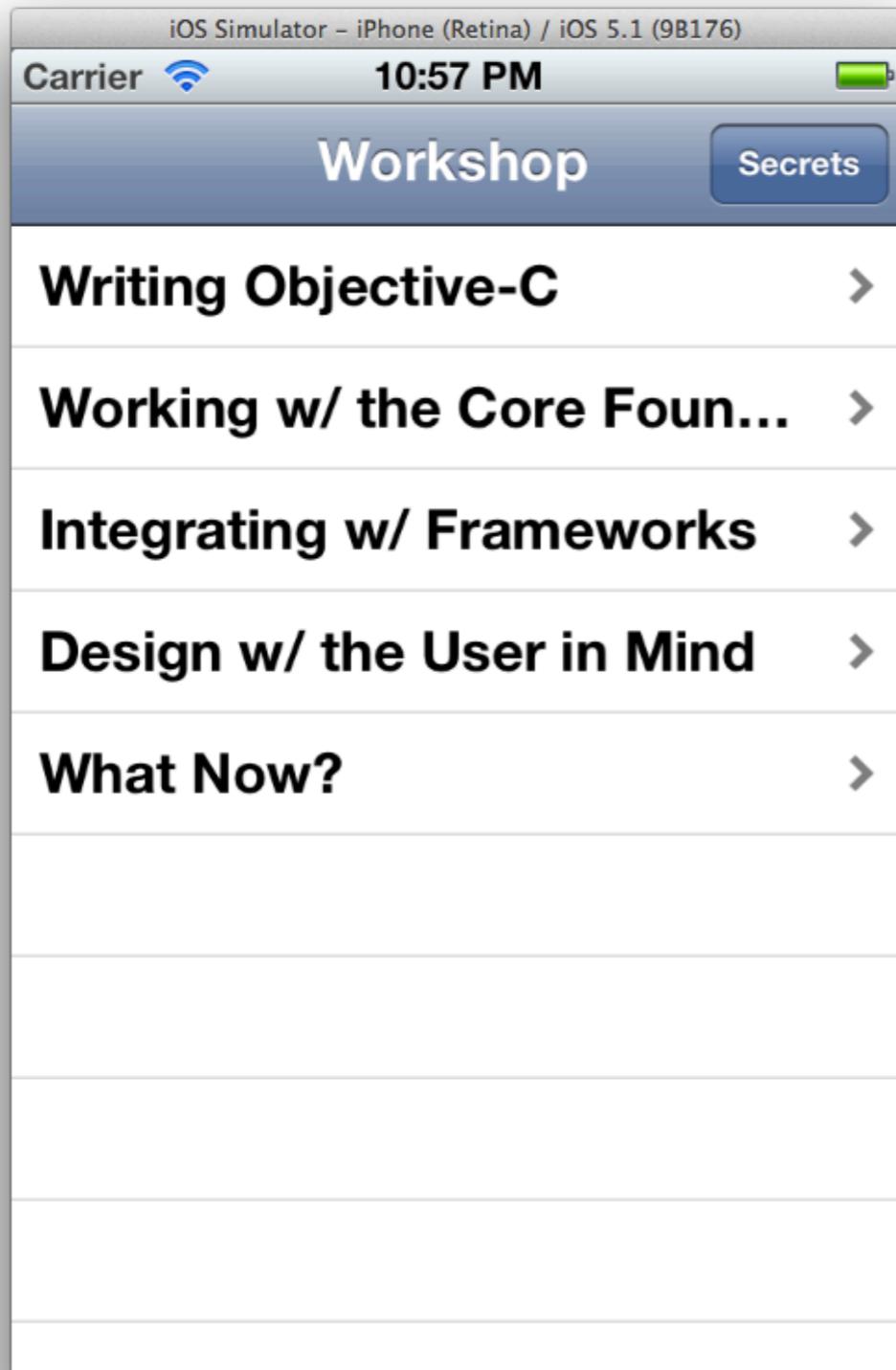
@synthesize title = _title;
@synthesize url = _url;

+ (NSArray *)workshopLessons
{
    WSLesson *lesson = [WSLesson new];
    lesson.title = @"Writing Objective-C";
    lesson.url = [NSURL URLWithString:@"https://developer.apple.com/library/ios/#referencelibrary/GettingStarted/WriteObjective-CCode/WriteObjective-CCode/WriteObjective-CCode.html"];

    return [[NSArray alloc] initWithObjects:lesson, nil];
}

@end
```

# WSMasterViewController



Let's populate the `WSMasterViewController`'s cells with `WSLesson` titles.

Then, we can use the `navigationController` to *push* on the `MSDetailViewController` when a cell is selected.

# Getting Started w/ Table Views

## Data Sources & Delegates

- ▶ **TODO (part 1a):** Populate each table cell's label text with a `WSLesson`'s title.
  - ▶ `tableView:cellForRowAtIndexPath:`
- ▶ **TODO (part 1b):** Use the `navigationController` to *push* the `detailViewController` onto the view stack when a user taps on a cell.
  - ▶ `tableView:didSelectRowAtIndexPath:`

# UINavigationController

A special kind of view controller. They manage a *stack* of child view controllers.

```
[self.navigationController pushViewController:self.detailViewController animated:YES];
```

A few Navigation Controller methods you may encounter...

- (id)initWithRootViewController:([UIViewController](#) \*)rootViewController
- (void)pushViewController:([UIViewController](#) \*)viewController animated:(BOOL)animated
- ([NSArray](#) \*)popViewControllerAnimated:(BOOL)animated

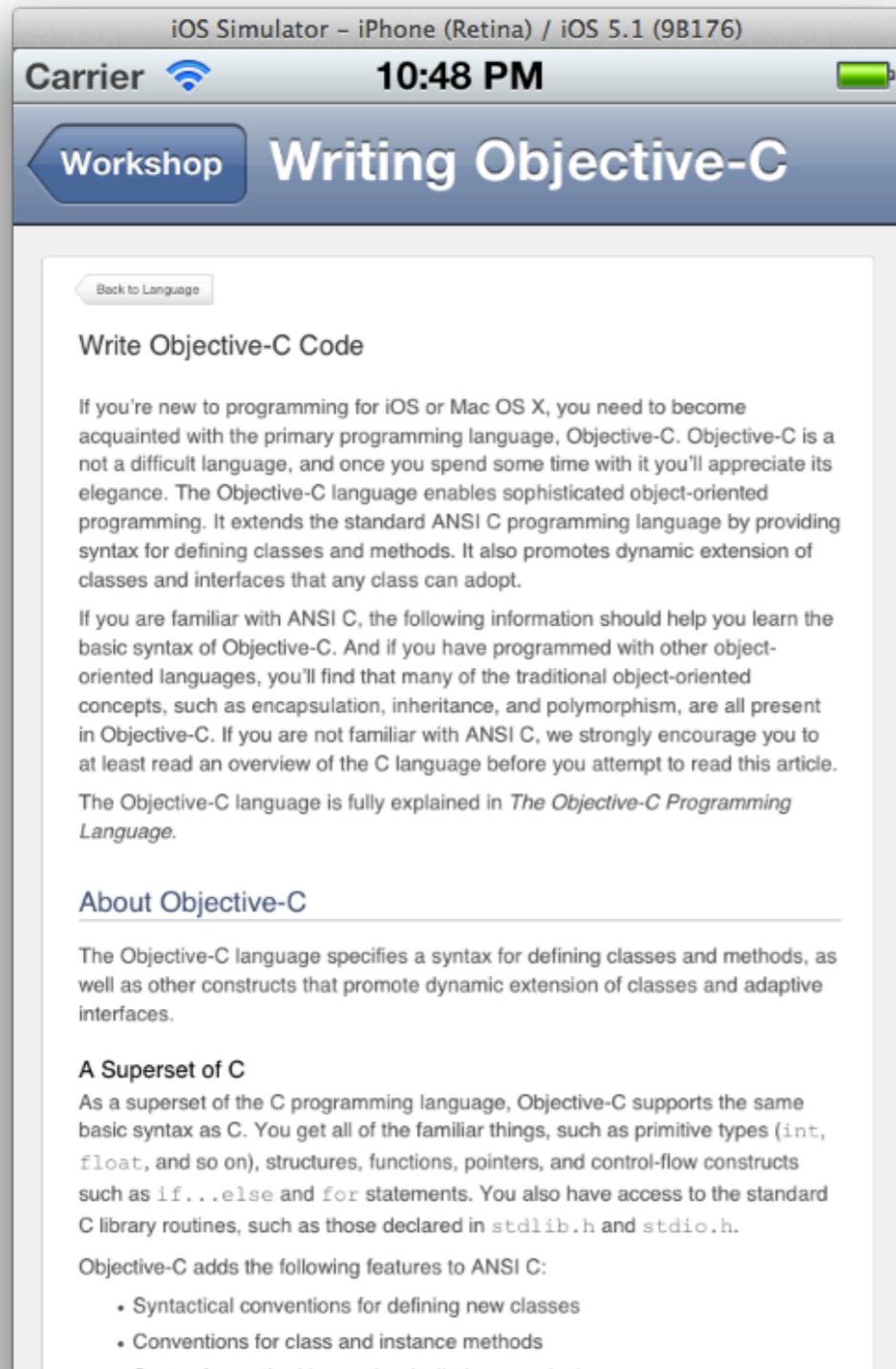
[UINavigationController Class Reference](#)

# Using WSDetailViewController

The detail view contains a `UIWebView` that will display lessons via the provided URL.

Let's populate the `WSDetailViewController` with a `WSLesson`'s title and lesson URL.

This will occur when a user has selected a particular cell in the `WSMasterViewController` list of lessons.



# Diving In

- ▶ **TODO (part 2a):** Add a `@property WSLesson` to `WSDetailViewController`. Don't forget to `@synthesize` in your implementation!
- ▶ **TODO (part 2b):** Set the selected lesson from `WSMasterViewController`'s *`didSelectRowAtIndexPath:`* on the `detailViewController`.

## WSDetailViewController.h

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

@interface WSDetailViewController : UIViewController

@property (strong, nonatomic) WSLesson *lesson;

@end
```

## WSDetailViewController.m

```
@implementation WSDetailViewController

@synthesize lesson = _lesson;
```

## WSMasterViewController.m

```
#pragma mark - Table view delegate

// Delegate method called when a user taps on a particular table cell at a given indexPath (section, row)
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    // creates a detail view controller, if one doesn't already exist
    if (!self.detailViewController) {
        self.detailViewController = [[WSDetailViewController alloc] initWithNibName:@"WSDetailViewController" bundle:nil];
    }

    WSLesson *lesson = [_workshopLessons objectAtIndex:indexPath.row];
    self.detailViewController.lesson = lesson;

    [self.navigationController pushViewController:self.detailViewController animated:YES];
}
```

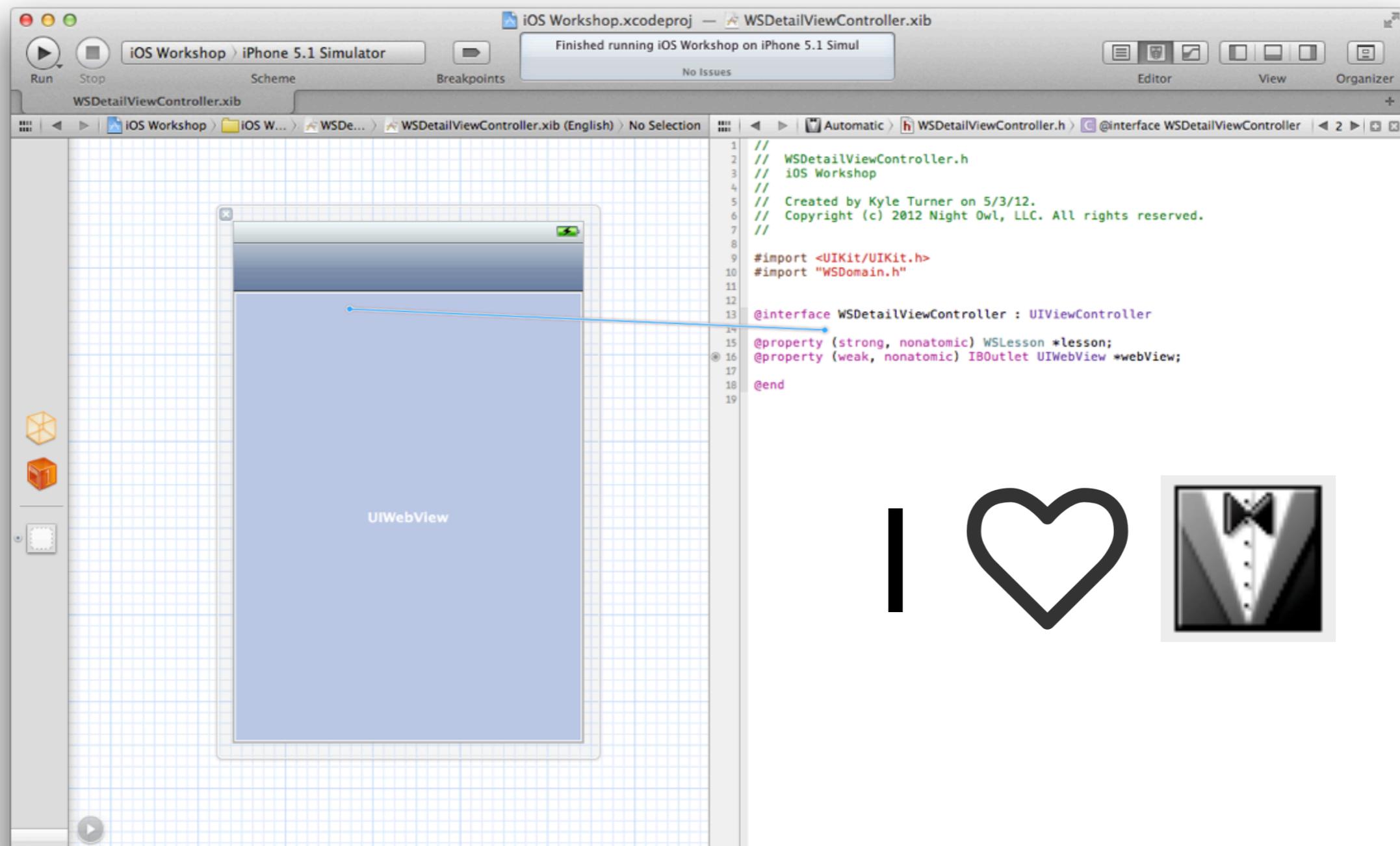
# Using the Detail View's WSLesson object

- ▶ **TODO (part 3a):** Set the detail view's navigation bar title to the lesson's title when the view loads.

```
self.title = self.lesson.title;
```

- ▶ **TODO (part 3b):** Add a UIWebView to the WSDetailViewController - it will soon load the lesson's URL!

# Wire up the WSDetailViewController's webView



# Loading Content via UIWebViews

- ▶ **TODO (part 4):** Load the detailViewController's Lesson URL in a webView.

```
[self.webView loadRequest:[NSURLRequest requestWithURL:self.lesson.url]];
```

Thanks for playing.

# More?

- *iOS Development Course From Stanford On iTunes Using iOS 5* – Free course provided by Apple & Stanford, covering beginner-advanced topics on iOS 5 development. Includes lecture videos, PowerPoints, assignments, etc.
- *Design With the User in Mind* - Overview of the Human Interface Guidelines.
- *Design Your App With Care* - What to look for when you're getting started on a project.
- *Human Interface Guidelines* - The Apple Bible on sticking to the user interface design standards.

**What about  
secrets??**

Launch the app, click  
“Secrets”, and rotate  
the Simulator upside  
down.

## Never Gonna Give You Up

We're no strangers to love  
You know the rules and so do I  
A full commitment's what I'm thinking of  
You wouldn't get this from any other guy

I just wanna tell you how I'm feeling  
Gotta make you understand

**Never gonna give you up,  
Never gonna let you down,  
Never gonna run around and desert you,  
Never gonna make you cry,  
Never gonna say goodbye,  
Never gonna tell a lie and hurt you**

We've known each other for so long  
Your heart's been aching but you're too shy to say it  
Inside we both know what's been going on  
We know the game and we're gonna play it

And if you ask me how I'm feeling  
Don't tell me you're too blind to see

We've known each other for so long  
Your heart's been aching but you're too shy to say it  
Inside we both know what's been going on  
We know the game and we're gonna play it

I just wanna tell you how I'm feeling  
Gotta make you understand