Navigation

IAP 2010 ❄

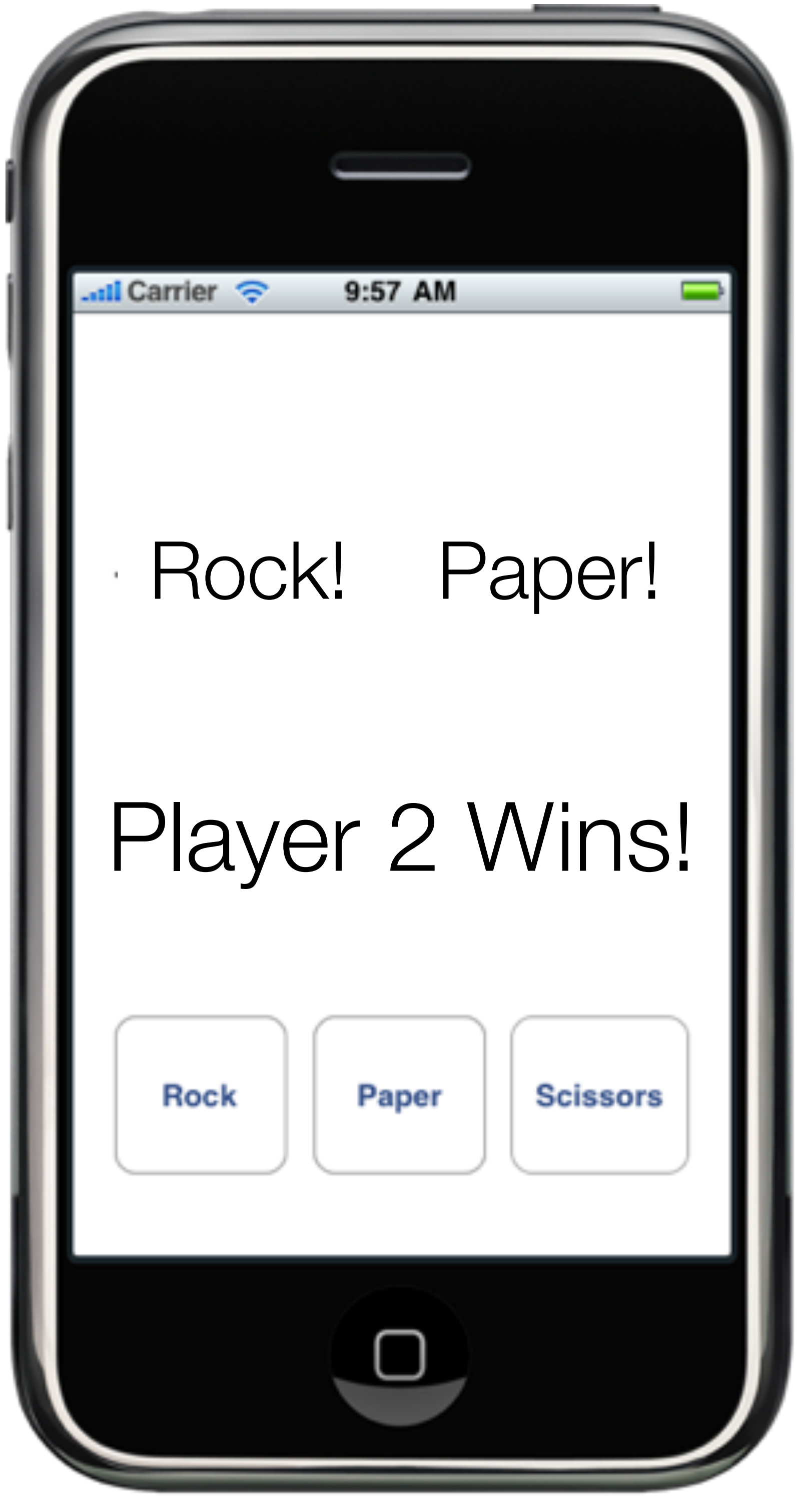
iphonedev.csail.mit.edu

edward benson / eob@csail.mit.edu

Today

- Programmatically Created Views
- Tab Bar Controllers
- Modal View Controllers &
- Tables
- Navigation Controllers

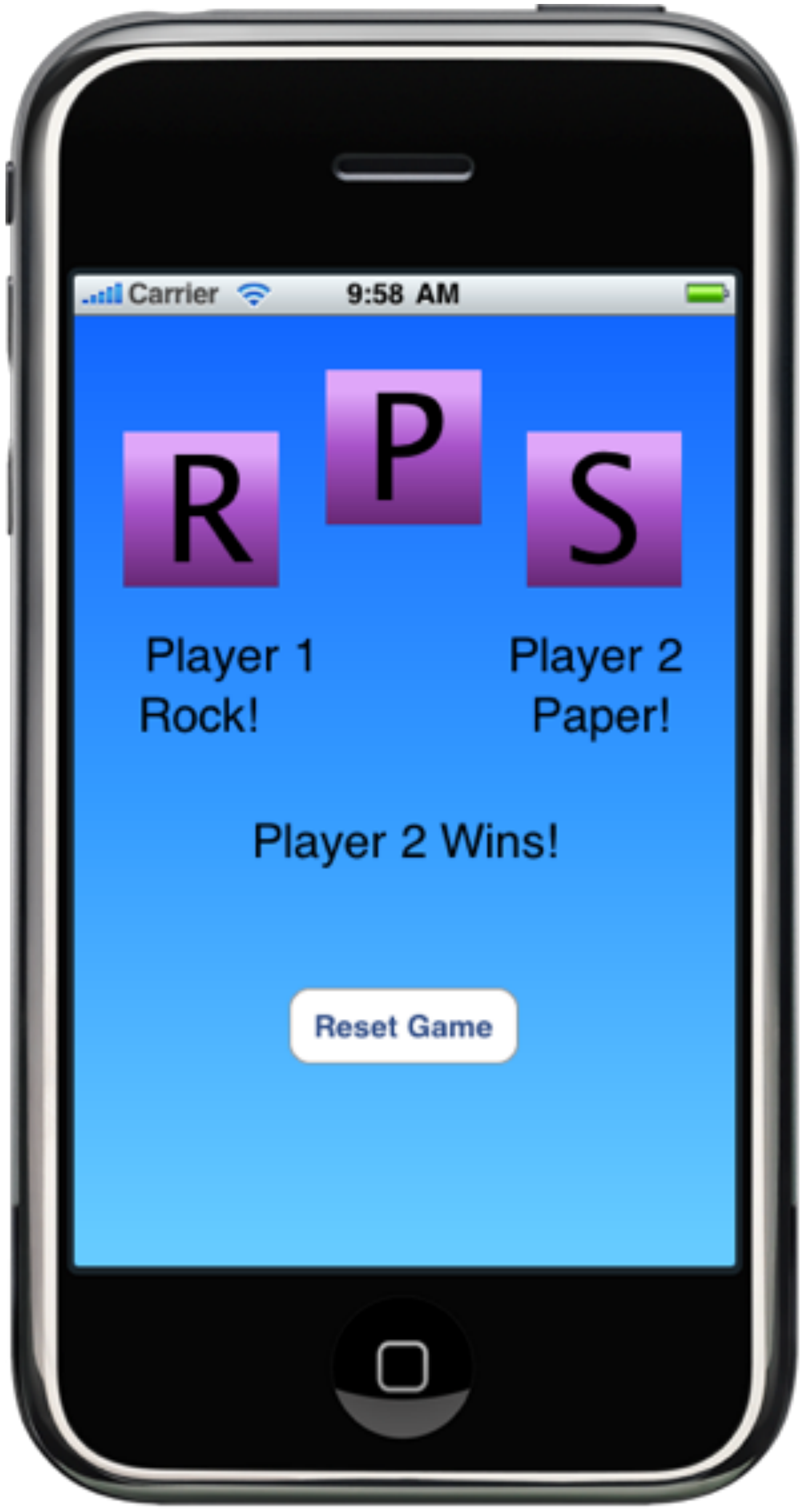
How many of you would pay 99c for this?



Rock! Paper!

Player 2 Wins!

How many of you would pay 99c for this?

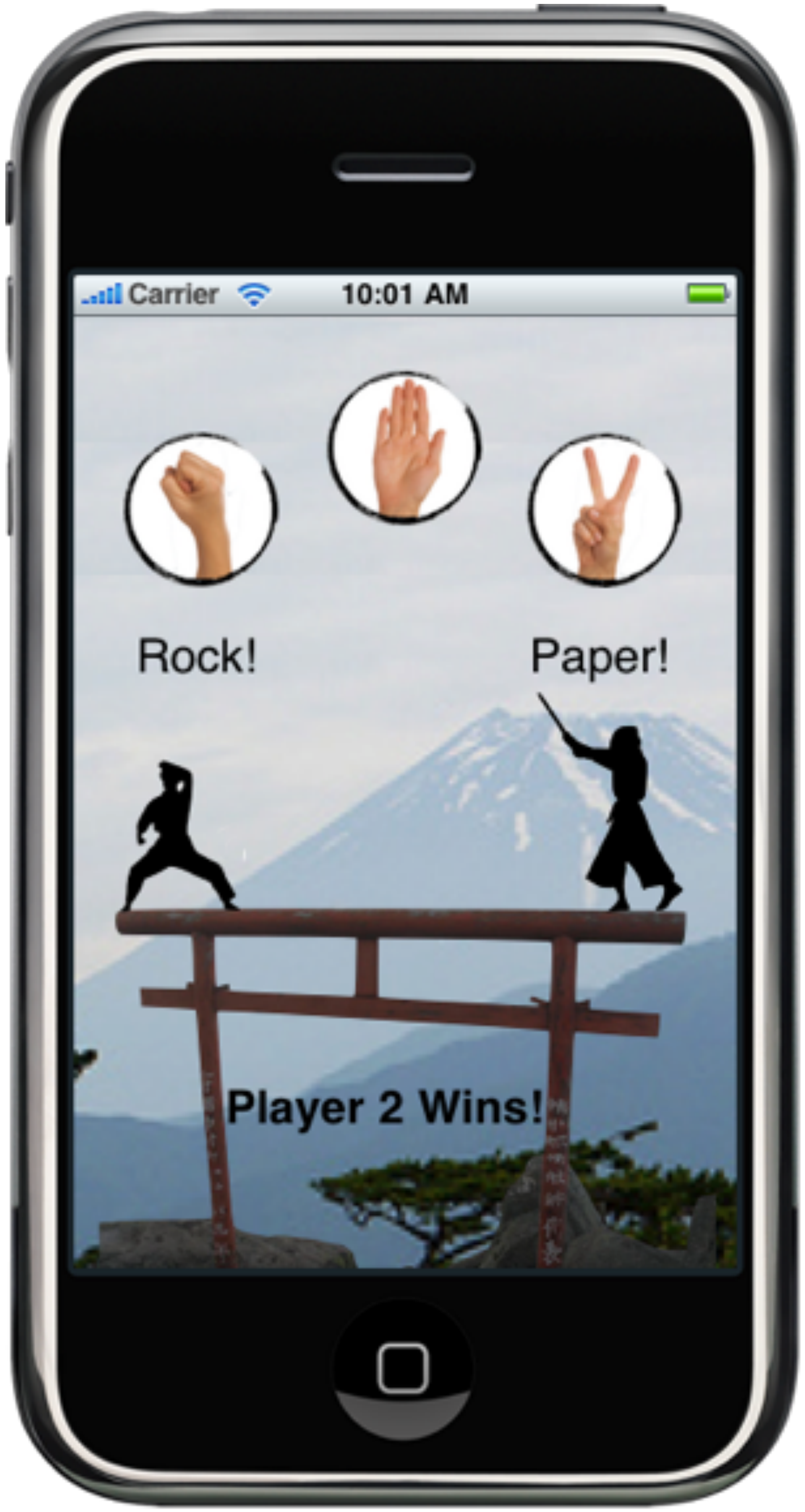


Player 1
Rock!

Player 2
Paper!

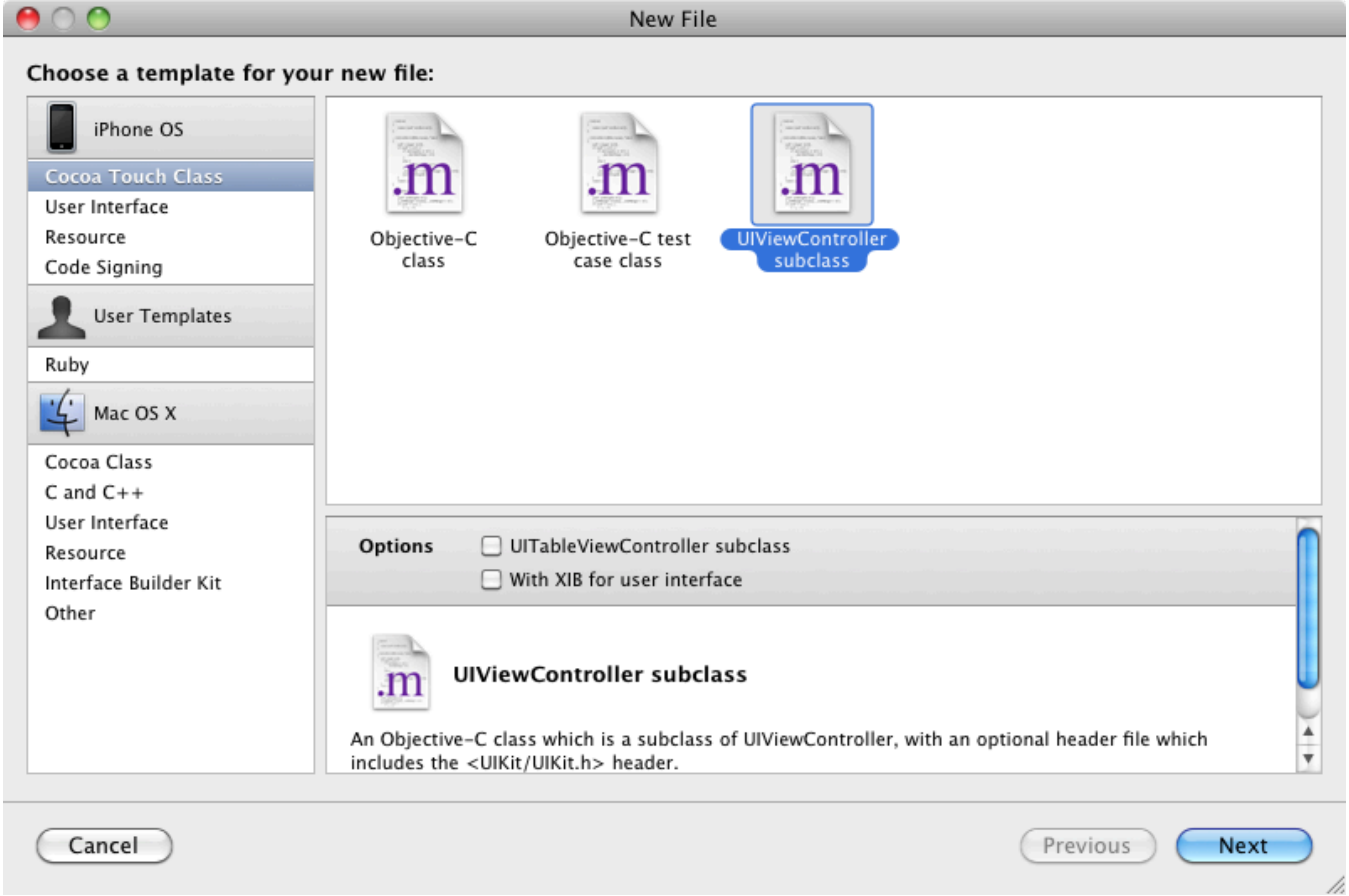
Player 2 Wins!

How many of you would pay 99c for this?



Interface is important.

Programmatically Created Views



Choose a template for your new file:

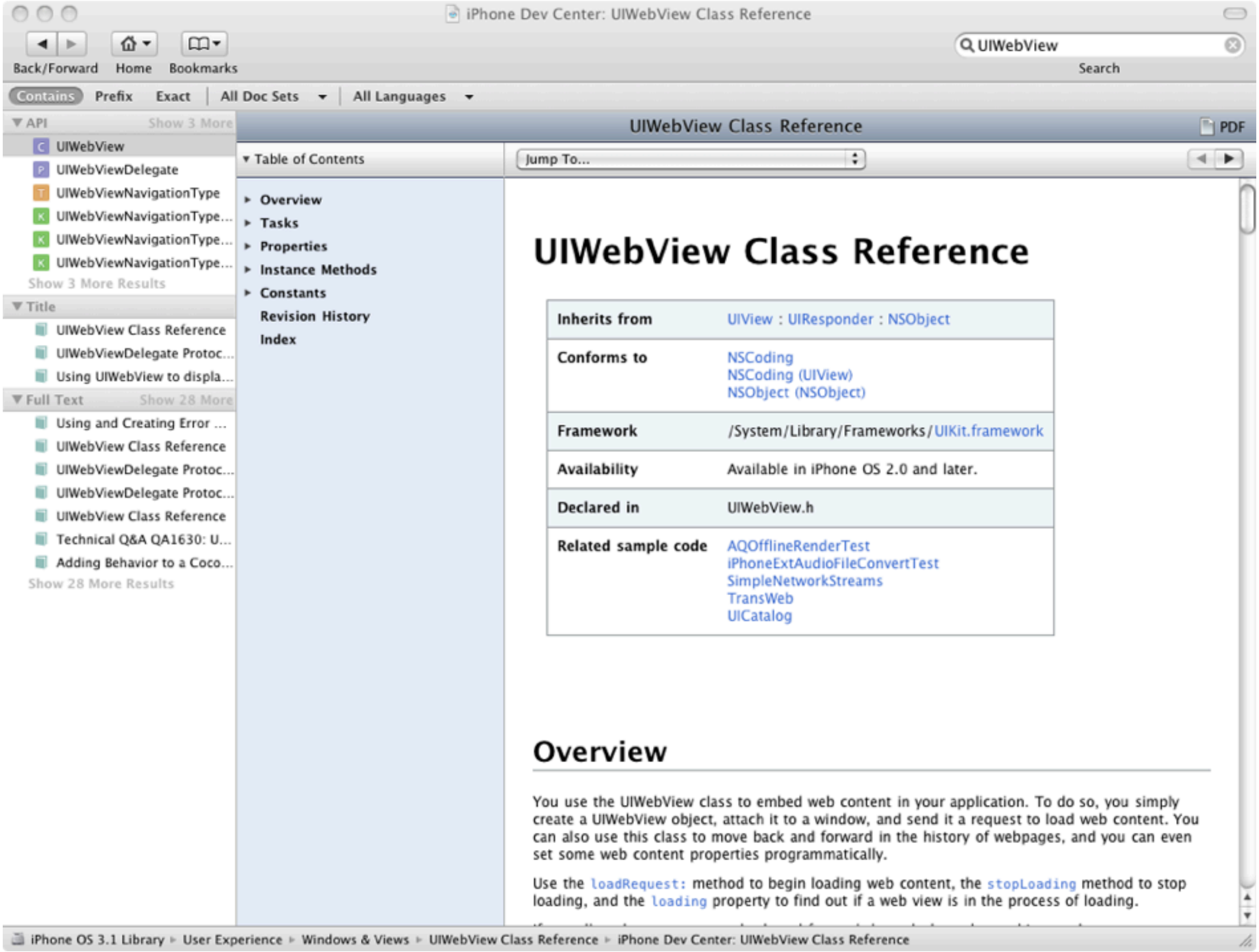
Options
□ UITableViewController subclass
□ With XIB for user interface

UIViewController subclass

An Objective-C class which is a subclass of UIViewController, with an optional header file which includes the <UIKit/UIKit.h> header.

Name it **AboutController**

```
/*
// Implement loadView to create a view
// hierarchy programmatically, without using a nib.

- (void)loadView {
}

*/
```



API Show 3 More

- UIWebView
- UIWebViewDelegate
- UIWebViewNavigationType
- UIWebViewNavigationType...
- UIWebViewNavigationType...
- UIWebViewNavigationType...

Show 3 More Results

Title

- UIWebView Class Reference
- UIWebViewDelegate Protoc...
- Using UIWebView to displa...

Full Text Show 28 More

- Using and Creating Error ...
- UIWebView Class Reference
- UIWebViewDelegate Protoc...
- UIWebViewDelegate Protoc...
- UIWebView Class Reference
- Technical Q&A QA1630: U...
- Adding Behavior to a Coco...

Show 28 More Results

UIWebView Class Reference

Table of Contents

UIWebView Class Reference

Inherits from	UIView : UIResponder : NSObject
Conforms to	NSCoding NSCoding (UIView) NSObject (NSObject)
Framework	/System/Library/Frameworks/UIKit.framework
Availability	Available in iPhone OS 2.0 and later.
Declared in	UIWebView.h
Related sample code	AQOfflineRenderTest iPhoneExtAudioFileConvertTest SimpleNetworkStreams TransWeb UICatalog

Overview

You use the UIWebView class to embed web content in your application. To do so, you simply create a UIWebView object, attach it to a window, and send it a request to load web content. You can also use this class to move back and forward in the history of webpages, and you can even set some web content properties programmatically.

Use the `loadRequest:` method to begin loading web content, the `stopLoading` method to stop loading, and the `loading` property to find out if a web view is in the process of loading.

iPhone OS 3.1 Library ▸ User Experience ▸ Windows & Views ▸ UIWebView Class Reference ▸ iPhone Dev Center: UIWebView Class Reference

```
- (void)loadView {
  NSURL *url = [NSURL URLWithString:@"http://www.google.com"];
}
```

```
- (void)loadView {
  NSURL *url = [NSURL URLWithString:@"http://www.google.com"];
  NSURLRequest *request = [NSURLRequest requestWithURL:url];
}
```

```
- (void)loadView {
  NSURL *url = [NSURL URLWithString:@"http://www.google.com"];
  NSURLRequest *request = [NSURLRequest requestWithURL:url];
  UIWebView *webView = [[UIWebView alloc] init];
}
```

```
- (void)loadView {
  NSURL *url = [NSURL URLWithString:@"http://www.google.com"];
  NSURLRequest *request = [NSURLRequest requestWithURL:url];
  UIWebView *webView = [[UIWebView alloc] init];
  [webView loadRequest:request];
}
```

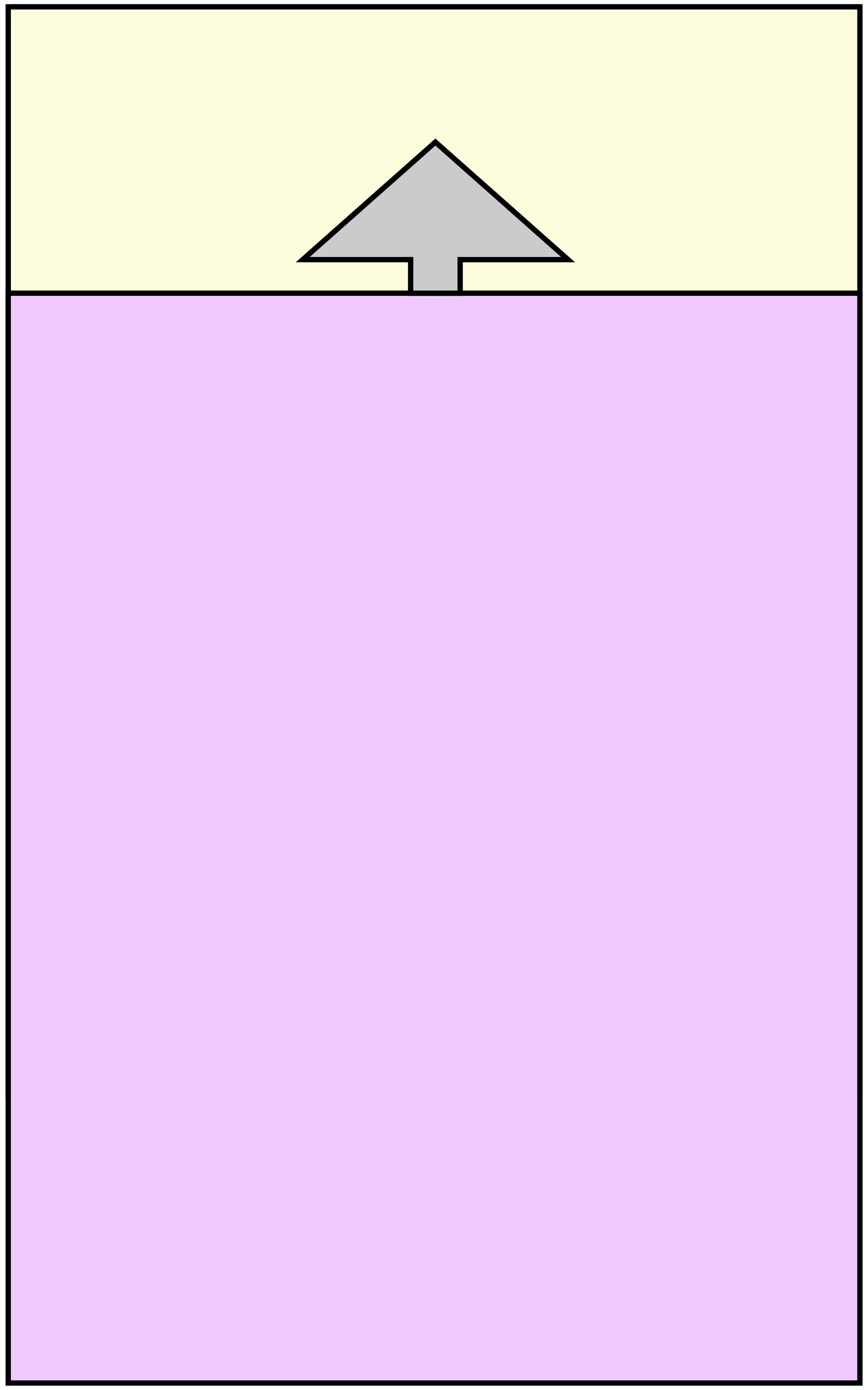
```
- (void)loadView {
  NSURL *url = [NSURL URLWithString:@"http://www.google.com"];
  NSURLRequest *request = [NSURLRequest requestWithURL:url];
  UIWebView *webView = [[UIWebView alloc] init];
  [webView loadRequest:request];

  self.view = webView;

}
```

```
- (void)loadView {
  NSURL *url = [NSURL URLWithString:@"http://www.google.com"];
  NSURLRequest *request = [NSURLRequest requestWithURL:url];
  UIWebView *webView = [[UIWebView alloc] init];
  [webView loadRequest:request];

  self.view = webView;

  [webView release];
}
```

Modal Views

(How do we show it?)

Modal View Controllers (we interrupt this message to bring you...)



RPSGameViewController.h

```
- (IBAction)aboutClicked:(id)sender;
```

RPSGameViewController.m

```
- (IBAction)aboutClicked:(id)sender {
  AboutController *ac = [[[AboutController alloc] init] autorelease];
  [self presentModalViewController:ac animated:YES];
}
```

Now let’s wire it up in IB..

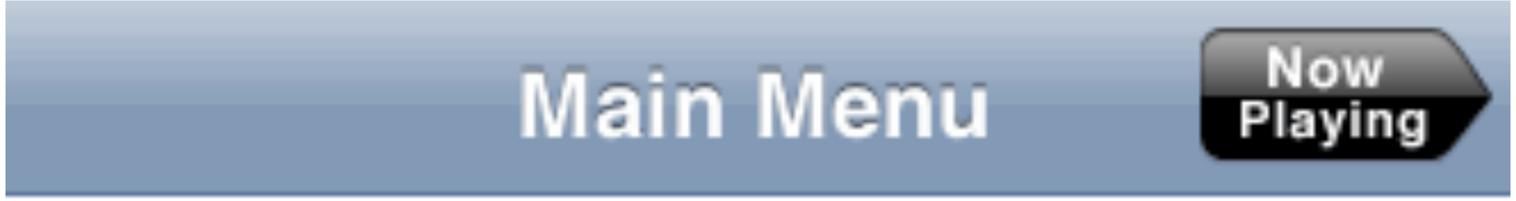
Ex3-1.zip

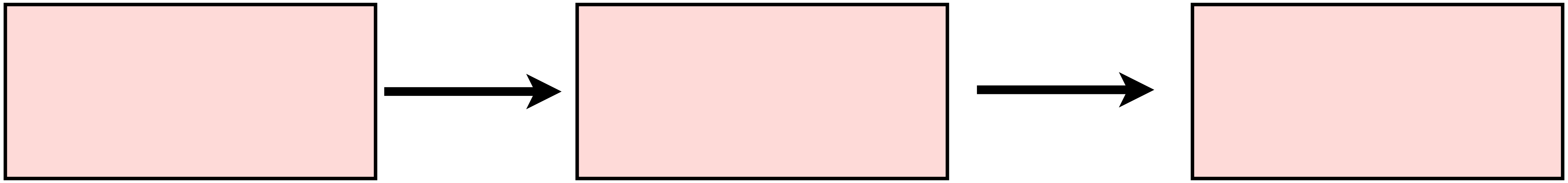
So how do we get rid of the modal view?

.. (in a minute)

Navigation Controllers

Navigation Controllers (Push, Pop views)



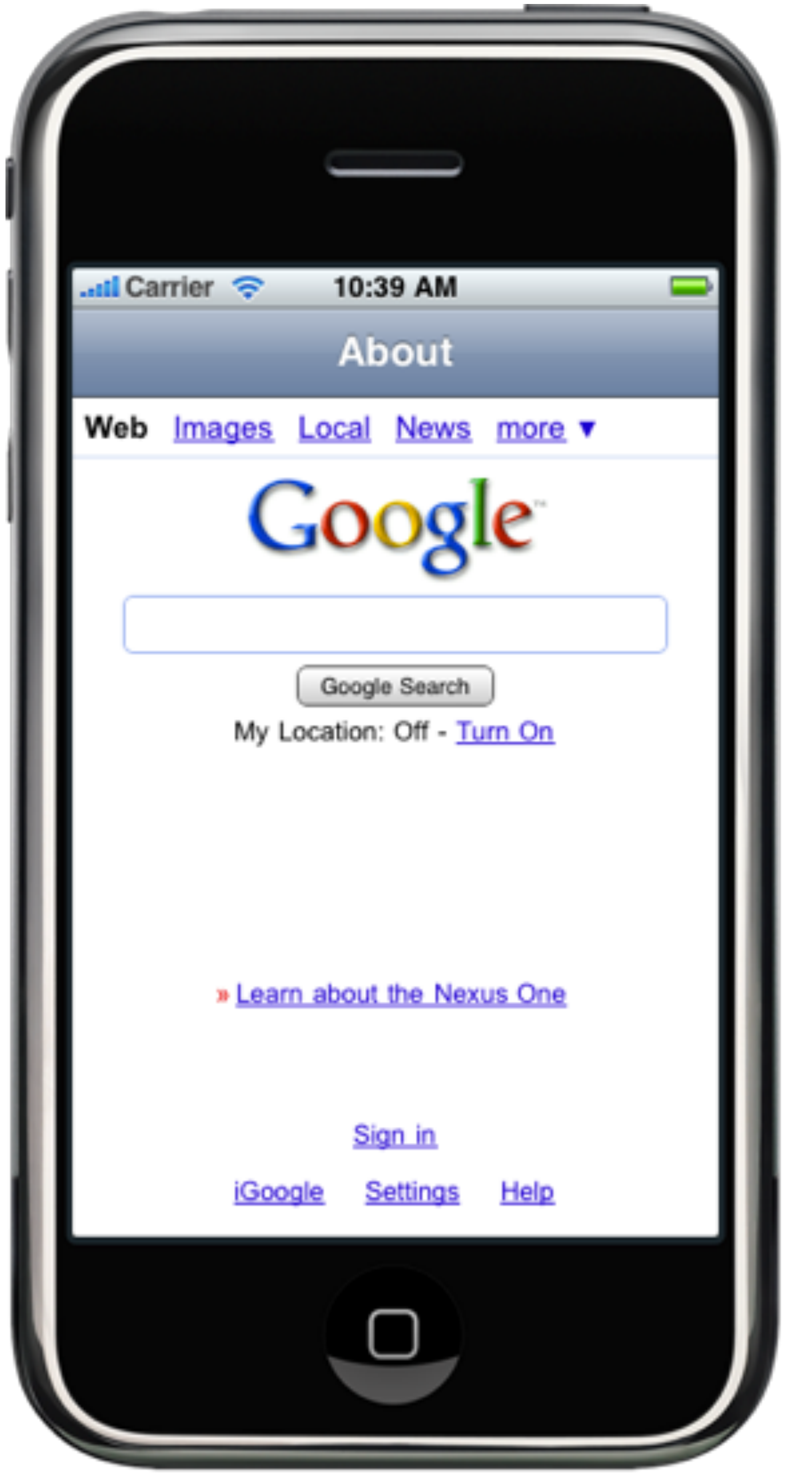


RPSGameViewController.m

```
- (IBAction)aboutClicked:(id)sender {
  AboutController *ac = [[[AboutController alloc] init] autorelease];
  UINavigationController *nav =
      [[UINavigationController alloc] initWithRootViewController:ac];
  [self presentModalViewController:nav animated:YES];
  [nav release];
}
```

AboutController.m

```
-(id)init {
  if (self = [super init]) {
    self.title = @"About";
  }
  return self;
}
```



AboutController.m

```
-(id)init {
  if (self = [super init]) {
     self.title = @"About";

     UIBarButtonItem *dismiss = [[UIBarButtonItem alloc] init];

  }
  return self;
}
```

AboutController.m

```
-(id)init {
  if (self = [super init]) {
     self.title = @"About";

     UIBarButtonItem *dismiss = [[UIBarButtonItem alloc] init];
     dismiss.title = @"Dismiss";

  }
  return self;
}
```

AboutController.m

```
-(id)init {
  if (self = [super init]) {
     self.title = @"About";

     UIBarButtonItem *dismiss = [[UIBarButtonItem alloc] init];
     dismiss.title = @"Dismiss";
     dismiss.target = self;

  }
  return self;
}
```

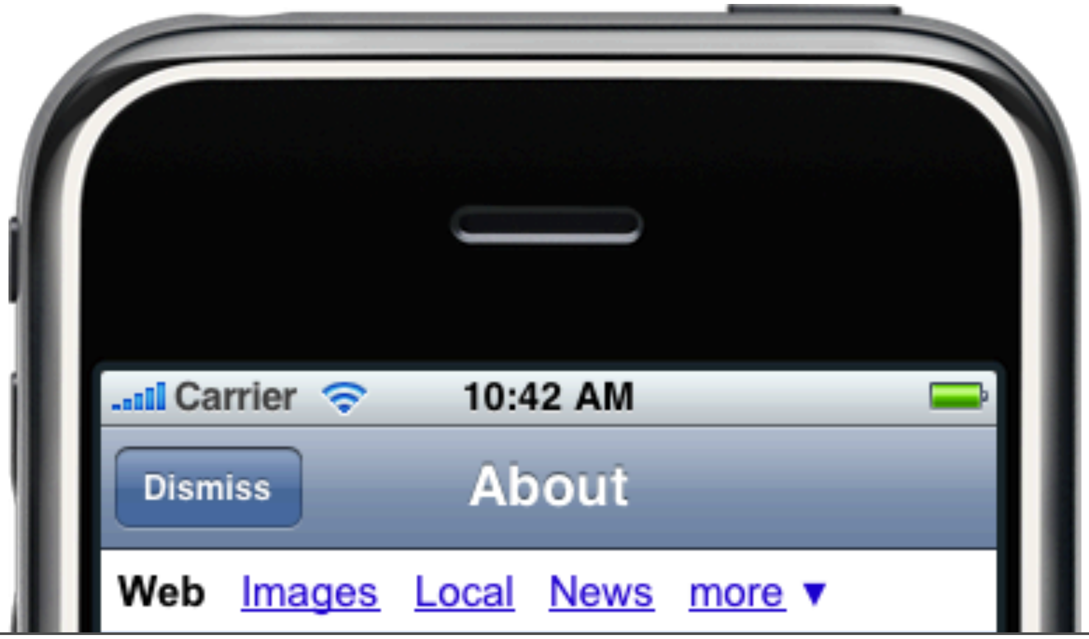
AboutController.m

```
-(id)init {
  if (self = [super init]) {
     self.title = @"About";

     UIBarButtonItem *dismiss = [[UIBarButtonItem alloc] init];
     dismiss.title = @"Dismiss";
     dismiss.target = self;
     dismiss.action = @selector(dismissClicked);

  }
  return self;
}
```

AboutController.m

```
-(id)init {
  if (self = [super init]) {
     self.title = @"About";

     UIBarButtonItem *dismiss = [[UIBarButtonItem alloc] init];
     dismiss.title = @"Dismiss";
     dismiss.target = self;
     dismiss.action = @selector(dismissClicked);
     self.navigationItem.leftBarButtonItem = dismiss;

  }
  return self;
}
```

Why does it crash?



AboutController.m

```
-(void)dismissClicked {
  [self dismissModalViewControllerAnimated:YES];
}
```

What happens if we don’t put this in the .h file?

Ex3-2.zip

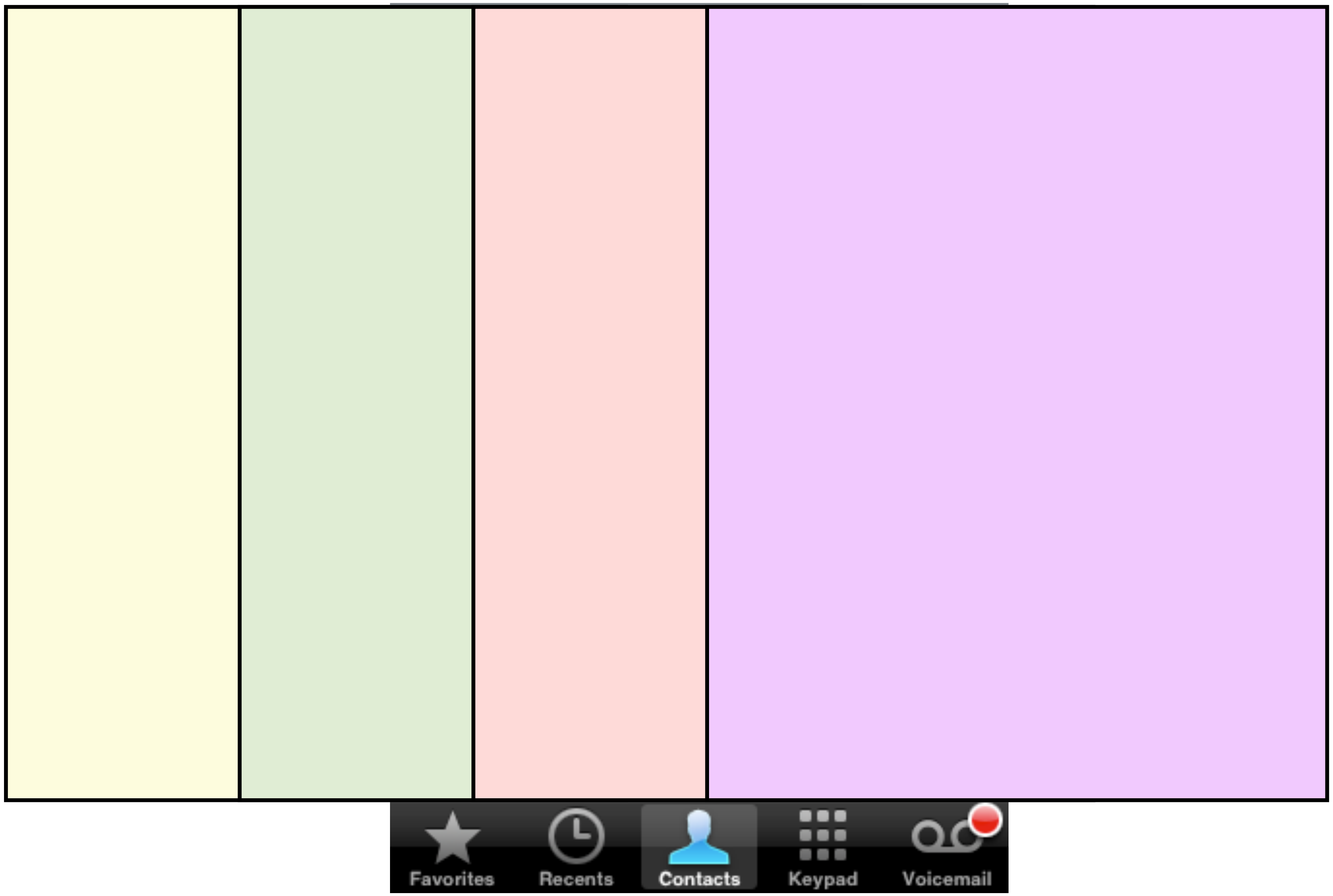
Navigation Controllers

We misused the NavigationController here.
It was just an easy way to get the Navigation **Bar** .

We'll revisit them in a bit to
see how they are really used.

Tab Bar Controllers

Tab Bars (Multiplex between views)



UITabBarController

“Wrapping Controllers”

We used a Navigation Controller by **wrapping** it around another controller

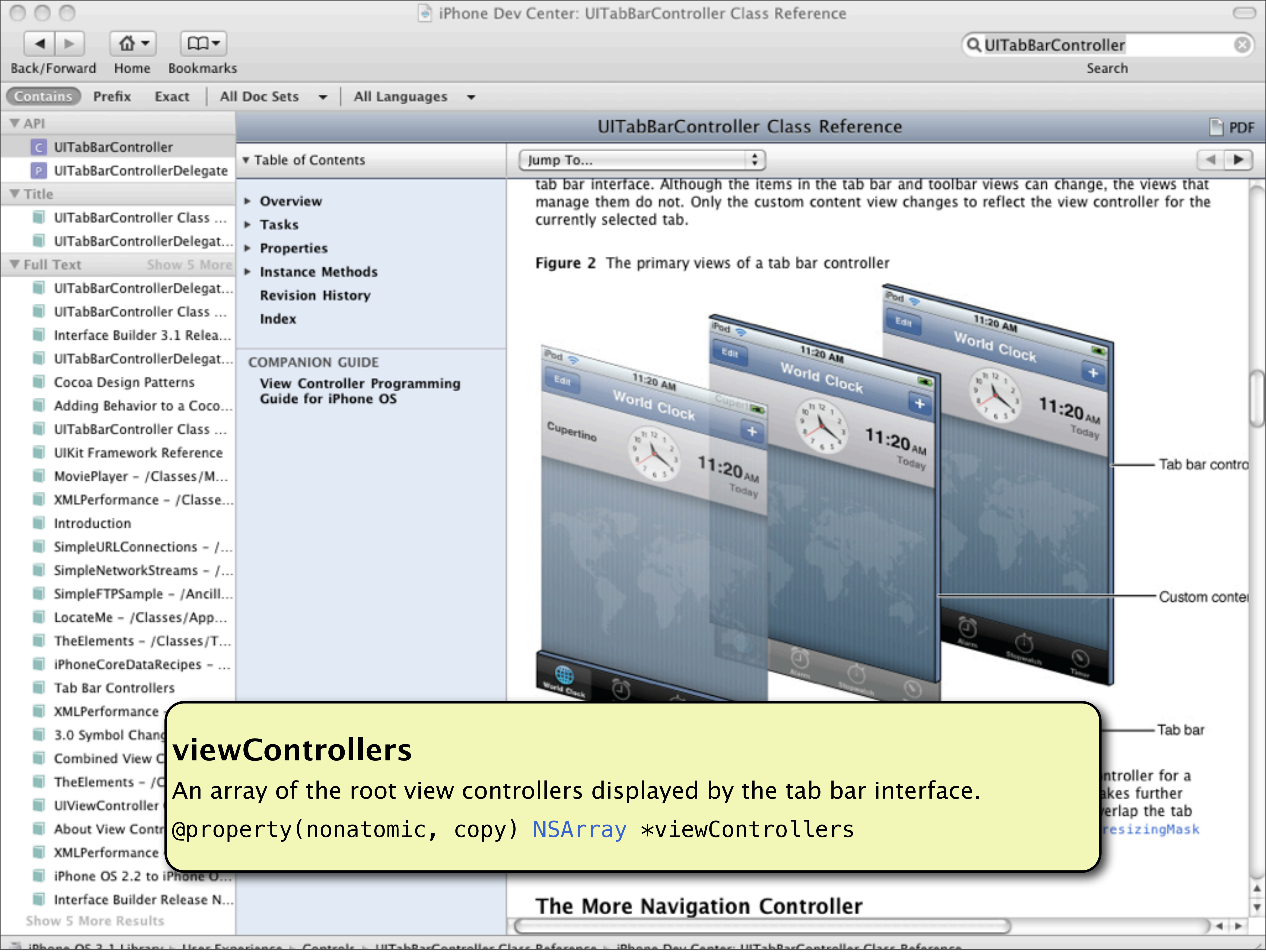
```
[[UINavigationController alloc] initWithRootViewController:ac];
```

It knew what title to show because of the **title** property on its child controller.

Remember our app delegate

```
- (void)applicationDidFinishLaunching:(UIApplication *)applicatio
{
  // Override point for customization after app launch
  gameViewController = [[RPSGameViewController alloc] init];
  [window addSubview:gameViewController.view];
  [window makeKeyAndVisible];
}
```

To use a tab bar controller, we just “wrap” controllers in a similar way.



▼ API

- UITabBarController
- UITabBarControllerDelegate

▼ Title

- UITabBarController Class ...
- UITabBarControllerDelegat...

▼ Full Text Show 5 More

- UITabBarControllerDelegat...
- UITabBarController Class ...
- Interface Builder 3.1 Relea...
- UITabBarControllerDelegat...
- Cocoa Design Patterns
- Adding Behavior to a Coco...
- UITabBarController Class ...
- UIKit Framework Reference
- MoviePlayer – /Classes/M...
- XMLPerformance – /Classe...
- Introduction
- SimpleURLConnections – /...
- SimpleNetworkStreams – /...
- SimpleFTPSample – /Ancill...
- LocateMe – /Classes/App...
- TheElements – /Classes/T...
- iPhoneCoreDataRecipes – ...
- Tab Bar Controllers
- XMLPerformance
- 3.0 Symbol Chang
- Combined View C
- TheElements – /C
- UIViewController
- About View Contr
- XMLPerformance
- iPhone OS 2.2 to iPhone O...
- Interface Builder Release N...

Show 5 More Results

UITabBarController Class Reference

▼ Table of Contents

COMPANION GUIDE

View Controller Programming Guide for iPhone OS

tab bar interface. Although the items in the tab bar and toolbar views can change, the views that manage them do not. Only the custom content view changes to reflect the view controller for the currently selected tab.

Figure 2 The primary views of a tab bar controller

ntroller for a akes further erlap the tab resizingMask

The More Navigation Controller

iPhone OS 3.1 Library › User Experience › Controls › UITabBarController Class Reference › iPhone Dev Center: UITabBarController Class Reference

viewControllers

An array of the root view controllers displayed by the tab bar interface.

```
@property(nonatomic, copy) NSArray *viewControllers
```

In the App Delegate

```
- (void)applicationDidFinishLaunching:(UIApplication *)application {
    // Override point for customization after app launch
   gameViewController = [[RPSGameViewController alloc] init];

   AboutController *a1 = [[[AboutController alloc] init] autorelease];
   AboutController *a2 = [[[AboutController alloc] init] autorelease];
   AboutController *a3 = [[[AboutController alloc] init] autorelease];

   UITabBarController *tabs = [[UITabBarController alloc] init];
   tabs.viewControllers = [NSArray arrayWithObjects:gameViewController, a1, a2, a3, nil];

   [window addSubview:tabs.view];
   [window makeKeyAndVisible];
}
```

Create three **AboutController** instances

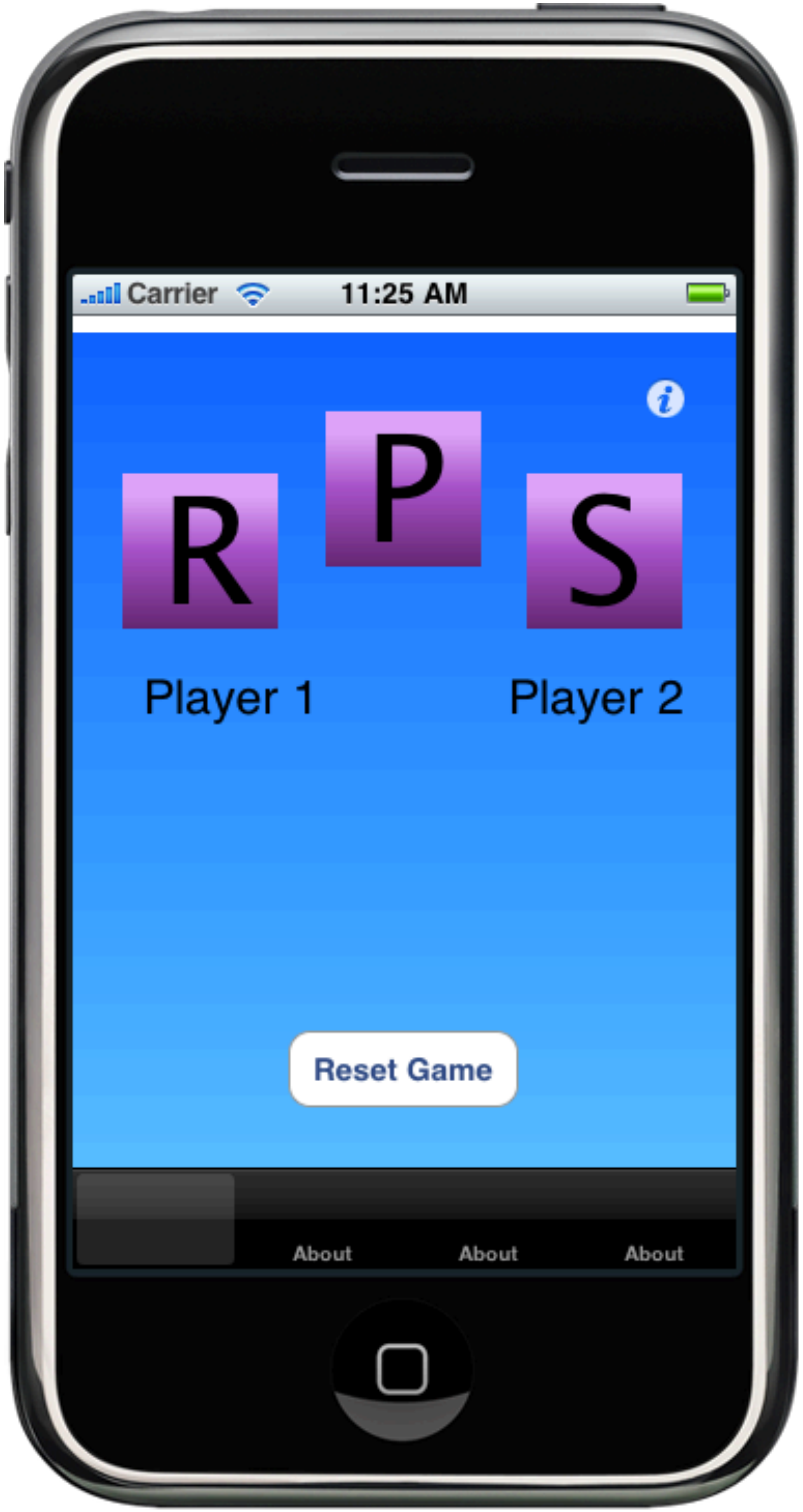
`#import "AboutController.h"` <- Remember, at the top

In the App Delegate

```
- (void)applicationDidFinishLaunching:(UIApplication *)application {
    // Override point for customization after app launch
   gameViewController = [[RPSGameViewController alloc] init];

   AboutController *a1 = [[[AboutController alloc] init] autorelease];
   AboutController *a2 = [[[AboutController alloc] init] autorelease];
   AboutController *a3 = [[[AboutController alloc] init] autorelease];

   UITabBarController *tabs = [[UITabBarController alloc] init];
   tabs.viewControllers = [NSArray arrayWithObjects:gameViewController, a1, a2, a3, nil];

   [window addSubview:tabs.view];
   [window makeKeyAndVisible];
}
```

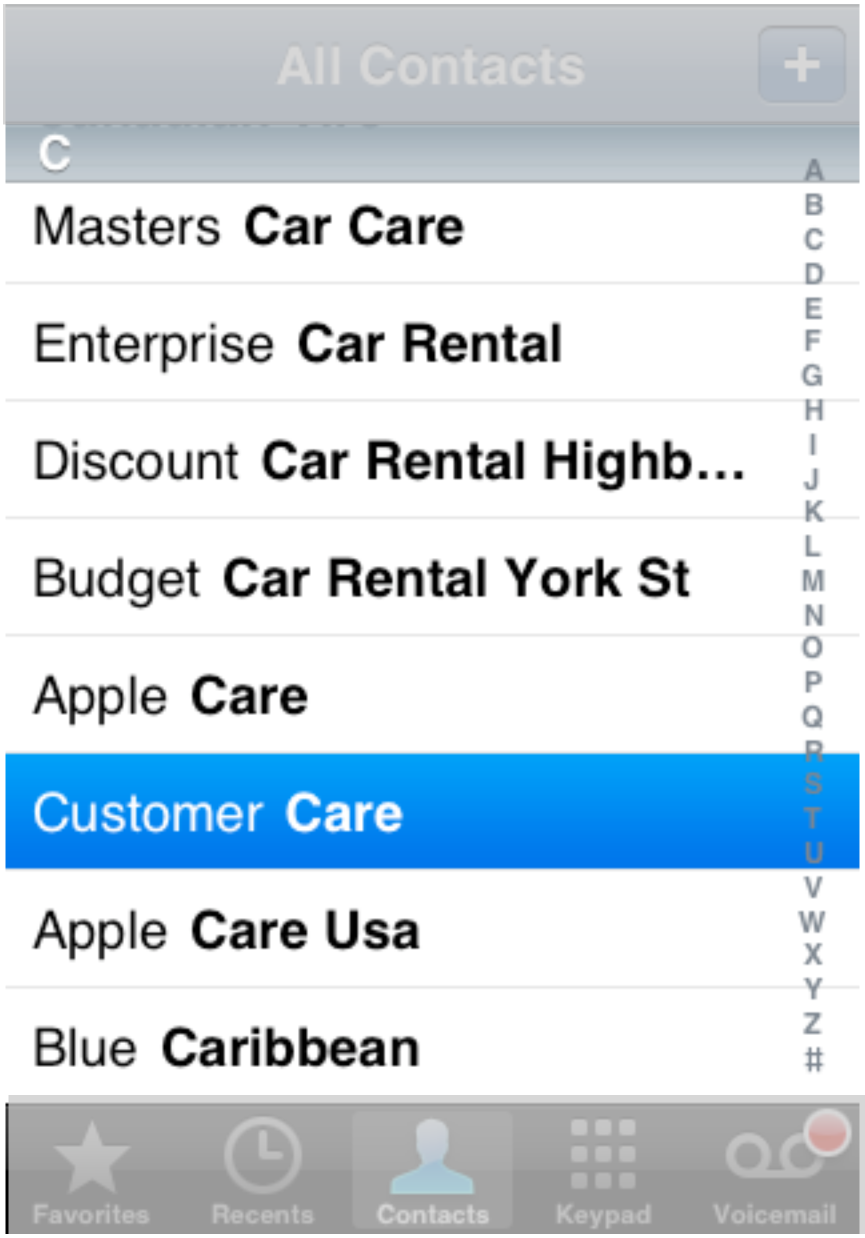
In the App Delegate

```
- (void)applicationDidFinishLaunching:(UIApplication *)application {
    // Override point for customization after app launch
   gameViewController = [[RPSGameViewController alloc] init];

   AboutController *a1 = [[[AboutController alloc] init] autorelease];
   AboutController *a2 = [[[AboutController alloc] init] autorelease];
   AboutController *a3 = [[[AboutController alloc] init] autorelease];

   UITabBarController *tabs = [[UITabBarController alloc] init];
   tabs.viewControllers = [NSArray arrayWithObjects:gameViewController, a1, a2, a3, nil];

   [window addSubview:tabs.view];
   [window makeKeyAndVisible];
}
```


Carrier
11:25 AM
R
P
S
Player 1
Player 2
Reset Game
About
About
About

Tables

Tables (Show lists)



Masters Car Care

Enterprise Car Rental

Discount Car Rental Highb...

Budget Car Rental York St

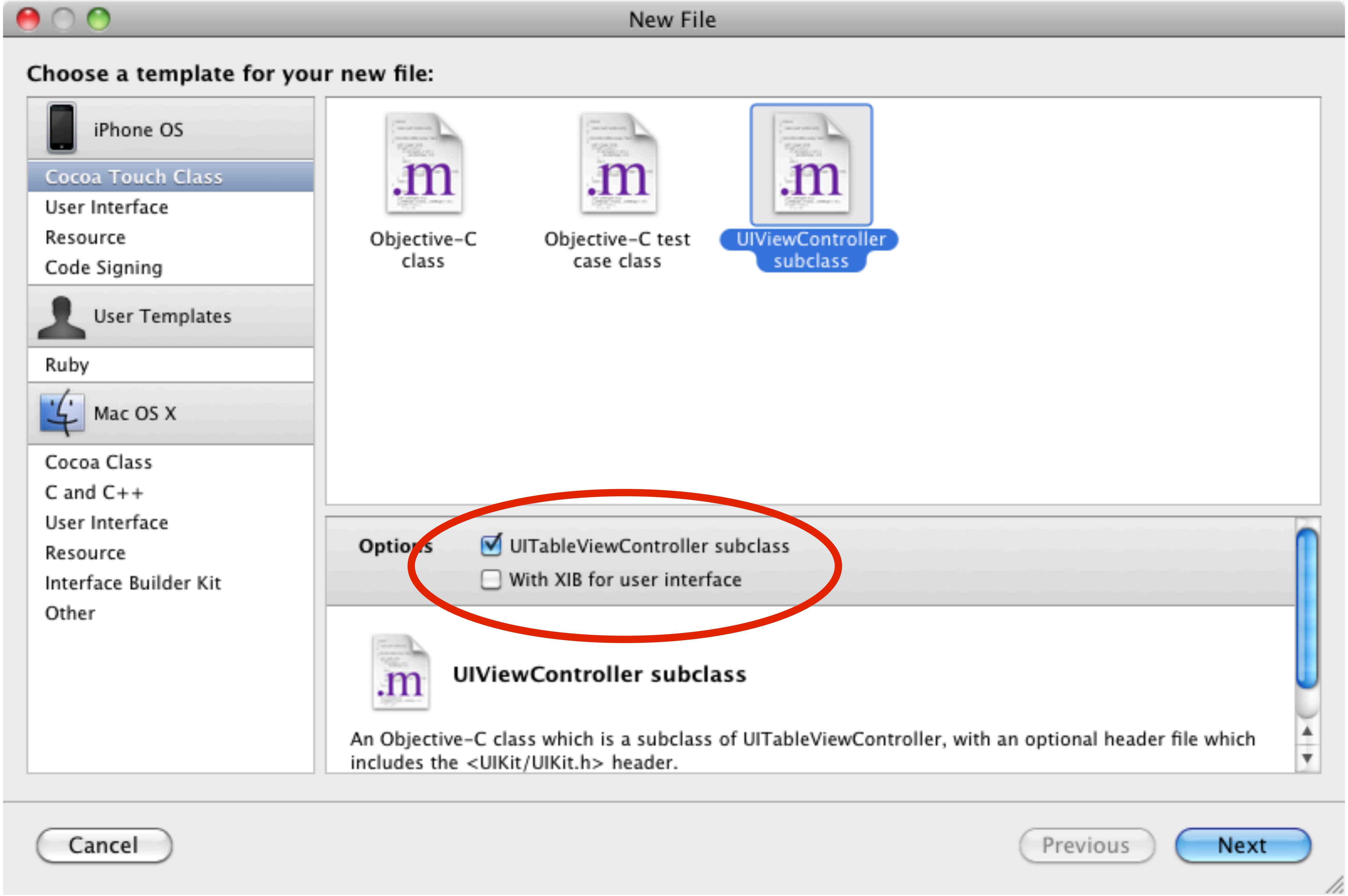
Apple Care

Customer Care

Apple Care Usa

Blue Caribbean

UITableViewController



Choose a template for your new file:

Options ☑ UITableViewController subclass
□ With XIB for user interface

UIViewController subclass

An Objective-C class which is a subclass of UITableViewController, with an optional header file which includes the <UIKit/UIKit.h> header.

Call it **GameHistory** Controller

In the App Delegate

Ex3-4.zip

```
- (void)applicationDidFinishLaunching:(UIApplication *)application {
    // Override point for customization after app launch
   gameViewController = [[RPSGameViewController alloc] init];

   GameHistoryController *history = [[[GameHistoryController alloc] init] autorelease];
   UINavigationController *historyWrapper = [[[UINavigationController alloc]
                                       initWithRootViewController:history] autorelease];

   UITabBarController *tabs = [[UITabBarController alloc] init];
   tabs.viewControllers = [NSArray arrayWithObjects:gameViewController,
                                                    historyWrapper,
                                                    nil];


   [window addSubview:tabs.view];
   [window makeKeyAndVisible];
}
```

Remove the AboutController instances,
Create a GameHistoryController
Wrap it in a NavigationController

`#import "GameHistoryController.h"` <- Remember, at the top

Looking at a Table Delegate

UITableViewDataSource Protocol

Methods that inform the table about its contents

UITableViewDelegate Protocol

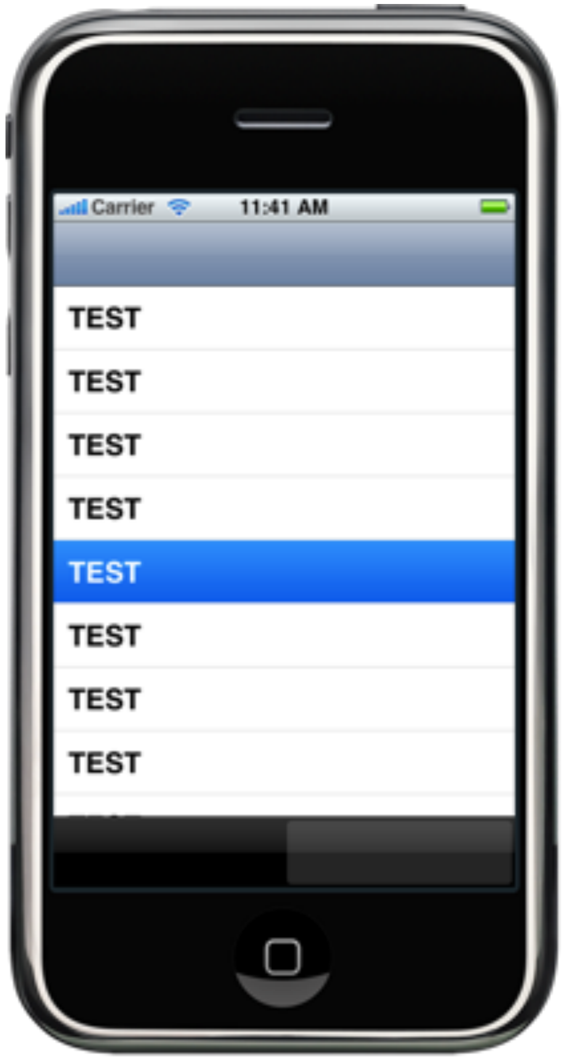
Methods related to user action on the table and table style manipulation

UITableViewController Class

A starting base for you

```
// Customize the number of rows in the table view.
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section {
    return 20;
}
```

```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {

....
    // Set up the cell...
   cell.text = @"TEST";
```



```
(void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath

  // Navigation logic may go here. Create and push another view controller.

GameHistoryController *g = [[GameHistoryController alloc] init];
[self.navigationController pushViewController:g animated:YES];
[g release];
```

Ex3-5.zip

(demo of other ways to manipulate the table)