

CS121 Views tutorial

Intro to views and re-intro to buttons.

Purple bubbles give you information you'll need to know.

Yellow Bubbles tell you what to do.

Orange bubbles tell you what you're not expected to understand yet. ☺

Choose a template for your project



Create a new project but this time make it an Empty Application.

Master-Detail Application



Tabbed Application

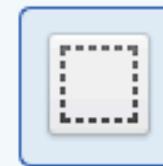
OpenGL Game



Utility Application



Page-Based Application



Empty Application



Single View Application

Name your project ViewTest. As in the previous tutorial use your name in Organization: FirstName LastName (e.g. Elizabeth Sweedyk)
Company Identifier: com.FirstNameLastName (e.g. com.ElizabethSweedyk)
Class Prefix: FirstInitialLastInitial (e.g. ES)

Cancel

Previous

Next

1. Open the Supporting Files folder.

The screenshot shows the Xcode interface with a project named "viewTest". The left sidebar displays the project structure, including a "Supporting Files" folder which contains the "main.m" file. The main editor window shows the content of "main.m". The code in "main.m" is as follows:

```
// main.m
// viewTest
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "AppDelegate.h"

int main(int argc, char * argv[])
{
    @autoreleasepool {
        return UIApplicationMain(argc, argv, nil, NSStringFromClass([AppDelegate class]));
    }
}
```

A yellow callout bubble points to the "Supporting Files" folder in the sidebar, and another yellow callout bubble points to the "main.m" file in the list. A pink callout bubble on the right side of the screen contains the explanatory text.

2. Open main.m.

This is our main! Main launches our app by instantiating the AppDelegate class.

An empty application still has an AppDelegate, which launches the app, but no ViewController is generated.

The screenshot shows the Xcode interface with the project 'viewTest' selected. In the left sidebar, the 'viewTest' group is expanded, showing files like 'ESAppDelegate.h', 'ESAppDelegate.m', 'InfoPlist.strings', and 'viewTest-Prefix.pch'. The main editor window displays the 'ESAppDelegate.h' file, which contains the following code:

```
// ESAppDelegate.h
// viewTest
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.

#import <UIKit/UIKit.h>

@interface ESAppDelegate : UIResponder <UIApplicationDelegate>
@property (strong, nonatomic) UIWindow *window;
@end
```

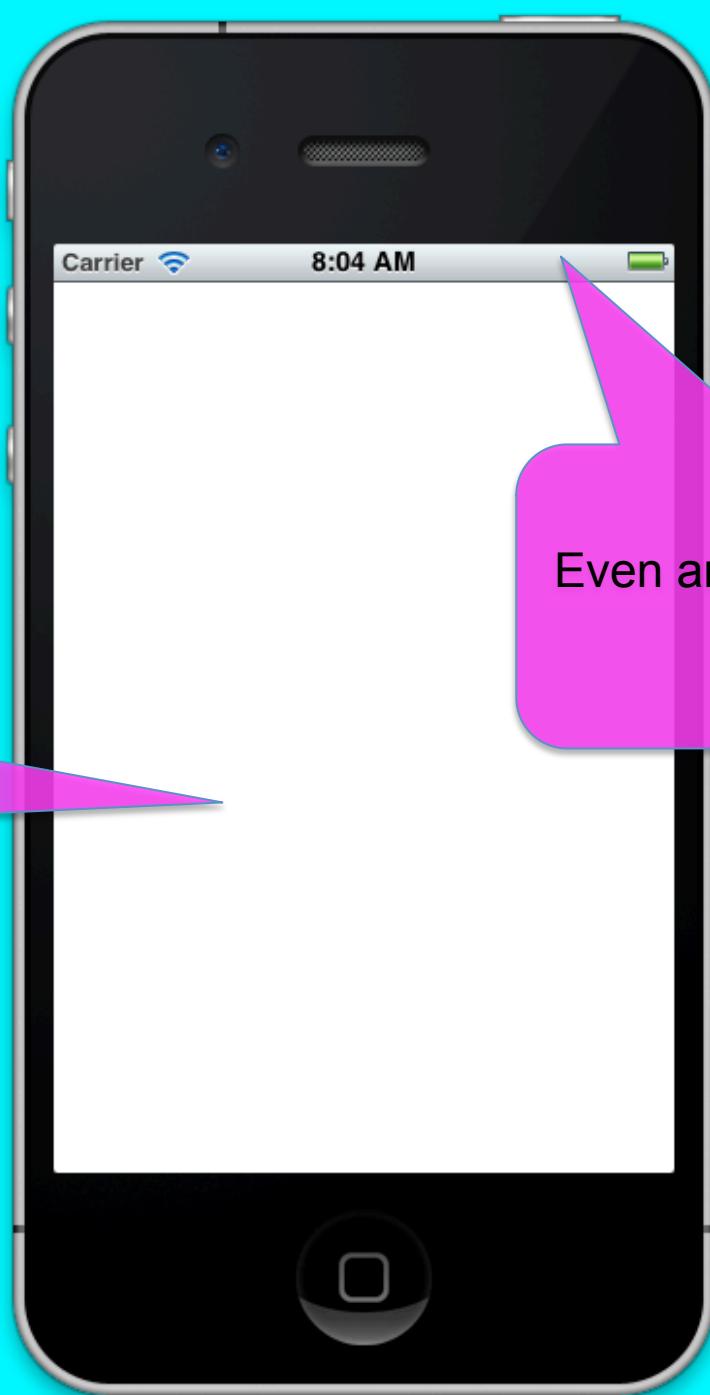
A yellow callout bubble points to the 'ESAppDelegate.h' file in the sidebar with the text: "Open the AppDelegate header file."

A pink callout bubble points to the code in the editor with the text: "Every app needs a window. Here is ours."

The right side of the Xcode interface shows the Utilities panel with sections for "Quick Help" and "No Quick Help". Below the editor, there is a list of button styles: "Push Button", "Gradient Button", and "Rounded Rect Button", each with a small icon and a brief description.

Run the app and check out the simulator.

The white region is our window.



Even an empty app has a status bar.

1. Open the AppDelegate source file.

```
//  
// ESAppDelegate  
// viewTest  
//  
// Created by Elizabeth Sweedyk on 8/14/14.  
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.  
//  
#import "ESAppDelegate.h"  
  
@implementation ESAppDelegate  
  
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];  
    // Override point for customization after application launch.  
    self.window.backgroundColor = [UIColor whiteColor];  
    [self.window makeKeyAndVisible];  
    return YES;  
}  
  
- (void)applicationWillResignActive:(UIApplication *)application  
{  
    // Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary  
    // interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the  
    // transition to the background state.  
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this  
    // method to pause the game.  
}  
  
- (void)applicationDidEnterBackground:(UIApplication *)application  
{  
    // This method to release shared resources, save user data, invalidate timers, and store enough application state  
    // information to restore your application to its current state in case it is terminated later.  
    // If your application supports background execution, this method is called instead of applicationWillTerminate: when the  
    // user quits.  
}  
  
- (void)applicationWillEnterForeground:(UIApplication *)application  
{  
    // Called as part of the transition from the background to the inactive state when the user tapped once on the home screen.  
    // on entering the background.  
}  
  
- (void)applicationDidBecomeActive:(UIApplication *)application  
{  
    // Restart any tasks that were paused (or not yet started) while the application was  
    // previously in the background, optionally refresh the user interface.  
}  
  
- (void)applicationWillTerminate:(UIApplication *)application  
{  
    // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.  
}
```

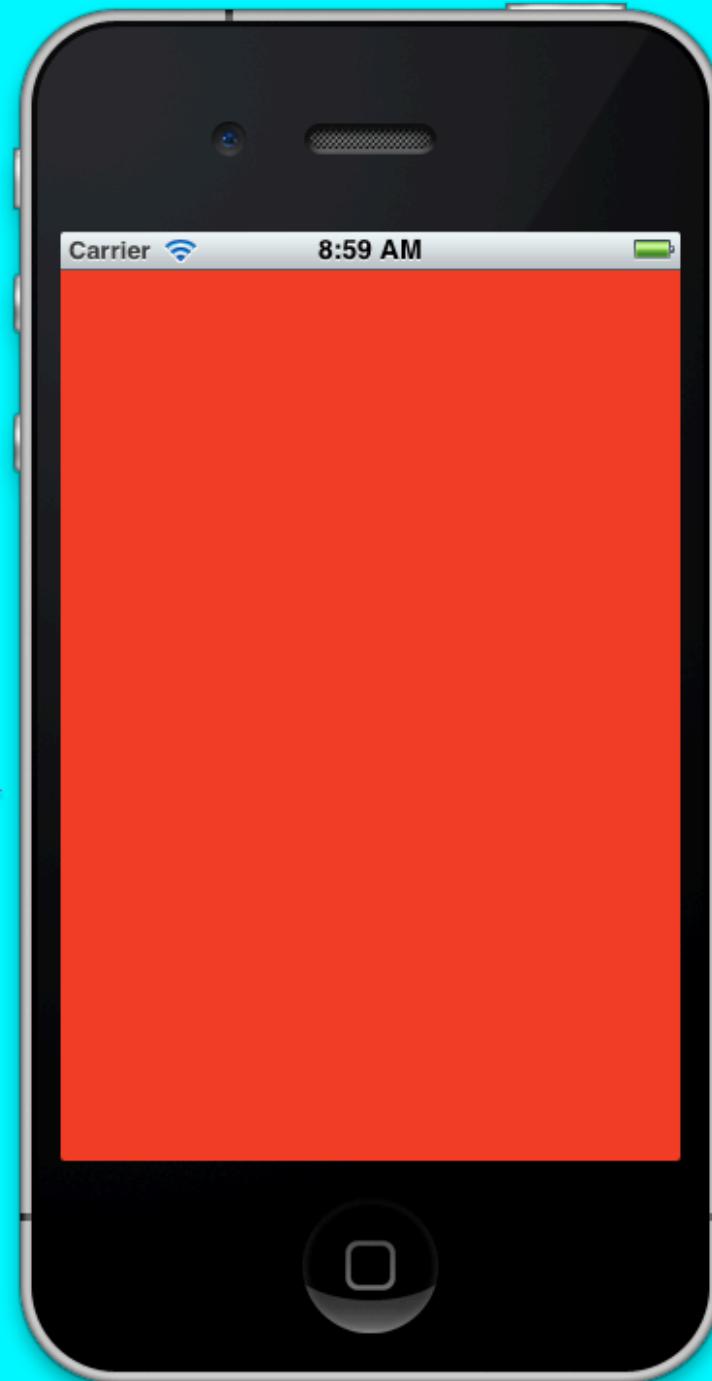
This line instantiates the window.

This line makes the window white.

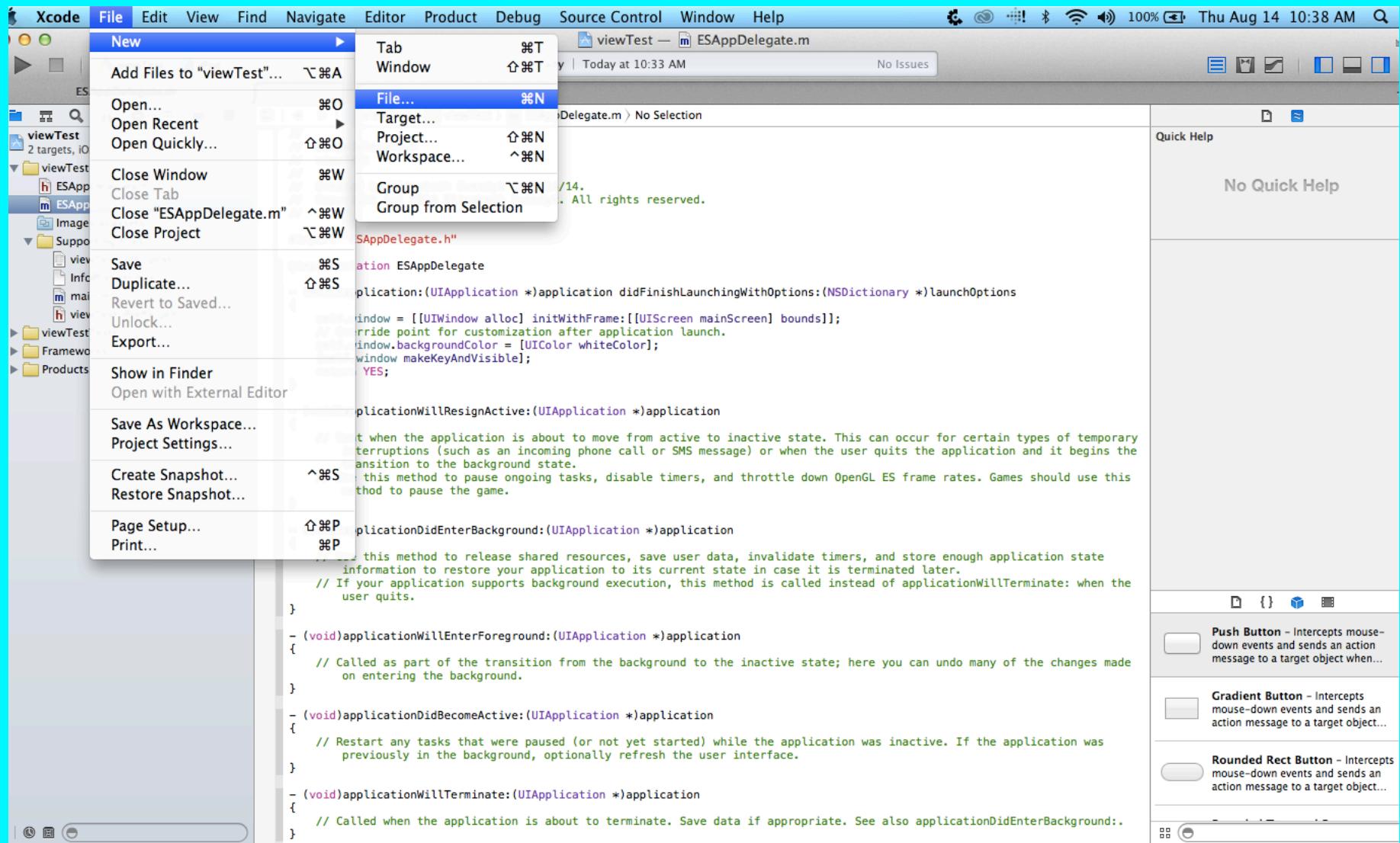
2. Change whiteColor to redColor.

Run the app and check out
the simulator.

Now our window is red.



Select the File tab
then New and New
File.



Choose a template for your new file:



Cocoa Touch

C and C++

User Interface

Core Data

Resource

Other

cocos2d v2.x



Cocoa

C and C++

User Interface

Core Data

Resource

Other

cocos2d v2.x



Objective-C class



UIViewController
subclass



Objective-C
category



Objective-C protocol

1. Select Cocoa Touch.

2. Select Objective-C Class.

Cancel

Previous

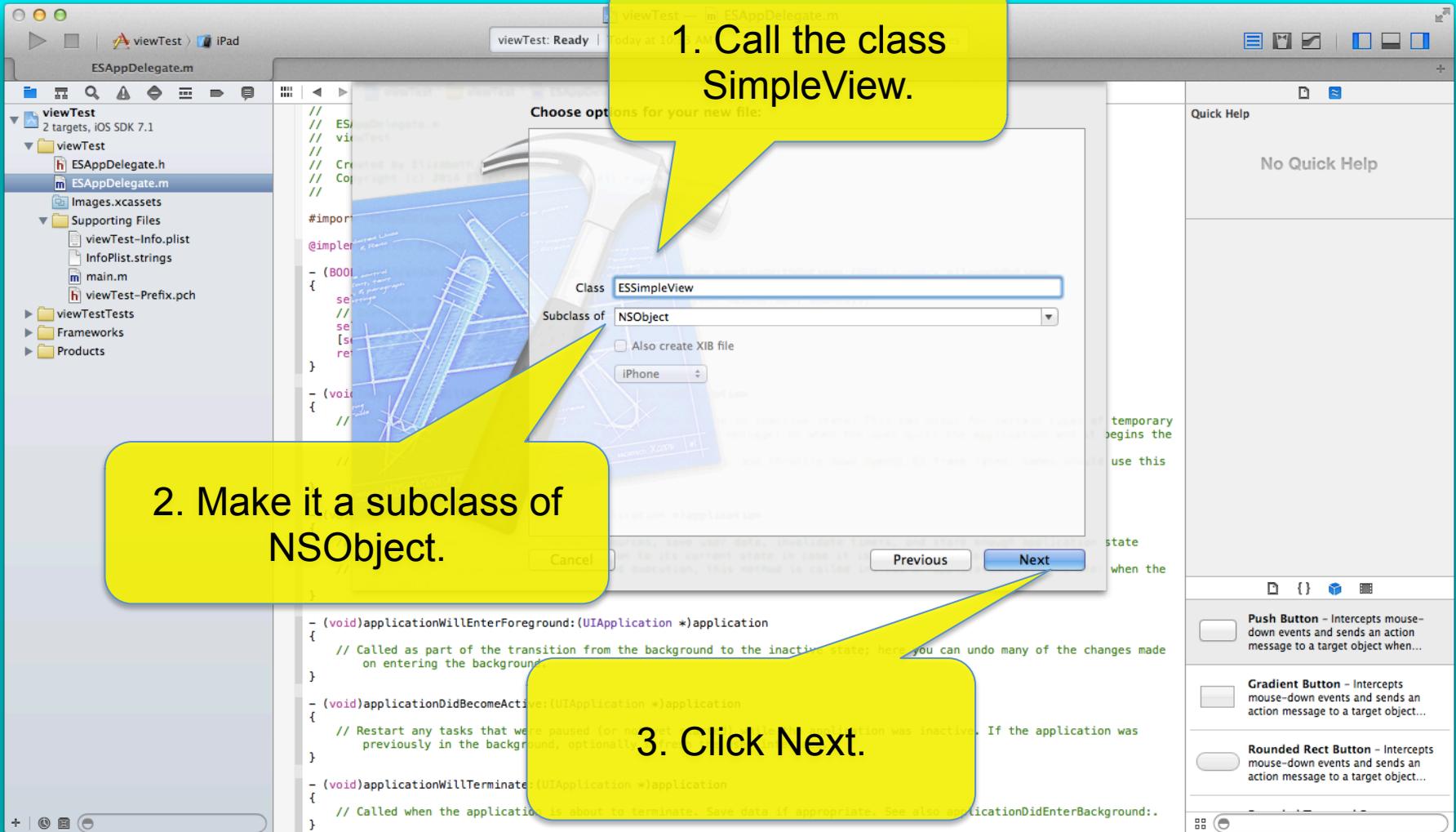
Next

3. Click Next.

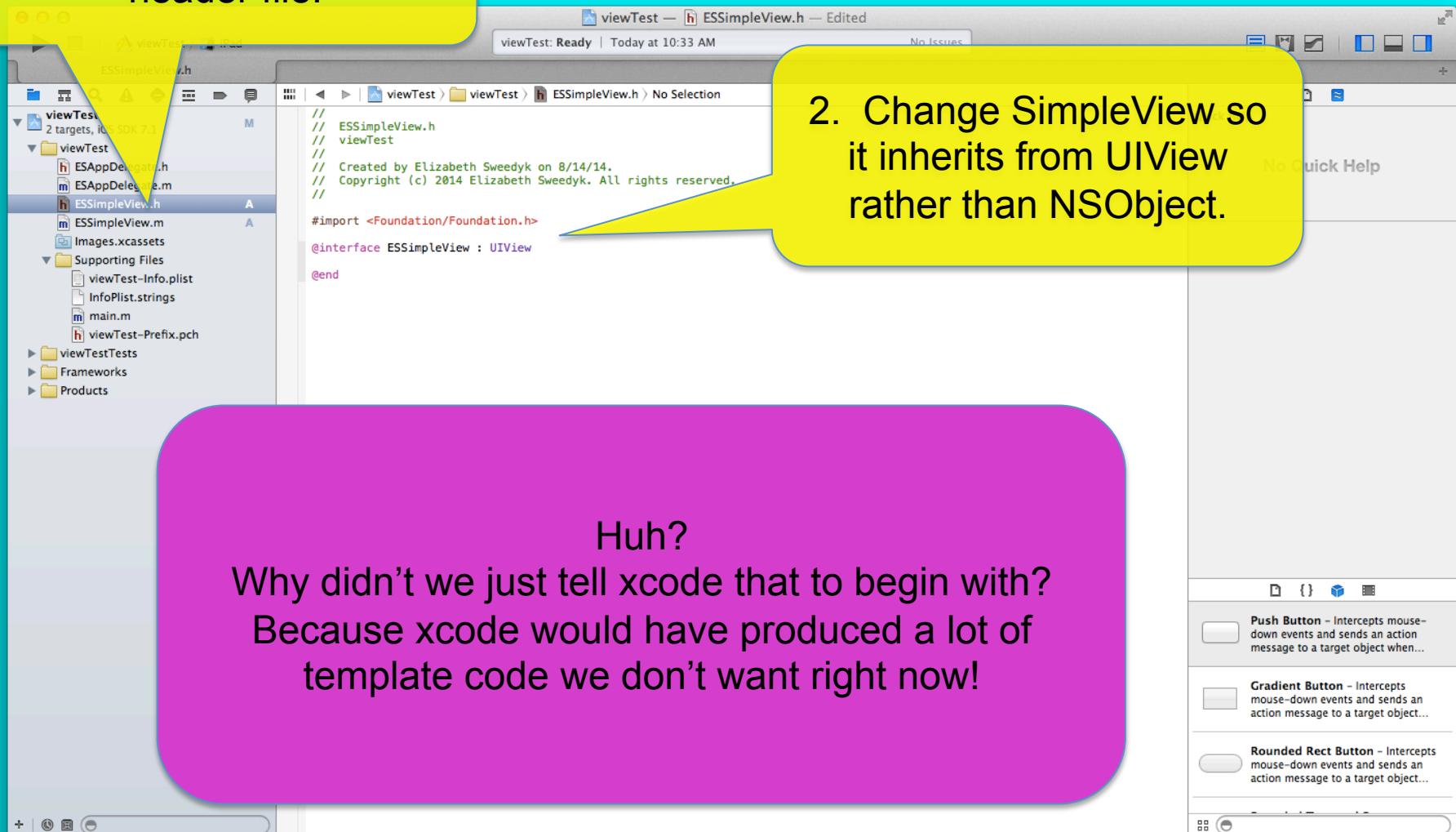
1. Call the class
SimpleView.

2. Make it a subclass of
NSObject.

3. Click Next.



1. Open the SimpleView header file.

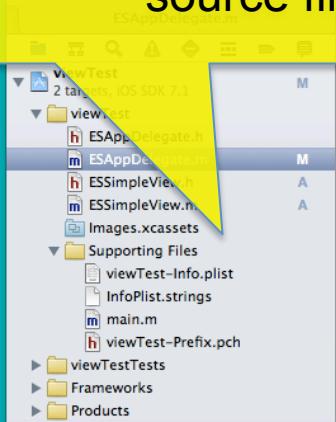


2. Change SimpleView so it inherits from UIView rather than NSObject.

Huh?

Why didn't we just tell xcode that to begin with?
Because xcode would have produced a lot of
template code we don't want right now!

1. Open the AppDelegate source file.



2. Import our class header.

Running viewTest on iPad

viewTest > ESAppDelegate.m - applicationDidFinishLaunchingWithOptions:

```
// ESSimpleView.h
#import "ESSimpleView.h"
#import "ESAppDelegate.h"

@implementation ESAppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
    // Override point for customization after application launch.

    // create view
    CGRect frame1 = CGRectMake(50,50,100,100);
    ESSimpleView *view1 = [[ESSimpleView alloc] initWithFrame:frame1];
    view1.backgroundColor = [UIColor whiteColor];
    [self.window addSubview:view1];

    self.window.backgroundColor = [UIColor redColor];
    [self.window makeKeyAndVisible];
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary
    // interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the
    // transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this
    // method to pause the game.
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate timers, and store enough application state
    // information to restore your application to its current state in case it is terminated later.
    // If your application supports background execution, this method is called instead of applicationWillTerminate: when the
    // user quits.
}
```

3. Insert this code.

Quick Help

No Quick Help

Search Documentation

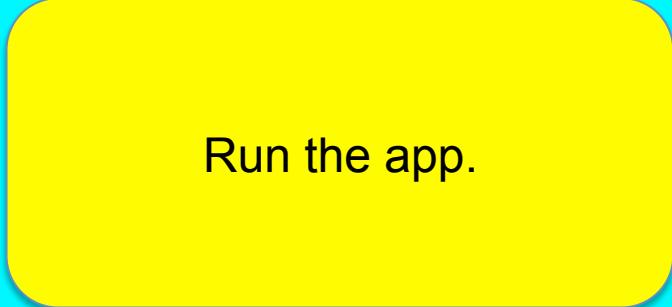
Push Button – Intercepts mouse-

down events and sends an action
message to a target object when...

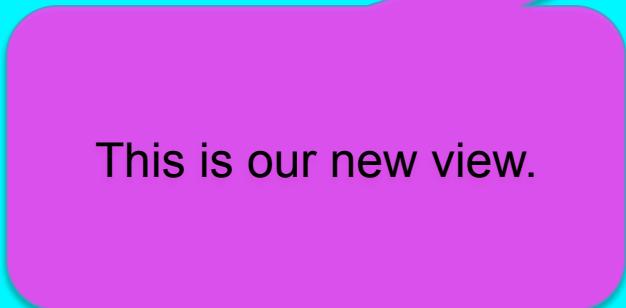
Gradient Button – Intercepts

mouse-down events and sends an
action message to a target object...

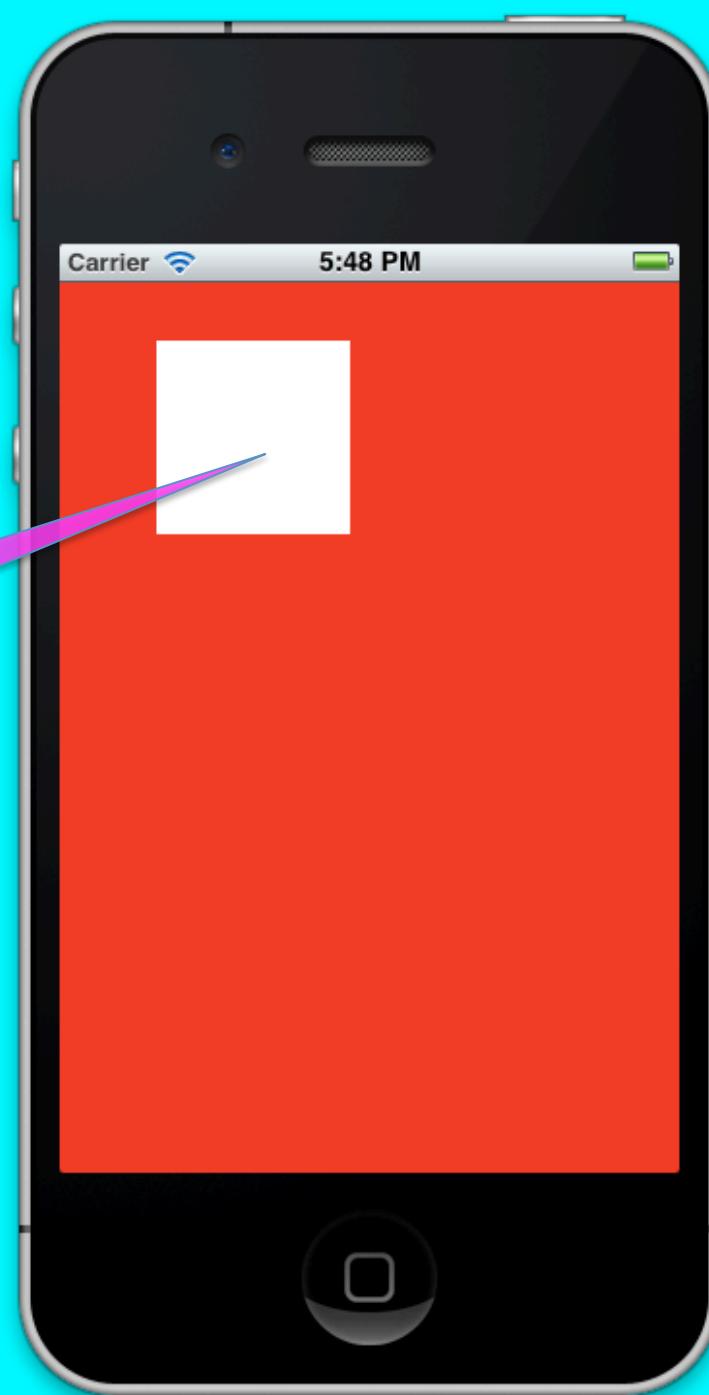
Rounded Rect Button – Intercepts
mouse-down events and sends an
action message to a target object...



Run the app.



This is our new view.



A view is like a subwindow. The key steps to creating a view are to:

1. Define its “frame” – the region it will occupy.
2. Add it as a subview.

It was not critical to set its background color but otherwise it would have been the same color as the window and we wouldn't have been able to see it.

The screenshot shows the Xcode interface with the ESAppDelegate.m file open in the editor. A callout bubble points from the text "It was not critical to set its background color but otherwise it would have been the same color as the window and we wouldn't have been able to see it." to the line of code where the view's background color is set to white.

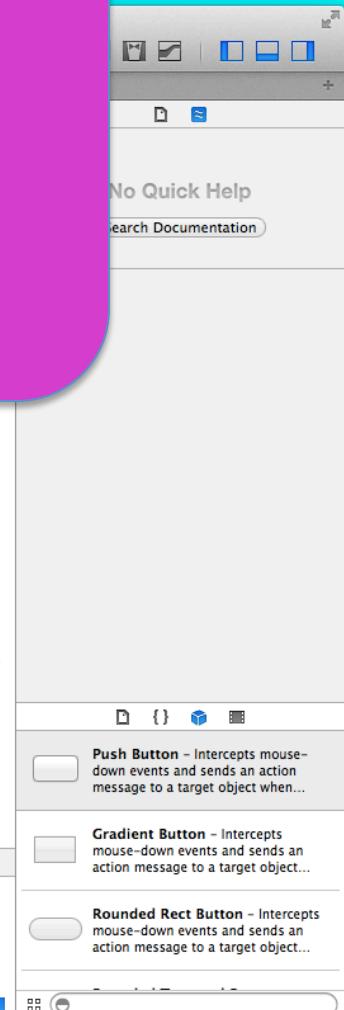
```
// Override point for customization after application launch

// create view
CGRect frame1 = CGRectMake(50,50,100,100);
ESSimpleView* view1 = [[ESSimpleView alloc] initWithFrame:frame1];
view1.backgroundColor = [UIColor whiteColor];
[self.window addSubview:view1];

self.window.backgroundColor = [UIColor redColor];
[self.window makeKeyAndVisible];
return YES;
}

-(void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this method to pause the game.
}

-(void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.
    // If your application supports background execution, this method is called instead of applicationWillTerminate: when the
    // user quits.
}
```



We can have multiple views!

The screenshot shows the Xcode interface with the file `ESAppDelegate.m` open. The code implements the `UIApplicationDelegate` protocol to handle application launch and window creation. A yellow callout bubble points to the closing brace of the `application:didFinishLaunchingWithOptions:` method, containing the text "Insert this code.".

```
#import "ESSimpleView.h"
#import "ESAppDelegate.h"

@implementation ESAppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    // Override point for customization after application launch.

    // create view
    CGRect frame1 = CGRectMake(50,50,100,100);
    ESSimpleView* view1 = [[ESSimpleView alloc] initWithFrame:frame1];
    view1.backgroundColor = [UIColor whiteColor];
    [self.window addSubview:view1];

    // create another view
    CGRect frame2 = CGRectMake(20,20,50,50);
    ESSimpleView* view2 = [[ESSimpleView alloc] initWithFrame:frame2];
    view2.backgroundColor = [UIColor blueColor];
    [self.window addSubview:view2];

    self.window.backgroundColor = [UIColor redColor];
    [self.window makeKeyAndVisible];
    return YES;
}

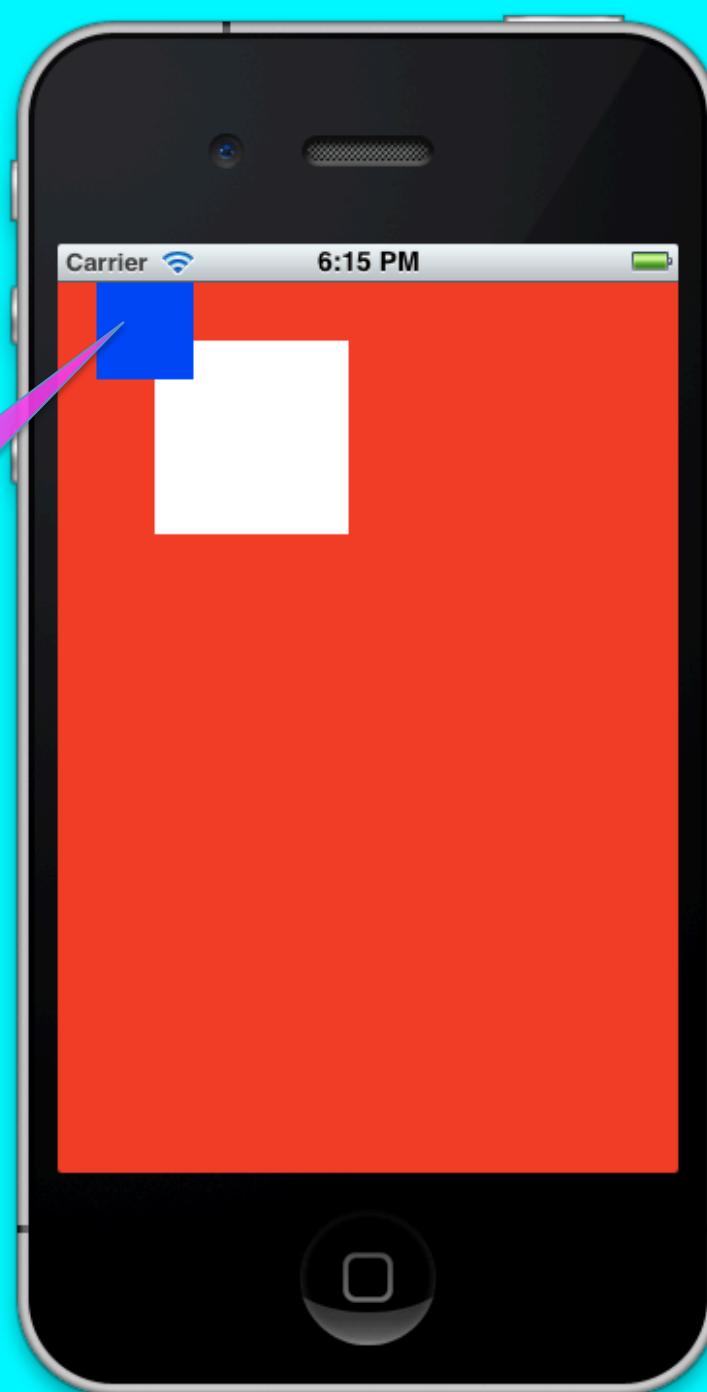
- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary
    // interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the
    // transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this
    // method to pause the game.
}
```

The Xcode interface includes a sidebar with project files like `ESSimpleView.m`, `viewTestTests`, and `Products`. The bottom status bar shows the date and time as 2014-08-14 19:46:50.055, and the output pane displays the message: "2014-08-14 19:46:50.055 viewTest[728:90b] Application windows are expected to have a root view controller at the end of application launch".

Insert this code.

Run the app.

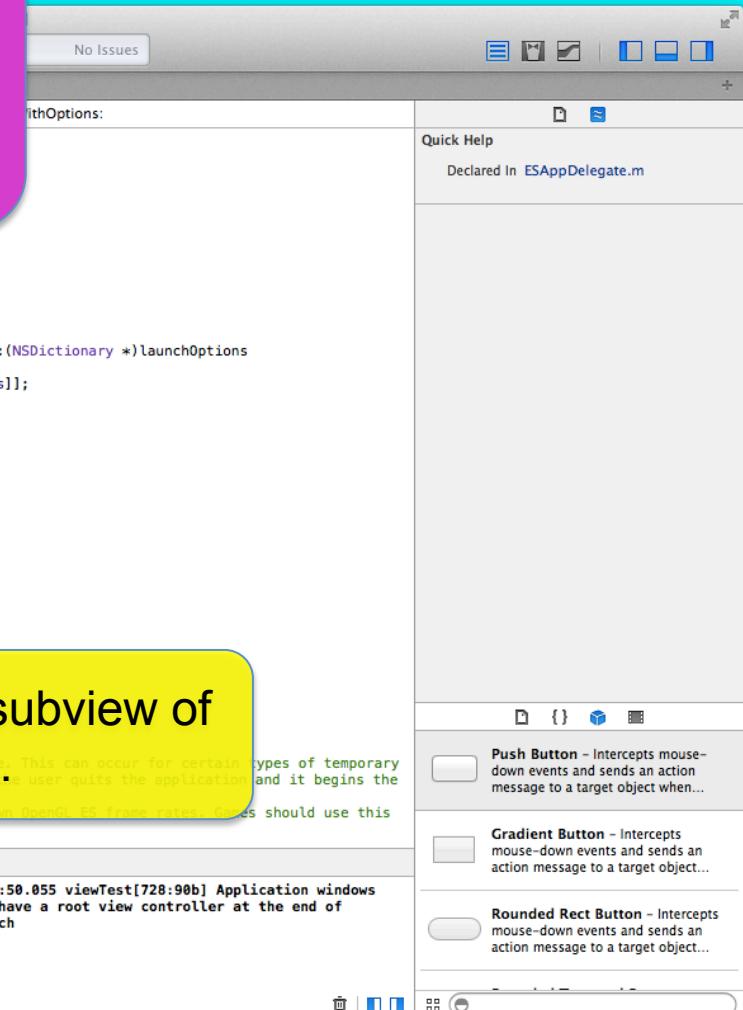
This is our new view.
Views are drawn in the
order they are added as
subviews, so the blue
view is drawn on top of
the red one.
(There is a way to
reorder the views.)



A view can have subviews.

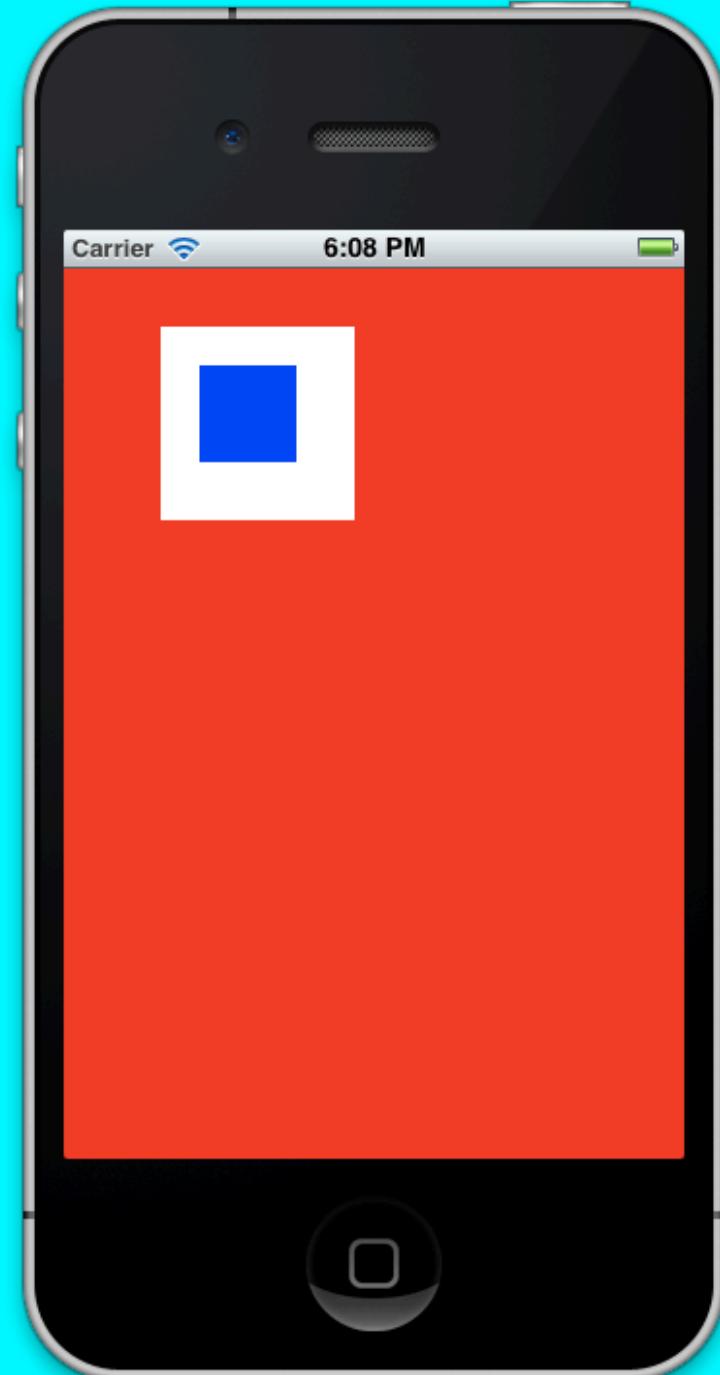
```
//  
#import "ESSimpleView.h"  
#import "ESAppDelegate.h"  
  
@implementation ESAppDelegate  
  
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];  
    // Override point for customization after application launch.  
  
    // create view  
    CGRect frame1 = CGRectMake(50,50,100,100);  
    ESSimpleView* view1 = [[ESSimpleView alloc] initWithFrame:frame1];  
    view1.backgroundColor = [UIColor whiteColor];  
    [self.window addSubview:view1];  
  
    // create another view  
    CGRect frame2 = CGRectMake(20,20,50,50);  
    ESSimpleView* view2 = [[ESSimpleView alloc] initWithFrame:frame2];  
    view2.backgroundColor = [UIColor blueColor];  
    [view1 addSubview:view2];  
  
    self.window.backgroundColor = [UIColor redColor];  
    [self.window makeKeyAndVisible];  
    return YES;  
}  
  
- (void)applicationWillResignActive:(UIApplication *)application  
{  
    // Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary  
    // interruptions (such as an incoming phone call or SMS message) or because the user quit the application and it begins the  
    // transition to the background state.  
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this  
    // method to pause the game.  
}
```

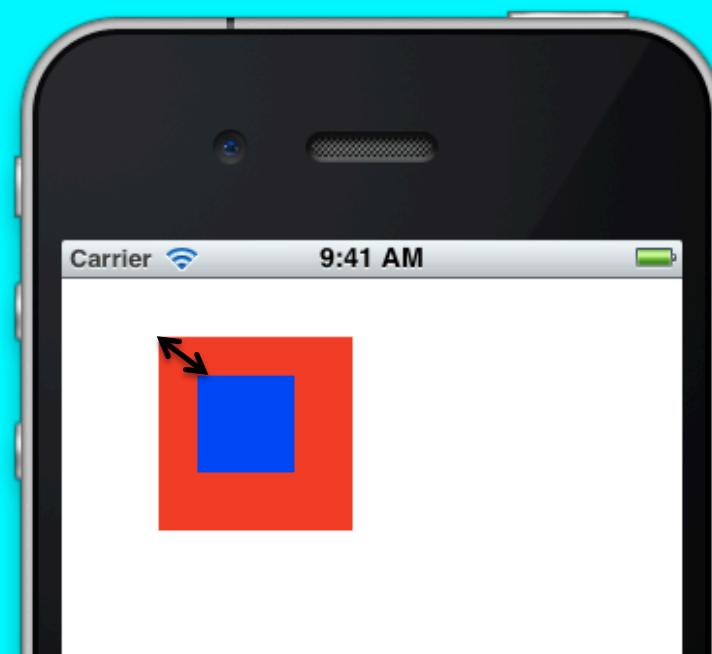
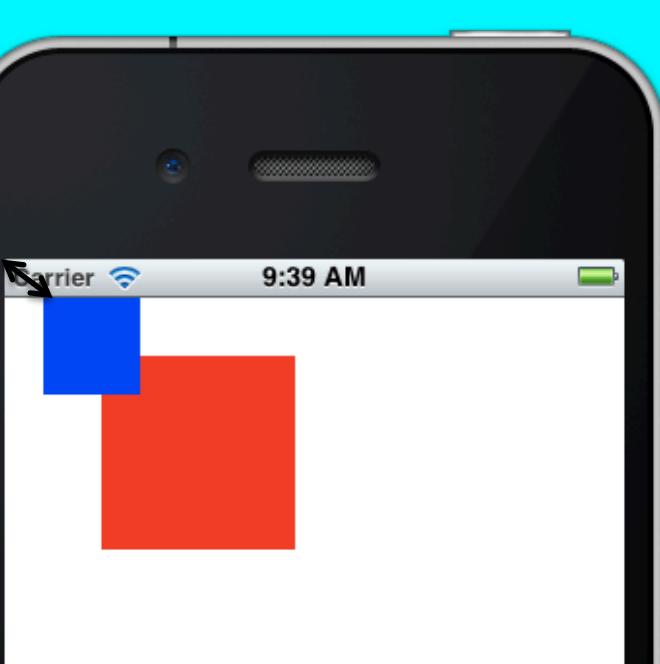
Make view2 a subview of view1.



Run the app.

Is this what you expected?



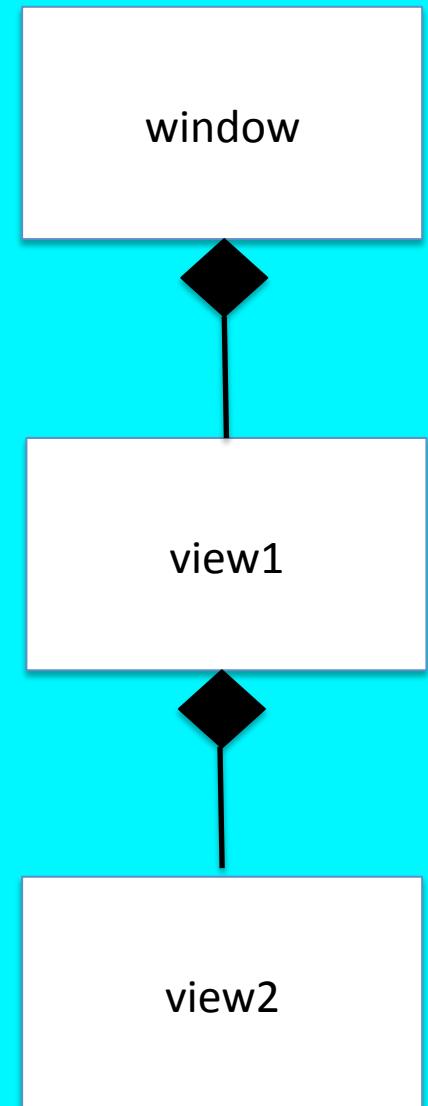
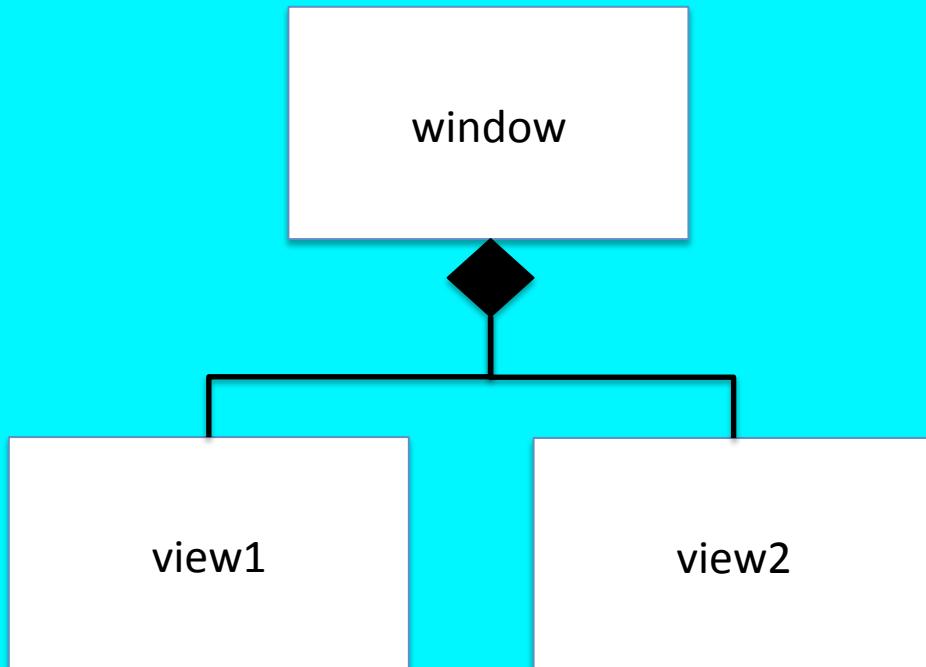


A view's frame is defined relative to its parent's frame.

On the left, the blue view is defined relative the window's origin, which is the top left of the screen. (The origin is actually occluded by the status bar. Depending on your setting the status bar may also be clear.)

On the right, the blue view is defined relative to the red view's origin, which is at 50,50 in the window's frame.

Here are diagrams of the two approaches!



We are being warned our app should have a root view controller. More about that later.

This button opens a console window. Open the window and you'll see a warning message.

```
// ESAppDelegate.m
// viewTest
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.

#import "ESSimpleView.h"
#import "ESAppDelegate.h"

@implementation ESAppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
    // Override point for customization after application launch.

    // create view
    CGRect frame1 = CGRectMake(50,50,100,100);
    ESSimpleView *view1 = [[ESSimpleView alloc] initWithFrame:frame1];
    view1.backgroundColor = [UIColor whiteColor];
    [self.window addSubview:view1];

    // create another view
    CGRect frame2 = CGRectMake(28,28,50,50);
    ESSimpleView *view2 = [[ESSimpleView alloc] initWithFrame:frame2];
    view2.backgroundColor = [UIColor blueColor];
    [view1 addSubview:view2];
}

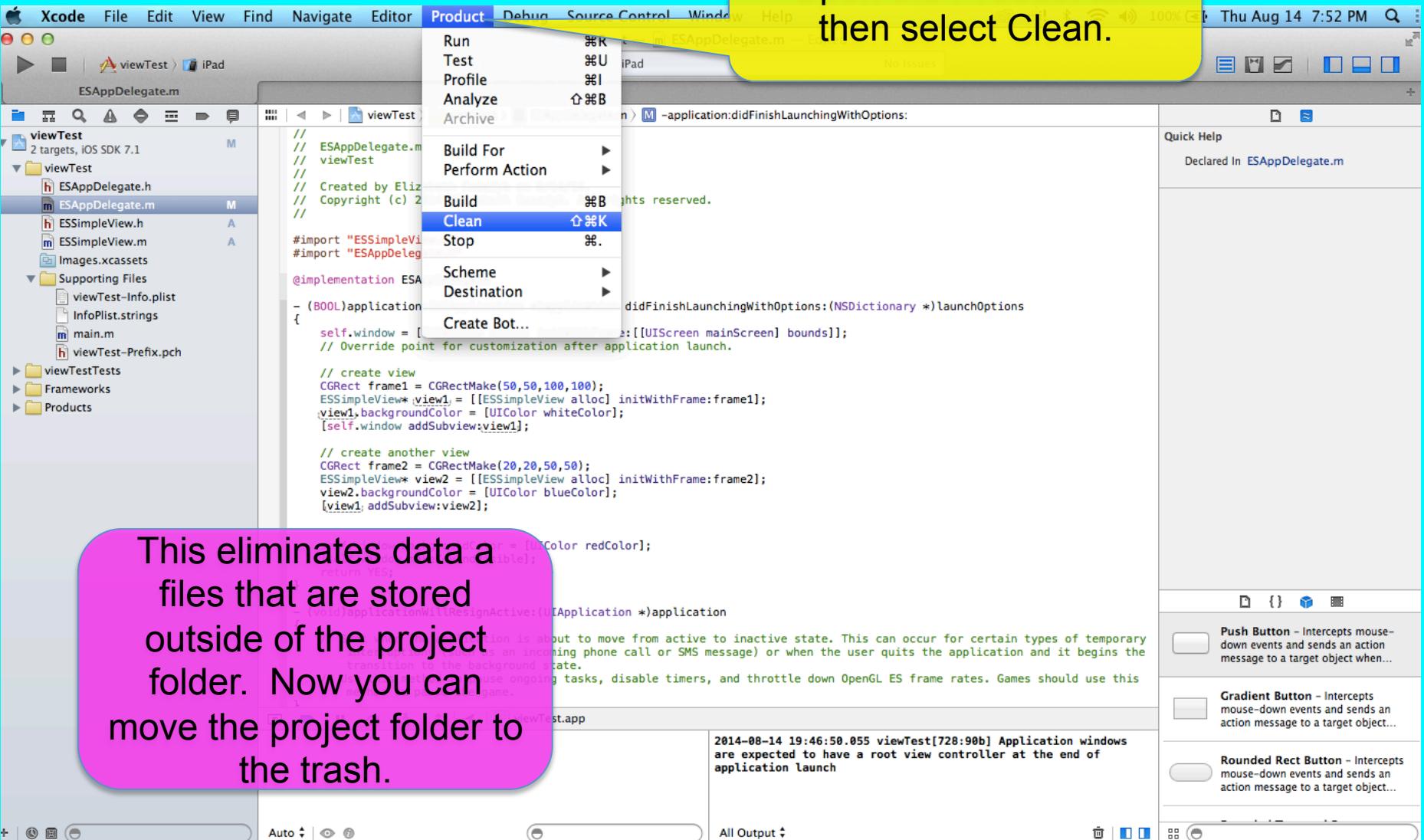
- (void)applicationWillResignActive:(UIApplication *)application
{
    // Called when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this method to pause the game.
}
```

2014-08-14 19:46:50.055 viewTest[728:90b] Application windows are expected to have a root view controller at the end of application launch

What we did was an experiment. Until you become an iOS expert, you would be wise to use the template code.

- Push the viewTest to github
- Xcode project use a lot of memory and not all of is freed when you simply move the project to trash. The next slide shows you the process for properly disposing of an Xcode project.

Open the Product tab
then select Clean.



1. Create a new single view iPad app named viewTutorial. Run the app. The screen should be white.

The screenshot shows the Xcode interface with the project 'viewTutorial' selected. The left sidebar shows files like ESAppDelegate.h, ESAppDelegate.m, Main.storyboard, and ESViewController.h. The main editor area displays the contents of ESAppDelegate.m. A yellow callout points to the file list on the left, and a pink callout points to the code in the editor. The right side of the interface includes a 'Quick Help' panel with documentation for View Controller, Collection View Controller, and Table View Controller.

```
// ESAppDelegate.m
// viewTutorial
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//

#import "ESAppDelegate.h"

@implementation ESAppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this method to pause the game.
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.
    // If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state: this is intended to be used to "undo" changes made in applicationWillResignActive:.
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.
}

@end
```

2. Open the AppDelegate source file.

Unlike the ViewTest app, this one wasn't created with code that sets the window to white!

These three files were created for us. Look back at the earlier slides and notice that they were not created in our ViewTest empty project.

```
// ESAppDelegate.m
// viewTutorial
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//

#import "ESAppDelegate.h"

@implementation ESAppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from the active to inactive state, which can occur for temporary interruptions (such as an incoming phone call or SMS message). In the inactive state, the application still running, but active.
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this method to pause the game.
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.
    // If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.
}

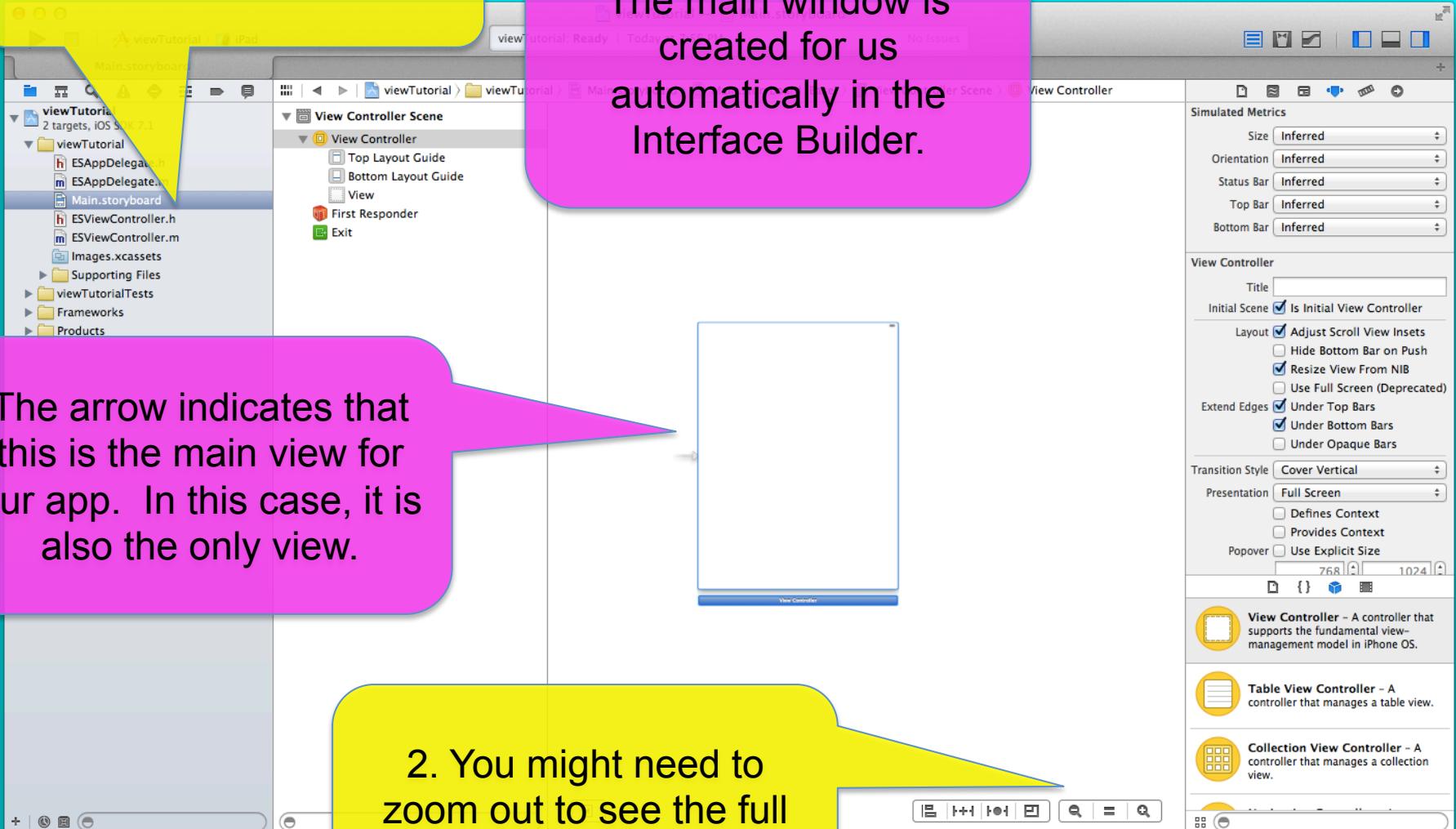
- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo many of the changes made on entering the background.
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.
}

@end
```

1. Open the storyboard.



The main window is
created for us
automatically in the
Interface Builder.

The arrow indicates that
this is the main view for
our app. In this case, it is
also the only view.

2. You might need to
zoom out to see the full
storyboard.

1. These buttons open and close the left and right panels. Try it!.

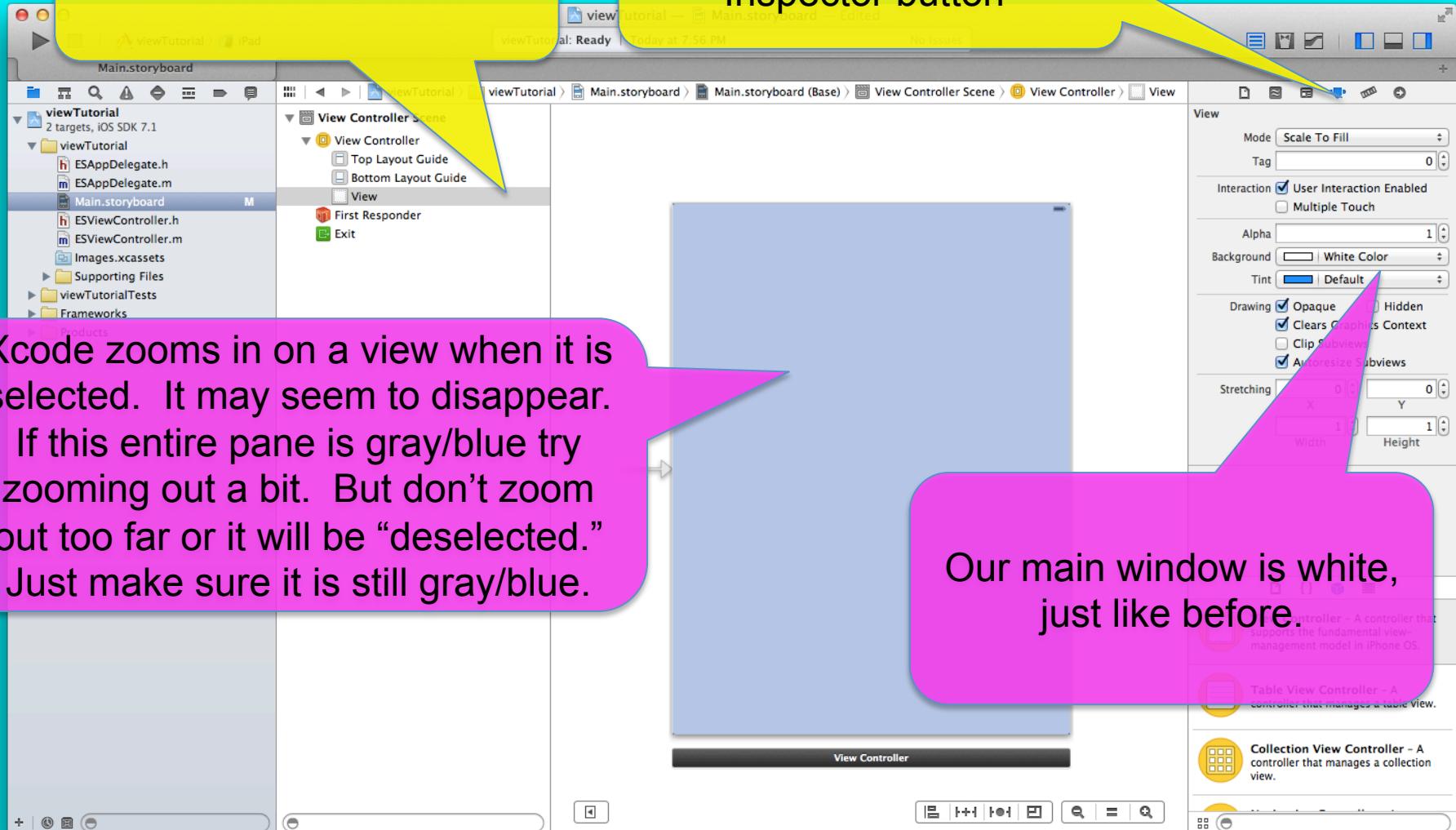
2. Click on the “Identity Inspector button”

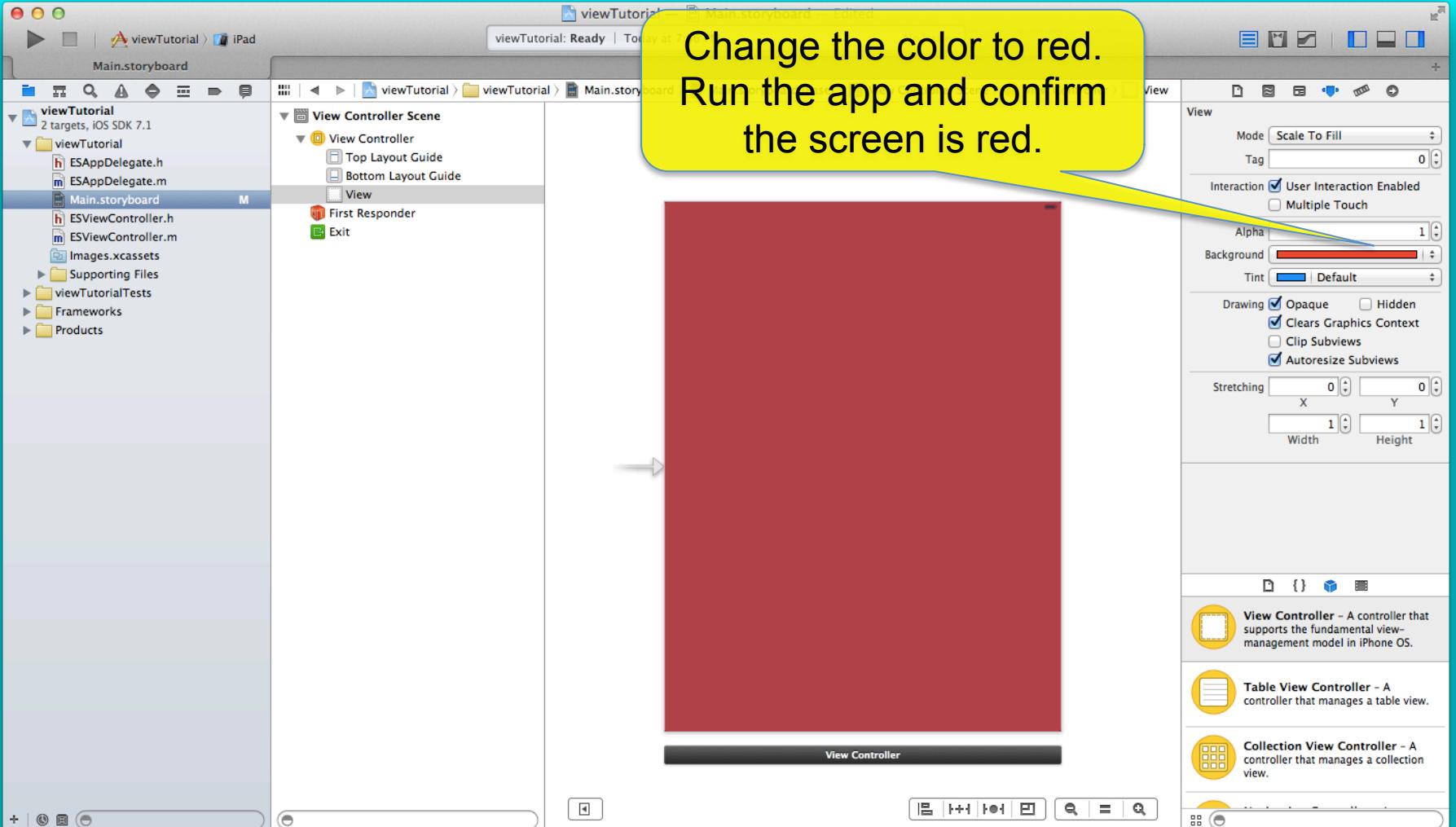
Our main window has a view controller called ESViewController. This is the root view controller. (Remember the warning we got in the ViewTest project. We won't get it with this project!)

The files for the ESViewController class are here.

1. Select the view.

2. Click on the “Attributes Inspector button”





1. Open the view controller source file.

The screenshot shows the Xcode interface with the project navigation bar at the top. Below it is the file browser showing the project structure: viewTutorial (target), viewTutorial (group), ESAppDelegate.h, ESAppDelegate.m, Main.storyboard, ESViewController.h, and ESViewController.m. The ESViewController.m file is selected and open in the main editor area. The code in the editor is:

```
// ESViewController.m
// viewTutorial
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.

#import "ESViewController.h"

@interface ESViewController : UIViewController

@end

@implementation ESViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    self.view.backgroundColor = [UIColor whiteColor];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

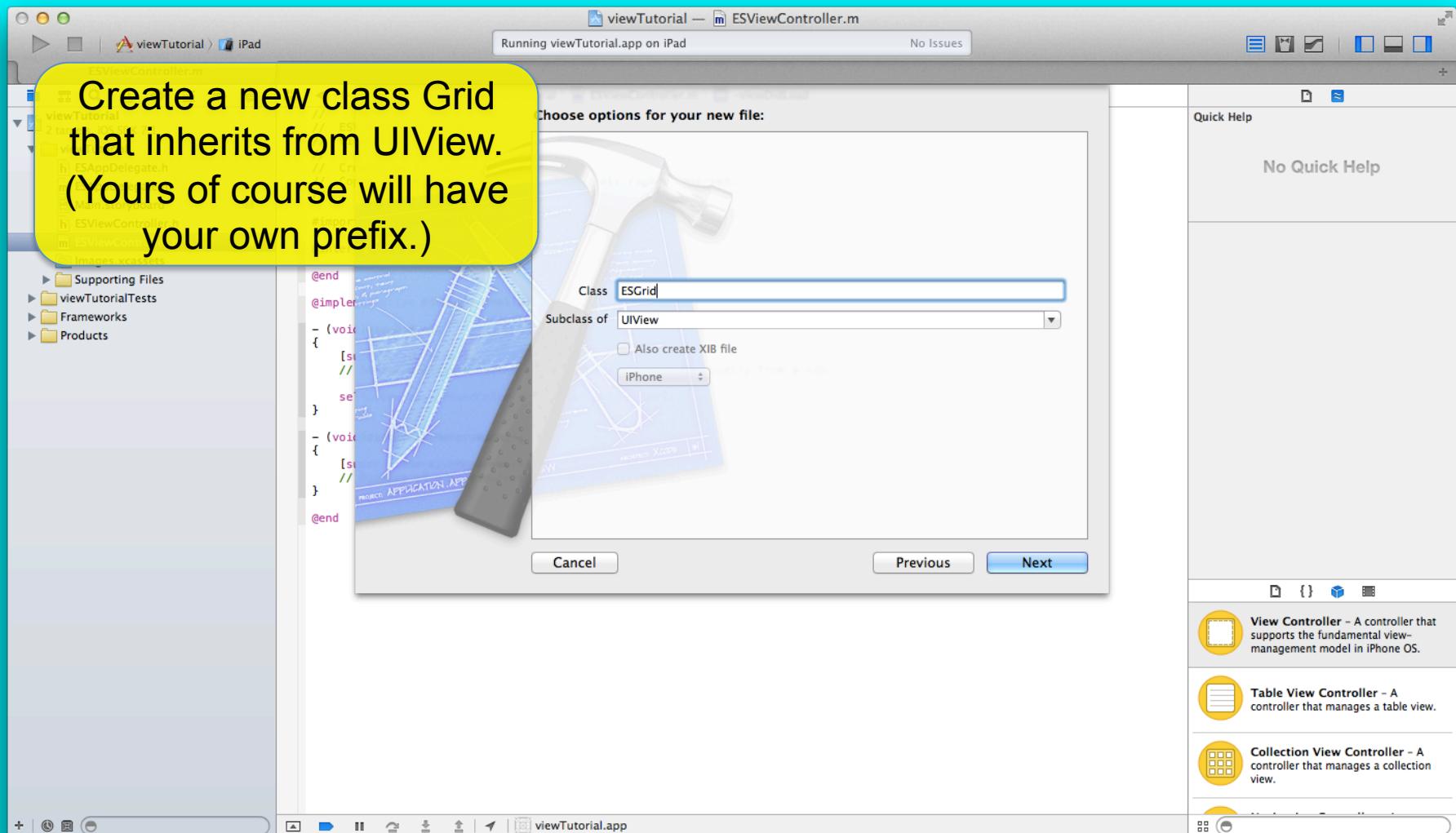
@end
```

A yellow callout bubble points to the ESViewController.m file in the file browser with the text "1. Open the view controller source file.". A blue callout bubble points to the line `self.view.backgroundColor = [UIColor whiteColor];` in the code with the text "2. Add this line of code. Run the app and confirm that the window is white again.". A pink callout bubble contains the explanatory text about Interface Builder.

The Xcode interface also includes a status bar at the bottom with icons for file, edit, and search.

The Interface Builder (IB) makes it very easy to create UI elements. But anything you can do through the IB, you can do in code as well. And what you do in code typically overrides what you do in the IB.

-  View Controller – A controller that supports the fundamental view-management model in iOS OS.
-  Table View Controller – A controller that manages a table view.
-  Collection View Controller – A controller that manages a collection view.



1. Open the
ViewController source
file.

2. Import Grid.h.

```
// ESViewController.m
// viewTutorial
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//

#import "ESGrid.h"
#import "ESViewController.h"

@interface ESViewController () {
    UIView* _gridView;
}

@end

@implementation ESViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    self.view.backgroundColor = [UIColor whiteColor];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

3. Create an instance variable to store our view.

The screenshot shows the Xcode interface with the project 'viewTutorial' selected. The file 'ESViewController.m' is open in the editor. The code implements a UIViewController with a gridView subview. A yellow callout points to the line where '_gridView' is created and added to the main view.

```
// ESViewController.m
// viewTutorial
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//

#import "ESGrid.h"
#import "ESViewController.h"

@interface ESViewController () {
    UIView* _gridView;
}

@end

@implementation ESViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    self.view.backgroundColor = [UIColor whiteColor];

    // create grid frame
    CGRect frame = self.view.frame;
    CGFloat x = CGRectGetWidth(frame)*.1;
    CGFloat y = CGRectGetHeight(frame)*.1;
    CGFloat size = MIN(CGRectGetWidth(frame), CGRectGetHeight(frame))*.80;

    CGRect gridFrame = CGRectMake(x, y, size, size);

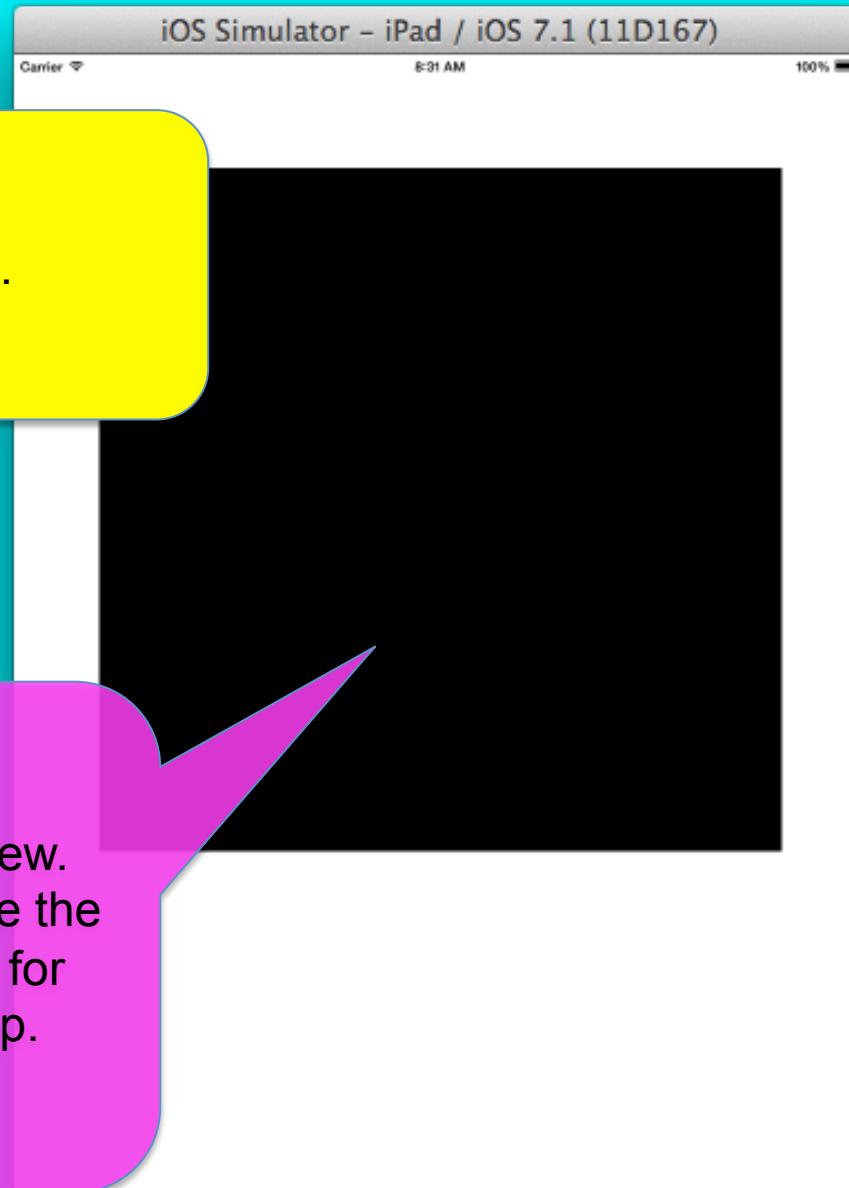
    // create grid view
    _gridView = [[ESGrid alloc] initWithFrame:gridFrame];
    _gridView.backgroundColor = [UIColor blackColor];
    [self.view addSubview:_gridView];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}
```

Add these line to
create `_gridView` and
add it as a subview.

In creating the frame for our grid, we size it relative to the main view. This allows our implementation to work for iPhone and iPad, whether in portrait or landscape mode.

	{}	Push Button – Intercepts mouse-down events and sends an action message to a target object when it's...
	{}	Gradient Button – Intercepts mouse-down events and sends an action message to a target object...
	{}	Rounded Rect Button – Intercepts mouse-down events and sends an action message to a target object...



viewTutorial — ESViewController.m — Edited

Running viewTutorial on iPad No Issues

viewTutorial — ESViewController.m — Edited

viewTutorial > viewTutorial > ESViewController.m > @interface ESViewController()

```
// ESViewController.m
// viewTutorial
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//

#import "ESGrid.h"
#import "ESViewController.h"

@interface ESViewController () {
    UIButton* _button;
    UIView* _gridView;
}

@end

@implementation ESViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    self.view.backgroundColor = [UIColor whiteColor];

    // create grid frame
    CGRect frame = self.view.frame;
    CGFloat x = CGRectGetWidth(frame)*.1;
    CGFloat y = CGRectGetHeight(frame)*.1;
    CGFloat size = MIN(CGRectGetWidth(frame), CGRectGetHeight(frame))*80;

    CGRect gridFrame = CGRectMake(x, y, size, size);

    // create grid view
    _gridView = [[ESGrid alloc] initWithFrame:gridFrame];
    _gridView.backgroundColor = [UIColor blackColor];
    [self.view addSubview:_gridView];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

Quick Help
No Quick Help

Push Button – Intercepts mouse-down events and sends an action message to a target object when it's...

Gradient Button – Intercepts mouse-down events and sends an action message to a target object...

Rounded Rect Button – Intercepts mouse-down events and sends an action message to a target object...

Let's
programmatically
add a button to our
grid. First create the
instance variable.

UIButton is a subclass of UIView. So we need to give it a frame. We can also set its background color.

The screenshot shows the Xcode interface with the file `ESViewController.m` open in the editor. The code is as follows:

```
// ESViewController.m
// viewTutorial
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//

#import "ESGrid.h"
#import "ESViewController.h"

@interface ESViewController () {
    UIButton* _button;
    UIVIew* _gridView;
}

@end

@implementation ESViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    self.view.backgroundColor = [UIColor whiteColor];

    // create grid frame
    CGRect frame = self.view.frame;
    CGFloat x = CGRectGetWidth(frame)*.1;
    CGFloat y = CGRectGetHeight(frame)*.1;
    CGFloat size = MIN(CGRectGetWidth(frame), CGRectGetHeight(frame))*80;

    CGRect gridFrame = CGRectMake(x, y, size, size);

    // create grid view
    _gridView = [[ESGrid alloc] initWithFrame:gridFrame];
    _gridView.backgroundColor = [UIColor blackColor];
    [self.view addSubview:_gridView];

    // create button
    CGFloat buttonSize = size/5.0;
    CGRect buttonFrame = CGRectMake(0, 0, buttonSize, buttonSize);
    _button = [[UIButton alloc] initWithFrame:buttonFrame];
    _button.backgroundColor = [UIColor redColor];
    [_gridView addSubview:_button];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
}
```

A pink callout bubble points to the first few lines of the code where the background color is set. A yellow callout bubble on the right side of the screen contains the text: "Add this code to create the button, set its color, and add it as a subview." A curly brace on the right side groups the code for creating the grid view and the button.

The Xcode interface includes the Project Navigator on the left, the Editor area with the code, the Utilities area on the right, and the bottom navigation bar.

iOS Simulator - iPad / iOS 7.1 (11D167)

Carrier

8:38 AM

100%



Run the app.

This is our button. It
doesn't do anything yet.
We have to configure it
appropriately.

We'll talk about "targets" in class.

1. Set the "target" for a button press.

The screenshot shows the Xcode interface with the file `ESViewController.m` open. The code is as follows:

```
-(void)viewDidLoad {
    // ...
    self.view.backgroundColor = [UIColor whiteColor];
    // create grid frame
    CGRect frame = self.view.frame;
    CGFloat x = CGRectGetWidth(frame)*.1;
    CGFloat y = CGRectGetHeight(frame)*.1;
    CGFloat size = MIN(CGRectGetWidth(frame), CGRectGetHeight(frame))*.80;
    CGRect gridFrame = CGRectMake(x, y, size, size);
    // create grid view
    ESGridView *_gridView = [[ESGrid alloc] initWithFrame:gridFrame];
    _gridView.backgroundColor = [UIColor blackColor];
    self.view addSubview:_gridView;
    // create button
    CGFloat buttonSize = size/5.0;
    CGRect buttonFrame = CGRectMake(0,0,buttonSize,buttonSize);
    _button = [[UIButton alloc] initWithFrame:buttonFrame];
    _button.backgroundColor = [UIColor redColor];
    [_gridView addSubview:_button];
    // create target for button
    [_button addTarget:self action:@selector(buttonPressed:) forControlEvents:UIControlEventTouchUpInside];
}
- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}
- (void)buttonPressed:(id)sender
{
    NSLog(@"You pressed the button!");
}
@end
```

2. Create the buttonPressed method.

3. Run the app and press the button.

A message appears in the console window.

Now you should configure the button programmatically so that its label says “Hit me” in blue. Also, when the button is pressed it should highlight momentarily.

You will probably find it useful to refer to Apple’s documentation for the `UIButton` class!

Push your ViewTutorial to github