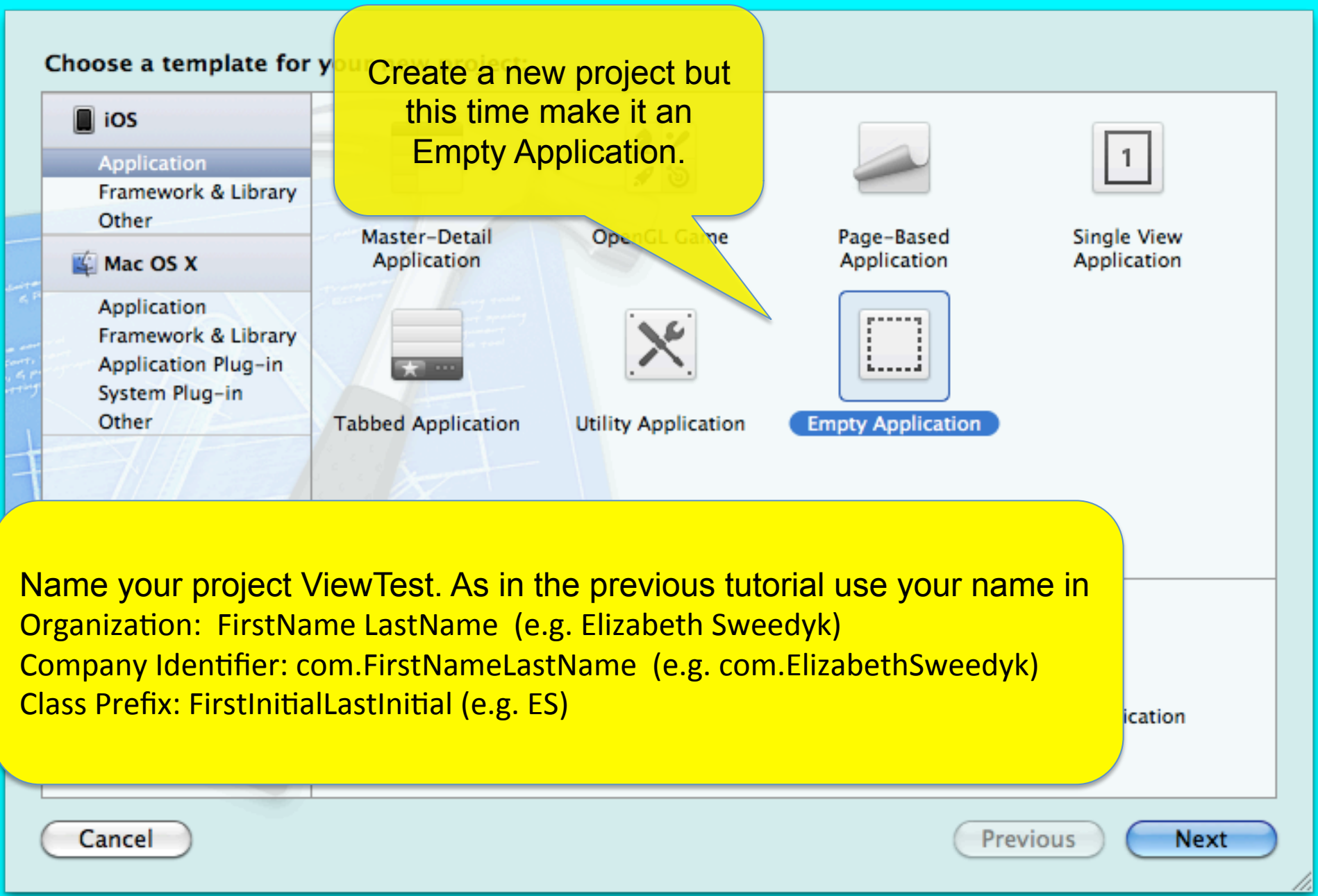
CS121 Views tutorial

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.

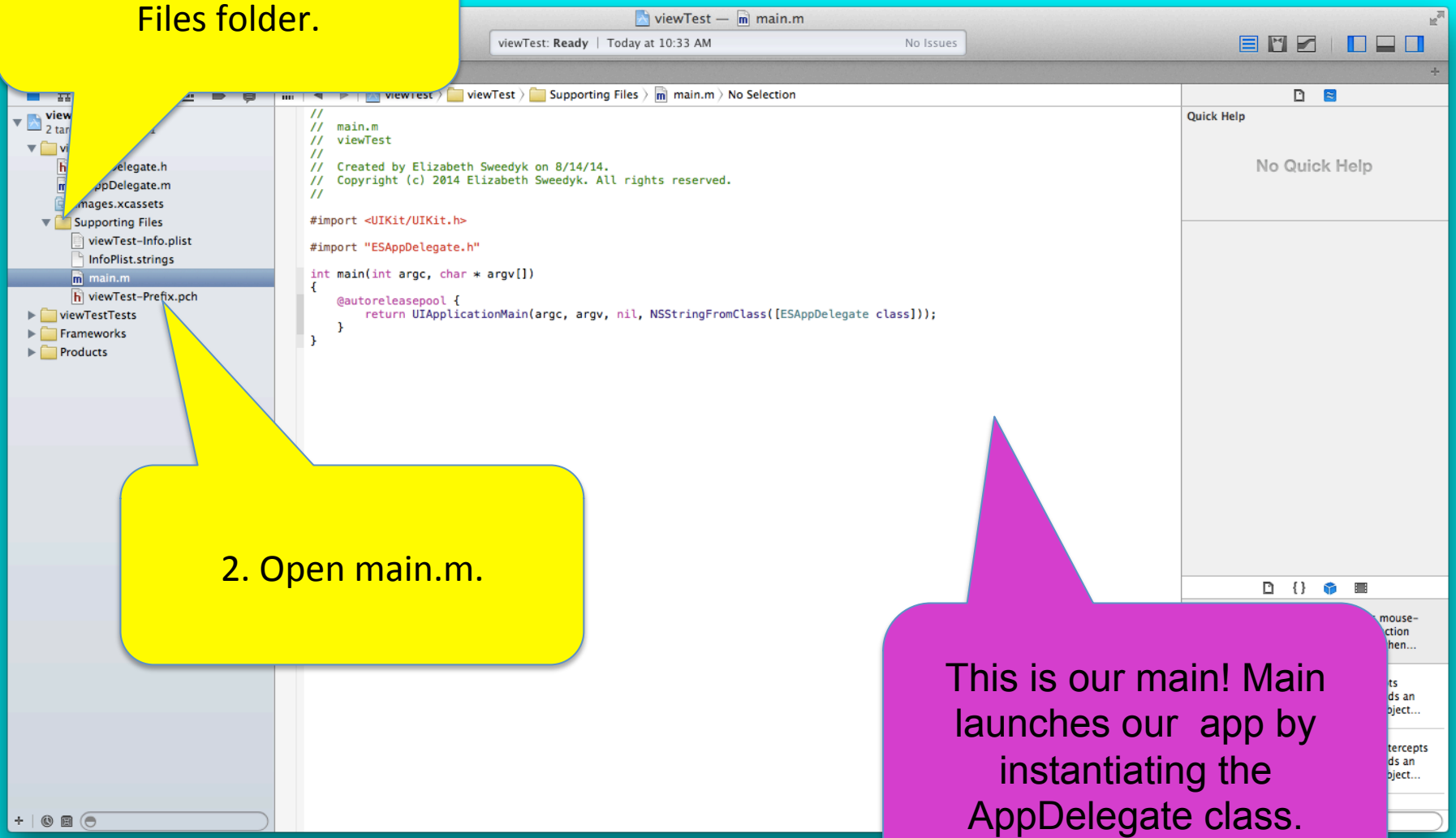
Orange bubbles tell you what you're not expected to understand yet. ☺



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

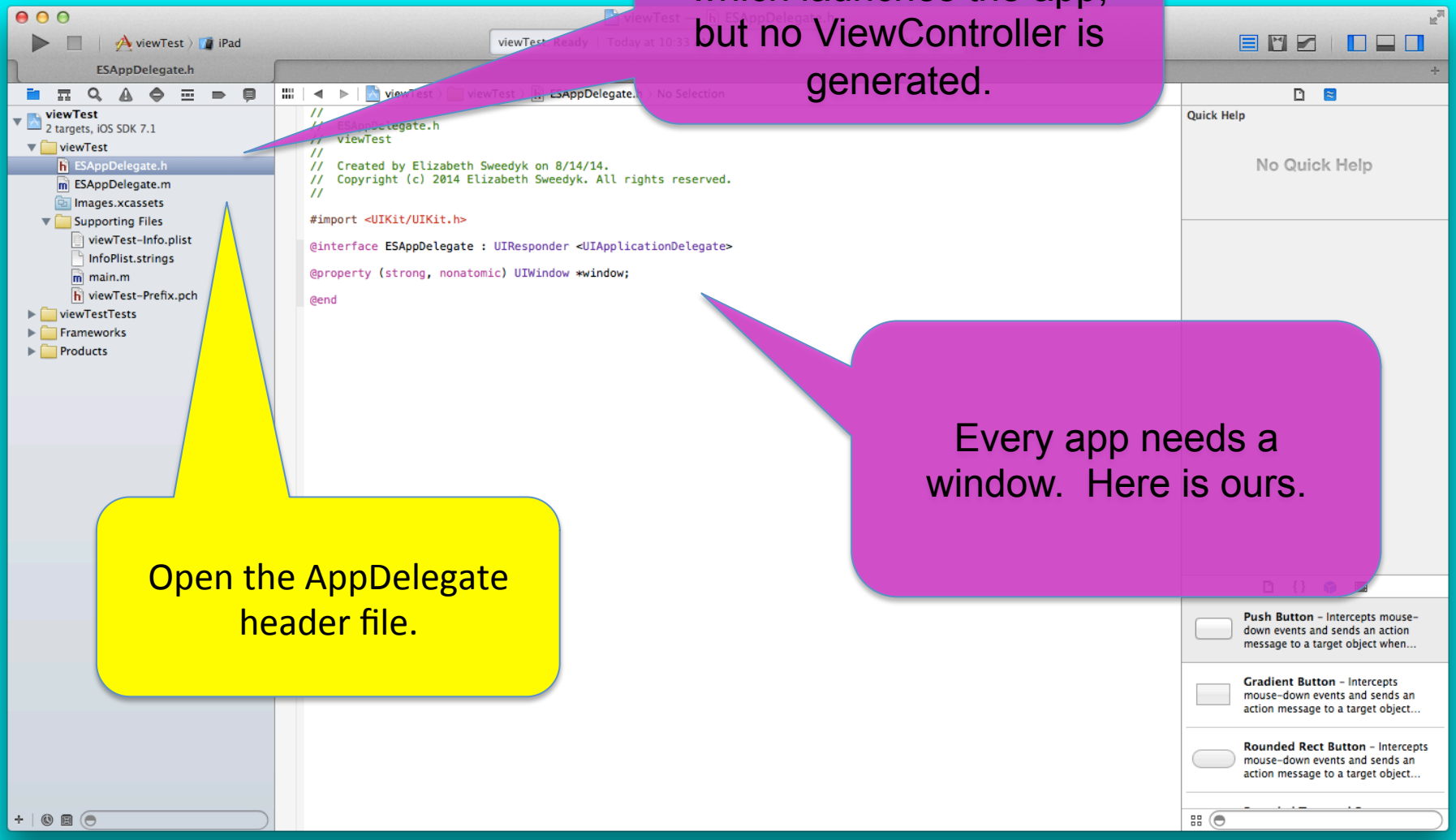
Name your project ViewTest. As in the previous tutorial use your name in
Organization: FirstName LastName (e.g. Elizabeth Sweedyk)
Company Identifier: com.FirstNameLastName (e.g. com.ElizabethSweedyk)
Class Prefix: FirstInitialLastInitial (e.g. ES)

1. Open the Supporting Files folder.

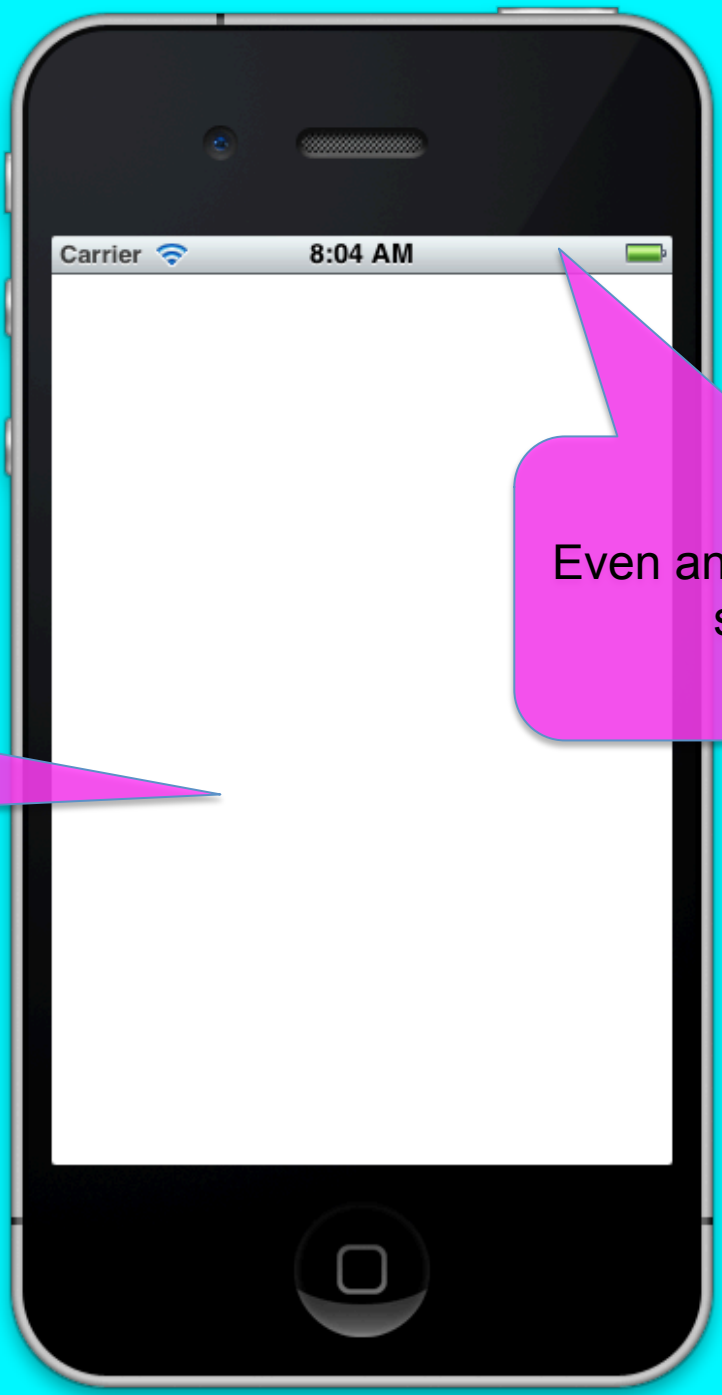


2. Open main.m.

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

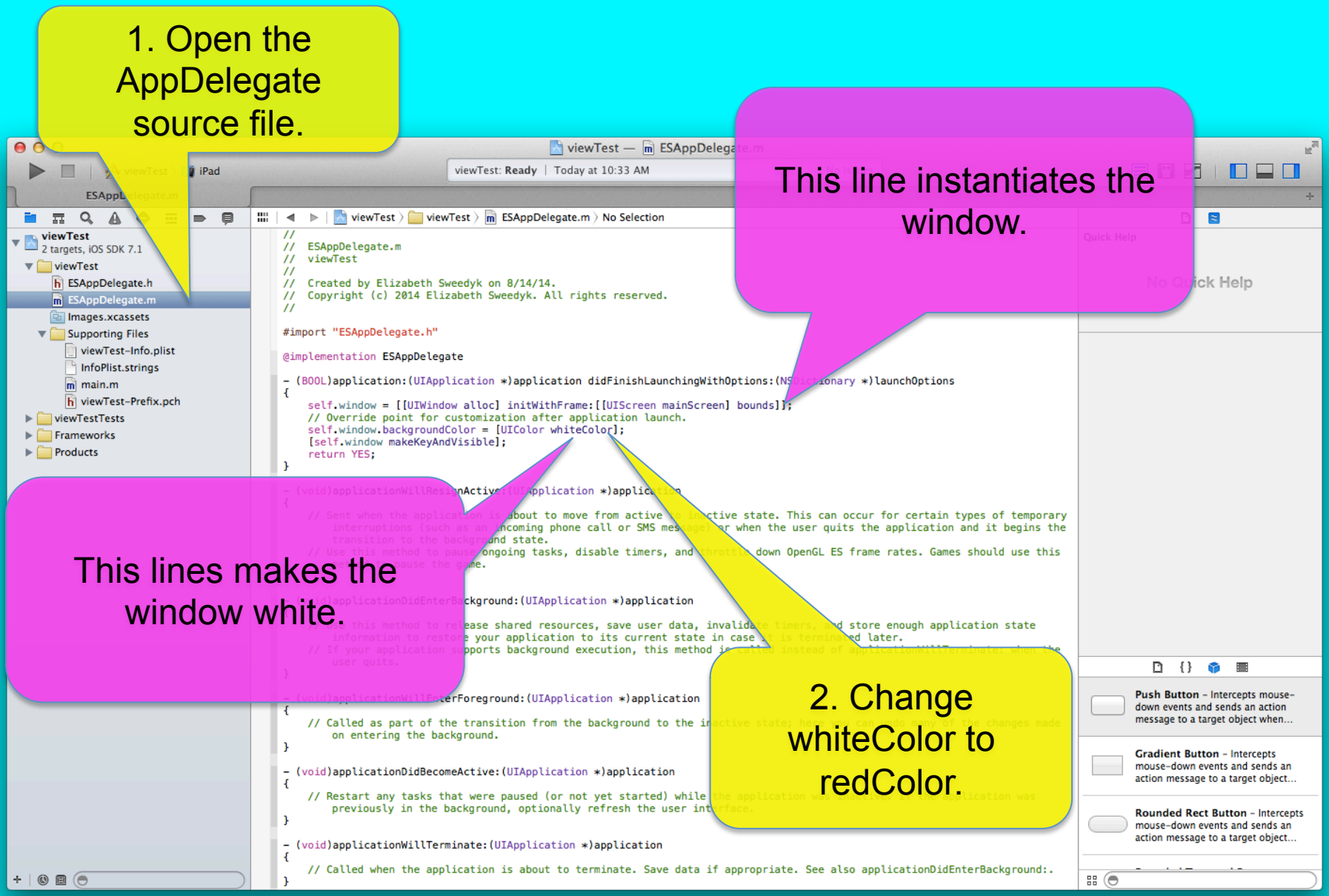

An empty application still has an AppDelegate, which launches the app, but no ViewController is generated.
Open the AppDelegate header file.
Every app needs a window. Here is ours.
viewTest
iPad
ESAppDelegate.h
viewTest
2 targets, iOS SDK 7.1
viewTest
ESAppDelegate.h
ESAppDelegate.m
Images.xcassets
Supporting Files
viewTest-Info.plist
InfoPlist.strings
main.m
viewTest-Prefix.pch
viewTestTests
Frameworks
Products
//
// 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
Quick Help
No Quick Help
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 and check out the simulator.



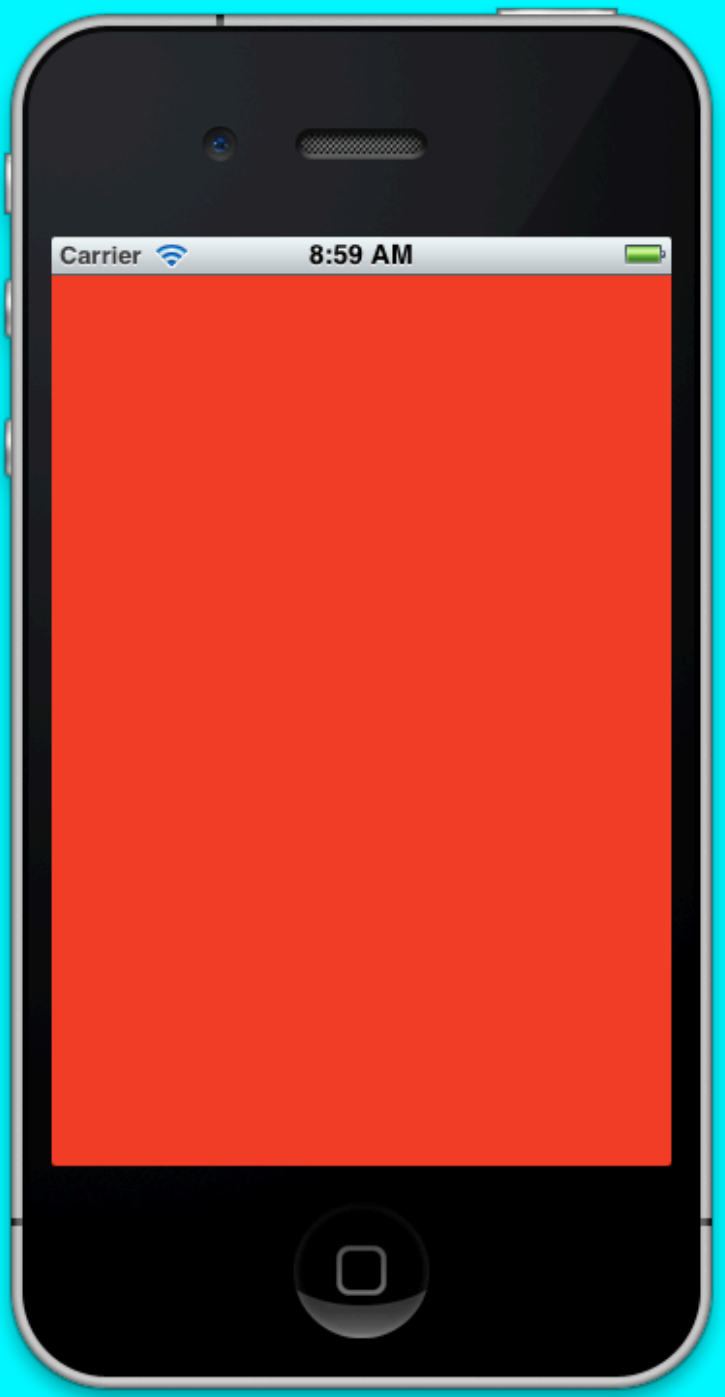
The white region is our window.

Even an empty app has a status bar.

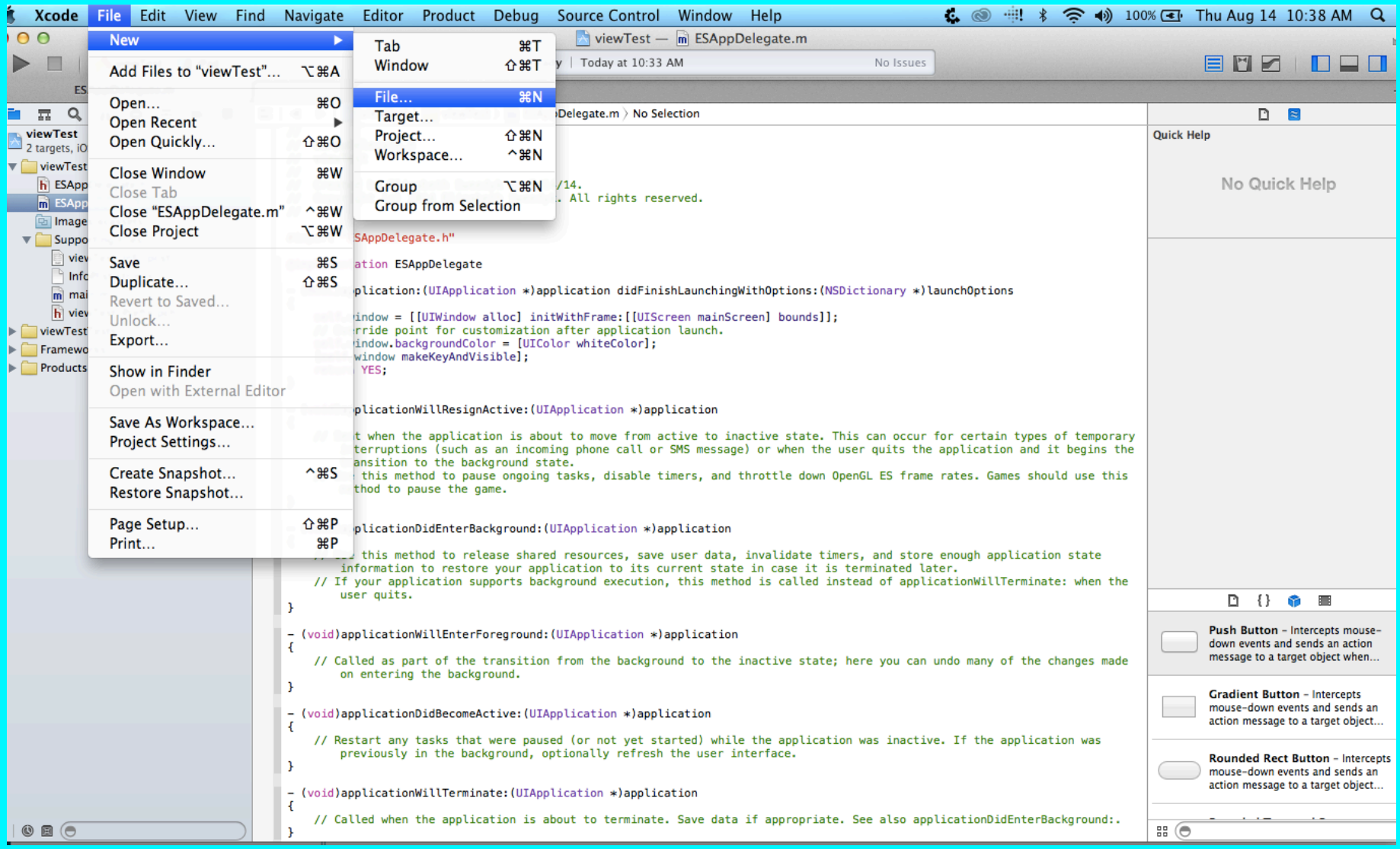

1. Open the AppDelegate source file.
This line instantiates the window.
This lines makes the window white.
2. Change whiteColor to redColor.
viewTest — ESAppDelegate.m
viewTest: Ready | Today at 10:33 AM
viewTest 〉 viewTest 〉 ESAppDelegate.m 〉 No Selection
viewTest
2 targets, iOS SDK 7.1
viewTest
ESAppDelegate.h
ESAppDelegate.m
Images.xcassets
Supporting Files
viewTest-Info.plist
InfoPlist.strings
main.m
viewTest-Prefix.pch
viewTestTests
Frameworks
Products
//
// ESAppDelegate.m
// 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;
}
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...

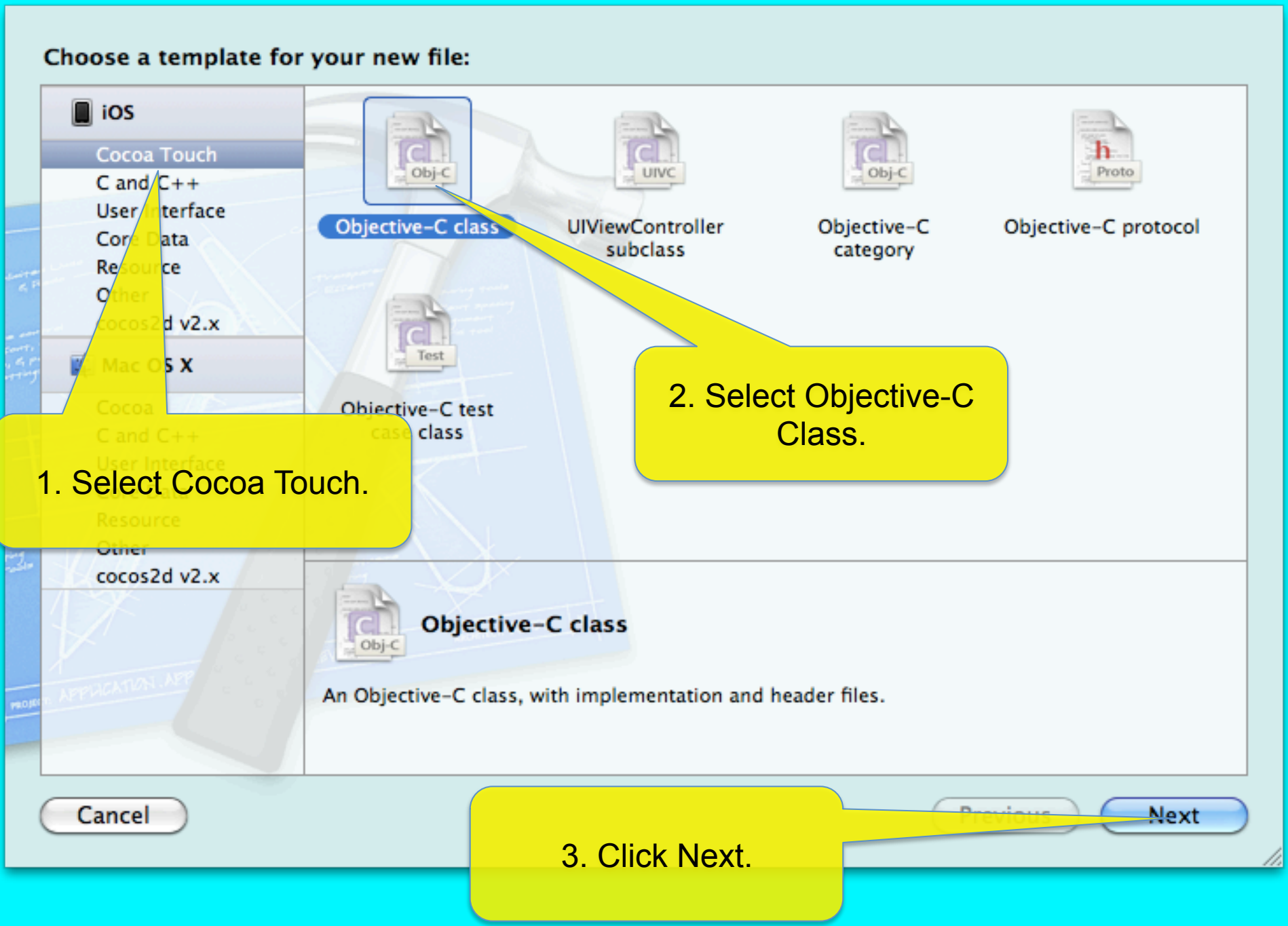
Run the app and check out the simulator.

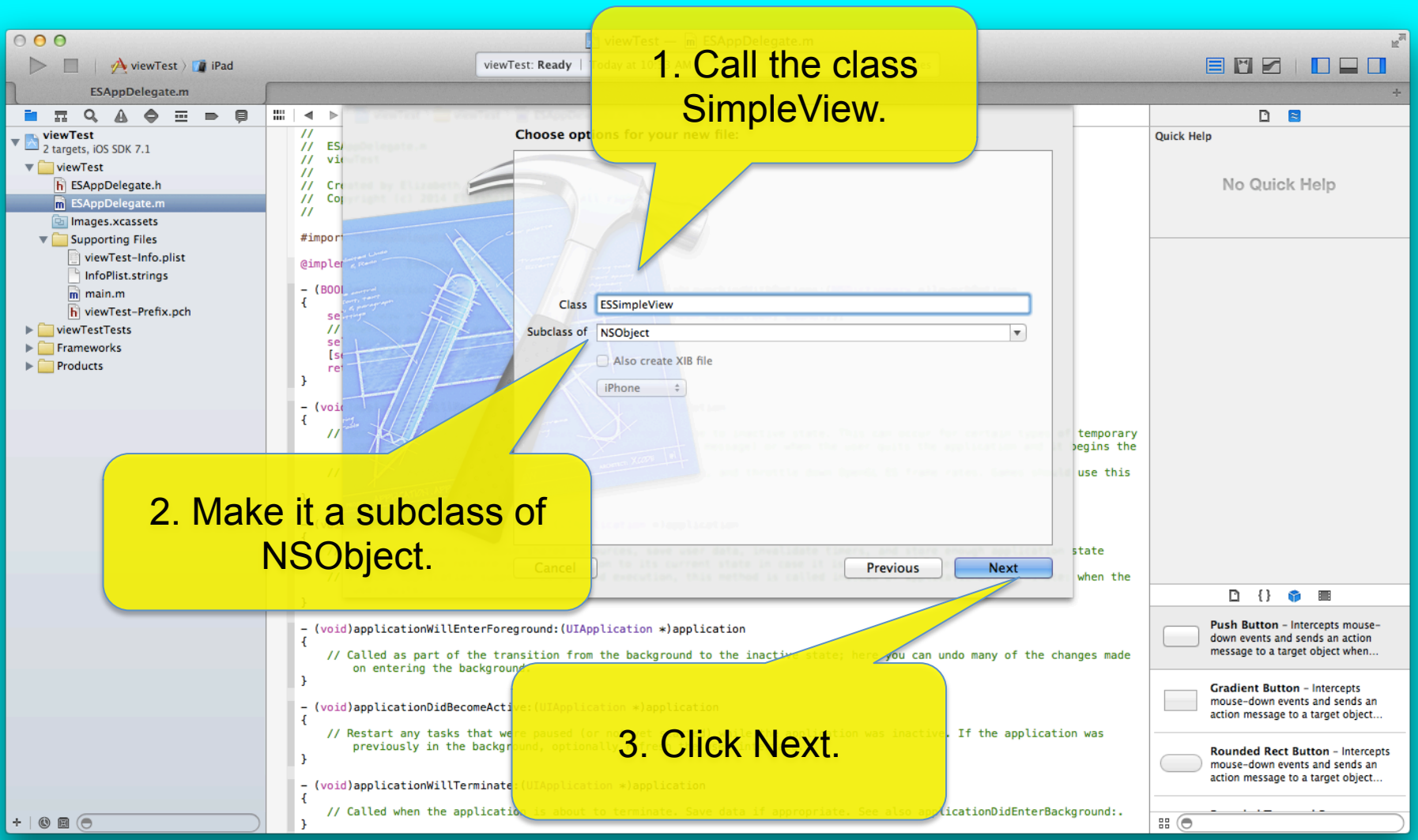
Now our window is red.



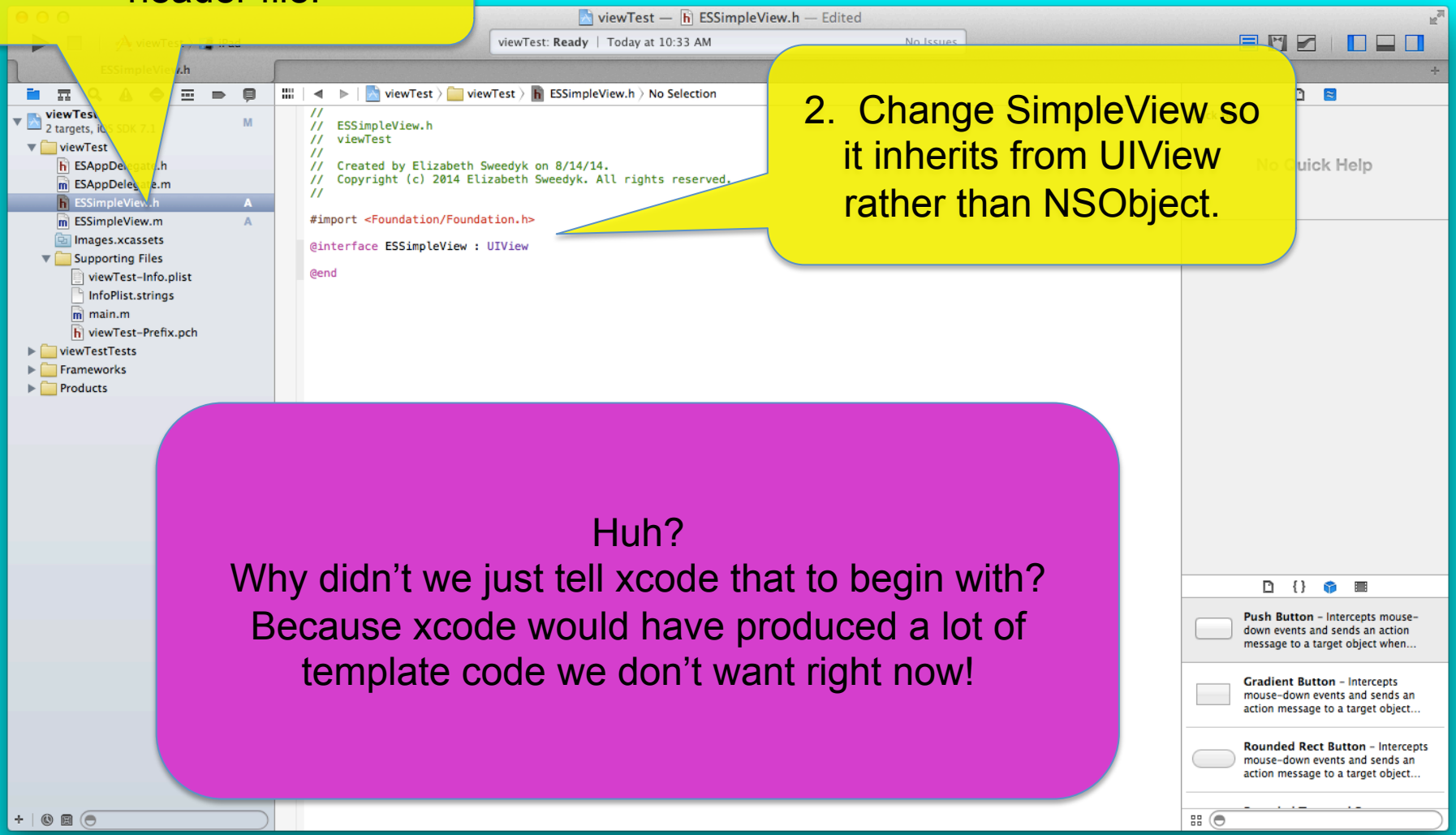
Select the File tab then New and New File.




Choose a template for your new file:
iOS
Cocoa Touch
C and C++
User Interface
Core Data
Resource
Other
cocos2d v2.x
Mac OS 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
Objective-C test case class
Objective-C class
An Objective-C class, with implementation and header files.
1. Select Cocoa Touch.
2. Select Objective-C Class.
3. Click Next.
Cancel
Previous
Next


1. Call the class SimpleView.
2. Make it a subclass of NSObject.
3. Click Next.
Class
ESSimpleView
Subclass of
NSObject
Also create XIB file
iPhone
Previous
Next
Cancel
Quick Help
No Quick Help
viewTest
2 targets, iOS SDK 7.1
ESAppDelegate.h
ESAppDelegate.m
Images.xcassets
Supporting Files
viewTest-Info.plist
InfoPlist.strings
main.m
viewTest-Prefix.pch
viewTestTests
Frameworks
Products
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...

1. Open the SimpleView header file.



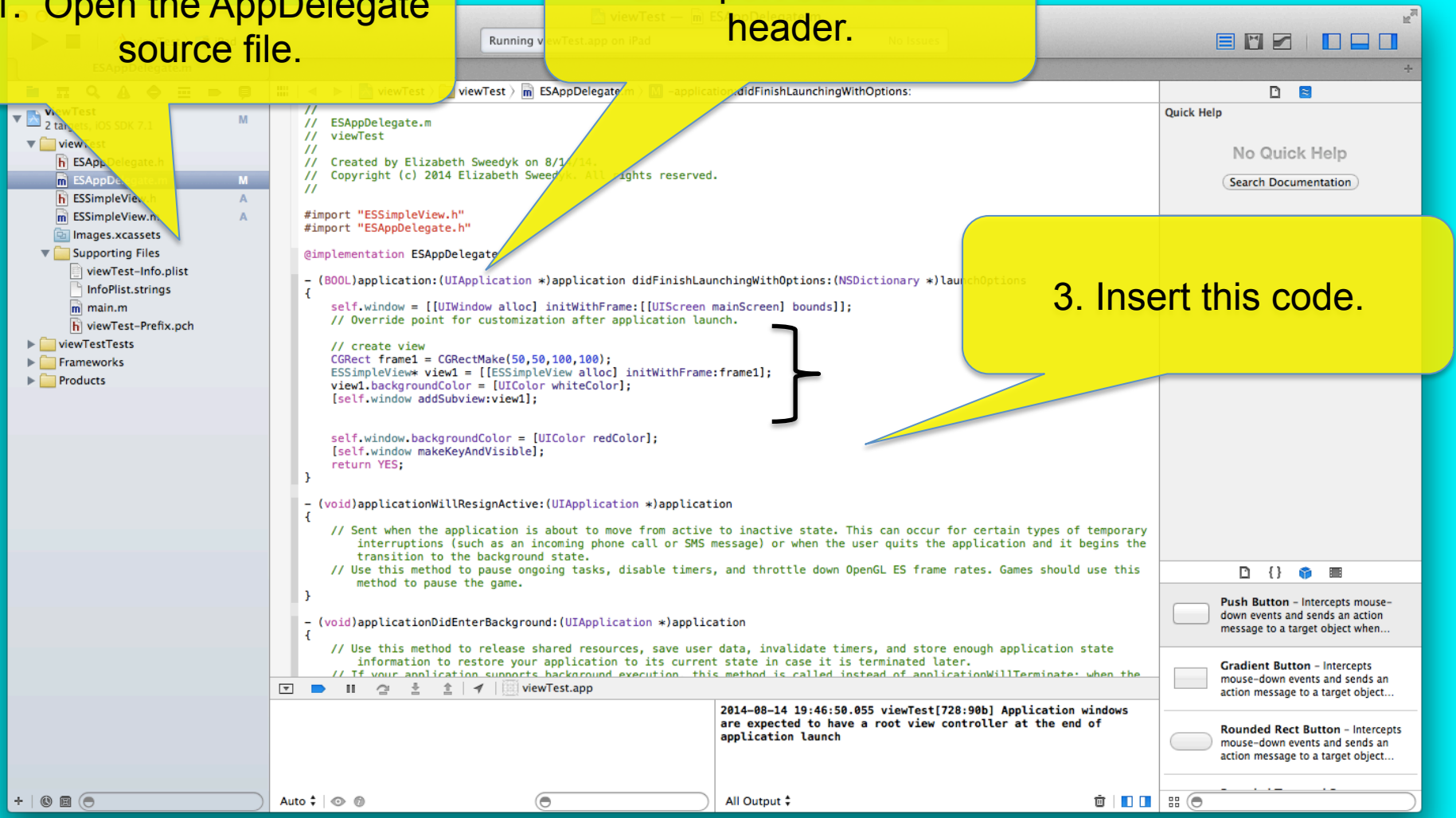
2. Change SimpleView so it inherits from UIView rather than NSObject.

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.

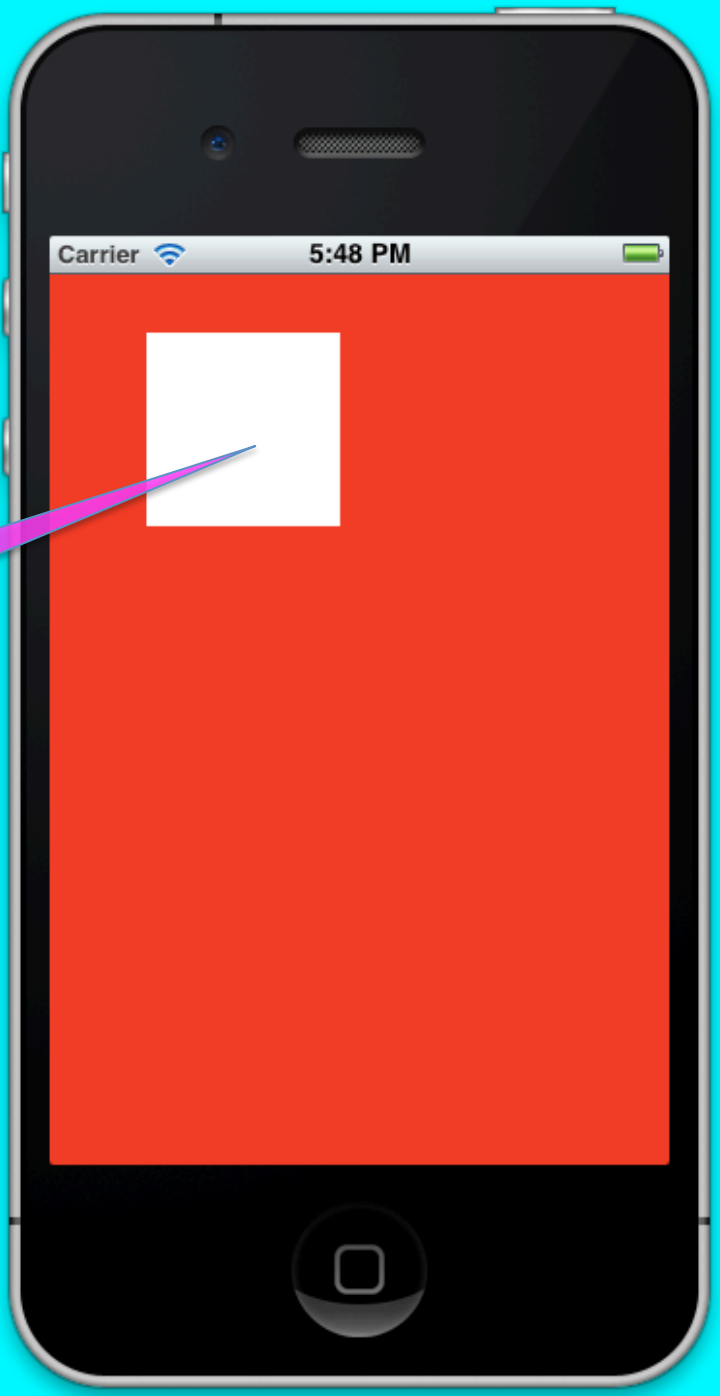
2. Import our class header.

3. Insert this code.



Run the app.

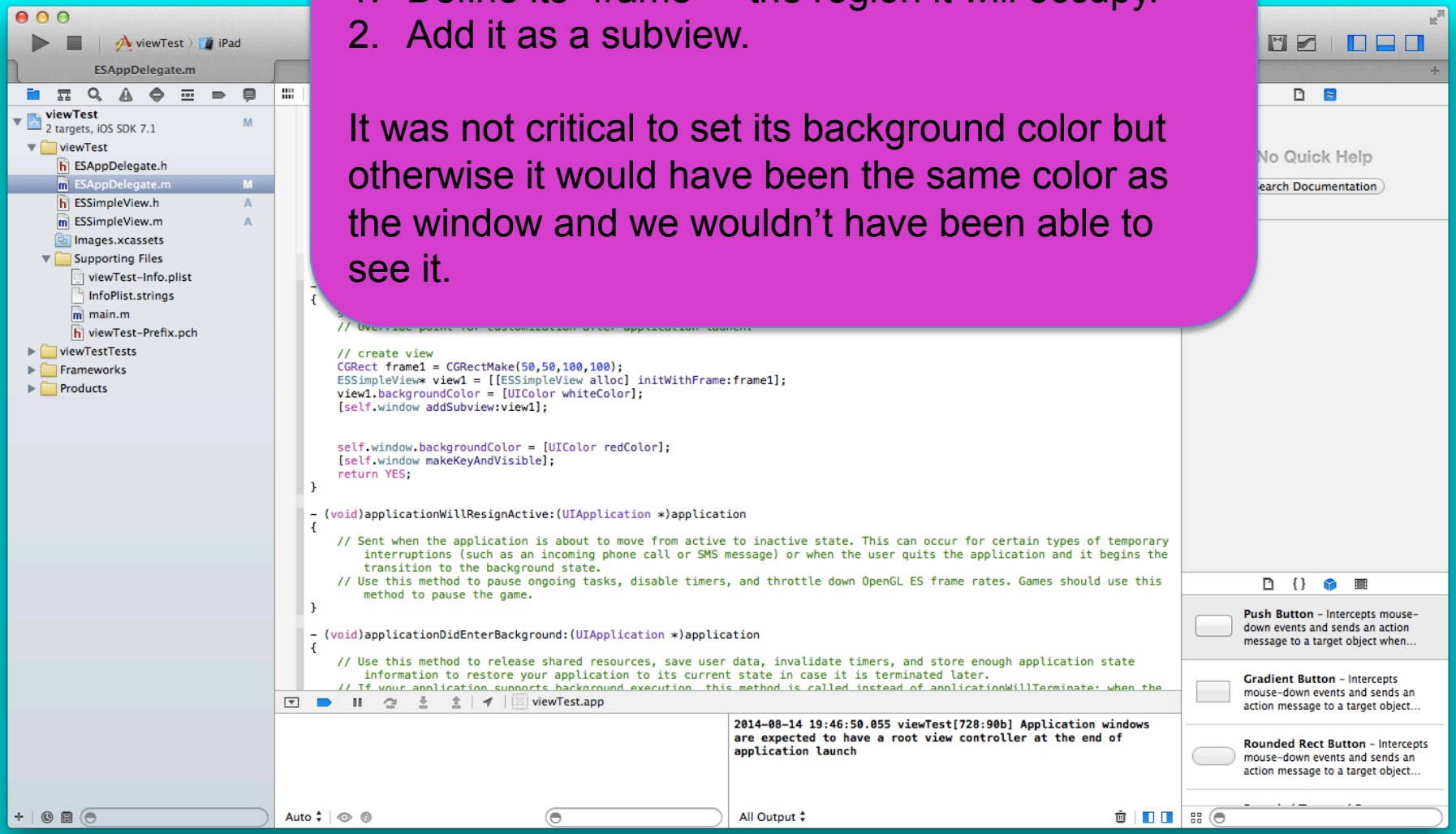
This is our new view.



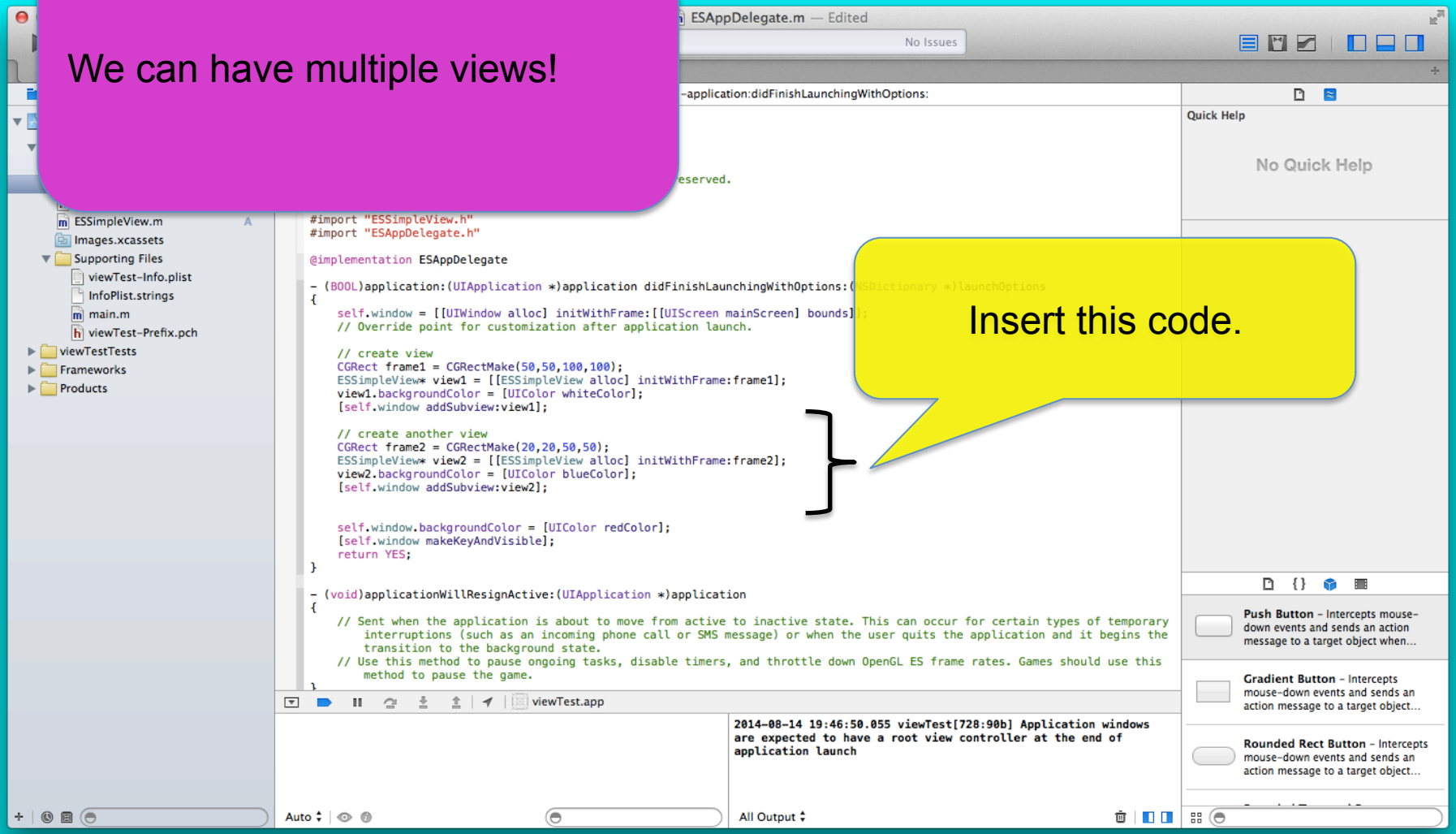
A view is like a subwindow. The key steps to creating a view are to:

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.

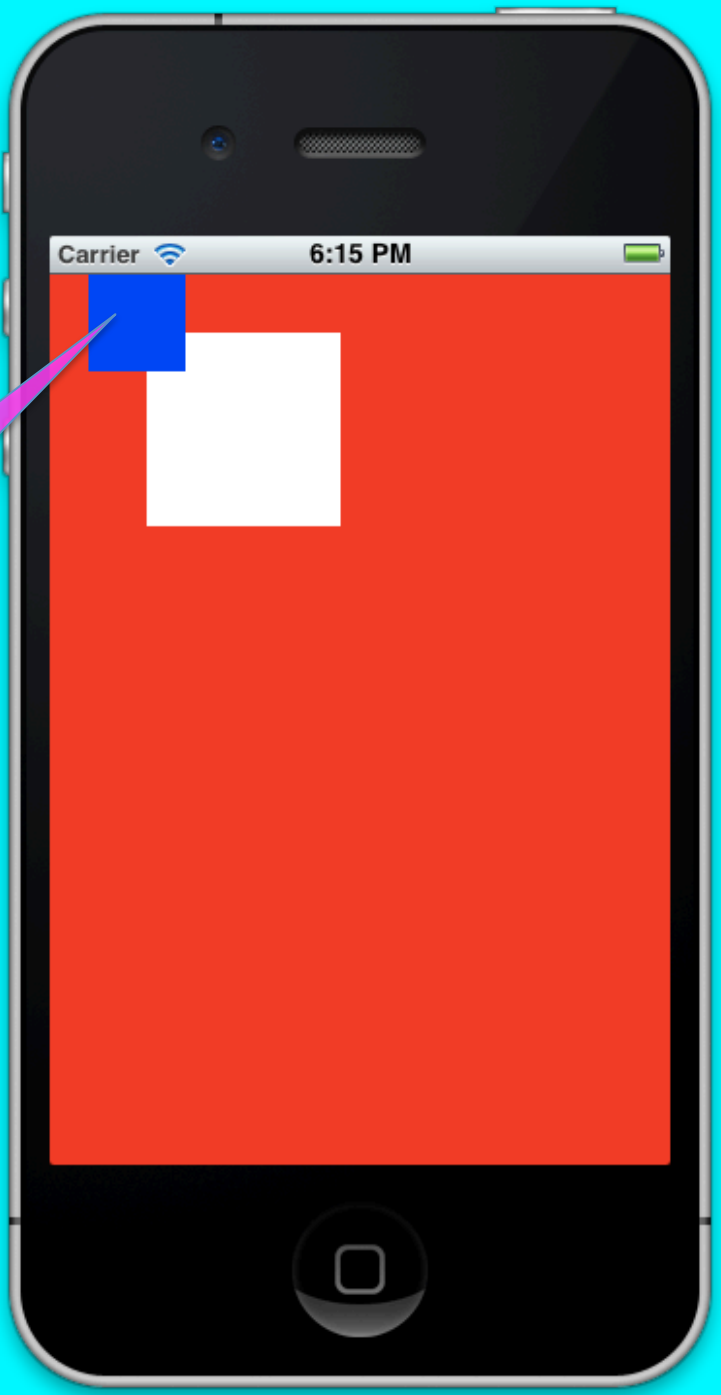


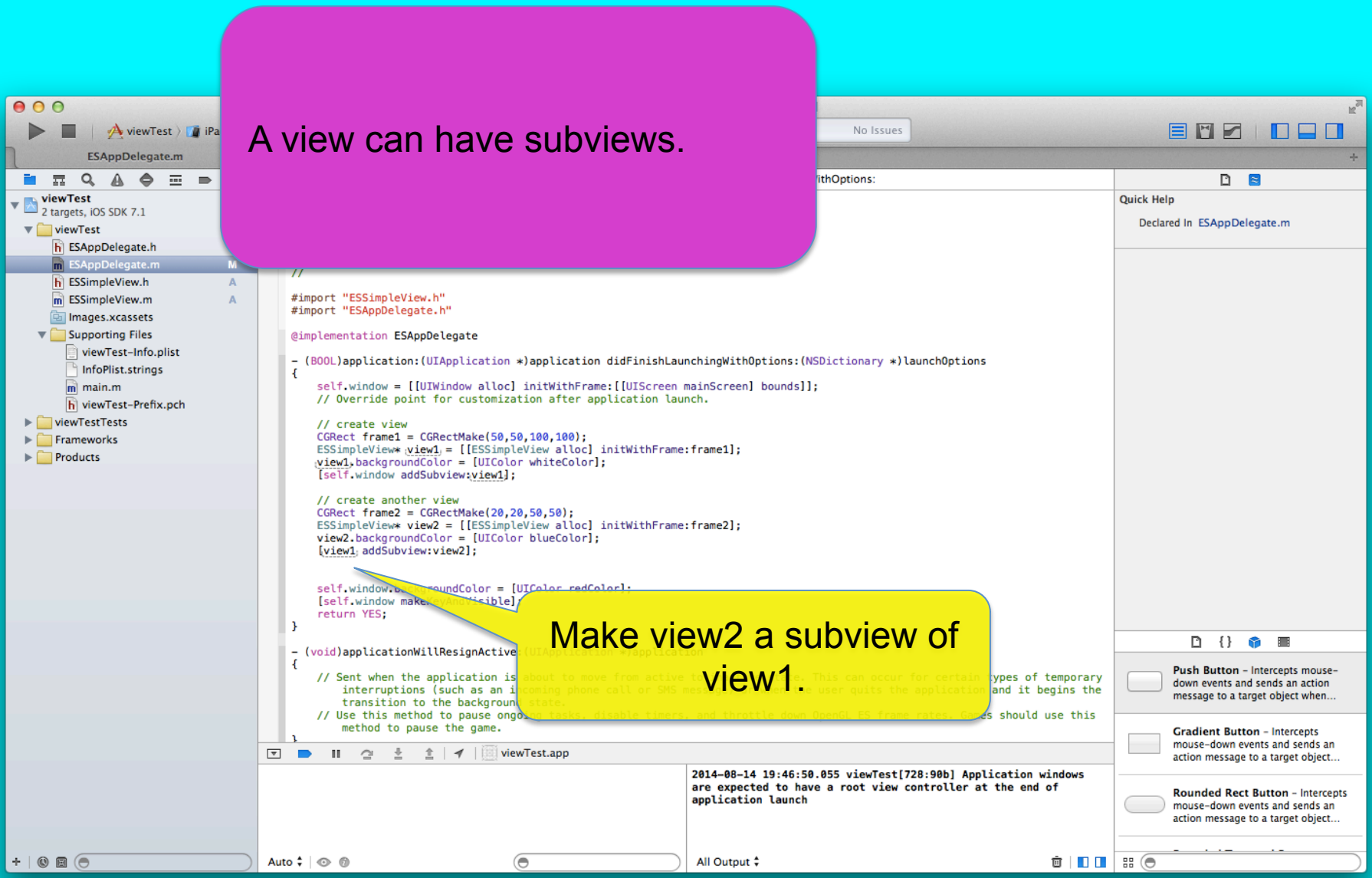
We can have multiple views!



Run the app.

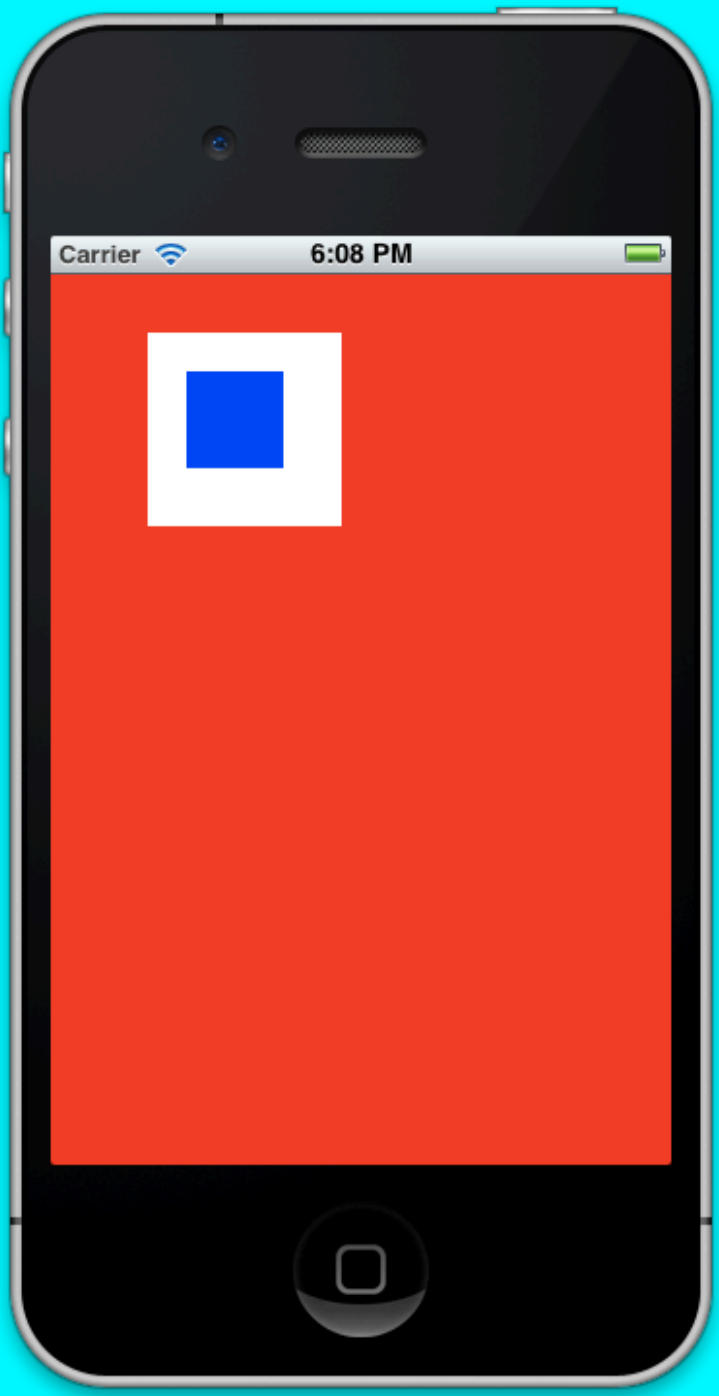
This is our new view. Views are drawn in the order they are added as subviews, so the blue view is drawn on top of the red one. (There is a way to reorder the views.)

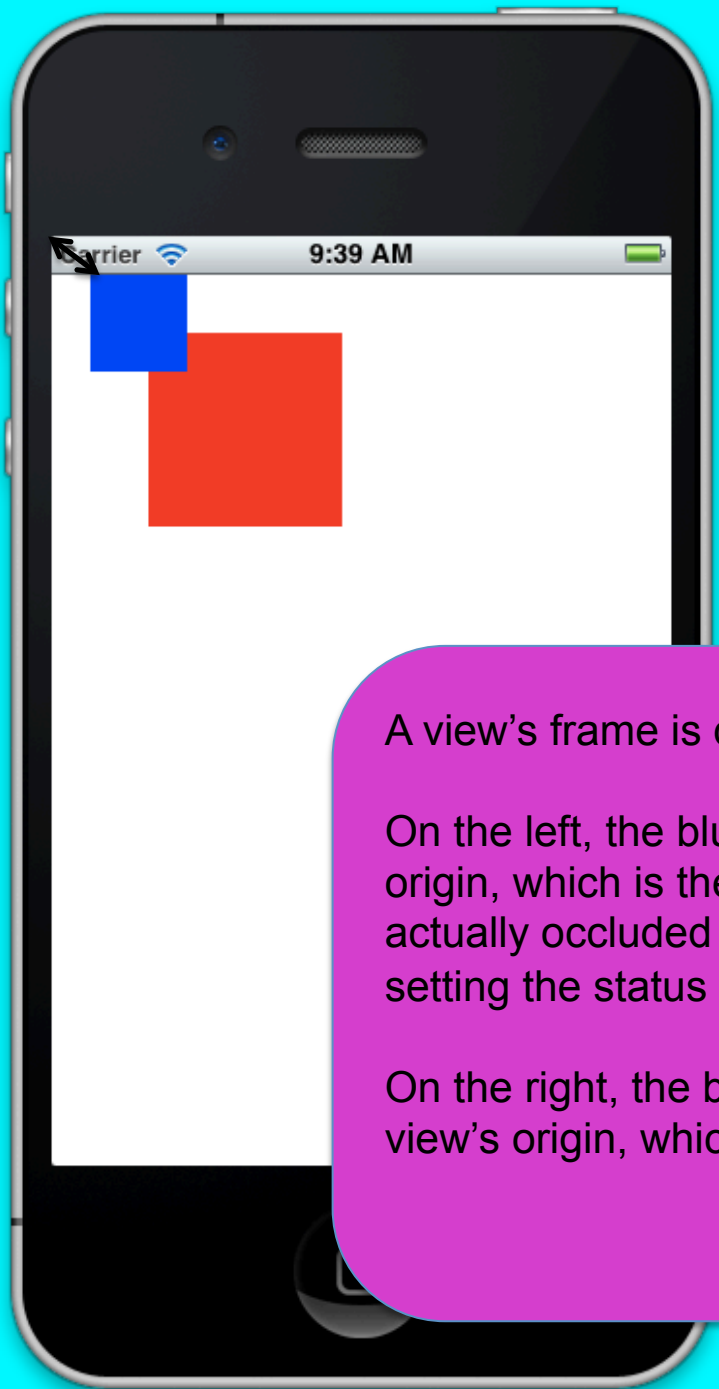


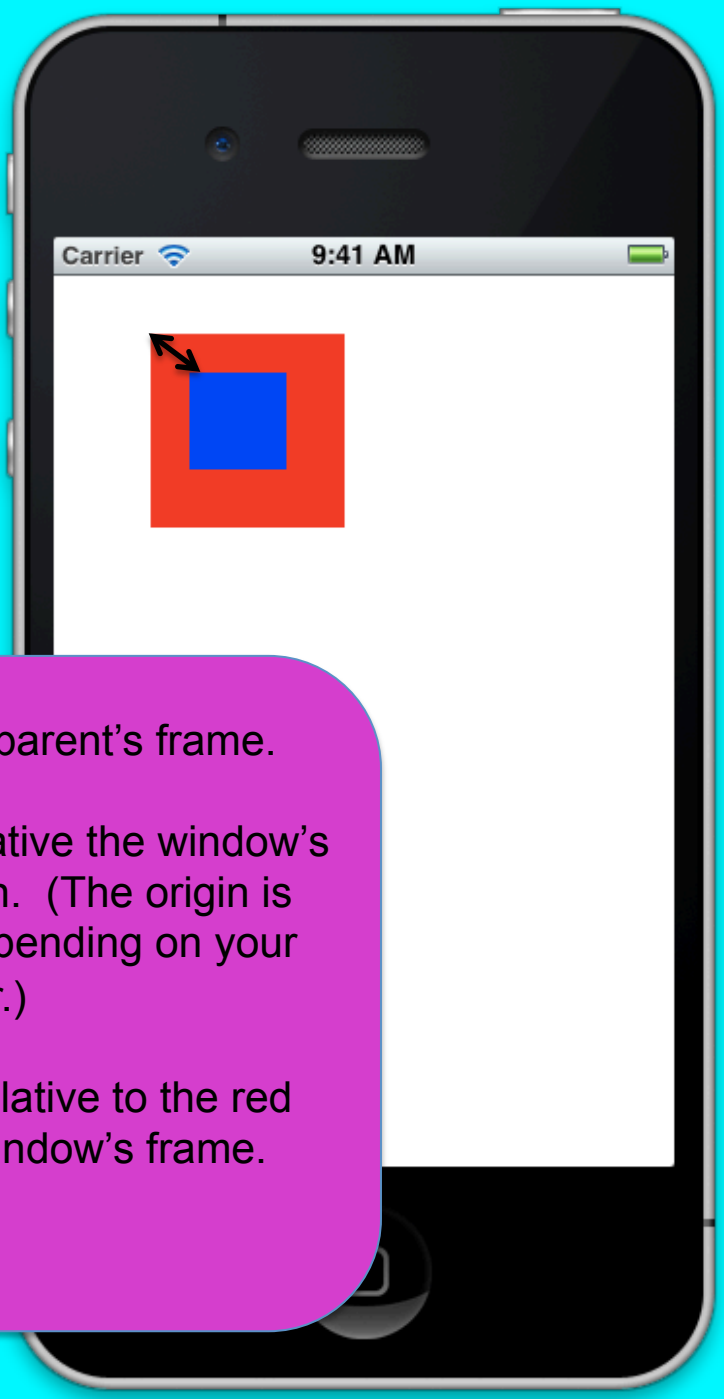

A view can have subviews.
viewTest
2 targets, iOS SDK 7.1
viewTest
ESAppDelegate.h
ESAppDelegate.m
ESSimpleView.h
ESSimpleView.m
Images.xcassets
Supporting Files
viewTest-Info.plist
InfoPlist.strings
main.m
viewTest-Prefix.pch
viewTestTests
Frameworks
Products
#import "ESSimpleView.h"
#import "ESAppDelegate.h"
@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];
return YES;
}
Make view2 a subview of view1.
Quick Help
Declared In ESAppDelegate.m
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...
viewTest.app
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
Auto
All Output

Run the app.

Is this what you expected?





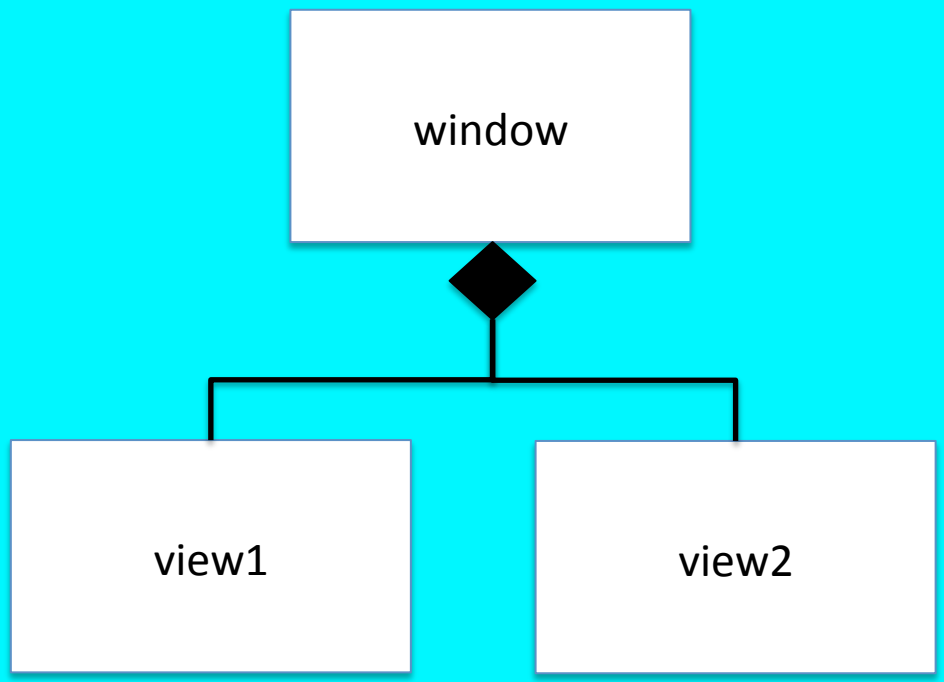


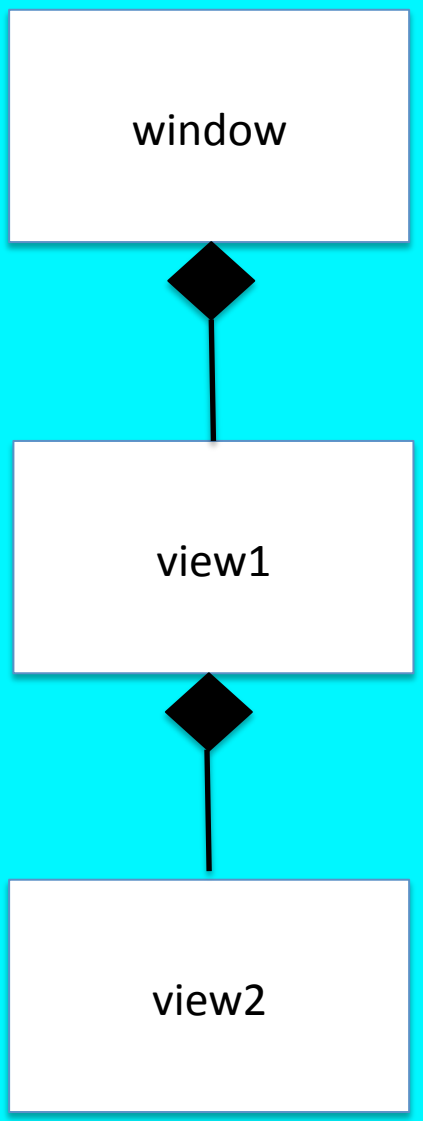
A view's frame is defined relative to its parent's frame.

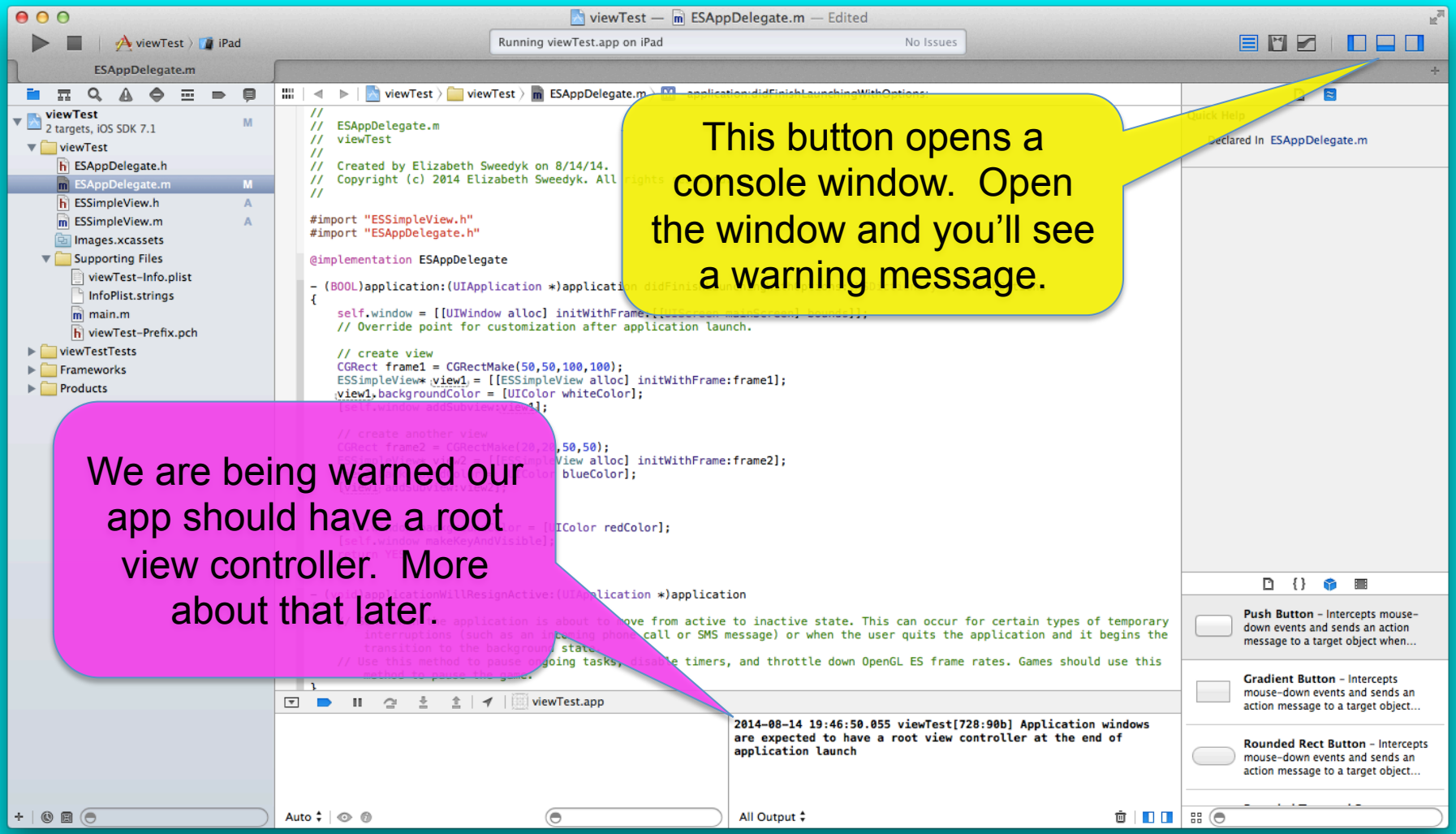
On the left, the blue view is defined relative the window's origin, which is the top left of the screen. (The origin is actually occluded by the status bar. Depending on your setting the status bar may also be clear.)

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!



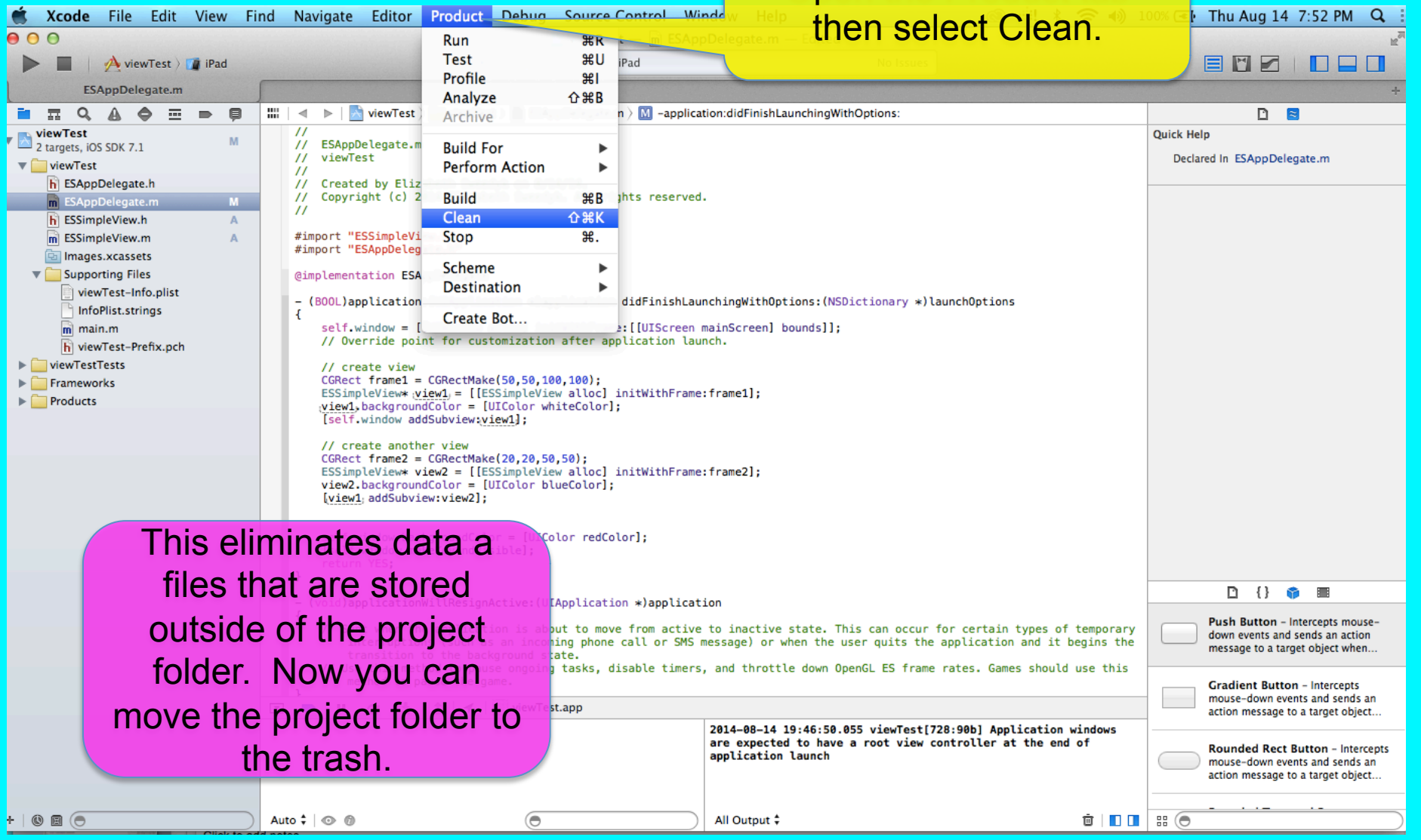



This button opens a console window. Open the window and you'll see a warning message.
We are being warned our app should have a root view controller. More about that later.
2014-08-14 19:46:50.055 viewTest[728:90b] Application windows are expected to have a root view controller at the end of application launch

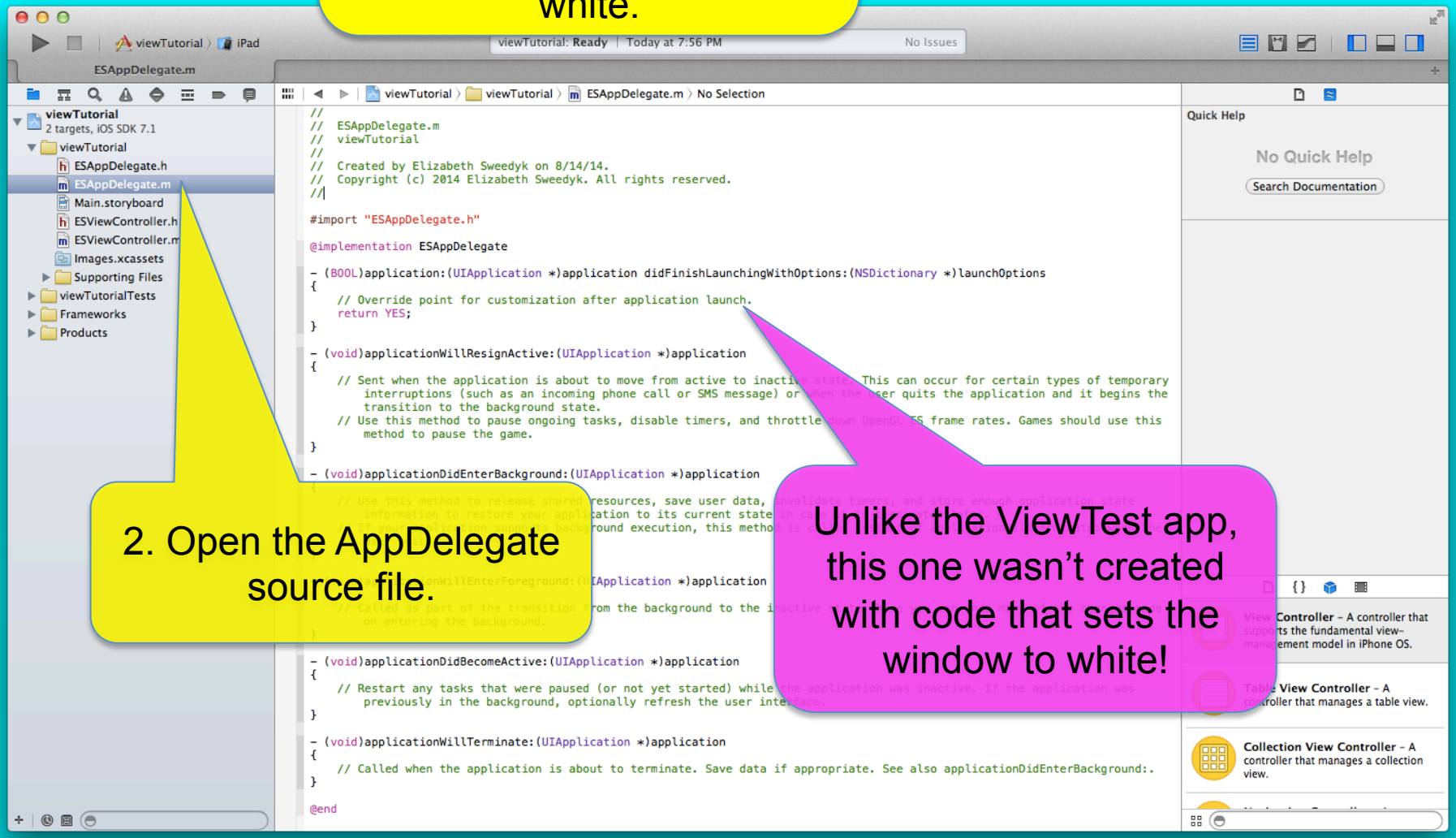
What we did was an experiment. Until you become an iOS expert, you would be wise to use the template code.

- 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.

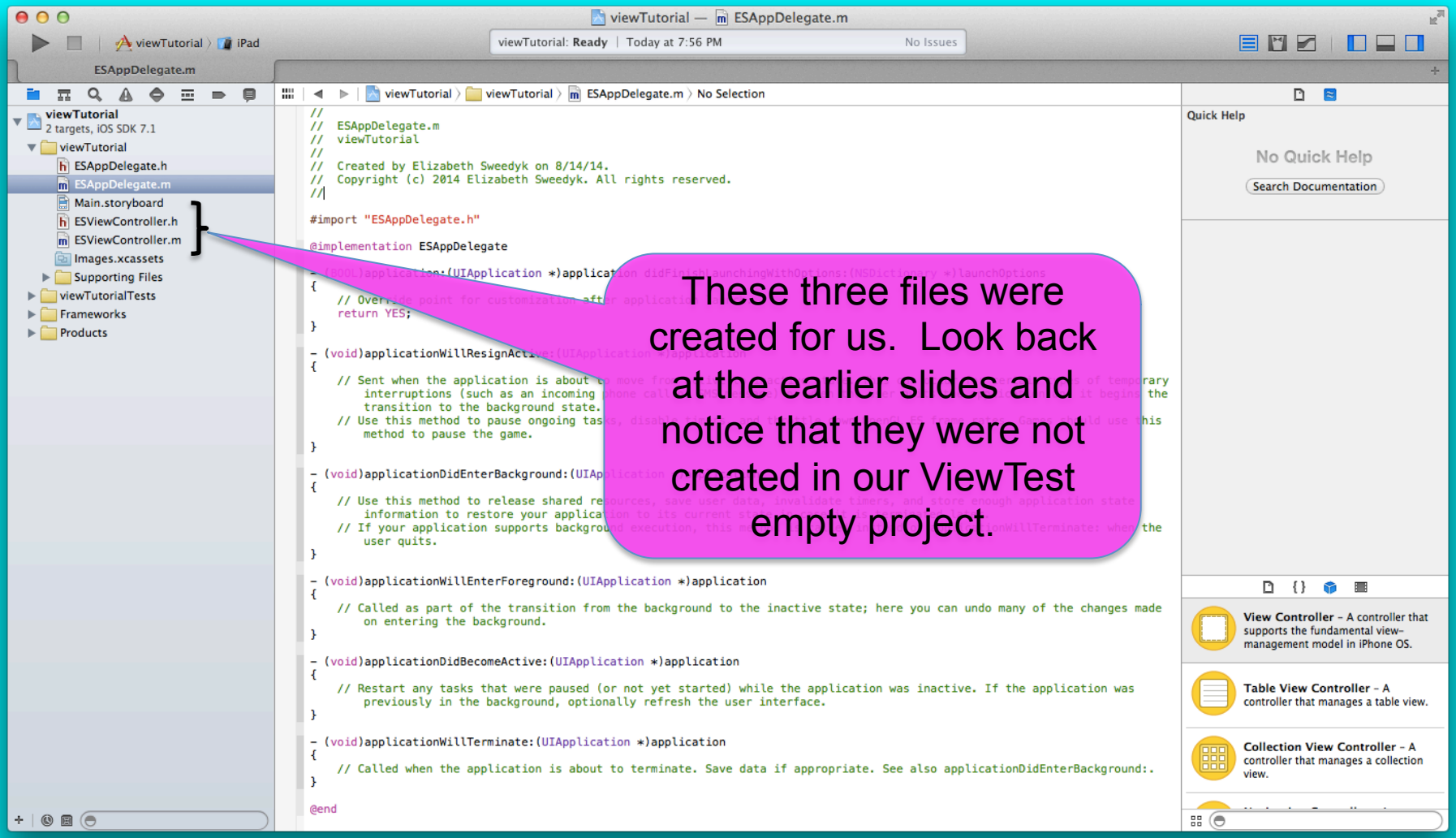

Open the Product tab then select Clean.
Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Thu Aug 14 7:52 PM
Run ⌘R
Test ⌘U
Profile ⌘I
Analyze ⇧⌘B
Archive
Build For
Perform Action
Build ⌘B
Clean ⇧⌘K
Stop ⌘.
Scheme
Destination
Create Bot...
viewTest
2 targets, iOS SDK 7.1
ESAppDelegate.h
ESAppDelegate.m
ESSimpleView.h
ESSimpleView.m
Images.xcassets
Supporting Files
viewTest-Info.plist
InfoPlist.strings
main.m
viewTest-Prefix.pch
viewTestTests
Frameworks
Products
-application:didFinishLaunchingWithOptions:
Quick Help
Declared In ESAppDelegate.m
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...
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
This eliminates data a files that are stored outside of the project folder. Now you can move the project folder to the trash.

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



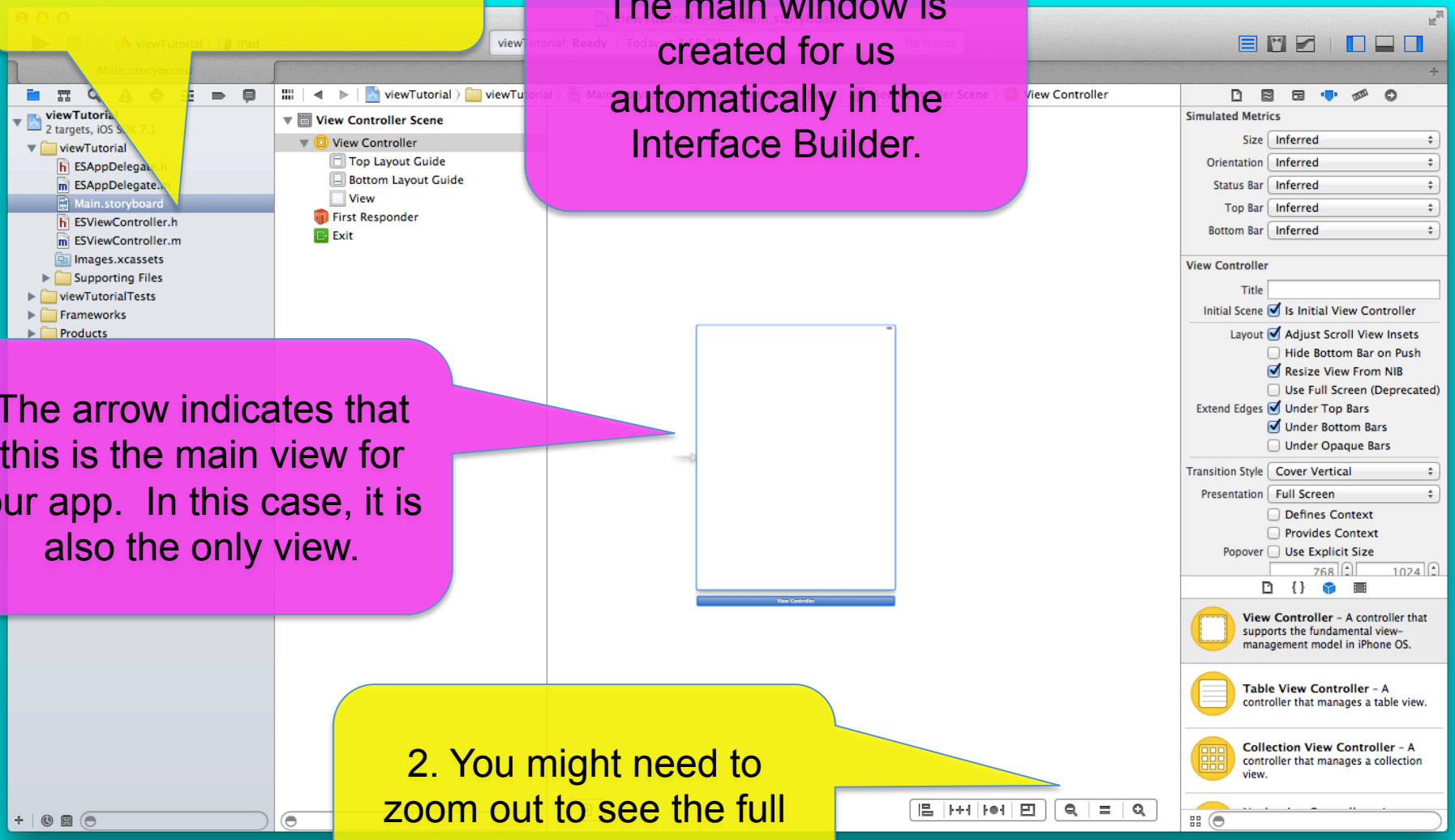
2. Open the AppDelegate source file.

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


viewTutorial — ESAppDelegate.m
viewTutorial: Ready | Today at 7:56 PM
No Issues
ESAppDelegate.m
viewTutorial
2 targets, iOS SDK 7.1
viewTutorial
ESAppDelegate.h
ESAppDelegate.m
Main.storyboard
ESViewController.h
ESViewController.m
Images.xcassets
Supporting Files
viewTutorialTests
Frameworks
Products
viewTutorial › viewTutorial › ESAppDelegate.m › No Selection
//
// ESAppDelegate.m
// viewTutorial
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//
#import "ESAppDelegate.h"
@implementation ESAppDelegate
{
return YES;
}
{
}
{
}
{
}
{
}
@end
Quick Help
No Quick Help
Search Documentation
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Collection View Controller – A controller that manages a collection view.
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.

1. Open the storyboard.

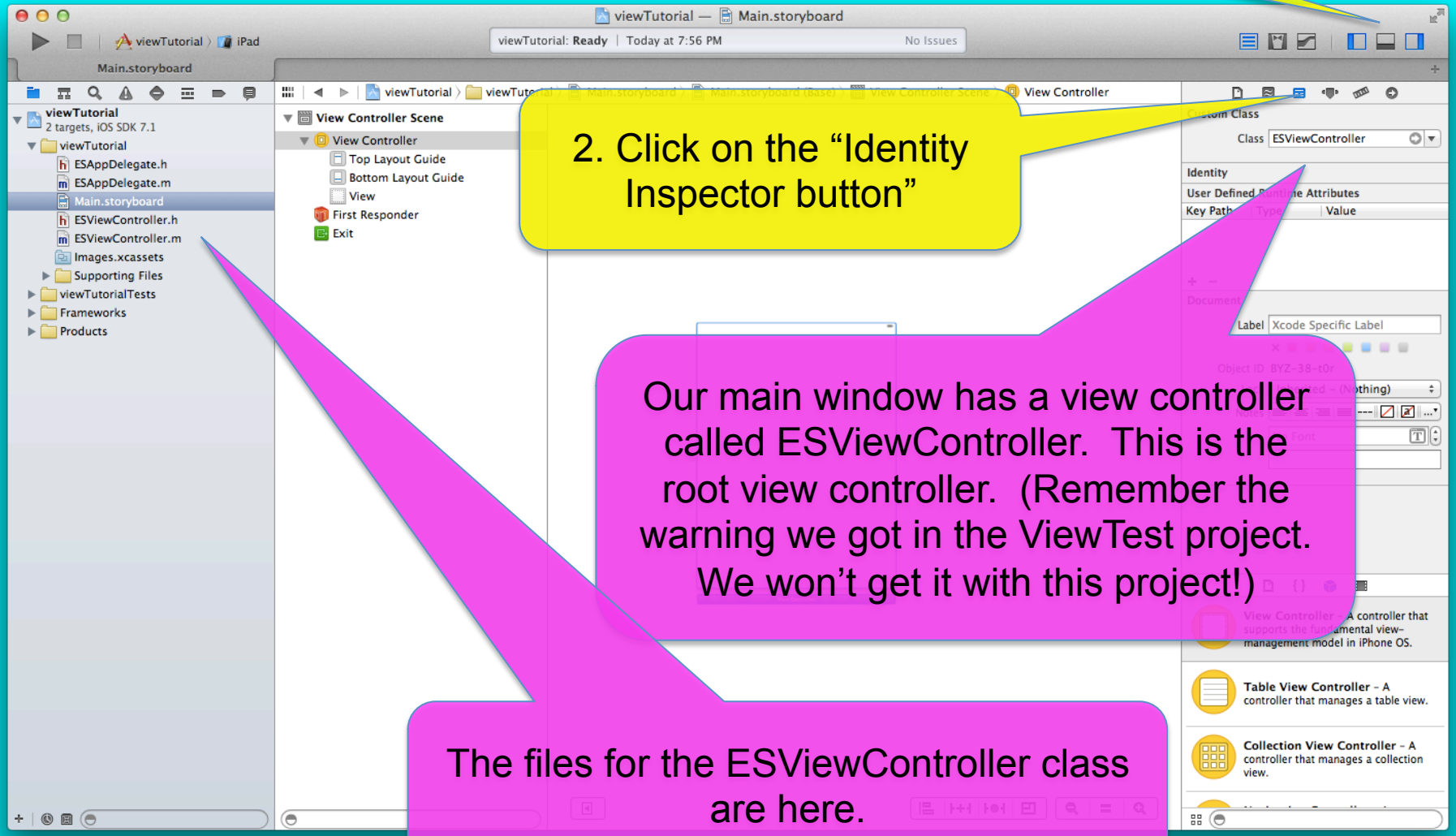
The main window is created for us automatically in the Interface Builder.



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!.



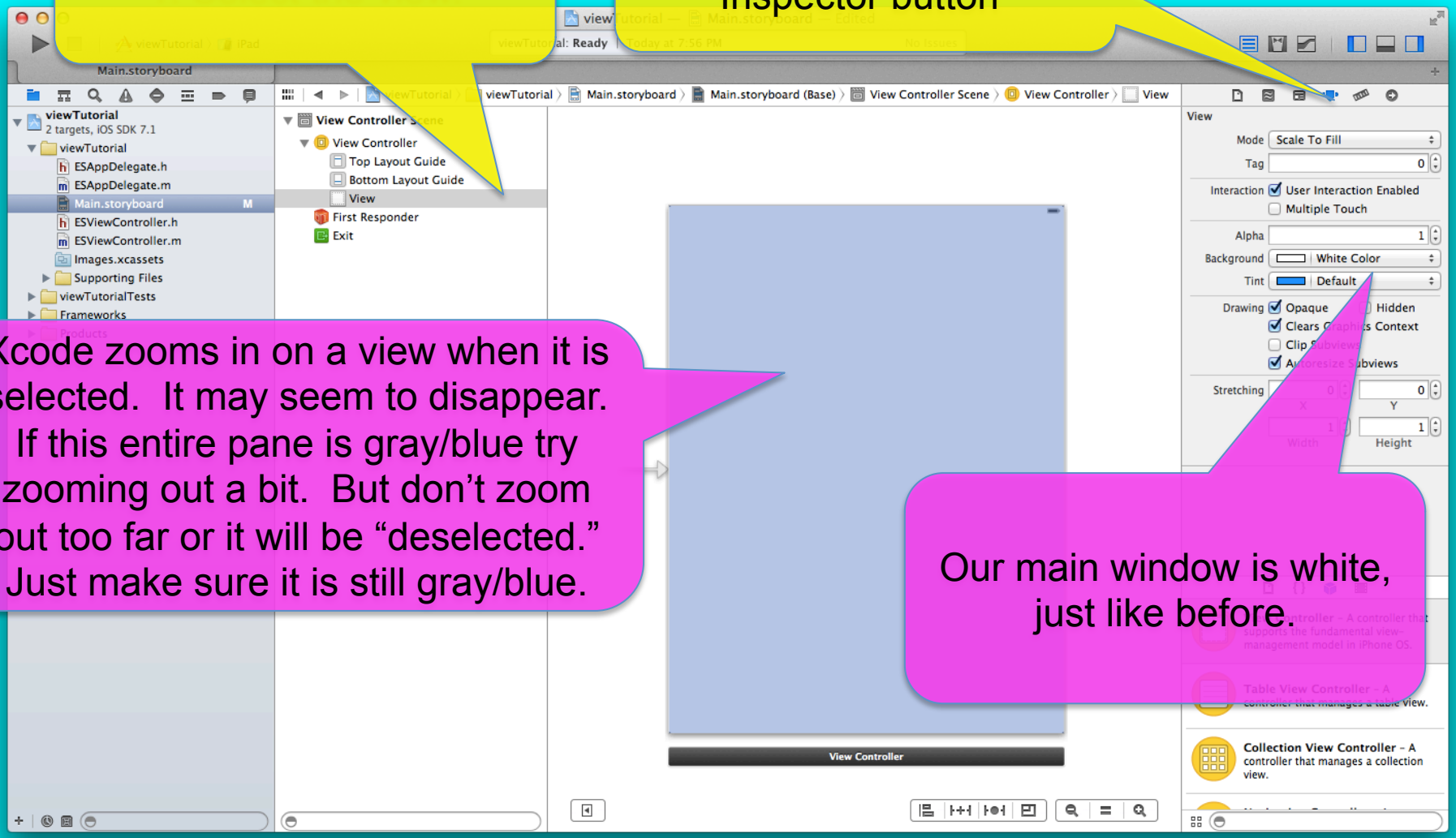
2. Click on the “Identity Inspector button”

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.

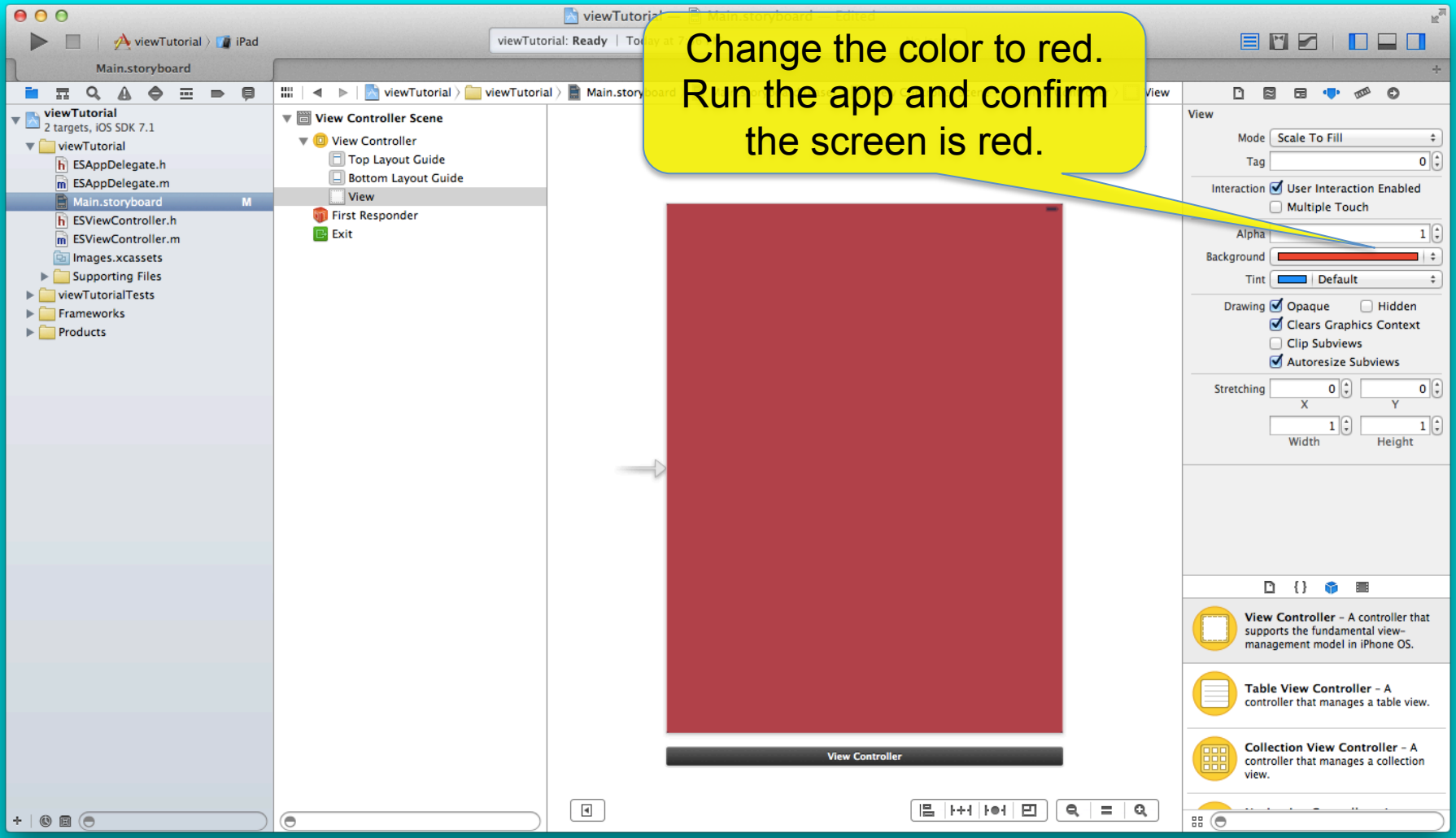
1. Select the view.

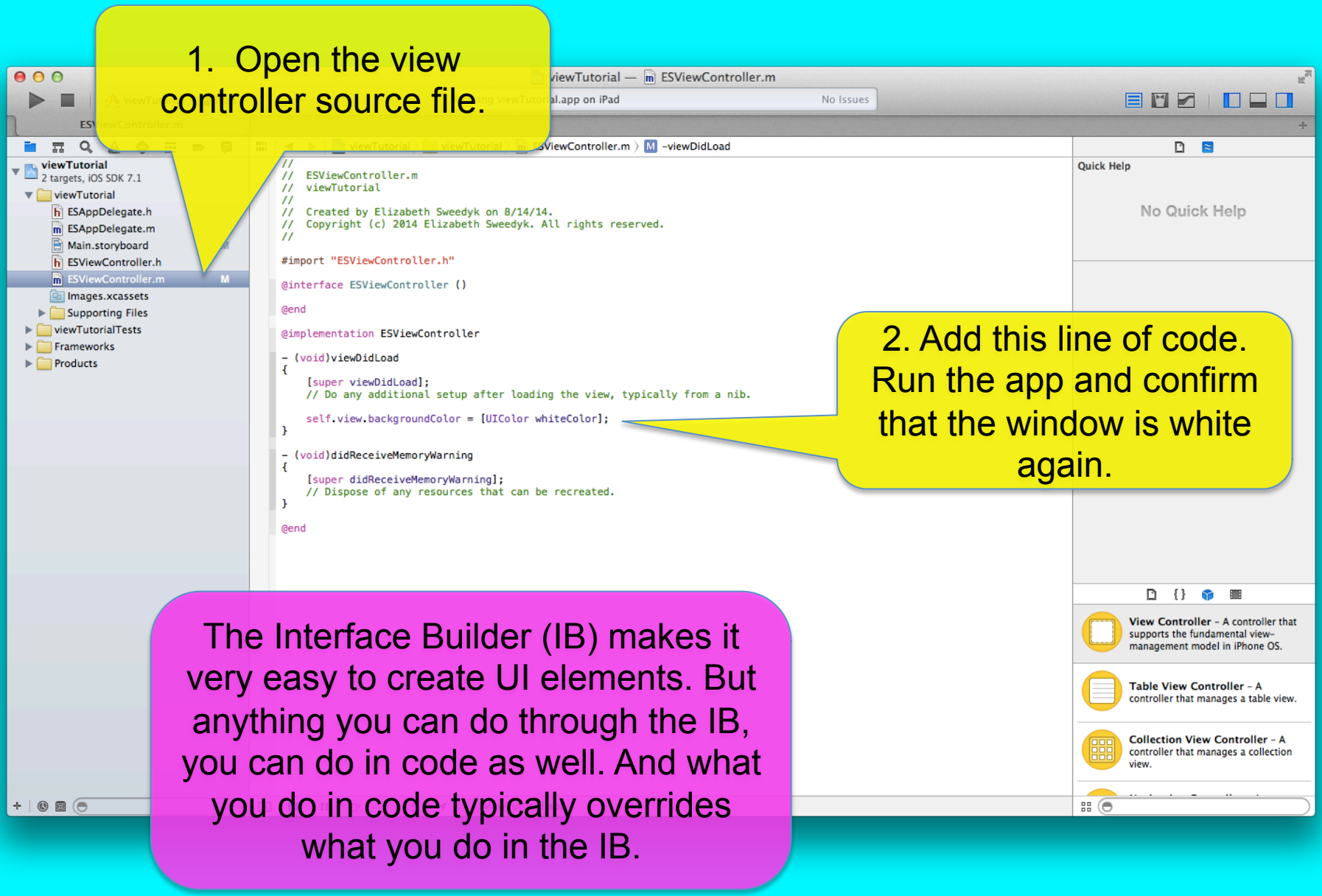
2. Click on the “Attributes Inspector button”

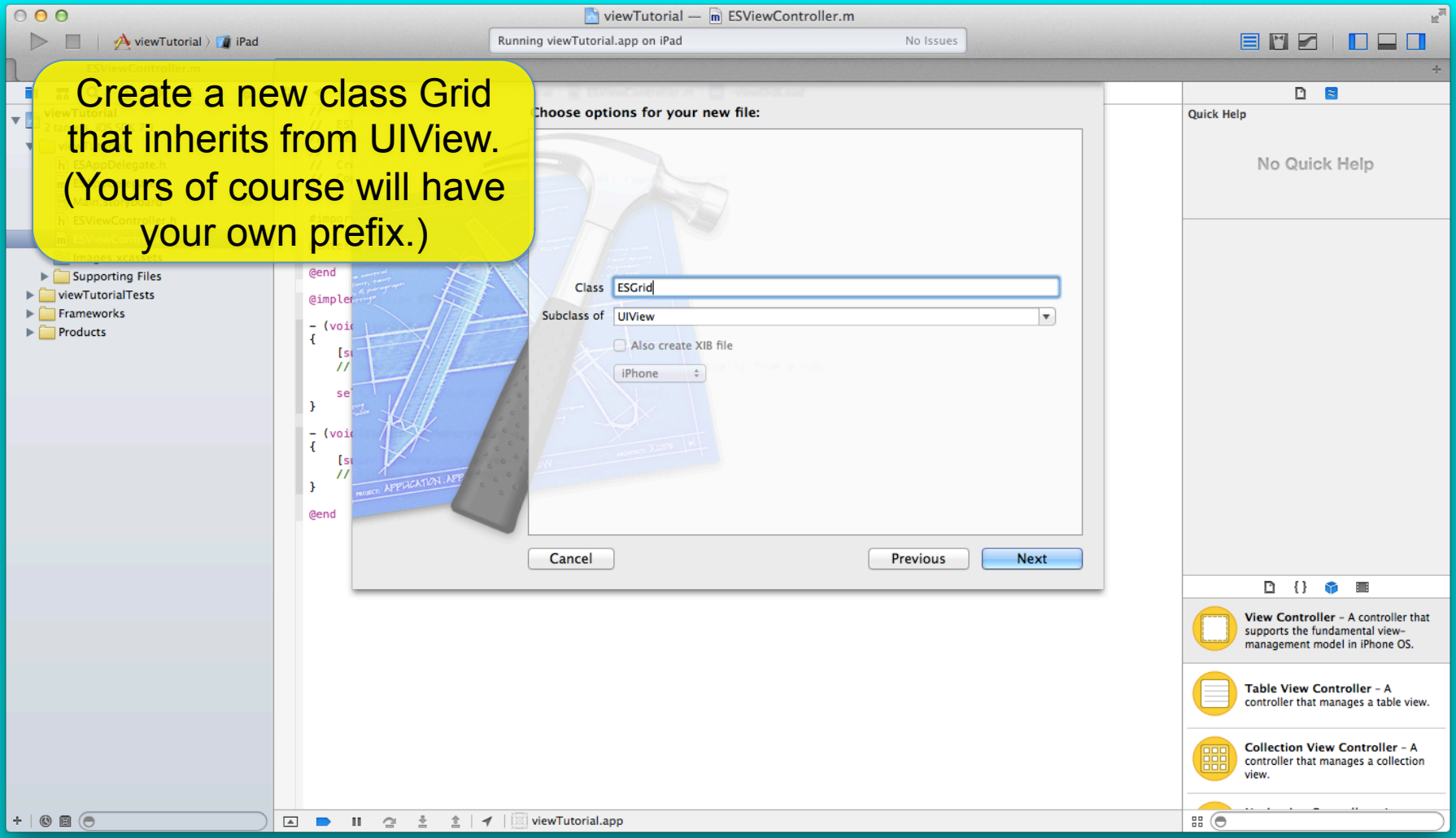


Xcode zooms in on a view when it is selected. It may seem to disappear. If this entire pane is gray/blue try zooming out a bit. But don’t zoom out too far or it will be “deselected.” Just make sure it is still gray/blue.

Our main window is white, just like before.

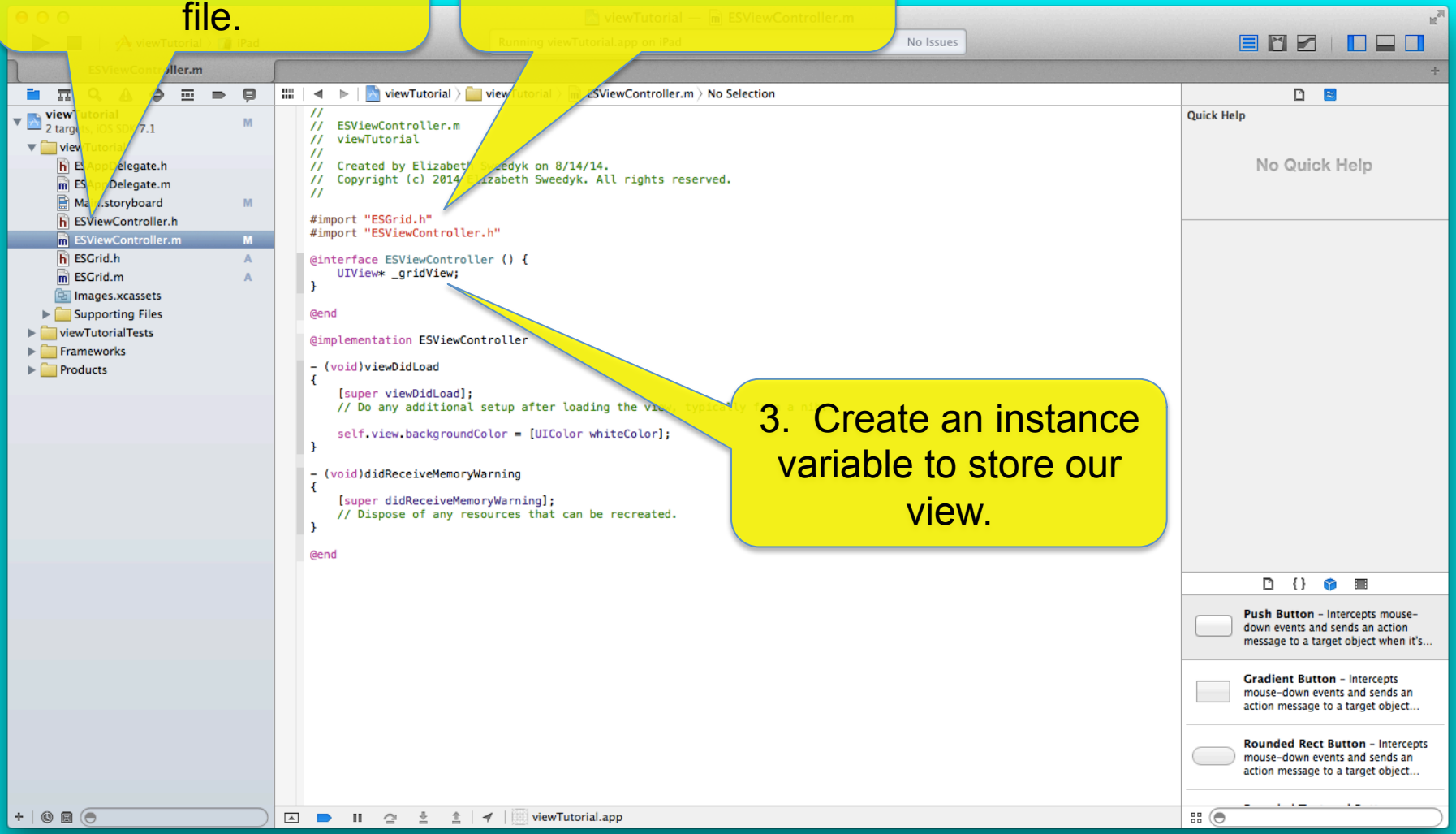

Change the color to red.
Run the app and confirm the screen is red.
viewTutorial
iPad
viewTutorial: Ready
Main.storyboard
viewTutorial
2 targets, iOS SDK 7.1
ESAppDelegate.h
ESAppDelegate.m
Main.storyboard
ESViewController.h
ESViewController.m
Images.xcassets
Supporting Files
viewTutorialTests
Frameworks
Products
View Controller Scene
View Controller
Top Layout Guide
Bottom Layout Guide
View
First Responder
Exit
View Controller
Mode
Scale To Fill
Tag
Interaction
User Interaction Enabled
Multiple Touch
Alpha
Background
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clip Subviews
Autoresize Subviews
Stretching
X
Y
Width
Height
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Collection View Controller – A controller that manages a collection view.


1. Open the view controller source file.
viewTutorial — ESViewController.m
No Issues
viewTutorial
2 targets, iOS SDK 7.1
viewTutorial
ESAppDelegate.h
ESAppDelegate.m
Main.storyboard
ESViewController.h
ESViewController.m
Images.xcassets
Supporting Files
viewTutorialTests
Frameworks
Products
ESViewController.m
-viewDidLoad
//
// ESViewController.m
// viewTutorial
//
// Created by Elizabeth Sweedyk on 8/14/14.
// Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//
#import "ESViewController.h"
@interface ESViewController ()
@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
Quick Help
No Quick Help
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Collection View Controller – A controller that manages a collection view.
2. Add this line of code. Run the app and confirm that the window is white again.
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.

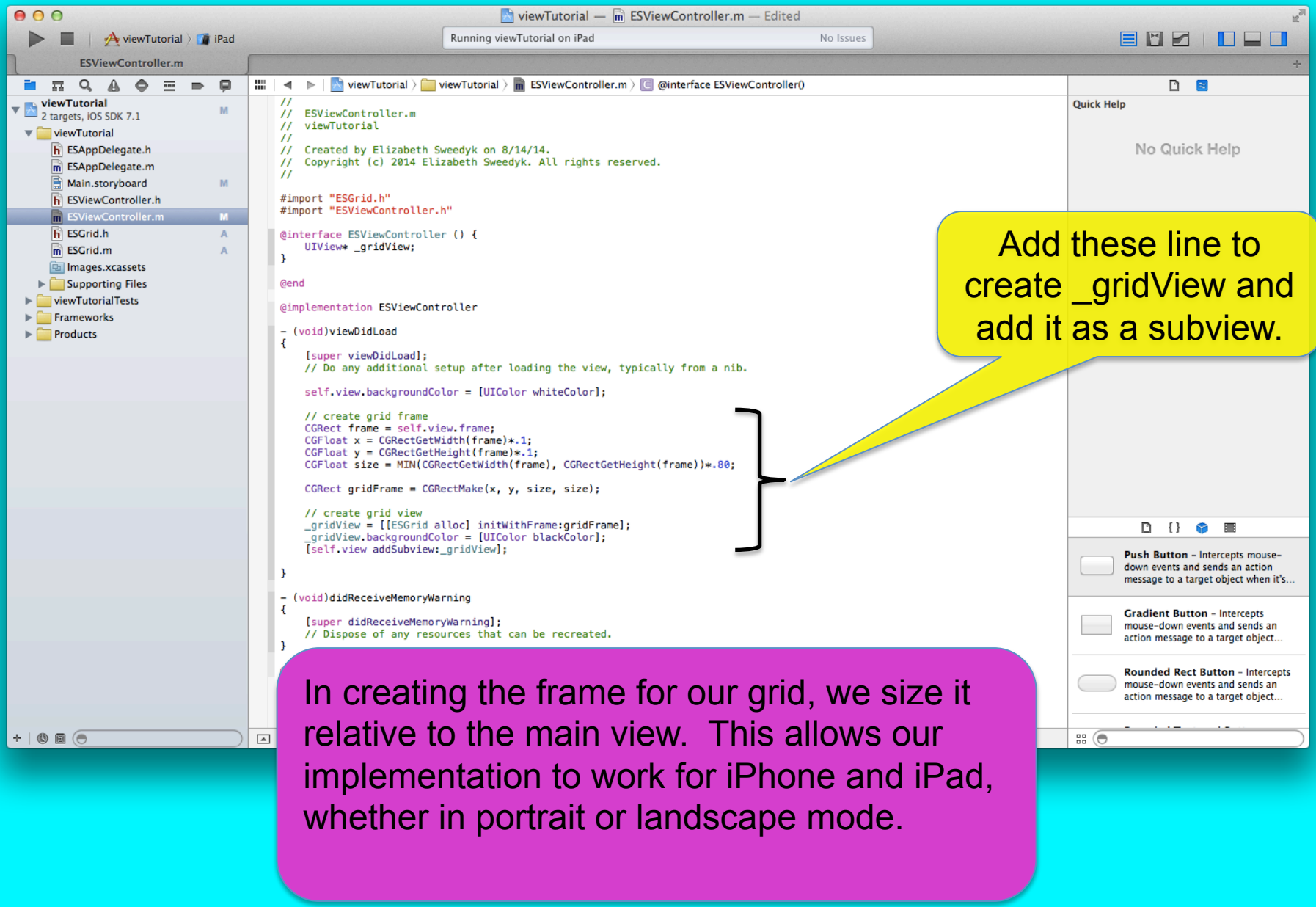

Create a new class Grid that inherits from UIView. (Yours of course will have your own prefix.)
viewTutorial — ESViewController.m
viewTutorial › iPad
Running viewTutorial.app on iPad
No Issues
Choose options for your new file:
Class
ESGrid
Subclass of
UIView
Also create XIB file
iPhone
Cancel
Previous
Next
Quick Help
No Quick Help
Supporting Files
viewTutorialTests
Frameworks
Products
View Controller – A controller that supports the fundamental view-management model in iPhone OS.
Table View Controller – A controller that manages a table view.
Collection View Controller – A controller that manages a collection view.
viewTutorial.app

1. Open the ViewController source file.

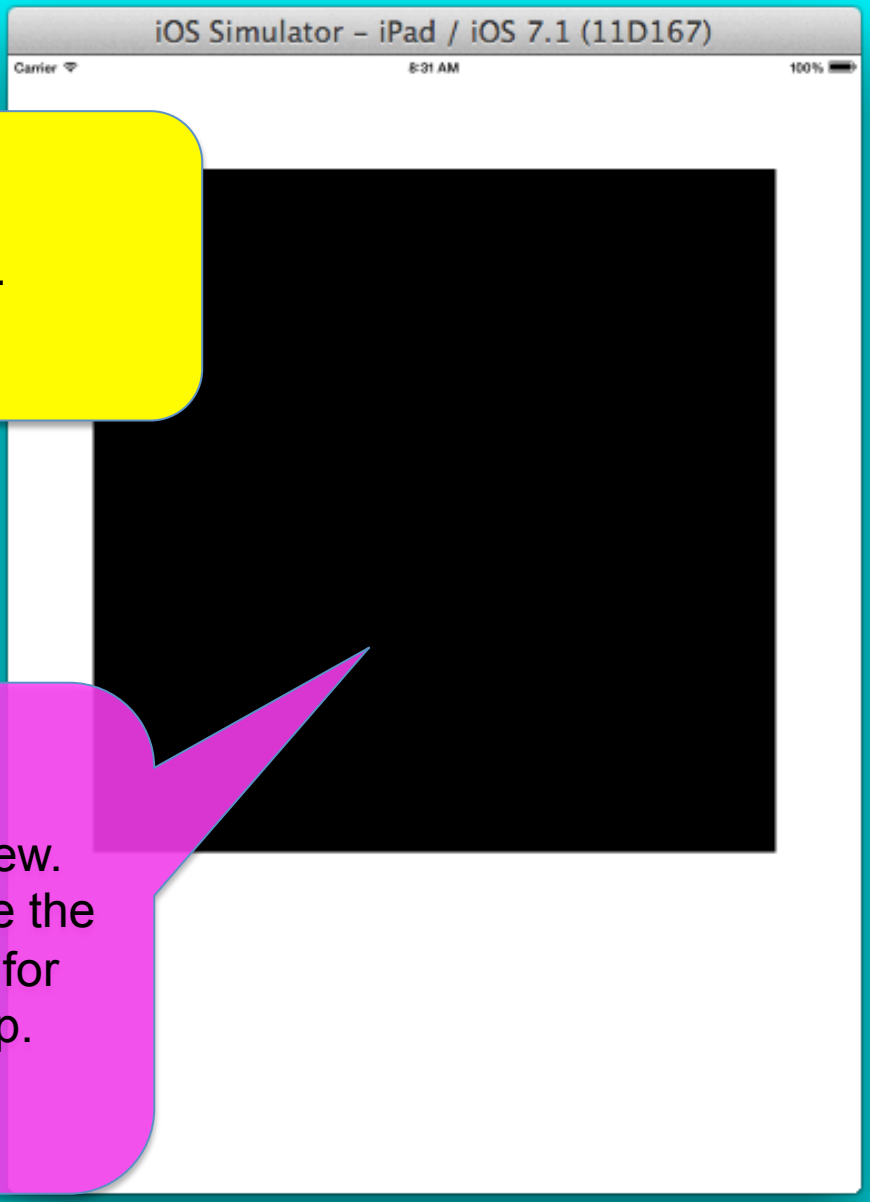
2. Import Grid.h.



3. Create an instance variable to store our view.

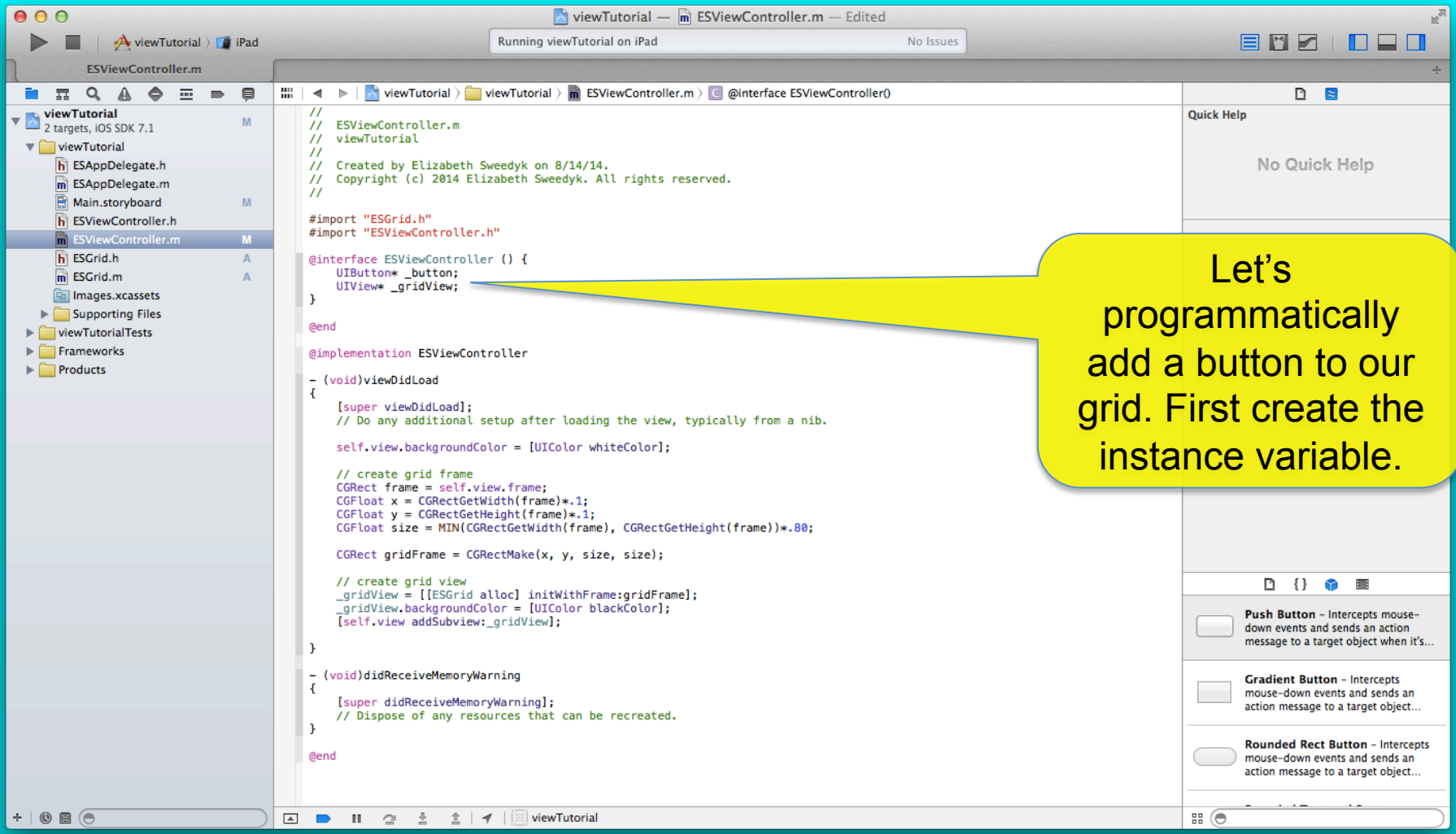



```
//
//  ESViewController.m
//  viewTutorial
//
//  Created by Elizabeth Sweedyk on 8/14/14.
//  Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//

#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.
}
```

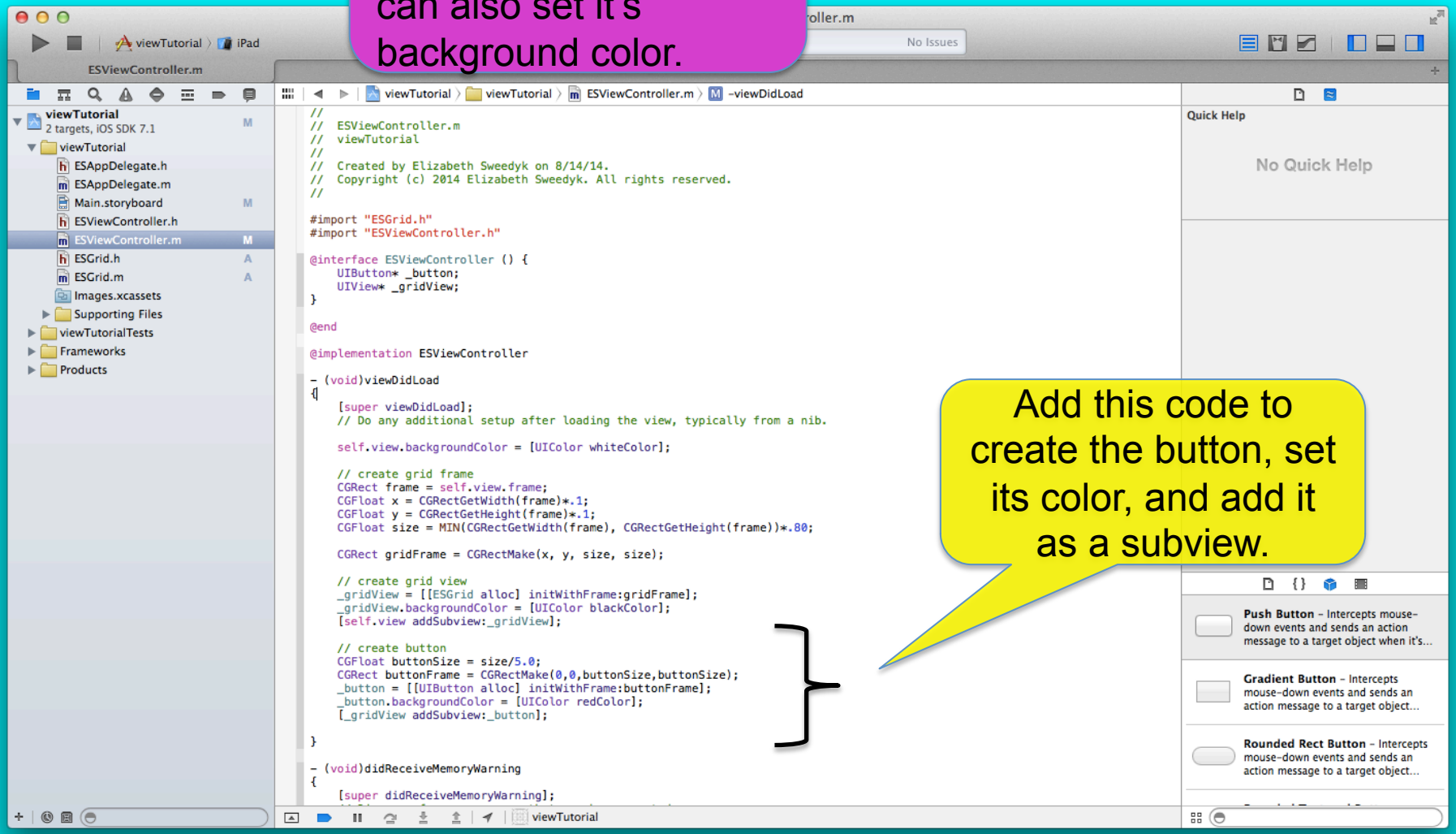


Run the app.

This is our grid view. You'll need to refine the location and size for your Sudoku app.


viewTutorial — ESViewController.m — Edited
Running viewTutorial on iPad
No Issues
viewTutorial
iPad
ESViewController.m
viewTutorial
2 targets, iOS SDK 7.1
ESAppDelegate.h
ESAppDelegate.m
Main.storyboard
ESViewController.h
ESViewController.m
ESGrid.h
ESGrid.m
Images.xcassets
Supporting Files
viewTutorialTests
Frameworks
Products
@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
Let's programmatically add a button to our grid. First create the instance variable.
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...
viewTutorial

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

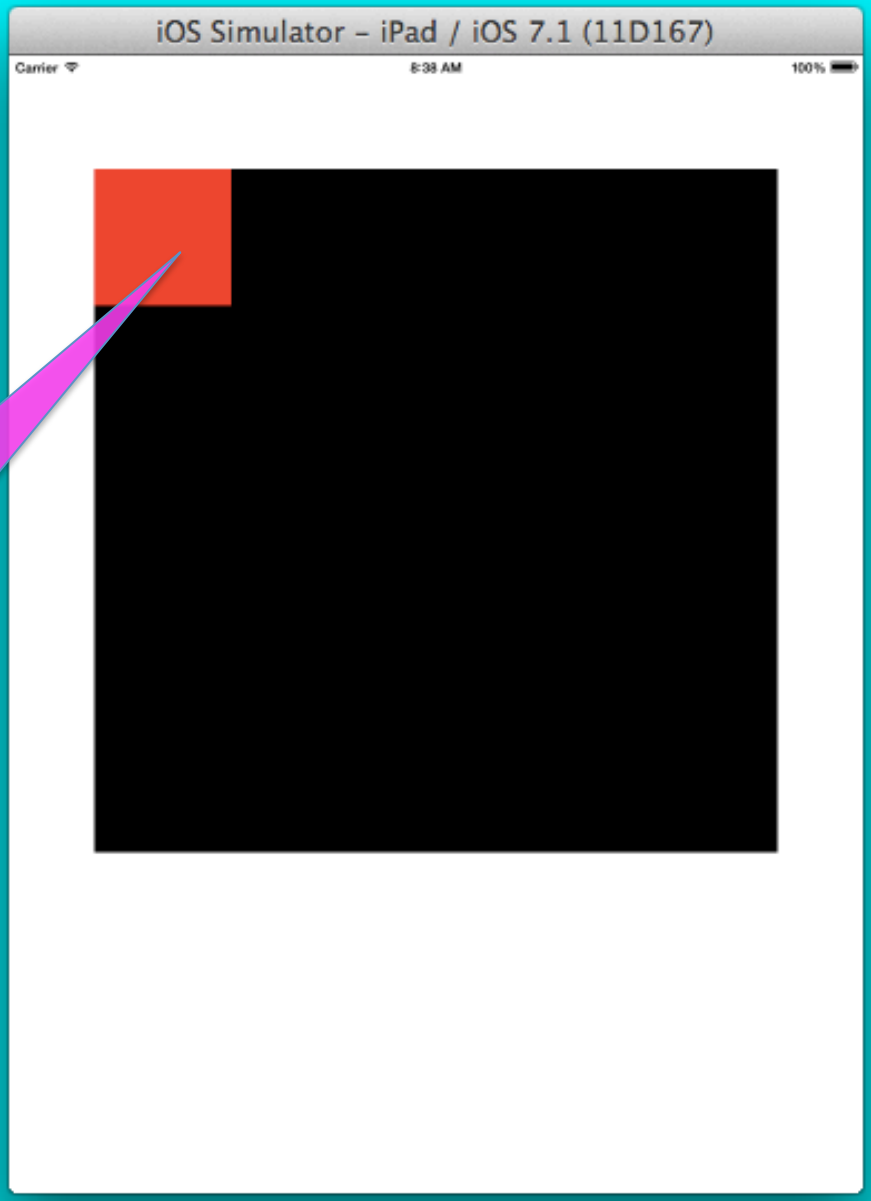



```
//
//  ESViewController.m
//  viewTutorial
//
//  Created by Elizabeth Sweedyk on 8/14/14.
//  Copyright (c) 2014 Elizabeth Sweedyk. All rights reserved.
//

#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];
```

Add this code to create the button, set its color, and add it as a subview.

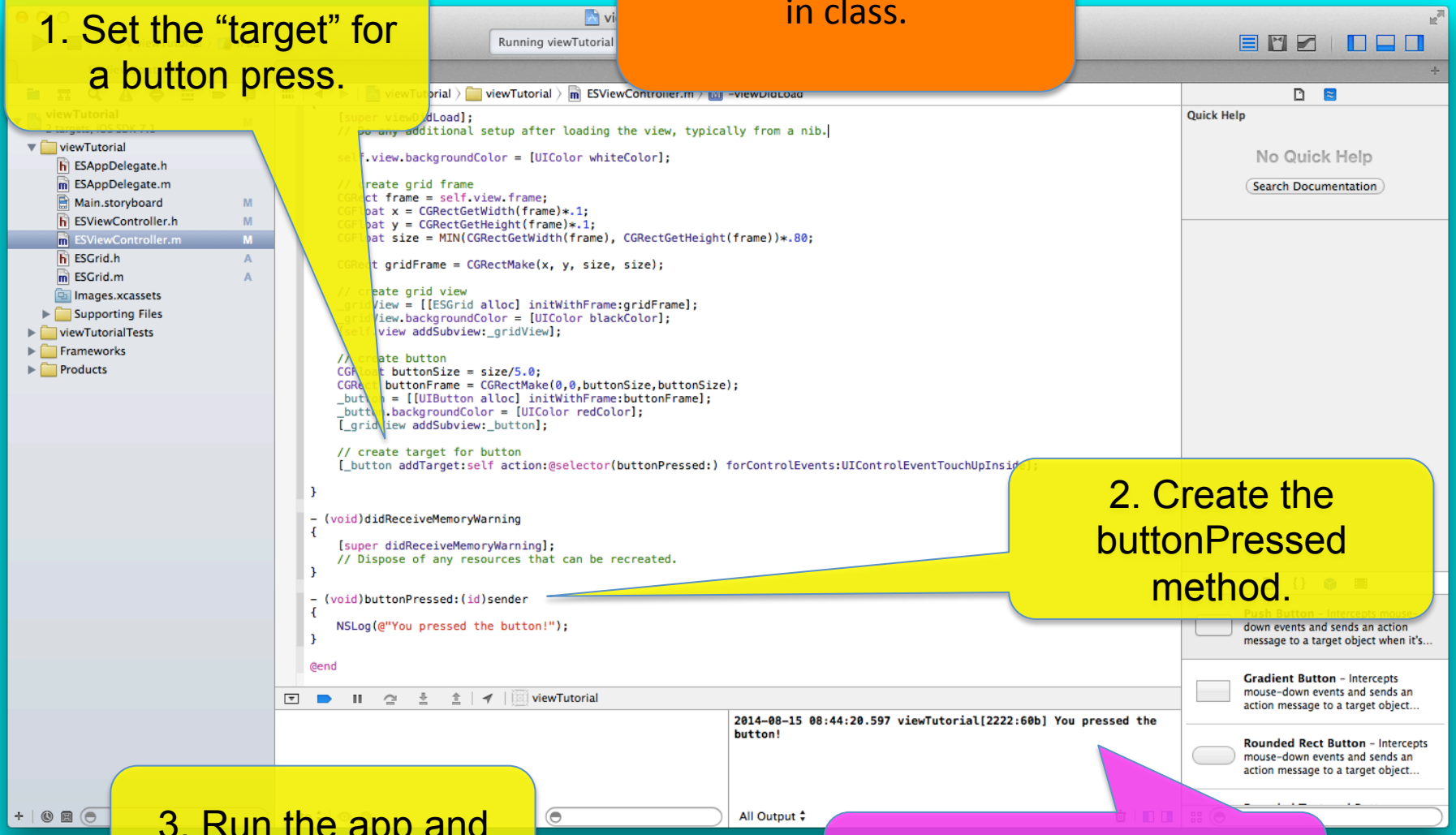
Run the app.

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.



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