

Who's Your Valentine?

Featuring contacts, shaking, and a little bit of love!

Launch Xcode



1.

New ►

Add Files... ⌘⌘A

Open... ⌘O

Open Recent ►

Open Quickly...

Close Window

Close Tab

Close Document ⌘⌘W

Close Workspace ⌘⌘W

Save ⌘S

Duplicate... ⌘⇧S

Revert to Saved...

Unlock...

Export...

Show in Finder

Open with External Editor

2.

Tab ⌘T

Window ⌘⌃T

File... ⌘N

Playground... ⌘⇧⌘N

Target...

Project... ⌘⌘N

Workspace... ⌘⌘N

Group ⌘⌘N

Group from Selection

By the end of the demo, our app will look through our phone's contacts (with permission of course!) and choose one lucky person to be our valentine.

Print...

⌘P

Choose a template for your new project:

iOS

Application

Framework & Library

Other

OS X

Application

Framework & Library

System Plug-in

Other

1.

Master-Detail Application

Game

2.

Page-Based Application

Single View Application

1

3.

Tabbed Application

Single View Application

This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a storyboard or nib file that contains the view.

Cancel

P

Next

Choose options for your new project:

Product Name: *

Organization Name:

Organization Identifier:

Bundle Identifier: *

Language: ▼

Devices: ▼

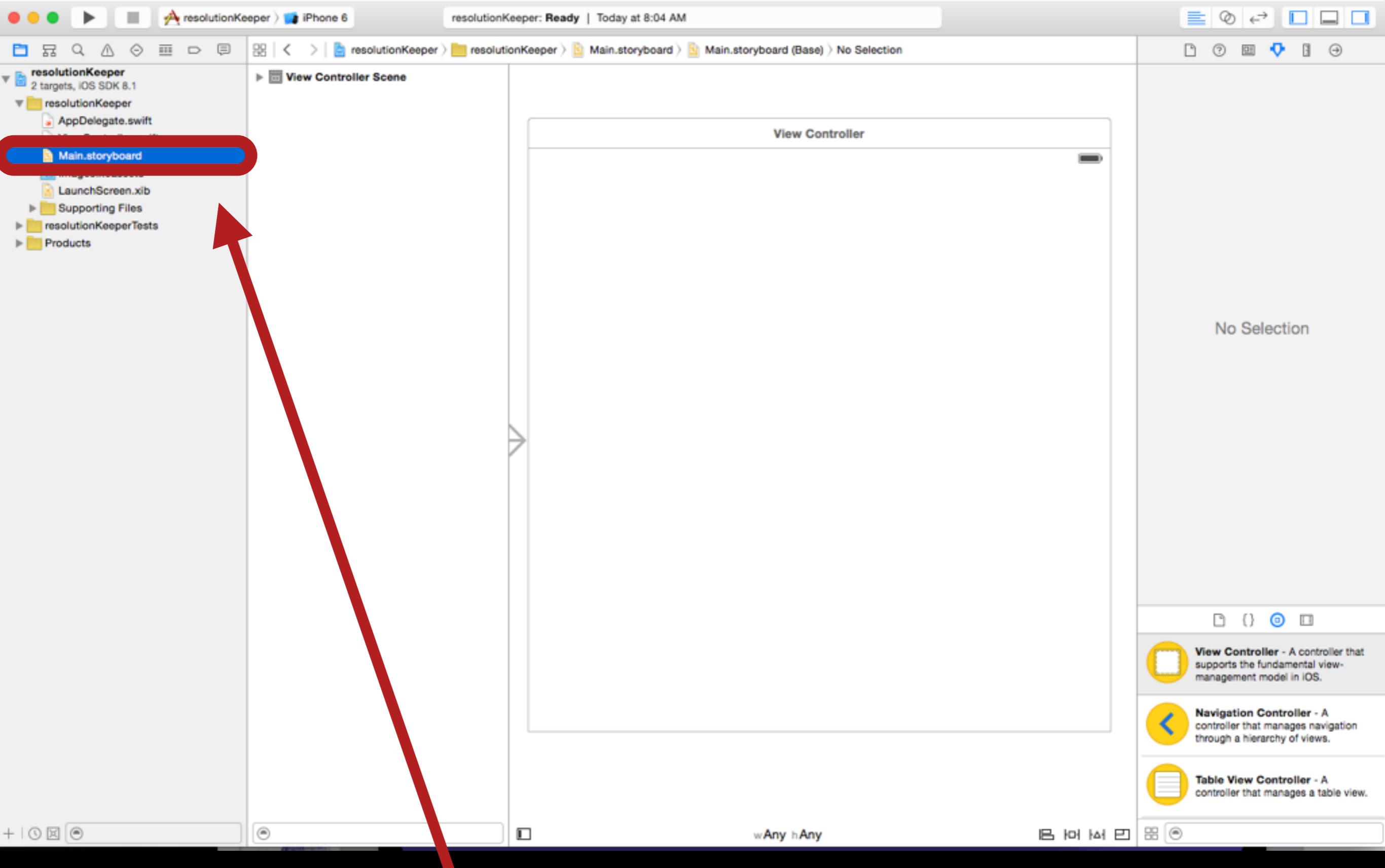
Use Core Data

Cancel

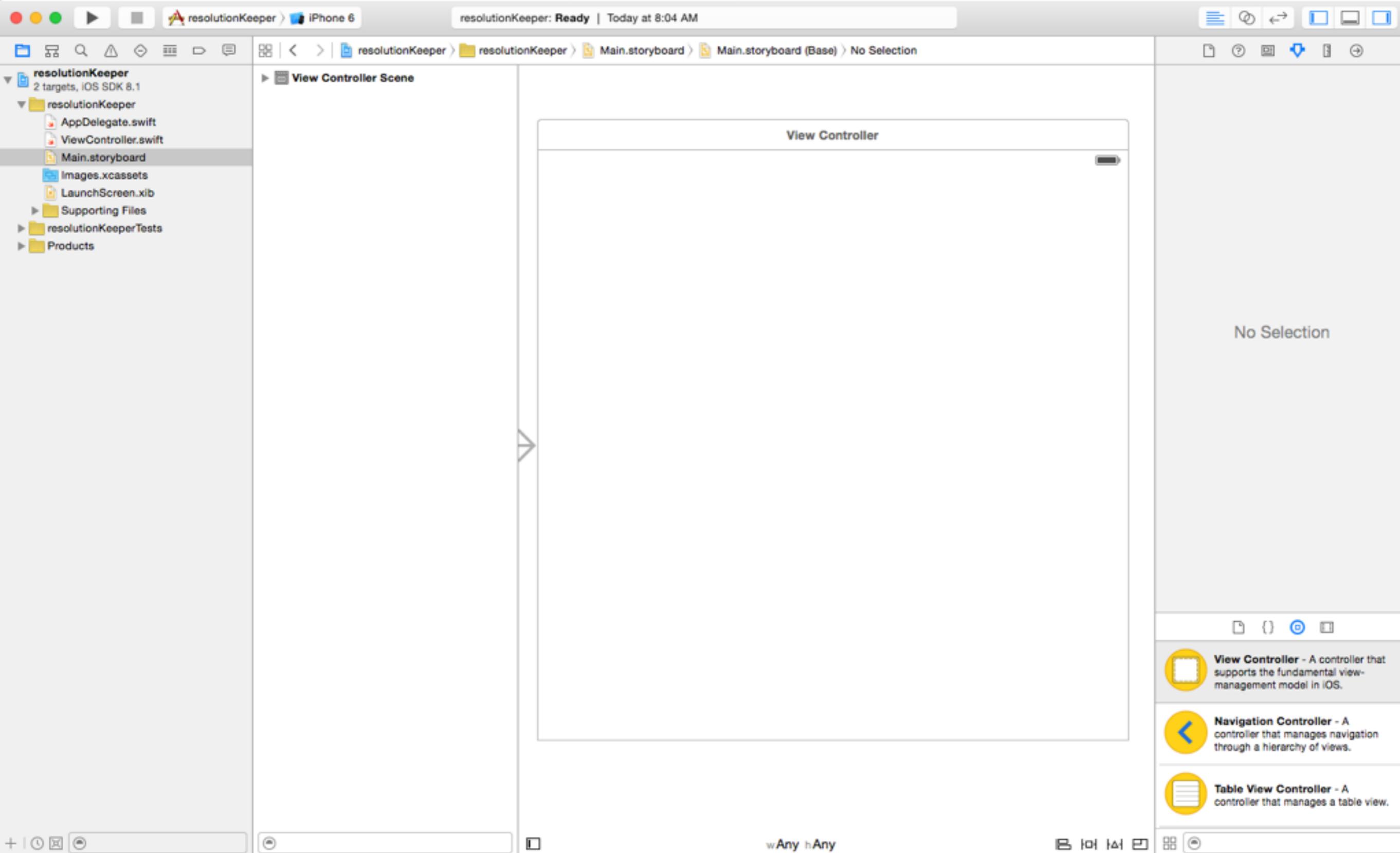
Previous

Next

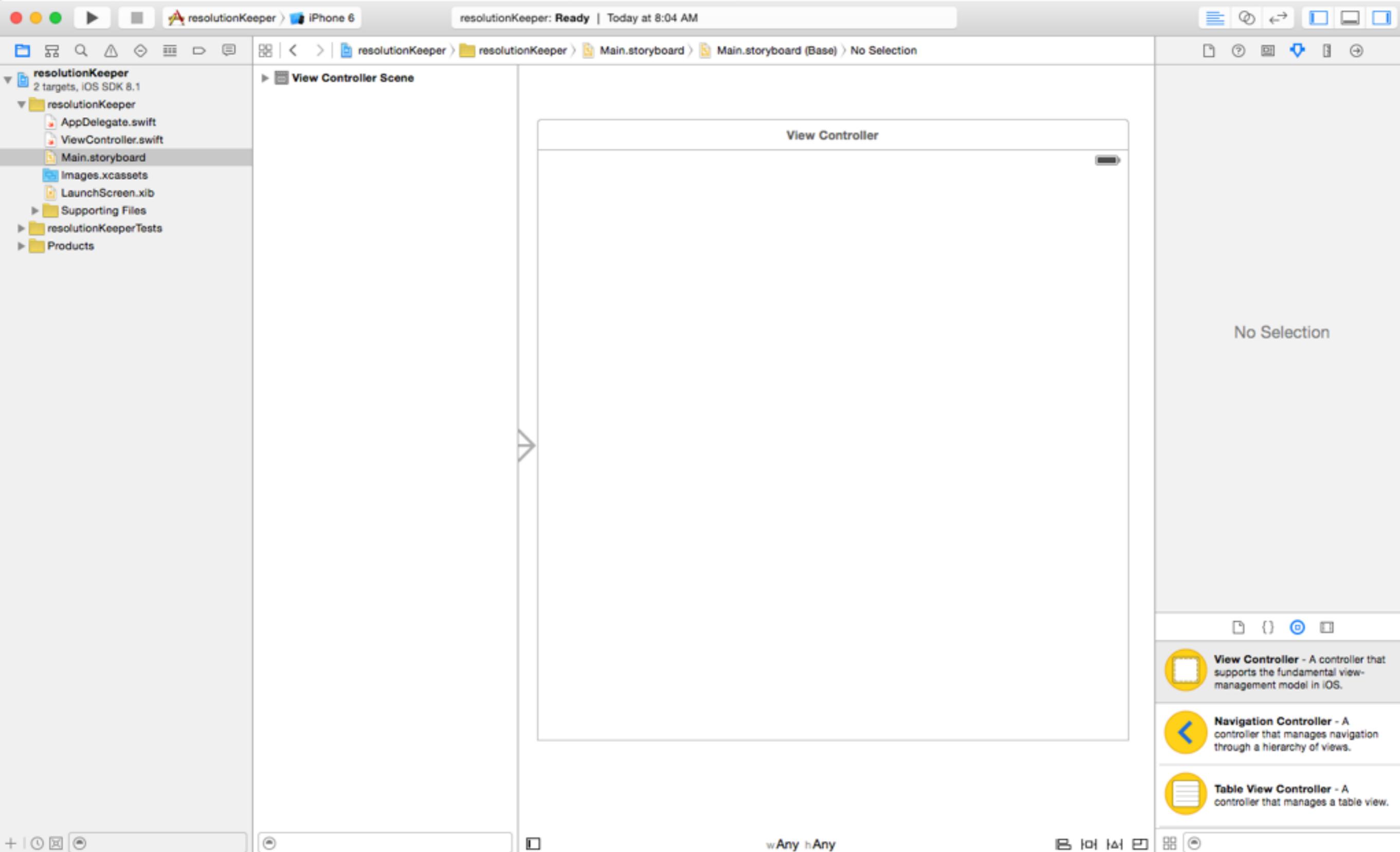
Follow the steps to create your project.



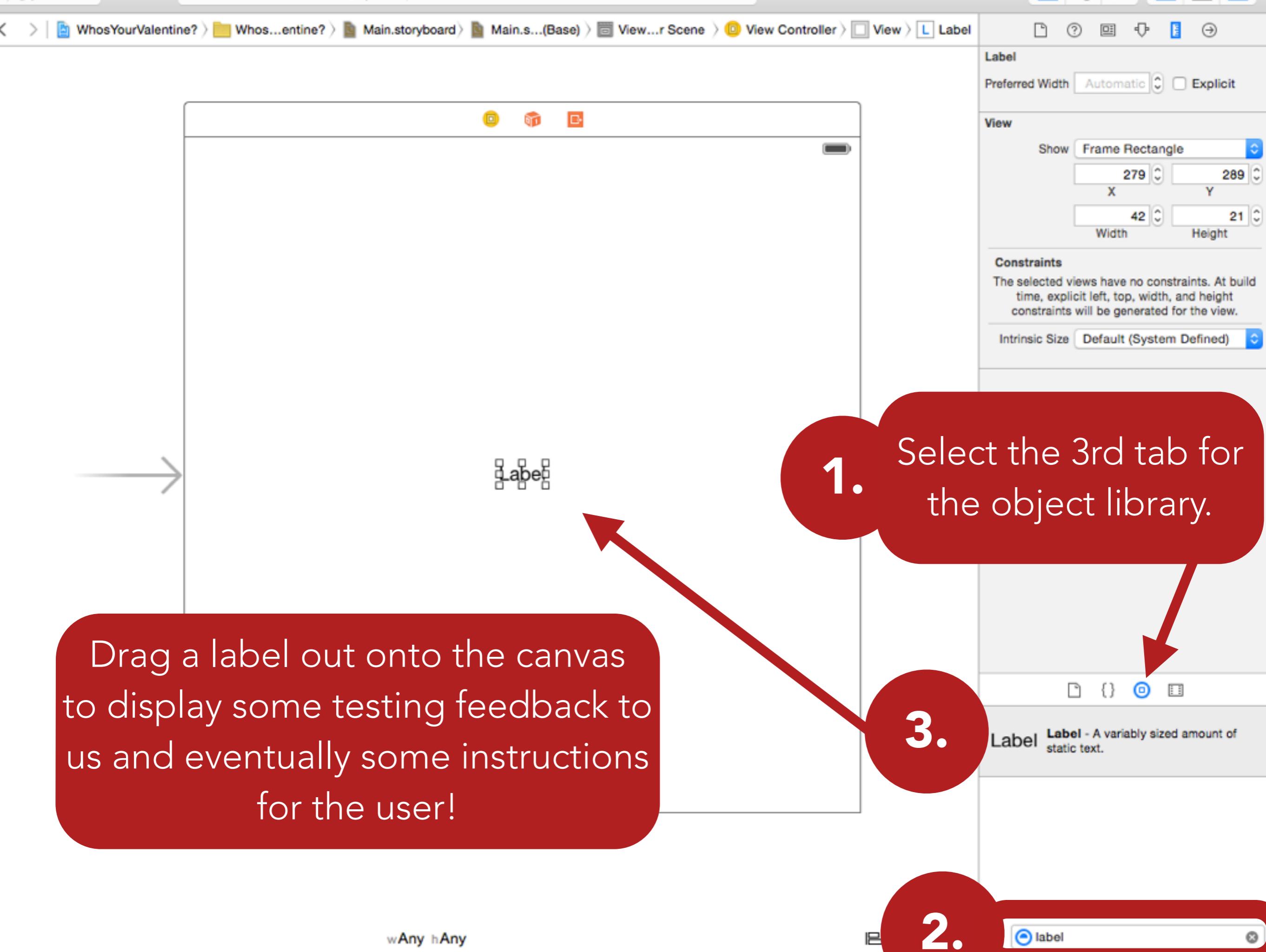
Now that the project has been started, head over to the Main Storyboard to start building the user interface.



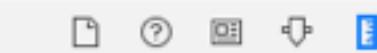
The Storyboard - where the user interface is created.



Notice that the View Controllers are by default set to an odd squarish shape. This is so you as the developer don't design only for one screen size.



WhosYourValentine? > Whos...entine? > Main.storyboard > Main.s...(Base) > View...r Scene > View Controller > View > Label



Center the label in the middle of the screen by adding a constraint for both horizontal and vertical center in container.



Label

Add New Alignment Constraints

- Leading Edges
- Trailing Edges
- Top Edges
- Bottom Edges
- Horizontal Centers
- Vertical Centers
- Baselines

2.

- Horizontal Center in Container
- Vertical Center in Container

1.



3.

Label - A variably sized static text.

The screenshot shows the Xcode interface with the following details:

- Project Navigator (Left):** Shows the project structure for "WhosYourValentine?". The "ViewController.swift" file is selected and highlighted with a red oval.
- Main Editor (Top Center):** Displays the Swift code for "ViewController.swift". The code includes comments about the file's creation and copyright, imports UIKit, and defines a ViewController class with viewDidLoad and didReceiveMemoryWarning methods.
- Assistant Editor (Right):** Shows the "Label" component settings. It includes sections for "Identity and Type", "Target Membership", "Text Settings" (with encoding set to Default - Unicode (UTF-8)), "Source Control", and a detailed description of the Label component.

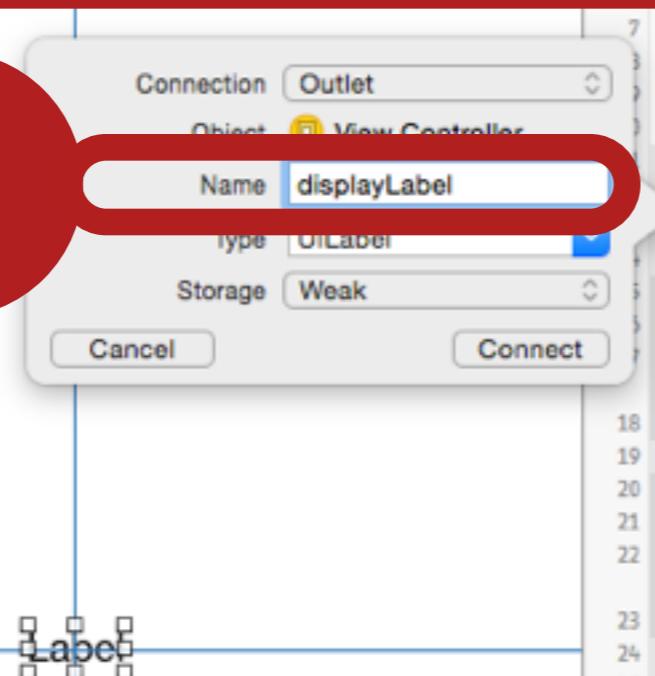
Now option click on ViewController.swift in the left-side panel to open Assistant Editor.

1.

Once in assistant editor mode control drag from our “displayLabel” uillabel into the space just above the “viewDidLoad” code.

Change the name to “`displayLabel`” and then hit connect.

2



```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the
        // view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be
        // recreated.
    }
}
```

We are creating an outlet that lets our code know information about our label and update it's contents.



ViewCo...ller.swift

motionEnded(_:,withEvent:)



```
1 //ViewController.swift
2 //WhosYourValentine?
3 //
4 //Created by Bliss Chapman on 2/14/15.
5 //Copyright (c) 2015 Bliss Chapman. All rights
6 //reserved.
7
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var displayLabel: UILabel!
14
15
16     override func viewDidLoad() {
17         super.viewDidLoad()
18         // Do any additional setup after loading the
19         // view, typically from a nib.
20     }
21
22     override func didReceiveMemoryWarning() {
23         super.didReceiveMemoryWarning()
24         // Dispose of any resources that can be
25         // recreated.
26     }
27
28     override func motionEnded(motion: UIEventSubtype,
29                               withEvent event: UIEvent) {
30
31         println("motion detected!!!!")
32     }
33 }
```

Identity and Type

Name ViewController.swift

Type Default - Swift Source

Location Relative to Group
ViewController.swift

Full Path /Users/blisschapman/Developer/WhosYourValentine?/WhosYourValentine?/ViewController.swift

Target Membership

WhosYourValentine?

WhosYourValentine?Tests

Text Settings

Text Encoding Default - Unicode (UTF-8)

Line Endings Default - OS X / Unix (LF)

Indent Using Spaces

Widths 4 Tab 4 Indent

Wrap lines

Source Control

Repository --

Type --

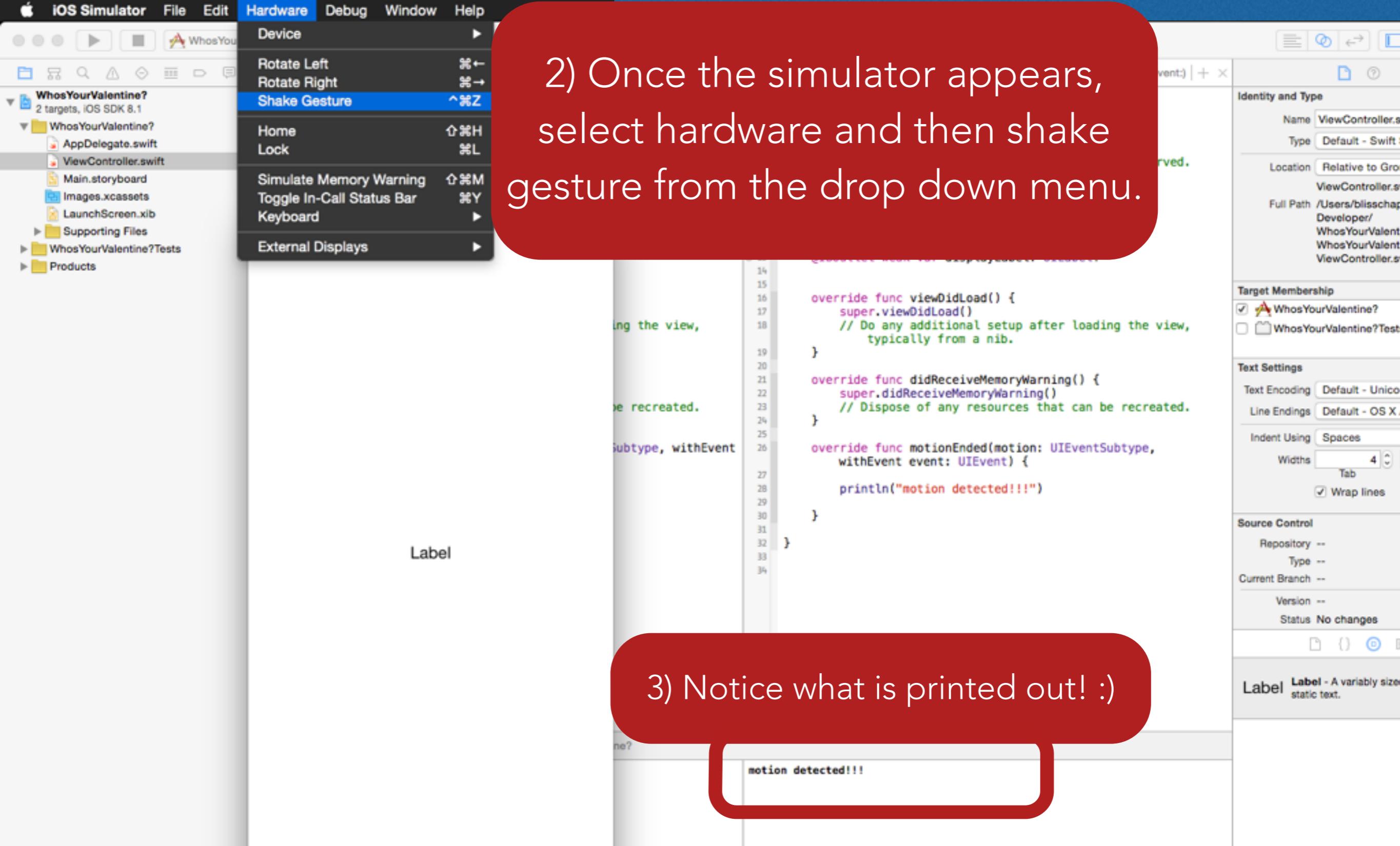
Current Branch --

Version --

Status No changes

1.

Add a function to detect if a motion event occurs.



1) Build and Run by selecting the triangle button in the upper left.

2) Once the simulator appears, select hardware and then shake gesture from the drop down menu.

3) Notice what is printed out! :)

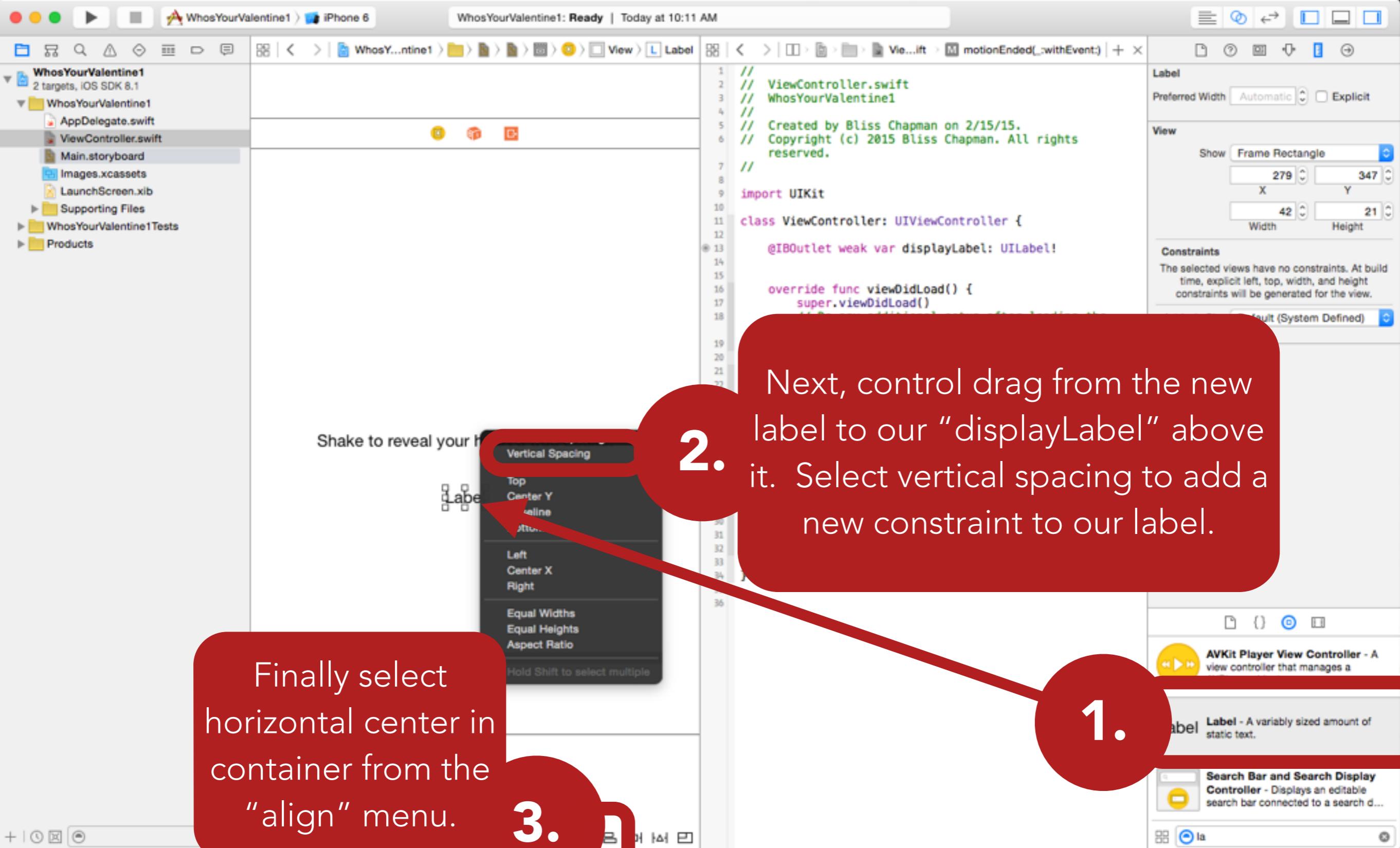
The screenshot shows the Xcode interface with the project "WhosYourValentine1" selected. The storyboard file "Main.storyboard" is open, displaying a single view controller with a label containing the text "Shake to reveal your heart's true desire.". A red circle with the number "1." is overlaid on the storyboard area, pointing to the label. The right side of the screen shows the "ViewController.swift" code:

```
1 // ViewController.swift
2 // WhosYourValentine1
3 // Created by Bliss Chapman on 2/15/15.
4 // Copyright (c) 2015 Bliss Chapman. All rights reserved.
5
6 import UIKit
7
8 class ViewController: UIViewController {
9
10    @IBOutlet weak var displayLabel: UILabel!
11
12    override func viewDidLoad() {
13        super.viewDidLoad()
14        // Do any additional setup after loading the view, typically from a nib.
15    }
16
17    override func didReceiveMemoryWarning() {
18        super.didReceiveMemoryWarning()
19        // Dispose of any resources that can be recreated.
20    }
21
22    override func motionEnded(motion: UIEventSubtype, withEvent event: UIEvent) {
23
24        println("motion detected!!!")
25    }
26
27
28 }
```

The Xcode interface includes the following panels on the right:

- Label**: Preferred Width: Automatic, Explicit checkbox.
- View**: Show: Frame Rectangle, X: 148, Y: 289, Width: 305, Height: 21.
- Constraints**: Shows a grid with a constraint labeled "Ag".
- Content Hugging Priority**: Horizontal: 251, Vertical: 251.
- Content Compression Resistance Priority**: Horizontal: 750, Vertical: 750.
- AVKit Player View Controller**: A view controller that manages an AVPlayer object.
- Label**: A variably sized amount of static text.
- Search Bar and Search Display Controller**: Displays an editable search bar connected to a search d...

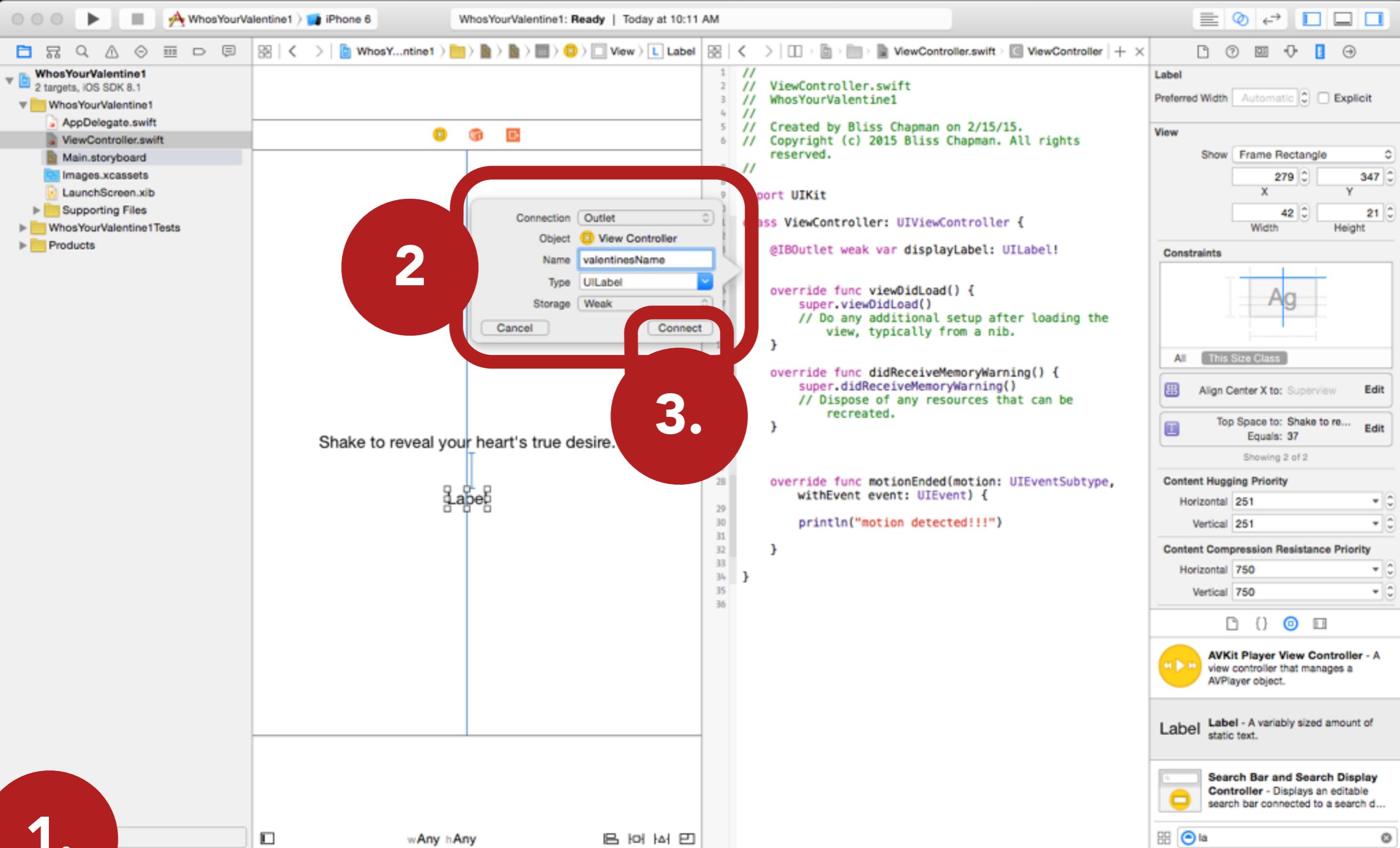
Now, edit the text of our label by double-clicking on it to give the user instructions.



Add a new label to the canvas by dragging it out from the object library.

Finally select horizontal center in container from the "align" menu.

Next, control drag from the new label to our "displayLabel" above it. Select vertical spacing to add a new constraint to our label.



Connect the label to our code via an outlet by control-dragging from the label to the space underneath `displayLabel`.

The screenshot shows the Xcode interface with the project "WhosYourValentine1" selected. The storyboard "Main.storyboard" is open, displaying a single view controller with a label placeholder. The code editor shows "ViewController.swift" with the following content:

```
// ViewController.swift
// WhosYourValentine1
//
// Created by Bliss Chapman on 2/15/15.
// Copyright (c) 2015 Bliss Chapman. All rights reserved.

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var valentinesName: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    override func motionEnded(motion: UIEventSubtype, withEvent event: UIEvent) {
        println("motion detected!!!")
    }
}
```

A red circle with the number "1." is overlaid on the storyboard, pointing to the label placeholder. A red oval highlights the line of code containing the IBOutlet declaration. The Xcode interface includes the file browser, navigation bar, and various inspector panels on the right.

Now delete the default text of our label by double clicking and deleting. Eventually, this is where our valentines name will appear.

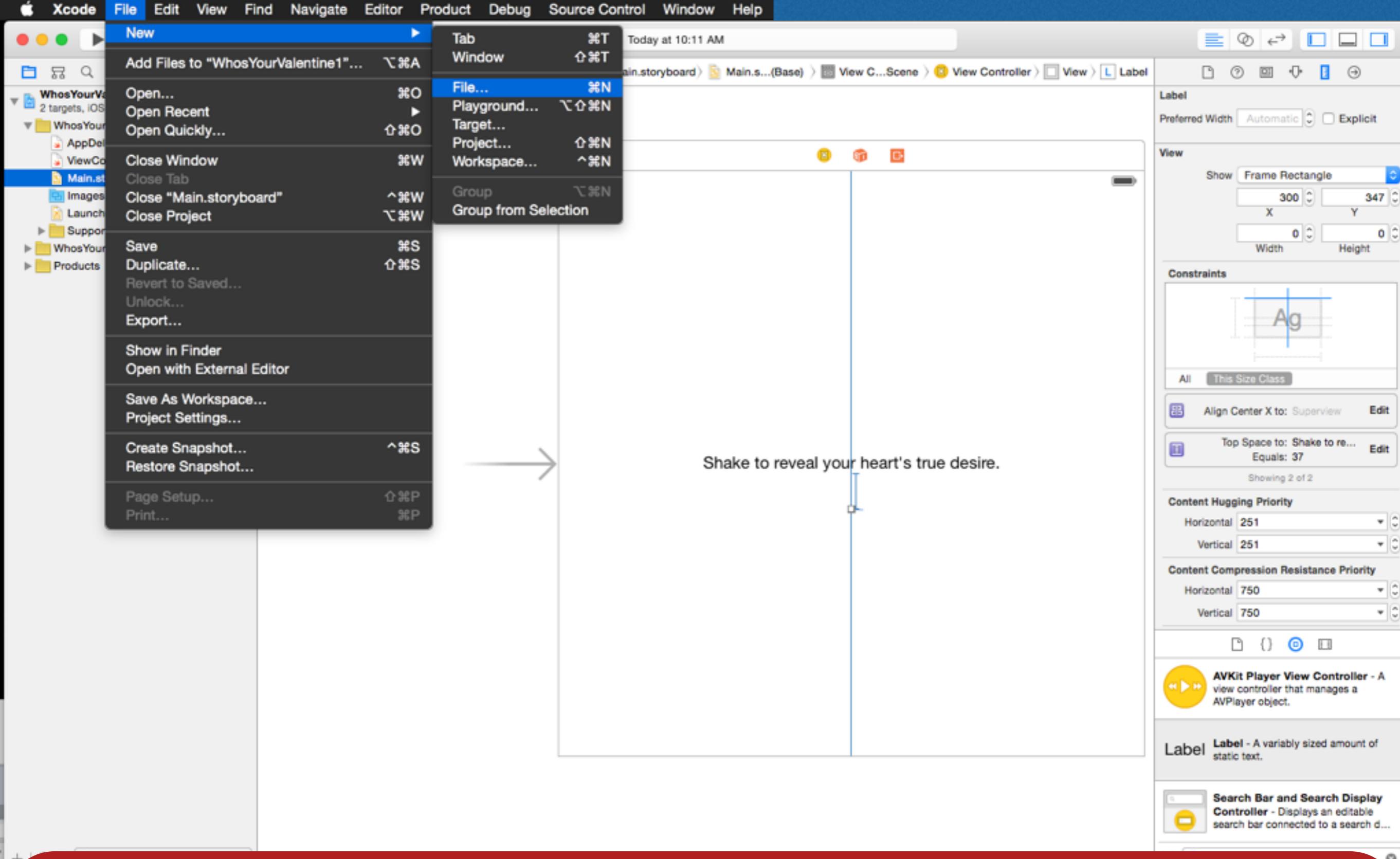
We are going to add a new file to our class which will serve as the “model” part of our Model-View-Controller paradigm. Recall that by the end of the demo, our app will look through our phone’s contacts (with permission of course!) and choose one lucky person to be our valentine.

We will write our model to help us deal with accessing the phone’s contacts. Then we will leverage this model in our view controller to display the results in our view.

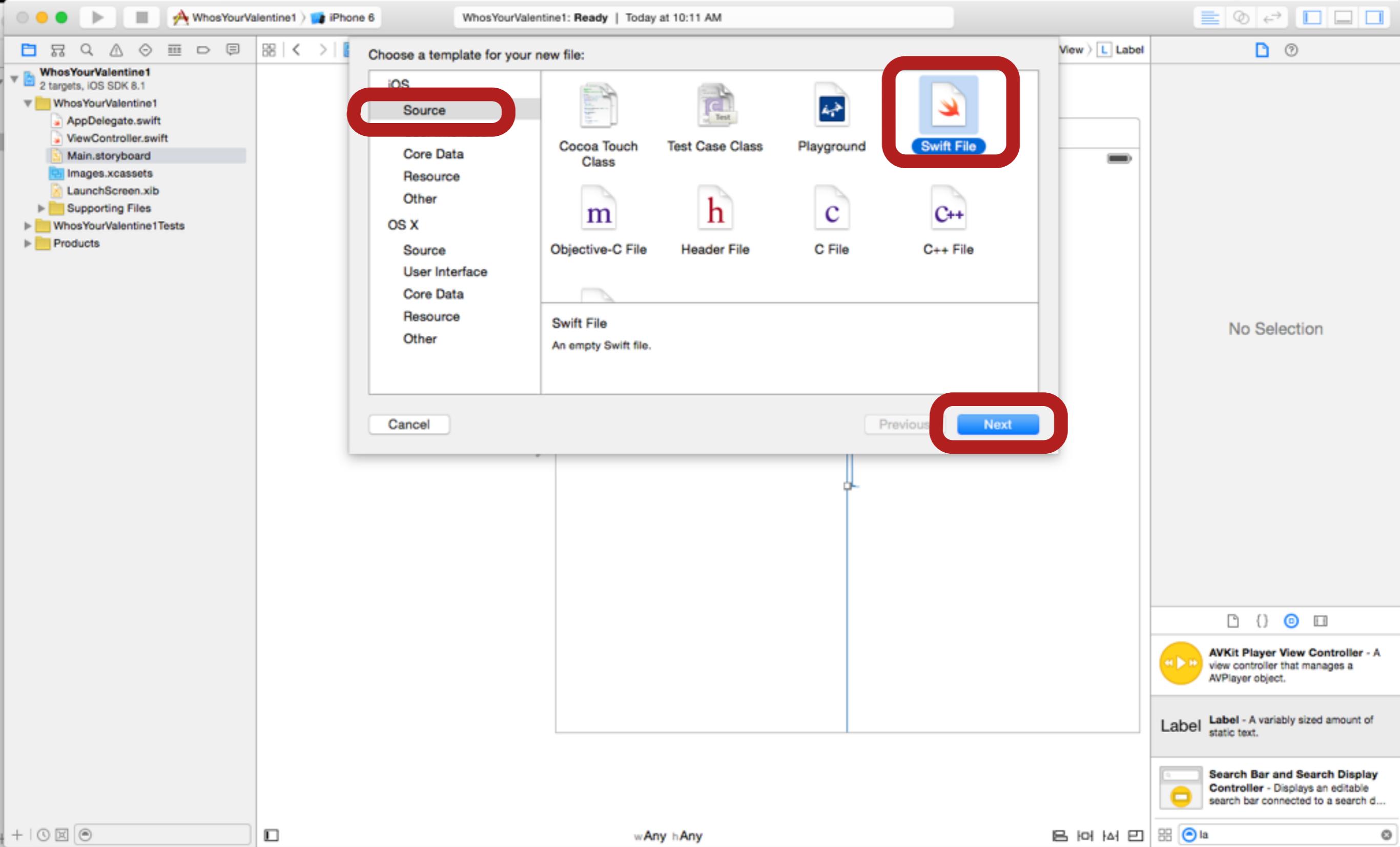
The storyboard interface includes:

- Label**: Preferred Width Automatic, Explicit checked.
- View**: Frame Rectangle (X: 300, Y: 347, Width: 0, Height: 0).
- Constraints**:
 - Align Center X to: Superview
 - Top Space to: Shake to re... Equals: 37
- Content Hugging Priority**: Horizontal 251, Vertical 251.
- Content Compression Resistance Priority**: Horizontal 750, Vertical 750.
- AVKit Player View Controller**: A view controller that manages an AVPlayer object.
- Label**: Label - A variably sized amount of static text.
- Search Bar and Search Display Controller**: Displays an editable search bar connected to a search d...

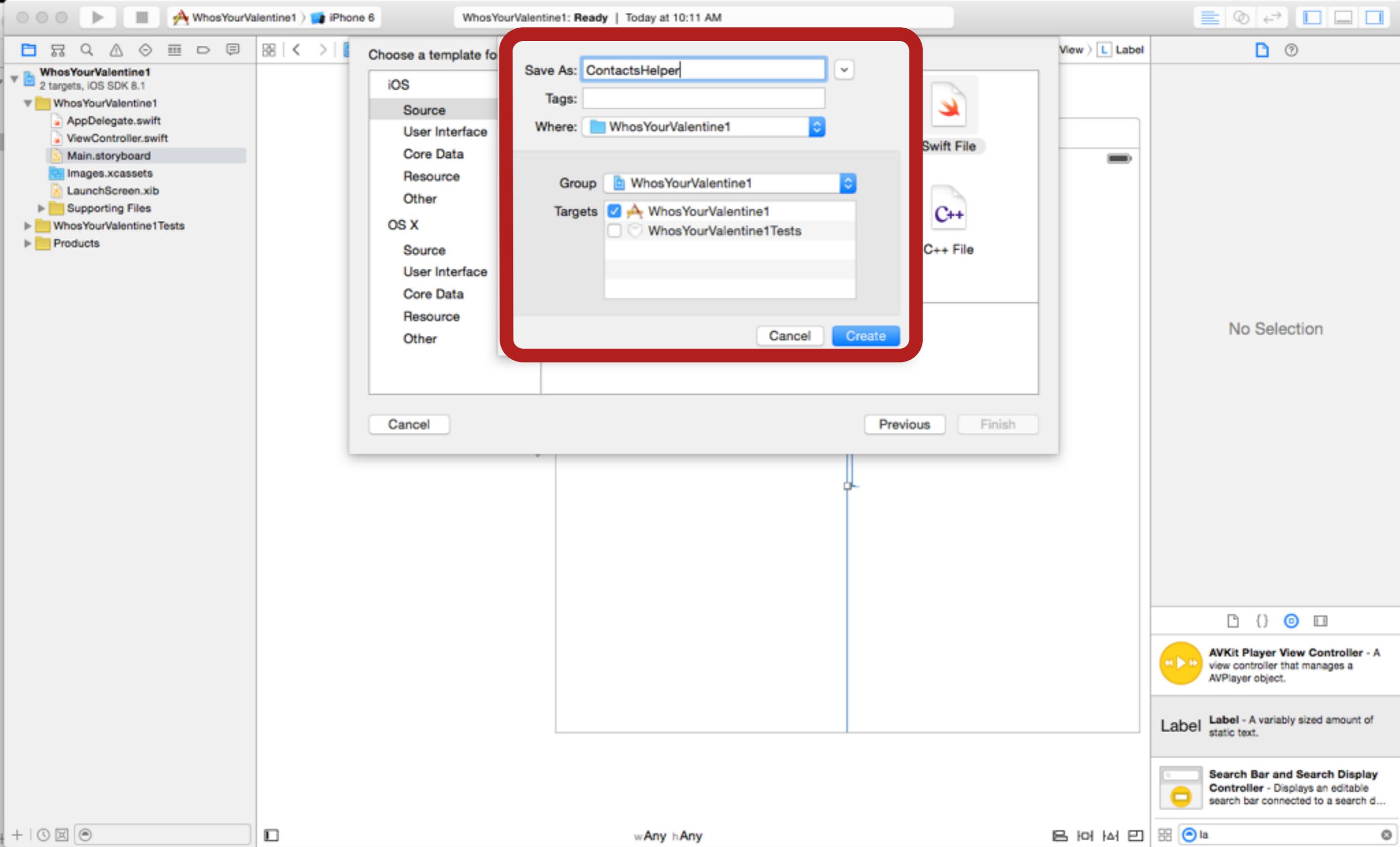
Open standard editor mode.



Create a new file.



Select Swift File from the iOS-Source menu. This is simply a blank file.



Name the new file “ContactsHelper” and then hit create.

WhosYourValentine1 > iPhone 6 WhosYourValentine1: Ready | Today at 10:11 AM

WhosYourValentine1 > WhosYourValentine1 > ContactsHelper.swift > No Selection

Identity and Type

Name: ContactsHelper.swift
Type: Default - Swift Source
Location: Relative to Group
Full Path: /Users/blisschapman/Developer/WhosYourValentine1/WhosYourValentine1/ContactsHelper.swift

Target Membership

✓ WhosYourValentine1
WhosYourValentine1Tests

Text Settings

Text Encoding: Unicode (UTF-8)
Line Endings: Default - OS X / Unix (LF)
Indent Using: Spaces
Widths: Tab 4 Indent 4
Wrap lines

Source Control

Repository: --
Type: --
Current Branch: --
Version: --
Status: No changes

AVKit Player View Controller - A view controller that manages a AVPlayer object.

Label - Label - A variably sized amount of static text.

Search Bar and Search Display Controller - Displays an editable search bar connected to a search d...

+ | 0 | 0 | 0 | 0

```
1 //  
2 // ContactsHelper.swift  
3 // WhosYourValentine1  
4 //  
5 // Created by Bliss Chapman on 2/15/15.  
6 // Copyright (c) 2015 Bliss Chapman. All rights reserved.  
7 //  
8  
9 import Foundation  
10  
11
```

Now we need to create a new class with all the methods needed to ask permission for access to the contacts and creating a list of all their names.

The screenshot shows the Xcode interface with the project 'WhosYourValentine1' selected. In the left sidebar, 'ContactsHelper.swift' is highlighted. The main editor area displays the following Swift code:

```
// ContactsHelper.swift
// WhosYourValentine1
//
// Created by Bliss Chapman on 2/15/15.
// Copyright (c) 2015 Bliss Chapman. All rights reserved.

import Foundation
import AddressBook

class ContactsHelper {
    init() {}

    var usersContactsNames: [String]?
    var addressBook: ABAddressBook!
}
```

First, import the AddressBook library which is required when dealing with contacts.

Second, create a new class called ContactsHelper. Be sure to add the init() function so that we can construct a new instance of this class in the future. Since we don't need any properties initialized with this class, we can leave it blank.

Third, add two properties to our class. One is an optional array of strings that we will use to hold the names of all the contacts. The other is an addressBook that is returned from one of the functions we are about to write.

Create a new class with all the methods needed to ask permission for access to the contacts and create a list of all their names.

WhosYourValentine1 > iPhone 6 WhosYourValentine1: Ready | Today at 10:39 AM

WhosYourValentine1 2 targets, iOS SDK 8.1

WhosYourValentine1

- AppDelegate.swift
- ViewController.swift
- ContactsHelper.swift
- Main.storyboard
- Images.xcassets
- LaunchScreen.xib

Supporting Files

WhosYourValentine1Tests

Products

```
1 // ContactsHelper.swift
2 // WhosYourValentine1
3 //
4 // Created by Bliss Chapman on 2/15/15.
5 // Copyright (c) 2015 Bliss Chapman. All rights reserved.
6 //
7
8 import Foundation
9 import AddressBook
10
11
12 class ContactsHelper {
13     init() {
14     }
15
16     var usersContactsNames: [String]?
17     var addressBook: ABAddressBook!
18 }
19
20
21 }
```

Don't worry if you are not sure what is going on,
its ok 😜 Please ask lots and lots of questions
because they help us ALL learn something.

Also, feel free to work with the person next to
you, I guarantee they don't bite 😊

Now, let's keep moving!!

```
1 //  
2 // ContactsHelper.swift  
3 // WhosYourValentine1  
4 //  
5 // Created by Bliss Chapman on 2/15/15.  
6 // Copyright (c) 2015 Bliss Chapman. All rights reserved.  
7 //  
8  
9 import Foundation  
10 import AddressBook  
11  
12  
13 class ContactsHelper {  
14     init() {}  
15  
16     var usersContactsNames: [String]?  
17     var addressBook: ABAddressBook?  
18  
19     //Determine our application's permission status  
20     func determinePermissionStatus() -> Bool {  
21  
22         let contactsPermissionStatus = ABAddressBookGetAuthorizationStatus()  
23         switch contactsPermissionStatus {  
24             case .Authorized:  
25             case .NotDetermined:  
26             case .Restricted:  
27             case .Denied:  
28         }  
29     }  
30 }
```

Identity and Type

Name ContactsH

Type Default -

Location Relative t

..../Contact

Full Path /Users/blis

Developer

WhosYour

ctsH

Our first function determines the permission status of our app regarding access to the user's contacts. Then, based on that permission we either continue to grab information or stop and wait for our user to change their privacy settings.

Note the four potential permission levels we must deal with.

AVKit Player V
view controller th
AVPlayer object.

Label Label - A variab
static text.

Search Bar and
Controller - Dis

```

8
9 import Foundation
10 import AddressBook
11
12
13 class ContactsHelper {
14
15     init() {
16 }
17
18     var usersContactsNames: [String]?
19     var addressBook: ABAddressBook!
20
21     //Determine our application's permission status
22     func determinePermissionStatus() -> Bool {
23
24         let contactsPermissionStatus = ABAddressBookGetAuthorizationStatus()
25
26         switch contactsPermissionStatus {
27             case .Authorized:
28                 return self.createAddressBook()
29             case .NotDetermined:
30                 var ok = false
31                 ABAddressBookRequestAccessWithCompletion(nil) {
32                     (granted:Bool, err:CFCError!) in
33                     dispatch_async(dispatch_get_main_queue()) {
34                         if granted {
35                             ok = self.createAddressBook()
36                         }
37                     }
38                 }
39                 if ok == true {
40                     return true
41                 }
42                 self.usersContactsNames = nil
43                 return false
44             case .Restricted:
45                 self.usersContactsNames = nil
46                 return false
47             case .Denied:
48                 self.usersContactsNames = nil
49                 return false
50         }
51     }
52
53 }
```

For `.Authorized`, we will call a function to create our address book.

For `.NotDetermined` we will request permission to use the users Contacts, and if the request is granted, then we will create our address book. Experienced developers, note the multithreading.

For `.Restricted` and `.Denied` we will simply set our list of contact's names to nil.

Fill in our different cases with the appropriate response.

```
29 case .NotDetermined:  
30     var ok = false  
31     ABAddressBookRequestAccessWithCompletion(nil) {  
32         (granted:Bool, err:CFError!) in  
33         dispatch_async(dispatch_get_main_queue()) {  
34             if granted {  
35                 ok = self.createAddressBook()  
36             }  
37         }  
38     }  
39     if ok == true {  
40         return true  
41     }  
42     self.usersContactsNames = nil  
43     return false  
44 case .Restricted:  
45     self.usersContactsNames = nil  
46     return false  
47 case .Denied:  
48     self.usersContactsNames = nil  
49     return false  
50 }  
51 }
```

```
53  
54 //Create address book  
55 func createAddressBook() -> Bool {  
56  
57  
58 }  
59  
60
```

Now, create the “createAddressBook” function that creates the address book if we have permission.

```
var ok = false
```

First, we check if our list of names has already been created. If it has, then we exit the function.

If it hasn't been created, we create an optional CFError (don't worry about "Unmanaged") and attempt to create a new addressBook.

If the addressBook we just created equals nil, something went wrong so we print out our error message and exit the function.

If it does exist, we assign it to our class property also called addressBook and then exit the function. Yay!

```
//Create address book
func createAddressBook() -> Bool {
    if usersContactsNames != nil {
        return true
    }
    var error: Unmanaged<CFError>?
    let addressBook: ABAddressBook? = ABAddressBookCreateWithOptions(nil, &error).takeRetainedValue()
    if addressBook == nil {
        println(error)
        self.addressBook = nil
        return false
    }
    self.addressBook = addressBook
    return true
}
```

Finally, we need a function that will take our just created addressBook and turn it into a string array of contact names.

First, we do a cautionary check to make sure we still have permission. If we don't, we exit the function. Then, if our addressBook does not equal nil which it shouldn't because we just created it, we initialize our array of contact names.

Second, we pull all the people names from our addressBook and keep them as an array of ABRecords.

Finally, we iterate over this array and append just the name of each of the contacts to our array of usersContactsNames. Just for confirmation, we then print out the array so we can see all our contacts names.

```
//Retrieve just the contact's names and store them in our usersContactsNames array.
func getContactsNames() {
    if !self.determinePermissionStatus() {
        return
    }
    if addressBook == nil {
        usersContactsNames = []
        let people = ABAddressBookCopyArrayOfAllPeople(addressBook).takeRetainedValue() as NSArray as [ABRecord]
        for person in people {
            var name: String = ABRecordCopyCompositeName(person).takeRetainedValue()
            self.usersContactsNames?.append(name)
        }
    }
    println(usersContactsNames!)
}
```

```
1 // ViewController.swift
2 // WhosYourValentine1
3 //
4 // Created by Bliss Chapman on 2/15/15.
5 // Copyright (c) 2015 Bliss Chapman. All rights reserved.
6 //
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var displayLabel: UILabel!
14     @IBOutlet weak var valentinesName: UILabel!
15
16     override func viewDidLoad() {
17         super.viewDidLoad()
18         // Do any additional setup after loading the view, typically from a nib.
19     }
20
21     override func didReceiveMemoryWarning() {
22         super.didReceiveMemoryWarning()
23         // Dispose of any resources that can be recreated.
24     }
25
26
27
28     override func motionEnded(motion: UIEventSubtype, withEvent event: UIEvent) {
29
30         println("motion detected!!!")
31     }
32
33 }
34
35
36 }
```

Yay, our model is complete and we now have some powerful methods we can leverage to find our valentine! Open the ViewController again.

Identity and Type

Name View

Type Definition

Location Relative

View

Full Path /Users/Bliss/Deve.../Who.../Who.../View...

Target Membership

WhosYourVal...

WhosYourVal...

Text Settings

Text Encoding Default

Line Endings Default

Indent Using Spaces

Widths

W...

Source Control

Repository --

Type --

Current Branch --

Version --

Status No c...

AVKit Player

view controller

AVPlayer o...

I - A...

ch B...

controlle...

search bar

```
1 //ViewController.swift
2 //WhosYourValentine1
3 //
4 // Created by Bliss Chapman on 2/15/15.
5 // Copyright (c) 2015 Bliss Chapman. All rights reserved.
6 //
7
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var displayLabel: UILabel!
14     @IBOutlet weak var valentinesName: UILabel!
15
16     var contactsHelper = ContactsHelper()
17
18     override func viewDidLoad() {
19         super.viewDidLoad()
20         // Do any additional setup after loading the view, typically from a nib.
21     }
22
23     override func didReceiveMemoryWarning() {
24         super.didReceiveMemoryWarning()
25         // Dispose of any resources that can be recreated.
26     }
27
28
29
30     override func motionEnded(motion: UIEventSubtype, withEvent event: UIEvent) {
31
32         println("motion detected!!!")
33
34     }
35
36
37
38 }
```

The first thing we need to do is initialize a new instance of our ContactsHelper class.

```
// Created by Bliss Chapman on 2/15/15.  
// Copyright (c) 2015 Bliss Chapman. All rights reserved.
```

```
import UIKit  
  
class ViewController: UIViewController {  
  
    @IBOutlet weak var displayLabel: UILabel!  
    @IBOutlet weak var valentinesName: UILabel!  
  
    var contactsHelper = ContactsHelper()  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
    }  
  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
  
    func makeArrayOfNames() {  
        if contactsHelper.determinePermissionStatus() == true {  
            contactsHelper.createAddressBook()  
            if contactsHelper.addressBook != nil {  
                contactsHelper.getContactsNames()  
            }  
        }  
    }  
  
    override func motionEnded(motion: UIEventSubtype, withEvent event: UIEvent) {  
        println("motion detected!!!")  
    }  
}
```

Our final function makes it super simple to make our list of potential valentines. It calls all our helper functions from our model at the appropriate time. Now, all that's left is to call this function when motion is detected and to display our valentine on screen. Almost there!!

In our motion-detecting function, we make an if-statement to distinguish between different potential motions. If the motion was a shake, we update our label and call our `makeArrayOfNames` function which leverages our model to create an array of potential valentines.

Next, we check if our array has a value using if-let syntax. If it does, we are going to generate a random number between 0 and the number of names in `potentialValentines`.

Finally, we update our `valentinesName` label to the element in our array at the index of the random number generated.

```
override func motionEnded(motion: UIEventSubtype, withEvent event: UIEvent) {  
    println("motion detected!!!")  
  
    if motion == UIEventSubtype.MotionShake {  
        displayLabel.text = "Your valentine is:"  
        makeArrayOfNames()  
        if let potentialValentines = contactsHelper.usersContactsNames {  
            var destinyFactor = arc4random_uniform(UInt32(potentialValentines.count))  
            valentinesName.text = potentialValentines[Int(destinyFactor)]  
        }  
    }  
}
```

WhosYourValentine1 > iPhone 6 Running WhosYourValentine1 on iPhone 6

WhosYourValentine1

WhosYourValentine1

ViewController.swift

ContactsHelper.swift

Main.storyboard

Images.xcassets

LaunchScreen.xib

Supporting Files

WhosYourValentine1Tests

Products

// ViewController.swift
// WhosYourValentine1
// Created by Bliss Chapman on 2/15/15.
// Copyright (c) 2015 Bliss Chapman. All rights reserved.

import UIKit

class ViewController: UIViewController {

 @IBOutlet weak var displayLabel: UILabel!
 @IBOutlet weak var valentinesName: UILabel!

 var contactsHelper = ContactsHelper()

 override func viewDidLoad() {
 super.viewDidLoad()
 // Do any additional setup after loading the view.
 }

 override func didReceiveMemoryWarning() {
 super.didReceiveMemoryWarning()
 // Dispose of any resources that can be recreated.
 }

 func makeArrayOfNames() {
 if contactsHelper.determinePermissions()
 contactsHelper.createAddressBook()
 if contactsHelper.addressBook != nil
 contactsHelper.getContactsNames()
 }
 }

 override func motionEnded(motion: UIEventSubtype, withEvent event: UIEvent!) {
 println("motion detected!!!")

 if motion == UIEventSubtype.MotionShake
 displayLabel.text = "Your Valentine!"
 makeArrayOfNames()
 if let potentialValentines = contactsHelper.names
 var destinyFactor = arc4random()%potentialValentines.count
 valentinesName.text = potentialValentines[destinyFactor]
 }
}

iOS Simulator - iPhone 6 - iPhone 6 / iOS 8.1 (12B411)
Carrier 11:41 AM

"WhosYourValentine1" Would Like to Access Your Contacts
Don't Allow OK

Identity and Type

Name ViewController.swift

Type Default - Swift Source

Location Relative to Group

ViewController.swift

Full Path /Users/blisschapman/Developer/WhosYourValentine1/WhosYourValentine1/ViewController.swift

Target Membership

✓ WhosYourValentine1

WhosYourValentine1Tests

Text Settings

Text Encoding Default - Unicode (UTF-8)

Line Endings Default - OS X / Unix (LF)

Indent Using Spaces

Widths 4 Tab Indent

Wrap lines

Source Control

Repository --

Type --

Current Branch --

Version --

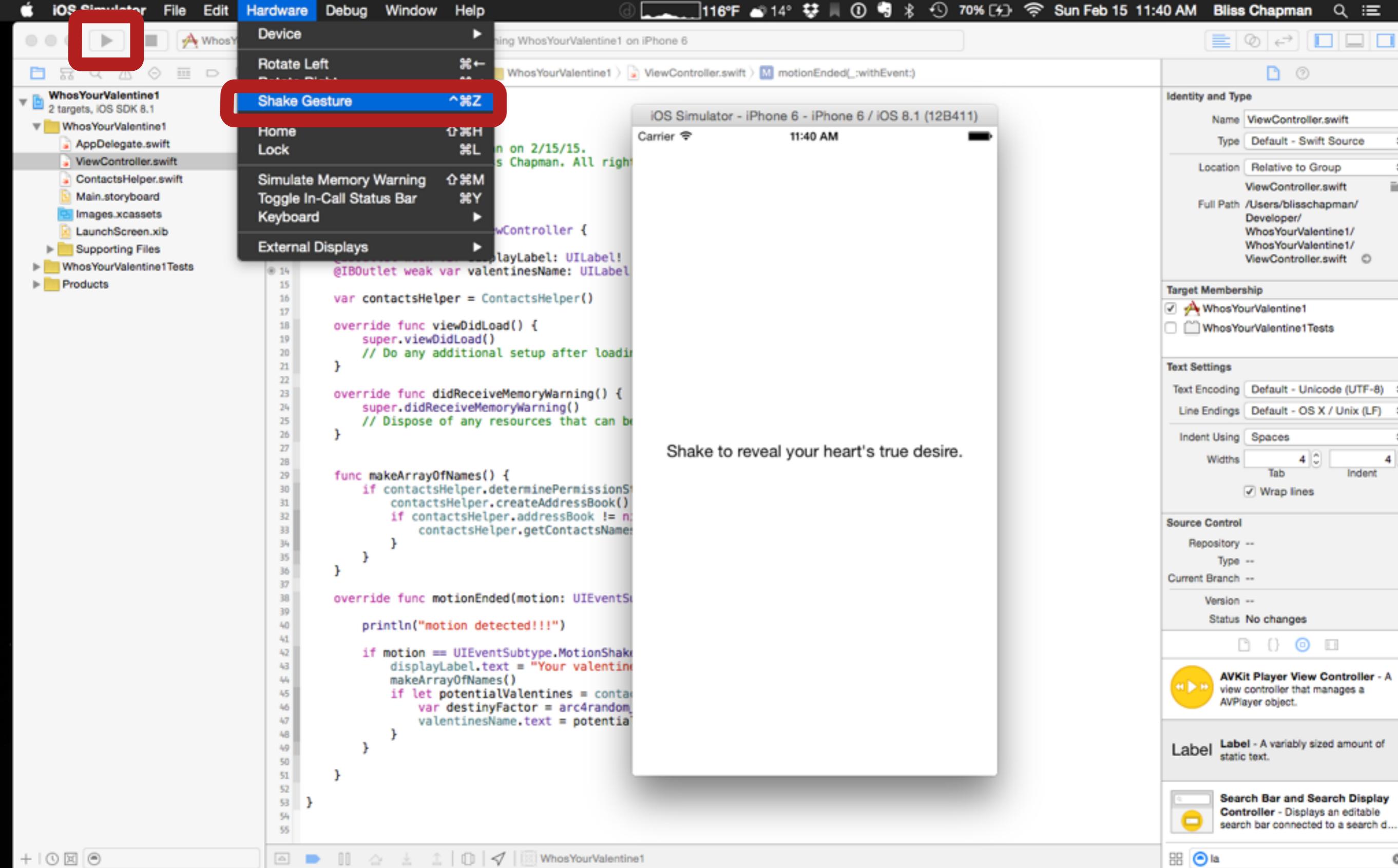
Status No changes

AVKit Player View Controller - A view controller that manages an AVPlayer object.

Label Label - A variable sized amount of static text.

Search Bar and Search Display Controller - Displays an editable search bar connected to a search display.

Yay! NOTE: for time-saving purposes, we did not include functionality to call our helper methods again if permission was not already granted. You may have to build and run again.



Build and Run.

WhosYourValentine1 > iPhone 6 Running WhosYourValentine1 on iPhone 6

WhosYourValentine1
2 targets, iOS SDK 8.1
WhosYourValentine1
 AppDelegate.swift
 ViewController.swift
 ContactsHelper.swift
 Main.storyboard
 Images.xcassets
 LaunchScreen.xib
 Supporting Files
WhosYourValentine1Tests
Products

```
8 import UIKit
9
10 class ViewController: UIViewController {
11
12     @IBOutlet weak var displayLabel: UILabel!
13     @IBOutlet weak var valentinesName: UILabel!
14
15     var potentialValentines = [String]()
16
17     override func motionEnded(motion: UIEventSubtype, withEvent event: UIEvent) {
18         if motion == .MotionShake {
19             let randomIndex = Int(arc4random_uniform(UInt32(potentialValentines.count)))
20             valentinesName.text = potentialValentines[randomIndex]
21         }
22     }
23
24     override func viewDidLoad() {
25         super.viewDidLoad()
26
27         // Do any additional setup after loading the view, typically from a nib.
28     }
29
30     override func didReceiveMemoryWarning() {
31         super.didReceiveMemoryWarning()
32     }
33
34     @IBAction func shakeButtonTapped(sender: UIButton) {
35         motionEnded(.MotionShake, withEvent: nil)
36     }
37
38     override func prefersStatusBarHidden() -> Bool {
39         return true
40     }
41
42     override func prefersStatusBarStyle(style: UIStatusBarStyle) -> Bool {
43         return true
44     }
45
46     var destinyFactor: Int = 0
47
48     var contactsNames: [String] = [
49         "Ruth, Yankee Doodle, Tim Cook, Leif Ericsson, Josiah Bartlett",
50         "motion detected!!!",
51         "[Daniel Higgins Jr., Hank M. Zakroff, David Taylor, Babe Ruth, Yankee Doodl", "Tim Cook, Leif Ericsson, Josiah Bartlett"
52     ]
53
54     var destinyFactor: Int = 0
55 }
```

iOS Simulator - iPhone 6 - iPhone 6 / iOS 8.1 (12B411)
Carrier WiFi 11:41 AM

typically from a nib.

Your valentine is:
Josiah Bartlett

Ruth, Yankee Doodle, Tim Cook, Leif Ericsson, Josiah Bartlett
motion detected!!!
[Daniel Higgins Jr., Hank M. Zakroff, David Taylor, Babe Ruth, Yankee Doodl
Tim Cook, Leif Ericsson, Josiah Bartlett]

All Output

Identity and Type
Name: ViewController.swift
Type: Default - Swift Source
Location: Relative to Group
ViewController.swift
Full Path: /Users/blisschapman/Developer/WhosYourValentine1/WhosYourValentine1/ViewController.swift

Target Membership
 WhosYourValentine1
 WhosYourValentine1Tests

Text Settings
Text Encoding: Default - Unicode (UTF-8)
Line Endings: Default - OS X / Unix (LF)
Indent Using: Spaces
Widths: 4 Tab Indent
 Wrap lines

Source Control
Repository --
Type --
Current Branch --
Version --
Status: No changes

AVKit Player View Controller - A view controller that manages an AVPlayer object.

Label - A variably sized amount of static text.

Search Bar and Search Display Controller - Displays an editable search bar connected to a search display.

Woohoo! Now, shake your simulator to your heart's content! Feel free to add custom contacts by clicking command-shift-h and opening up the contacts application. It makes it even more dramatic!

Congratulations
iOS Developer!

Thank You

Thanks so much for coming. Remember, if you have any questions please, please ask!

There are lots of past demos available on the website for you to work through and fantastic resources we recommend to further you on your iOS development journey.

Best of luck and hope to see you next week!