

---

# View Controllers

## MK Map View

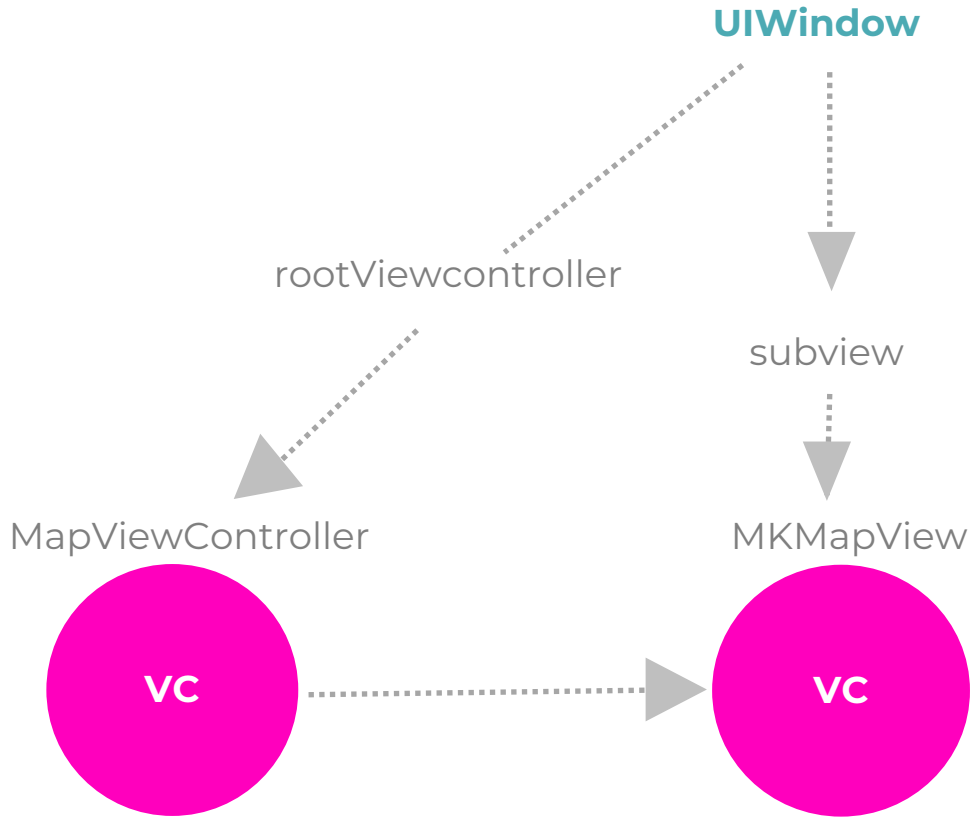


## View Controller

Manages view hierarchy

Creates view objects

Handles events associated with view objects



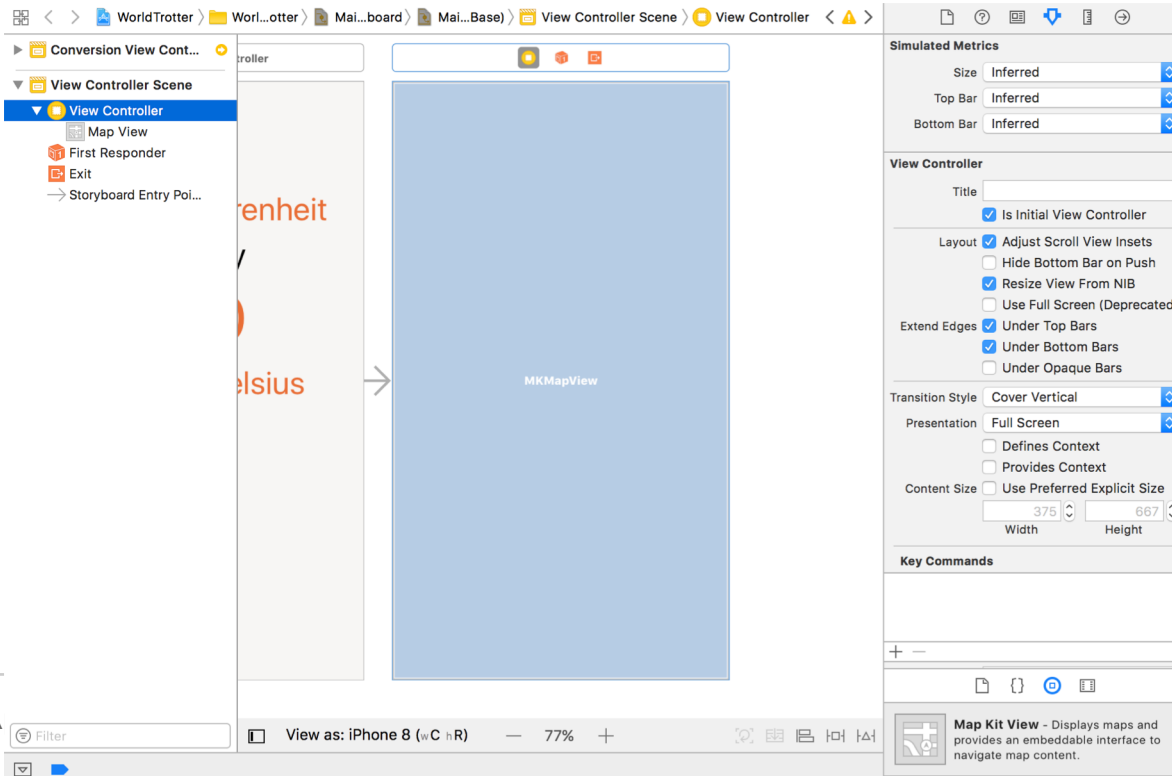
- 2 ways to create ViewController:
1. Using storyboard in IB
  2. Programmatically overriding method `loadView()`

Each storyboard has only one Initial view controller. It is an entry point to the storyboard. This view controller shows first.

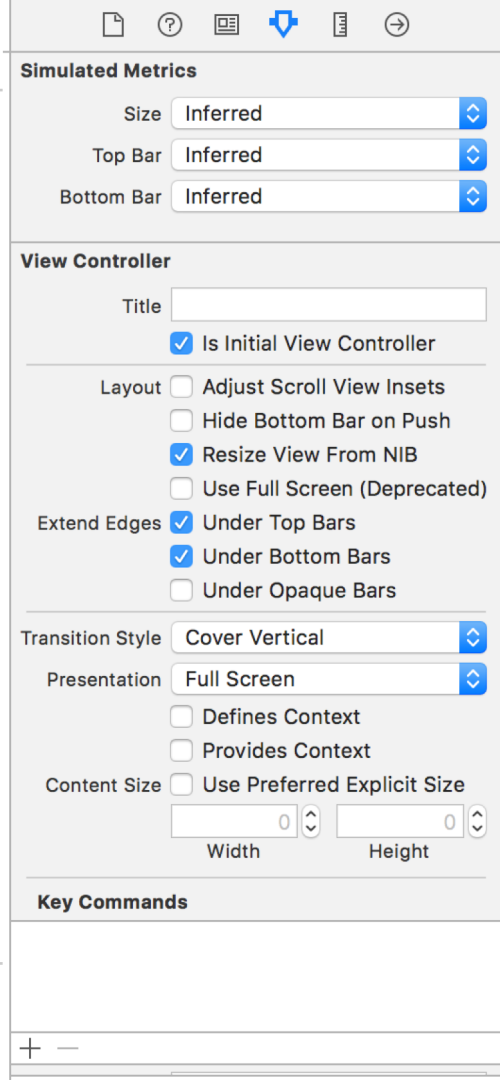
Open Main.storyboard.

Drag View Controller onto canvas. Select the View of the ViewController – Delete.

Drag Map Kit View from object library (search) onto controller to set map view.



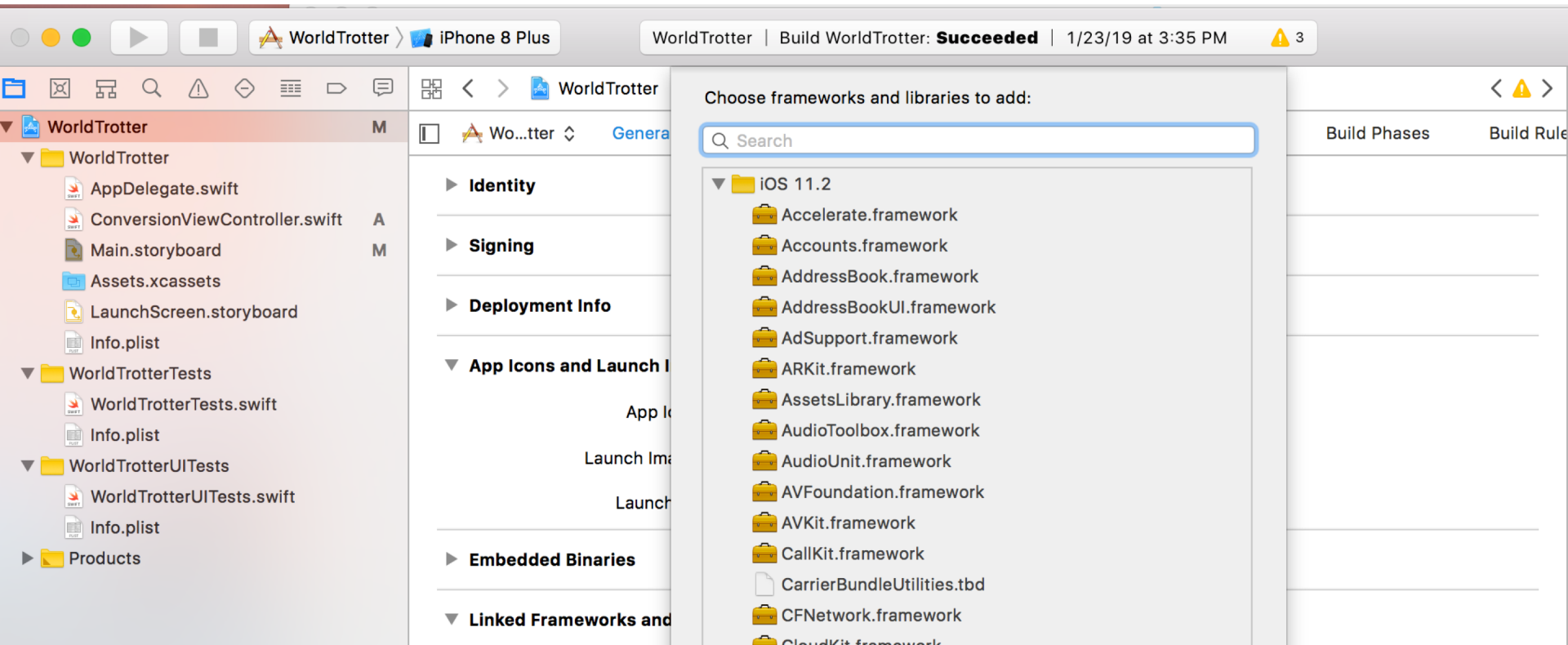
Select View Controller and open Attributes inspector.  
Check Is Initial View Controller  
Notice the change of direction of the gray arrow  
Now it points to View Controller now, not Conversion View Controller.  
Test and run the app.



Import MapKit framework and link it to the view to load map view.

WorldTrotter project > Project Settings > General > Linked Frameworks and Libraries

Check + sign and search for MapKit framework. Click Add. Build and run to test map view.

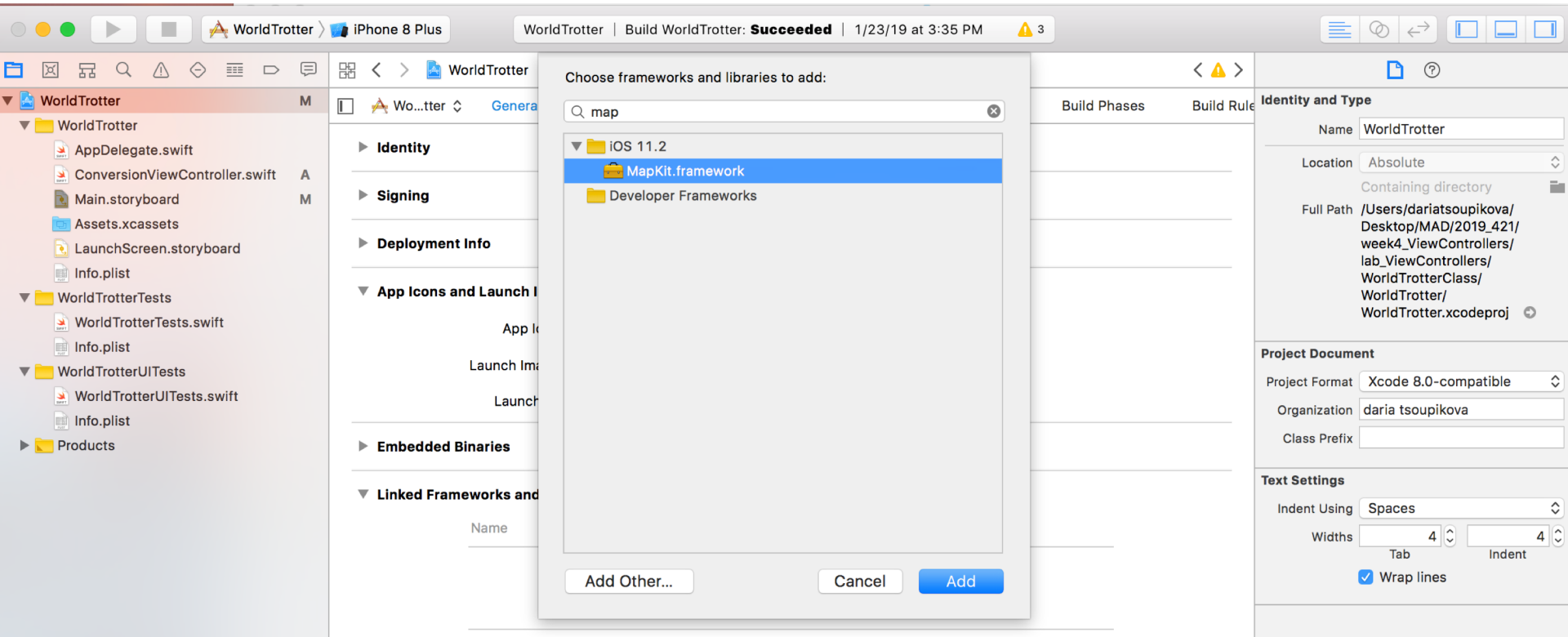


Add the MapKit framework

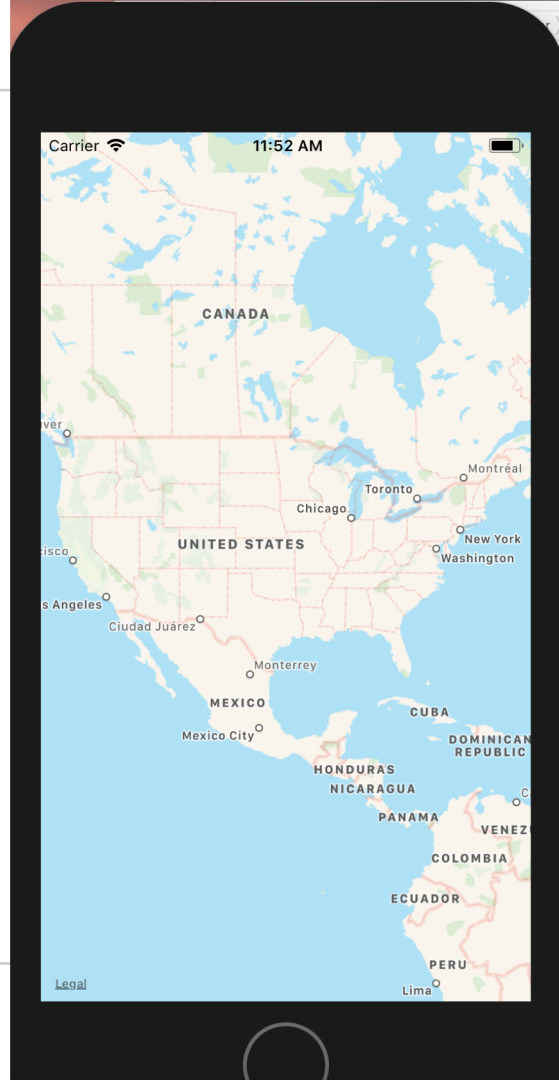
Import MapKit framework and link it to the view to load map view.

WorldTrotter project > Project Settings > General > Linked Frameworks and Libraries

Check + sign and search for MapKit framework. Click Add. Build and run to test map view.



Build and run to test map view.

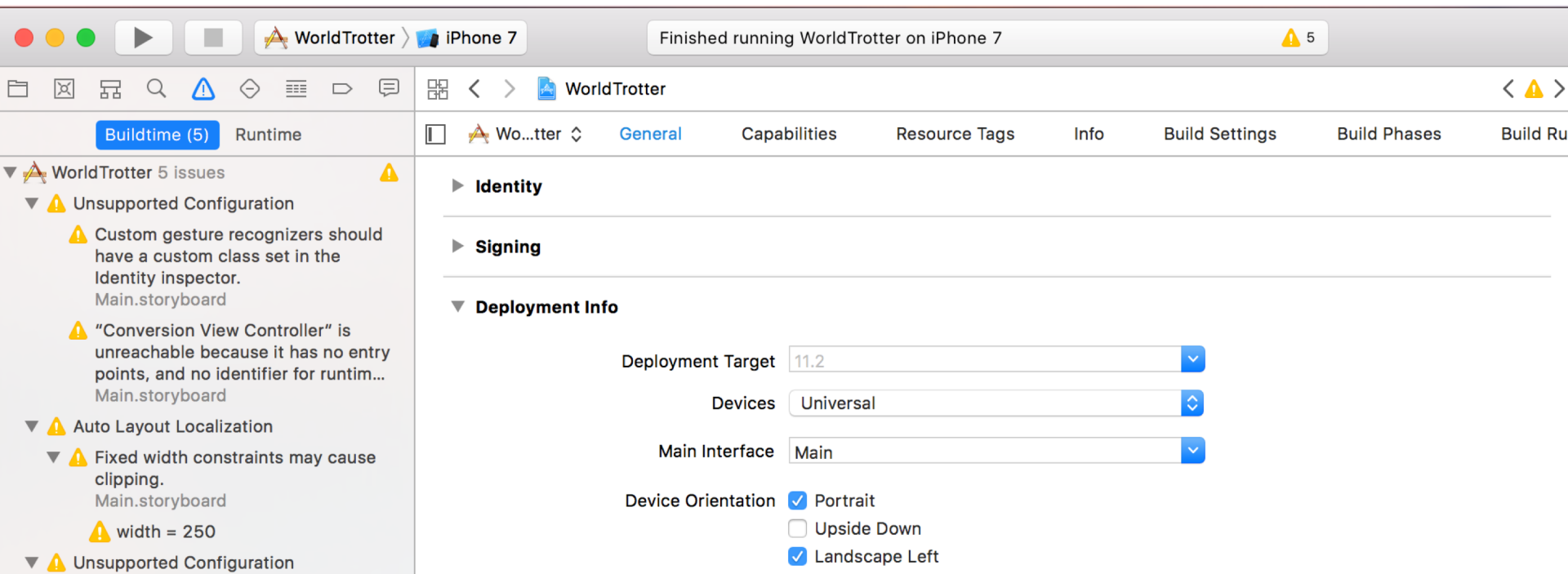




Each app has one main interface (main.storyboard)

When the app launches, the initial view controller gets set as the rootViewController of the window.

The main interface is set in the project settings >General > Deployment >Main Interface  
Main corresponds to Main. Storyboard.



Add Transition

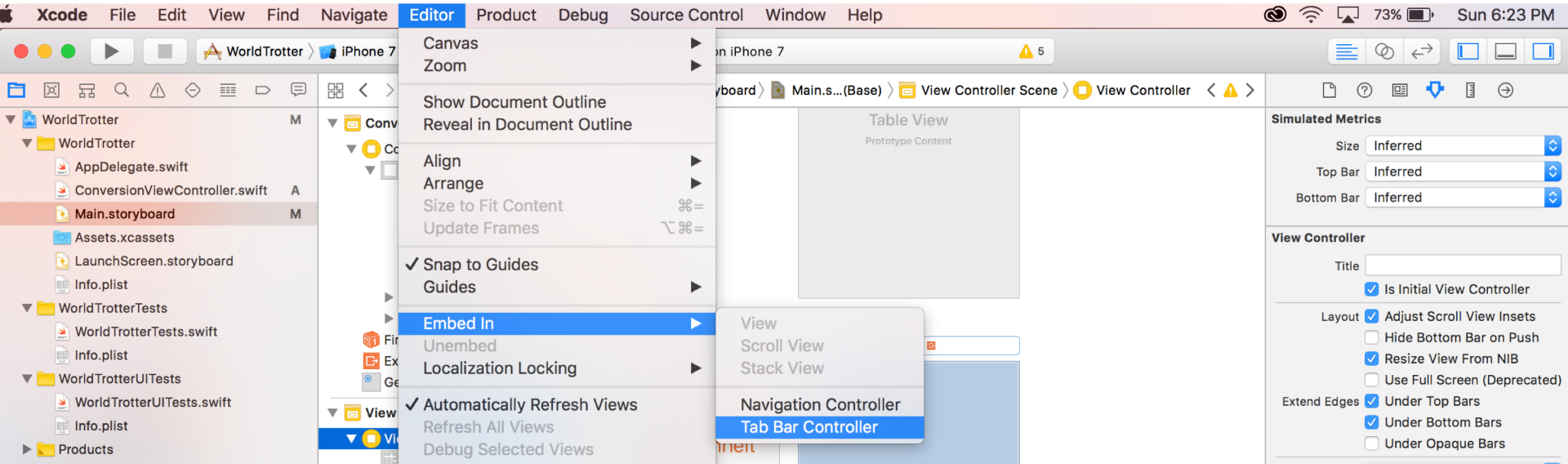
UITabBarController allows swap between the View Controllers.

Keeps array of view controllers

Maintains a menu to select view controllers

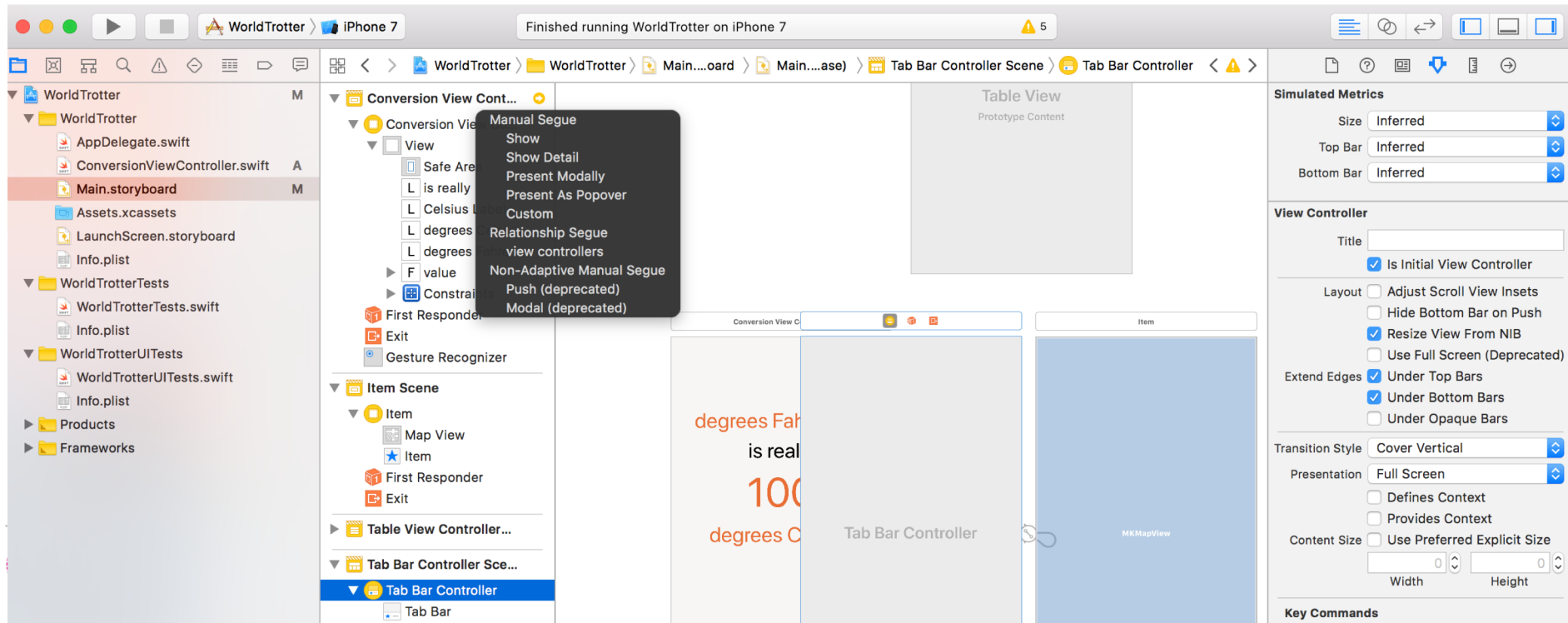
Open Main.storyboard> select View Controller

Editor > Embed In > Tab Bar Controller . Relationship arrow will be added to pointing from TabBar Controller to View Controller

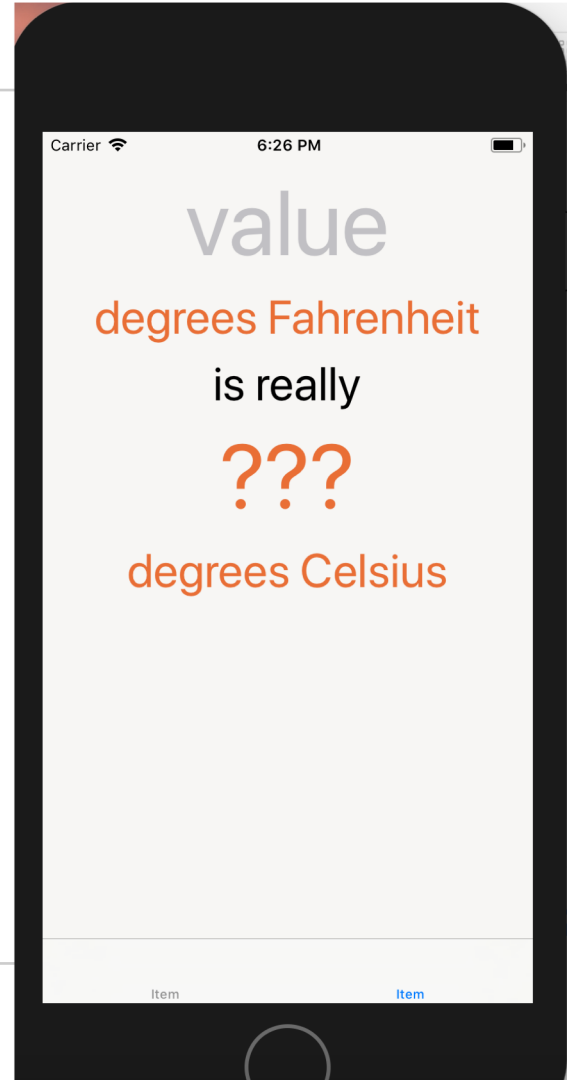
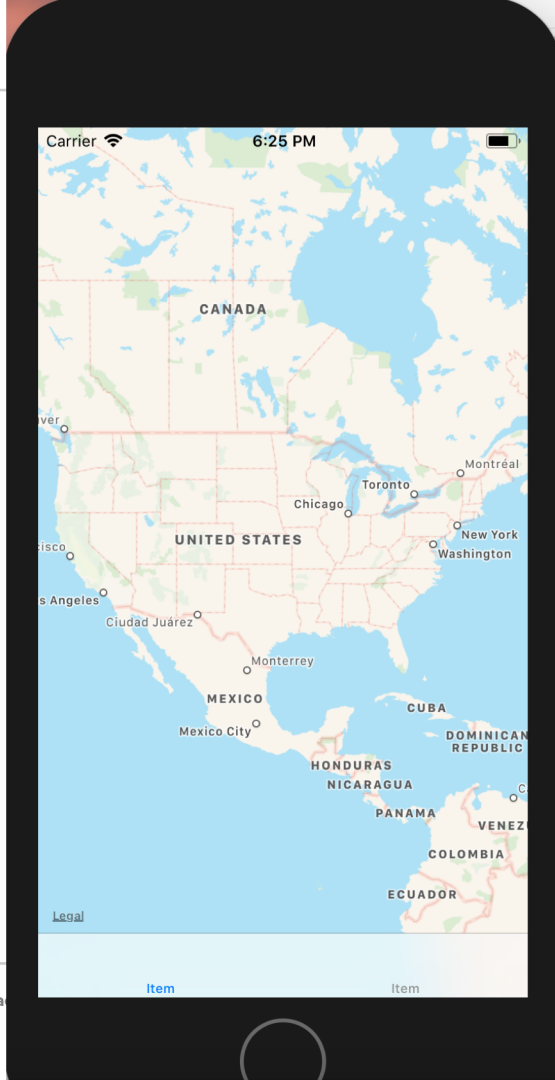


Add Conversion ViewController to TabBarController to switch between the screens:

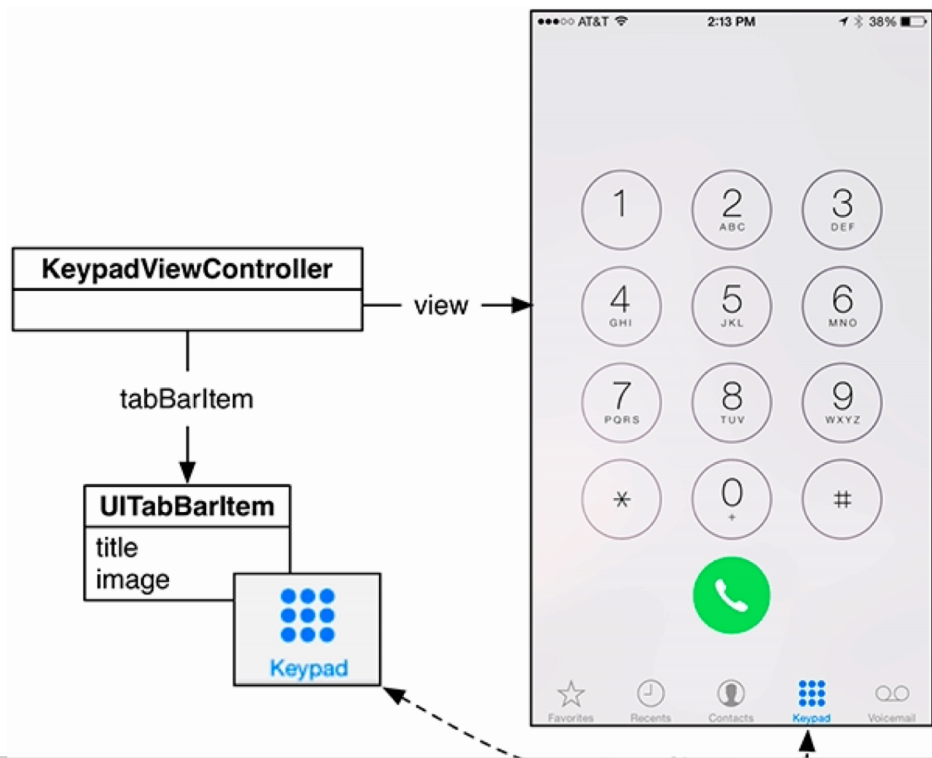
Control-drag from the TabBarController to the ConversionViewController  
Choose: Relationship Segue > view controllers. Build, run and test switching views.



Tab menu bars to switch between the screens:



Each tab bar in the menu can display a title and an image. (currently untitled “Item”)



Add image by adding it to assets  
Assets are files to be selected at runtime

Open Assets.xcassets

download from

<https://www.bignerdranch.com/solutions/iOSProgramming6ed.zip>

Under Ch. 5 Xcode Assets

ConvertIcon.png

ConvertIcon@2x.png

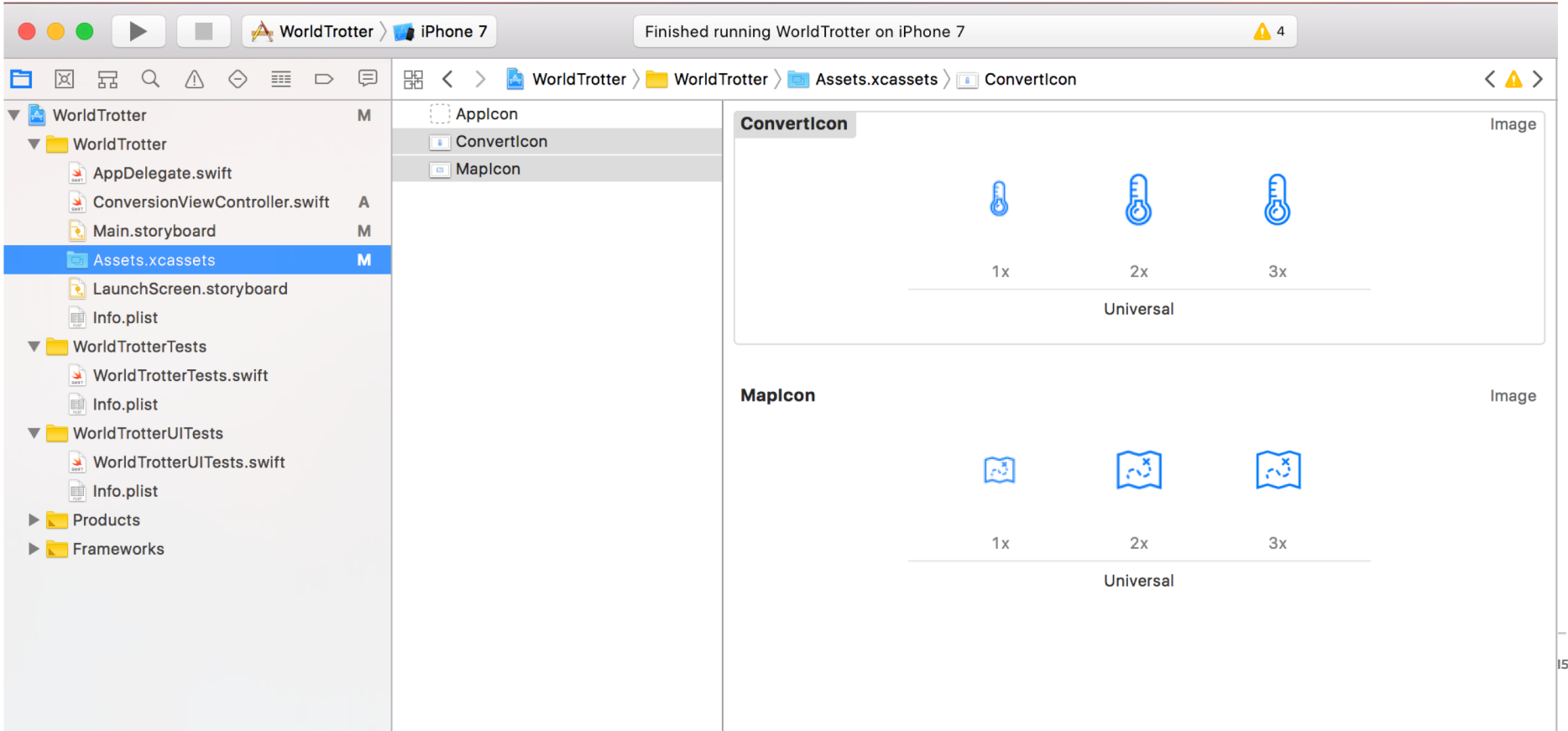
ConvertIcon@3x.png

MapIcon.png

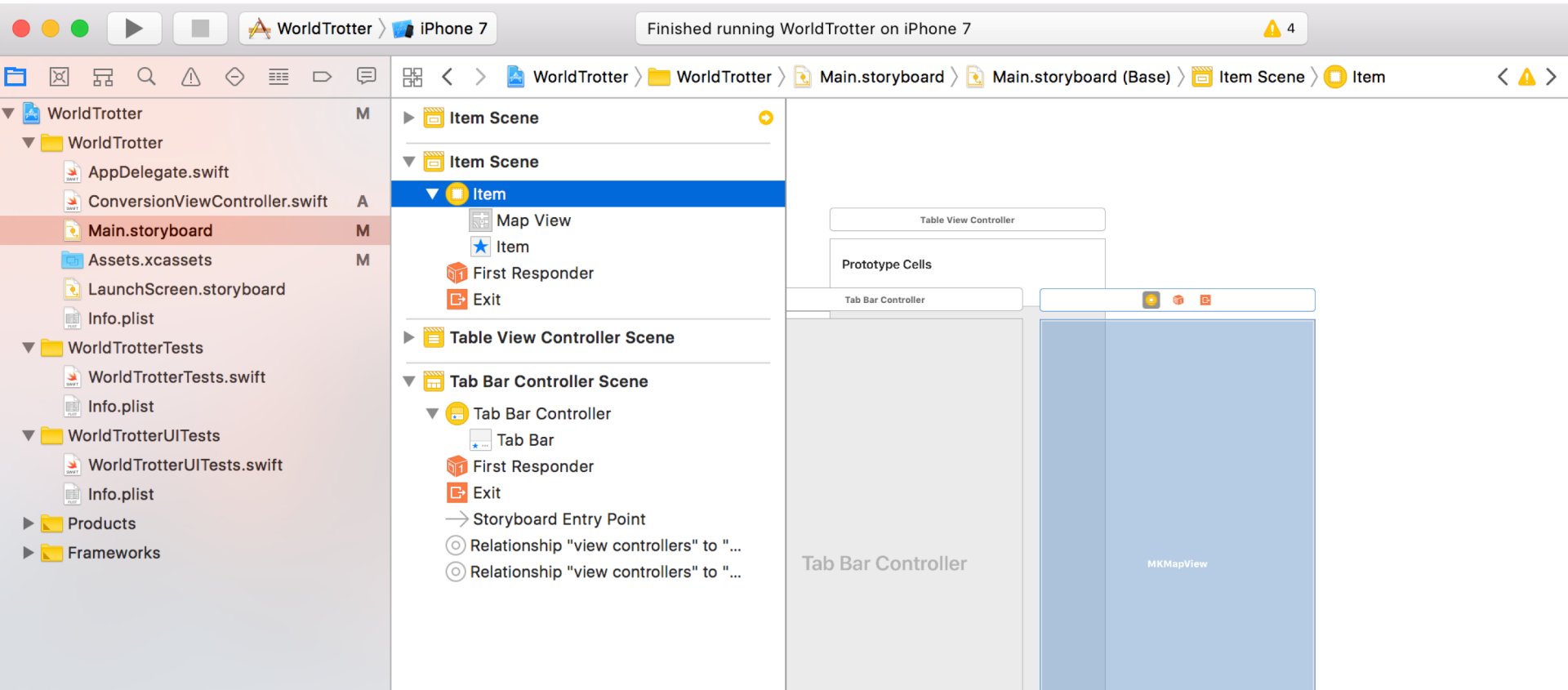
MapIcon@2x.png

MapIcon@3x.png

Drag these images to images set list of the Assets Catalog.

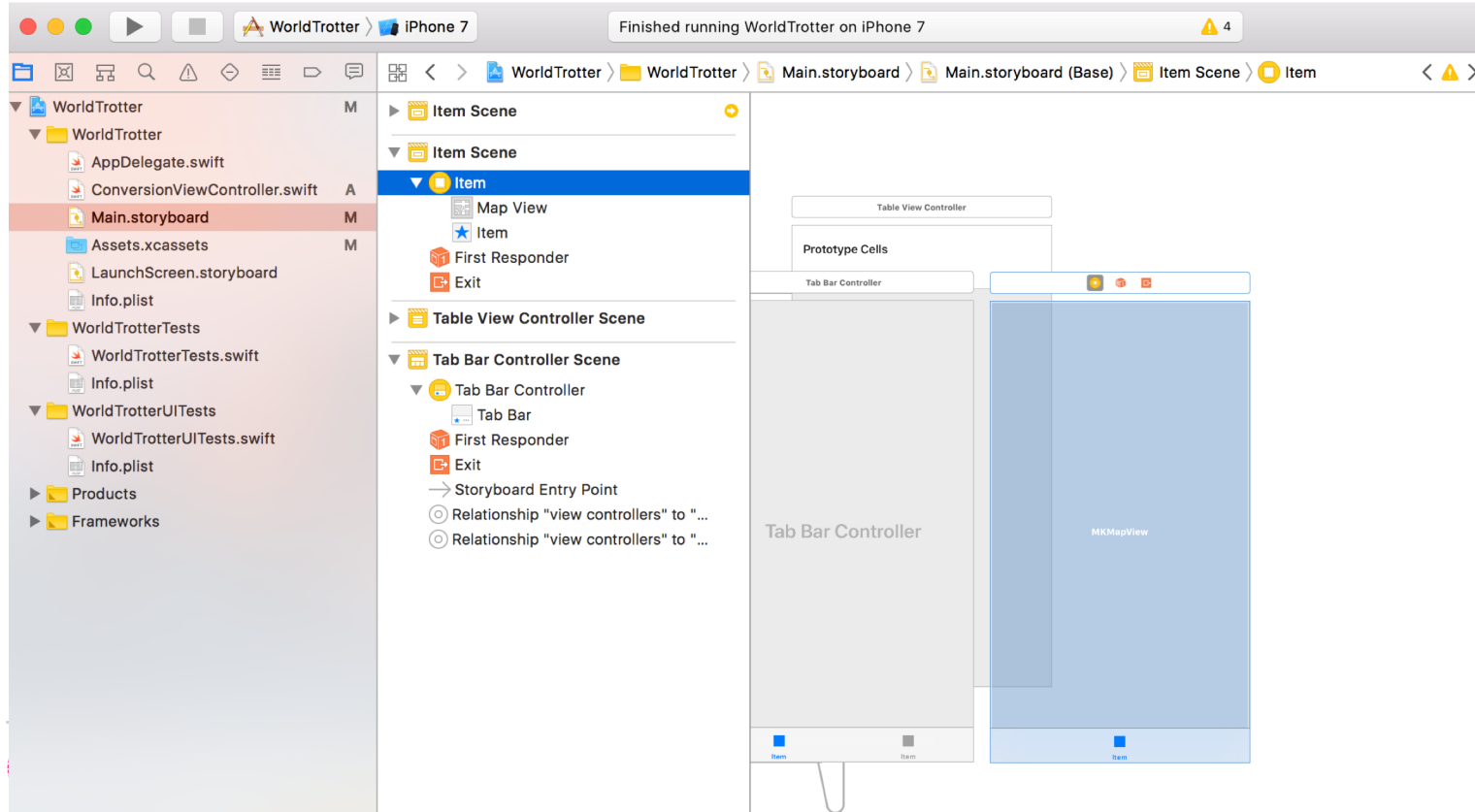


In Main.storyboard select ViewController – it is named Item now

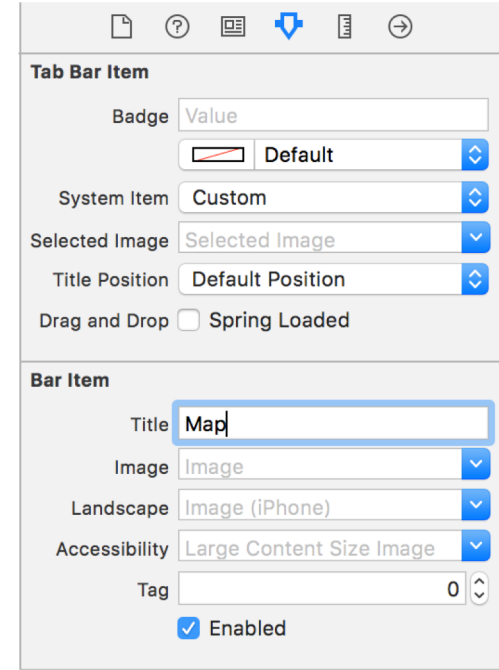
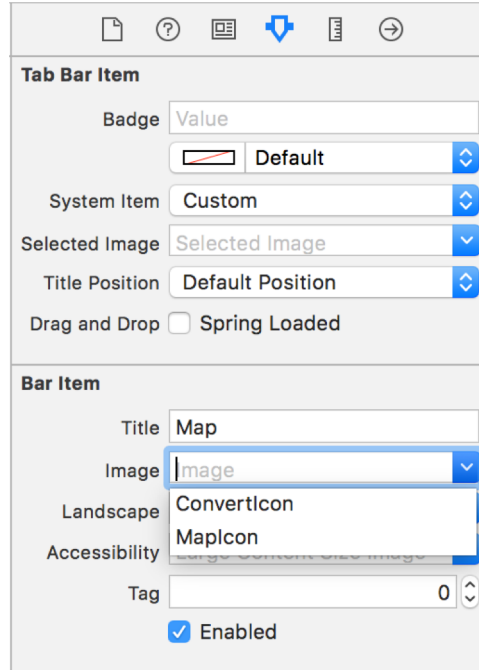




Notice the tab bar is added to the bottom of the Interface

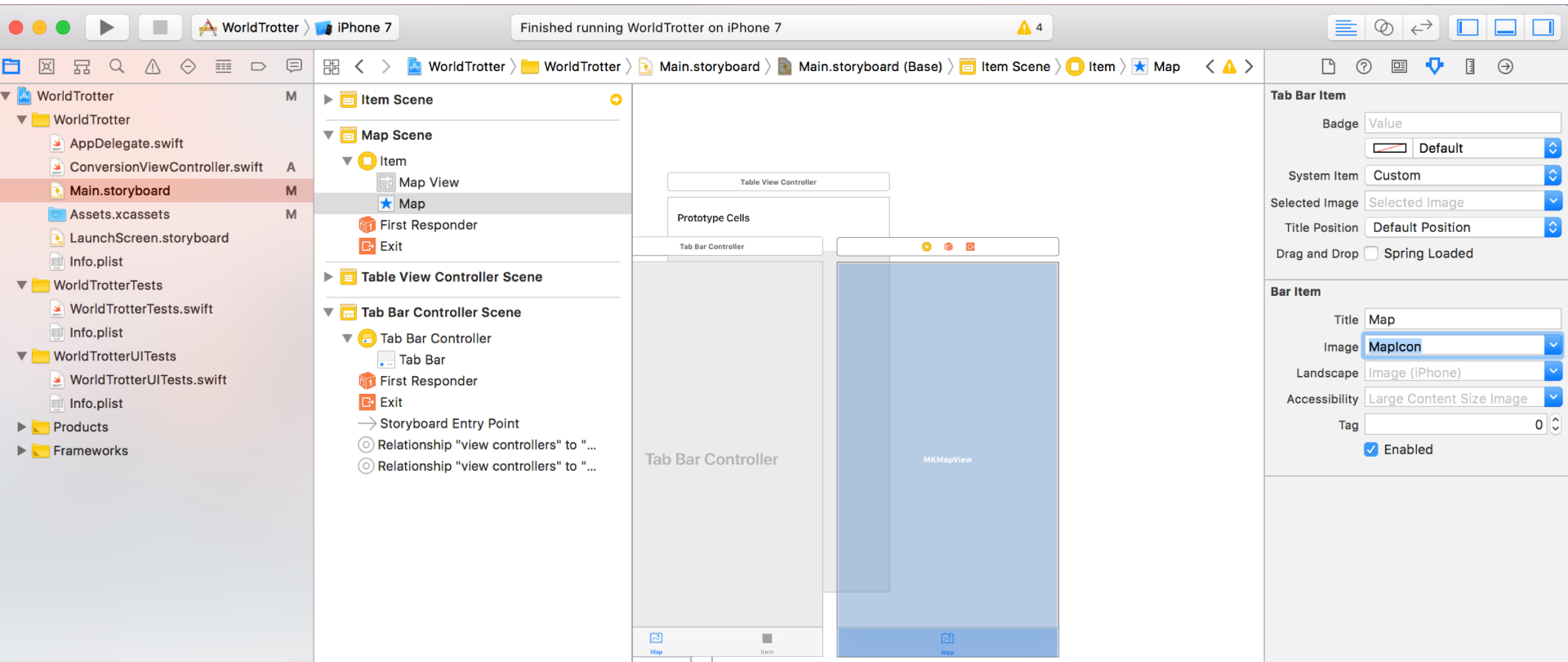


Select this tab bar and open attributes  
Change name to Map  
Choose MapIcon as Image



Modify the menu

- Select this tab bar and open attributes
- Change name to Map
- Choose MapIcon as Image



# Select tab bar in ConversionViewController Change name to "Convert" Choose ConvertIcon as Image

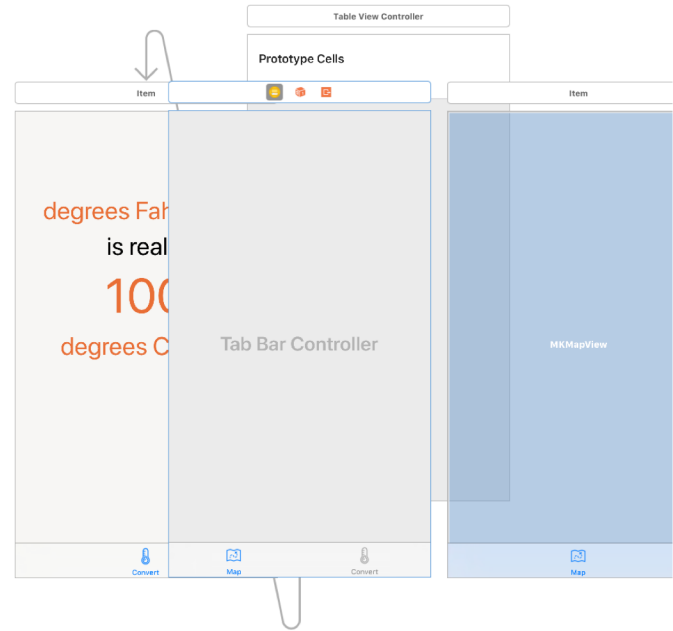
The screenshot shows the Xcode interface with a storyboard on the left and the Attributes Inspector on the right. The storyboard displays a tab bar at the bottom with a tab labeled "Convert". The main content area shows a text view with the text "degrees Fahrenheit is really 100 degrees Celsius". The Attributes Inspector on the right is set to the "Tab Bar Item" section, showing the following settings:

- Badge: Value
- System Item: Default
- System Item: Custom
- Selected Image: Selected Image
- Title Position: Default Position
- Drag and Drop:  Spring Loaded
- Bar Item:
  - Title: Convert
  - Image: ConvertIcon
  - Landscape: Image (iPhone)
  - Accessibility: Large Content Size Image
  - Tag: 0
  - Enabled

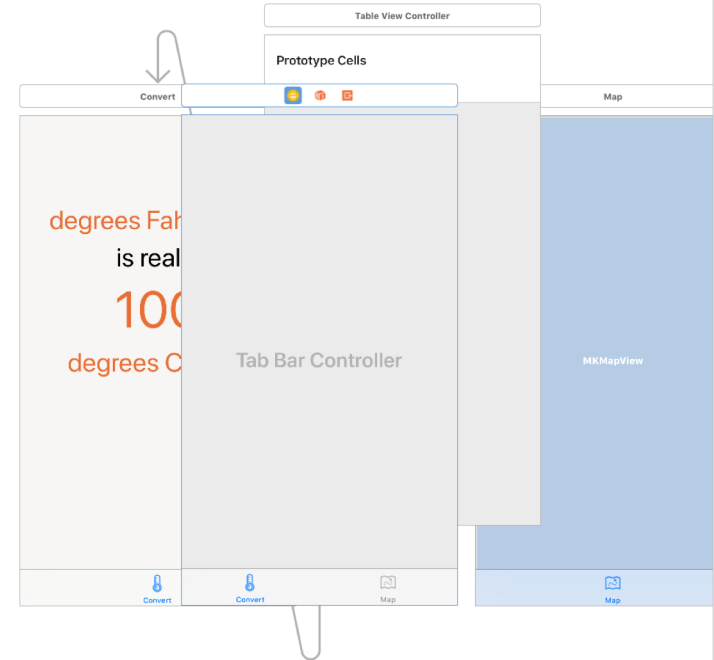
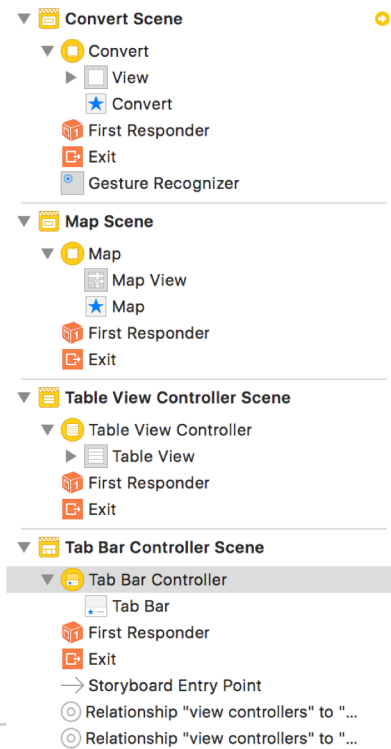
Select Tab Bar Controller on canvas (or in storyboard)  
Drag the Convert tab to be on the left

WorldTrotter > WorldTrotter > Main...oard > Main...ase > Tab Bar Controller Scene > Tab Bar Controller < >

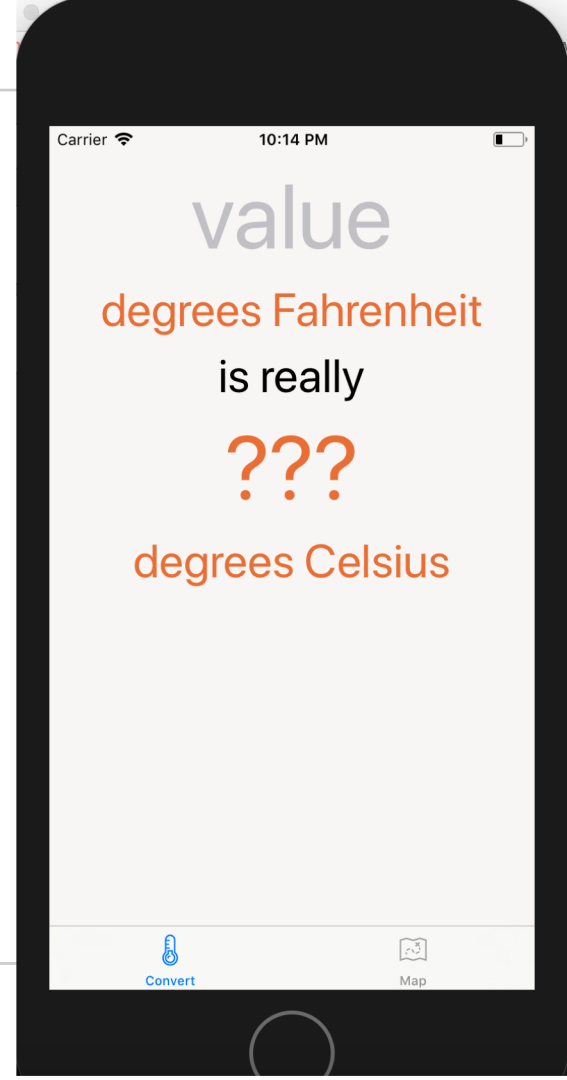
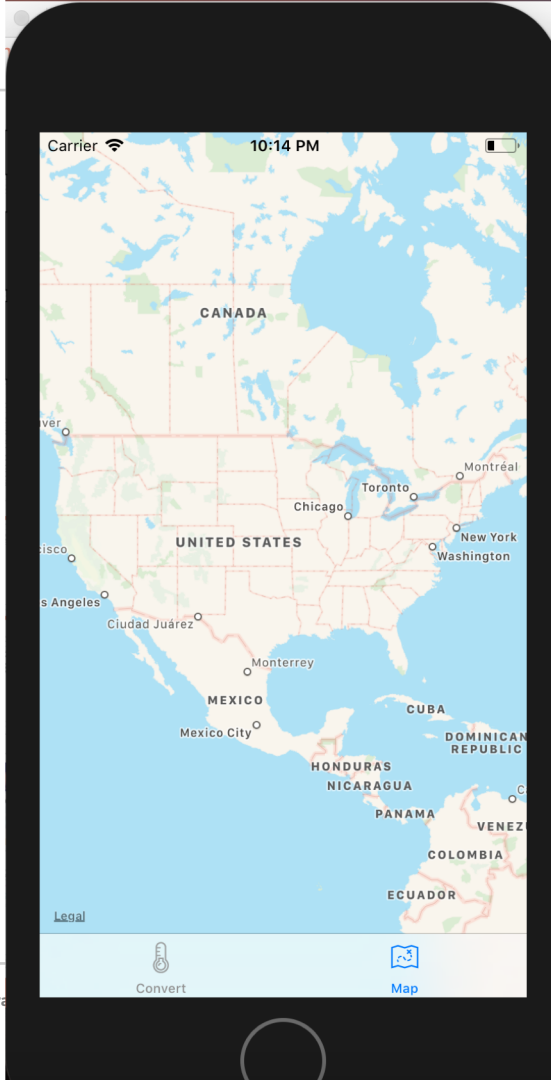
- Convert Scene
  - Item
    - View
      - Convert
    - First Responder
    - Exit
    - Gesture Recognizer
- Map Scene
  - Item
    - Map View
      - Map
    - First Responder
    - Exit
- Table View Controller Scene
- Tab Bar Controller Scene
  - Tab Bar Controller**
    - Tab Bar
    - First Responder
    - Exit
    - Storyboard Entry Point
    - Relationship "view controllers" to "..."
    - Relationship "view controllers" to "..."



Select Tab Bar Controller on canvas (or in storyboard)  
Drag the Convert tab to be on the left

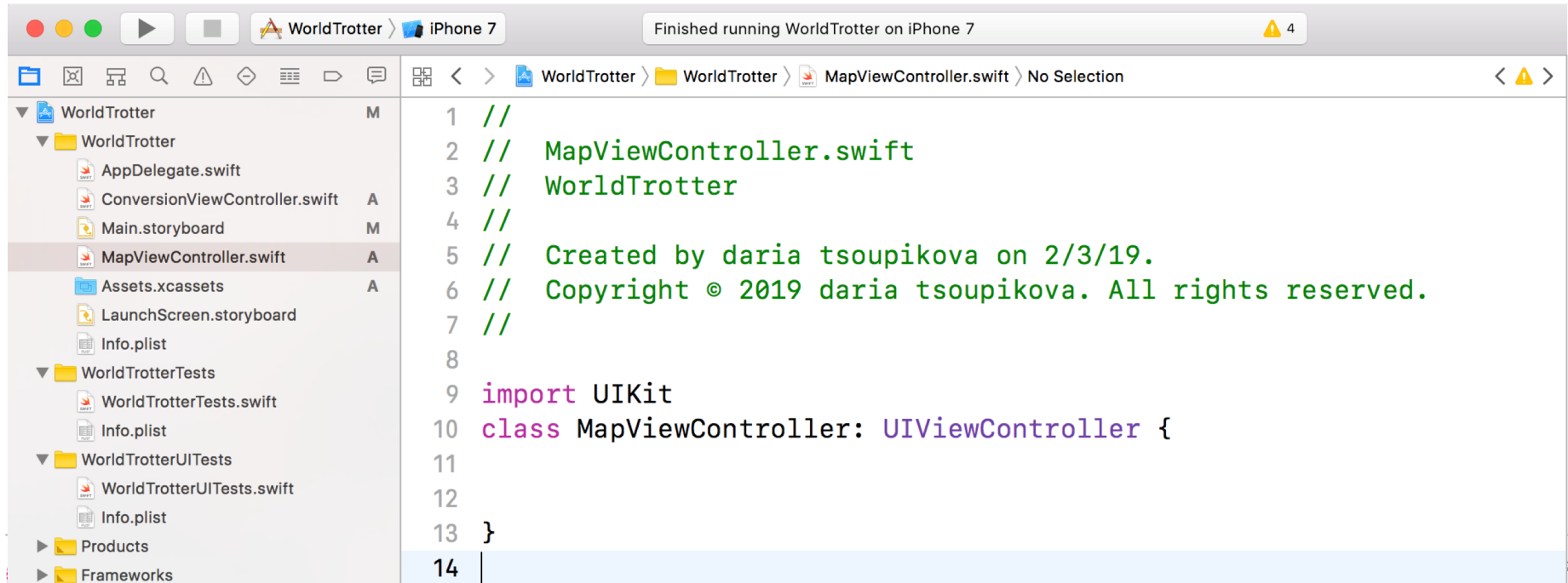


## Build, run and test



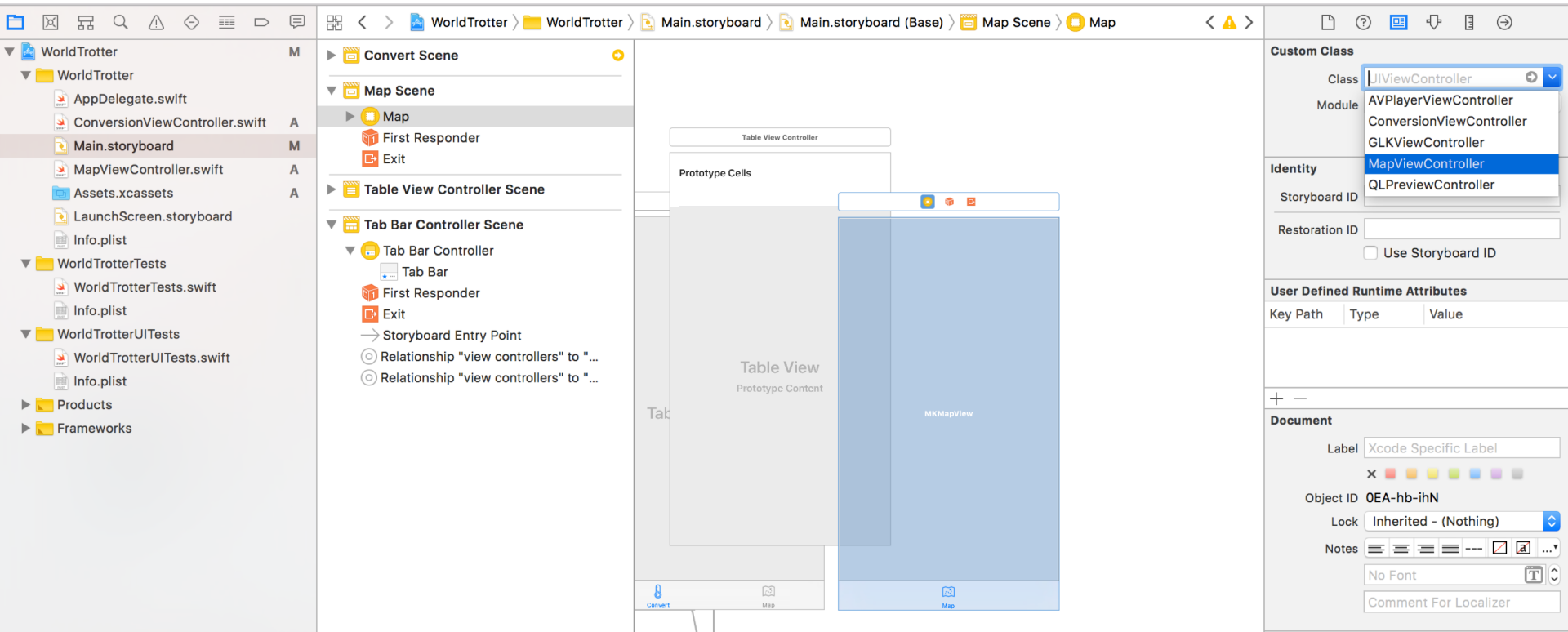
MapViewController is not needed until user taps on the menu item

Create new Swift file "MapViewController"  
Define a UIViewController subclass MapViewController





Open Main.Storyboard and select map's view controller  
Open identity inspector and change the class to MapViewController class.



Memory is limited. Test and see how long it takes to load the map...

**Lazy loading** make app run faster and makes memory more efficient.  
Defers allocation until user needs it.

- Optimization
- The controller's view is not created until it needs to be presented on the screen
- Saves memory
- Improves performance

In ConversionViewController.swift

Update viewDidLoad()

```
override func viewDidLoad() {  
super.viewDidLoad()
```

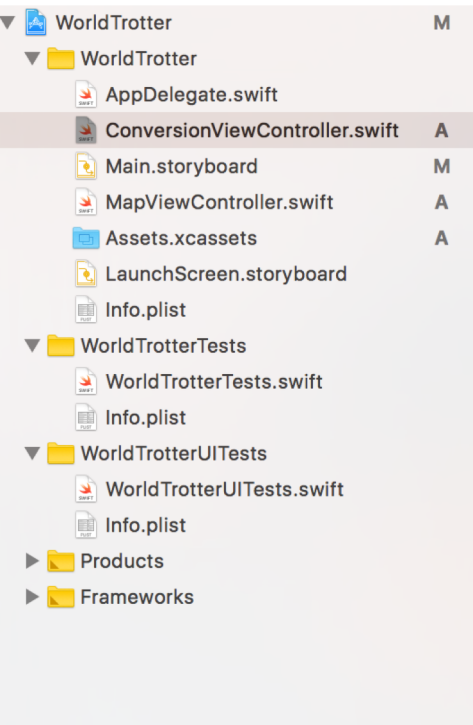
```
print("ConversionViewController loaded its view")
```

```
updateCelsiusLabel()
```

```
}
```

## In ConversionViewController.swift

### Update viewDidLoad()

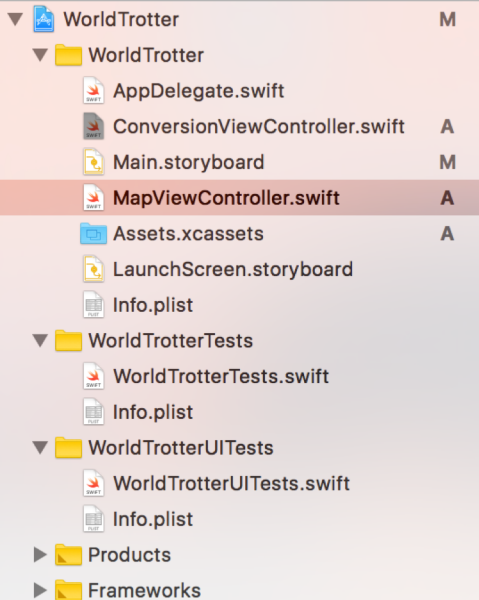


```
69         fahrenheitValue = nil
70     }
71 }
72
73 @IBAction func dismissKeyboard(_ sender:
74     UITapGestureRecognizer) {
75     textField.resignFirstResponder()
76 }
77
78 override func viewDidLoad() {
79     super.viewDidLoad()
80     print("ConversionViewController loaded its view")
81
82     updateCelsiusLabel()
83 }
84
```

In MapViewController.swift

Update viewDidLoad()

```
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    print("MapViewController loaded its view")  
  
}
```



```
1 //
2 // MapViewController.swift
3 // WorldTrotter
4 //
5 // Created by daria tsoupikova on 2/3/19.
6 // Copyright © 2019 daria tsoupikova. All rights reserved.
7 //
8
9 import UIKit
10 class MapViewController: UIViewController {
11
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15
16         print("MapViewController loaded its view")
17
18
19     }
20
21
22 }
```

## Build, run and test.

The image shows a screenshot of the Xcode IDE. On the left, a simulator displays a mobile application interface with the following text: "value", "degrees Fahrenheit", "is really", "???", and "degrees Celsius". At the bottom of the simulator are two buttons labeled "Convert" and "Map".

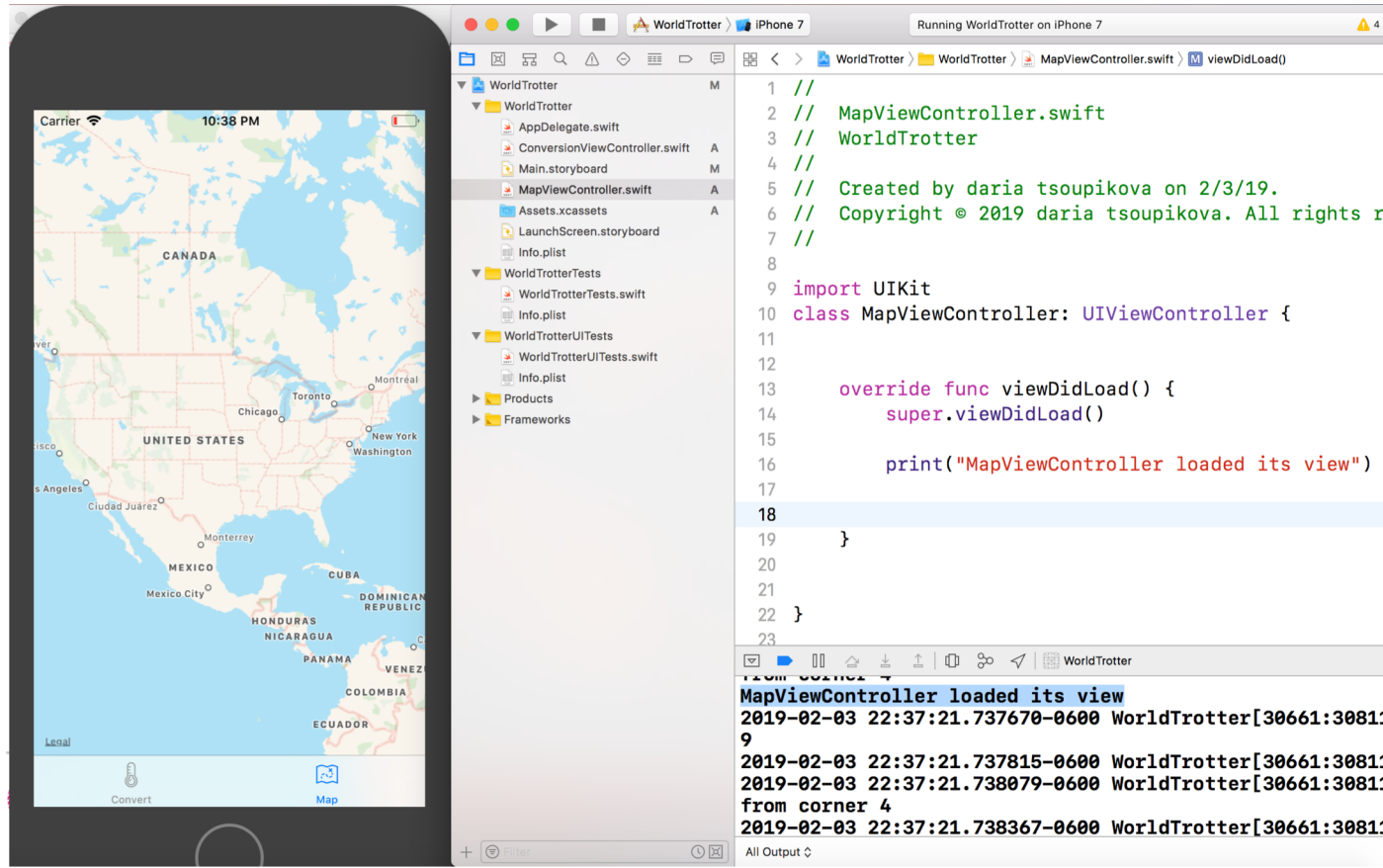
The main area of the IDE shows the Swift source code for `MapViewController.swift`. The code is as follows:

```
1 //  
2 // MapViewController.swift  
3 // WorldTrotter  
4 //  
5 // Created by daria tsouplikova on 2/3/19.  
6 // Copyright © 2019 daria tsouplikova. All rights reserved.  
7 //  
8  
9 import UIKit  
10 class MapViewController: UIViewController {  
11  
12  
13     override func viewDidLoad() {  
14         super.viewDidLoad()  
15  
16         print("MapViewController loaded its view")  
17  
18  
19     }  
20  
21  
22 }  
23
```

The bottom of the IDE shows the console output:

```
ConversionViewController loaded its view  
2019-02-03 22:36:03.537198-0600 WorldTrotter[30661:3081:  
MobileCoreServices.framework  
2019-02-03 22:36:03.541215-0600 WorldTrotter[30661:3081:  
MobileCoreServices.framework
```

# Build, run and test.



The image shows a screenshot of an iPhone simulator on the left and the Xcode IDE on the right. The simulator displays a map of North America with labels for Canada, United States, Mexico, and other countries. The time is 10:38 PM. The Xcode IDE shows the project structure for 'WorldTrotter' and the code for 'MapViewController.swift'. The code includes a class definition and an override for the 'viewDidLoad()' method that prints a message. The console output shows the message 'MapViewController loaded its view' being printed.

```
1 //  
2 // MapViewController.swift  
3 // WorldTrotter  
4 //  
5 // Created by daria tsoupikova on 2/3/19.  
6 // Copyright © 2019 daria tsoupikova. All rights reserved.  
7 //  
8  
9 import UIKit  
10 class MapViewController: UIViewController {  
11  
12  
13     override func viewDidLoad() {  
14         super.viewDidLoad()  
15  
16         print("MapViewController loaded its view")  
17  
18     }  
19 }  
20  
21  
22 }  
23
```

MapViewController loaded its view  
2019-02-03 22:37:21.737670-0600 WorldTrotter[30661:3081:  
9  
2019-02-03 22:37:21.737815-0600 WorldTrotter[30661:3081:  
2019-02-03 22:37:21.738079-0600 WorldTrotter[30661:3081:  
from corner 4  
2019-02-03 22:37:21.738367-0600 WorldTrotter[30661:3081:



## Lazy loading

2 methods:

- Override viewDidLoad() method
  - This method is called after the view controller's interface file is loaded.
- Override viewWillAppear(\_: ) method
  - This method is called just before a controller's view is added to the window.

Use this method if configuration needs once during the the run of the app.

Use this method if you need the configuration to be done each time the view controller's view appears on the screen.

## **init(coder:)**

Initializes UIViewController instances created from a storyboard

## **init(nibName:bundle:)**

designated initializer for UIViewController (for instances created without storyboard, programmatically). Called once on each viewController as it is created.

## **loadView()**

Is overridden to create a view controller's view programmatically

## **viewDidLoad()**

Is overridden to configure views created by loading an interface file. This method is called after the view controller is created.

## **viewWillAppear()**

Is overridden to configure view created by loading an interface file. This method is called every time view controller is moved onscreen.

## **init(coder:)**

Initializes UIViewController instances created from a storyboard

## **init(nibName:bundle:)**

designated initializer for UIViewController (for instances created without storyboard, programmatically). Called once on each viewController as it is created.

## **loadView()**

Is overridden to create a view controller's view programmatically

## **viewDidLoad()**

Is overridden to configure views created by loading an interface file. This method is called after the view controller is created.

## **viewWillAppear()**

Is overridden to configure view created by loading an interface file. This method is called every time view controller is moved onscreen.