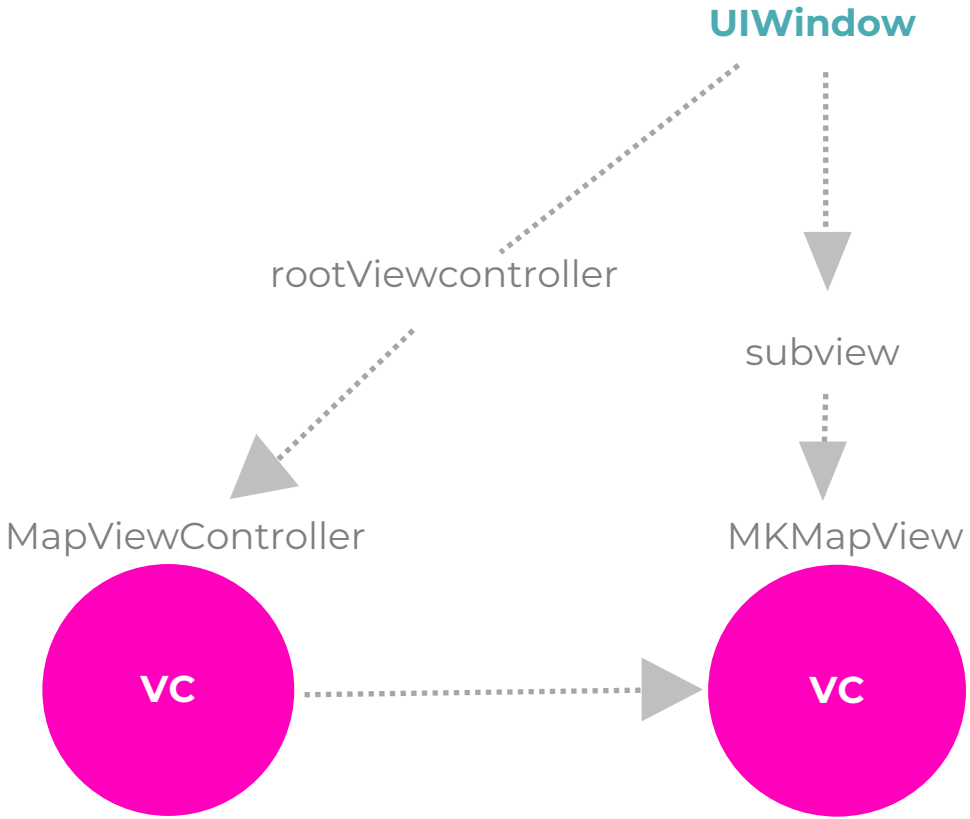
View Controllers

MK Map View

Daria Tsoupikova
Sabine Krauss

View Controller

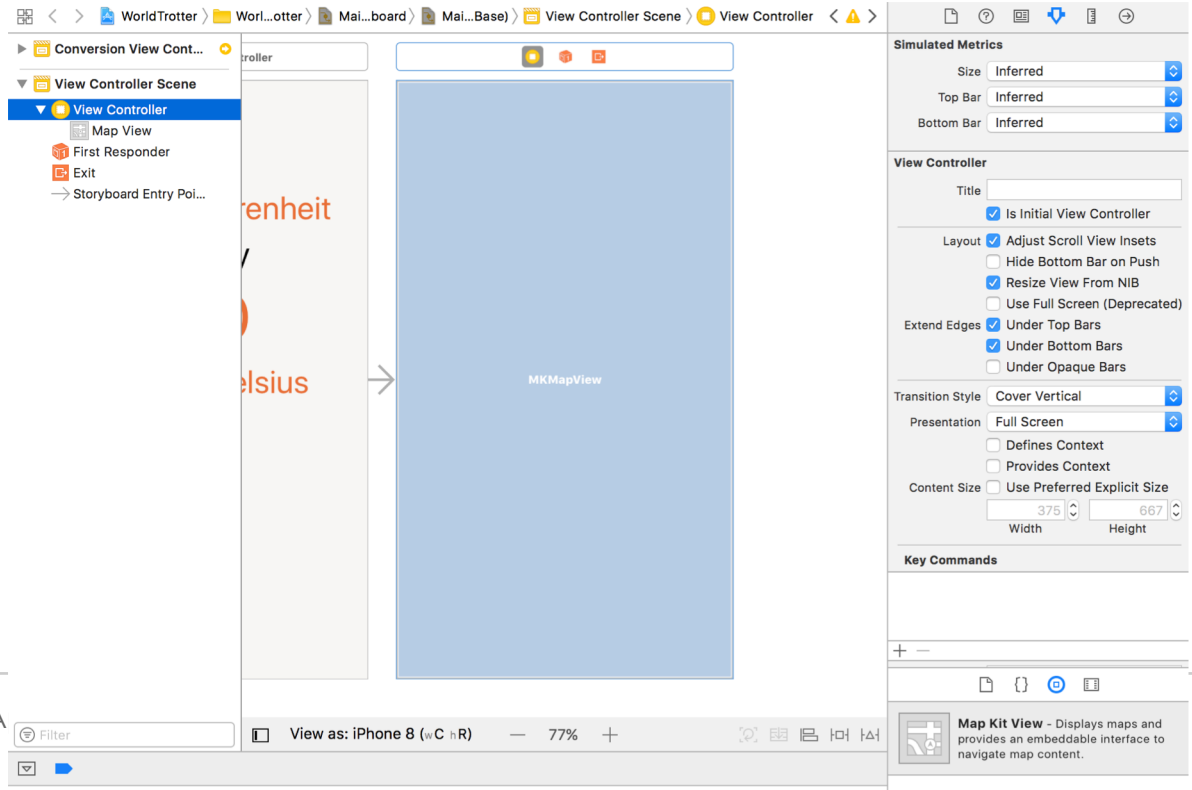
Manages view hierarchy
Creates view objects
Handles events associated with view objects



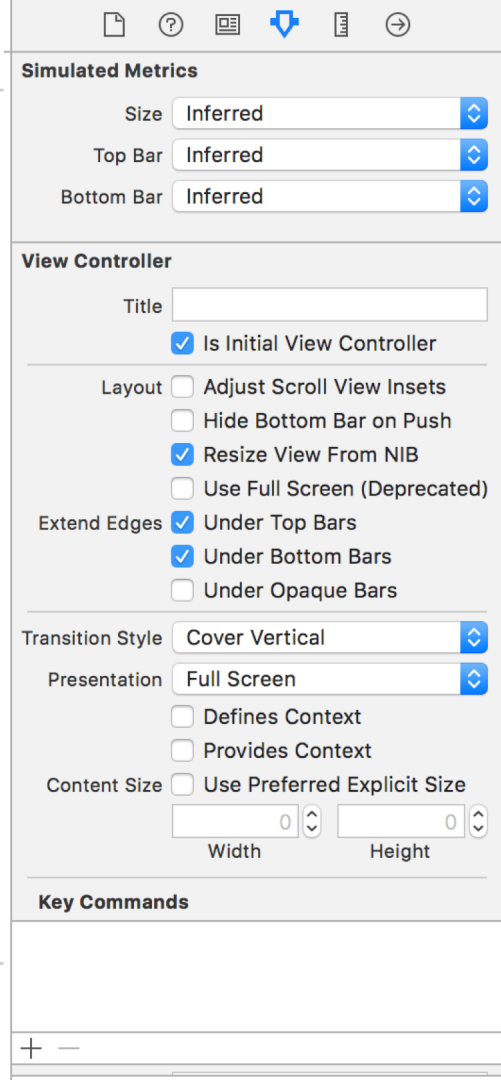
2 ways to create ViewController:
1. Using storyboard in IB
2. Programmatically overriding method loadView()

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

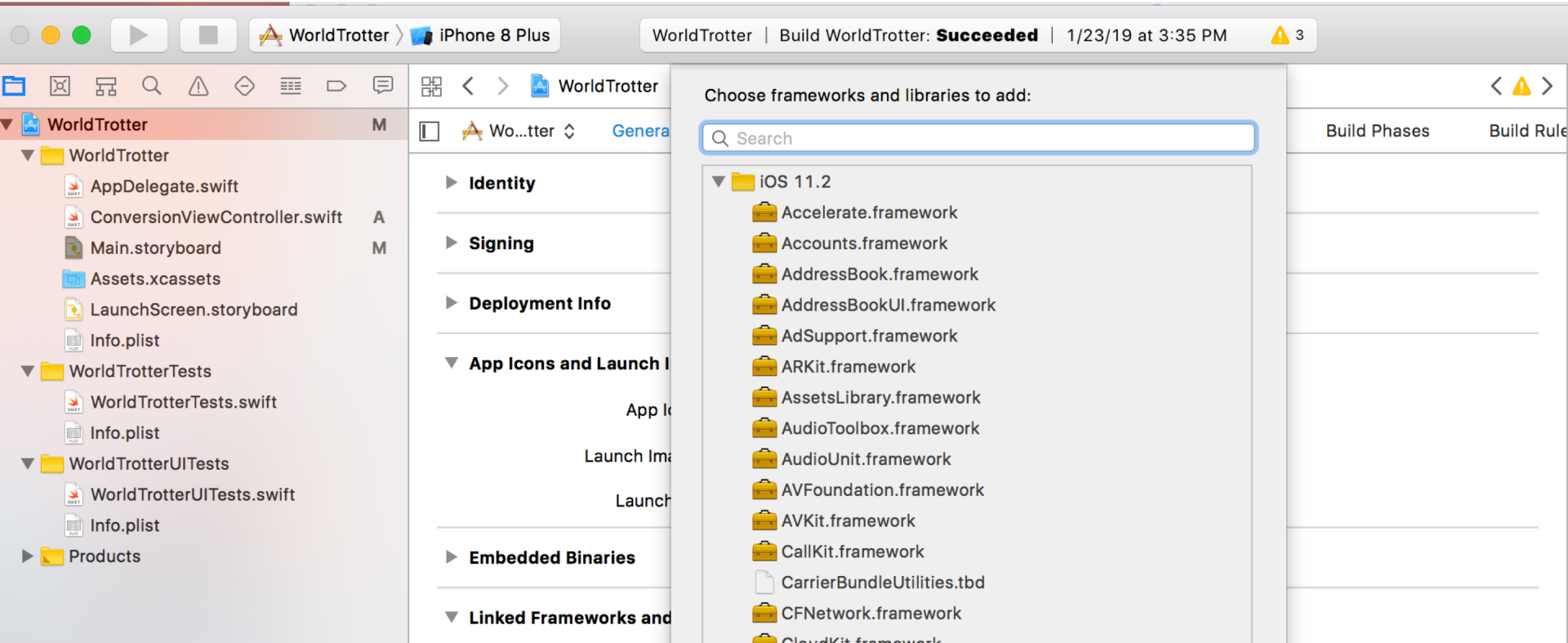
Open Main.storyboard.
Drag View Controller onto canvas. Select the View of the ViewController – Delete.
Drag Map Kit View from object library (search) onto controller to set map view.



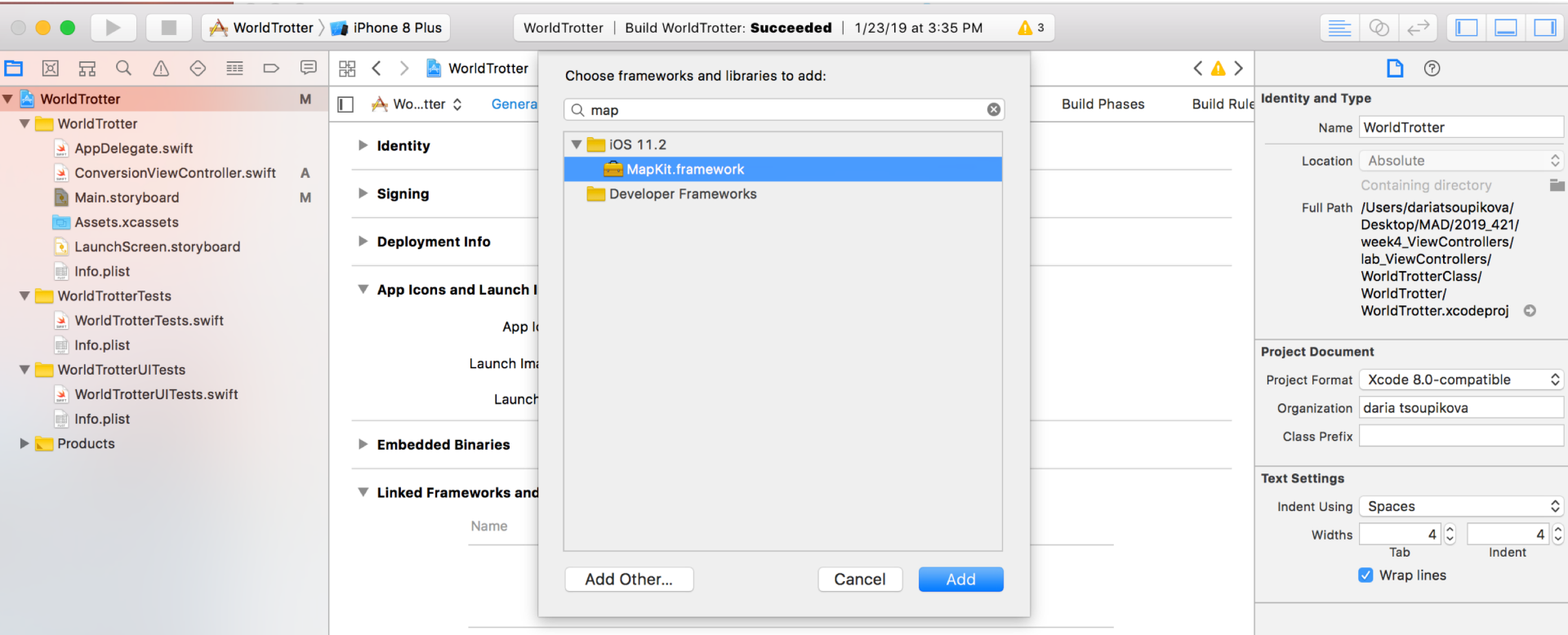
Select View Controller and open Attributes inspector.
Check Is Initial View Controller
Notice the change of direction of the gray arrow
Now it points to View Controller now, not Conversion View Controller.
Test and run the app.



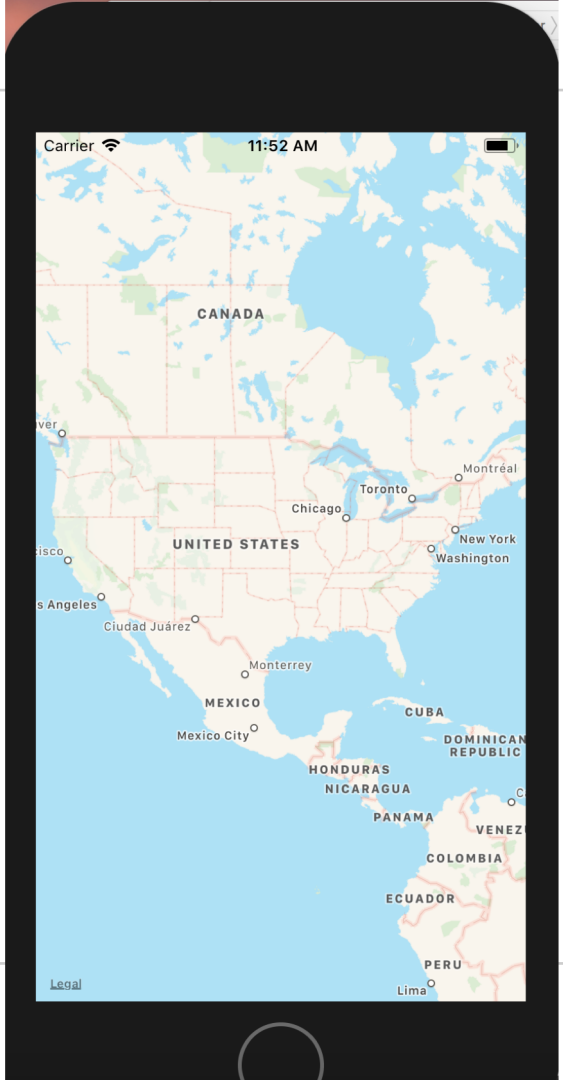
Import MapKit framework and link it to the view to load map view.
WorldTrotter project > Project Settings > General > Linked Frameworks and Libraries
Check + sign and search for MapKit framework. Click Add. Build and run to test map view.



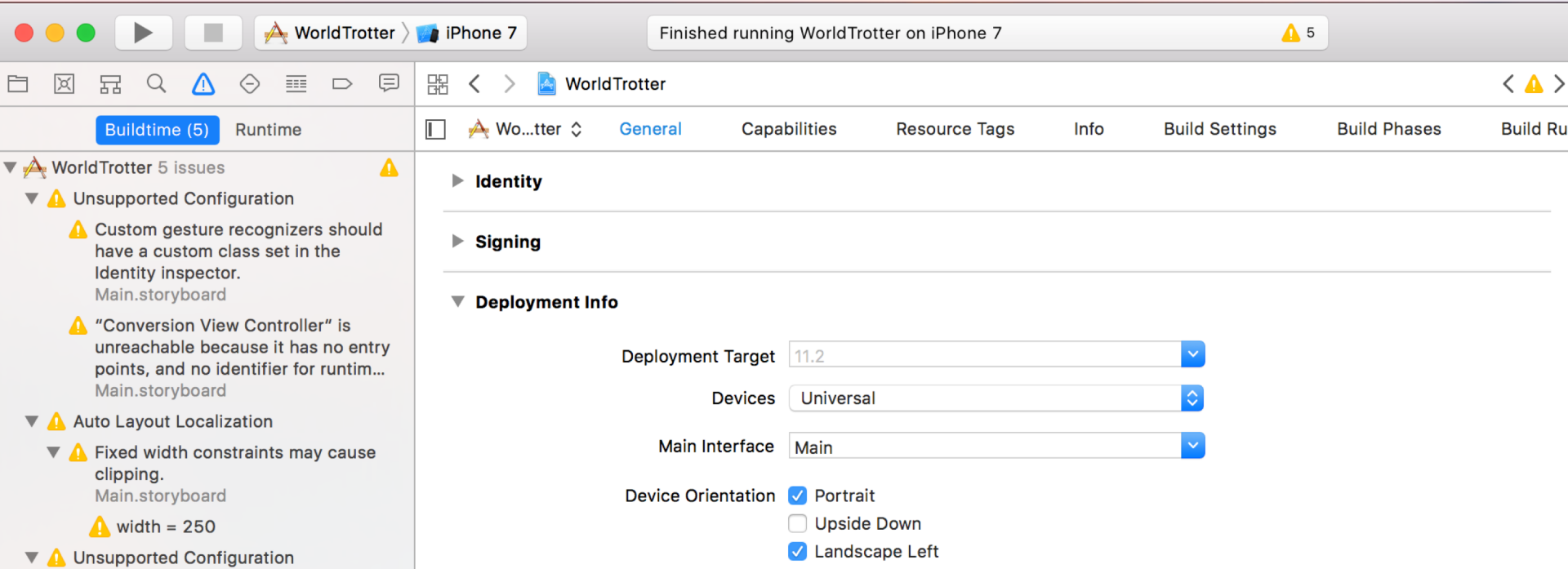
Import MapKit framework and link it to the view to load map view.
WorldTrotter project > Project Settings > General > Linked Frameworks and Libraries
Check + sign and search for MapKit framework. Click Add. Build and run to test map view.



Build and run to test map view.



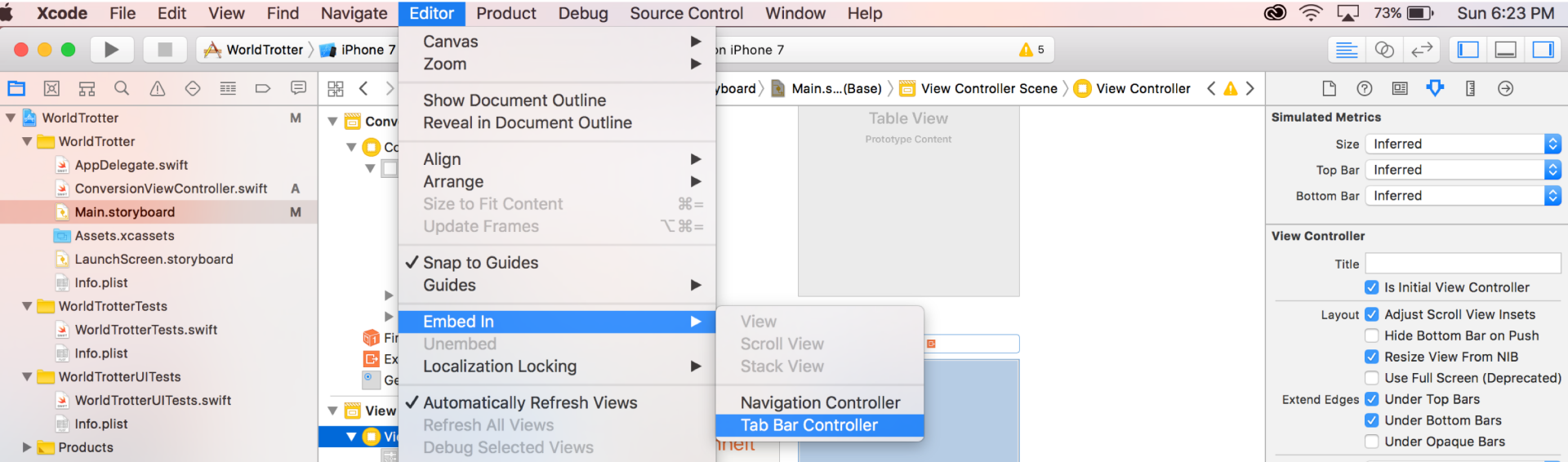
Each app has one main interface (main.storyboard)
When the app launches, the initial view controller gets set as the rootViewController of the window.
The main interface is set in the project settings >General > Deployment >Main Interface
Main corresponds to Main. Storyboard.



UITabBarController allows swap between the View Controllers.
Keeps array of view controllers
Maintains a menu to select view controllers

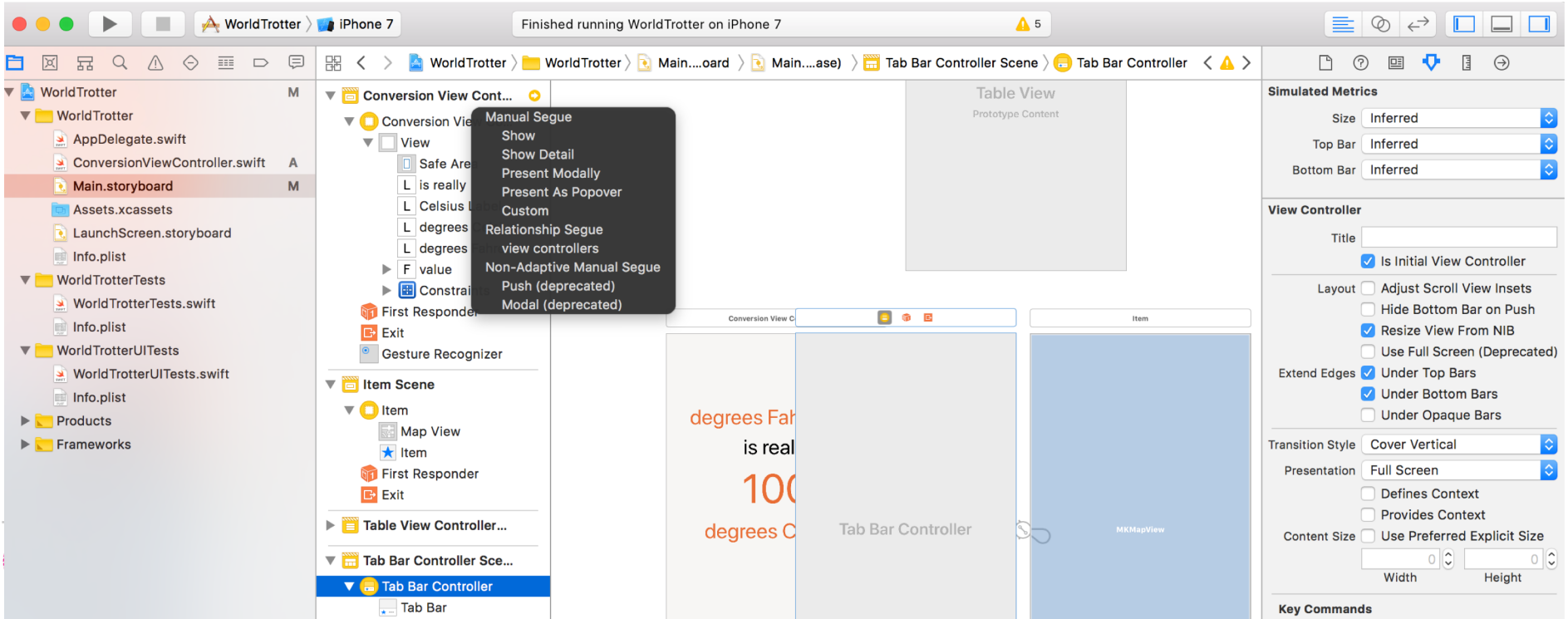
Open Main.storyboard> select View Controller

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

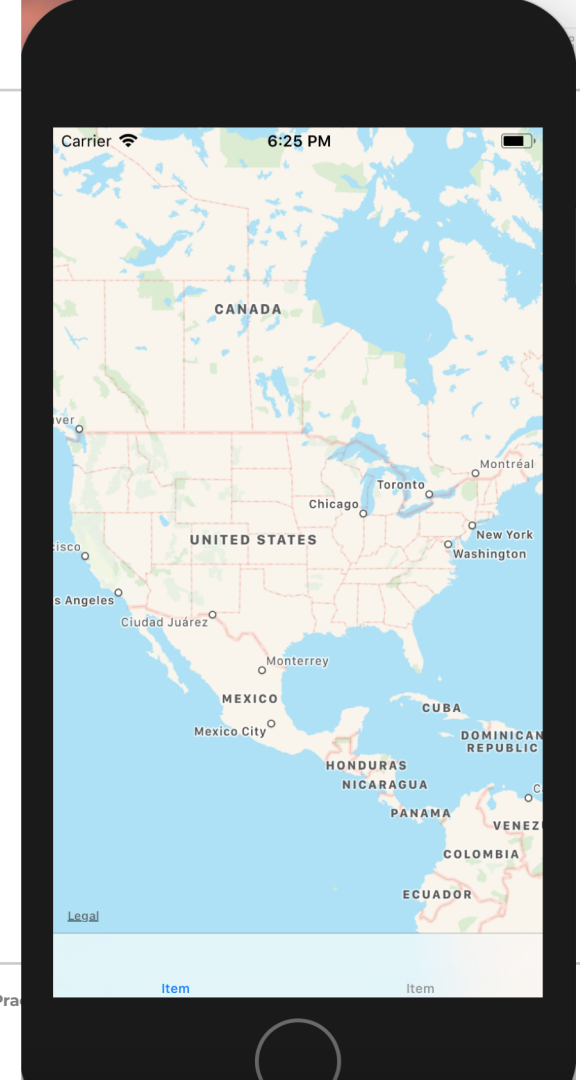


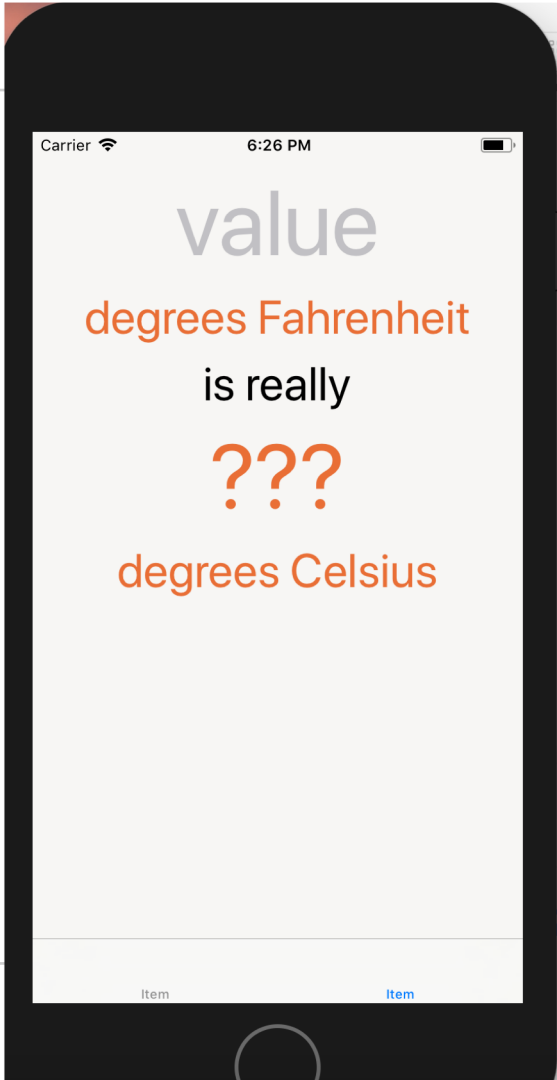
Add Conversion ViewController to TabBarController to switch between the screens:

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

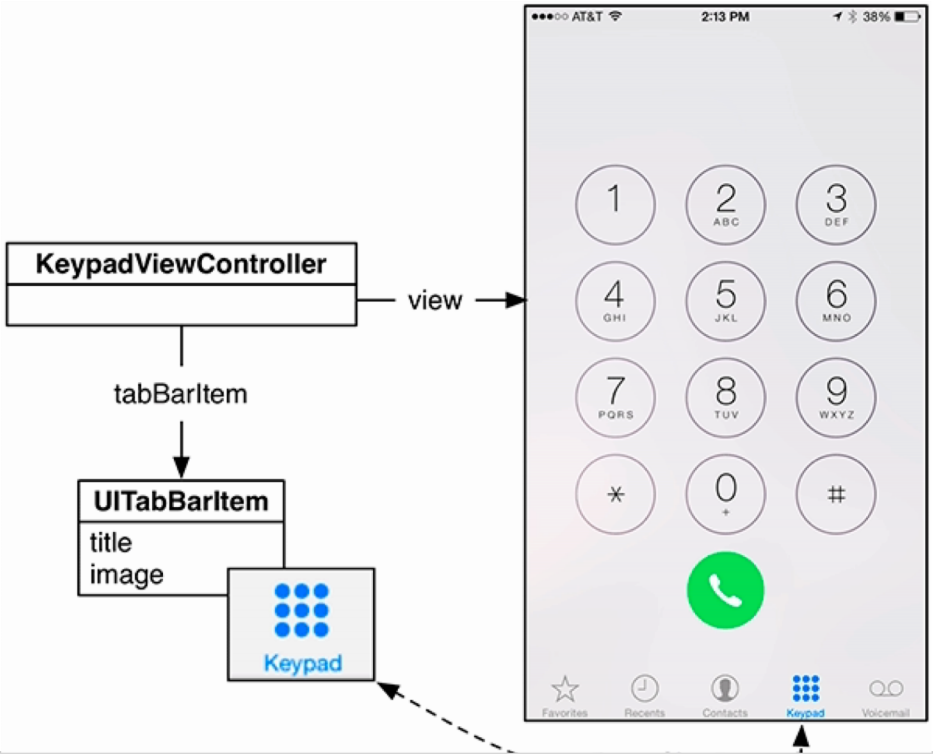


Tab menu bars to switch between the screens:





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



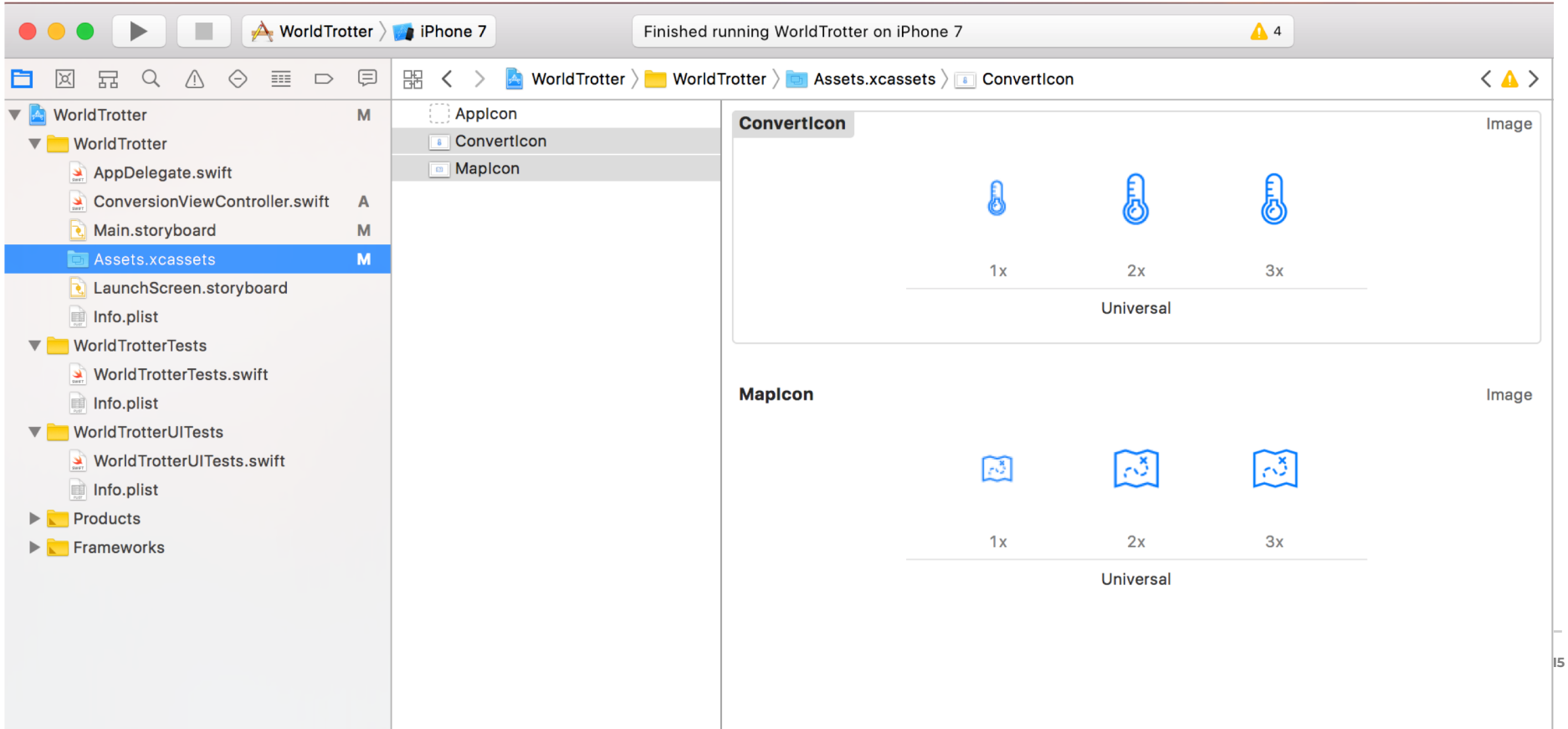
Add image by adding it to assets
Assets are files to be selected at runtime

Open Assets.xcassets

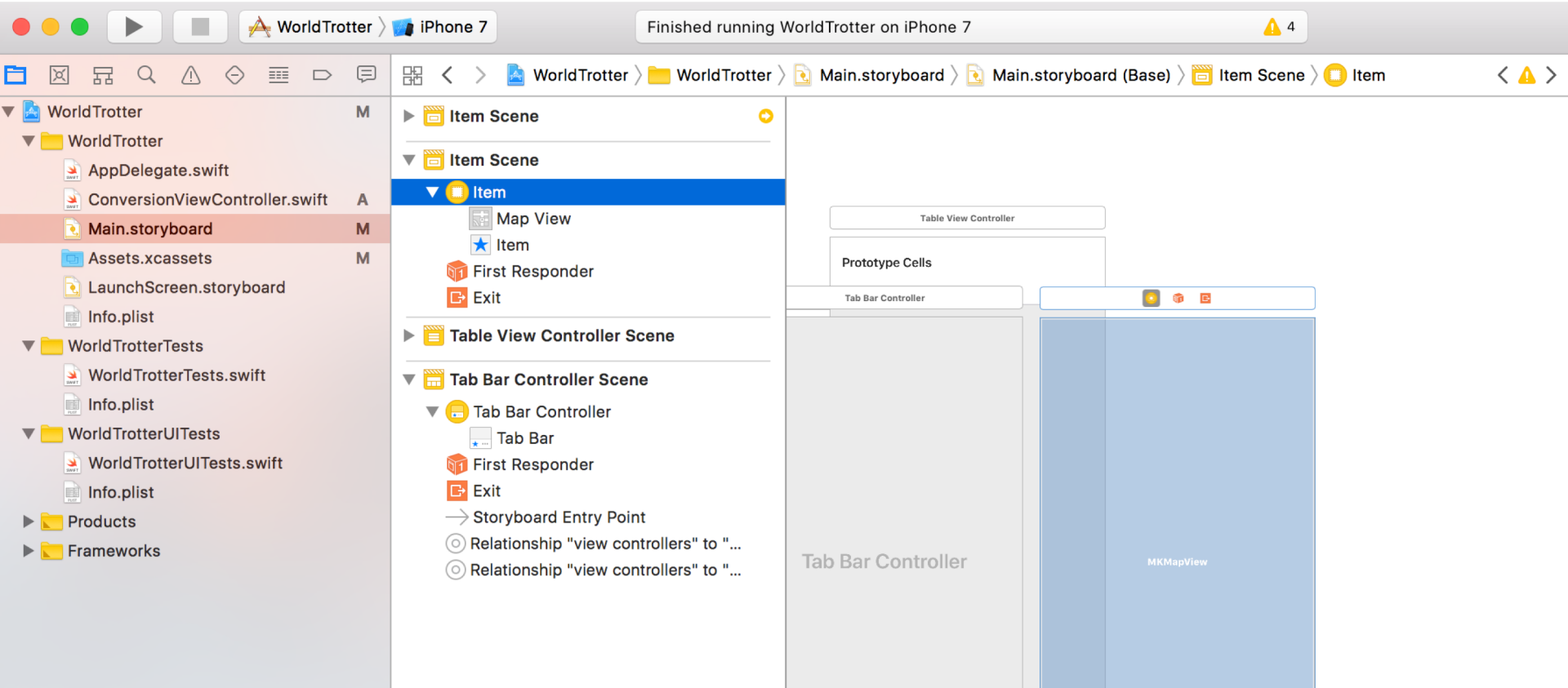
download from
https://www.bignerdranch.com/spolutions/iOSProgramming6ed.zip

Under Ch. 5 Xcode Assets
ConvertIcon.png
ConvertIcon@2x.png
ConvertIcon@3x.png
MapIcon.png
MapIcon@2x.png
MapIcon@3x.png

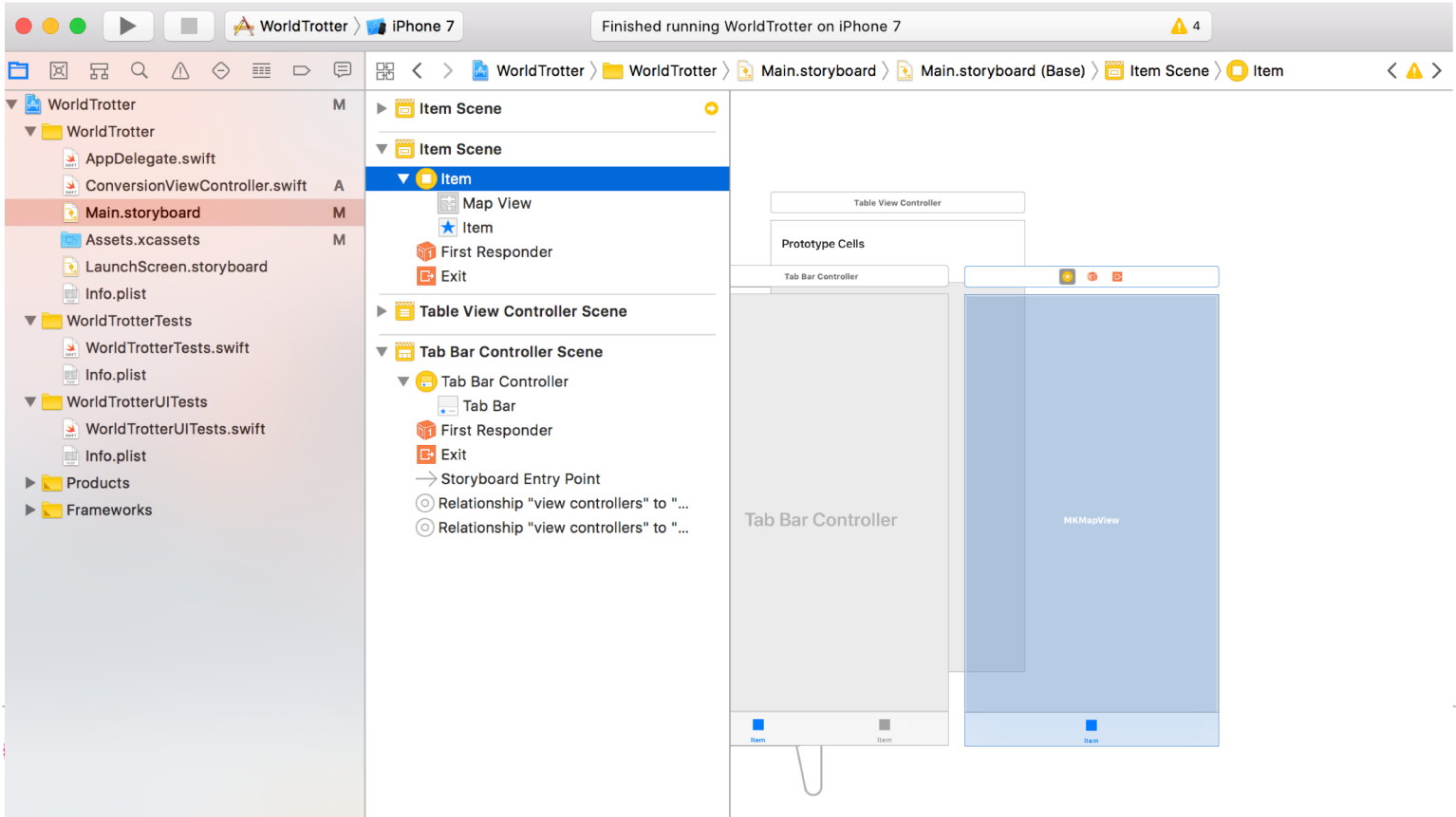
Drag these images to images set list of the Assets Catalog.



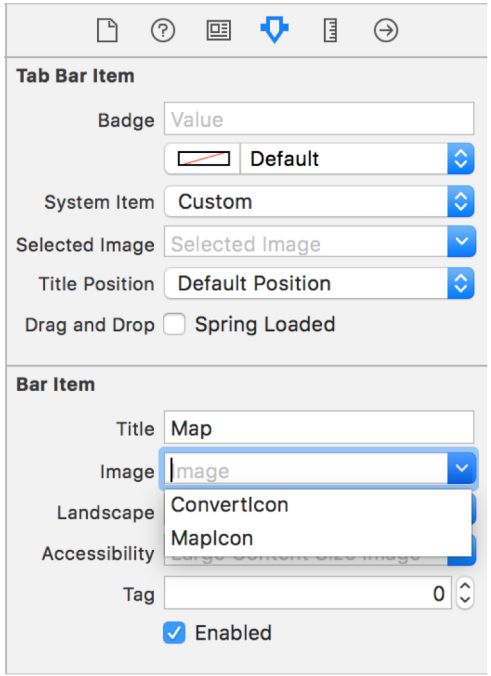
In Main.storyboard select ViewController – it is named Item now

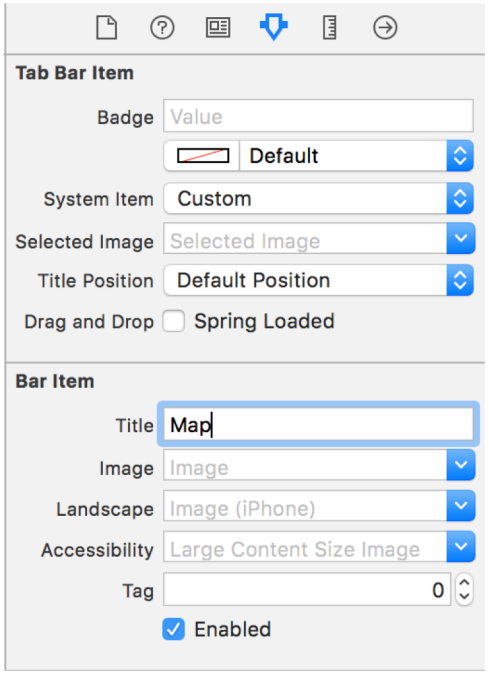


Notice the tab bar is added to the bottom of the Interface

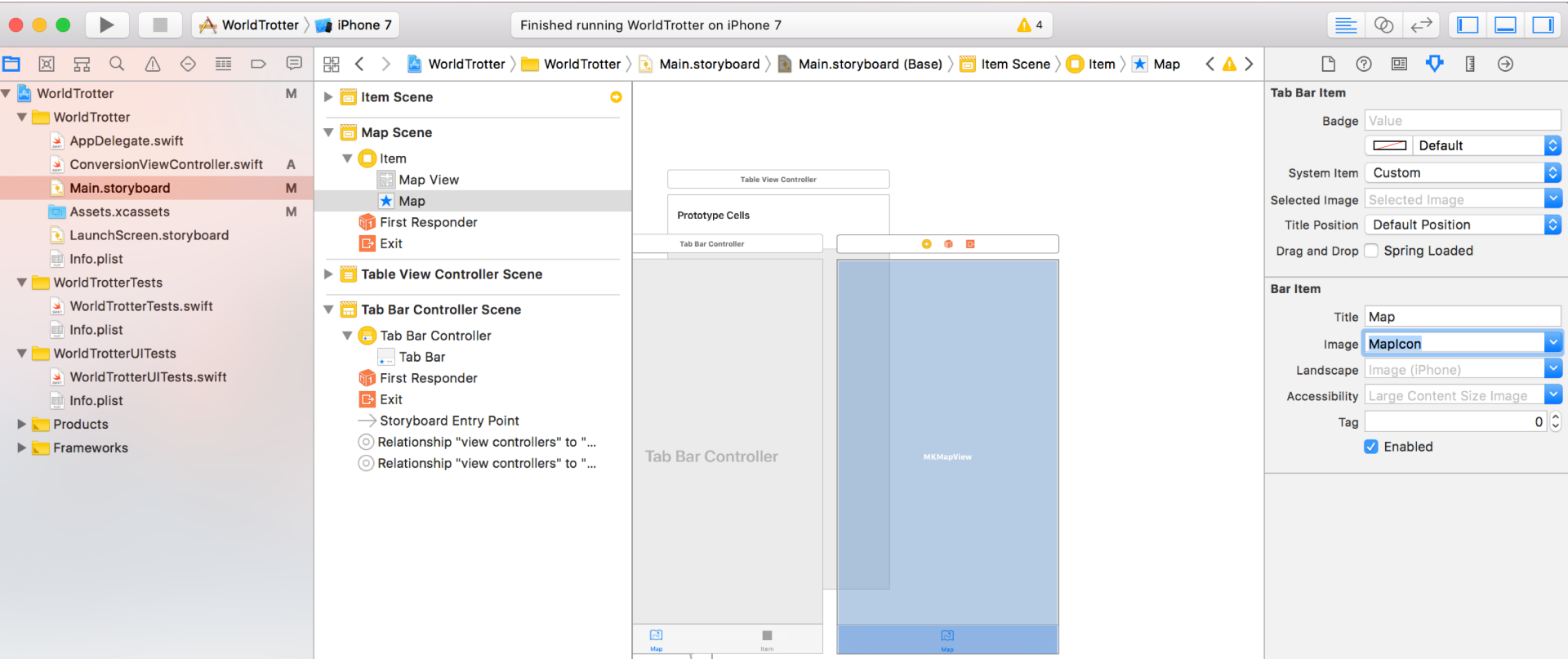


Select this tab bar and open attributes
Change name to Map
Choose MapIcon as Image

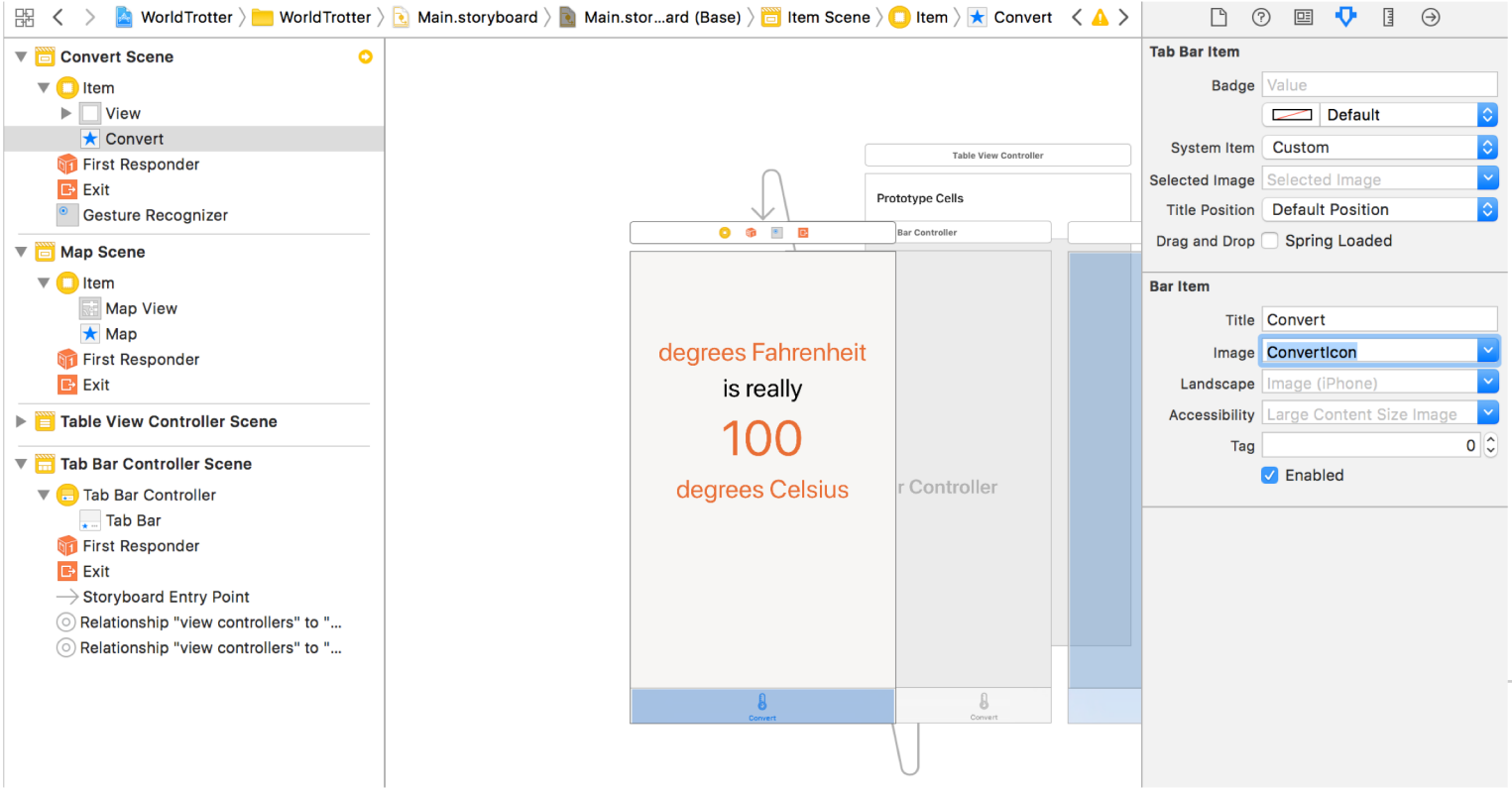




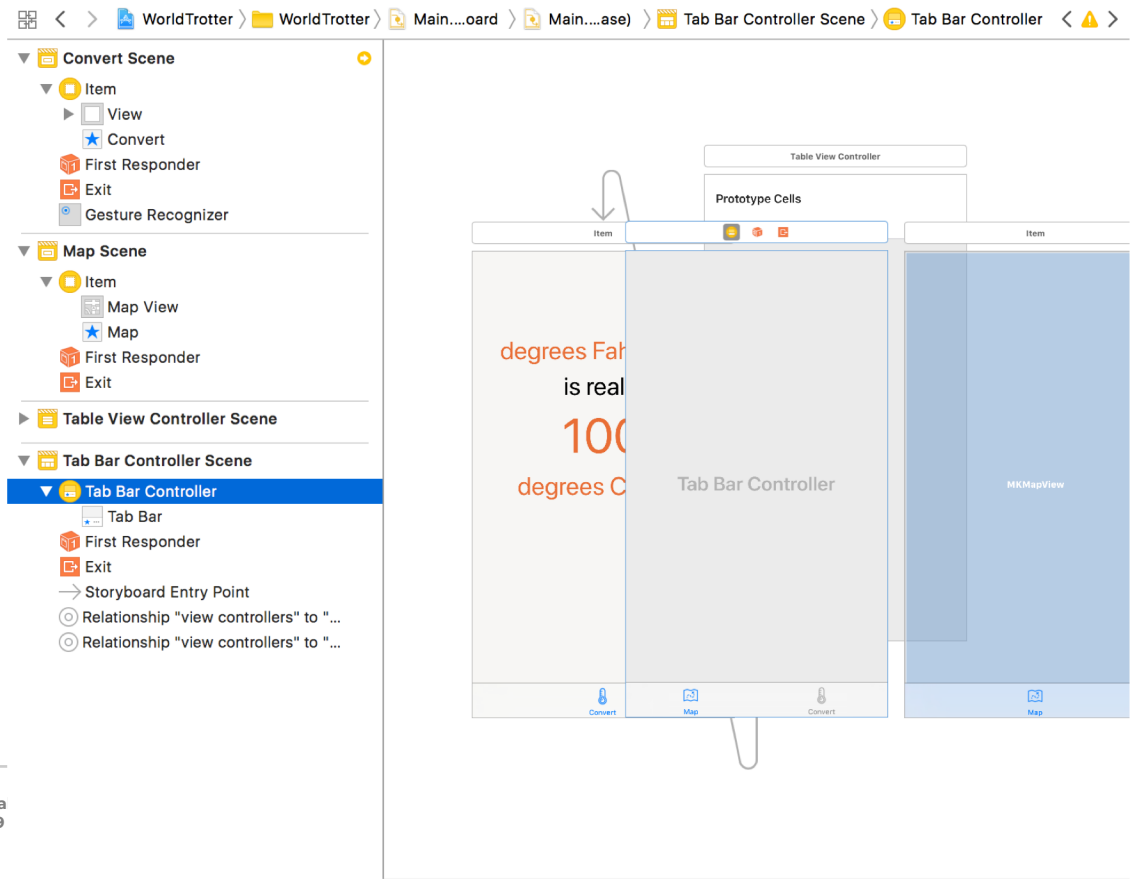
Select this tab bar and open attributes
Change name to Map
Choose MapIcon as Image



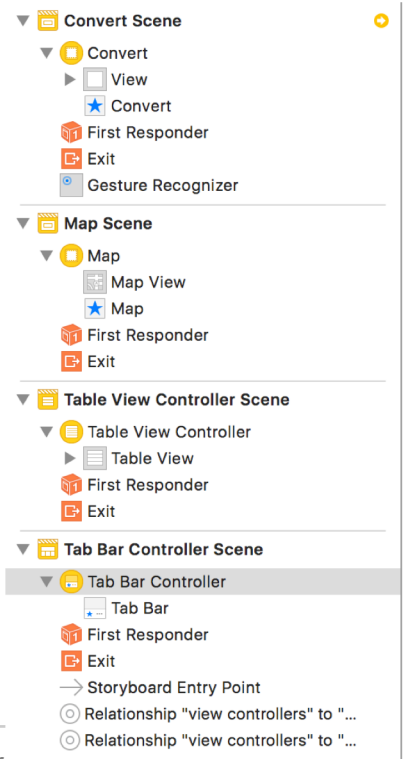
Select tab bar in ConversionViewController
Change name to “Convert”
Choose ConvertIcon as Image

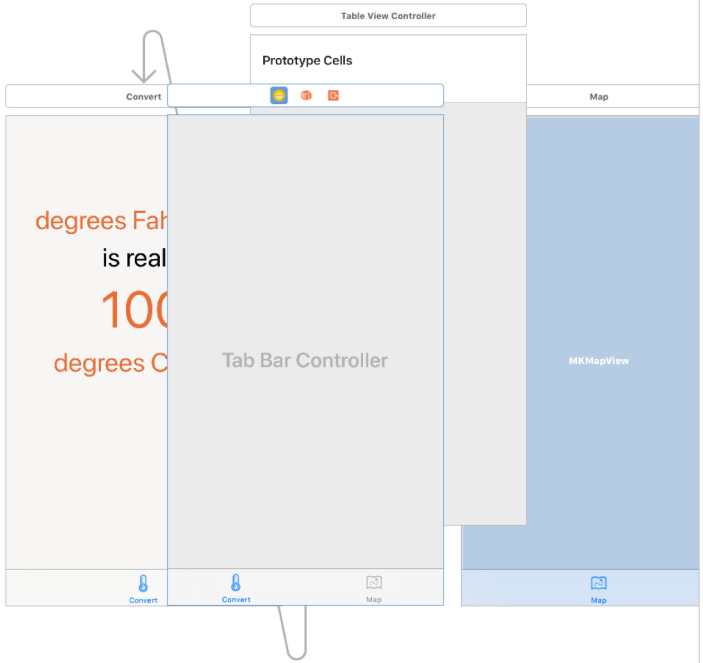


Select Tab Bar Controller on canvas (or in storyboard)
Drag the Convert tab to be on the left

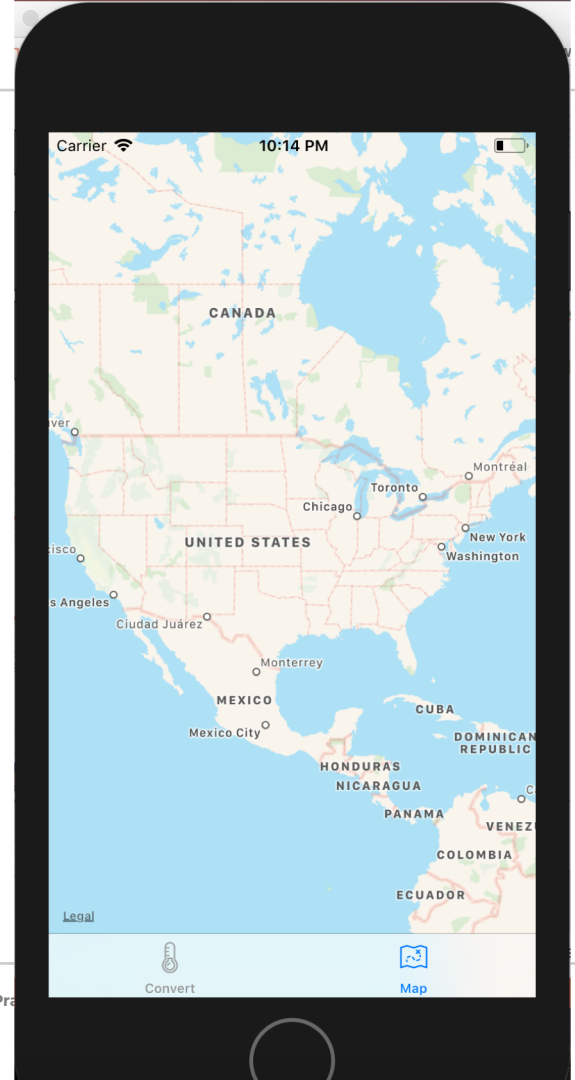


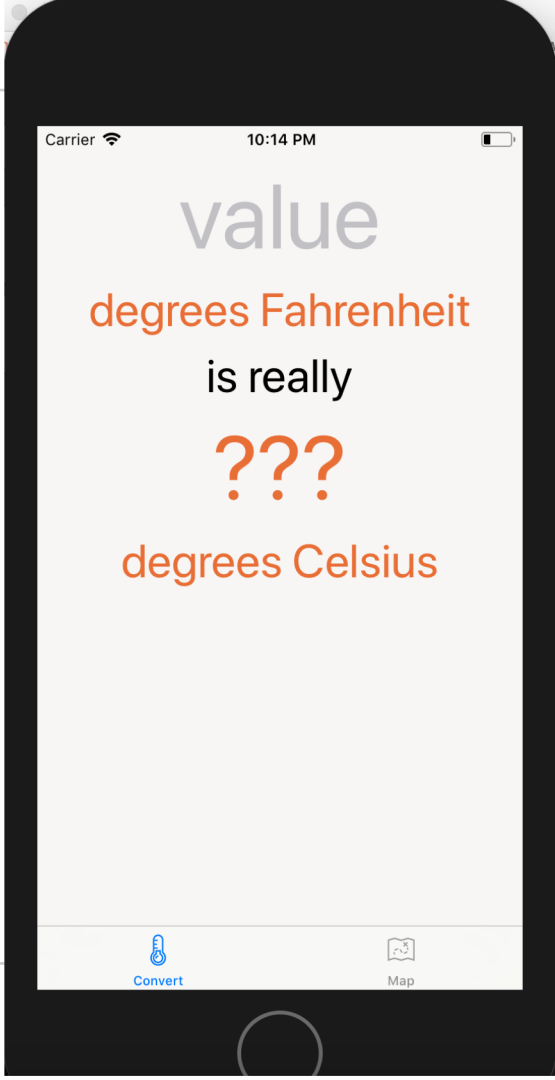
Select Tab Bar Controller on canvas (or in storyboard)
Drag the Convert tab to be on the left





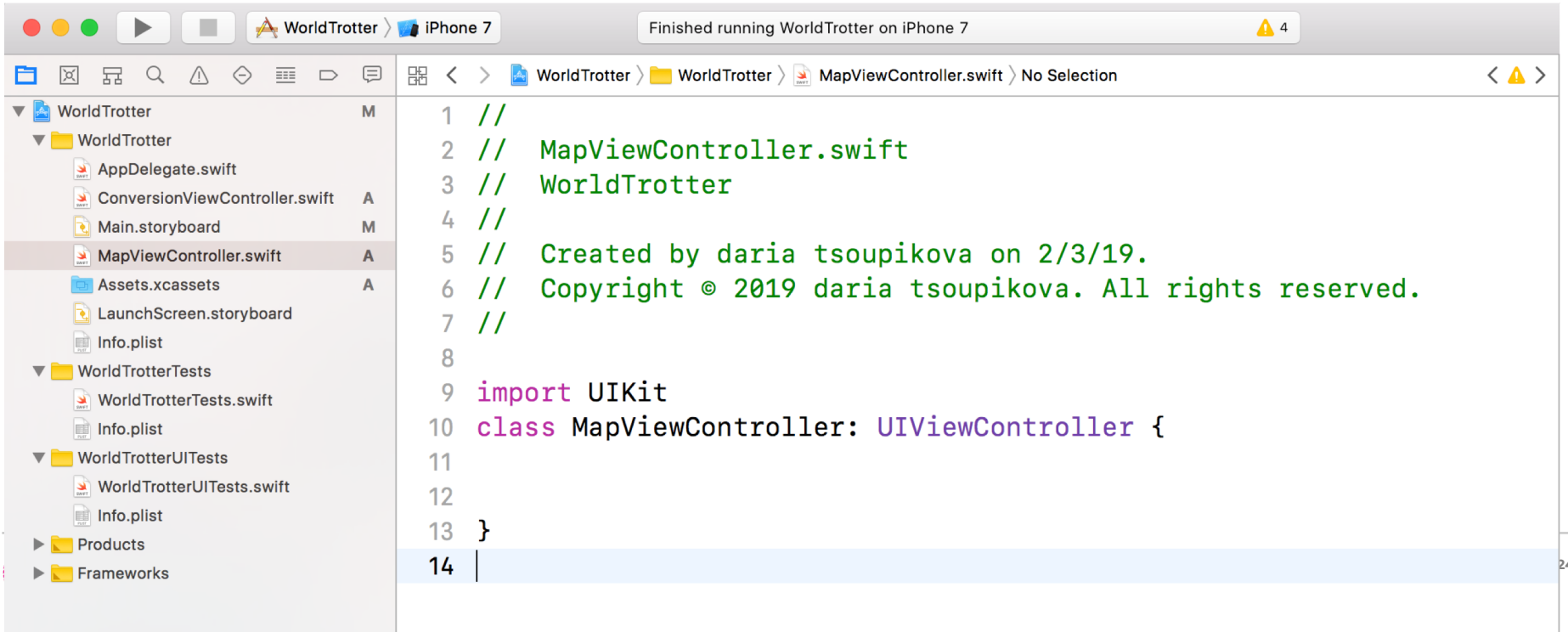
Build, run and test



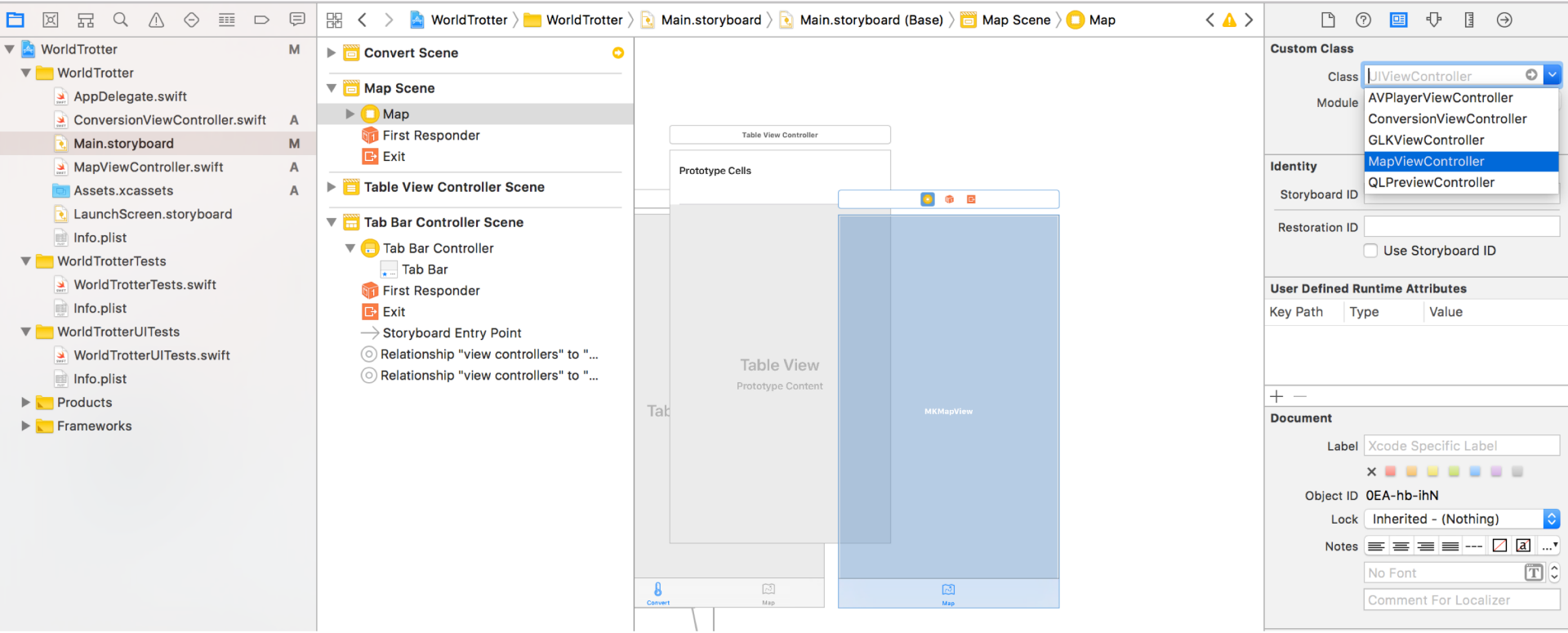


MapViewController is not needed until user taps on the menu item

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

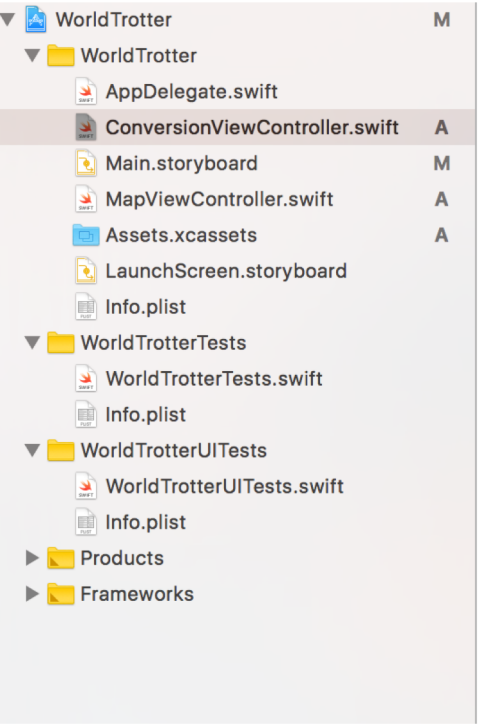
In ConversionViewController.swift

Update viewDidLoad()

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

print("ConversionViewController loaded its view")

updateCelsiusLabel()

}
```

In ConversionViewController.swift

Update viewDidLoad()

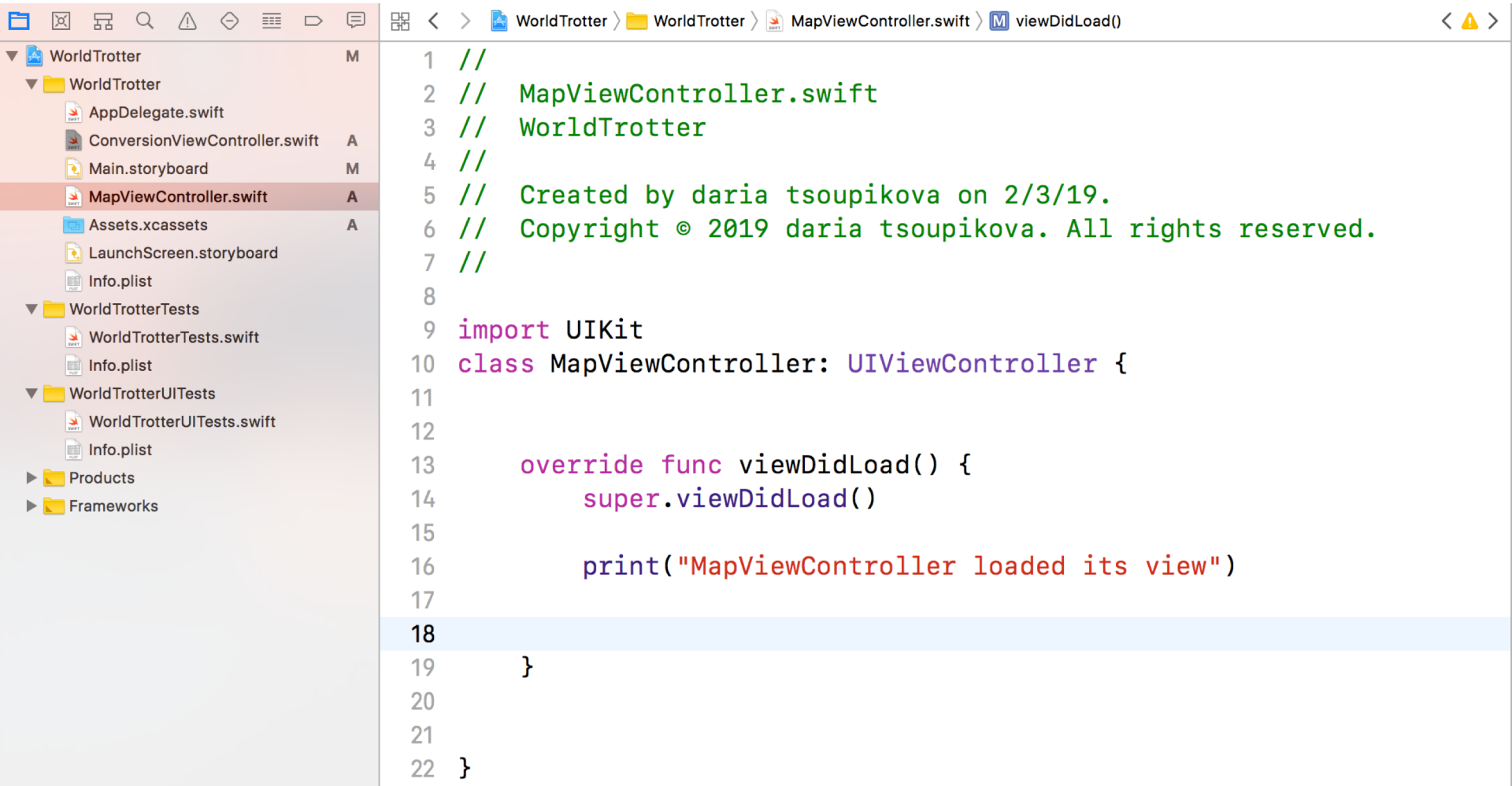



```
69              fahrenheitValue = nil
70          }
71      }
        @IBAction func dismissKeyboard(_ sender:
            UITapGestureRecognizer) {
73          textField.resignFirstResponder()
74      }
75
76
77      override func viewDidLoad() {
78          super.viewDidLoad()
79
80          print("ConversionViewController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

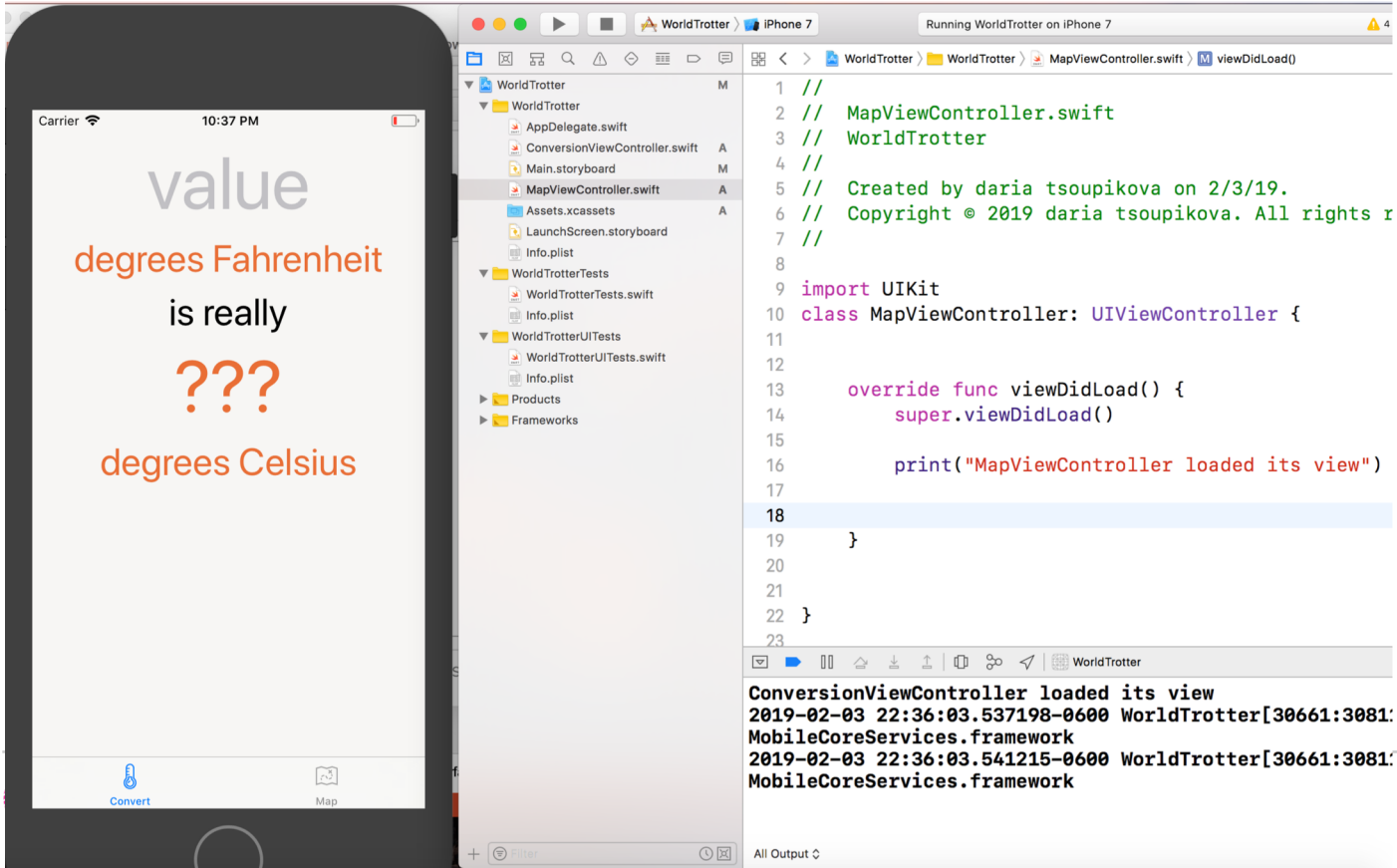
In MapViewController.swift

Update viewDidLoad()

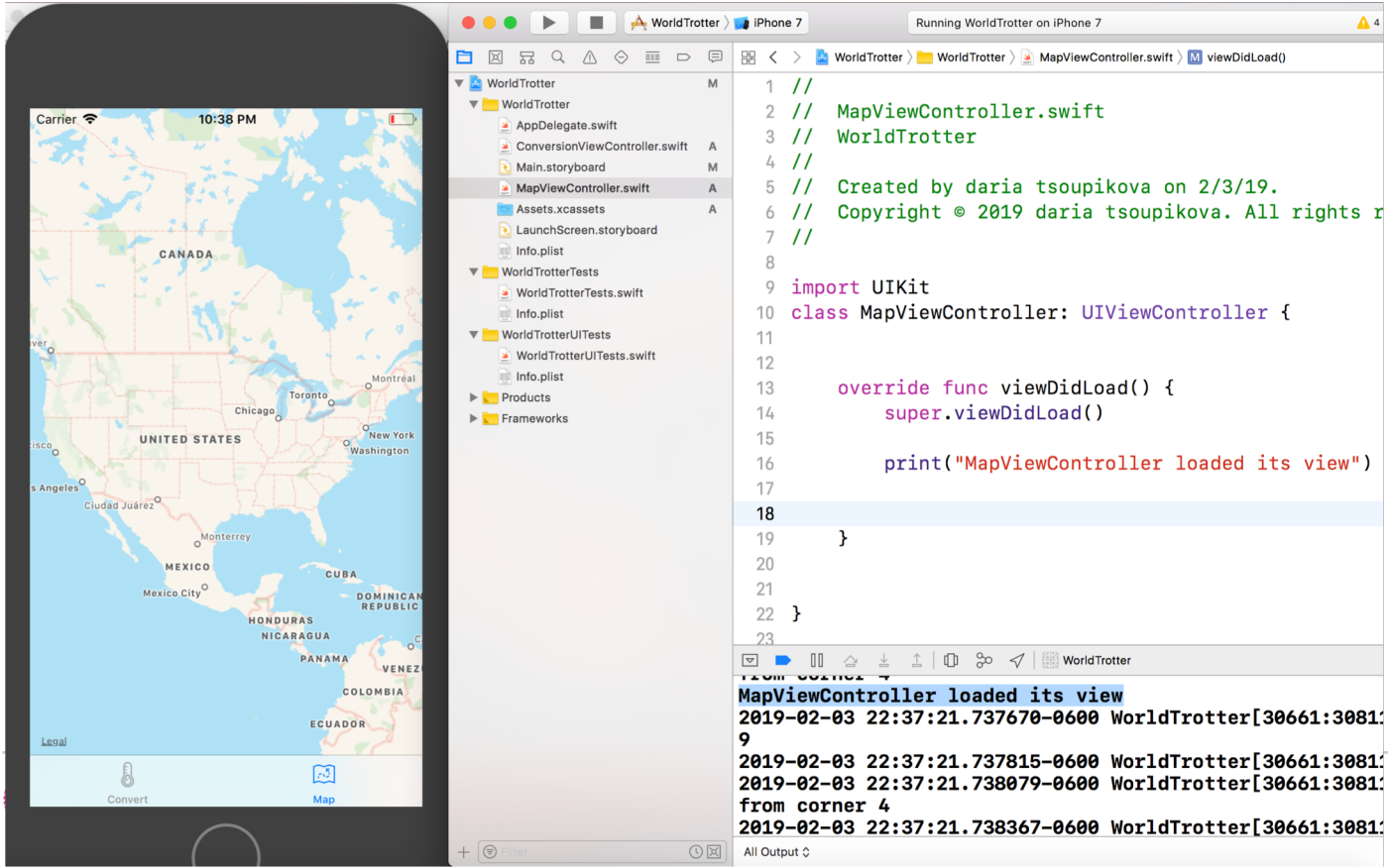
```
override func viewDidLoad() {
super.viewDidLoad()

print("MapViewController loaded its view")


}
```


WorldTrotter
WorldTrotter
AppDelegate.swift
ConversionViewController.swift A
Main.storyboard M
MapViewController.swift A
Assets.xcassets A
LaunchScreen.storyboard
Info.plist
WorldTrotterTests
WorldTrotterTests.swift
Info.plist
WorldTrotterUITests
WorldTrotterUITests.swift
Info.plist
Products
Frameworks
WorldTrotter WorldTrotter MapViewController.swift viewDidLoad()
//
// MapViewController.swift
// WorldTrotter
//
// Created by daria tsoupikova on 2/3/19.
// Copyright © 2019 daria tsoupikova. All rights reserved.
//
import UIKit
class MapViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
print("MapViewController loaded its view")
}
}

Build, run and text.



Build, run and text.



Lazy loading

2 methods:

- Override viewDidLoad() method

- This method is called after the view controller’s interface file is loaded.

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

- Override viewWillAppear(_:) method

- This method is called just before a controller’s view is added to the window.

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 s 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 n interface file. This methods is called after the view controller is created.

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

init(coder:)
Initializes UIViewController instances created from s 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 n interface file. This methods is called after the view controller is created.

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