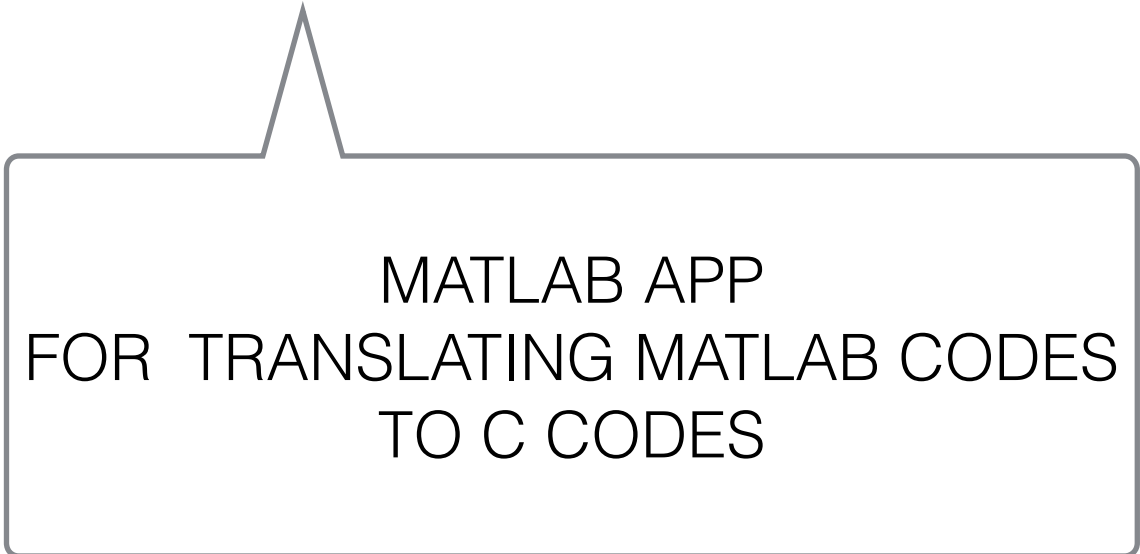


# SWIFT and MATLAB

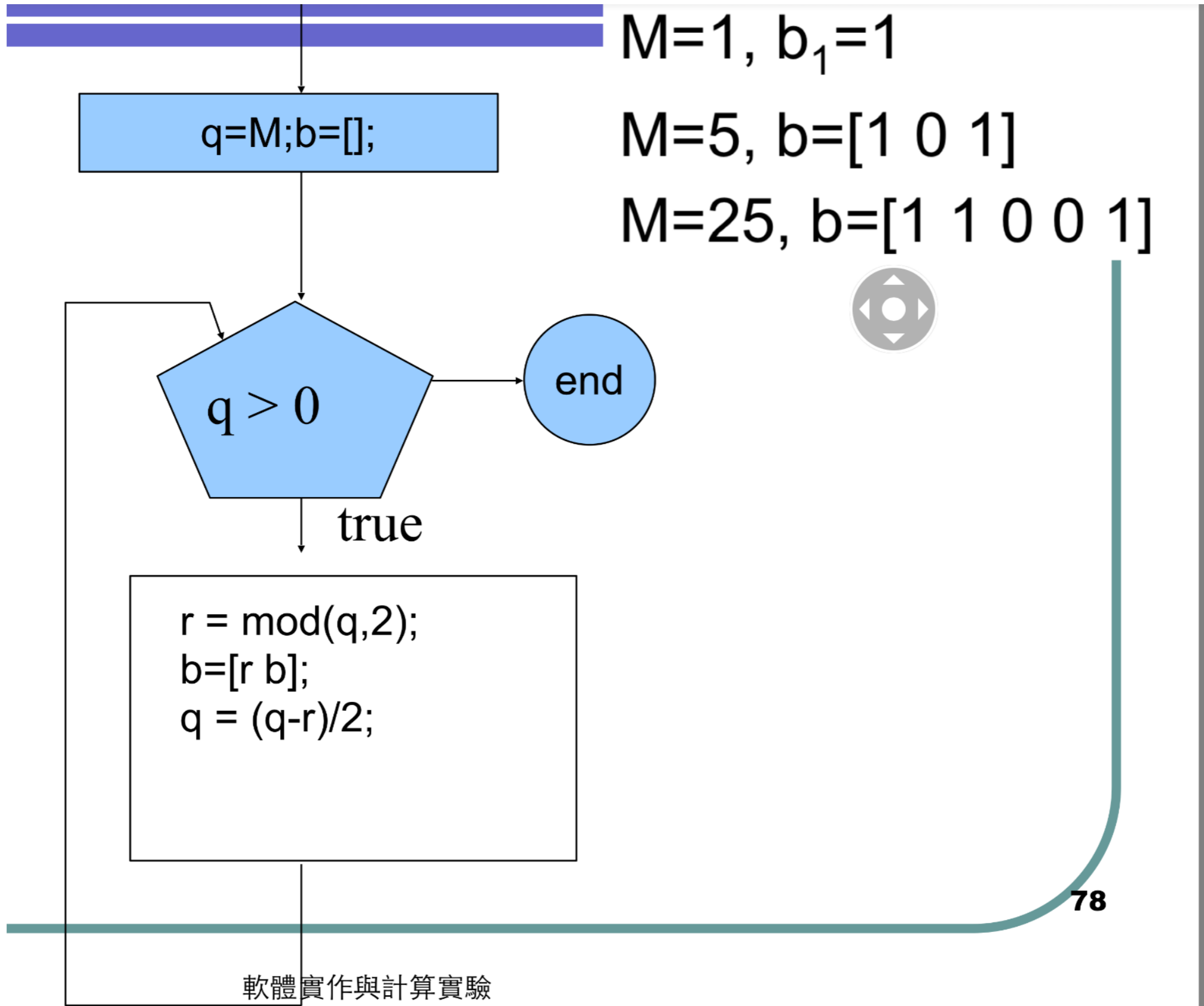
2015

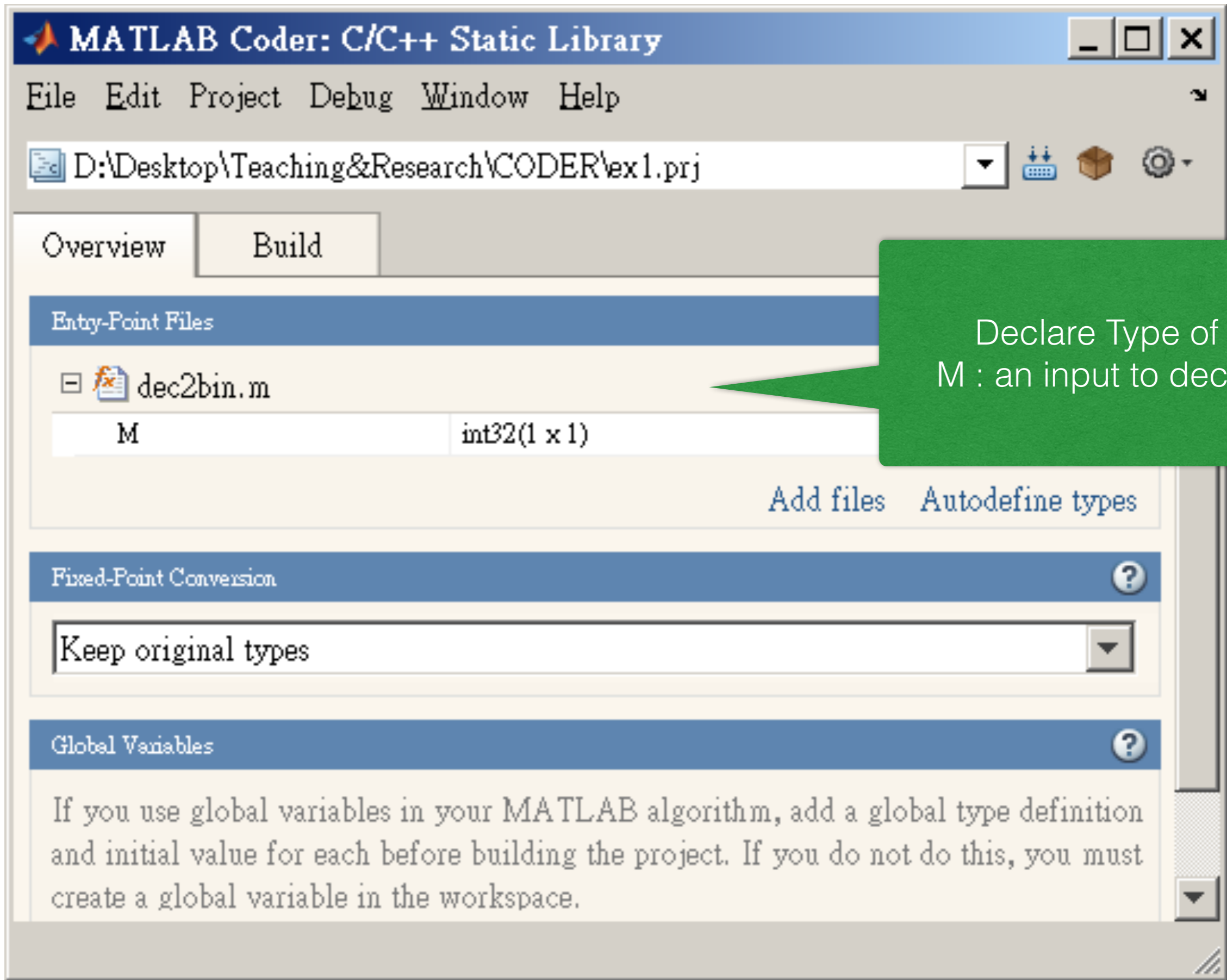
LAB326 AM NDHU

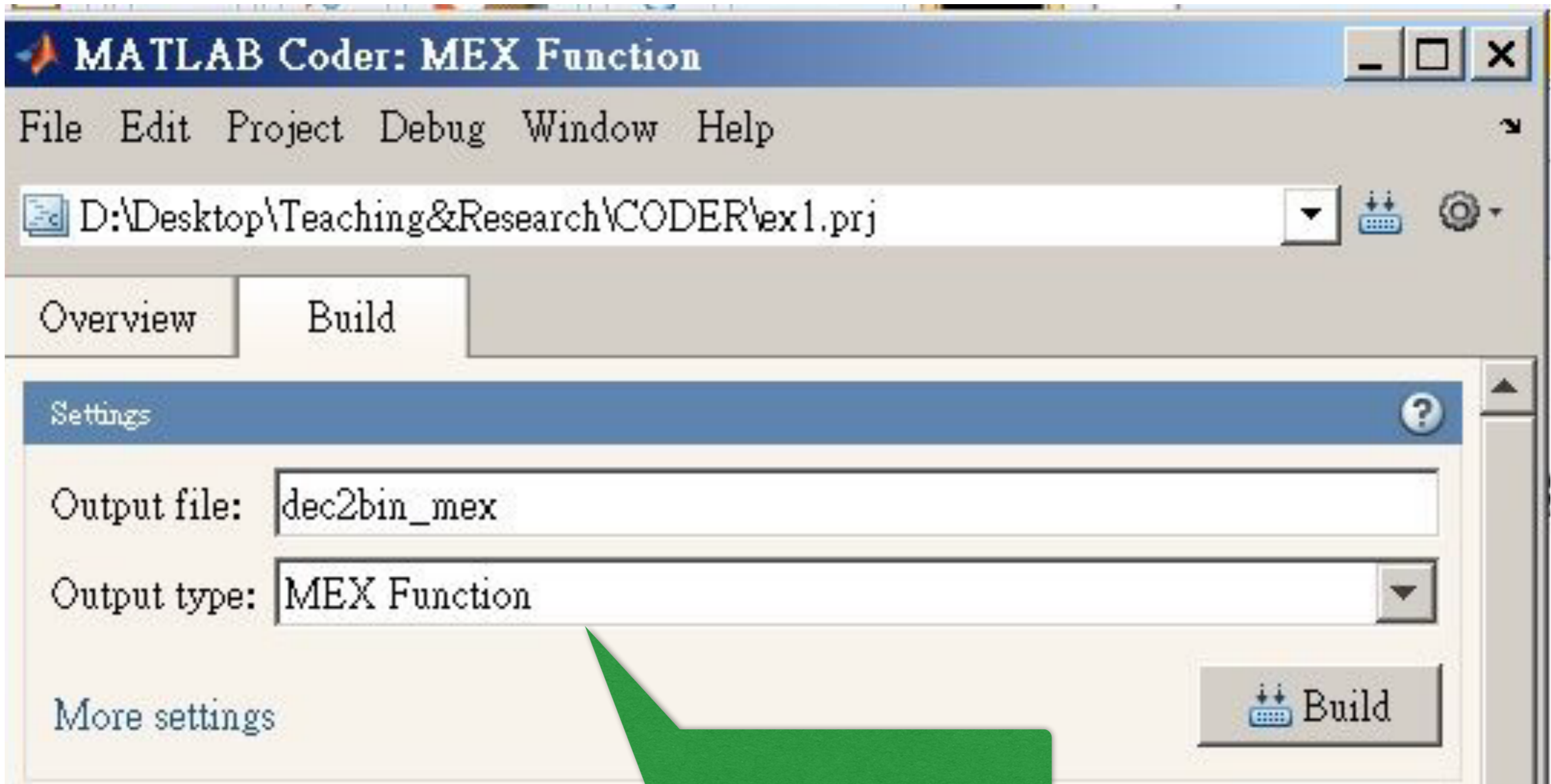
# MATLAB CODER



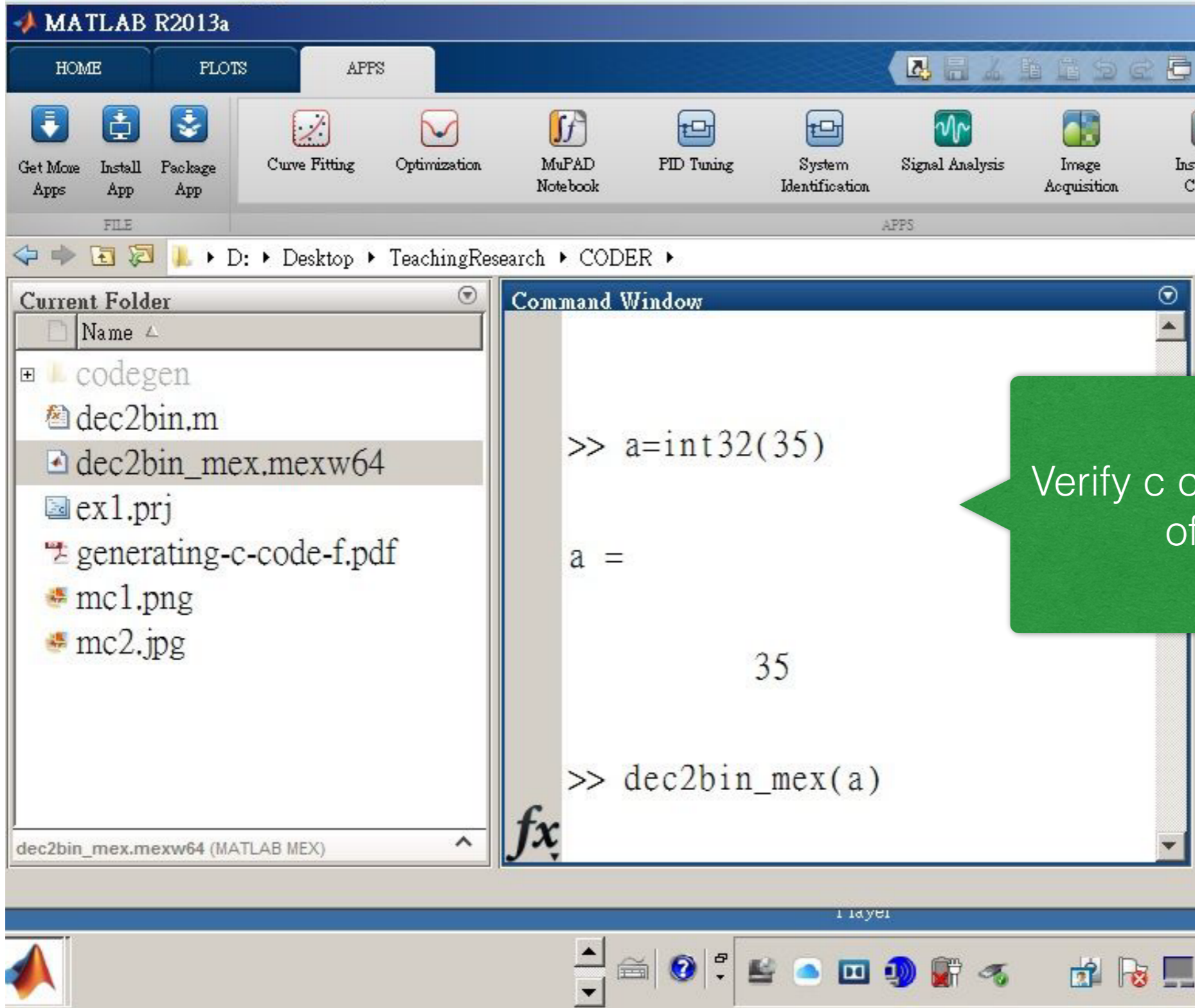
MATLAB APP  
FOR TRANSLATING MATLAB CODES  
TO C CODES







MATLAB EXECUTABLE  
CODES



Verify c codes in form of MEX

SWIFT PROGRAMMING

PDFS

SWIFT CALLS C



×



PLAYGROUND

# Welcome to Xcode

Version 6.0.1 (6A317)



**Get started with a playground**  
Explore new ideas quickly and easily.



**Create a new Xcode project**  
Start building a new iPhone, iPad or Mac application.



**Check out an existing project**  
Start working on something from an SCM repository



Show this window when Xcode launches



**MyPlayground.playground**

~/Desktop



**ex1**

~/Desktop



**jm**

~/Desktop

Open another project...

MyPlayground.playground — Edited

MyPlayground.playground > No Selection

```
// Playground – noun: a place where
    people can play

import UIKit

var str = "Hello, playground"
var A=20
let B=50
A=A+B
println(str)
var a=1
a+=A
println(str)
str=str+String(a)
var loop=0
for var i=0; i<39; ++i {
    loop+=i
}
```

SWIFT

```
"Hello, playground"
20
50
70
"Hello, playground"
1
71
"Hello, playgro...
"Hello, playgroun...
0
(39 times)
```

APP

The image shows a screenshot of the Xcode IDE. The top toolbar includes 'Run' and 'Stop' buttons. The left sidebar shows a project named 'ex1' with a file tree containing 'AppDelegate.swift', 'ViewController.swift', 'Main.storyboard', 'Images.xcassets', 'LaunchScreen.xib', 'Supporting Files', 'ex1Tests', and 'Products'. The main editor window displays the Swift code for 'AppDelegate.swift'. The code includes a comment 'Override point for customization after application launch.' and a function 'applicationWillResignActive'.

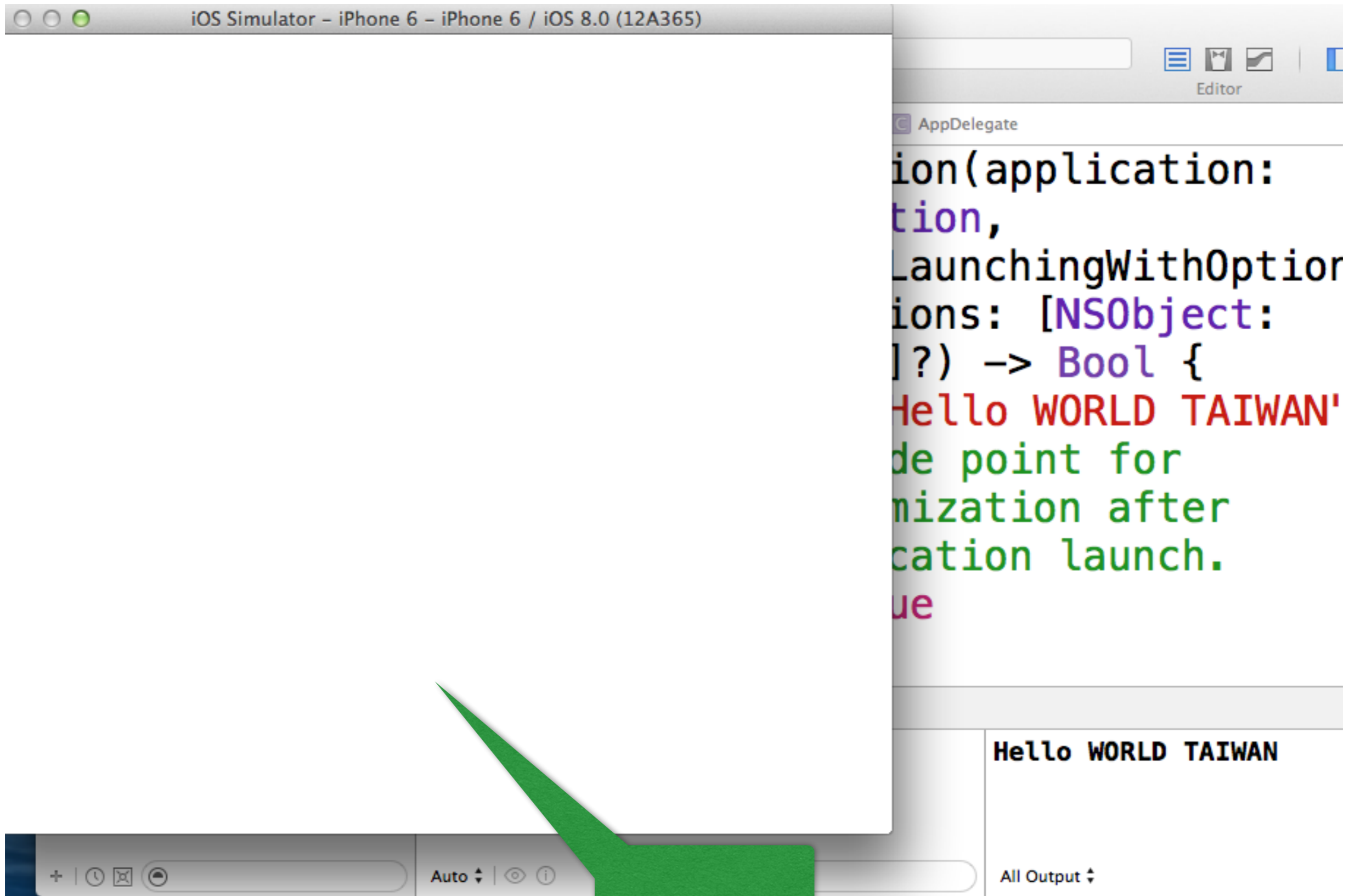
```
func applicationWillResignActive(
    didFinishLaunchingWithOptions
    launchOptions: [NSObject:
    AnyObject]?) -> Bool {
    println("Hello WORLD TAIWAN")
    // Override point for
    // customization after
    // application launch.
    return true
}
```

Below the code editor is the 'OUTPUT CONSOLE' window, which displays the text 'Hello WORLD TAIWAN'.

OUTPUT CONSOLE

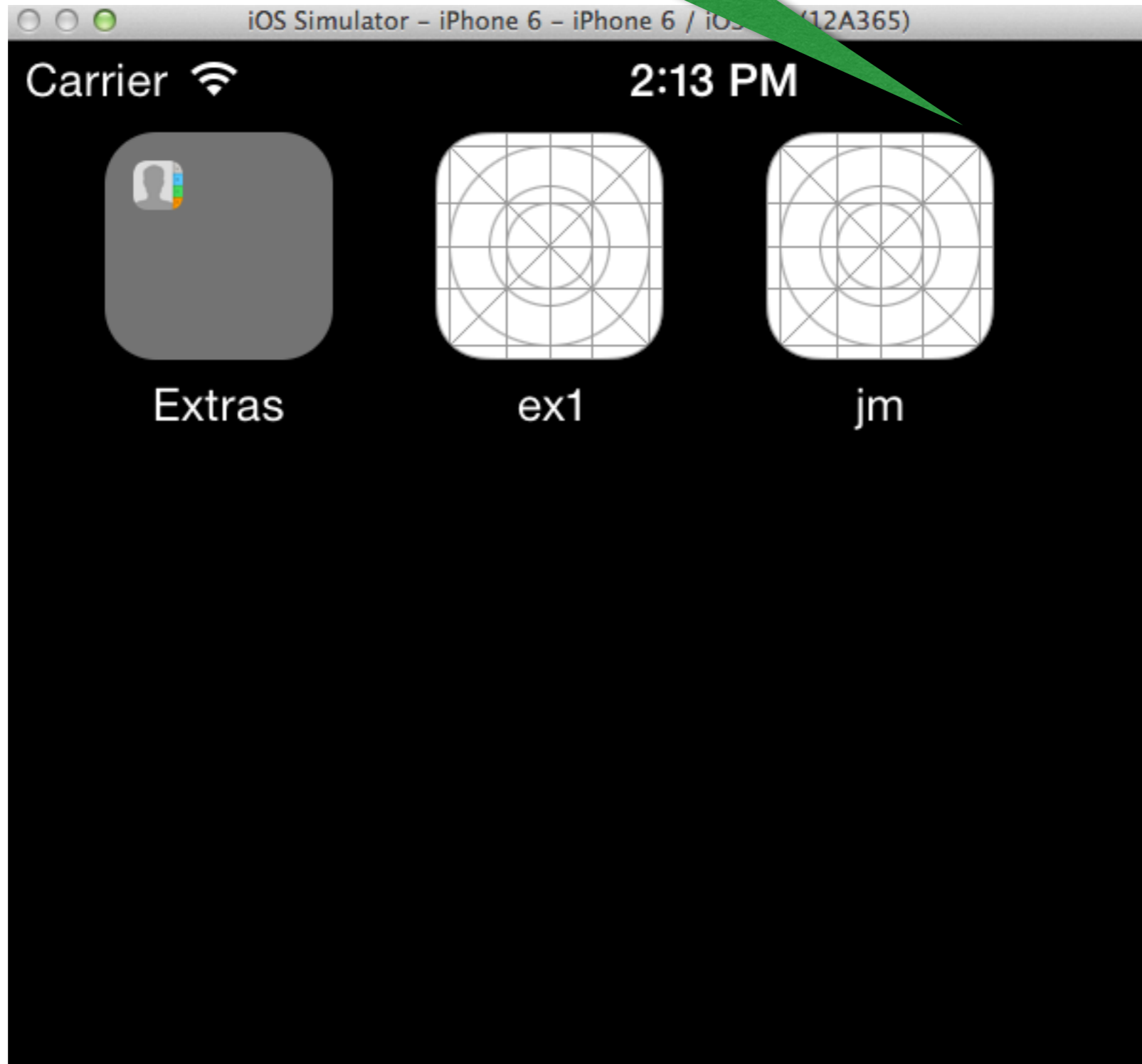
Hello WORLD TAIWAN

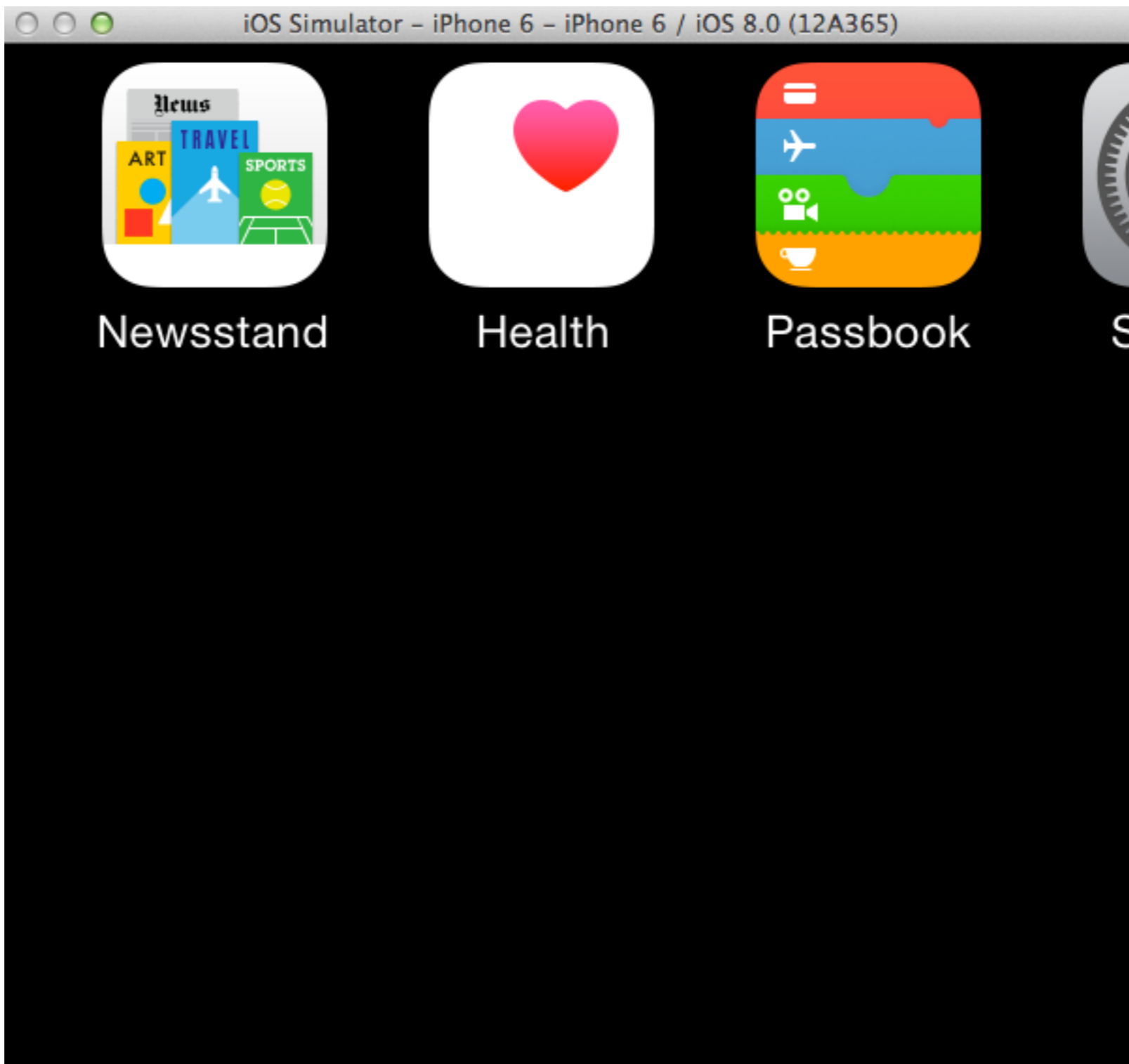
All Output



IPHONE 6

# IPHOME SIMULATORS





[http://www.minwt.com/ios/  
3676.html](http://www.minwt.com/ios/3676.html)

電火球

SAMPLE CODE

WWDC-2014



FREE SWIFT  
TUTORIALS



## Swift Tutorial Part 2: A Simple iOS App



Ray Wenderlich on June 14, 2014



If you're new here, you may want to subscribe to my [RSS feed](#) or follow me on [Twitter](#). Thanks for visiting!

**Update 12/18/14:** Updated for Xcode 6.1.1.

Welcome back to our Swift tutorial series!

In the [first Swift tutorial](#), you learned the basics of the Swift language, and created your very own tip calculator class.

In this second Swift tutorial, you will learn how to make a simple iOS app. Specifically, you will create a user interface for your tip calculator class that you developed last time.

I will be writing this tutorial in a manner so that it is useful for both complete beginners to iOS, and seasoned iOS developers transitioning to Swift.

For this Swift tutorial, you will need the latest public copy of Xcode (Xcode 6.1.1 at the time of writing this Swift tutorial). You do not need any prior Swift or Objective-C experience, but it would help if you had some prior programming experi-



*Create a simple iOS app in this Swift tutorial!*

<http://www.raywenderlich.com/74904/swift-tutorial-part-2-simple-ios-app>

```
//  
// File.swift  
// test  
//  
// Created by LAB326 on 2015/3/17.  
// Copyright (c) 2015年 LAB326. All rights  
// reserved.  
//  
  
import Foundation  
class TipCalculatorModel {  
  
    var total: Double  
    var taxPct: Double  
    var subtotal: Double {  
        get {  
            return total / (taxPct + 1)  
        }  
    }  
}
```



Carrier 

2:32 PM



# Tip Calculator

Total

100.00

Tip Percentage

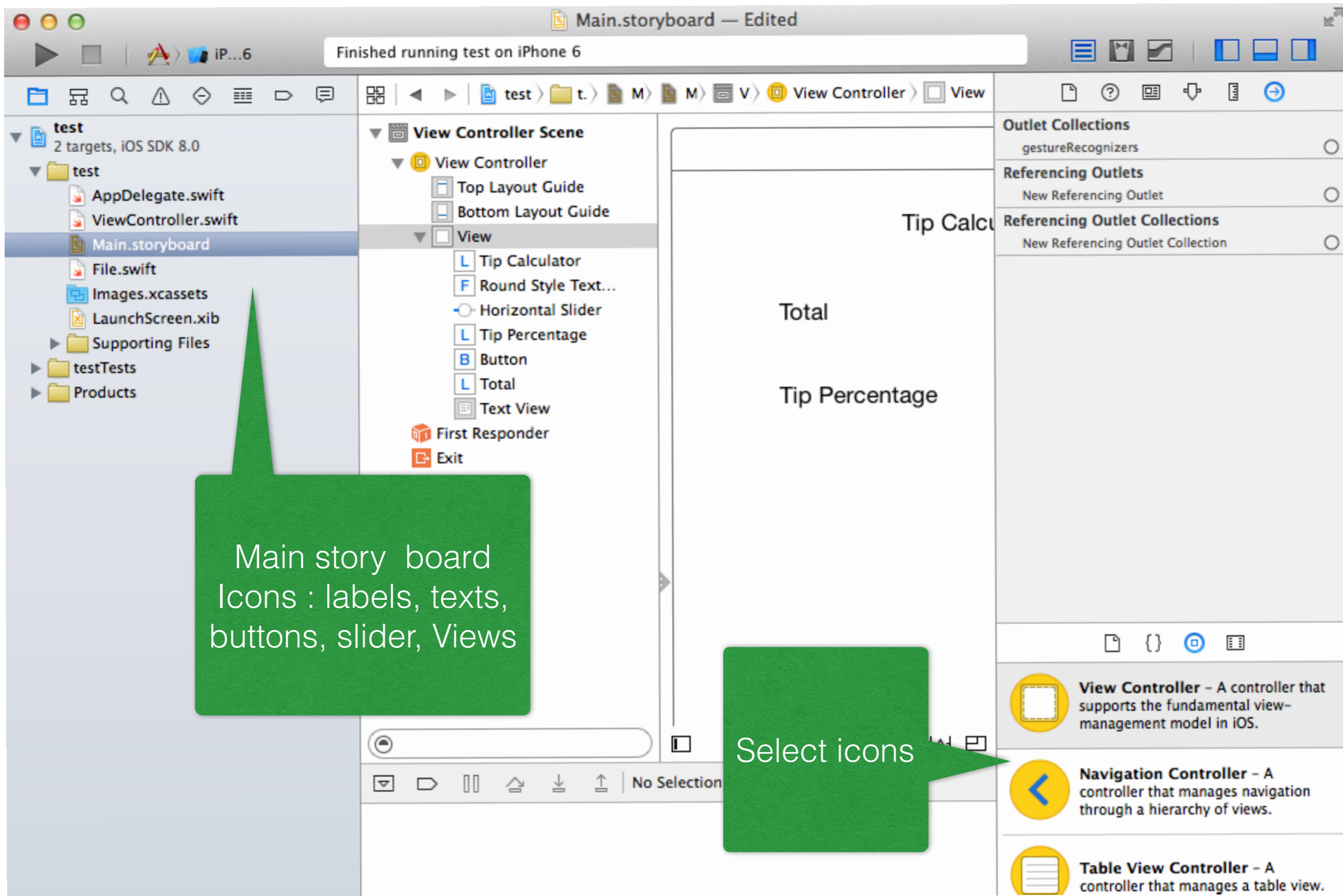


Button

20%: 19.3363375847019

18%: 17.4027038262317

15%: 14.5022531885264



Main story board  
Icons : labels, texts,  
buttons, slider, Views

Select icons

**Outlet Collections**  
gestureRecognizers ○

**Referencing Outlets**  
New Referencing Outlet ○

**Referencing Outlet Collections**  
New Referencing Outlet Collection ○

---

**View Controller** - A controller that supports the fundamental view-management model in iOS.

**Navigation Controller** - A controller that manages navigation through a hierarchy of views.

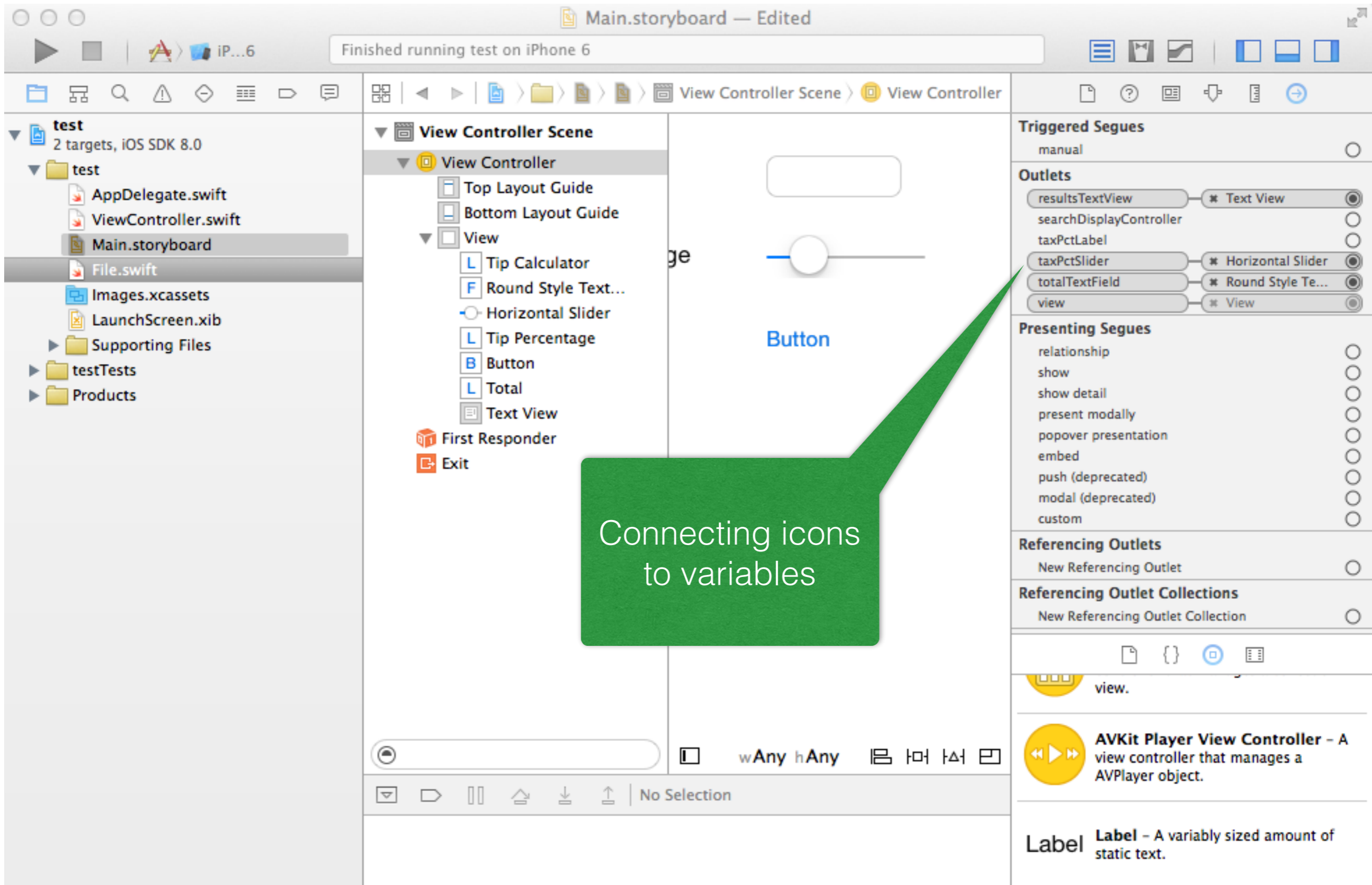
**Table View Controller** - A controller that manages a table view.

```
ViewController.swift
test > test > ViewController.swift > calculateTapped(:)
//
// Created by LAB326 on 2015/3/17.
// Copyright (c) 2015年 LAB326. All rights
// reserved.
//

import UIKit

class ViewController: UIViewController {
    @IBOutlet var totalTextField :
        UITextField!
    @IBOutlet var taxPctSlider : UISlider!
    @IBOutlet var taxPctLabel : UILabel!
    @IBOutlet var resultsTextView :
        UITextView!
    let tipCalc = TipCalculatorModel(total:
        33.25, taxPct: 0.06)
}
```

Add variables



Connecting icons to variables

**Triggered Segues**  
manual

**Outlets**

resultsTextView	* Text View	<input checked="" type="radio"/>
searchDisplayController		<input type="radio"/>
taxPctLabel		<input type="radio"/>
taxPctSlider	* Horizontal Slider	<input checked="" type="radio"/>
totalTextField	* Round Style Te...	<input checked="" type="radio"/>
view	* View	<input checked="" type="radio"/>

**Presenting Segues**

relationship	<input type="radio"/>
show	<input type="radio"/>
show detail	<input type="radio"/>
present modally	<input type="radio"/>
popover presentation	<input type="radio"/>
embed	<input type="radio"/>
push (deprecated)	<input type="radio"/>
modal (deprecated)	<input type="radio"/>
custom	<input type="radio"/>

**Referencing Outlets**  
New Referencing Outlet

**Referencing Outlet Collections**  
New Referencing Outlet Collection

view.

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

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

```
@IBAction func calculateTapped(sender :
    AnyObject) {
    tipCalc.total = Double
        ((totalTextField.text as
            NSString).doubleValue)
    // 2
    let possibleTips = tipCalc.
        returnPossibleTips()
    var results = ""
    // 3
    for (tipPct, tipValue) in
        possibleTips {
        // 4
        results += "\(tipPct)%: \
            (tipValue)\n"
    }
}
```