

Making App Pie

IOS10, SWIFT PROGRAMMING, TUTORIAL

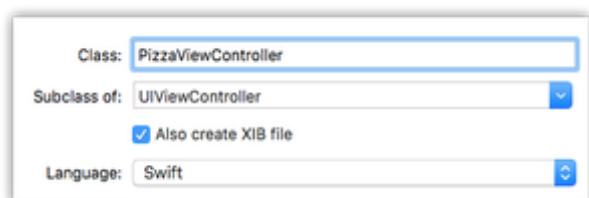
How to Make a Tab Bar Controller in Swift 3.0 Code

JULY 14, 2016 | STEVEN LIPTON | LEAVE A COMMENT

While Navigation controllers often have the limelight when it comes to Xcode's view controllers, *tab bar controllers* are better for independent tasks in the same app, or for different ways of working with the same model. In this lesson we'll take a look at how to make them in Swift programmatically. If you are interested in tab bar controllers on the storyboard, You might want to read this post (<https://makeapppie.com/2016/07/06/tab-bar-controllers-in-xcode-8-storyboards/>). For passing data between tabs read here (<https://makeapppie.com/2015/02/04/swift-swift-tutorials-passing-data-in-tab-bar-controllers/>).

Setting Up

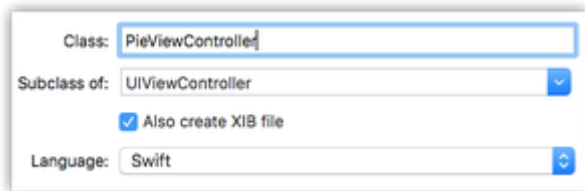
While there are very easy storyboard ways (<https://makeapppie.com/2016/07/06/tab-bar-controllers-in-xcode-8-storyboards/>) of making tab bar controllers, we can do much of this programmatically. Start with a single view template and create a Swift project called **TabProgDemo**. Once the project loads, on the keyboard hit **Command-N**. Create a **Cocoa Touch Class** named **PizzaViewController**, subclassing of **UIViewController**. Make the language **Swift**, and check the mark to make a **Xib file for an iPhone** like this:



([https://appie.files.wordpress.com/2016/07/2016-07-](https://appie.files.wordpress.com/2016/07/2016-07-12_10-38-481.png)

12_10-38-481.png)

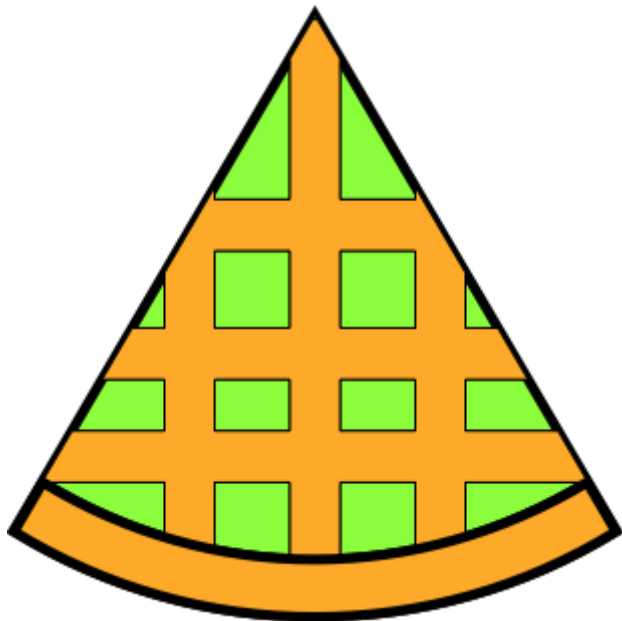
Press **Command-N** again and Make another **Cocoa Touch Class** like the first, subclassing **UIViewController**. Name it **PieViewController** again do not creating a XIB file.



([https://apppie.files.wordpress.com/2016/07/2016-07-](https://apppie.files.wordpress.com/2016/07/2016-07-13_06-44-57.png)

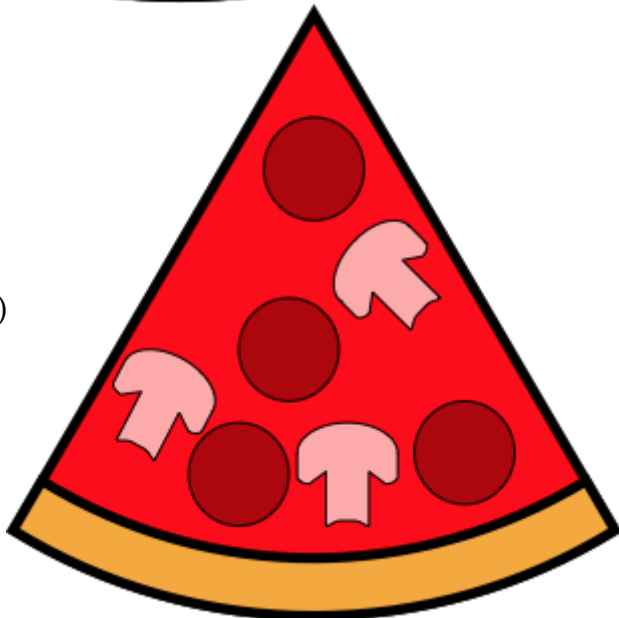
[13_06-44-57.png](https://apppie.files.wordpress.com/2016/07/2016-07-13_06-44-57.png))

I have created a file of image assets for this lesson. You can get them here as a zip file (<https://apppie.files.wordpress.com/2016/07/pizza-and-pie-icons.zip>) or you can right click and save the images below:



([https://apppie.files.wordpress.com/2016/07/pie-](https://apppie.files.wordpress.com/2016/07/pie-icon.png)

[icon.png](https://apppie.files.wordpress.com/2016/07/pie-icon.png))



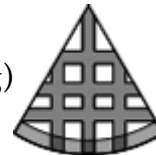
(<https://apppie.files.wordpress.com/2016/07/pizza-icon.png>)



(https://apppie.files.wordpress.com/2016/07/pizza_bar_icon2x.png)



(https://apppie.files.wordpress.com/2016/07/pizza_bar_icon.png)



(https://apppie.files.wordpress.com/2016/07/pie_bar_icon2x1.png)



(https://apppie.files.wordpress.com/2016/07/pie_bar_icon.png)

Save these images on your drive. Open up your **Assets.xcassets** folder in Xcode. Select all the files in finder and drag them into the assets folder.



(https://apppie.files.wordpress.com/2016/07/2016-07-13_06-15-30.png)

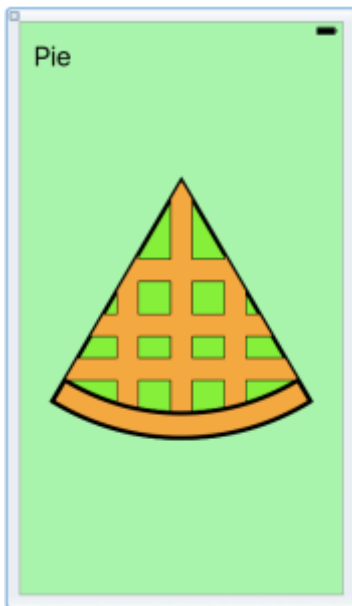
13_06-15-30.png)

Go to the the **PieViewController.xib** file. Make sure in the lower left of Interface builder it reads the default setting:

☐ View as: iPhone 6s (wC hR)

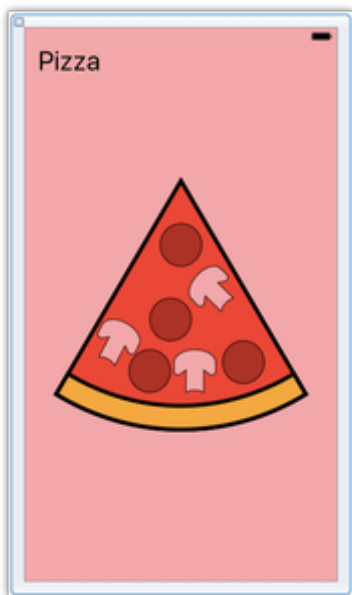
(<https://apppie.files.wordpress.com/2016/07/view-as-6s.png>)

If, not set it to iPhone 6s by clicking it and selecting the device. Set the background color to **Light Green(#AAFFAA)**. Drag out a label and title it **Pie** at **32 points**. Add the color **Pie Icon**. Your XIB should look like this:



(https://apppie.files.wordpress.com/2016/07/2016-07-13_06-46-52.png)

Select the PizzaViewController.xib file. Set up the Xib the same way as the pie controller. Use a **Light Red(#FFAAAA)** for the background:



(https://apppie.files.wordpress.com/2016/07/2016-07-13_06-30-30.png)

Tab Bar Programmatically

Click open the AppDelegate.swift file. While much of this can be important stuff, we'll clear it out so we can concentrate on tab bar controllers. Replace the class with the following:

```
1 class AppDelegate: UIResponder, UIApplicationDelegate {
2
3     var window: UIWindow?
4     func application(application: UIApplication,
5                     didFinishLaunchingWithOptions launchOptions: [NS
6                     return true
7     }
8 }
```

Let's put this code together part by part. First add this under the function declaration:

```
1 let tabBarController = UITabBarController()
```

Creates an instance of a `UITabBarController`. This will be the controller we'll configure in this function. We are doing this here so it is global to the application, one of the few times you want to do something globally.

Create the two view controllers from the XIB files. In full accordance with the adage "Life is uncertain, eat dessert first", we'll make the first page on launch be the pie page.

```
1 let tabViewController1 = PieViewController(  
2     nibName: "PieViewController",  
3     bundle: nil)  
4 let tabViewController2 = PizzaViewController(  
5     nibName: "PizzaViewController",  
6     bundle: nil)
```

Add these three lines:

```
1 let controllers = [tabViewController1, tabViewController2]  
2 tabBarController.viewControllers = controllers  
3 window?.rootViewController = tabBarController
```

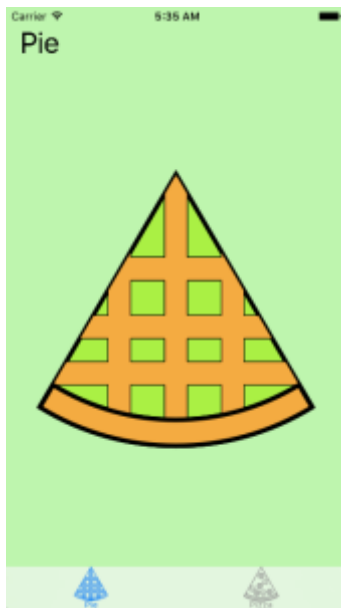
`UITabBarController` has a property called `viewControllers`, which is an array of the view controllers in the order displayed in the tab bar. Our first line of this segment makes an array of view controllers in the order we want them to present. Next we assign the array to the tab bar controller. The last line assigns the controller as the root view controller.

We have our controller set up, but no titles or icons. This code assigns the title and image.

```
1 tabViewController1.tabBarItem = UITabBarItem(  
2     title: "Pie",  
3     image: UIImage(named: "pie_bar_icon"),  
4     tag: 1)  
5 tabViewController2.tabBarItem = UITabBarItem(  
6     title: "Pizza",  
7     image: UIImage(named: "pizza_bar_icon"),  
8     tag: 2)
```

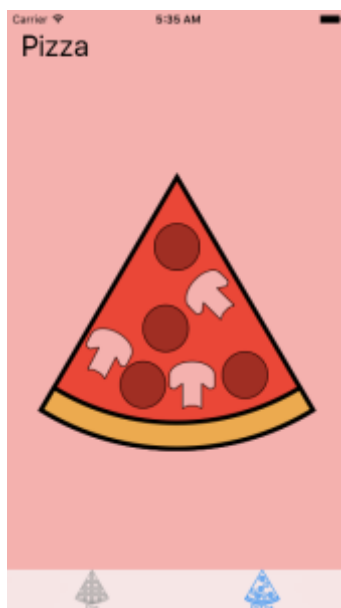
Each view controller has a property `tabBarItem` which contains the icon and text for the tab bar. We use the `UITabBarItem(title:String, image:UIImage, tag:Int)` initializer to create the tab bar items. In our example, we set the first one to **Pie** and use the **pie_bar_icon** image for the first. The second uses **Pizza** and the **pizza_bar_icon** image.

Set your simulator to the iPhone 6s. Build and run. You get the Pie first:



(https://apppie.files.wordpress.com/2016/07/2016-07-14_05-35-42.png)

You'll notice the tabs the bottom. Tap the pizza tab.



(https://apppie.files.wordpress.com/2016/07/2016-07-14_05-36-10.png)

You have working tabs.

◀ [APPDELEGATE](#) ▶ [APPLE SWIFT](#) ▶ [PROGRAMMATICALLY](#) ▶ [SWIFT](#) ▶ [TAB BAR](#)
[CONTROLLER PROGRAMMATICALLY](#) ▶ [TAB BAR ICONS](#) ▶ [TAB BAR SWIFT](#) ▶ [UI TAB](#)
[BAR CONTROLLER](#) ▶ [UI TAB](#)
[ITEM](#) ▶ [UITABBAR](#) ▶ [UITABBARCONTROLLER](#) ▶ [UITABBARITEM](#) ▶ [XCODE SWIFT](#)