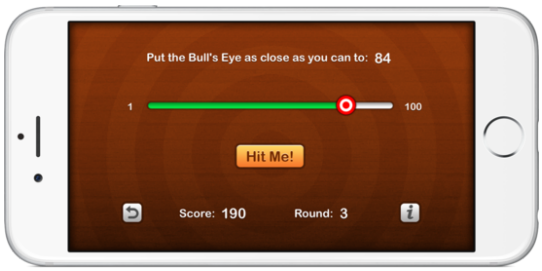
Ray Wenderlich para mim 25 de jul

José, welcome to the **raywenderlich.com iOS Apprentice Email Course**, the swiftest way (pardon the pun) to get started with iOS development!

In this course you'll learn how to make your first iOS app using Apple's Swift 4.0 programming language (and Xcode 9).

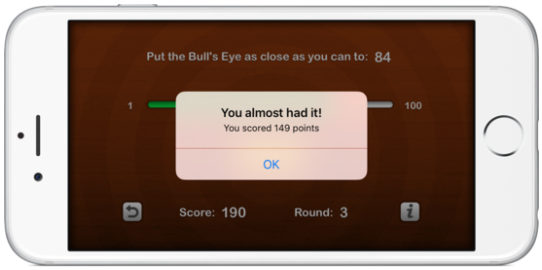
Everybody likes games, right? So, you'll start by building a simple but fun iPhone game named **Bull's Eye**. It will teach you the basics of iPhone programming.

This is what the **Bull's Eye** game will look like when you're finished:



The objective of the game is to put the bull's eye, which is on a slider that goes from 1 to 100, as close to a randomly chosen target value as you can. In the screenshot above, the aim is to put the bull's eye at 84. Because you can't see the current value of the slider, you'll have to "eyeball" it.

When you're confident of your estimate, you press the "Hit Me!" button and a popup, also known as an alert, will tell you what your score is:



The closer to the target value you are, the more points you score. After you dismiss the alert popup by pressing the OK button, a new round begins with a new random target. The game repeats until the player presses the "Start Over" button (the curly arrow in the bottom-left corner), which resets the score to 0.

This game probably won't make you an instant millionaire on the App Store, but even future millionaires have to start somewhere!

About This Course

This course is a collaboration between Matthijs Hollemans, Fahim Farook, Andy Obusek, and myself - Ray Wenderlich. It's based on section 1 of our best-selling book The iOS Apprentice.

Even if you've never programmed before or if you're new to iOS, you should be able to follow along with the step-by-step instructions and create your first app. Don't worry - this course has a ton of illustrations to prevent you from getting lost. Not everything might make sense right away, but hang in there and all will become clear in time.

Writing your own iOS apps is a lot of fun, but it's also hard work. If you have the imagination and perseverance, there is no limit to what you can make your apps do. It is our sincere belief that this course is a great way to get started with iOS and Swift development, but you do have to put in the time and effort. By creating this course, we've done our part. The rest is up to you…

Make a programming to-do list

And speaking of hard work, it's already time for your first challenge!

Now that you've seen what the game will look like and what the gameplay rules are, your challenge is to make a list of all the things that you think you'll need to do in order to build this game. It's OK if you draw a blank, but give it a shot anyway.

We'll give you an example:

The app needs to put the "Hit Me!" button on the screen and show an alert popup when the user presses it.

Try to think of other things the app needs to do – it doesn't matter if you don't actually know how to accomplish these tasks. The first step is to figure out **what** you need to do; **how** to do these things is not important yet.

Once you know what you want, you can also figure out how to do it, even if you have to ask someone or look it up. But the "what" comes first. (You'd be surprised at how many people start writing code without a clear idea of what they're actually trying to achieve. No wonder they get stuck!)

Whenever I start working on a new app, I first make a list of all the different pieces of functionality I think the app will need. This becomes my programming to-do list. Having a list that breaks up a design into several smaller steps is a great way to deal with the complexity of a project.

You may have a cool idea for an app but when you sit down to write the program the whole thing can seem overwhelming. There is so much to do… and where to begin? By cutting up the workload into small steps you make the project less daunting – you can always find a step that is simple and small enough to make a good starting point and take it from there.

It's no big deal if this challenge is giving you difficulty. You're new to all of this! As your understanding grows of how software and the development process works, it will become easier to identify the different parts that make up a design, and to split it into manageable pieces.

> **Challenge:** If you haven't already, stop reading this email and write your programming to-do list. If you'd like to compare what you wrote with ours, stop by the discussion thread on our forums. We'd love to see what you came up with! :]

Hello, World!

In this part of the course, we'll get you started with the very basics: creating a "Hello, World" app. And to do that, the first order of business is to download and install Xcode.

Xcode is the development tool for iOS apps. It has a text editor where you'll type in your source code and it has a visual editor for designing your app's user interface.

You can download Xcode for free from the Mac App Store. This requires at least an up-to-date version of macOS, so if you're still running an older version of macOS, you'll first have to upgrade to the latest version of macOS (also available for free from the Mac App Store). Get ready for a big download, as the full Xcode package is about 5 GB.

Important: You may already have a version of Xcode on your system that came pre-installed with your version of macOS. That version could be hopelessly outdated, so don't use it. Apple puts out new releases on a regular basis and you are encouraged to always develop with the latest Xcode and the latest available SDK on the latest version of macOS.

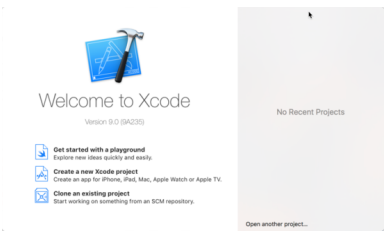
We wrote this revision of this course with **Xcode version 9** and the **iOS 11** SDK. By the time you're reading this, the version numbers might have gone up again. We will do our best to keep this up-to-date with new releases of the development tools and iOS versions, but don't panic if the screenshots don't correspond 100% to what you see on your screen. In most cases, the differences will be minor.

Once you have Xcode installed, it's time to start coding!

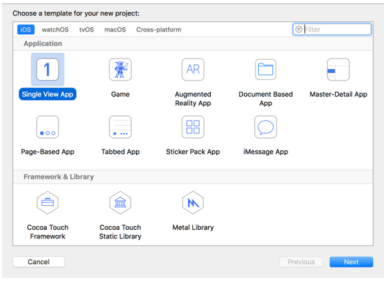
Create a new project

➤ Launch Xcode. If you have trouble locating the Xcode application, you can find it in the folder **/Applications/Xcode** or in your Launchpad. You may wish to place its icon in your dock for easy access.

Xcode shows the "Welcome to Xcode" window when it starts:



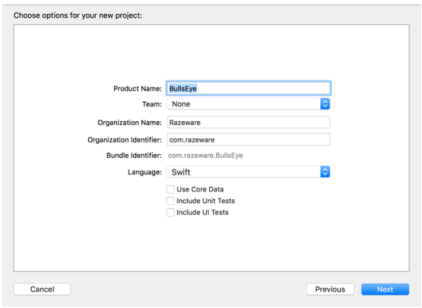
➤ Choose **Create a new Xcode project**. The main Xcode window appears with an assistant that lets you choose a template:



Xcode has bundled templates for a variety of application styles. Xcode will make a pre-configured project for you based on the template you choose. The new project will already include some of the source files you need. These templates are handy because they can save you a lot of typing. They are ready-made starting points.

➤ Select **Single View App** and press **Next**.

This opens a screen where you can enter options for the new app:

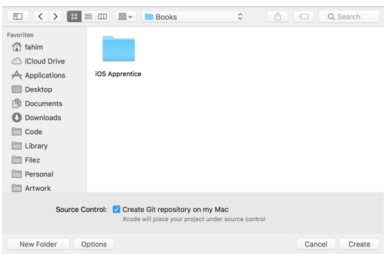


➤ Fill out these options as follows:

- Product Name: **BullsEye**. If you want to use proper English, you can name the project Bull's Eye instead of BullsEye, but it's best to avoid spaces and other special characters in project names.
- Team: If you already are a member of the Apple Developer Program, this will show your team name. For now, it's best to leave this setting alone; we'll get back to this later on.
- Organization Name: Fill in your own name here or the name of your company.
- Organization Identifier: Ours says "com.razeware". That is the identifier we use for the apps at our company, which is named Razeware. As is customary, it is our domain name written in reverse. You should use your own identifier here. Pick something that is unique to you, either the domain name of your website (but backwards) or simply your own name. You can always change this later.
- Language: **Swift**

Make sure the three options at the bottom – Use Core Data, Include Unit Tests, and Include UI Tests – are **not** selected. You won't be using those in this project.

➤ Press **Next**. Now Xcode will ask where to save your project:



➤ Choose a location for the project files, for example the Desktop or your Documents folder.

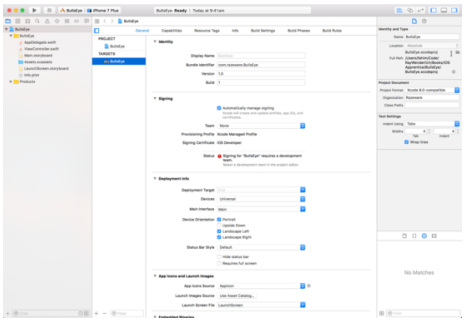
Xcode will automatically make a new folder for the project using the Product Name that you entered in the previous step (in your case BullsEye), so you don't need to make a new folder yourself.

At the bottom there is a checkbox that says, "Create Git repository on My Mac". You can ignore this for now. You'll learn about the Git version control system later on.

➤ Press **Create** to finish.

Xcode will now create a new project named BullsEye, based on the Single View Application template, in the folder you specified.

When it is done, the screen should look something like this:

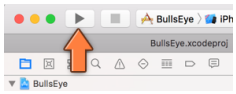


There may be small differences with what you're seeing on your own computer if you're using a version of Xcode newer than my version. Rest assured, any differences will only be superficial.

> **Note:** If you don't see a file named ViewController.swift in the list on the left but instead have ViewController.h and ViewController.m, then you picked the wrong language (Objective-C) when you created the project. Start over and be sure to choose Swift as the programming language.

Run your project

➤ Press the **Run** button in the top-left corner:



> **Note:** If this is the first time you're using Xcode, it may ask you to enable developer mode. Click **Enable** and enter your password to allow Xcode to make these changes.
> Also, make sure that you do not have your iPhone or iPad plugged in at this point to your computer, for example, for charging. If you do, it might switch to the actual device instead of the Simulator for running the app and since you are not yet set up for running on device, this could result in errors that might leave you scratching your head :]

Xcode will labor for a bit and then launch your brand new app in the iOS Simulator. The app may not look like much yet – and there is not anything you can do with it either – but this is an important first milestone in your journey!

Congratulations on making it through the first part of the course! But don't stop here, this is just the beginning.

In my next email, we'll start creating Bull's Eye and diving into your first few lines of Swift. See you soon! :]

- Ray

If you'd like to stop receiving the iOS Apprentice Email Course but still stay subscribed to raywenderlich.com Weekly, click here.
