iOS Setup Guide

The iOS build setup for Unity involves fewer steps than Android, but it more restrictive since it involves Apple™. To build for iOS devices for free, you will need to:

1. Get an Apple ID. If you've got an iOS device, you probably already have this
2. Use a computer with OSX 10.10 or greater and an iOS device (iOS 8.0 or greater*)
3. Install Xcode 7 or greater & register your Apple ID with it
4. Install the Unity editor extension that allows Unity to make Xcode projects. (It is not installed by default when you install Unity.)
5. Build & sign your app with Xcode

This will allow us to "sideload" a build of our Unity app onto a device connected to an OSX computer via USB. (Note: you may be able to use an older version of OSX and Xcode, but you will not be able to sideload apps for free - you will need a developer account.)

Unity's manual on iOS building is a bit better than their Android manual: see here for reference.

Developer Account

You can sideload apps onto an iOS device connected to your computer for free (Xcode 7 or greater), but if you want to put your app on the app store or distribute it more widely outside of the app store (e.g. email it to a friend), you will need a developer account. This costs $99/year.

Quick comparison to Android. You can build your Android apps and share them via a URL to the `.apk` file for free. If you want to publish your app to the Google Play store, you need to pay a one-time $25 fee. If you want to publish your app to the Amazon Fire store, it's free.

Xcode

Xcode is the Apple IDE for building apps for Mac devices. If you've got OSX >= 10.10, just head grab the free download from the Mac app store. After it has finished installing, add your Apple ID to Xcode using this guide.

Notes:

- You need Xcode 7 or higher to load your apps on your phone for free without a developer account. You'll need at least OSX 10.10 to be able to run Xcode 7.0 - 7.2. For Xcode >=7.3, you'll need OSX 10.11.
- You can download older versions of Xcode from the Apple developer site here. You should be able to sign in/register with your Apple ID to get free access to limited developer features.

Unity Xcode Build Package

Go to File -> Build Settings -> iOS -> Open download. Install the package and restart Unity.

Building and Signing a Unity project

When building & installing an app for iOS, there is a signing procedure for verifying that the app is allowed to be installed on the target device.

1. Open a Unity project
2. Set up the build

    - Go to File -> Build Settings. Select iOS.
    - Click the "Player Settings" button. Under "Other Settings", set the bundle identifier to something like "com.Mike.ParticlesFun".
3. Make sure your iOS device is connected. Hit "Build And Run" under the build window.
4. This will likely fail if it is the first time you are building an app. You need to sign the app with your apple ID.
    - Click on the failure notification to open the Xcode project. (Alternatively, go to where you choose to save the build and click the `.xcodeproj` file.)
    - Make sure that your device is listed in the top bar next to the play and stop buttons (see here).
    - Click on your project in the left sidebar to get to your app's general settings menu (see here).
    - In the general settings, under the "Signing" section, make sure that "Automatically manage signing" is checked and the "Team" dropdown has your name/ID selected.
    - Hit the play button to build & run your app on your iOS device.
    - If you get an error that says "Could not launch...", follow the steps on screen.
5. Celebrate! (Or see troubleshooting below)

Troubleshooting

- If you get an error that says "Could not launch...", follow the instructions in the pop up. These will guide you to setting up your iOS device to trust apps that you build & install from Xcode.
- If you get an error that says "The maximum number of apps for free development profiles has been reached", follow this SO answer to remove some of the prior apps you have built and installed on the device.
- There are a lot of things that were installed, so the first debugging step is to reboot & reconnect. Disconnect the device, close Unity. Reconnect the device and restart Unity.
- If that doesn't work, you may need to reboot your computer.
- Google is your friend! If these don't solve the problem, try googling any errors or messages you are getting.

Unity 5 Remote

We've just gone through the process for building an app that can run on device. That's generally the process you'll want to go through for deploying an app to a device (or the Google Play store), but that's not the best process for development.

Unity Remote is an Android/iOS app that allows us to run our app on our device while still using our computer to see the console, tweak things in the inspector, etc. This is incredibly helpful for development.

1. Download Unity Remote from the iOS store
2. Open a Unity project.
3. Go to Edit -> Project Settings -> Editor. Set "Device" to "Any iOS Device."
4. Open up the Unity Remote 5 app on your device. Then hit play in Unity on your computer.

Again, if it doesn't work - disconnect, restart Unity and reconnect your device.

Distributing the App

You don't have as many options as Android here...

1. Sideload the app onto a specific device over USB. That's the process we went through in this PDF.
2. Get a $99/year developer account in order to publish apps to the app store (or to be able to sign the app and distribute it outside the app store).