



# .NET Developer's Guide to Augmented Reality in iOS

Building Immersive Apps Using Xamarin,  
ARKit, and C#

---

Lee Englestone

Apress®

# **.NET Developer's Guide to Augmented Reality in iOS**

**Building Immersive Apps Using  
Xamarin, ARKit, and C#**

**Lee Englestone**

**Apress®**

# ***.NET Developer's Guide to Augmented Reality in iOS: Building Immersive Apps Using Xamarin, ARKit, and C#***

Lee Englestone  
Offerton, UK

ISBN-13 (pbk): 978-1-4842-6769-1

<https://doi.org/10.1007/978-1-4842-6770-7>

ISBN-13 (electronic): 978-1-4842-6770-7

Copyright © 2021 by Lee Englestone

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Joan Murray

Development Editor: Laura Berendson

Coordinating Editor: Jill Balzano

Cover image designed by Freepik ([www.freepik.com](http://www.freepik.com))

Distributed to the book trade worldwide by Springer Science+Business Media LLC, 1 New York Plaza, Suite 4600, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [booktranslations@springernature.com](mailto:booktranslations@springernature.com); for reprint, paperback, or audio rights, please e-mail [bookpermissions@springernature.com](mailto:bookpermissions@springernature.com).

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at [www.apress.com/9781484267691](http://www.apress.com/9781484267691). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*This book is dedicated to my family.*

*To my amazing wife who is the kindest person you could ever hope to meet and my children, Holly and Aaron, who bring me so much joy.*

# Table of Contents

- About the Author ..... xi
- About the Technical Reviewer ..... xiii
- Preface ..... xv
- Introduction ..... xvii
- Chapter 1: Setting Up Your Environment ..... 1
  - Apple ID..... 1
  - Suitable iOS Device..... 2
  - Install Xcode..... 2
  - Install Visual Studio for Mac..... 3
  - Create a New Project in Xcode..... 4
    - Step 1. Choose a project template ..... 5
    - Step 2. Provide project details..... 6
    - Step 3. Provide a project location..... 7
    - Step 4. View the new project..... 8
    - Step 5. Select a Team or sign in with an Apple ID ..... 9
    - Step 6. Change the deployment target..... 10
    - Step 7. Trust the developer..... 11
    - Step 8. Done ..... 13
  - Create a New Project in Visual Studio for Mac..... 13
    - Step 1. Create a new project and select a project type..... 13
    - Step 2. Provide app details..... 14
    - Step 3. Provide project details..... 15
    - Step 4. Choose a deployment device and run ..... 16
    - Set Camera Permissions ..... 17
  - Summary..... 18

TABLE OF CONTENTS

**Chapter 2: Basic Concepts..... 19**

    Scene View..... 19

    Session ..... 20

    SceneKit..... 21

    Positioning ..... 21

    World Origin ..... 23

    World Alignment..... 24

    Gravity..... 25

    GravityAndHeading..... 25

    Camera..... 25

    Size ..... 26

    Configuration..... 28

    Summary..... 29

**Chapter 3: Nodes, Geometries, Materials, and Anchors ..... 31**

    Nodes..... 31

    Opacity ..... 32

    Geometries..... 33

    Built-in Geometry Shapes ..... 33

    Materials ..... 35

        Solid Color Material ..... 35

    Image Material ..... 36

    Material Fill Mode ..... 37

    Anchors..... 38

    Things to Try..... 38

    Summary..... 39

**Chapter 4: Built-in AR Guides ..... 41**

    Show Feature Points ..... 41

    Show World Origin and Coordinate Axis..... 42

    Show Statistics ..... 44

Coaching Overlay .....	45
Summary.....	48
<b>Chapter 5: Animations .....</b>	<b>49</b>
Animating Opacity.....	49
Animating Scale .....	50
Animating Position .....	51
Animating Orientation .....	51
Repeat Behavior.....	52
Animation Easing .....	53
Combining Animations .....	54
Waiting .....	54
Summary.....	55
<b>Chapter 6: Constraints .....</b>	<b>57</b>
BillboardConstraint .....	57
LookAtConstraint.....	57
Other Constraints .....	59
Things to Try.....	59
Summary.....	59
<b>Chapter 7: Lighting .....</b>	<b>61</b>
Automatically Add Default Lighting .....	61
Automatically Update Default Lighting.....	61
Light Types .....	62
Adding Shadows .....	63
Things to Try.....	67
Summary.....	68
<b>Chapter 8: Video and Sound .....</b>	<b>69</b>
Playing Sound .....	69
Playing Video.....	70
Things to Try.....	72
Summary.....	72

TABLE OF CONTENTS

**Chapter 9: Plane Detection ..... 73**

    Detecting Planes ..... 73

    Remembering Planes..... 73

    ARSCNViewDelegate (Scene View Delegate) ..... 74

    Plane Detection Example ..... 76

        Turning Off Plane Detection ..... 81

        Possible Applications..... 82

        Things to Try ..... 82

    Summary..... 83

**Chapter 10: Image Detection ..... 85**

    Adding Images as App Resources..... 85

        Detecting the Images ..... 88

        Dynamically Adding Images to Detect..... 91

        Things to Try ..... 92

    Summary..... 93

**Chapter 11: Face Tracking and Expression Detection ..... 95**

    Tracking Faces ..... 95

    Recognize Facial Expressions..... 99

    Things to Try..... 105

    Summary..... 105

**Chapter 12: Touch Gestures and Interaction ..... 107**

    Gesture Recognizers ..... 107

    Hooking Up Gesture Recognizers..... 108

    Tapping ..... 109

    Pinching ..... 110

    Rotating..... 111

    Panning ..... 112

    Swiping ..... 113



Long Press .....	113
Things to Try.....	114
Summary.....	115
<b>Chapter 13: 3D Models .....</b>	<b>117</b>
Importing 3D Models.....	117
Creating Your Own 3D Models in Blender .....	119
Add Shadows, Animations, and Make Interactive .....	121
Things to Try.....	121
Summary.....	121
<b>Chapter 14: Physics.....</b>	<b>123</b>
Giving an Item a Rigid Structure .....	123
Applying Gravity to an Object.....	124
Combining Gravity and Solid Objects .....	125
Applying Force .....	128
Things to Try.....	132
Summary.....	133
<b>Chapter 15: Object Detection .....</b>	<b>135</b>
Scanning and Saving Object Spatial Data.....	135
Recognizing Scanned Objects.....	137
Things to Try .....	139
Summary.....	140
<b>Chapter 16: Body Tracking .....</b>	<b>141</b>
Detecting a Body in a Scene .....	141
Capturing Body Motion.....	148
Potential Applications .....	149
Things to Try.....	150
Summary.....	150

TABLE OF CONTENTS

**Chapter 17: Publishing to the App Store ..... 151**

    App Store Submission To-Do List..... 151

    Set Up Icons for the App ..... 152

    Set Up Launch Screen Image..... 153

    Set Up App ID and Entitlements ..... 154

    Create and Install an App Store Provisioning Profile..... 158

    Update Build Release Configuration..... 162

    Set Up the App in App Store Connect..... 164

    Build the App and Submit It to Apple ..... 169

    Summary..... 178

**Index..... 179**

# About the Author



**Lee Englestone** is an innovative, hands-on software development manager and technical lead, based in Stockport, England. He has been a .NET developer for many years, writing code for Windows, web, mobile, cloud, and Augmented Reality (AR) applications in his spare time. He believes that there are many new exciting opportunities for developers in the area of Augmented Reality and is excited to share them with his fellow .NET developers. In recognition of his community contributions, he has been awarded a Microsoft MVP in Developer Technologies. He can be

contacted through LinkedIn at <https://www.linkedin.com/in/leeenglestone/> or on Twitter at <https://twitter.com/LeeEnglestone>.

# About the Technical Reviewer



**Nishith Pathak** is India's first and only Artificial Intelligence (AI) Most Valuable Professional (MVP), a Microsoft Regional Director (RD), and a lead architect, speaker, AI thinker, innovator, and strategist. Nishith's expertise lies in helping Fortune 100 companies design and architect next-generation solutions that incorporate AI, ML, cognitive services, Blockchain, and many more. He also laid his expertise in defining and strategizing technology road maps for customers and companies using emerging technologies. He sits on several technical advisory boards across the globe. He has also authored more than half a dozen international

books for Springer Publication, United States. The last three books were on Artificial Intelligence (AI). Earlier, Nishith has also played the role of a PAN account enterprise architect where he was responsible for the overall architecture to design in multiple projects. He is an internationally acclaimed speaker on technologies like AI, IOT, and Blockchain and regularly speaks at various technical conferences. He advises and mentors a lot of startups as a community initiative.

For his expertise on Artificial Intelligence, Microsoft have awarded him the first Most Valuable Professional (MVP) from India in the Artificial Intelligence category. He is the only Artificial Intelligence MVP in India to date. Globally, he is among 19 MVPs on AI, recognized by Microsoft for their sheer expertise on AI. He has also received the "Microsoft Regional Director" award which is awarded to 150 of the world's top technology visionaries chosen specifically for their proven cross-platform expertise.

Nishith is a gold member and sits on the advisory board of various national and international computer science societies and organizations. Nishith is currently working as Global Chief Technologist for Emerging Technologies and Advanced Analytics for DXC Technology where he is focused on using emerging technologies helping companies architect solutions based on these technologies, laying out technology road maps, and curating the startup ecosystem. He can be contacted at [nispathak@gmail.com](mailto:nispathak@gmail.com) or through LinkedIn at [www.linkedin.com/in/nishithpathak/](http://www.linkedin.com/in/nishithpathak/).

# Preface

I have always loved futuristic science fiction films and TV series. As a child, I remember fondly watching *Star Trek* and *Star Wars* with my dad most evenings after school. You name it, if it was set in the future, I probably watched it.

But these days, it is not the lasers or spaceships that I marvel at. Being a software developer, it is the glowing, floating futuristic-looking computer interfaces that spark my imagination. It seems in some distant future we are destined to shed our monitors, mice, and keyboards and replace them with floating transparent holograms using only our hands to interact with them.

Having followed Augmented Reality very closely for some time, I believe that future is already here, and to prove it, I will show you how to build the interfaces of tomorrow, today. As .NET developers, we have the enviable ability to develop apps for a wide range of uses including Windows, web, console, cloud, serverless, ML, AI, mobile, and now Augmented Reality.

I'm very excited to share how you and fellow .NET developers can use .NET and C# to create amazing apps that use Augmented Reality to open up a new world of possibilities, to be at the forefront of unique AR experiences that will take the world by storm.

# Introduction

Welcome to the start of your journey into the exciting world of Augmented Reality development.

Being a .NET developer and iPhone user for many years, I was shocked when I accidentally stumbled onto the Augmented Reality functionality that .NET developers can target and leverage to create fantastic user experiences; I was amazed that more .NET developers weren't using the techniques outlined in this book to create their own AR apps for their own iOS devices, especially at a time when Augmented Reality is becoming increasingly popular in our society and adopted by a growing number of industries and markets.

In fact, I was so impressed by the possibilities that Augmented Reality offers .NET developers that I started experimenting with it as much as possible, as well as discussing it at .NET user groups around the country to show fellow .NET developers the kind of experiences they could achieve.

But I didn't stop there, I wondered how else I could reach even more .NET developers to share with them this new way of creating immersive user experiences. I even wondered if I could write a book introducing the concept of .NET: this book. And if you are reading this, you are interested in learning how to create the next generation of Augmented Reality user experiences and have come to the right place.

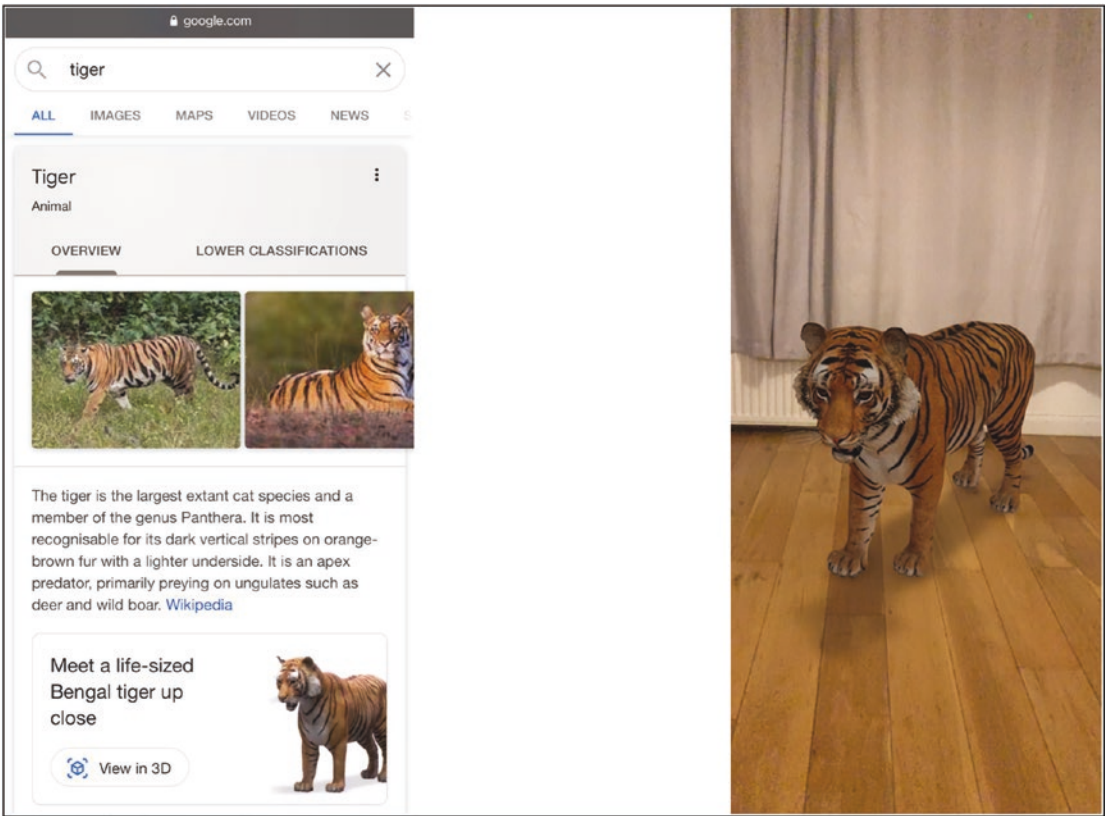
Before we get stuck into coding and creating, it is worth giving you an overview of Augmented Reality and how to use this book.

## What Is Augmented Reality?

Augmented Reality (or AR for short) is the process of a computer program or mobile app making something appear in your immediate (real-world) environment that isn't actually there. That object's position is maintained in the real world and as you move around and change your perspective.

## INTRODUCTION

All of the major technology companies are finding uses for Augmented Reality including Facebook, Apple, Google, and Microsoft. At the time of writing, if you Google “tiger” on a mobile device, you are presented with a screen where if you press “View in 3D,” you can view the animal in Augmented Reality as shown in the image on the right in Figure 1.



**Figure 1.** *Augmented reality is becoming increasingly popular*

Augmented Reality isn’t to be confused with Virtual Reality (or VR for short), where you wear a headset that completely occludes your vision and everything you see is a product of the running computer program or mobile app.

One of the most popular VR devices on the market is Facebook’s Oculus range of devices with perhaps their most popular VR game being *Beat Saber* where one must use slice cubes that are approaching the player in VR to the rhythm of the music.

For completion, you may also have heard the terms XR (Extended Reality) and MR (Mixed Reality) previously. These terms are generally referring to *both* Augmented and Virtual Reality. We won’t use or discuss the terms XR and MR further in this book; you may just be interested to know that those terms encompass both AR and VR.

## Who Is This Book For?

This book is for .NET developers that want to use C# and Xamarin to create Augmented Reality apps for their iOS devices. While experience with Xamarin cross-platform apps and Visual Studio for Mac is helpful, it is not necessary, as we will cover the environment setup and basic concepts early on.

You may already be familiar with Augmented Reality, you may just be curious to learn more about what is possible with it, or you may already have an idea for an Augmented Reality app that you wish to create.

The only thing you need is your imagination.

## How to Read This Book

This book has been written to introduce you to the fundamental building blocks and concepts used to create Augmented Reality experiences, then show how these can be combined together, then move on to explain more advanced concepts. At each stage, code samples are provided to allow you to experiment for yourself the concept being described.

While I encourage you to understand each topic before moving on to the next, rest assured most code samples are stand-alone and can be run without previous code samples. So if you want to try out image detection before fully understanding how to work with simple geometries, feel free to do so.

## Code Samples

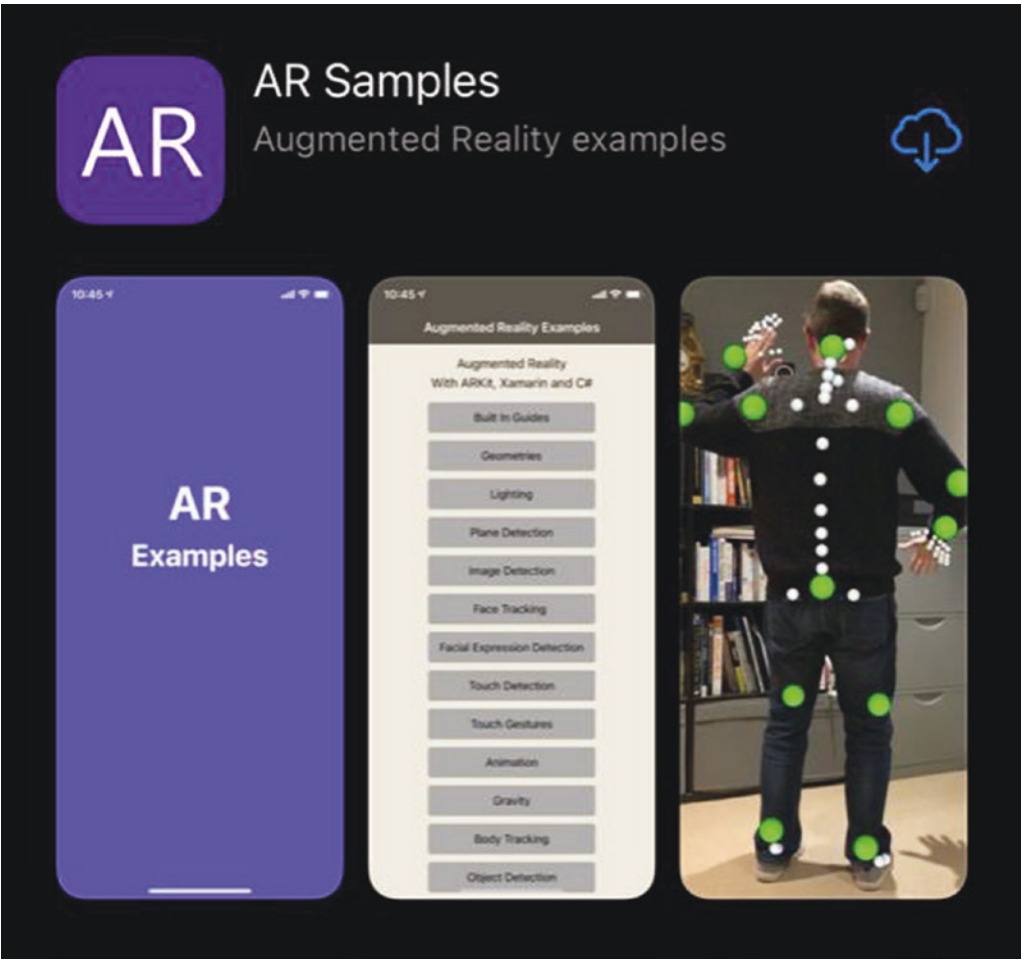
Code samples in this book have been written with clarity in mind. I'm sure in most cases the code could be optimized or better structured. They have been written to aid understanding and may not be the same code I would necessarily recommend be used for production.

Code samples can also be found in the source code for the companion app on GitHub.



# Companion App

All of the example functionality described in this book including face detection, 3D models, and so on can be experienced using the free companion app available in the iOS App Store as shown in Figure 2. The code for the companion app itself is in GitHub so that you can clone it to your computer and play around with the code yourself.



**Figure 2.** Example ARKit functionality is shown in the companion app

# The Rise of Augmented Reality

The rise of Augmented Reality has been a slow but steady one. That's not necessarily a bad thing given how Virtual Reality has suffered from being overhyped for over a decade and only recently delivering what it originally promised.

The slow adoption of Augmented Reality on the other hand has been relative to the increasing sophistication of our mobile devices, especially their processors and cameras. It is not uncommon for some mobile phones to boast three or four different cameras, some of which can determine the depth of field of what they are being directed at. Those cameras along with increasingly powerful processors are continually helping make Augmented Reality environments more accurate and allowing increasingly advanced experiences.

It isn't surprising then that a number of Augmented Reality frameworks allowing developers to leverage these devices have sprung up, the most common being ARCore for Android and ARKit for iOS.

You need only look at the popularity of one of the very first Augmented Reality games, *Pokémon Go*, that launched in 2016 which made over 30 billion (yes, billion) dollars for evidence that the general public are not only willing to embrace Augmented Reality but they desire to and are willing to pay to do so.

## ARKit

In 2017, Apple introduced its Augmented Reality framework ARKit to the world and has been improving it every year since. ARKit makes use of your device's camera, gyroscope, and accelerometer to determine the direction your device is pointing and the differences between camera frames so it can maintain the Augmented Reality experience.

### ARKit 1.0 (September 2017)

ARKit was first introduced in iOS 11.0 with the ability to track the world and environment around the user. This initial release also allowed for face tracking and horizontal plane detection.

### ARKit 1.5 (March 2018)

In their first ARKit update in iOS 11.3, Apple added the much requested ability to detect vertical planes such as walls as well as simple image recognition.

## **ARKit 2.0 (September 2018)**

Later that year in iOS 12 ARKit was extended to allow 3D object recognition as well as scene persistence and sharing.

## **ARKit 3.0 (September 2019)**

In iOS 13 Apple added support for people occlusion, multiple face tracking, as well as human body tracking.

## **ARKit 3.5 (March 2020)**

In iOS 13.4 Apple added LiDAR scanning to iPad Pro devices improving plane detection as well as introducing Scene Geometry allowing users to create topological maps of their surroundings and even use semantic classification which can identify everyday objects such as chairs and tables. Though this is just for iPad Pro at the time of writing, it may be supported by future versions of iPhone and is rumored to be in iPhone 12.

# **Augmented Reality for .NET Developers Using Xamarin**

Until fairly recently, if you wanted to create a native Augmented Reality app for iOS using ARKit, you would have to write it in Objective-C or Swift using Xcode.

Note: This book covers Augmented Reality development using Xamarin and ARKit which is Apple's Augmented Reality framework. This is not to be confused with ARCore which is Android's Augmented Reality framework.

A company called Xamarin (founded in 2011), seeing how people wanted to develop mobile apps for different platforms, created a framework called Mono which allowed C# developers to write C# code and produce apps able to run on macOS, Linux, Android, and iOS.

So popular was Xamarin in fact that in February 2016 Microsoft came along and acquired them and ported ARKit (among other frameworks) to .NET. Soon after that Microsoft made it possible for .NET developers to use Visual Studio for Mac, Xamarin, and ARKit to create Augmented Reality applications and deploy them to iOS devices.

Interestingly while this is a fantastic way to allow .NET developers to create Augmented Reality iOS apps using C#, there aren't many people doing it just yet. On the upside, you can be one of the first .NET developers to create and share some amazing AR experiences; on the downside, most of the ARKit code samples you will find online will be in Objective-C or Swift, and it can take a slight bit of head-scratching to work out how to do the same thing in .NET.

This is where this book comes in (I have spent many an hour translating Swift ARKit examples to C# so you don't have to). You're welcome.

## The Future of Augmented Reality

We are on the verge of an Augmented Reality revolution. While we are currently addicted to our little black screens to get our daily dose of news, social interaction, and entertainment, they will soon be replaced by Augmented Reality experiences.

Admittedly these Augmented Reality experiences are currently predominantly on our mobile devices; however, there are an ever-increasing number of manufacturers producing Augmented Reality headsets that will soon flood the market. It is therefore only logical that there will be a large demand of Augmented Reality software development and experiences needed to operate on these headsets.

Once businesses, marketers, and entrepreneurs start to realize the potential and leverage the power of Augmented Reality, we will see a gold rush of Augmented Reality experiences. Believe me, you want to be on the ground floor, build your AR skills, and ride the rocket that will be Augmented Reality development.

## Summary

We discussed Augmented Reality and how you can use your existing knowledge of C# and .NET along with the ARKit framework to create your own Augmented Reality experiences. We discussed just how exciting the future of Augmented Reality is going to be, especially for us .NET developers who get to create experiences for them.

So now that you can't wait to start producing Augmented Reality experiences, you'll want to check out the next chapter, "Setting Up Your Environment," which will tell you exactly what you need to do to get started as quickly as possible.