Beginning Swift Games Development for iOS

Develop 2D and 3D games Using Apple's SceneKit and SpriteKit

Second Edition

James Goodwill
Wesley Matlock

APRESS®

Beginning Swift Games Development for iOS

Develop 2D and 3D games Using Apple's
SceneKit and SpriteKit

Second Edition

James Goodwill
Wesley Matlock

Apress®

Beginning Swift Games Development for iOS: Develop 2D and 3D games Using Apple's SceneKit and SpriteKit

James Goodwill
Highlands Ranch, Colorado, USA

Wesley Matlock
Kansas City, Missouri, USA

To Christy Goodwill,

my awesome wife, you have supported me all of these years.
I could not have done a fraction of what I have without you. I love you.

Wesley Matlock - for Amy

Contents at a Glance

Contents

About the Authors

James Goodwill is a ten-time published author of books about leading technologies such as iOS, Swift, Objective C, Grails, Groovy, Java Servlets, JavaServer Pages (JSP), Tomcat, and Struts. He is a senior enterprise iOS and Java consultant in the Denver metro area and frequent speaker and article writer. You can find additional resources about Sprite Kit, Swift, and James himself at his blog at www.jgoodwill.org. You can also follow James on Twitter at https://twitter.com/jamesgoodwill.

Wesley Matlock is a professional independent iOS consultant in the Kansas City metro area. He has more than 20 years of development experience in several different platforms. He first started doing mobile development on the Compaq iPaq in the early 2000s. Today he enjoys developing on the iOS platform and bringing new ideas to life.

About the Technical Reviewer

Bruce Wade is a software engineer from British Columbia, Canada. He started software development when he was 16 years old by coding his first website. He went on to study Computer Information Systems ad DeVry Institute of Technology in Calgary, then to further enhance his skills he studied Visual & Game Programming at The Art Institute Vancouver. Over the years he has worked for large corporations as well as several start-ups. His software experience has led him to utilize many different technologies including C/C++, Python, Objective-C, Swift, Postgres, and JavaScript. In 2012 he started the company Warply Designed to focus on mobile 2D/3D and OS X development. Aside from hacking out new ideas, he enjoys spending time hiking with his Boxer Rasco, working out, and exploring new adventures.

Acknowledgments

This book could not have been written without the incredible folks at Apress. The idea of a Swift iOS gaming book began with a conversation with Steve Anglin and came to life with a great discussion about gaming and Apple with Michelle Lowman. Mark Powers and James Markham kept the book on the rails and brought it safely into the station. Bruce Wade made sure all of the technical statements made sense and the code compiled and ran successfully. I thank you all.

I want to send out a special thanks to Wes Matlock for taking over the Scene Kit section of the book when my father passed. I just did not have the time or energy to complete the second section of the book, and Wes stepped in without hesitation.

I also want to thank Deborah Saez for the wonderful artwork in the book. I highly recommend her. She is both very talented and a very hard worker. You can find her at www.deborahsaez.com/. Look her up.

Finally and most importantly, I want to thank the three girls in my life: Christy (my wonderful wife) and our daughters, Abby (who supplied a ton of inspiration) and Emma (who did a great technical review of the book). You three are the most important people in my life.

—James Goodwill

Introduction

Which Version of Swift Is Covered in This book?

This book covers version 3 of Swift and iOS version 10. As new versions are released, we will update the source for this book at both the Apress.com web site and James Goodwill's blog at www.jgoodwill.org.

What This Book Is

Game apps are one of the most popular categories in the Apple iTunes App Store. Well, the introduction of the new Swift programming language will make game development even more appealing and easier to existing and future iOS app developers. In response, James Goodwill, Wesley Matlock, and Apress introduce you to this book, *Beginning Swift Games Development for iOS*.

In this book, you'll learn the fundamental elements of the new Swift language as applied to game development for iOS in 2D and 3D worlds using both Sprite Kit and Scene Kit, respectively.

What You Need to Know

This book assumes you have a basic understanding of how to create applications for the iPhone using Xcode. You will also need a basic understanding of Apple's new programming language, Swift 3. We assume that you can download, install, and use the latest version of Xcode to create an application and run it on the iPhone simulator.

What You Need to Have

In terms of hardware, you need an Intel-based Macintosh running Mountain Lion (OS X 10.8) or later. Regarding software, you need Xcode 8.x since that is the current version to include Swift 3. You can download Xcode from the App Store or Apple's developer web site at http://developer.apple.com.

What's in This Book?

In Chapter 1, you'll learn about what Sprite Kit is and how you create a new Sprite Kit game using Xcode. You will then dive in and create the beginnings of a Sprite Kit game starting from scratch. You will learn about `SKNode`s and their subclasses, and you'll use an `SKSpriteNode` to add both a background node and a player node.

In Chapter 2, we will step back a bit and give you a deeper look at Sprite Kit scenes, including how scenes are built and why the order they are built in can change your game. The chapter will close with a discussion of Sprite Kit coordinate systems and anchor points as they relate to `SKNode`s.

In Chapter 3, you'll work with Sprite Kit's physics engine and collision detection. The chapter will begin with a discussion of `SKPhysicsBody`—the class used to simulate collision detection. You will then turn on gravity in the game world and see how that affects the nodes. After that, you will add a touch responder to propel the `playerNode` up into space, and finally you will learn how to handle node collisions.

In Chapter 4, you'll start adding some real functionality to your game. You'll begin by making some small changes to the current `GameScene`. After that, you will add additional orb nodes to collide into. You will then add scrolling to your scene, allowing you to make it look like the player is flying through space collecting orbs. Finally, you will start using the phone's accelerometer to move the player along the x-axis.

In Chapter 5, you'll refactor the orb node's layout one last time with the goal of enhancing playability. After that, you will learn how you can use `SKActions` to move an `SKSpriteNode` back and forth across the scene and then make that same node rotate forever. The chapter will close with a look at how you can add colorizing effects to an `SKSpriteNode` using a colorize action.

In Chapter 6, you'll see how to define particle emitters and how to leverage them in Sprite Kit games. After that, you will learn how you can use them to add engine exhaust to the `playerNode` whenever an impulse is applied to the `physicsBody`.

In Chapter 7, you'll see how you can use `SKLabelNode`s to add text to your Sprite Kit games. Specifically, you'll see how you how to add a label that keeps up with the number of impulses remaining for the spaceman to use, and then you'll see how you can add scoring to the game to keep up with the number of orbs the spaceman has collected.

In Chapter 8, you'll learn how to implement scene transitions using Sprite Kit's `SKTransition` class. You will look at some of the different types of built-in transitions Sprite Kit makes available to you. You will also see how you can control each scene during a transition. At the end of the chapter, you will take your newfound knowledge and add a menu scene to your SuperSpaceMan game.

In Chapter 9, you'll learn some Sprite Kit best practices; specifically, you will see how you can create your own subclasses of `SKSpriteNode` so that you can better reuse your nodes. You will then move on to changing your game to load all the sprites into a single texture atlas that you can reference when creating all future sprites. After that, you will move on to externalizing some of your game data so that designers and testers can change the game play. Finally, you will close out the chapter when you prune your node tree of all nodes that have fallen off the bottom of the screen.

In Chapter 10, you'll learn about what Scene Kit is and how to create a new Scene Kit game using Xcode. You will then dive in and create the beginnings of a Scene Kit game starting from scratch. You will learn to about `SCNScene` and `SCNodes` with a Scene Kit primer.

In Chapter 11, you'll learn more about the scene graph and some of the basics of Scene Kit. You will start to create your game by loading the spaceman from his Collada file. You will also learn about the Scene Kit primitive geometries by adding these as objects for the spaceman to avoid.

In Chapter 12, you'll learn how Scene Kit uses lighting and the type of lighting that is available to you in Scene Kit. You will also examine how materials are added onto the `SCNNode`, as well as how the camera is used within the scene.

In Chapter 13, you'll learn about the basics of animating the objects in your game. You will see a couple of different ways to animate the nodes to give you more than one way to do your animations. Once you have completed this chapter, all your objects will move within the scene.

In Chapter 14, you'll learn about collision detection within the scene. You will learn how to move the spaceman around the scene. Once you have the spaceman moving, you will learn how to detect when the spaceman runs into an obstacle.

In Chapter 15, you'll learn how to use a Sprite Kit scene within the Scene Kit scene. The chapter will show you how to create a screen to show you a timer that you will start when the user starts the game. The chapter will also show you how to display a "game over" screen and then restart the game.

In Chapter 16, you will learn the basics of the SceneKit Editor. This chapter will give you a basic understanding of creating a scene and various nodes visually in the editor. No coding will be done, but rather drag and dropping nodes and using the various editors to manipulate the objects.

Part

I

Introduction to Spritekit

Swift and Sprite Kit

In this part of this book, we will cover the basics of Sprite Kit including how you render and animate sprites, add physics and collision detection, and control your game play with the accelerometer. You will also look at how you add particle emitters to enhance the appearance of your game. We will cover everything you need to know to create your own Sprite Kit game.

Setting Up Your Game Scene and Adding Your First Sprites

SpriteKit is Apple's exciting 2D game framework that was first released in September 2013 with iOS 7. It is an animation and graphics rendering framework that gives you the power to easily animate textured images, play video, render text, and add particle effects. It also includes an integrated physics library. SpriteKit is the first-ever game engine formally built into the iOS SDK.

In this chapter you will learn what SpriteKit is and how to create a new SpriteKit game using Xcode. You will then move on and create the beginnings of a SpriteKit game starting from scratch. You'll learn about `SKNodes` and their subclasses and you'll use an `SKSpriteNode` to add both a background node and a player node to your game.

What You Need to Know and Have

This section of this book assumes you have a basic understanding of how to build iPhone applications using Xcode and the Xcode Simulator. It also assumes you have a basic knowledge of the iOS/Mac programming language Swift. If you are not familiar with Swift, there is a brief introduction in the appendix at the back of this book.

This book doesn't cover how to program. It focuses only on SpriteKit game programming.

To complete all the examples in the book, you will need to have an Intel-based Macintosh running OS X 10.11 (El Capitan) or newer. You will also need Xcode 8+ installed. You can find both of these in the Apple App Store.

Introducing SuperSpaceMan

We feel the best way to learn anything is to do it. Therefore, in this book you are going to dive right in and create your own game. You will start off with the basic code for a 2D game, and you will add new features to the game as we introduce new topics with each chapter. At the end of the book, you will have a complete game.

The game you are going to create is inspired by Sega's popular Sonic Jump Fever (https://itunes.apple.com/us/app/sonic-jump-fever/id794528112?mt=8). It is a vertical scroller that accelerates the main character through obstacles and collectables, increasing your score as you collect rings.

This game is similar in that it is a vertical scroller, but your main character is going to be a spaceman who hurtles through space collecting power orbs while trying to avoid black holes that will destroy him.

Creating a Swift SpriteKit Project

Before you can get started, you will need to create a Swift SpriteKit project. Open Xcode and complete the following steps:

1. Click File ➤ New ➤ Project.

2. Select iOS.

3. Select the Game icon from the Application group. The choose template dialog should look like Figure 1-1.
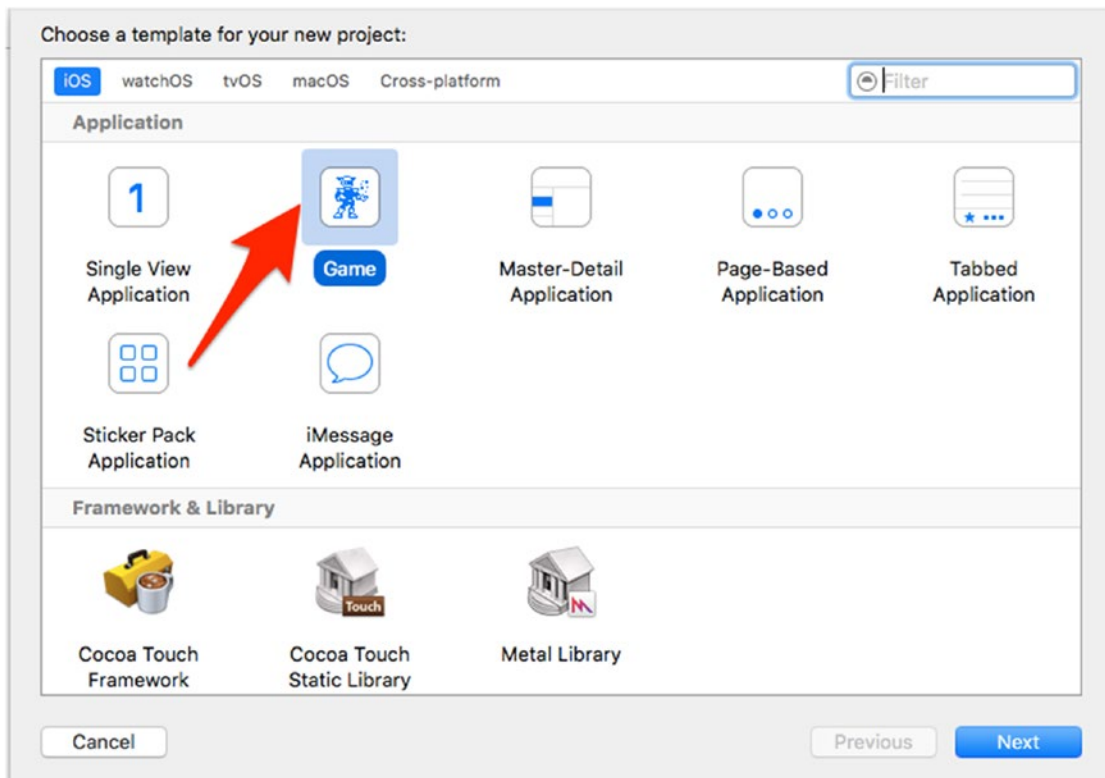
Figure 1-1. *The choose template dialog*

4. To move on, click the Next button.

5. Enter **SuperSpaceMan** for Product Name, **Apress** for Organization
    Name, and **com.apress** for Organization Identifier.

6. Make sure that Swift is the selected language, SpriteKit is the
    selected game technology, and iPhone is the selected device.

7. Before you click Next, look at Figure 1-2. If everything looks like that,
    click Next and select a good place to store your project files. Click
    Create.

Choose options for your new project:

Product Name:	SuperSpaceMan
Team:	Add account...
Organization Name:	Apress
Organization Identifier:	com.apress
Bundle Identifier:	com.apress.SuperSpaceMan
Language:	Swift
Game Technology:	SpriteKit
Devices:	iPhone

☐ Integrate GameplayKit
☐ Include Unit Tests
☐ Include UI Tests

Cancel   Previous   Next

Figure 1-2. The choose project options dialog

Note   You will notice you are creating an iPhone-only game. That's only because the game you are
creating lends itself better to the iPhone. Everything we cover in this book translates to the iPad just
as well.

You now have a working SpriteKit project. Go ahead and click the Play button to see what you have created. If everything went OK, you will see your new app running in the simulator.

> **Note** The Xcode simulator may take a while to start on some slower machines. Simulating SpriteKit apps can be very taxing on your processors.

It doesn't do a whole lot yet, but there is more to it than displaying "Hello, World!" Tap the simulator screen a few times. You will see rotating boxes displayed wherever you tap. Depending on where you tapped, you should see something similar to Figure 1-3.
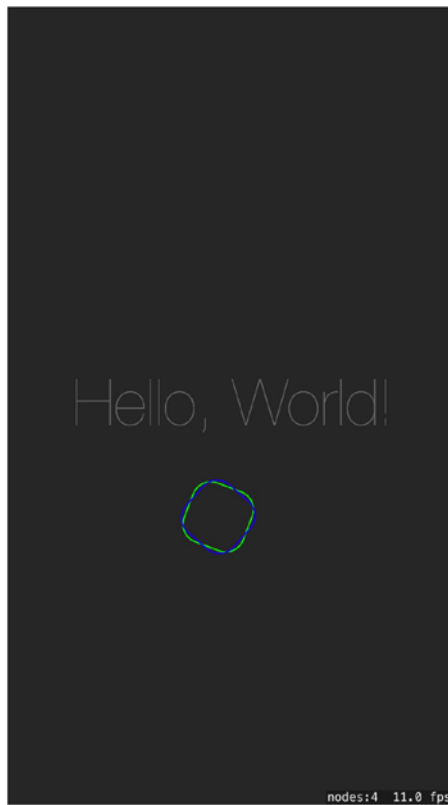
Figure 1-3. The SpriteKit sample application