

# Contents

<b>WeSplit</b>	<b>5</b>
WeSplit: Introduction	
Understanding the basic structure of a SwiftUI app	
Creating a form	
Adding a navigation bar	
Modifying program state	
Binding state to user interface controls	
Creating views in a loop	
Reading text from the user with TextField	
Creating pickers in a form	
Adding a segmented control for tip percentages	
Calculating the total per person	
WeSplit: Wrap up	
<b>Guess the Flag</b>	<b>46</b>
Guess the Flag: Introduction	
Using stacks to arrange views	
Colors and frames	
Gradients	
Buttons and images	
Showing alert messages	
Stacking up buttons	
Showing the player's score with an alert	
Styling our flags	
Guess the Flag: Wrap up	
<b>Views and Modifiers</b>	<b>71</b>
Views and modifiers: Introduction	
Why does SwiftUI use structs for views?	

- What is behind the main SwiftUI view?
- Why modifier order matters
- Why does SwiftUI use “some View” for its view type?
- Conditional modifiers
- Environment modifiers
- Views as properties
- View composition
- Custom modifiers
- Custom containers
- Views and modifiers: Wrap up

## **BetterRest** **98**

- BetterRest: Introduction
- Entering numbers with Stepper
- Selecting dates and times with DatePicker
- Working with dates
- Training a model with Create ML
- Building a basic layout
- Connecting SwiftUI to Core ML
- Cleaning up the user interface
- BetterRest: Wrap up

## **Word Scramble** **129**

- Word Scramble: Introduction
- Introducing List, your best friend
- Loading resources from your app bundle
- Working with strings
- Adding to a list of words
- Running code when our app launches
- Validating words with UITextChecker
- Word Scramble: Wrap up

## **Animation** **154**

- Animation: Introduction

Creating implicit animations  
Customizing animations in SwiftUI  
Animating bindings  
Creating explicit animations  
Controlling the animation stack  
Animating gestures  
Showing and hiding views with transitions  
Building custom transitions using ViewModifier  
Animation: Wrap up