

CUSTOM CONTROLS IN iOS



HANDS-ON CHALLENGES

Custom Controls in iOS

Catie & Jessy Catterwaul

Copyright ©2017 Razeware LLC.

Notice of Rights

All rights reserved. No part of this book or corresponding materials (such as text, images, or source code) may be reproduced or distributed by any means without prior written permission of the copyright owner.

Notice of Liability

This challenge and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use of other dealing in the software.

Trademarks

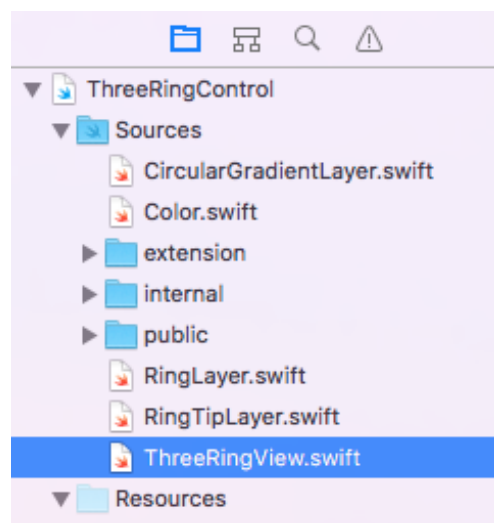
All trademarks and registered trademarks appearing in this book are the property of their own respective owners.

Challenge #8: Control Animation

By Catie & Jessy Catterwaul

Currently, there's no way to alter ThreeRingView's animation duration. In this challenge, you'll add that ability to its public API.

Open ThreeRingControl.playground, and select ThreeRingView.swift in the Sources folder. (Hit command-1 first, to show the Project navigator, if it is hidden.)



Add a new property in the public section of the API, to store the animation duration.

```
public final class ThreeRingView: UIView {  
  //MARK: public  
  public var animationDuration = 1.5
```

Then, use that property in the animate method, in place of the hard-coded constant that you had before.

```
func animate(  
  ring: Ring,
```

```
        value: CGFloat
    ) {
        CATransaction.begin()
        CATransaction.setAnimationDuration(animationDuration)
        defer {
            CATransaction.commit()
        }

        ringLayers[ring]!.animate(value: value)
    }
}
```

Back in the playground, set this new property in the for loop near the top, to a value of your choosing.

```
for threeRingView in threeRingViews {
    threeRingView.innerRingValue = 0.9
    threeRingView.middleRingValue = 0.3
    threeRingView.outerRingValue = 0.6

    // Snappy timing!
    threeRingView.animationDuration = 0.5
}
```

Click on each of the three-ring views, in the live view. The one on top hasn't changed meaningfully from the demo, because `animationDuration` does not affect what happens when the rings' values are set directly. The bottom view, however, will animate with the duration you assigned!

Stay Tuned!

In the next challenge, you'll use the `animationDuration` API, as well as `ThreeRingView`'s other public properties, in a buildable app.