

# **iOS Animation with Swift**

Hands-On Challenges

# iOS Animation with Swift Hands-On Challenges

Copyright © 2014 Razeware LLC.

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.

This book 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 or other dealings in the software.

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



# Challenge I: Shape animation

In this challenge you will get your hands dirty with shape animations. You've seen how to squish the circular avatar shape to an oval, but now you will make it morph into a plain square.

Don't forget to open the challenge starter project before coding your solution. The challenge starter project develops the demo further to a point in which the searching for opponenet animation calls a new method on the avatar view called `animateToSquare()`. Your task is to add the shape animation to that new method.

Start by defining a new `CABasicAnimation` on a `path` property:

```
let morphAnimation = CABasicAnimation(keyPath: "path")
morphAnimation.duration = 0.25
morphAnimation.fromValue = circleLayer.path
morphAnimation.toValue = UIBezierPath(rect: bounds).CGPath
morphAnimation.removedOnCompletion = false
morphAnimation.fillMode = kCAFillModeBoth
```

Notice how easy it is to create a square shape by using the convenience `UIBezierPath(rect:)` initializer. The rest of the code is similar to what you already know how to do from the demo video.

Next add the two lines needed to add the animation to both the shape layer and the mask layer of the avatar view:

```
circleLayer.addAnimation(morphAnimation, forKey:nil)
maskLayer.addAnimation(morphAnimation, forKey: nil)
```

Build and run the project to see how CoreAnimation animates the circular avatars to square ones in the very end of the faux search for opponent animation.

