

# CALayers

Hands-on Challenges

# CALayers Hands-On Challenges

Copyright © 2015 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 D: Final Challenge

You'll probably have noticed that the avatar's image view appears to bleed outside of the border when a corner radius is applied:



In this final challenge, you will take everything you've learned and put it together to work around this problem. Here are a few hints to get you started, and if you get stuck check out the challenge solution!

## Hints

You only need to modify one file: **AvatarView.swift**.

1. Remove the `imageView` and all corresponding code.
2. Replace it with a `CAShapeLayer`. You're choosing a shape layer so that it will stroke a desired path (the circle).
  - a. Create a property for it and call it `layerAvatar`.
  - b. In `setup()`, set the fill color to `nil`, line width to 10, contents gravity to `kCAGravityResizeAspectFill`, and add it as a sublayer to the main layer.
  - c. In `configure()`, set its contents based on `imageAvatar` and set its stroke color based on `strokeColor`.
  - d. In `layoutSubviews()`, calculate `layerAvatar`'s height as the bounds height – margin – the label height. Use this to set the frame to the correct spot. Draw a diagram if necessary.



3. Here's the trick:

- a. Create a second shape layer called `maskLayer` – an oval inside `layerAvatar`'s bounds.
- b. Set `layerAvatar`'s mask to `maskLayer`. This cuts out everything outside the path.
- c. Set `layerAvatar`'s path to `maskLayer`'s path. This makes the shape layer stroke the circular path.

Build and run, and you should no longer see a bleeding edge!

