

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 F: Basic Layer Animations

In this challenge you will re-create the login button animation you created for Challenge A. This time you will use `CABasicAnimation` and animate the button's layer.

Before you create your first layer animation, you need to remove the one in place. Scroll to `viewWillAppear(animated:)` and remove the following two lines:

```
loginButton.center.y += 30.0
loginButton.alpha = 0.0
```

Then remove the line that animates the login button inside `viewDidAppear(animated:)`:

```
UIView.animateWithDuration(0.5, delay: 0.5,
    usingSpringWithDamping: 0.5, ...
```

Build and run the app to make sure that the login button appears just fine under the text fields.

Now it's time to move that layer! At the bottom of `viewWillAppear(animated:)` add:

```
loginButton.layer.position.y += 300.0
```

This should position the button just below the screen edge. Now scroll to `viewDidAppear(animated:)` and add at the bottom the code to animate the button on its way up the screen:

```
let flyUp = CABasicAnimation(keyPath: "position.y")
flyUp.toValue = loginButton.layer.position.y - 300
flyUp.duration = 1.0
flyUp.setValue("loginButton", forKey: "name")
flyUp.delegate = self
loginButton.layer.addAnimation(flyUp, forKey: nil)
```

Much like for the text fields in the video tutorial demo you prepare a `CABasicAnimation` and give the animation a name so that you can adjust the final layer position in the `animationDidStop(anim:, finished:)` delegate method.

Speaking of which you will also need to add the code to set the button layer's position in that delegate method. Scroll to `animationDidStop` and inside the first if statement add (new code is highlighted):



```
if let name = nameValue {  
    if name == "loginButton" {  
        loginButton.layer.position.y -= 300  
    }  
    if ...
```

This should nicely show the button at its final position after the animation has complete.

