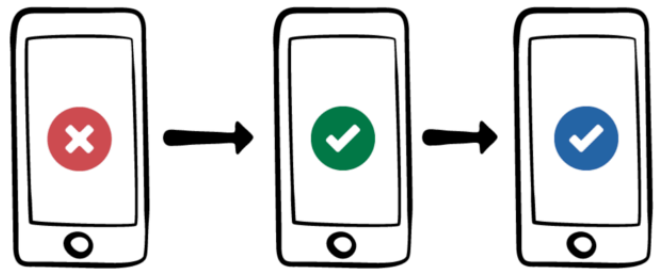


# BEGINNING iOS UNIT & UI TESTING



**HANDS-ON CHALLENGES**

## Beginning iOS Unit and UI Testing

Joshua Greene

Copyright ©2016 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.

# 6 Recording UI Tests - Challenge

By Joshua Greene

While the Xcode recorder can be finicky at times, it can really make it easy to write UI tests when it works well!

You still have to write assertions yourself, but this is much easier when you already have the rest of the code written for you. ;]

In this final challenge, you'll add one final UI test to the project.

## Challenge

Write a new UI test that navigates to the About screen and taps on the "Show/Hide Copyright Notice" button to show the RW logo and copyright. Then, tap the button again to hide the logo and copyright.

## Hints

You can use the Xcode UI Recorder or write the entire test manually— that's up to you. I used the recorder to find how to reference the RW image. You can do this too by navigating to the About screen, tapping the image to get a reference to it and then deleting the `tap()` call.

Here's a couple tips if you do use the recorder:

- Remember to scroll by swiping on the cell text. This should help the recorder determine the correct element.
- If the recorder crashes (it probably will at least once), you *don't* have to restart the entire app. Instead, you can simply press "Ok" on the alert and then start the recorder again by pressing the red, record button.

You should be able to do this challenge without too much trouble. If you get stuck,

check out the next page for a sample solution.

## Solution

Open `StackReviewUITests.swift` and add the following test method to the class:

```
func testAboutDetailsCanHideShowCopyrightNotice() {  
    tapPancakeHousesButtonIfNeeded()  
    app.navigationBars.buttons["About"].tap()  
  
    let showHideCopyrightNoticeButton = app.buttons["Show/Hide Copyright  
Notice"]  
    showHideCopyrightNoticeButton.tap()  
  
    XCTAssertTrue(app.images["rw_logo"].exists)  
  
    showHideCopyrightNoticeButton.tap()  
    XCTAssertFalse(app.images["rw_logo"].exists)  
}
```