

Introducing

Auto Layout

Hands-On Challenges

Introduction to Auto Layout Challenge Instructions

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.



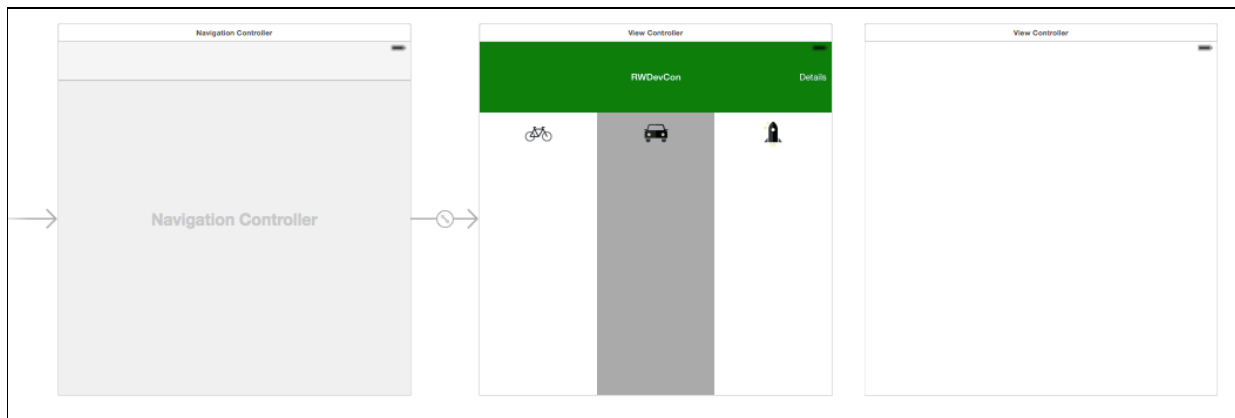
From Design to Layout

You're now familiar with all kinds of ways to add constraints – from buttons, menus, dragging between views, code, and the visual format language. You're well on your way to creating even more interesting layouts for your apps!

In this challenge, you'll reproduce one more view controller for the **About RWDevCon** app from a screenshot.

Another view controller

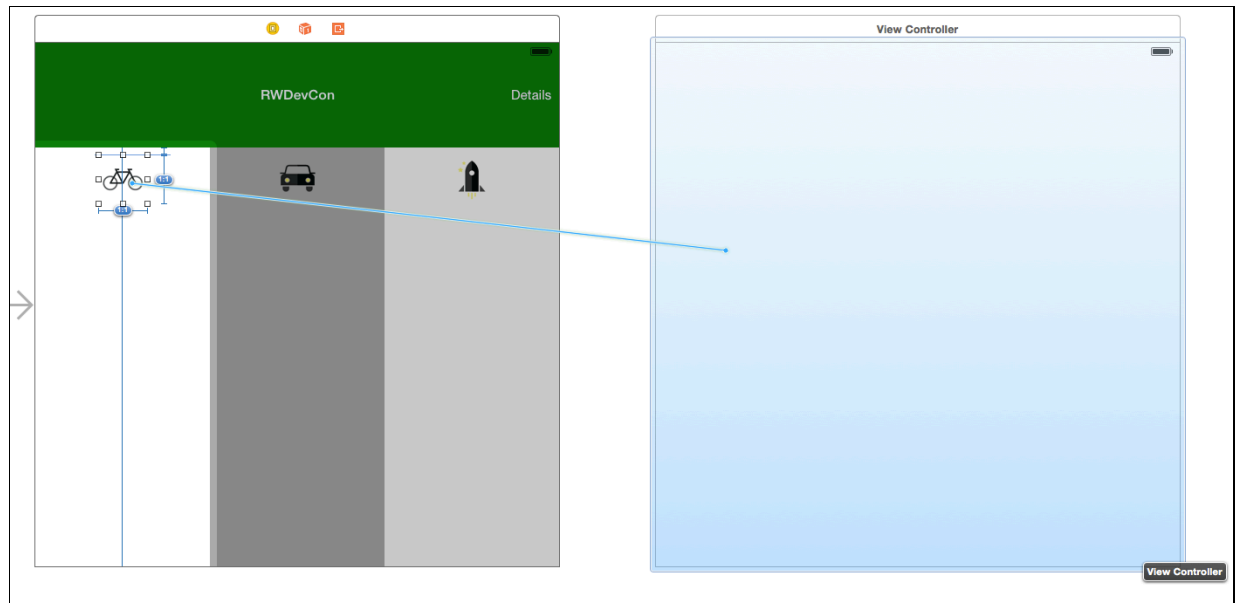
Open the storyboard and drag out another view controller.



Just in case you want to test the new view controller on device, you can connect a segue from one of the buttons.

Control-drag from the beginner (bicycle) button to your new view controller.





In the popup, select **Show** in the **Action Segue** heading.

Now when you run the app, you can tap on the bicycle to show the second view controller.

Target design

You'll find the required images in the asset catalog: **beginner** for the bicycle icon and **rwdevcon-challenge** for the logo.

Also, here's the text to paste into the label:

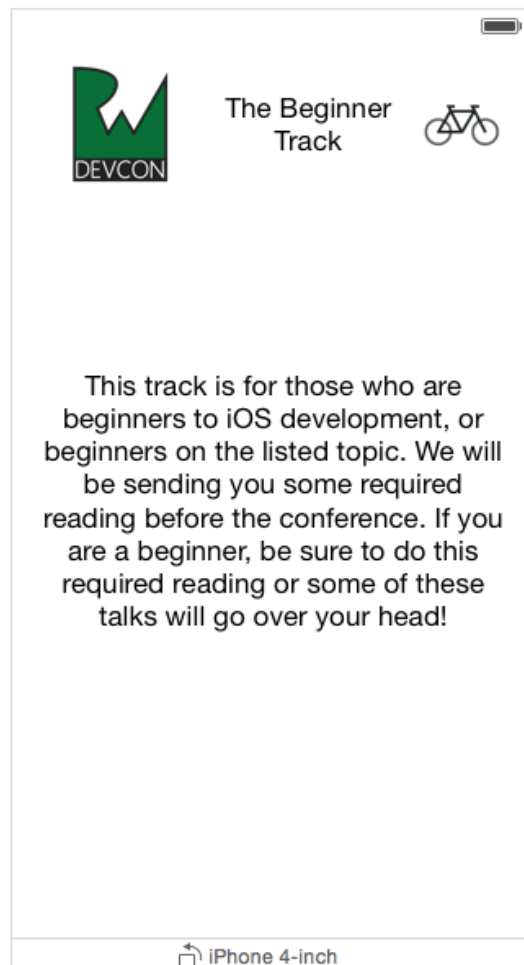
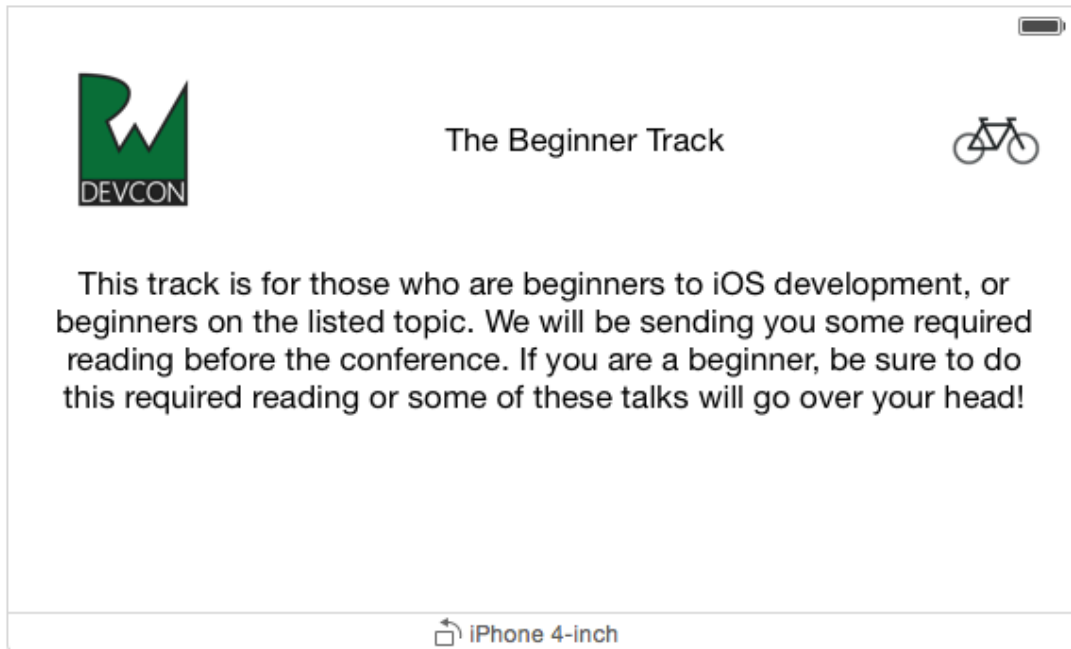
This track is for those who are beginners to iOS development, or beginners on the listed topic. We will be sending you some required reading before the conference. If you are a beginner, be sure to do this required reading or some of these talks will go over your head!

If you're confident, just look at the screenshot below and get started! If you get stuck, there are a few hints on the following pages.

I'd suggest building this in interface builder but if you want an über challenge, add a new class for the view controller and build it in code!

Here's what the view controller should look like, in a few sizes and interface orientations:







The Beginner Track



This track is for those who are beginners to iOS development, or beginners on the listed topic. We will be sending you some required reading before the conference. If you are a beginner, be sure to do this required reading or some of these talks will go over your head!

iPhone 5.5-inch



The Beginner Track



This track is for those who are beginners to iOS development, or beginners on the listed topic. We will be sending you some required reading before the conference. If you are a beginner, be sure to do this required reading or some of these talks will go over your head!

iPhone 5.5-inch



Hints

Here are some hints, going from top to bottom in the layout.

1. Note that the RWDevCon icon, “The Beginner Track” label, and the bicycle are all **Center Y** aligned. That is, you could draw a straight horizontal line through their centers.
2. The easiest way to get the two images is just in an image view with a fixed size. The RWDevCon image is 90x90 points, and the bicycle image is 57x57 points.
3. “The Beginner Track” has its leading and trailing space to its neighbors set to keep it between the two images. The label has its number of lines set to 0 so it can wrap nicely if needed.
4. The blurb of text is vertically centered in the superview.

Solution

For a sample solution, check out the resources for this session. You’ll find an Xcode project with the view controller built out in the storyboard.

