
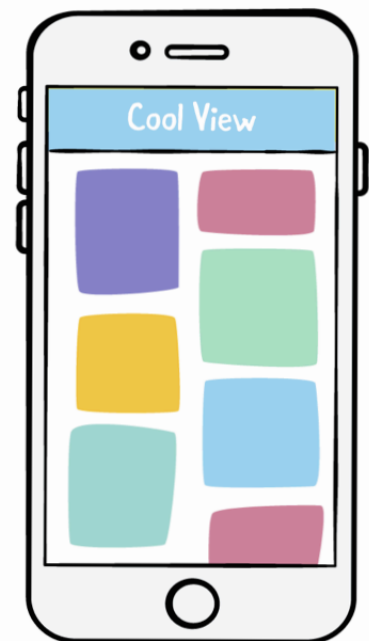


CUSTOM 
COLLECTION
 VIEW
LAYOUT 



HANDS-ON CHALLENGES

Custom Collection View Layout

Michael Briscoe

Copyright ©2017 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.

Overlapping Cells

Now that you've got the carousel layout set up, let's enhance it with overlapping cells—sort of like a deck of cards. You'll also want the "featured" cell to appear on top of all the other cells. This trick will give the carousel layout a sense of depth.

There are just two lines of code to change the layout properties and attributes to easily accomplish this task.

Hints:

1. You'll need to change the layout's line spacing to implement overlapping.
2. Change the `zIndex` of the scrolling items so that the "featured" cell has the highest value.

Before you turn the page for our solution, be sure to give it a try for yourself first!

Solution

Overlapping Cells

Open **MasterViewController.swift** and add the following line of code to the end of `viewDidLoad()`:

```
layout.minimumLineSpacing = -(layout.itemSize.width * 0.5)
```

This code sets the spacing between lines to a negative value, bringing the cells closer together and appearing to overlap.

Do a build and run to see this effect.



We're getting close, but because the cells share the same `zIndex` they all seem to blend together. Let's fix that.

Changing Z Order

Open **CharacterFlowLayout.swift** and add the following to the end of `changeLayoutAttributes(_:)`:

```
attributes.zIndex = Int(alpha * 10)
```

Here we are changing the cell's `zIndex` attribute based on the changing `alpha` value, ensuring that the cell with the highest opacity will also have the highest `z` order.

Build and run again to see the difference.

Beginning Auto Layout



Now the "featured" cell pops out above the rest!