


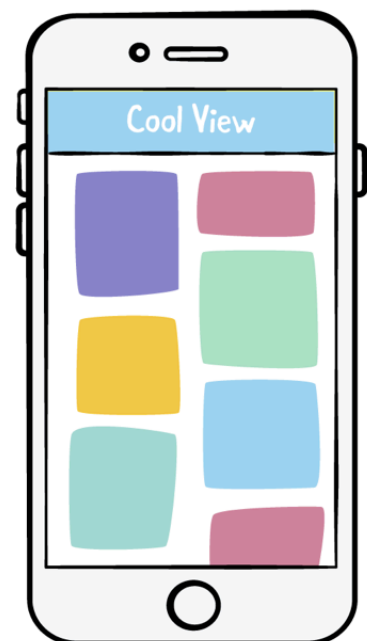


CUSTOM   
COLLECTION  
 VIEW  
LAYOUT 



 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.

## Adding a description label to custom cell

Our playing card looks a little sparse. It would be great if we knew a little bit more about each character. Let's add a label below the character's title to display more information.

### Hints:

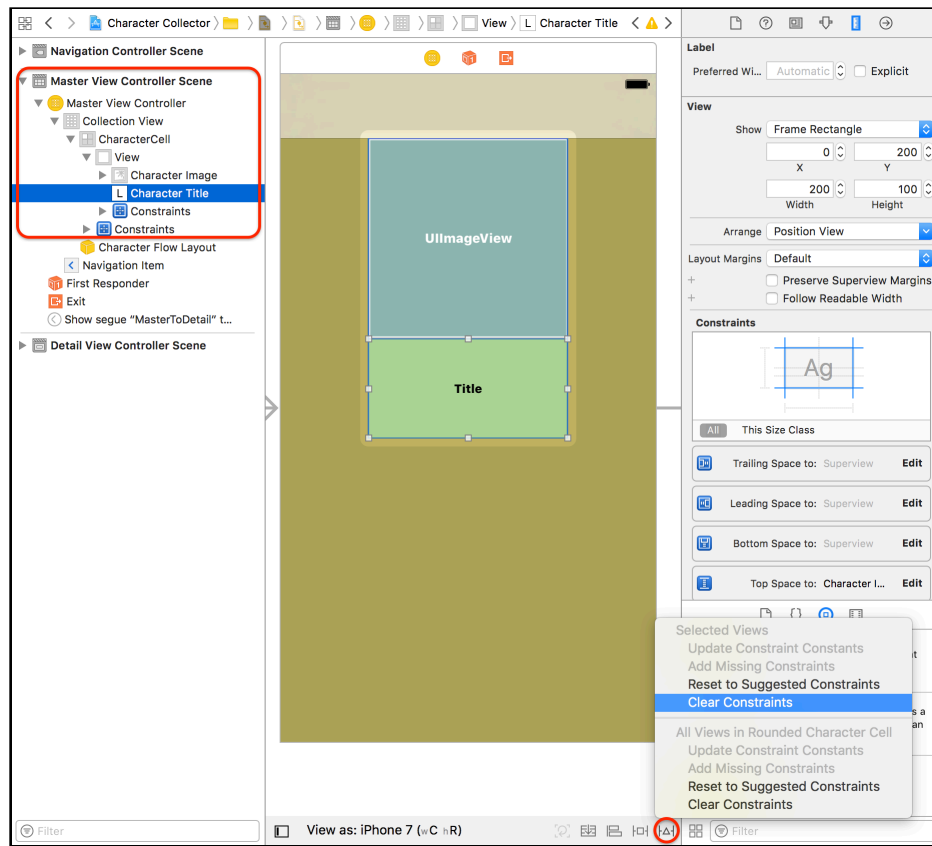
1. First, update the storyboard with a new label.
2. Add an outlet to reference the label.
3. Update the custom cell class to implement the display of the character's description.

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

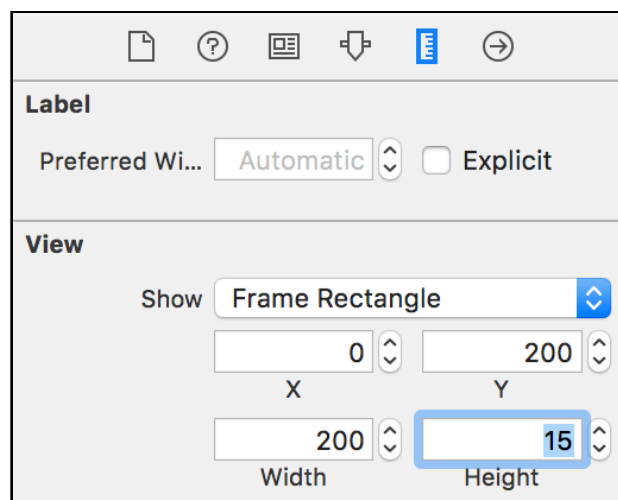
## Solution

Open **Main.storyboard** and expand the **Master View Controller Scene** from the Document Outline. Expand the **Collection View** until you can see *Character Title*.

Select **Character Title**, then press the **Resolve Auto Layout Issues** button. Choose **Selected Views/Clear Constraints**:



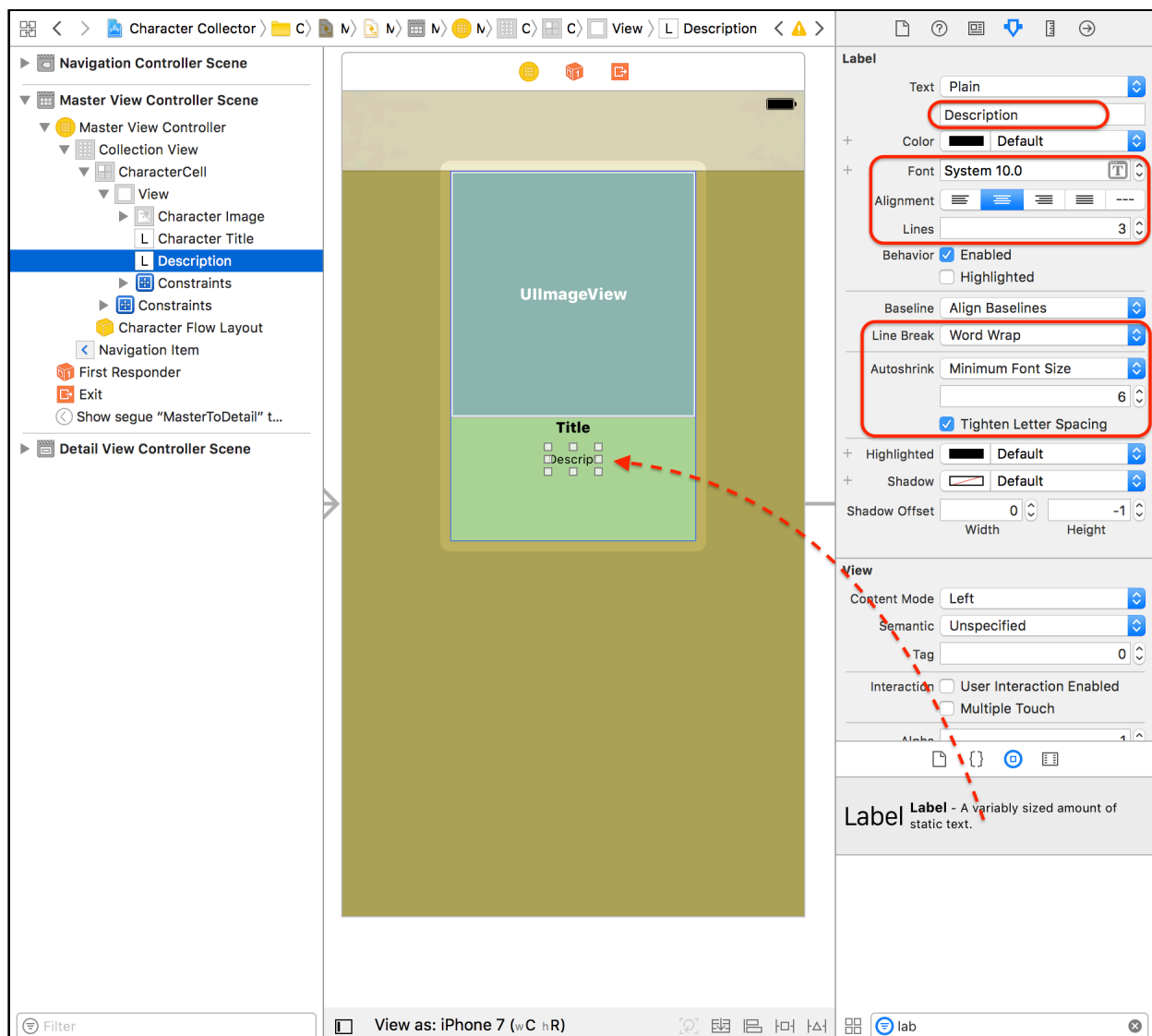
With *Character Title* still selected set its **Height** to **15**:



Now, drag a **Label** to the CharacterCell, just below the *Title* label.

With the label selected choose the **Attributes Inspector** and change the following settings:

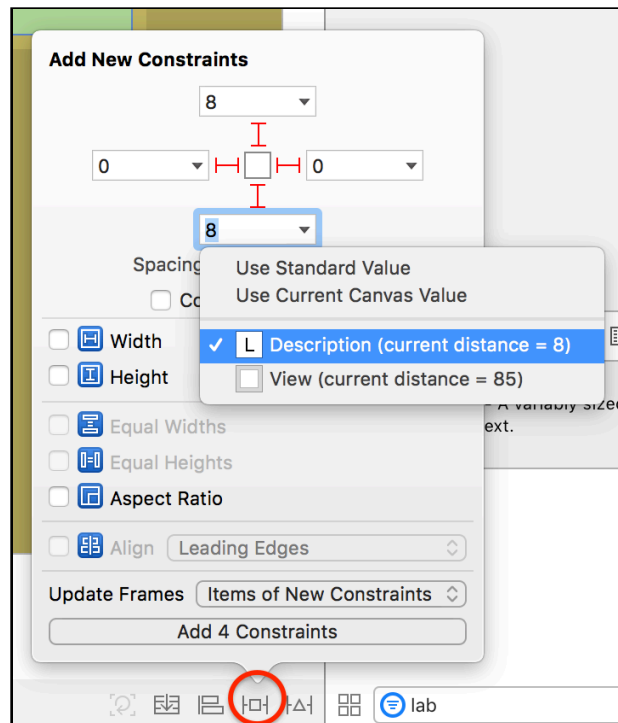
1. **Title** to Description
2. **Font** to System 10.0
3. **Alignment** to Center
4. **Lines** to 3
5. **Line Break** to Word Wrap
6. **Autoshrink** to Minimum Font Size, 6
7. **Tighten Letter Spacing** to enabled



Next select the **Title** label and click the **Pin** button and add the constraints shown below:

- **Top Constraint** = 8 (in relation to Character Image)
- **Leading Constraint** = 0
- **Trailing Constraint** = 0
- **Bottom Constraint** = 8 (in relation to the Description label)

Make sure that **Update Frames** is set to Items of New Constraints, then click **Add 4 Constraints**.



Switch to the **RoundedCharacterCell.swift** file and add an outlet to the top of the class:

```
@IBOutlet weak var characterInfo: UILabel!
```

While we're here, let's modify the character variable's didSet block by adding the following line just below the line that sets characterTitle.text:

```
characterInfo.text = theCharacter.description
```

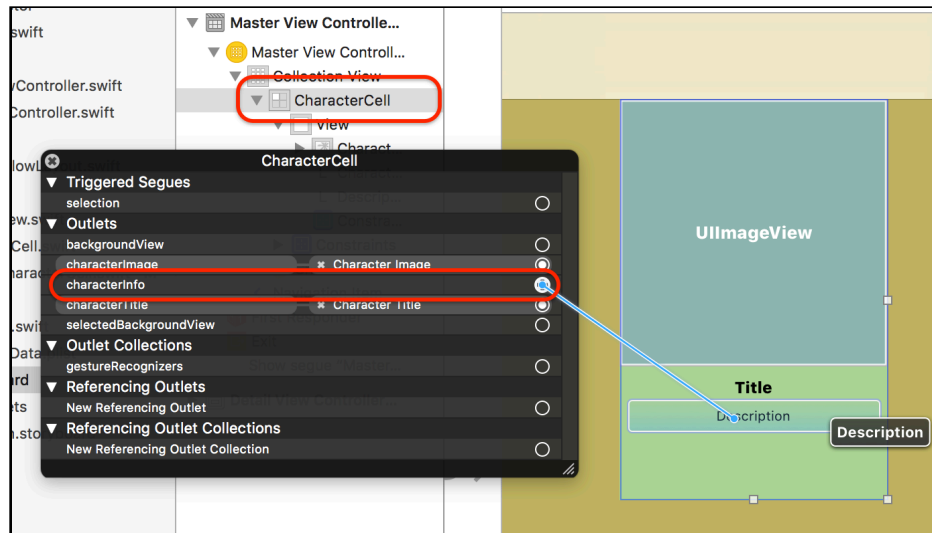
This will set the Description label to a brief description whenever the character variable is updated.

Now add this to the end of prepareForReuse():

```
characterInfo.text = ""
```

This is just a little cleanup so that the label is empty when it enters the reuse pool.

Finally, we need to connect the outlet to the Description label. Open **Main.storyboard** and **control-click** on the CharacterCell to bring up it's connections panel. Connect the **characterInfo** outlet to the Description field.



Now do a build and run.



The character cards look a lot better with some *slightly* humorous character descriptions!