

# Collection Views

Hands-On Challenges

# Collection Views

## Hands-On Challenges

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.

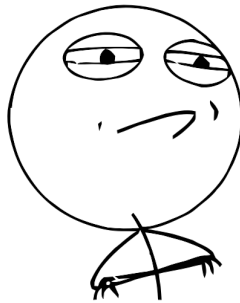
All trademarks and registered trademarks appearing in this book are the property of their respective owners.



# Challenge B: Looking Good!

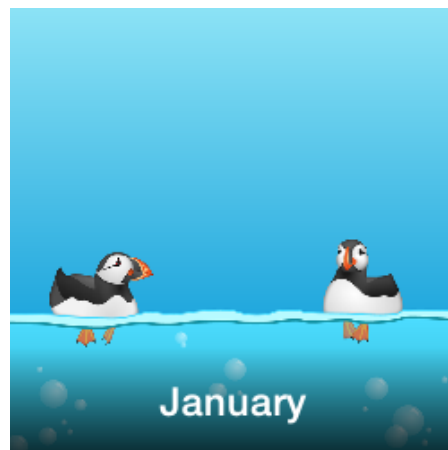
In the demo you created a custom cell, and added an image view which displays a thumbnail of the relevant the wallpaper. It looks pretty good as it is, but I bet you can do even better.

## CHALLENGE ACCEPTED



Your challenge this time to add a label to the custom cell which displays the paper's caption, as well as adding a gradient background behind the label so the caption can be read even on lighter wallpapers.

When you're finished, the cell should look something like this:



Pretty nice, right?

**Hint:** Have a dig around in the project folder and you might just come across a custom view class that you can use for the gradient background.

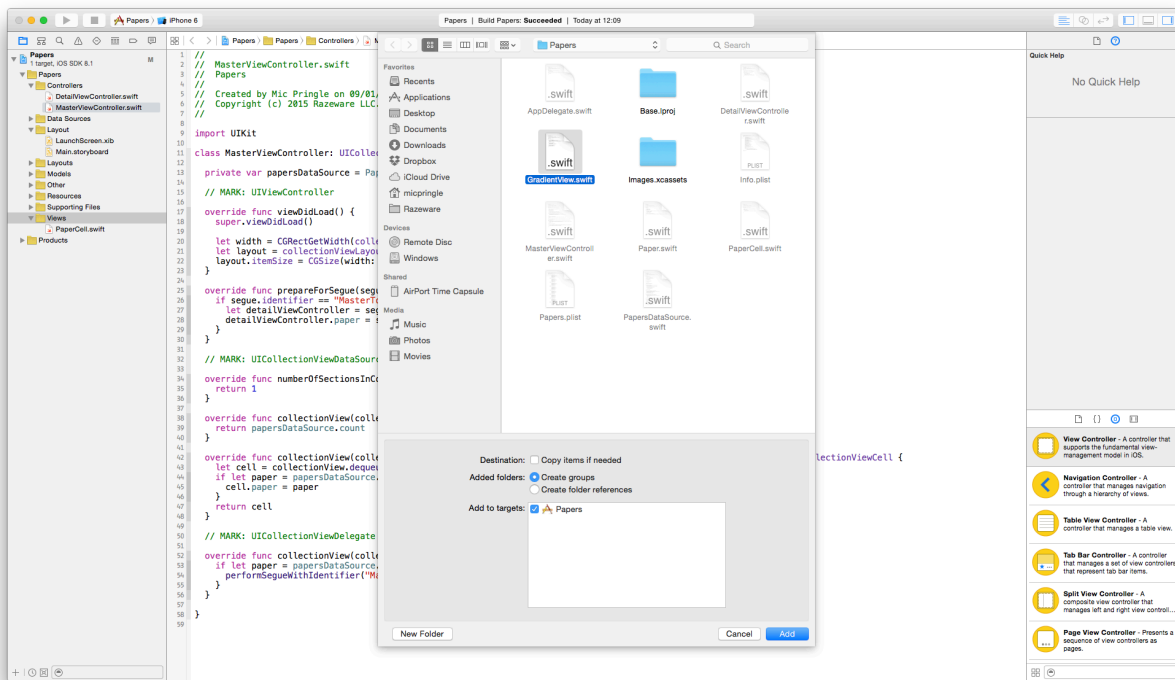
Before you turn the page for our solution, be sure to give it a try for yourself first!



# Solution

A ready-made Gradient View class has been provided for you in the project, so the first thing you need to do is import that into Xcode.

Right-click on the **Views** group and choose **Add Files to "Papers"...**. In the dialog box that appears, choose the **GradientView.swift** file and then click **Add**.



This class is simply a UIView subclass that draws a gradient as it's background.

Next, open **PaperCell.swift** from the **Views** group, and add the following IBOutlet property declarations just below the existing one:

```
@IBOutlet private weak var gradientView: GradientView!
@IBOutlet private weak var captionLabel: UILabel!
```

Here you're simply adding two outlets; one for the gradient background view, and one for the caption label.

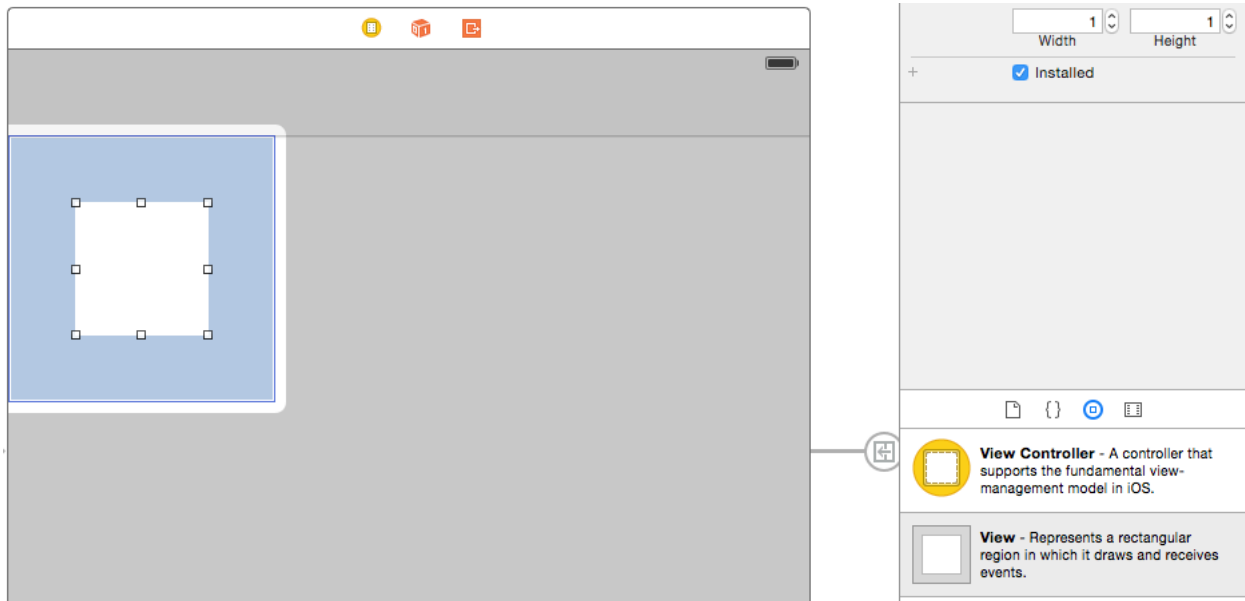
The final change you need to make to PaperCell is to update the `paper` property observer so that it sets the label's text to the paper's caption whenever it's set.

Add the following just below where you set the image on `paperImageView`:

```
captionLabel.text = paper.caption
```

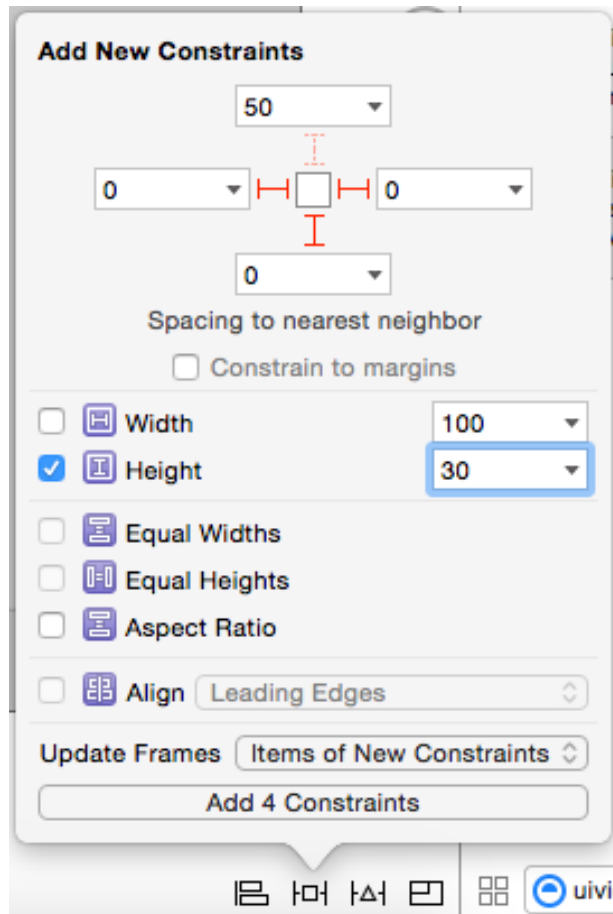


Now, open **Main.storyboard** from the **Layout** group. Find **View** in the **Object Library** and drag one onto the collection view cell. It appears quite near the bottom of the list, so it might be quicker for you to search by entering "uiview" in the search field.



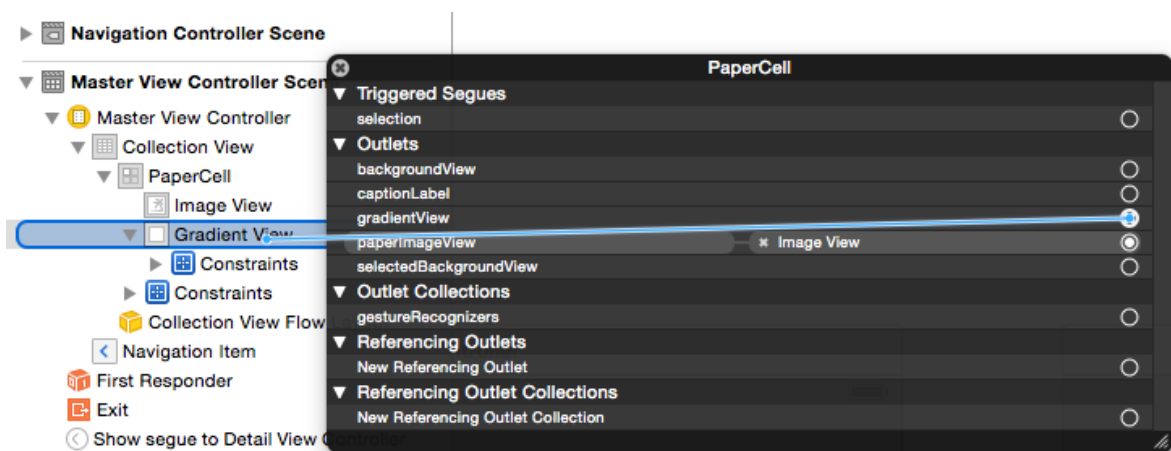
Select the view you just added, and in the Identity Inspector set the **Custom Class** to **Gradient View**. Next, in the Attributes Inspector change the **Background** color to **Light Gray Color**. Finally, click the Pin button located at the bottom of the storyboard canvas and add the following constraints:



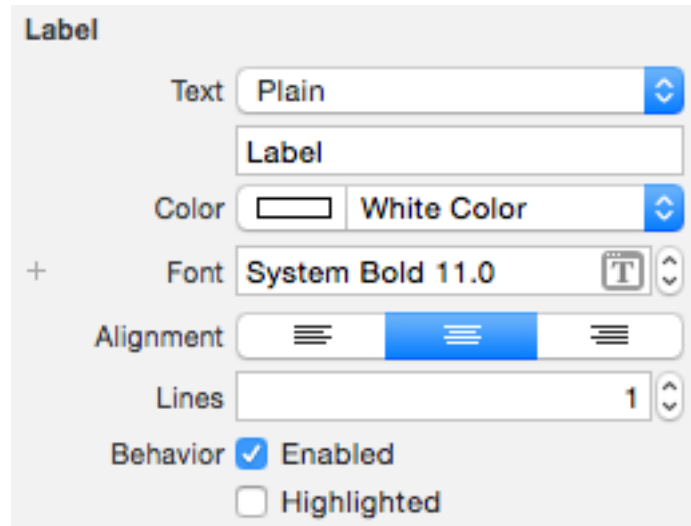


Make sure you uncheck **Constrain to margins**, and select **Items of New Constraints** from the Update Frames menu. Click **Add 4 Constraints** to add the constraints.

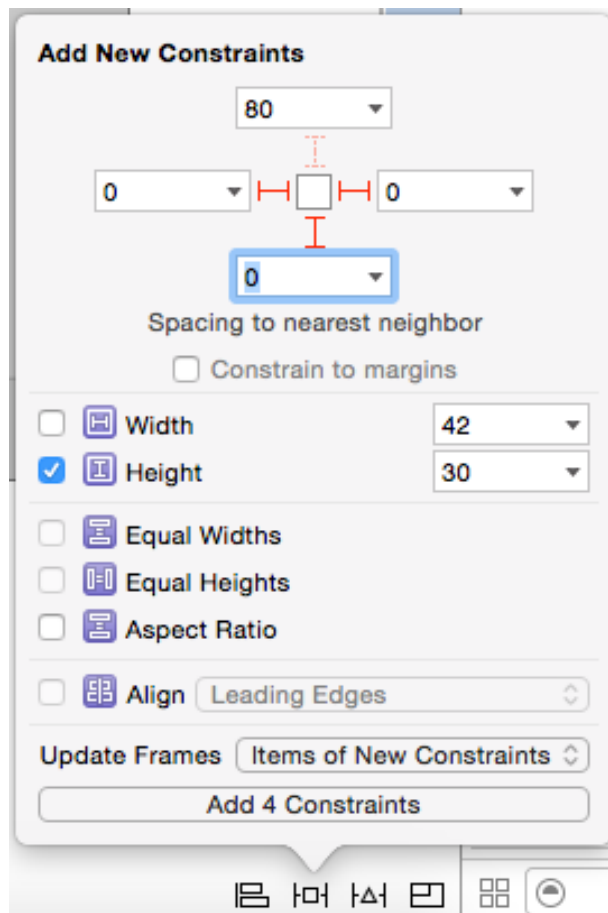
Next, right-click on PaperCell in the Document Outline and drag from the **gradientView** outlet in the popup to the **Gradient View** object in the Document Outline to connect the two:



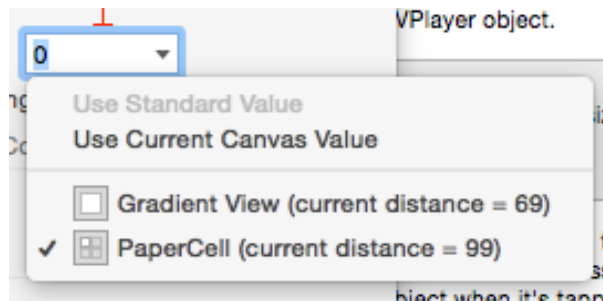
With the gradient view in-place, the last thing to do is to add the caption label. Drag a **Label** from the Object Library onto the cell. Select it, and in the Attributes Inspector set its **Color** to White Color, its **Font** to System Bold 11.0, and its **Alignment** to Center:



Then, again using the Pin button, add the following constraints:



Make sure you uncheck **Constrain to margins**, and select **Items of New Constraints** from the Update Frames menu. For the bottom constraint, make sure it's relative to PaperCell and not Gradient View:



Click **Add 4 Constraints** to add the constraints.

The final step is to hook up the labels outlet. As you did before, right-click on on PaperCell in the Document Outline and drag from the **captionLabel** outlet in the popup to the **Label** object in the Document Outline to connect the two.

Build and run. You'll see that the cells now display the caption associated with each paper, on top of a nice subtle gradient.

