Table Views in iOS

Hands-On Challenges

Table Views: Beginning to Advanced Hands-On Challenges

Copyright © 2014 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 per- mission of the copyright owner.

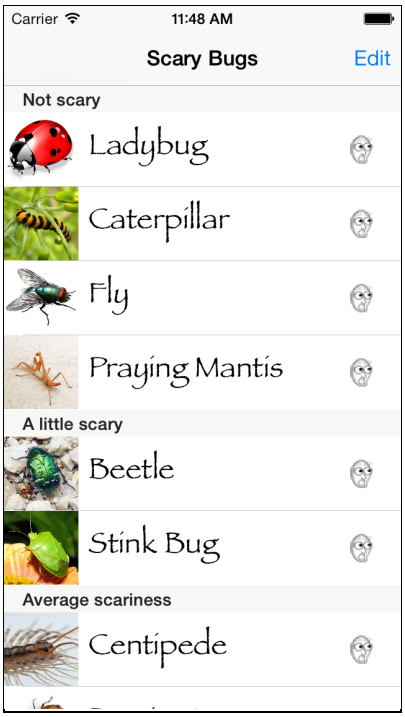
This book and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connec- tion 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 #6: Custom Rows

Your Scary Bug app is coming along well so far, but currently it looks like many other apps out there. It would be nice to have a custom look for your cells!

In this challenge, you will customize the scary bug cells so they look like this:



See if you can do this on your own based on what you learned on the video. If you get stuck, follow along with the full walkthrough below!

Full Walkthrough

Open the Scary Bugs project where you left it off in the last challenge, or use the starter project provided by the instructor.

First, open **BugTableViewController.m** and add this method that I forgot to tell you to add in the previous challenge:

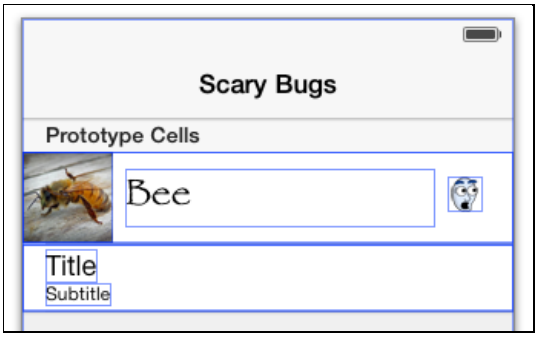
```
- (BOOL)tableView:(UITableView *)tableView
canMoveRowAtIndexPath:(NSIndexPath *)indexPath {
  BugSection *bugSection = self.bugSections[indexPath.section];
  if (indexPath.row >= bugSection.bugs.count && [self isEditing]) {
    return NO;
```

```
  } else {
    return YES;
  }
}
```

This prevents the user from trying to move the special "Add Bug" row.

Open **Main.storyboard**, select your table view, and set the **Prototype Cells** to 2. Select the second cell, and set the **Identifier** to **NewRowCell**.

Next select the first cell and set the **Row Height** to **60**, and the **Style** to **Custom**. Then layout the cell so it looks something like this:



The "shocked face" image shown here is **shockedface2_full.png**, which is already included in your project.

Try to get the Auto Layout constraints for this cell working if you can. **Tip**: you will need a width and height constraint on the bug image view to 60x60 (in addition to several other constraints on this view and others) to get this to work.

> **Note**: Stuck with the Auto Layout constraints? Check out our Auto Layout video tutorial, or the solution for this challenge.

Next, you need to create a class for your new table view cell so you can easily associate the subviews to outlets. To do this, create a new file with the **iOS\Cocoa Touch\Objective-C class** template, name it **ScaryBugCell**, and make it a subclass of **UITableViewCell**.

Then open **ScaryBugCell.h** and replace the contents with the following:

```
#import <UIKit/UIKit.h>

@interface ScaryBugCell : UITableViewCell

@property (nonatomic, weak) IBOutlet UIImageView *bugImageView;
@property (nonatomic, weak) IBOutlet UILabel *bugNameLabel;
@property (nonatomic, weak) IBOutlet UIImageView *howScaryImageView;
```

```
@end
```

Here you have declared properties for the three subviews of your cell. You could have created these by draging from the subviews in the Storyboard editor to this class in the assistant editor, but I wanted to show you that doing it manually like this is also a valid option.

Back in **Main.storyboard**, select your new cell and open the 3rd tab (the Identity Inspector). Set the class of your cell to your new **ScaryBugCell** class.

With the new cell still selected, switch to the 6th tab (the Connections inspector). You will see the new **bugImageView**, **bugNameLabel**, and **howScaryImageView** outlets listed here. Drag from the circles to the right of each outlet to their associated subview in the Storyboard editor to connect them.

Open **BugTableViewController.m** and import your new cell class:

```
#import "ScaryBugCell.h"
```

Next, you need to modify `tableView:cellForRowAtIndexPath:` to use your new cell. Replace the method with the following:

```
- (UITableViewCell *)tableView:(UITableView *)tableView
  cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
  UITableViewCell *cell;

  BugSection *bugSection = self.bugSections[indexPath.section];
  if (indexPath.row >= bugSection.bugs.count && [self isEditing]) {
    cell = [tableView dequeueReusableCellWithIdentifier:@"NewRowCell"
      forIndexPath:indexPath];
    cell.textLabel.text = @"Add Bug";
    cell.detailTextLabel.text = nil;
  } else {
    cell = [tableView dequeueReusableCellWithIdentifier:@"BugCell"
      forIndexPath:indexPath];
    ScaryBugCell *bugCell = (ScaryBugCell *)cell;
    ScaryBug *bug = bugSection.bugs[indexPath.row];
    bugCell.bugImageView.image = bug.image;
    bugCell.bugNameLabel.text = bug.name;
    if (bug.howScary > ScaryFactorAverageScary) {
      bugCell.howScaryImageView.image =
        [UIImage imageNamed:@"shockedface2_full"];
    } else {
      bugCell.howScaryImageView.image =
        [UIImage imageNamed:@"shockedface2_empty"];
```

```
    }
  }

  return cell;
}
```

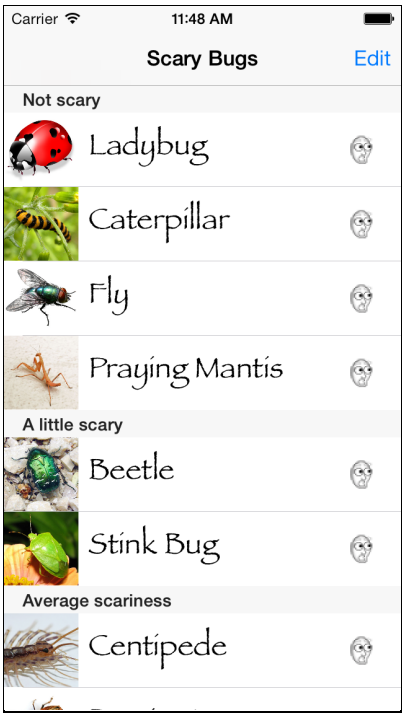
Note how you dequeue two different types of cells here, based on whether it is a row for the scary bug, or for the special "Add Bug" row. You can use this same technique to add as many different types of prototype cells as you would like.

Finally, add this last method:

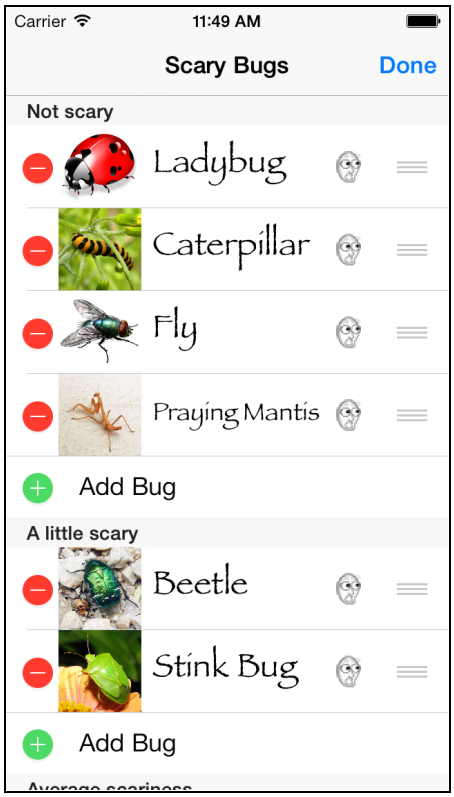
```
- (CGFloat)tableView:(UITableView *)tableView
  heightForRowAtIndexPath:(NSIndexPath *)indexPath {
  BugSection *bugSection = self.bugSections[indexPath.section];
  if (indexPath.row >= bugSection.bugs.count && [self isEditing])
  {
    return 44;
  } else {
    return 60;
  }
}
```

This is required when you have dynamic cells with different heights, so the table view knows the proper height to use for each cell.

Build and run, and enjoy your new look:



If you got the Auto Layout constraints working, it shoud look like this in edit mode:



Congratulations, your bugs are now strutting in style!