Collection Views

Hands-On Challenges

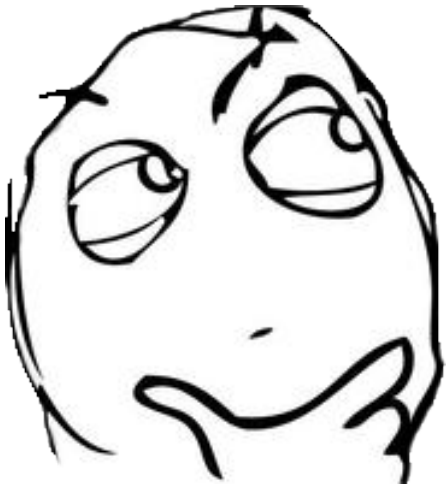
Collection Views
Hands-On Challenges

Challenge A: Tap, Tap, Tap

With your collection view all setup, and hooked up to the papers data source via the appropriate UICollectionViewDataSource methods, it's time to think about cell selection.

The collection view as it stands lacks any sort of user interaction, even though you have everything that you need to make it possible. Ideally, tapping on a cell should push the detail view controller that's already setup in the storyboard onto the navigation stack and display the corresponding wall paper.

But, just like UITableView, there are two different ways you can handle cell selection: using a Selection Segue in Interface Builder, or using the appropriate UICollectionViewDelegate method. So, which one should you choose?



How about both! :]

Your challenge this time is to first implement cell selection using a Selection Segue, and then to re-implement it using the necessary delegate method.

> **Note**: It's important that you implement the two different approaches *in this order*, as you'll be building out the delegate method later in the video series.

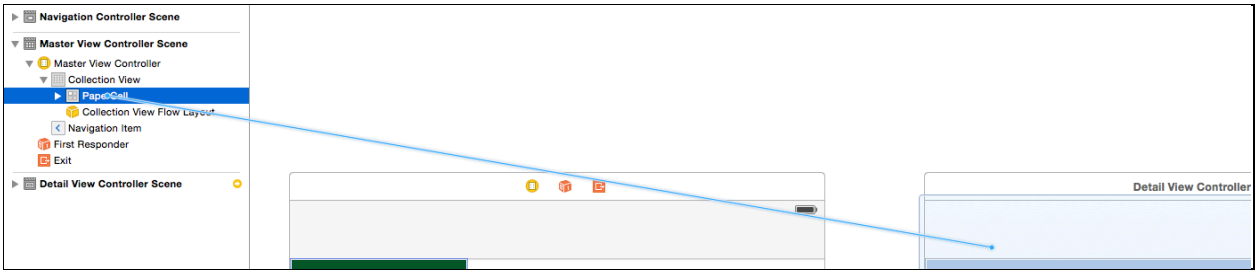
Before you turn the page for our solution, be sure to give it a try for yourself first!

Solution

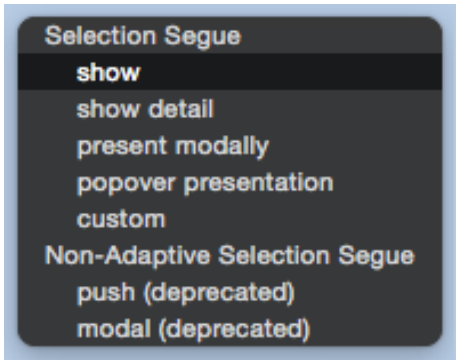
Cell Selection In Interface Builder

Open **Main.storyboard** from the **Layout** group and, using the disclosure triangles, expand Master View Controller Scene in the Document Outline until you find the **PaperCell** object.

Next, ctrl+drag from **PaperCell** to the **Detail View Controller** on the storyboard canvas to create a segue:

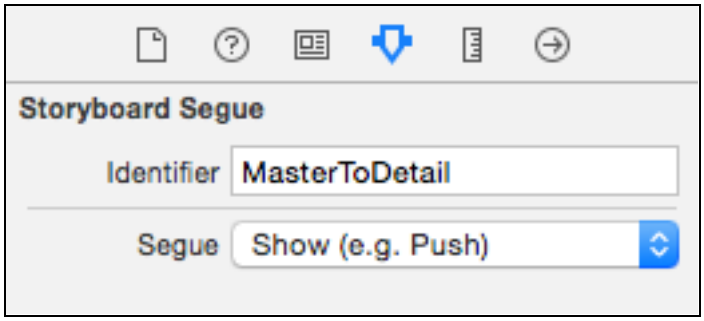


From the resulting popup menu, from the **Selection Segue** section, choose **show**:



The final step to creating a selection segue in Interface Builder is to give the segue an identifier so that it can be referenced from code.

Select the segue you just created, either in the Document Outline or in the storyboard canvas, and then in the Attributes Inspector enter **MasterToDetail** in the **Identifier** text box, like so:



Build and run. Tapping on a cell now pushes the detail view controller onto the navigation stack, but you'll notice the selected wall paper isn't being displayed. You're going to fix that now.

Open **MasterViewController.swift** from the **Controllers** group, and just below `viewDidLoad()`, add the following method:

```
override func prepareForSegue(segue: UIStoryboardSegue, sender:
    AnyObject?) {
  if segue.identifier == "MasterToDetail" {
    if let indexPath = collectionView!
        .indexPathsForSelectedItems().first as? NSIndexPath {
      if let paper = papersDataSource.
          paperForItemAtIndexPath(indexPath) {
        let detailViewController = segue.destinationViewController
            as DetailViewController
        detailViewController.paper = paper
      }
    }
  }
}
```

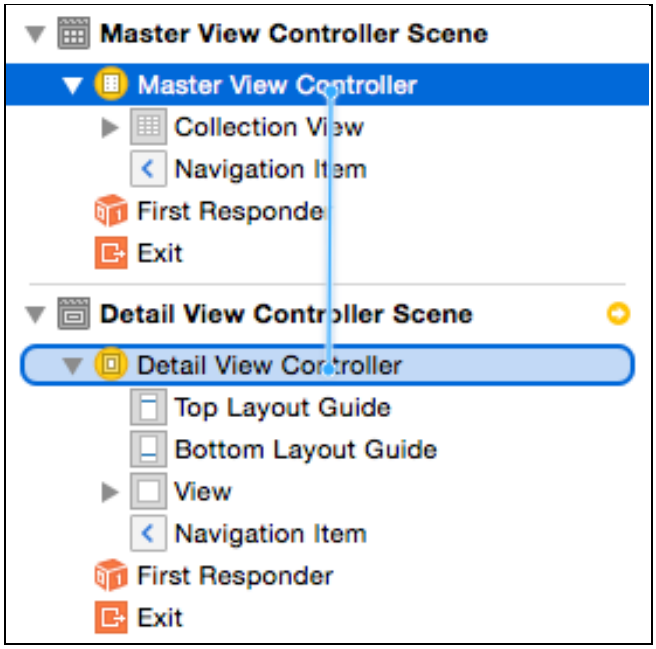
Here you first make sure you're working with the correct segue by checking its identifier, before getting the index path of the selected cell. Since collection views support multiple cell selection, `indexPathsForSelectedItems()` returns an array, so you simply take the first index path. Next, you ask the papers data source for the paper corresponding to that index path, before finally setting it as the paper on the detail view controller.

Build and run. You can now tap any cell and the corresponding wall paper will be displayed by the detail view controller.

In the next section, you'll re-implement cell selection using the appropriate delegate method, instead of a selection segue.

Cell Selection Using The Delegate

Open **Main.storyboard** and delete the segue you created earlier. Next, ctrl+drag from **Master View Controller** to **Detail View Controller** in the Document Outline to create a manual segue:



From the resulting popup menu, from the **Manual Segue** section, choose **show**. Then, select the segue you just created, either in the Document Outline or in the storyboard canvas, and in the Attributes Inspector enter **MasterToDetail** in the **Identifier** text box.

Open **MasterViewController.swift** from the **Controllers** group, and at the bottom of the class add the following:

```
// MARK: UICollectionViewDelegate

override func collectionView(collectionView: UICollectionView,
    didSelectItemAtIndexPath indexPath: NSIndexPath) {
  if let paper = papersDataSource
      .paperForItemAtIndexPath(indexPath) {
    performSegueWithIdentifier("MasterToDetail", sender: paper)
  }
}
```

Here you implement the delegate method `collectionView(_:didSelectItemAtIndexPath:)` which is called by the collection view whenever a user selects a cell. You simply ask the papers data source for the paper corresponding to the given index path, and then trigger the segue manually, passing the paper as the sender. This is a neat little trick if your `prepareForSegue(_:sender:)` implementation needs access to the selected object, like it does here, and saves you have to track that object using a property if it's not used anywhere else.

Finally, replace the exisiting implementation of `prepareForSegue(_:sender:)` with the following:

```
override func prepareForSegue(segue: UIStoryboardSegue, sender:
    AnyObject?) {
  if segue.identifier == "MasterToDetail" {
    let detailViewController = segue.destinationViewController
        as DetailViewController
    detailViewController.paper = sender as? Paper
  }
}
```

Here you've simplified the implementation somewhat by removing the code that asks the collection view for the selected index path, and already have access to selected paper since it's passed from `collectionView(_:didSelectItemAtIndexPath:)`. The final step is to cast the `sender` variable to an instance of `Paper` so it can be set on the detail view controller without throwing a compiler error.

Build and run. Like before, you can tap any cell and the corresponding wall paper will be displayed by the detail view controller.

Using the selection delegate method as opposed to a selection segue offers a lot more control over how selection is handled, as you'll see later in the video series.