

UICollectionView

Introduction

The world is once again craving yet another photo app to help them manage their photos! You're going to build one.

Setup

- Create a new iOS project
- Use the Single View template
- Add 10 images from your phone to your project

Remember to commit your changes to git incrementally and push your local commits to GitHub as often as possible.

User Stories

As a user, I want to view all my photos in a grid

- Display photographs using cells, creating a custom `UICollectionViewCell` class where each cell displays a photo

As a user, I want to have my photos initially ordered into groups by subject

- Group photographs by subject by using sections in the layout, and have a supplementary view that's a label with the title of the section

As a user, I want to organize my photos into groups by location or subject

- Show a segmented control that allows the user to pick from sectioning the photos by group or by location. You may want to embed your view controller in a Navigation Controller so that you can add this segment control to the navigation bar.

Stretch User Stories

As a user, I want to pretty up and enhance the app

- Add a decoration view to the page of floral motifs by subclassing the flow layout
- Add the ability to delete a photo if the user double-taps a photo
- Modify the flow layout subclass to have a custom animation when an item is deleted so that the item appears to fall off the bottom of the screen
- Change the layout object to have custom attributes for each cell that makes it feel more like photos in an album: tilt each photo slightly and shift their positions randomly just a bit, etc

As a user, I want to be able to organize photos into piles and pinch to zoom in and expand a pile

- Add a segmented control to allow swapping between different layouts

- Add pinch gesture recognizer
- Set up an indexPath in the custom layout to track which items should be animated
- Set up multiple properties (e.g., pinch scale, pinch centre) in the custom layout subclass to track the difference and animate expanding accordingly
- If you upscale image in each cell as user pinches out, you may need to swap between the original image and one with larger resolution to prevent pixelate as the cell grows

As a user, I want to be able to drag and drop to rearrange my photos

- Add a long press gesture recognizer
- Tapping on an item should only be recognized if long press gesture fails
- When long press gesture is recognized, scale the selected item up by 20% and fade it out (70% opacity) to give an illusion that the item flows on top of other items (same effect as iOS springboard)
- When long press gesture ends, restore the default attributes
- CADisplayLink can be used to improve FPS and scrolling performance
- You could extend `UICollectionViewDataSource` and implement your own delegate call back that mimics `UITableView`'s
`-tableView:moveRowAtIndexPath:toIndexPath:` and
`-tableView:canMoveRowAtIndexPath:` (e.g.,
`-collectionView:moveItemAtIndexPath:toIndexPath:` and
`-collectionView:canMoveItemAtIndexPath:`)