ScrollView Image Galleries

Learning Objectives

We're going to use UIScrollView to make two different types of image viewers: an image gallery and an image detail view. Then we're going to connect them together, and add a page control.

Note: part 1 and part 2 can be done in either order.

Setup

- Use the Single View template
- You can use any images: Download from the internet.
- **Source Control:** Remember to create and push your project to github.

Instructions

Part 1: Image gallery view

- In the storyboard, add a `UIScrollView` to your root view.
- Set up auto layout constraints, and attach an `IBOutlet` for the scroll view
- Make sure to connect the delegate of your scroll view to your view controller, and declare that you implement `UIScrollViewDelegate` in your header file.

- In your view controller's `-viewDidLoad` method, create three `UIImageView` and place them inside the `UIScrollView` next to each other. Now would be a good time to consider your layout strategy: are you going to use explicit constraints, or set the frame and rely on `translatesAutoresizingMaskIntoConstraints`?
- If you've chosen frames & masks, be sure to set the `contentSize` of your scroll view to include the combined width of your image views.
- If you've chosen autolayout, `contentSize` will be inferred, but be sure to set height and width constraints for your image views or they'll scale to the size of the image they are displaying.
- Enable paging on the scroll view in the storyboard, to have the scroll view lock to "pages" when scrolling.

Source Control: Remember to commit your progress to git

Part 2: Image detail view

- Create a new view controller
- Set it to be the initial view controller in the storyboard (don't delete the view controller from part 1).
- Just like in part 1, create a scroll view, set up constraints, and wire up your delegate and outlet.
- In the storyboard or in code, place a single `UIImageView` inside your scroll view. Set up layout constraints to attach it to the top, bottom, left, and right edges of the scroll view.
- This time *do not* set a height or width constraint for the image view. By default the image view will take its size from the size of the image itself.
- Enable pinch-to-zoom:
    1. set the scroll view's minimum and maximum zoom scale.
    2. implement the `UIScrollViewDelegate` method `viewForZoomingInScrollView:` to tell the scroll view which view to zoom.

Source Control: Remember to commit your progress to git

Part 2.5: Connecting the two views

- Allow users to tap one of the image views from part 1 to present the detail view controller with the appropriate image shown.
    - To do this you will need to use a `UITapGestureRecognizer`
    - You will also need to connect a segue between the gallery and the detail view controllers directly. Set an identifier, so you can manually trigger it when a user taps.
    - In the tap gesture's action, figure out which image view the tap occurred in, and trigger a segue (hint: if you use `performSegueWithIdentifier:sender:`, you can pass user data using the `sender` parameter).
- Consider embedding the initial view controller in a `UINavigationController`, to give your app a back button.

Part 3: Add a page control to part 1

- Add a `UIPageControl` to the root view of the view controller you made in part 1.
- Using the `UIScrollViewDelegate` method `scrollViewDidScroll:`, set the `currentPage` property to the correct page index as the user scrolls.

Stretch Goal

- Make the detail image view zoomed out by default.
- Enable tapping the `UIPageControl` to scroll to the next page.
- Refactor your code so that instead of three image views you have an array of three image name strings. Create and place your image views by iterating over this array. Now add a new image or two.

- To the detail view, add a HUD that shows your position as you zoom and pan around the image (something a bit like this: http://i.imgur.com/gGu6DAy.png)

Resources

Ray Wenderlich Tutorial on ScrollViews: (Start at the section called - "Scrolling and Zooming a Large Image")

http://www.raywenderlich.com/10518/how-to-use-uiscrollview-to-scroll-and-zoom-content