📖 LambdaSchool / **ios-projects**

| Branch: master ▾ | **ios-projects** / Sprint 6 / **Module 3 - UIView Animation** / | Create new file | Upload files | Find file | History |

armadsen Add README for module 6.4 project　　　　　　　Latest commit 5b1c303 on Aug 29, 2018

..

📄 README.md　　　　　　Add README for module 6.4 project　　　　　　4 months ago

📖 **README.md**
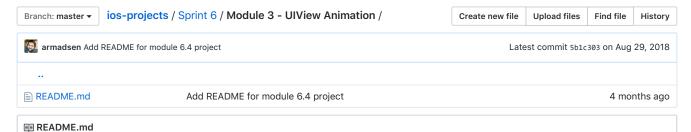
Scatter and Gather Animations

Your project scatters the letters of the word "Lambda" by animating them to new locations. This project helps you practice the concepts learned in Lambda's iOS module 6.4.

After completing the lesson material and this project, you'll be able to animate a number of `UIView` and `CALayer` properties.

Preview your project: https://youtu.be/w8zJVBPrmnY

Set up the project

Follow these steps to set up your project skeleton:

1. Create a new single-view project.
2. Drag a navigation controller into the `Main.storyboard` .
3. Delete the default table view controller.
4. Make your view controller the root view for the navigation controller.
5. Make your navigation controller the entry point for the project by setting it as the initial view controller.

Create a bar button that calls back to your main view controller

These steps enable you to run your animation with a single tap:

1. Create a new bar button item in your view controller's navigation item.
2. Name it `toggle`
3. Connect it to your view controller so it will run `toggle(_:)` as an `@IBAction` .
4. Establish a boolean variable called `shouldScramble` that you toggle back and forth. You'll animate your views to either scramble them or gather them together when the toggle button is tapped.

Create views

Build the items you want to animate, either by hand or in Interface Builder. You'll need six labels (one for each letter) and one image view (for the Lambda logo). Your PM will give you the art for the Lambda logo.

Connect these views to your view controller so they can be animated by your `toggle(_:)` method.

Animate

When scattering:

- Fade out your logo view
- Move your letters to random locations

- Assign them a random background color and text color
- Use a custom transform to rotate the views (`letter.transform = CGAffineTransform(rotationAngle: random_angle)`)
- Incorporate as many other custom animations as you like.
- Your animation should take between 2 and 4 seconds.

When gathering:

- Fade in your logo view
- Reset all the custom properties you previously assigned to the letters.
- Animate the letters back to their starting position, either as a line or however you have them set up by default.

Stretch

Here are some stretch goals to investigate:

- Add 3-D transforms to your layers instead of a simple 2D rotation.
- Animate shadows onto and away from your views.
- Add keyframe sequences to perform multiple tasks when scattering your views.

References

Here are some helpful resources for your project:

- `UIView` reference (Xcode, also https://developer.apple.com/documentation/uikit/uiview). See the `Animations` section for a list of animatable properties.
- `CALayer` animatable properties: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual /CoreAnimation_guide/AnimatableProperties/AnimatableProperties.html
- Core animation: https://developer.apple.com/documentation/quartzcore
- Affine Transforms in iOS: see Xcode docs for `CGAffineTransform` and `CATransform3D` , Quartz 2D: https://developer.apple.com/library/archive/documentation/GraphicsImaging/Conceptual/drawingwithquartz2d /dq_affine/dq_affine.html, Wolfram: http://mathworld.wolfram.com/AffineTransformation.html, Wikipedia: https://en.wikipedia.org/wiki/Affine_transformation, University of Texas: https://www.cs.utexas.edu/users/fussell /courses/cs384g-fall2011/lectures/lecture07-Affine.pdf
- Using `arc4random` and `arc4random_uniform` : https://www.freebsd.org/cgi/man.cgi?query=arc4random, SO: https://stackoverflow.com/questions/32552336/generating-random-numbers-with-swift