IvanCaldwell / **AutoLayout-II**
forked from LambdaSchool/AutoLayout-II

No description, website, or topics provided. Edit

Manage topics

1 commit **1** branch **0** releases **1** contributor

Branch: **master** ▾ | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download** ▾

This branch is even with LambdaSchool:master.

erica first commit

Clone with HTTPS ? Use SSH

Use Git or checkout with SVN using the web URL.

`https://github.com/IvanCaldwell/AutoLayo`

Open in Desktop **Download ZIP**

✓ images	first commit	
✓ photo.imageset	first commit	7 hours ago
✓ photo2.imageset	first commit	7 hours ago
✓ README.md	first commit	7 hours ago

README.md

Module Challenge: Auto Layout II - Adaptive Auto Layout

This challenge allows you to push forward on the concepts and techniques learned in today's guided lesson and apply them in a concrete project. Your lesson explored Auto Layout. You demonstrate continued proficiency by creating an application that showcases the same features you learned in class.

Instructions

Read these instructions carefully. Understand exactly what is expected *before* starting this Challenge.

This is an individual assessment but you are permitted to consult with and support other members of your cohort. You are encouraged to follow the twenty-minute rule and seek support from your PM and Instructor in your cohort help channel on Slack.

Commits

Commit your code regularly and meaningfully. This helps both you (in case you ever need to return to old code for any number of reasons) and your project manager.

Description

In this challenge, you build a series of interfaces using view (xib) files to lay out your content.

Questions

Demonstrate your understanding of today's concepts by answering the following free-form questions. Submit them as a text file along with your project.

- What is auto layout and why do iOS developers use it?
- When do you use compression resistance? When do you use content hugging?
- When do you adjust priorities and why?
- When should you nest stack views? What challenges are involved in nesting stack views?
- How does text autoshrink work and what are its advantages and disadvantages?

Project Note

Today you are not building a full project. You will create several view files (xib files) and then build the contents according to the following challenges.

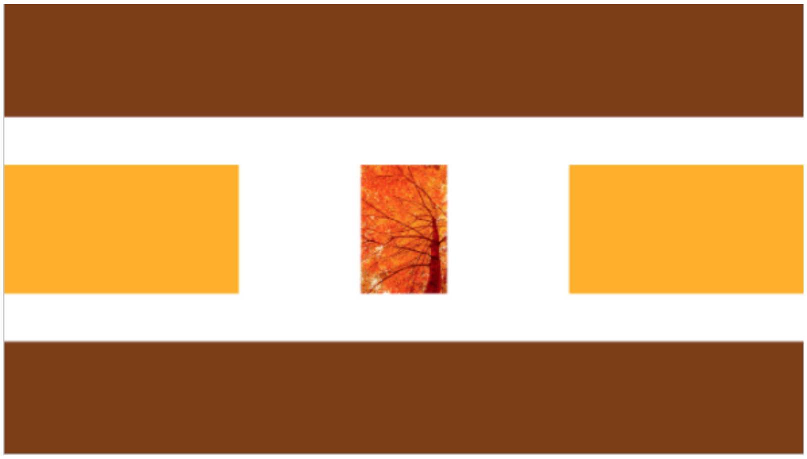
To set up, drag the image sets into your primary asset catalog.

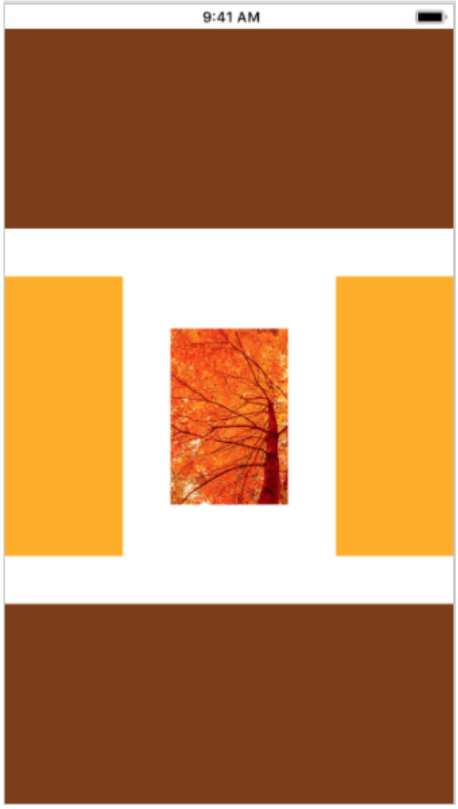
Challenge 1

Duplicate this layout. There are 2 brown views, two orange views, and the image view, which uses the image provided from "photo.imageset". The brown views take up 1/4 each of the safe space. There are 40 pixels between the brown views and the orange views and between the orange views and the image.

The orange and brown views are flush to the sides.

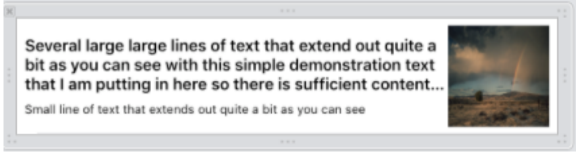
You may not manually resize the image frame. Work with whatever frame you dragged out from the image library.





Challenge 2

This table view cell is 120 pixels high. On the left are two labels with 8 points between them and 8 points to the leading edge. One has three lines of large text. One has one line of small text. The image view on the right is 100x100 and is 20 points away from the two labels and 8 points from its trailing edge.

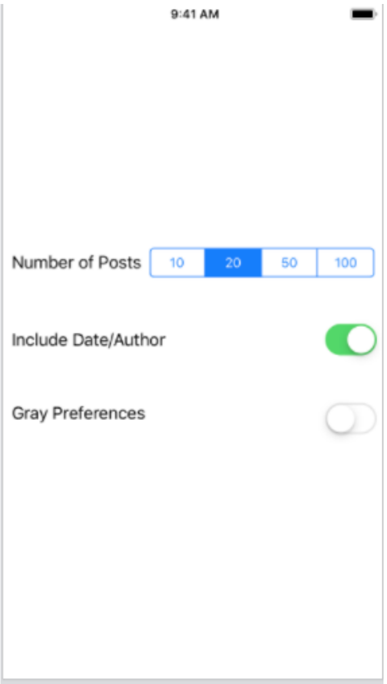


Several large large lines of text that extend out quite a bit as you can see with this simple demonstration text that I am putting in here so there is sufficient content...

Small line of text that extends out quite a bit as you can see

Challenge 3

Create this settings screen. It consists of three options with labels on the left and controls on the right (a segmented control and two switches). Leave 50 points between each line. Controls are right aligned. The labels are left aligned. The set of settings is centered vertically. There are at least 8 points between each label and control. There are at least 8 points of spacing between the view edges and any content.



Number of Posts

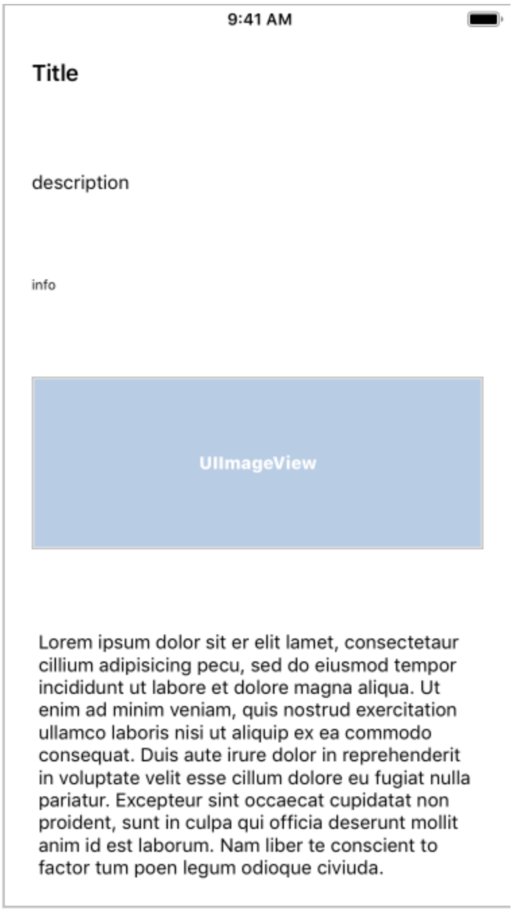
Include Date/Author

Gray Preferences

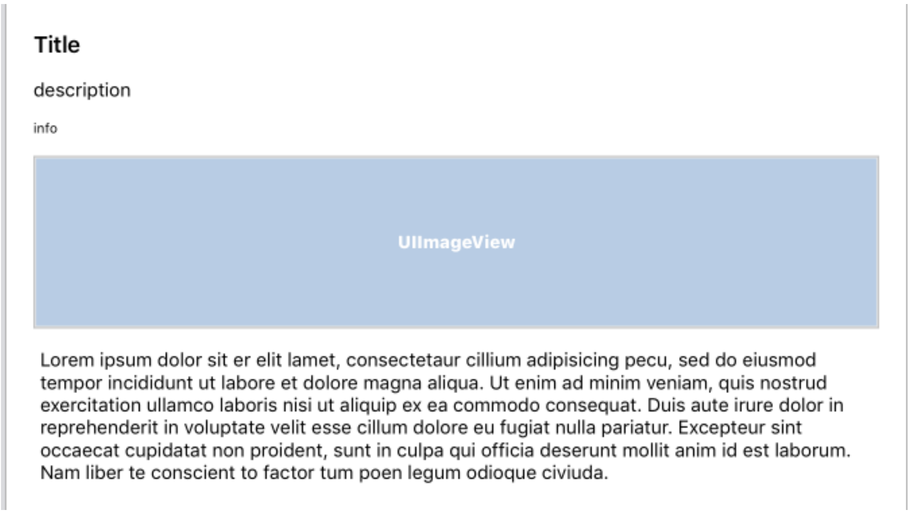
Challenge 4

Create this layout. It has a large title, followed by a normal body-sized description, followed by a small info line. There is next an image view, whose height is fixed to 128 points. It uses aspect fill. The layout finishes with a text view that always occupies 1/3 of the available height.

The layout is padded to each size by 20 points of spacing.



Lorem ipsum dolor sit er elit lamet, consectetaur cillium adipisicing pecu, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Nam liber te conscient to factor tum poen legum odioque civiuda.



Lorem ipsum dolor sit er elit lamet, consectetaur cillium adipisicing pecu, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Nam liber te conscient to factor tum poen legum odioque civiuda.

Minimum Viable Product

Your finished project must include all of the following requirements:

- Your layouts work properly in both portrait and landscape orientations.
- Your layouts work properly on several different target screens, including a wide variety of geometries.

Stretch Problems

After finishing your required elements, push your work further. These goals may or may not be things you learned in this module but they build on the material you just studied. Time allowing, stretch your limits and see if you can deliver on the following optional goals:

- Update Challenge 4 to provide separate layouts for vertical and horizontal presentation.
- Add a "like" button to the cell of Challenge 2, placing it on the left side of your layout.

Progress Dashboard

Download checked items