WordCollage
Lesson 2

Description

Add a new Label and layout constraints to the collage.

Learning Outcomes

- Recognize how adaptive layouts affect the appearance of an app running on different hardware and in different device orientations.
- Practice adding and customizing a user interface component.
- Discover how to add constraints to interface components.
- Experiment with size classes, and auto layout constraints to customize the appearance of a user interface.
- Test the runtime appearance of a user interface with Interface Builder previews and the iOS Simulator.

Vocabulary

Interface Builder	storyboard	canvas
user interface	iOS Simulator	adaptive user interface
Object Library	Label	Attributes Inspector
Document Outline	Assistant Editor	auto layout
constraint	size class	orientation

Materials

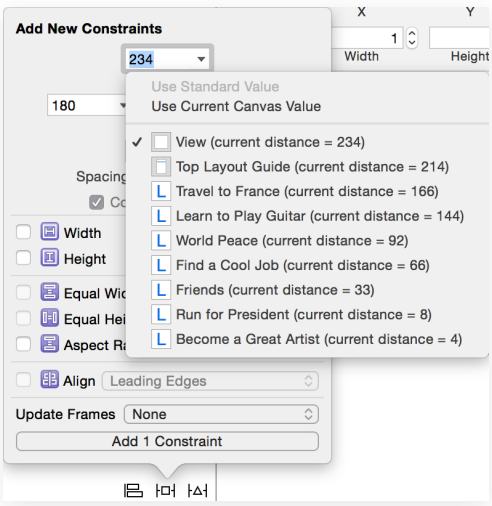
- **WordCollage Lesson 2** Xcode project
- **Adaptive User Interfaces** presentation

Opening

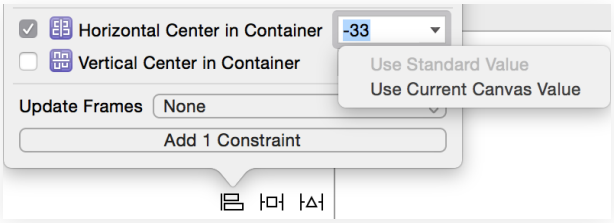
Why does the app look different in the iOS Simulator, compared to what we see in Interface Builder?

Agenda

- Use the Project Navigator (⌘1) to select **Main.storyboard**.
- Run the app (⌘R), and observe how the visual layout of the collage appears different in the iOS Simulator.
- Present the concept of Adaptive User Interfaces.
- Using the Object Library (⌥⌘L), place a new Label on the interface. Change the Label contents (e.g. "Learn to Code") and use the Attributes Inspector (⌥⌘4) to to change the font family, size and color (e.g. 51pt Avenir Next Ultra Light).
- Use the Label handles to expand its size, and adjust the Label position.
- Run the app (⌘R), and observe how the Label position appears differently in the iOS Simulator.
- Explain that position constraints must be added to the Label to influence its position.
- With the Label selected, use the Pin control to select a Vertical Space constraint relative to the View.



- Discuss how Interface Builder displays a vertical blue bar representing the Vertical Space constraint.
- Explain how missing constraints result in Interface Builder displaying Auto Layout issues in orange.
- With the Label selected, use theAlign control to select a Center X Alignment constraint based on the current position of the Label.



- Discuss how Interface Builder displays another vertical blue bar representing the Center X Alignment constraint.
- Using the Show Document Outline control (▣) in the lower left corner of the canvas, ensure that the document outline is visible.
- Discuss how Interface Builder displays one remaining Auto Layout issue in orange, and use the Issue Navigator (⌘4) or the Document Outline disclosure arrow (▸) to observe the details of the remaining Auto Layout issue.
- With the Label selected, use the menu item *Editor* > *Resolve Auto Layout Issues* > *Update Frames* (⌥⌘=) so the frame matches the constraint. Alternatively, use the menu item *Editor* > *Resolve Auto Layout Issues* > *Update Constraints* (⇧⌘=) so the constraints match the frame.
- Run the app (⌘R) and observe how the Label appears in a better position, but still appears somewhat different.
- Using Interface Builder, select the Compact Width | Regular Height size class.
- Explain how different size classes apply to different devices and orientations.
- While viewing the canvas in Interface Builder, open the Assistant Editor (⌥⌘↩), and use the Assistant Editor jump bar to select the Preview item.
- Delete the default iPhone 4-inch preview, and use the Add button in the lower left corner of the Preview to add an iPhone 4.7-inch preview.
- Discuss how the preview closely resembles the app in the iOS Simulator.
- Within the Interface Builder canvas, select the recently added Label, adjust its position, update the constraints (⇧⌘=), and observe how the preview automatically reflects the change.
- Run the app (⌘R) and observe how the Label appears as expected within the iOS Simulator.
- Rotate the app (⌘→) within the iOS Simulator, and observe how the label appears in a different position when in a landscape orientation.

- Using Interface Builder, select the Any Width | Compact Height size class, and rotate the orientation of the preview.
- Select the recently added Label, adjust its position, update the constraints (⇧⌘=), and observe how the preview automatically reflects the change.
- Run the app (⌘R), rotate the app (⌘→) in the Simulator, and observe the Label appearing in the expected position.

Closing

How do the physical screens on different iOS devices vary? How many different kinds of screens are there? Have you seen the principles of adaptive layouts elsewhere?

Modifications And Extensions

- Use additional size classes and constraints, and simulate different devices, to create an interface that adapts to additional screen sizes and orientations.

Resources

iOS Developer Program https://developer.apple.com/programs/ios/

Start Developing iOS Apps Today https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/

iOS Technology Overview https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/

iOS App Programming Guide: About iOS App Programming https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Introduction/Introduction.html

Xcode Overview: Build a User Interface https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode_Overview/edit_user_interface.html

Adding an Object to Your Interface https://developer.apple.com/library/ios/recipes/xcode_help-IB_objects_media/Chapters/AddingObject.html

Adaptive User Interfaces https://developer.apple.com/design/adaptivity/

Designing for Multiple Size Classes https://developer.apple.com/library/ios/recipes/xcode_help-IB_adaptive_sizes/chapters/AboutAdaptiveSizeDesign.html

Auto Layout Guide: Resolving Auto Layout Issues https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/AutolayoutPG/ResolvingIssues/ResolvingIssues.html

Previewing Your Layout https://developer.apple.com/library/ios/recipes/xcode_help-interface_builder/Chapters/PreviewingLayouts.html