2.1 Lecture - Weather App

You will learn how to create the UI (user interface) for a simple weather app. You can use Auto Layout to design the look and feel of your app, and its appearance on different sized iPhone devices.

2.2 Lecture - Design the Weather App UI

The first step is to block out the user interface using the Storyboard file. Add the key components and put them in temporary positions. You'll update these positions using Auto Layout in the next step.

1. Add the weather icons to your asset catalog
2. Set the background color to a bright yellow color: #F0F418
3. Add a city UILabel to the top middle
   1. Text: Your city name (i.e.: "Rochester")
4. Add a date UILabel centered below the city label
   1. Text: Today's date (i.e.: "March 16")
5. Add a white UIView to the center:
   1. Size: 400x400 points
   2. Rename it to: "WeatherView"
6. Inside the white subview add:
   1. Add a UIImageView to hold the current weather icon
      1. Size: 50x50 points
      2. Position: top left corner
   2. Add a weather temperature UILabel in the center
      1. Set the text: "86"
      2. Font: System Bold 80.0
   3. Add a degree UILabel to the top right corner of the temperature label
      1. Text: Use the keyboard shortcut: Alt + 0: "°"
      2. Font: System 40.0
   4. Add a weather summary UILabel centered below the temperature label
      1. Text: "Sunny"

    2. Font: System 17.0

2.3 Lecture - Create Auto Layout Connections

Your UI needs to adapt to different screen sizes. On iPhone you can't design for static pixel positions, instead you will establish rules for where UI needs to move. You can setup relative positions and widths to enable your UI to adapt to wider or taller iPhones.

Auto Layout Part 1

1. Date and City labels
   1. Add constraints to center horizontally and vertically
2. WeatherView
   1. Center vertically and horizontally
   2. Add aspect ratio 1:1
   3. Equal widths to view (containing view)
      1. Constant: 0
      2. Multiplier: 70%
3. Update frames using: `Command + Option + =`

Auto Layout Part 2

1. Weather icon
   1. Add the leading and top space to container constraints
2. Degree label
   1. Add a top layout constraint
   2. Add a horizontal spacing constraint with constant: 0
3. Weather summary
   1. Add a vertical spacing and center horizontally constraint
4. Center the white square inside the container (view)
5. Add a ratio percentage based
6. Update the frames using: `Command + Option + =`

2.4 Bug Fix - Troubleshooting Auto Layout Constraints

If you make a mistake adding constraints you can end of with rules that confuse the iPhone layout. For example, you might tell a view to be 40 points wide and as a separate rule tell it that it must be 80 points wide. When layout constraints can't be followed you have unsatisfied constraints.

Fix Constraints Quickly

1. Use undo: `Command + Z`
2. Use redo: `Command + Shift + Z`
3. Remove all constraints for selected views and start over
4. Start over from the beginning and try again
5. Remove any constraints that are conflicting to resolve layout issues (delete key)
6. Resize labels using `Command + =` before adding constraints
7. Add more constraints to define both size and position

Color Coded Constraints

1. Blue: constraints are good
2. Orange: constraints are missing (ambiguous layout)
3. Red: constraints are conflicting

2.5 Lecture - Size Classes and Auto Layout

Size classes allow you to make bigger UI changes to your application to support portrait and landscape orientations on iPhone. You can also provide a custom user interface for iPhone 6S+ and the iPad. Size classes are also important if your app supports popups or multitasking where your app can appear on the side of another app on iPad.
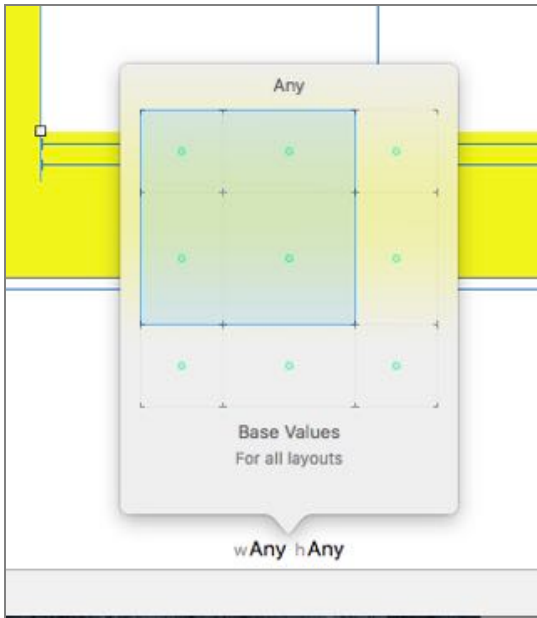
Default Size Class

When you start an iPhone app you will see your storyboard has the Any x Any size class. It is best to design your UI using this size class to understand adaptive layouts.

Four Size Classes

Size classes enable you to create custom, rearranged, or resized interfaces. Switch to one of these to further customize the look and feel. You can adjust font sizes, positions, and sizes of UIView elements.

1. Compact x Compact = iPhone Landscape
2. Compact x Regular = iPhone Portrait
3. Regular x Regular = iPad
4. Regular x Compact = iPhone 6S+

Installing and Uninstalling Constraints

You can further customize each of these layouts by installing or uninstalling constraints (Command + Delete) a constraint or view.

2.6 Lecture - App Icons for App Store and iOS Devices

App Icons should be 1024x1024 big when you create them and you'll need to export smaller versions to include within Xcode. Using Sketch you can make new app icons at the resolution 1024x1024.

Submitting an iPhone App

When you upload your app to Apple for the App Store or Test Flight, you must upload a 1024x1024 JPG or PNG image.

iPhone App Icons

Your iPhone app project has an AppIcon entry in the Asset Catalog that has a list of all the sizes required (180x180 through 29x29). The easiest way to generate these icons is to use the Mac app, IconKit, to export the required files from your 1024x1024 icon.

Adhoc App Icon

If you distribute your app adhoc (not via TestFlight) you will need to add the iTunesArtwork and iTunesArtwork@2x files to your Xcode project (not asset catalog).

Links

- Icon and Image Sizes - apple.com
- App Icons on iPad and iPhone Technical Q&A - apple.com
- Sketch App - sketchapp.com
- IconKit - Mac App Store

2.7 Lecture - Background Images for the Weather App

You can add a background image for your weather app by adding new images and setting up a UIImageView to match the size of the screen with an aspect fill.

If you use images, you would need to have a collection of different images to cycle through based on the current weather conditions. Apple uses animated backgrounds, while other weather apps use images from Flickr.