Designing and Building With Accessibility in Mind.

Hung Truong
@hungtruong
SwiftFest 2018

Hung Truong

iOS Developer

Location support in iOS Simulator

New developer tools

Backlight level setting

Dictionary popover

Customize UI

Newsstand Kit

Improved PDF support

Full page curl transition

Core Image

OpenGL ES debugger

vFcore and vImage libraries

GL Kit

Page View Controller

OpenGL ES game extensions

Fast forward and rewind streaming content

Access to LED flash

Storyboarding

Data protection for Core Data

Forward and reverse geocoding

New view animations

Automatic Reference Counting

GameKit achievement notification banners

AirPlay from AV Foundation apps

Push updates

Ranking-style leaderboards

AirDrop from Activity sheet

Background asset downloads

UI Dynamics

Inter-app audio

Map snapshots

3D map view

MFi game controllers

Dynamic type size

Sprite Kit

Guided Access API

Directions API

Peer-to-peer connectivity

60-fps video capture

compositors

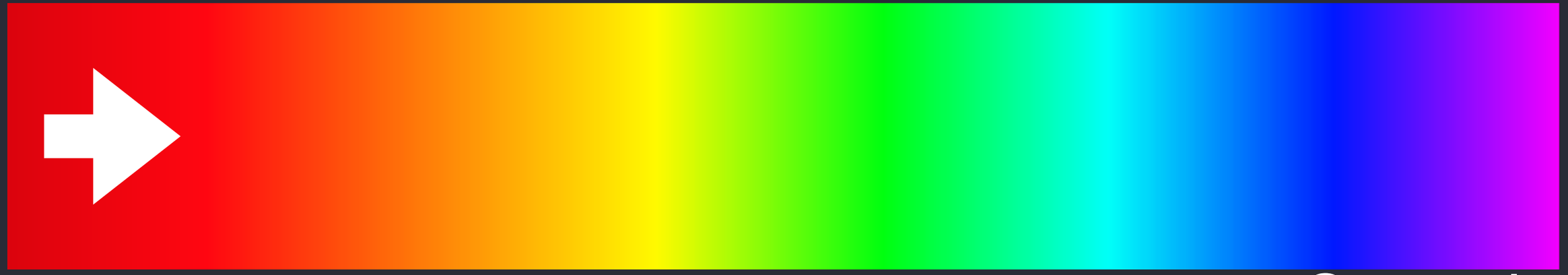
New multitasking APIs

Barcode scanning

Contacts
Interactive notifications
Cache Delete
Smart Card API
Window
SceneKit physically based rendering
Camera sensor data
Live information
Recent addresses
SiriKit
Xcode source editor extensions
iCloud for Developer ID
SiriKit ride booking
Apple Pay on the web
Book-a-ride extension
Live Photos editing
Xcode editor extensions
Message apps
Smaller Xcode download
Native VoIP experience
am alert extension
HomeKit Air Purifier
Xcode thread sanitizer
e Photos capture
iMessage extensions
HomeKit accessories
HomeKit Doorbell
Reservations Maps extension
notifications
VoIP extensions
Wide color
CarPlay Maps instrument cluster
Stickers
Grid View
lKit sharing
RAW photo editing
SiriKit Workouts
Memory debugger
Pixar USD model support
DEV

¯_(ツ)_/¯

Spectrum of Caring About Accessibility



¯_(ツ)_/¯

Care a lot!

What is Accessibility?

- For this talk: designing software so that it is usable for people who have disabilities, focusing on these four categories:
    - Visual
    - Auditory
    - Motor
    - Cognitive

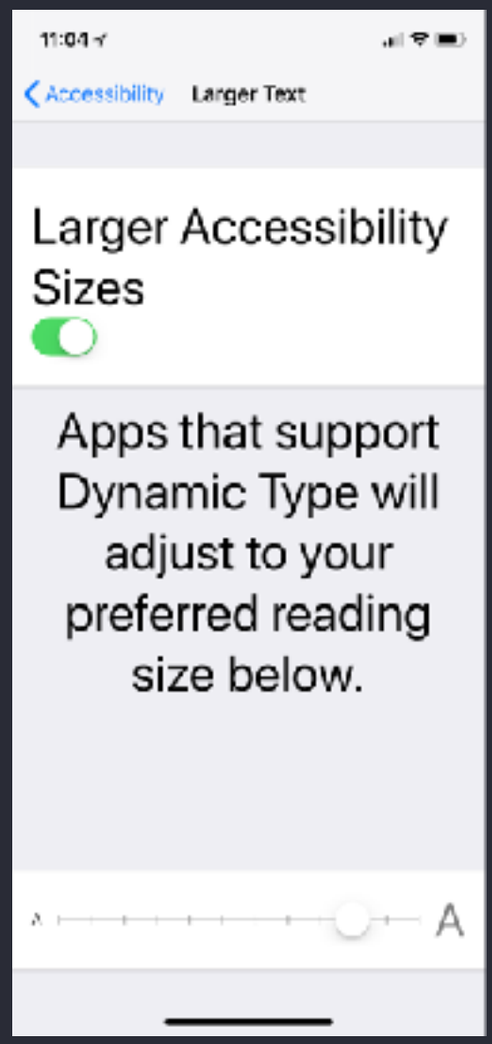
“Temporary” Disability

- One arm in sling
- Dilated pupils at optometrist
- Temporary hearing loss from a concert

Some Things We Won’t Cover

(But are still important)

- Dynamic Text: Useful for users with problems seeing small text



Some Things We Won’t Cover

(But are still important)

- WCAG (Web Content Accessibility Guidelines) 2.0: Follow guidelines for displaying text with enough background contrast *

* *https://www.w3.org/WAI/WCAG20/quickref/#qr-visual-audio-contrast-contrast*

Some Things We Won’t Cover

(But are still important)

- WCAG (Web Content Accessibility Guidelines) 2.0: Follow guidelines for displaying text with enough background contrast *

* *https://www.w3.org/WAI/WCAG20/quickref/#qr-visual-audio-contrast-contrast*

Some Things We Won’t Cover

(But are still important)

- WCAG (Web Content Accessibility Guidelines) 2.0: Follow guidelines for displaying text with enough background contrast *

* *https://www.w3.org/WAI/WCAG20/quickref/#qr-visual-audio-contrast-contrast*

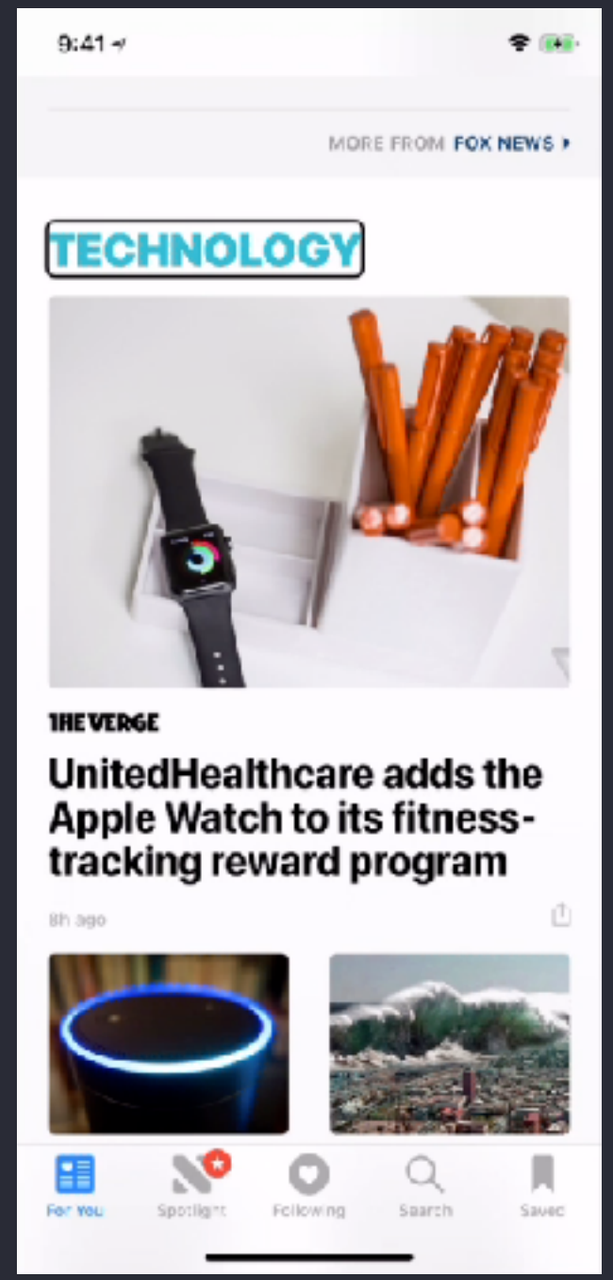
What this talk will go over

- Quick intro to VoiceOver
    - Set up shortcuts so you can audit your app!
- A live coding demo!
    - Increasing usability by supporting VoiceOver features
    - Considerations for custom UI controls and elements
- Lessons learned from supporting accessibility with VoiceOver

What is VoiceOver?

According to Apple:

VoiceOver is a screen reader that interacts with objects in your apps so users can drive the interface even if they can't see it.



Let’s just set up VoiceOver!

Explore VoiceOver: UIKit Playground

- Swipe to move back and forth between elements on the screen
- Touch and drag to explore on the screen
- Swipe up and down to adjust values
- Double tap to activate
- Twist gesture to change mode (speaking rate, custom actions)
- Three finger swipe to scroll through scrollviews

Demo: Pizza Time!

- An app to rate Pizza on a scale of 1 to 5
- Tap on some links to call for pizza or get directions to the pizza!
- That’s it.

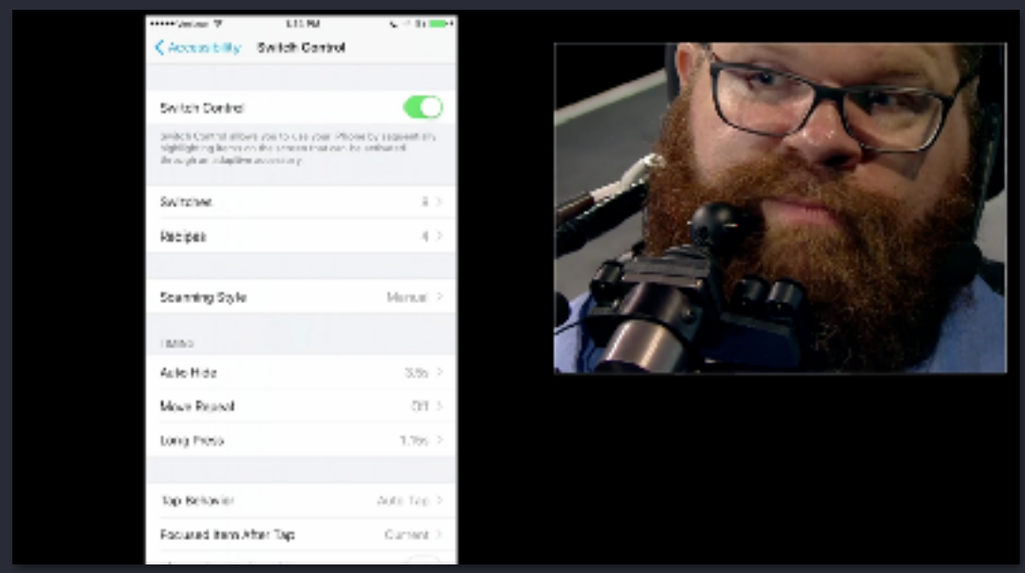
UIAccessibility Overview

```
extension NSObject {
    // Determines "visibility" of view to VoiceOver
    open var isAccessibilityElement: Bool

    // Static label, defaults to text of label or button
    // Remember to set for images!
    open var accessibilityLabel: NSString?

    // Traits for view such as Header, Adjustable, etc.
    open var accessibilityTraits: UIAccessibilityTraits

    // Dynamic value of a view with state, e.g. "4 of 5 stars selected"
    open var accessibilityValue: NSString?

    // A string that is read to provide additional context when needed
    // (Use sparingly)
    open var accessibilityHint: NSString?
}
```

Best Practices! Lessons Learned!

Consider Different Usage Patterns

Users may interact with your app by swiping left and right, panning, or via switch control.

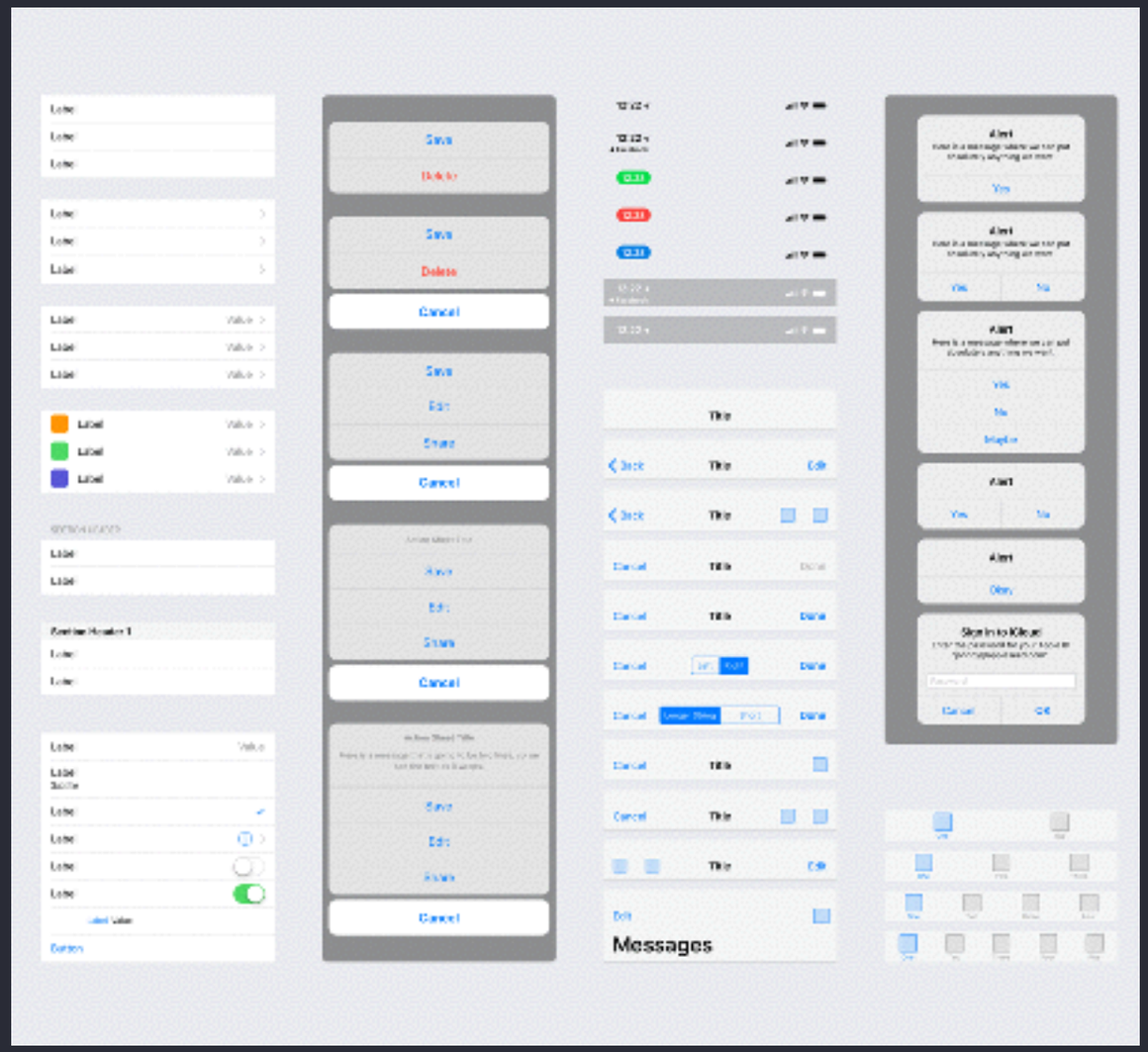


Video: Todd Stabelfeldt using Switch Control

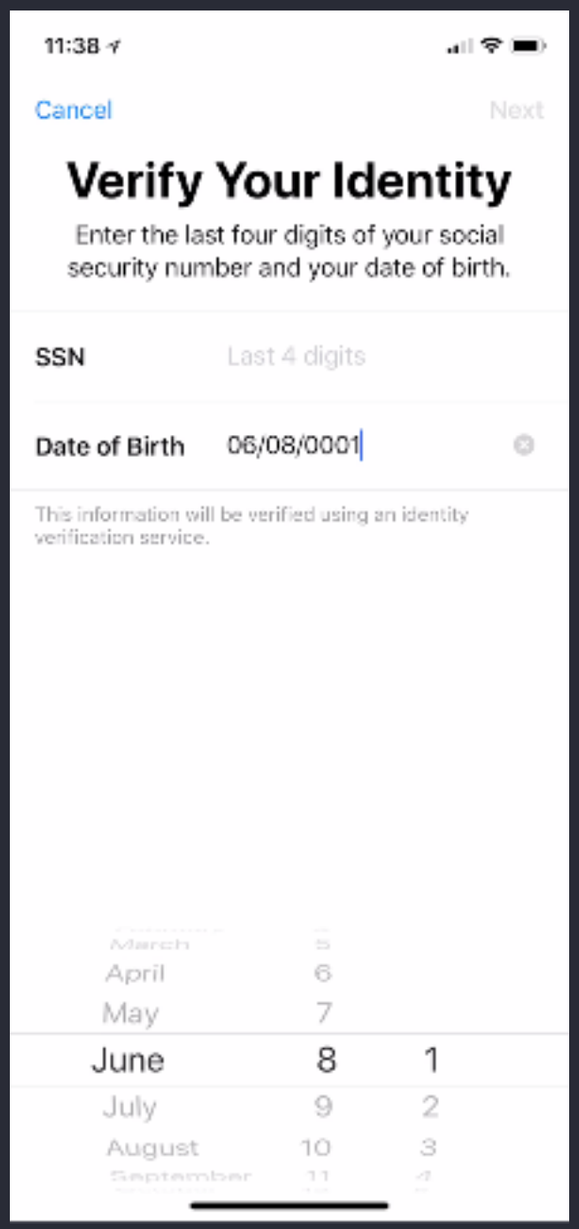
Side note, check out “Convenience for You is Independence for Me” from WWDC 2017

Use Apple UIKit Controls When Possible

- Built-in Accessibility works out of the box (mostly)
- Use UIAppearance protocol to customize your app’s look and feel
- Kinda boring though…



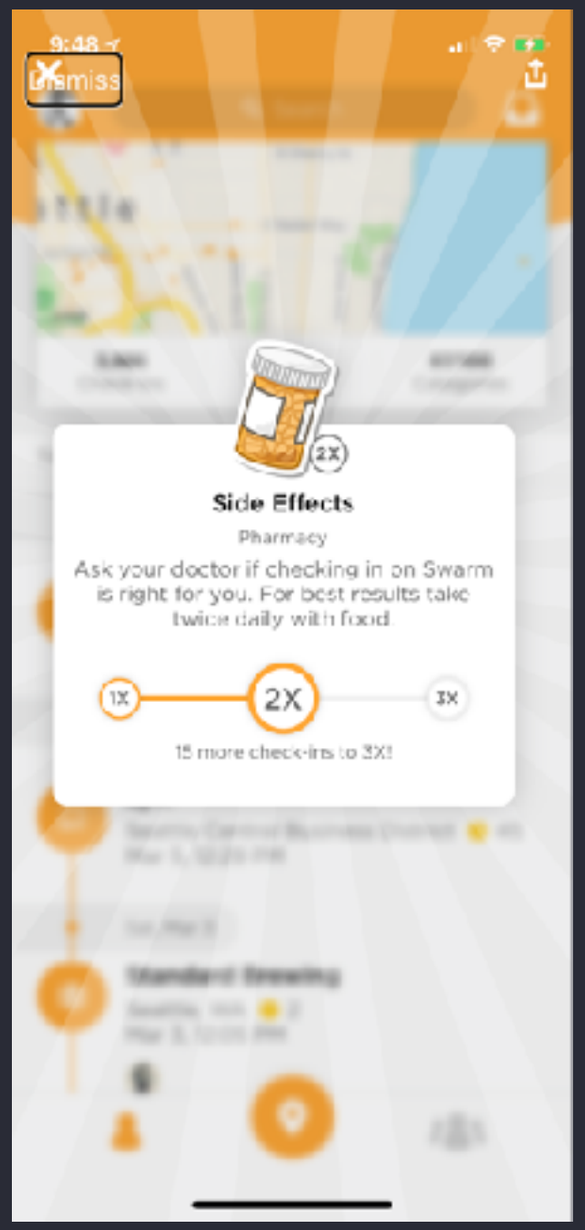
Use Sensible Defaults!



Custom Modals and Alerts are hard!

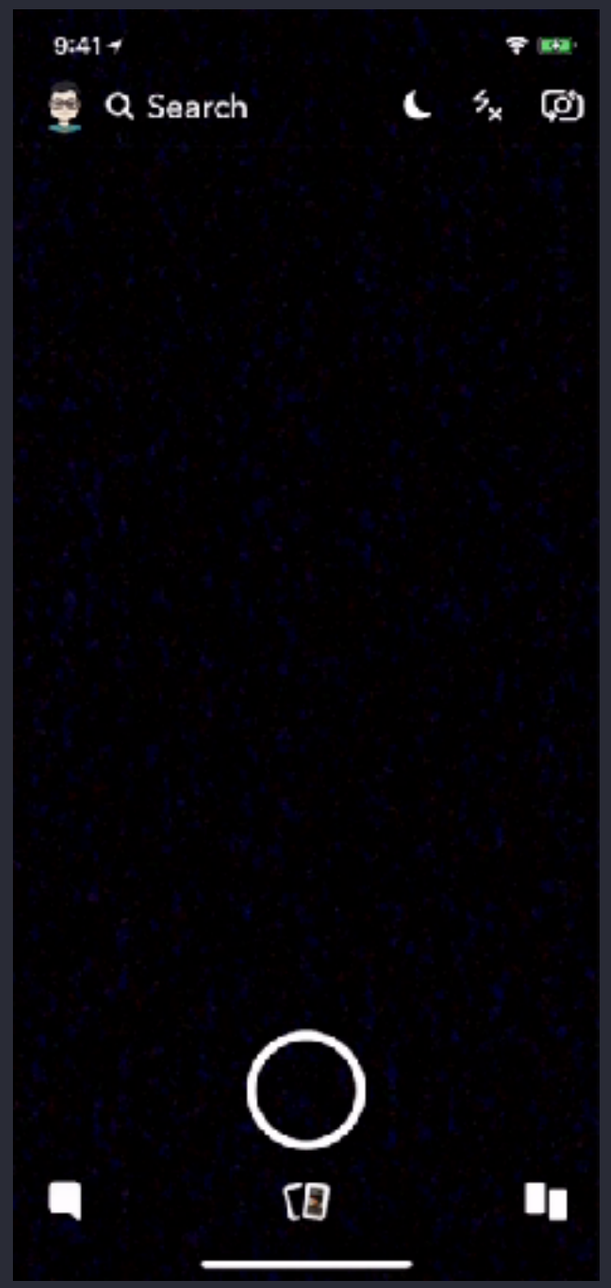
Make sure you can’t get stuck behind it, dismiss it if an action is required, etc.

Use *accessibilityViewIsModal* property to ignore sibling views.



Provide Alternatives to Gesture-based Actions

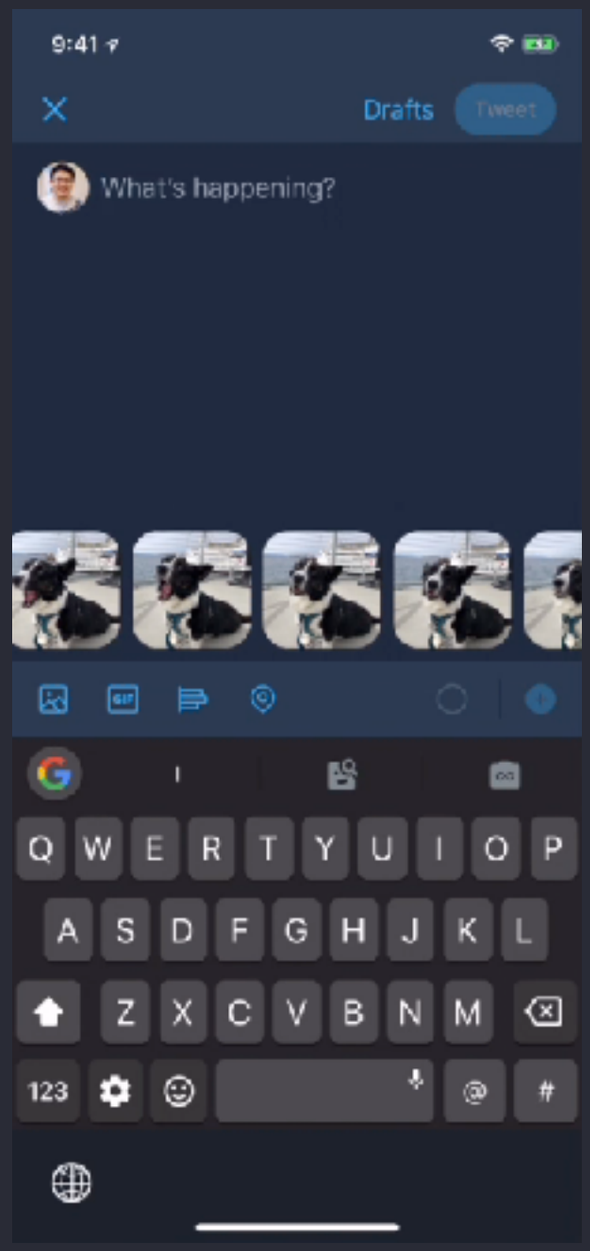
Use *UIAccessibilityCustomAction* to provide VoiceOver users a way to use your feature!



Consider Images

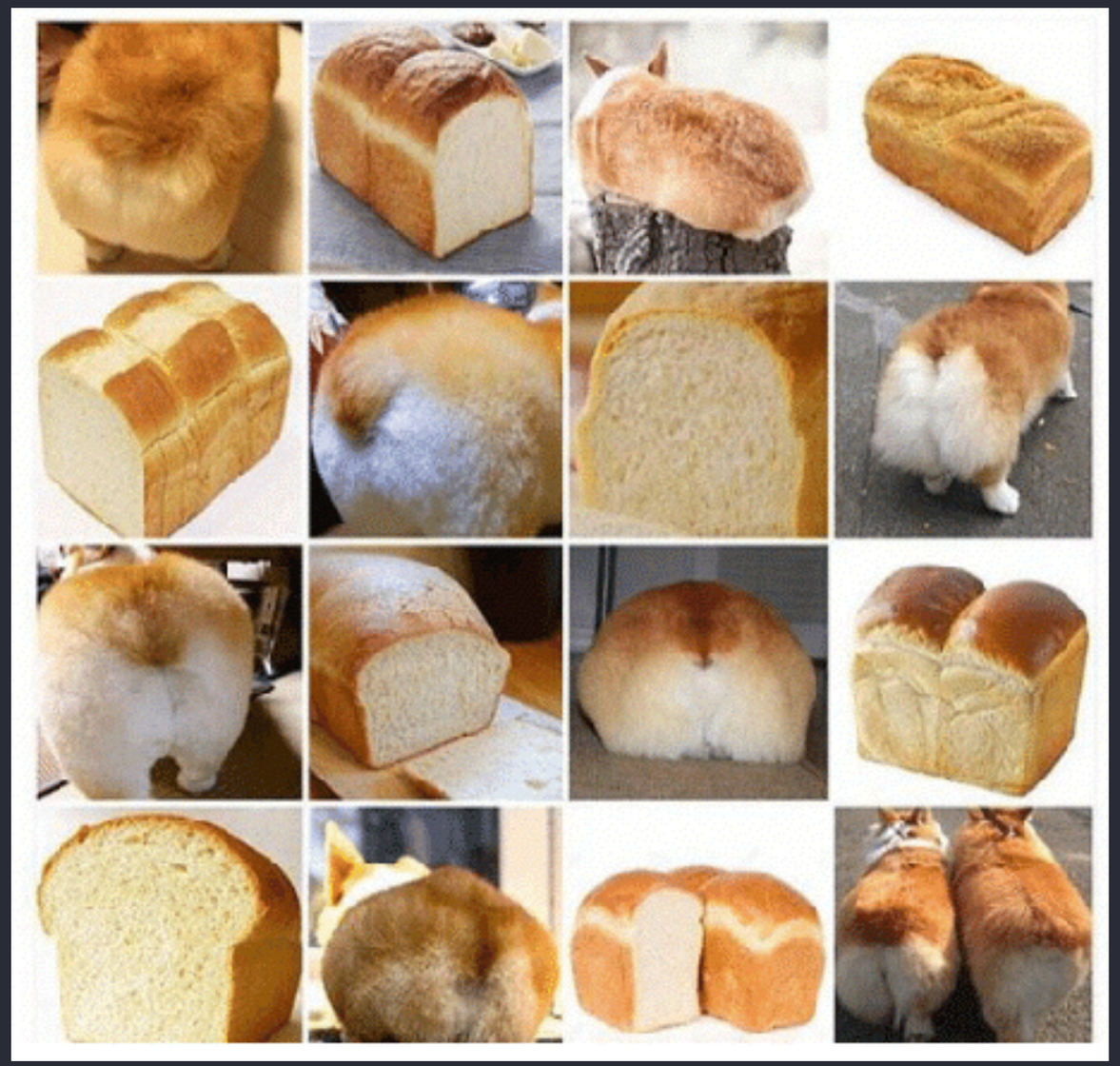
VoiceOver will read your image’s asset name by default.

Ignore the image or provide alternative text if relevant!



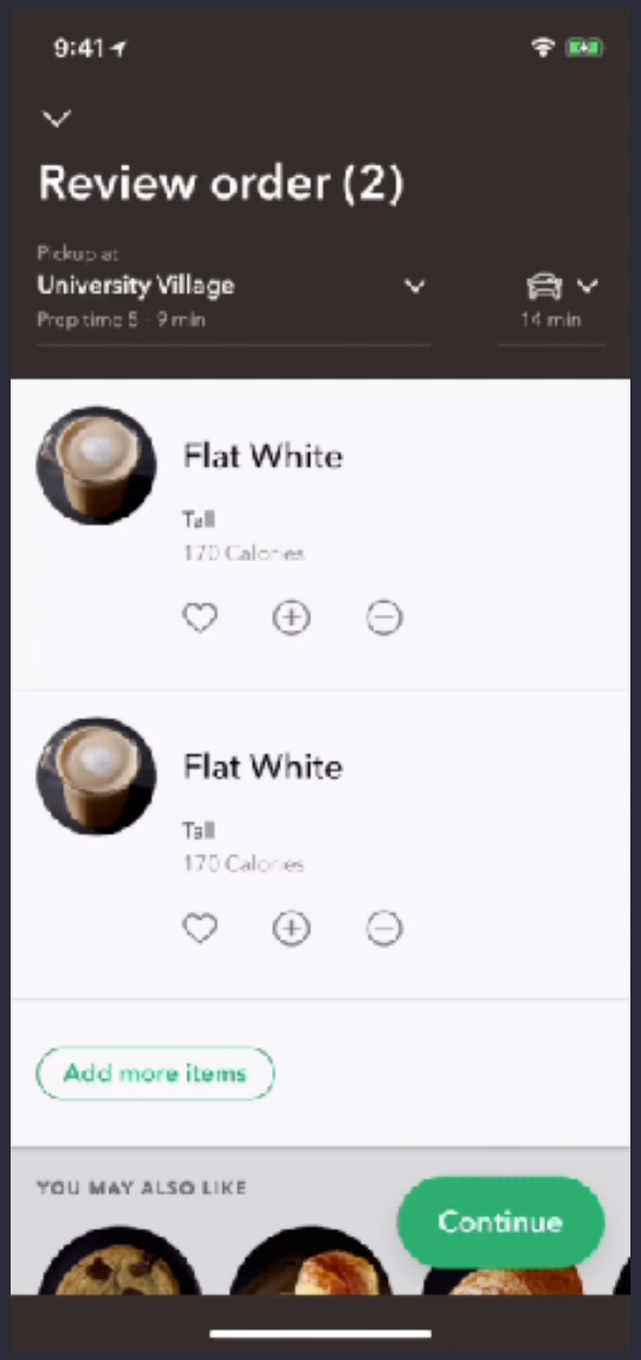
Maybe Use AI!

But don’t get too carried away.



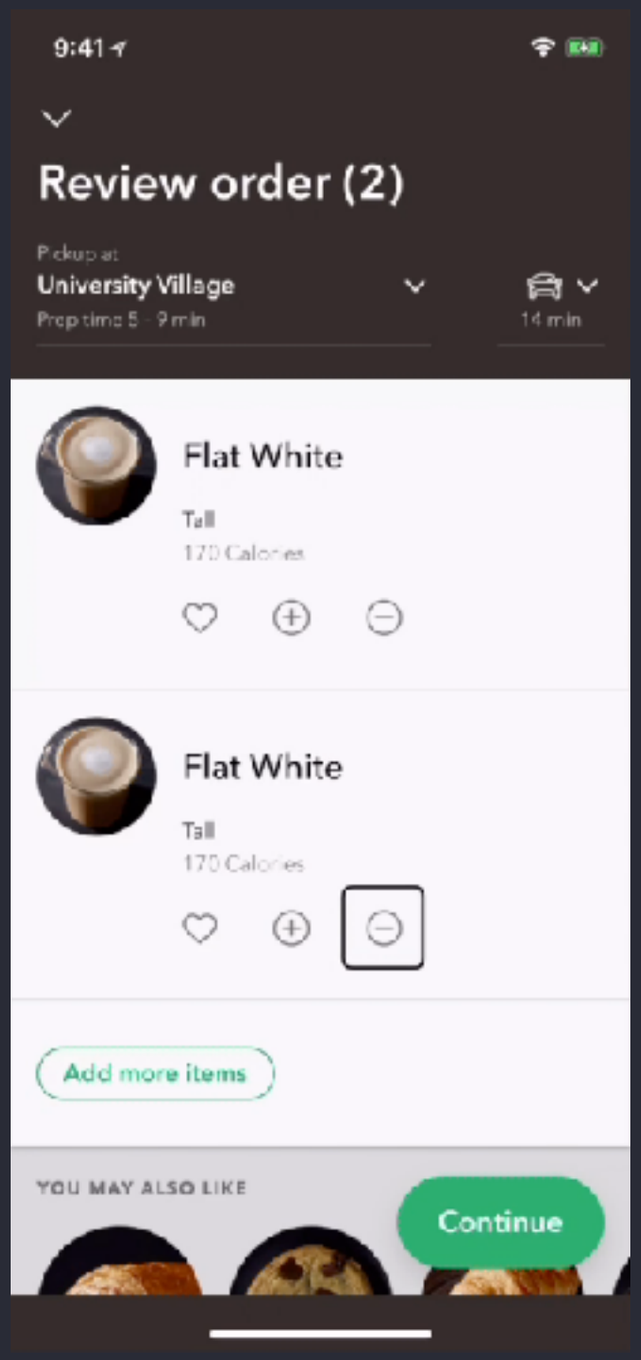
Case Study: Starbucks!

How to support a button in a view that pops up and disappears in 3 seconds?



Case Study: Answer!

Maintain two code paths: enable an “Are you sure?” alert with VoiceOver turned on, and the snack bar when VoiceOver is turned off.



Extra Tips! (Nice to have)

Magic Tap

```
//Two finger double tap that performs a “default” action
func accessibilityPerformMagicTap() -> Bool {
    //Do magic here
    return true
}
```

Escape

```
// 'Z' shaped gesture to dismiss a modal or
cancel an alert
func accessibilityPerformEscape() -> Bool {
    //Do magic here
    return true
}
```

Conclusion

- Supporting accessibility comes down to awareness, best practices and a little extra work
- Apple’s VoiceOver is a powerful tool to increase the usefulness of your app
- If you can’t make everything accessible, make the most important things work, and test against regressions

Next Steps

- Run an accessibility audit on your app. Go through the most common flow and ensure that all users can complete it.
- Assess your company/team's culture and values. Find ammo to support putting resources into accessibility!
- Find other a11y-minded peers in your company and start a group.
- Identify customers who use assistive technologies and gather feedback/partner with them on making your apps more usable.

Further Reading/Viewing + Questions?

- Convenience For You is Independence For Me (WWDC 2017)
- What's New in Accessibility (WWDC 2017)
- Building Apps With Dynamic Type (WWDC 2017)
- WCAG Contrast Standards
- The Accessibility Song (James Dempsey and the Breakpoints)

Follow me on Twitter @hungtruong!