PhoneGap Accessibility
Michael Jordan  |  Accessibility Engineer  |  @majornista
Adobe has a cross-product accessibility team

Supports accessibility in

- Product requirements
- Product development
- Standards committees
- Relationships with assistive technology vendors
- Information for end users and authors
What is PhoneGap?

- PhoneGap makes native mobile apps using HTML, CSS and JavaScript
- Supported platforms: iOS, Android, BlackBerry, Windows Phone 8, Windows 8
- New platforms: Tizen, Firefox OS, Ubuntu Mobile
- Build apps locally with your own SDKs, or in the cloud with PhoneGap Build
- Plugins for device native features (alarms, alerts, camera, LEDs, etc.)
- Over 400,000 developers use PhoneGap
- Tens of thousands of PhoneGap apps are in app stores
What is Cordova?

- Adobe bought Nitobi in October 2011
- Announced the open-sourcing of PhoneGap
- Submitted the PhoneGap codebase to the Apache Foundation as “Cordova”
  - http://cordova.apache.org/
- PhoneGap = Cordova
How it works

- Author builds a site with HTML, CSS, JS, images
- Author adds plugins for non-web functionality
  - Calendar, barcodes, battery, Bluetooth, flashlight
  - Plugins can also be written separately
- PhoneGap bundles the site with the operating system's WebView
- PhoneGap sends the app to the compiler for each operating system
- The compiler generates a native executable
  - .ipa for iOS, .apk for Android, etc.
- The author deploys the app through the OS app store or by sideloading
What works already

- PhoneGap supports (most of) what mobile browsers already do
  - basic HTML accessibility
  - some ARIA support
  - touch navigation
  - screen reader/magnifier support

- General web accessibility rules still apply
Demo 1: Conference app

- Best Practices for Post-Production & Emerging Forms of Audio Description
  - Robert Pearson
  - 8:00AM–8:50AM

- [Case Study] Improving accessibility without compromising user experience
  - Myeon Kim & Sarah Clatterbuck
  - 8:00AM–8:50AM

- PDF/UA: What Is It? Why It Is Relevant?
  - Kushal Agarwal & Shannon Kelly
  - 8:00AM–8:50AM

- Accessibility Features of HTML5
  - Mark Sadecki
  - 8:00AM–8:50AM
Challenges we face with mobile web accessibility

- Screen readers on touch devices intercept gestures
- Subscribing to the OS-level text size preferences is different on every platform
- Sometimes WAI-ARIA isn’t enough
  - Some design patterns, like Sliders, don’t work
  - Live regions have limitations
    - You can't stop them once they've started
    - Sometimes they don't start
Mobile app accessibility vs. web accessibility

- On the web, screen reader sniffing telegraphs disability status
- Some sites have used Assistive Technology data to move users to a “text only” site
- Web sites *do not* get this information

- On mobile, your own device is (mostly) your private agent
  - You can set default font sizes, screen reader & captioning settings, etc.
- All native mobile apps have access to this data
- Mobile apps can adapt more easily to Assistive Technology status
- With PhoneGap, we’re building a mobile app from web technology
  - So...
we made phonegap-mobile-accessibility

- Gives a developer access to the OS APIs mobile apps have, but the web doesn't
  - Assistive technology status
  - Accessibility event models
  - Access to text-to-speech engines
  - Generic and OS-specific functions
Developers need to

- Provide a logical reading order
- Manage application view states
- Add alt text where it belongs
- Using good semantic HTML
  - Including WAI-ARIA, where it’s needed
- Manage focus
- Design with appropriate text contrast
- Use phonegap-mobile-accessibility to check device’s state
New Mobile Accessibility APIs

- `isScreenReaderRunning()`
- `speak()/stop()`
- `postNotification(string, callback)`
- `isClosedCaptioningEnabled()`
- `getTextZoom()`
- `setTextZoom(int textZoom)`
- `updateTextZoom()`
- `usePreferredTextZoom(boolean enabled) (iOS 7+, Android, Windows 8.1, Windows Phone 8.1)`
- `add/removeEventListener() for status changes`
- `isGuidedAccessEnabled() (iOS)`
- `isMonoAudioEnabled() (iOS)`
- `isInvertColorsEnabled() (iOS)`
- `isHighContrastEnabled()/getHighContrastScheme() (Windows 8.1, Windows Phone 8.1)`
MobileAccessibilityNotifications

- SCREEN_READER_STATUS_CHANGED
- CLOSED_CAPTIONING_STATUS_CHANGED
- GUIDED_ACCESS_STATUS_CHANGED (iOS)
- INVERT_COLORS_STATUS_CHANGED (iOS)
- MONO_AUDIO_STATUS_CHANGED (iOS)
- TOUCH_EXPLORATION_STATUS_CHANGED (Android)
- HIGH_CONTRAST_CHANGED (Windows 8.1, Windows Phone 8.1)
Benefits of using web technology to make mobile apps

- One codebase can reach all platforms at once
- Apps are easier to develop and modify than native
- More developers know web technology than any native platform
- …and all the accessibility rules still apply
- The OS-native WebView usually already has accessibility support
Thanks!

- [http://adobe.com/accessibility](http://adobe.com/accessibility)

- [http://cordova.apache.org](http://cordova.apache.org)
- [http://phonegap.com](http://phonegap.com)

- [https://github.com/phonegap/phonegap-mobile-accessibility](https://github.com/phonegap/phonegap-mobile-accessibility)
- [https://github.com/majornista/phonegap-mobile-accessibility-test](https://github.com/majornista/phonegap-mobile-accessibility-test)

- mattmay@adobe.com
- mijordan@adobe.com

- [http://twitter.com/adobeaccess](http://twitter.com/adobeaccess)
- [http://twitter.com/majornista](http://twitter.com/majornista)
- [http://twitter.com/mattmay](http://twitter.com/mattmay)
Demo 2: Test app

PHONEGAP
- DEVICE IS READY
- SCREEN READER IS OFF
- CLOSED CAPTIONING IS ON
- SPEAK SOMETHING
- STOP SPEAKING
- USE PREFERRED TEXT ZOOM

IOS SPECIFIC
- GUIDED ACCESS IS OFF
- INVERT COLORS IS OFF
- AUDIO IS STEREO
- POST ANNOUNCEMENT NOTIFICATION
- POST LAYOUT CHANGED NOTIFICATION
- POST PAGE SCROLLED NOTIFICATION
- POST SCREEN CHANGED
Demo 2: Test app