Google Accessibility Group

Google Access Engineering

Improving Access To Web Platforms, Content, and Applications at Google

Jonas Klink (klink@google.com)
Accessibility Product Manager / Software Engineer

http://google.com/Accessibility

Who am I?

- PM/SWE, part of a Dev team dedicated to Access Engineering
- With Google for the past 4 years
- Education
  - MS in Computer Science (Chalmers Uni. of Tech., Sweden)
  - PhD work in Computer Science (University of Washington, Seattle)
- Research on Education and Technology for the visually impaired
- Project work includes:
  - Client-side: Toolbar, Desktop Search, Chrome
  - Web Apps: Gmail, Apps, Blogger, Maps, Transit, …

What is Google Access Engineering?

Information access is at the core of Google's mission

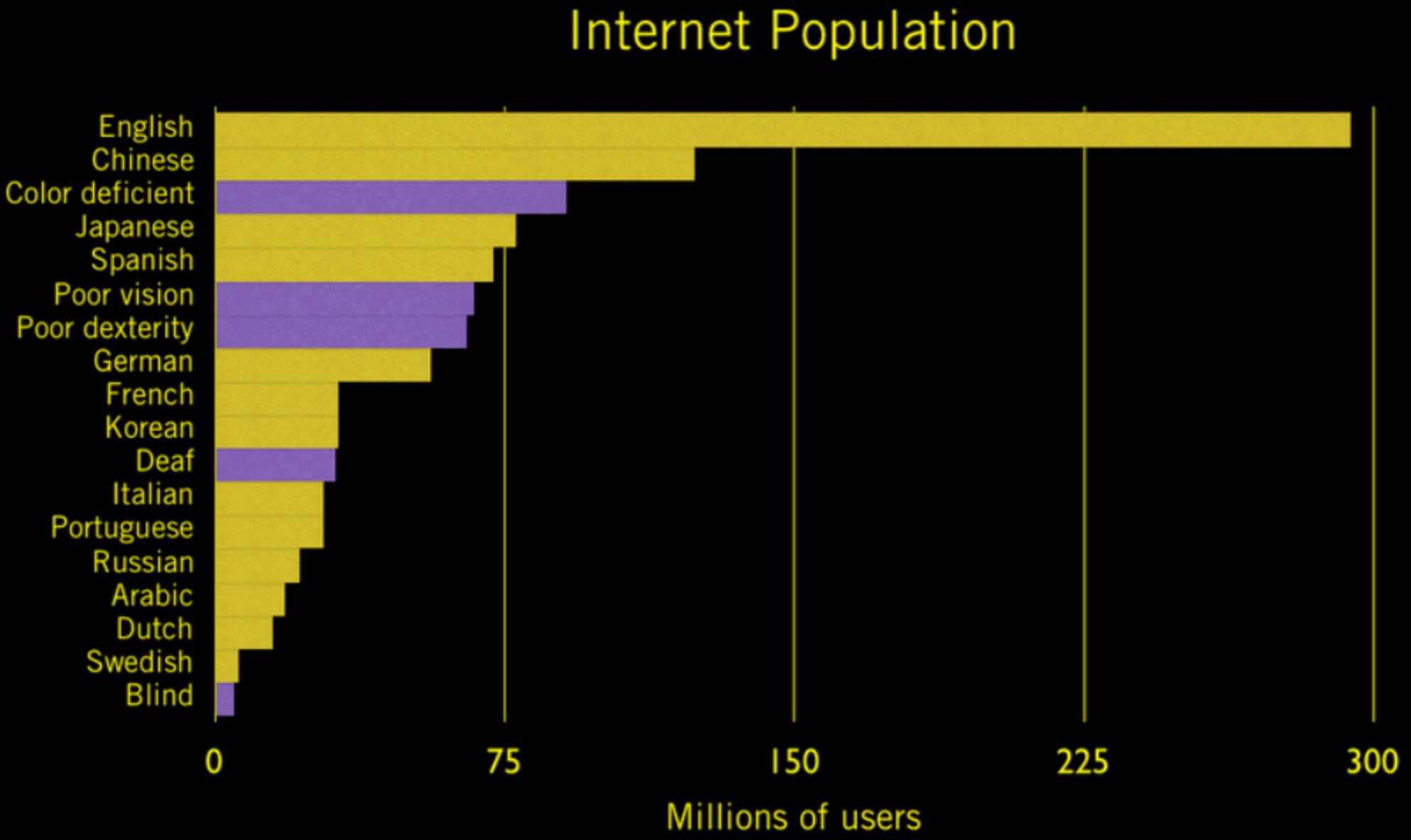
- To make the world's information *universally accessible and useful*
- Adapting to the information channels that your user can *most effectively use*

Access involves understanding two key issues:

- How differently abled users and/or different setups *access information*
- How developers *enable apps* to function with *assistive technologies and settings*

Visit google.com/accessibility for resources and feedback!

Why? Users!



http://google.com/Accessibility

Outline

The Web as a Platform

Mobile: Android

Accessibility in the Cloud

Q & A

The Web as a Platform

http://google.com/Accessibility

FAIL!

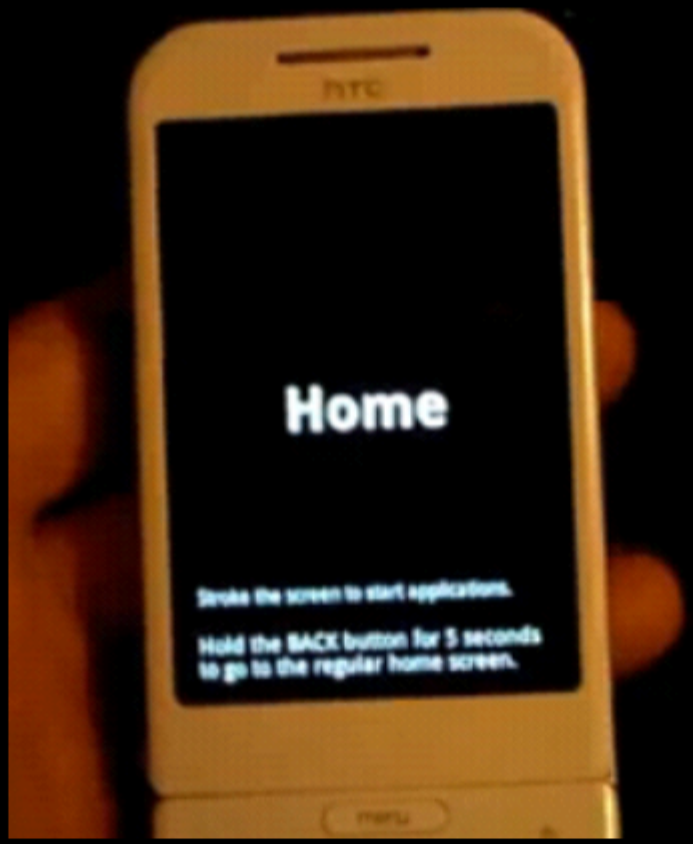
http://google.com/Accessibility

The Web as a Platform

- Platform layers are changing
  1. Low-level support framework (TTS, fonts, themes)
  2. JavaScript APIs
  3. Web Applications (GWS, Gmail, Docs)
- Graceful Degradation vs. Progressive Enhancement
- The Web has the distributed data
  - Universal Access Engineering makes it available through any channel
- Personalization and user goals are key
  - Every level in the stack is customizable
  - APIs provides the muscle
  - Focus on workflows, rather than UI components
  - User is less dependent on the applications

Example: Google Reader Access

- Extremely keyboard friendly
  - Access keyboard shortcut through '?' or Reader Help Center
  - Navigate items with 'j' and 'k'
  - Keyboard bindings available for starring, sharing, commenting, etc
- Delivers screen reader augmentation
  - Follow link 'click here for ARIA enhanced Google Reader'
  - Screen reader support in ARIA-enabled browsers
- Applies magnification lens for low-vision users
  - Follows keyboard navigation
  - Provides customization through '-' and '='
- Zero impact on latency!



Mobile: Android

Eyes-Free Navigation on Android

- Built on Open Source TTS library (in production)
- Alternative eyes-free Home screen
  - Configurable to replace default Home screen
  - Home is where finger first touches
  - Configurable shortcuts laid out in square pattern around Home
  - Includes quick access to Signal, Time, Battery, Location, Search, Applications, Voicemail & Shortcuts
  - Feedback through haptic and speech channels

Video: Eyes-free Kit on Android System

Eyes-Free: Talking Dialer on Android

- Built on Open Source TTS library (in production)
- Eyes-free dialing on touch screen possible
  - Keypad number 5 always located at center of square
  - Numbers 1-9 laid out around number 5 (0 located past 8)
  - Dialing made possible by gesturing in desired direction, while maintaining touch
  - Feedback through haptic and speech channels

Video: Talking Dialer: Hands-Free Communication Device

Non-visual feedback through TalkBack

- Available under Settings->Accessibility
- TalkBack
  - Relying on API available in Android 1.6+ SDK (Donut)
  - Standard UI elements produce spoken feedback during interaction
  - Free voices available for en, fr, it, de & es
- KickBack
  - Producing haptic feedback
- SoundBack
  - Producing non-spoken auditory feedback
- Instructions available on YT EyesFree channel

DIY: Accessibility APIs on Android

- Developers: customize your Access experience!
- Open Source development APIs
  - android.accessibilityservice.AccessibilityService
    - Accessibility service runs in the background and receives callbacks by the system when AccessibilityEvents are fired.
  - android.accessibilityservice.AccessibilityServiceInfo
    - Describes an AccessibilityService, e.g. what feedback type is generated: audible (but not spoken), spoken, haptic or visual.
  - android.view.accessibility.AccessibilityEvent
    - Represents accessibility events that are sent by the system when something notable happens in the UI.

Accessibility in the Cloud

http://google.com/Accessibility

Chromium Access

- Based on Open Source WebKit
  - Open development and standards compliance
  - Multi-process and sandboxed
  - Minimal UI - browser fading into the background
- Access work in progress
  - Keyboard navigation
  - Developing screen reader support & WAI-ARIA
  - Full page zoom
  - High Contrast Support (through Extensions)

Extended Access through Extensions

- URL: chrome.google.com/extensions
- Light-weight and easy to author (HTML/CSS/JS)
  - Anyone can create!
- Open Source and extensible
  - Anyone can make it his/her own!
- Secure (multi-process), fast (v8) and powerful
  - We've got your back!
- Overlay putting desired information and behavior at your fingertips
  - Information the way YOU want it!
  - Quick navigation, page re-styling & re-structuring, etc

AxsJAX

- Keyboard navigation in <30 mins!
  - Open Source
  - Can be applied to any page
  - Overlay can provide additional keyboard navigation
  - Any DOM node can be spoken
  - Design for overlays - content in DOM access-friendly
- Lens effect
  - DOM-node level magnification
- Available for many Google products
  - GWS, Gmail, Calendar
  - Reader, Scholar, Books

URL: http://code.google.com/p/google-axsjax/wiki/Showcase

Captions

- Talk:Video Captioning at Google and YouTube
  - Friday, Mar 26th, 4:20pm in Del Mar AB
- Largest online repository of captioned video
- Auto-captions & auto-timing
- Benefits all users!
  - Improved indexing and in-video navigation
  - Access for non-native speakers
  - Multiple tracks & languages
  - Real-time translation

Conclusion

Collaboration and openness benefit everyone

Customization is key
Configure once, work everywhere

Focus on workflows rather than widgets

Develop solutions with little or no latency impact

Thank you for coming!

?

Q & A

google.com/accessibility