

# Make Your Apps Accessible to All

Heath Borders

Jive Software Mobile Technical Lead

@heathborders

# Two talks in one!

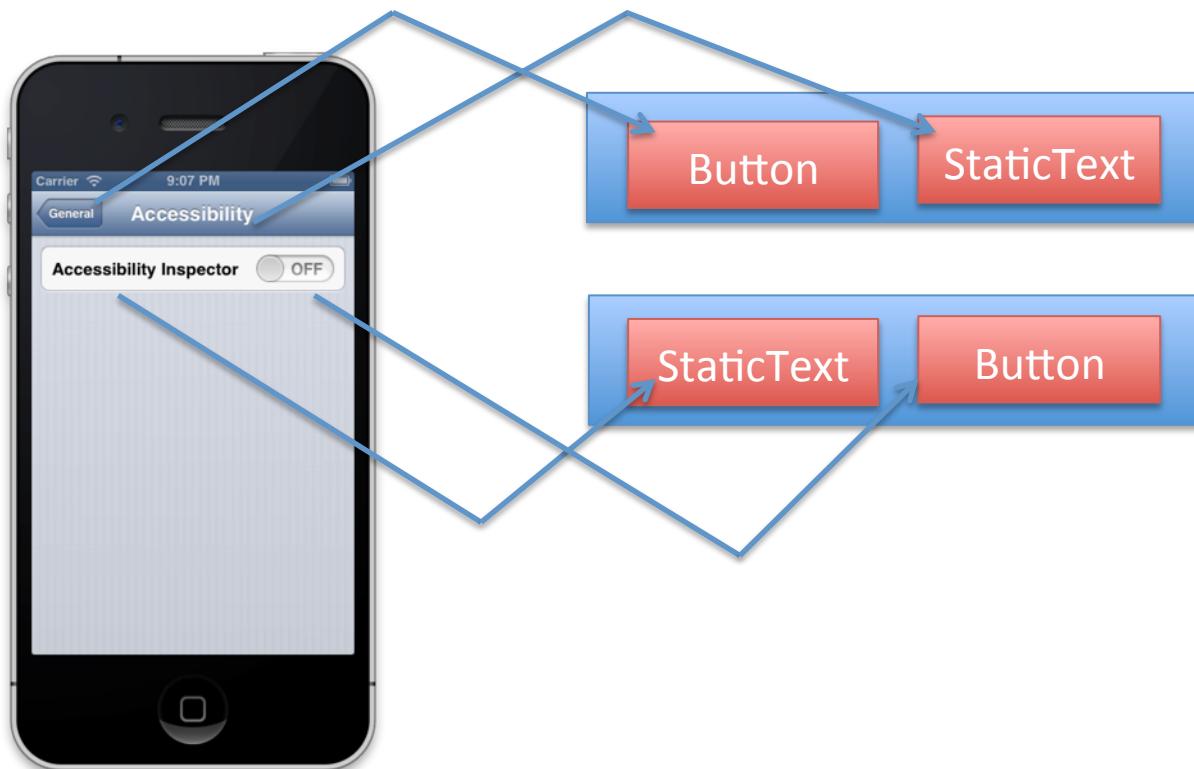


Android

iPhone

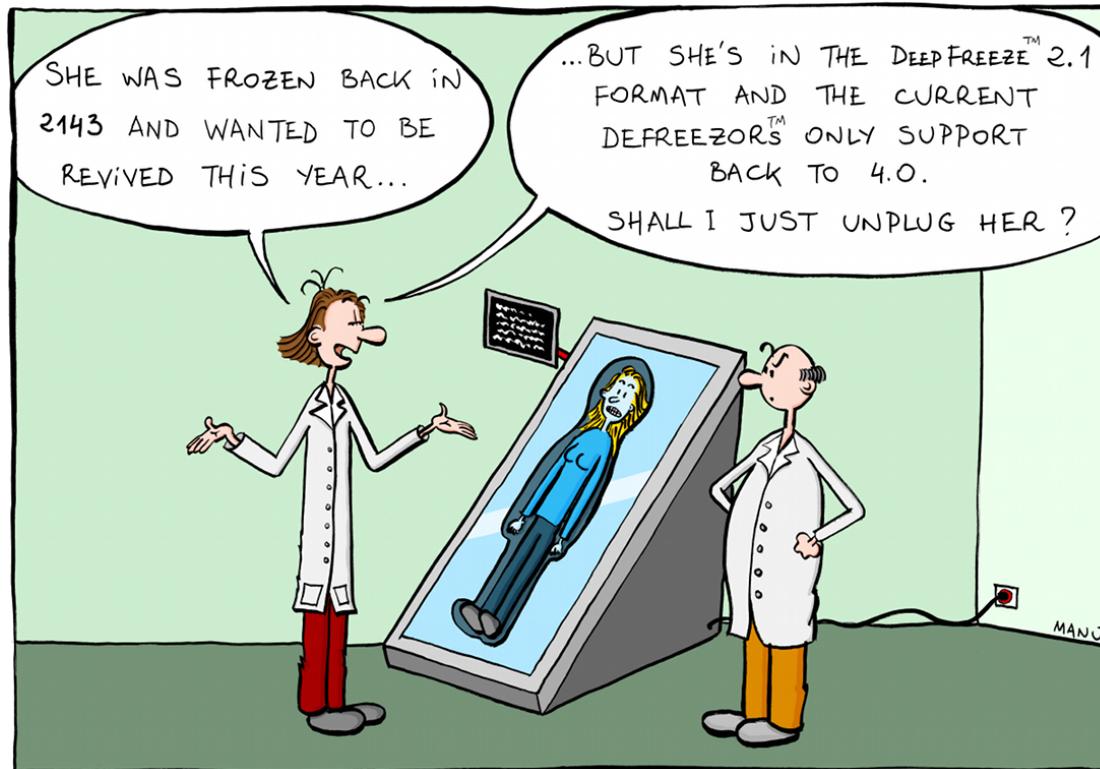
# UIAccessibility

- Informal Protocol
- Separate DOM hierarchy for screen readers



# Compatibility

- Available since iOS3
- New APIs with every major iOS version since



# UIAccessibilityTraits

- The capabilities of your accessibility element
- Examples
  - UIAccessibilityTraitButton
  - UIAccessibilityTraitStaticText
  - Lots more

# Sensible Defaults

```
UILabel *label = [UILabel new];
```

```
label.text = @”foo”;
```

```
label.isAccessibilityElement = YES;
```

StaticText:“Foo”

```
UIImage *image = [UIImage imageNamed:@”foo  
bar.png”];
```

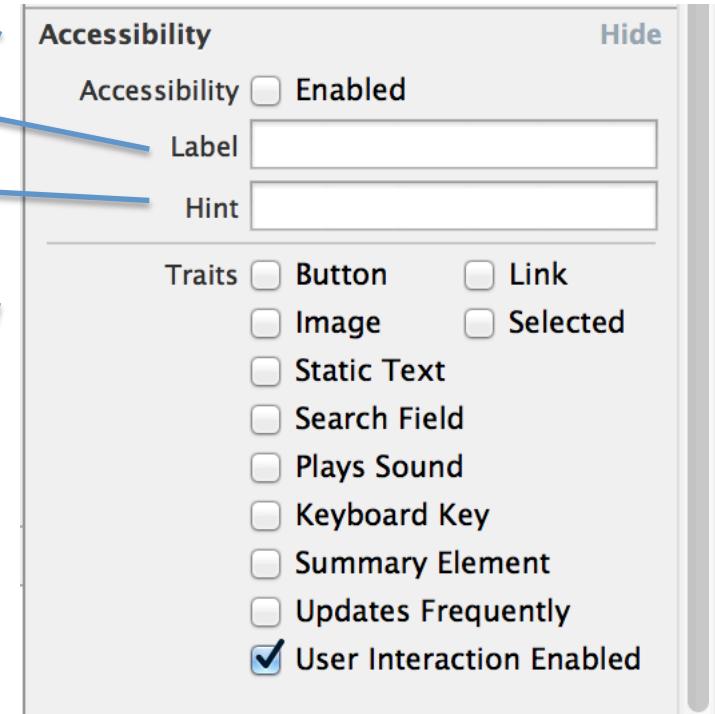
```
UIImageView *iv = [[UIImageView alloc]  
initWithImage:image];
```

```
iv.isAccessibilityElement = YES;
```

Image: “foo bar”

# Normal Usage

- UIAccessibility informal protocol
  - `isAccessibilityElement`
  - `accessibilityLabel`
  - `accessibilityHint`
  - `accessibilityValue`
- Attributes → 



# Example

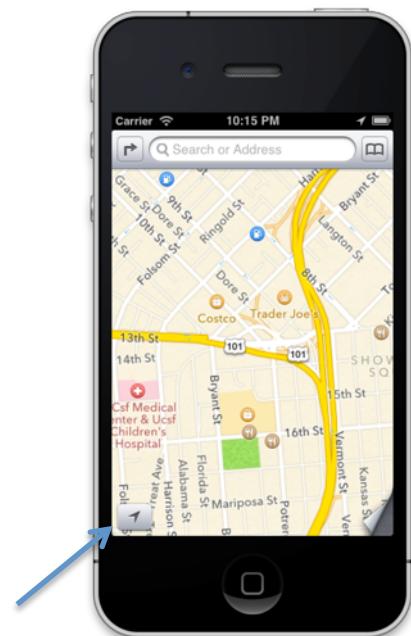
- Maps Location Tracking Button
  - Label: Tracking
  - Hint: Activate to change tracking options
  - Value: Off



- Value: On



- Value: On with heading

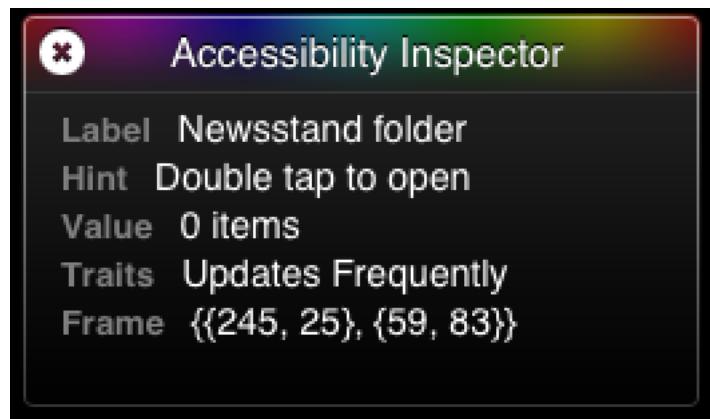


# Testing VoiceOver

- VoiceOver navigation only available on device
  - Settings->General->Accessibility->VoiceOver
  - Set up accessibility shortcut (Triple-click Home)
  - VoiceOver practice to discover gestures
    - Triple-tap with 3 fingers for the screen curtain
  - <https://developer.apple.com/library/ios/technotes/TestingAccessibilityOfiOSApps/TestAccessibilityonYourDevicewithVoiceOver/TestAccessibilityonYourDevicewithVoiceOver.html>

# Testing VoiceOver

- iOS Simulator has Accessibility Inspector
  - Settings->General->Accessibility
  - Only for verifying simple accessibility values
  - Can't test navigation, can't screen-read



# UIWebView

- VoiceOver finds web-accessible elements automatically
- Web Accessibility is beyond the scope of this talk
  - <http://www.w3.org/WAI/>

# Simple UITableViews

- Don't change `isAccessibilityElement` explicitly on any `UILabels`.
  - Cell navigation won't work
- `accessibilityLabel` built from child `UILabels`
  - Can override by setting directly

# Simple UICollectionViews

- Cell navigation doesn't work out of the box
  - Call `-[UICollectionViewCell setAccessibilityElement:YES]` to enable cell navigation

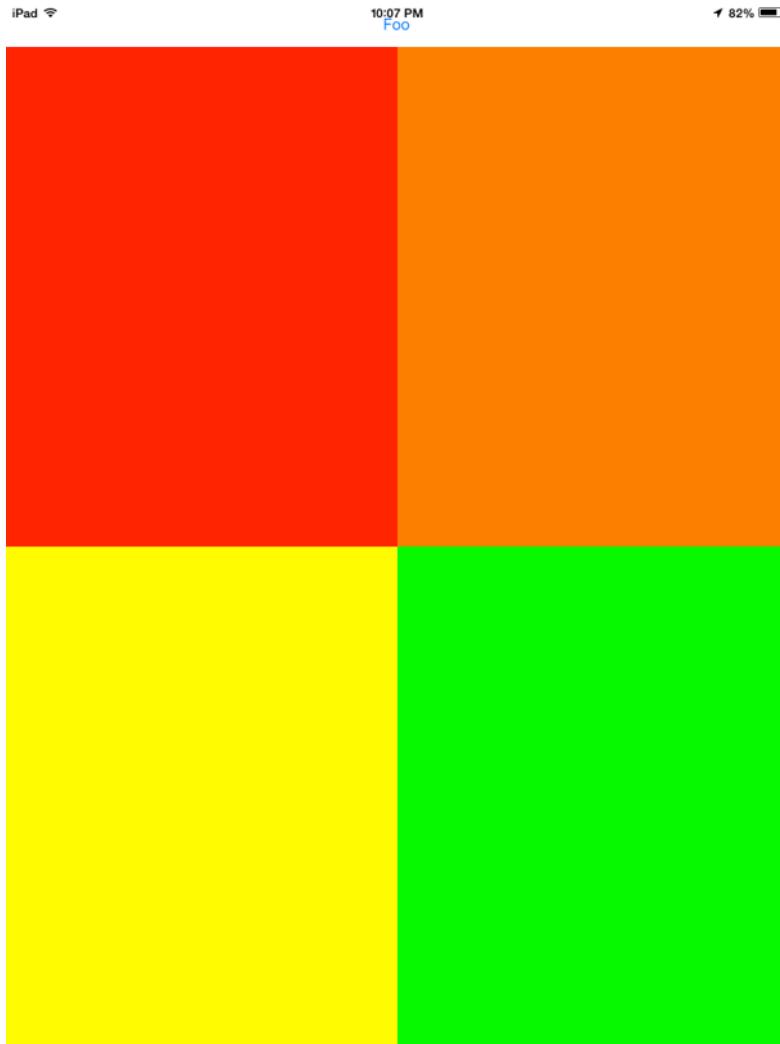
# UIAccessibilityContainer

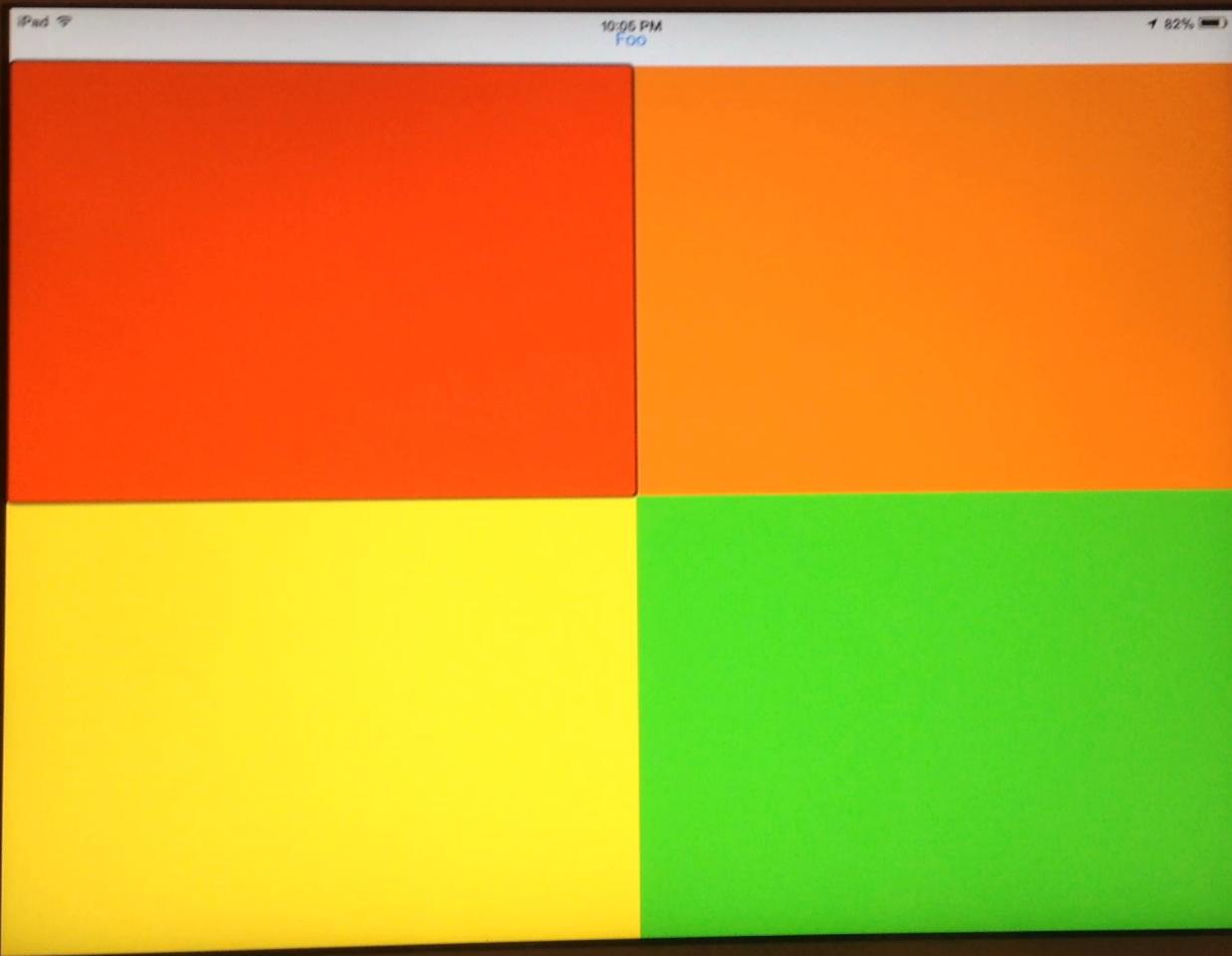
- Programmatically build a UIAccessibility DOM
- Tracks screen-level touches of UIAccessibilityElements

# UIAccessibilityNotifications

- UIAccessibilityPostNotification
  - UIAccessibilityLayoutChangedNotification
    - Small changes, like a control appear/disappear
  - UIAccessibilityScreenChangedNotification
    - Big changes that affect the whole screen.

# DEMO





# Code!

```
- (void)updateAccessibilityElements {  
  
    CGRect topHalfRect, bottomHalfRect;  
    CGRectDivide(self.bounds, &topHalfRect, &bottomHalfRect,  
    floor(CGRectGetHeight(self.bounds) / 2), CGRectGetMinYEdge);  
  
    CGRect topLeftRect, topRightRect;  
    CGRectDivide(topHalfRect, &topLeftRect, &topRightRect,  
    floor(CGRectGetWidth(self.bounds) / 2), CGRectGetMinXEdge);  
  
    CGRect bottomLeftRect, bottomRightRect;  
    CGRectDivide(bottomHalfRect, &bottomLeftRect, &bottomRightRect,  
    floor(CGRectGetWidth(self.bounds) / 2), CGRectGetMinXEdge);  
  
    CGRect topLeftAccessibilityFrame = [self convertRect:topLeftRect toView:nil];  
    CGRect topRightAccessibilityFrame = [self convertRect:topRightRect toView:nil];  
    CGRect bottomLeftAccessibilityFrame = [self convertRect:bottomLeftRect toView:nil];  
    CGRect bottomRightAccessibilityFrame = [self convertRect:bottomRightRect toView:nil];
```

# Code! (Part 2)

```
UIAccessibilityElement *topLeftAccessibilityElement = [[UIAccessibilityElement alloc]
initWithAccessibilityContainer:self];
    topLeftAccessibilityElement.accessibilityLabel = @"Top Left";
    topLeftAccessibilityElement.accessibilityFrame = topLeftAccessibilityFrame;
    topLeftAccessibilityElement.accessibilityTraits = UIAccessibilityTraitButton;

UIAccessibilityElement *topRightAccessibilityElement, bottomLeftAccessibilityElement,
bottomRightAccessibilityElement; // just like topLeftAccessibilityElement...

self.accessibilityElements = @[ topLeftAccessibilityElement,
topRightAccessibilityElement, bottomLeftAccessibilityElement,
bottomRightAccessibilityElement, ];

if (self.window) {
    UIAccessibilityPostNotification(UIAccessibilityLayoutChangedNotification,
        @"Layout Changed");
}
}
```

# Code! (Part 3: This time, it's personal)

```
- (BOOL)isAccessibilityElement {
    return NO;
}
- (NSInteger)accessibilityElementCount {
    NSUInteger accessibilityElementCount = [self.accessibilityElements count];
    return accessibilityElementCount;
}
- (id)accessibilityElementAtIndex:(NSInteger)index {
    if (index < [self.accessibilityElements count]) {
        UIAccessibilityElement *accessibilityElement = self.accessibilityElements[index];
        return accessibilityElement;
    } else {
        return nil;
    }
}
- (NSInteger)indexOfAccessibilityElement:(id)element {
    NSUInteger index = [self.accessibilityElements indexOfObject:element];
    return index;
}
```

# UIAccessibilityAction

- accessibilityIncrement
- accessibilityDecrement
  - Used UIAccessibilityTraitAdjustable
- accessibilityScroll:
- accessibilityPerformEscape
  - Used for modal dialogs
- accessibilityPerformMagicTap
  - Used for single-purpose functions in apps

# UIAccessibilityFocus

- Callbacks when voiceover focuses.
- accessibilityElementDidBecomeFocused
- accessibilityElementDidLoseFocus
- accessibilityElementIsFocused

# UIContentSizeCategory

- New in iOS7
- Settings->General->Text Size
- Programmatic access
  - UIApplication.preferredContentSizeCategory
  - UIContentSizeCategoryDidChangeNotification
- Manage font size to UIContentSizeCategory yourself
- See also +[UIFont preferredFontForTextStyle:]

# A seamless transition

Transition Music for the Classroom  
Pre-Made and Text Editable Templates

Templates provided in 12 color options

## Transition Music

Move to Next Location

Line Up at the Door

Clean Up Time

Pair Share Time

Carpet Time

Mrs. Litette's Learning Detectives

Includes directions on linking your own sound clips\*

# android.accessibility

- android:contentDescription
  - android.view.View#setContentDescription(String)
  - Spoken when user navigates to the view.
- Focus Navigation
  - Test with the D-Pad in the simulator
    - android:focusable
    - android.view.View#setFocusable(boolean)

# Complex View Focus Navigation

- android:nextFocusDown
- android:nextFocusLeft
- android:nextFocusRight
- android:nextFocusUp

# Testing

- Use TalkBack and Explore by Touch on Device
  - Launch “Explore by Touch” Tutorial
  - Not present by default on the emulator
  - Can install the APK yourself!
    - [https://code.google.com/p/eyes-free/downloads/detail?name=com.google.android.marvin.talkback-v3.4.0\\_r99.apk](https://code.google.com/p/eyes-free/downloads/detail?name=com.google.android.marvin.talkback-v3.4.0_r99.apk)
    - ./adb install com.google.android.marvin.talkback-v3.4.0\_r99.apk
  - Jittery, but works!
- Talkback with gestures (scrolling, zooming)
- Ensure focusable controls are >48dp square

# Accessibility Services

- Augment 3<sup>rd</sup> party applications!
  - Or the whole system!
- Walk the Android Accessibility DOM yourself
- Respond to user interactions
- Respond to notifications

# AndroidManifest.xml

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <service android:name=".GuidebookAccessibilityService"
        android:label="@string/service_label"
        android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE">
        <intent-filter>
            <action android:name="android.accessibilityservice.AccessibilityService" />
        </intent-filter>

        <meta-data
            android:name="android.accessibilityservice"
            android:resource="@xml/accessibility_service_config" />
    </service>
</application>
```

# accessibility\_service\_config.xml

```
<accessibility-service xmlns:android="http://schemas.android.com/apk/res/android"  
    android:accessibilityEventTypes="typeAllMask"  
    android:accessibilityFeedbackType="feedbackAudible"  
    android:accessibilityFlags="flagDefault"  
    android:canRetrieveWindowContent="true"  
    android:description="@string/accessibility_service_description"  
    android:notificationTimeout="100"  
    android:packageNames="com.guidebook.android" />
```

- android:settingsActivity
  - Optional



Required for DOM  
access!!

# android:packageNames

- Declares the packages you want to augment
  - Comma-separated
  - Use “android” to augment system packages and observe the home screen
- Discover your system packages
  - adb shell; pm list packages –f
  - Then make a best guess.

# android:accessibilityEventTypes

Declares which events you want to receive

- typeViewClicked
- typeViewLongClicked
- typeViewSelected
- typeViewFocused
- typeViewTextChanged
- typeWindowStateChanged
- typeNotificationStateChanged
- typeViewHoverEnter
- typeViewHoverExit
- typeTouchExplorationGestureStart
- typeTouchExplorationGestureEnd
- typeWindowContentChanged
- typeViewScrolled
- typeViewTextSelectionChanged
- typeAllMask

# android:accessibilityFeedbackType

- feedbackSpoken
- feedbackHaptic
- feedbackAudible
- feedbackVisual
- feedbackGeneric
- feedbackAllMask
- Not sure if this is strictly required to be accurate.

# android:accessibilityFlags

- flagDefault
- flagIncludeNotImportantViews
  - Mostly layoutManager views
- flagRequestTouchExplorationMode
  - Enable Explore by Touch
- flagRequestEnhancedWebAccessibility
  - ???
- flagReportViewIds
  - AccessibilityNodeInfos include view FQNs. Useful for UI Test Runners
- flagRequestFilterKeyEvents
  - Implement global shortcuts. Exclusive to a single service

# android.accessibilityservice.AccessibilityService

- `onServiceConnected()`
- `onAccessibilityEvent(AccessibilityEvent)`
  - Required
- `onGesture(int)`
  - Only works in Explore by Touch mode
- `onKeyEvent(KeyEvent)`
- `onInterrupt()`
  - Required
- `onUnbind()`

# Handy Methods/Objects

- AccessibilityEvent
  - Can have a null source!
- AccessibilityNodeInfo
  - A node of the Accessibility DOM
- AccessibilityService.getRootInActiveWindow()
  - Can return null!

# performAction(int)

- ACTION\_ACCESSIBILITY\_FOCUS
- ACTION\_CLEAR\_ACCESSIBILITY\_FOCUS
- ACTION\_CLEAR\_FOCUS
- ACTION\_CLEAR\_SELECTION
- ACTION\_CLICK
- ACTION\_COPY
- ACTION\_CUT
- ACTION\_FOCUS
- ACTION\_LONG\_CLICK
- ACTION\_NEXT\_AT\_MOVEMENT\_GRANULARITY
- ACTION\_PASTE
- ACTION\_PREVIOUS\_AT\_MOVEMENT\_GRANULARITY
- ACTION\_SCROLL\_BACKWARD
- ACTION\_SCROLL\_FORWARD
- ACTION\_SELECT
- ACTION\_SET\_SELECTION

# DEMO



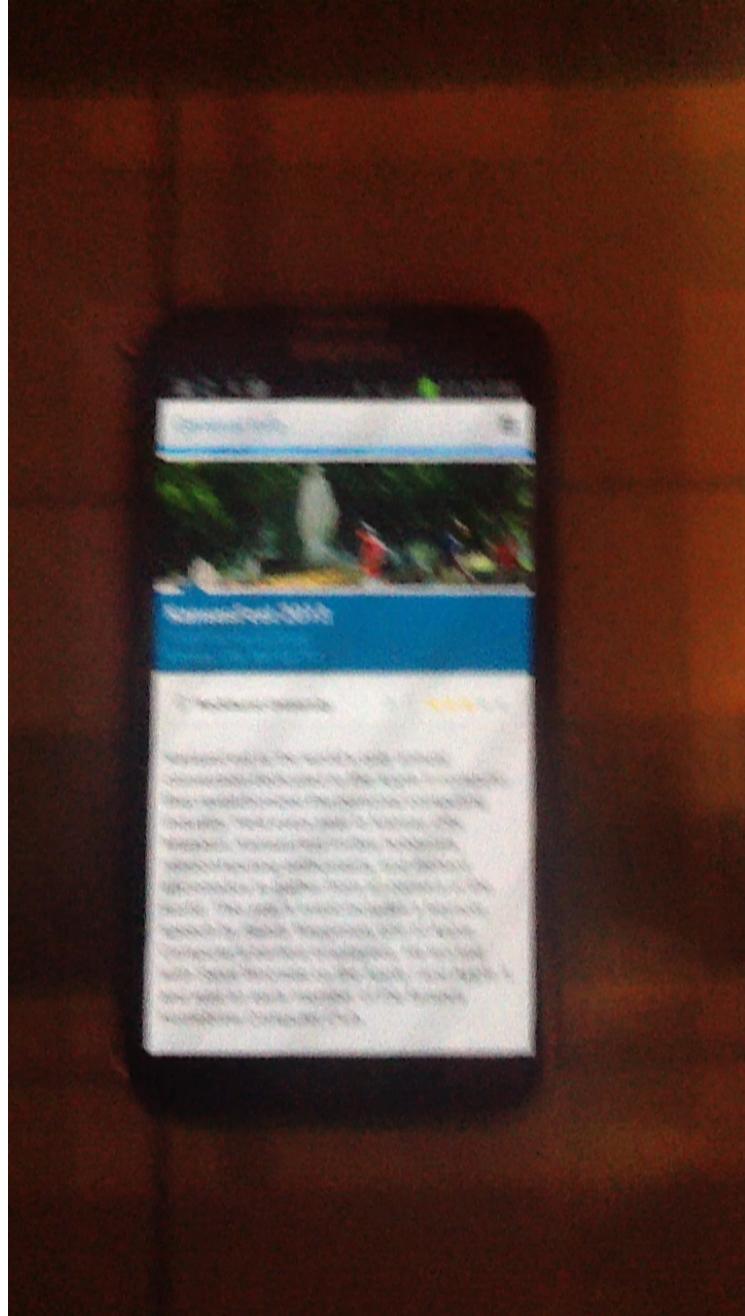
Impossible-to-see  
yellow rectangle  
surrounds the  
stars

# DEMO

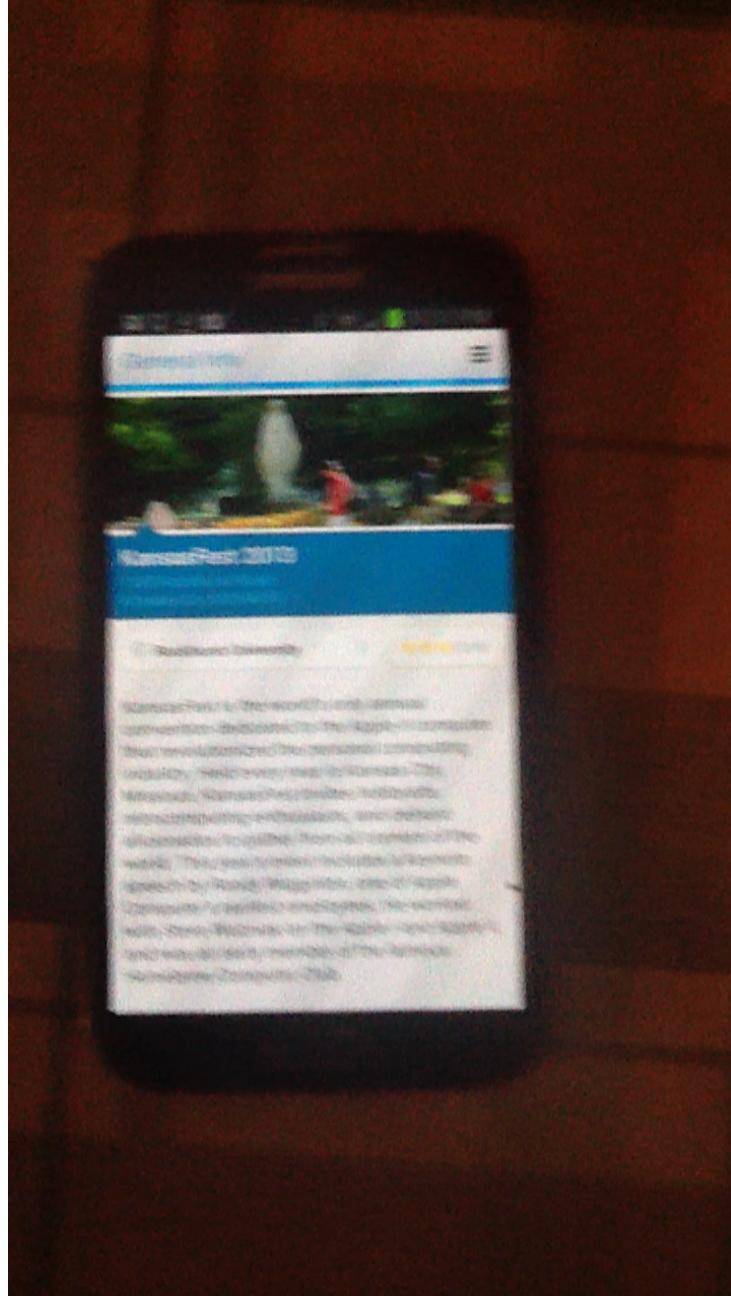


No-quite-so-impossible-to-see white rectangle surrounds the “Overall” score

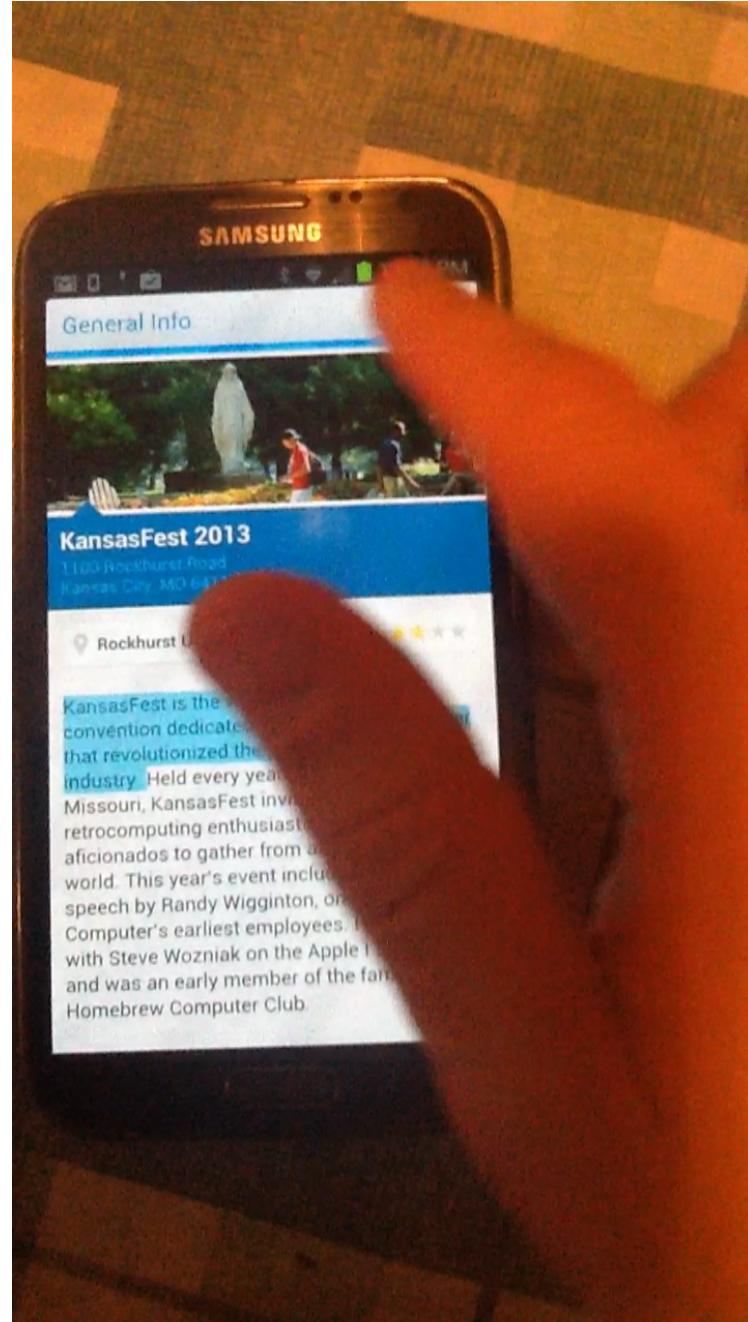
Plain



Explore-by-touch



## Accessibility Service



- <https://developer.apple.com/library/ios/technotes/TestingAccessibilityOfiOSApps/TestAccessibilityonYourDevicewithVoiceOver/TestAccessibilityonYourDevicewithVoiceOver.html>
- <http://www.w3.org/WAI/>
- <http://mattgemmell.com/2010/12/19/accessibility-for-iphone-and-ipad-apps/>
- <http://assistivetechology.about.com/>
- <https://code.google.com/p/eyes-free/>
- [http://eyes-free.googlecode.com/svn/trunk/documentation/android\\_access/index.html](http://eyes-free.googlecode.com/svn/trunk/documentation/android_access/index.html)
- <http://developer.android.com/guide/topics/ui/accessibility/index.html>
- <https://github.com/hborders/mobile-accessibility>

# Questions?

