

Adding Accessibility



Jim Wilson

MOBILE SOLUTIONS DEVELOPER & ARCHITECT

@hedgehogjim blog.jwhh.com



What to Expect from This Module



Goal of Accessibility

Screen Readers and D-pad Navigation

Provide Screen Reader Descriptions

Control D-pad Navigation Behavior

Automate Accessibility Testing

Goal of Accessibility

Usability that goes beyond our traditional expectations of how users physically interact with a device



Achieving the Goal of Accessibility

Expand apps' supported usage scenarios

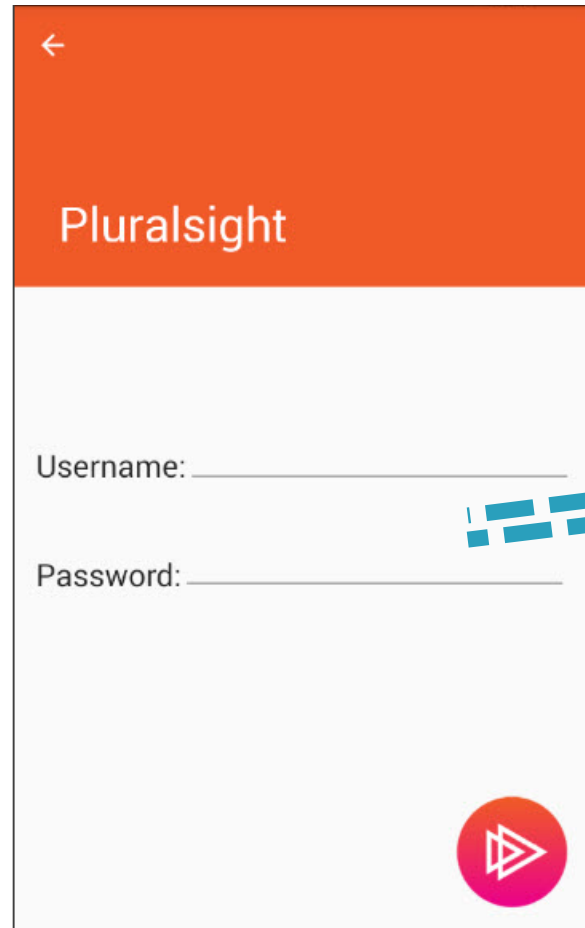
- Empower users with physical limitations
- Support interaction capabilities of non-phone/non-tablet devices

Interaction assumptions we need to change

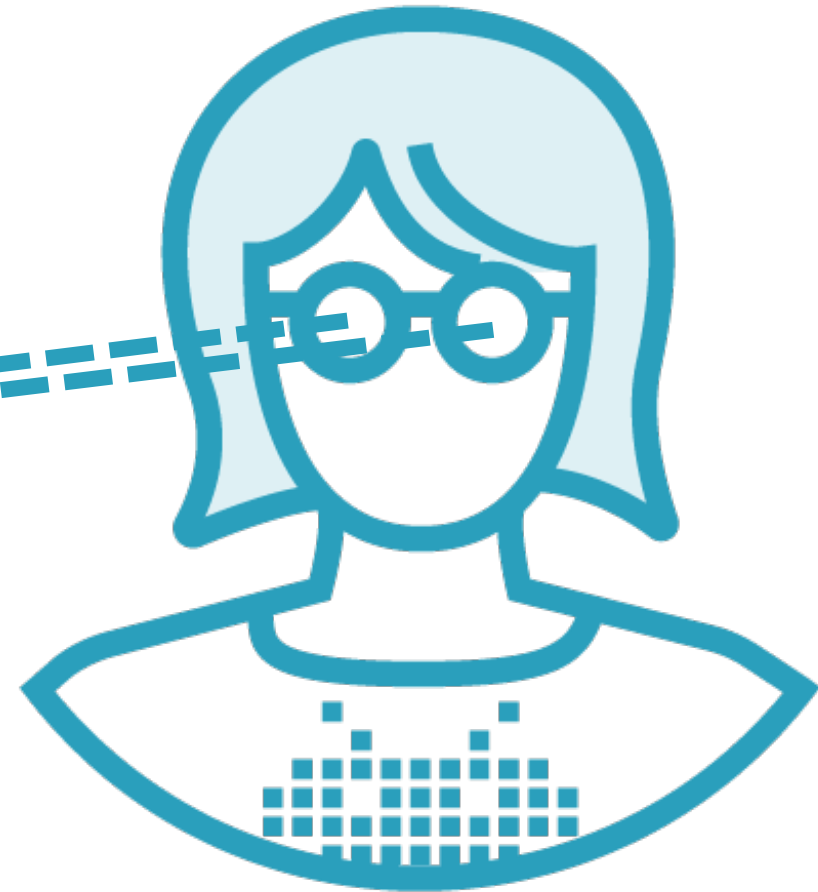
- Information presentation
- Navigation



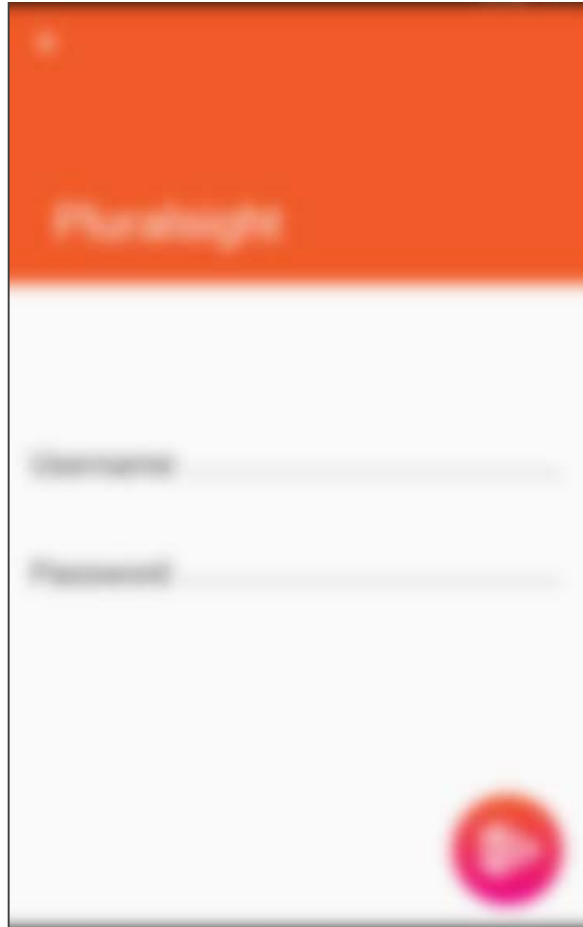
Information Presentation



A login form for Pluralsight. It features an orange header bar with a back arrow and the text 'Pluralsight'. Below the header, there are two input fields: 'Username:' and 'Password:'. A red circular icon with a white play button symbol is located in the bottom right corner of the form.



Information Presentation



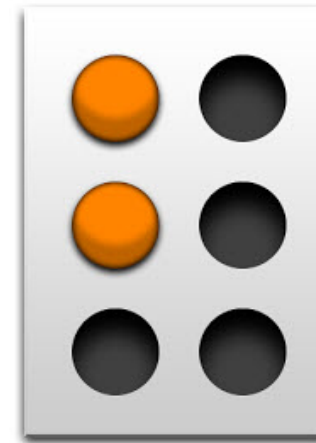
Information Presentation: Screen Readers



TalkBack

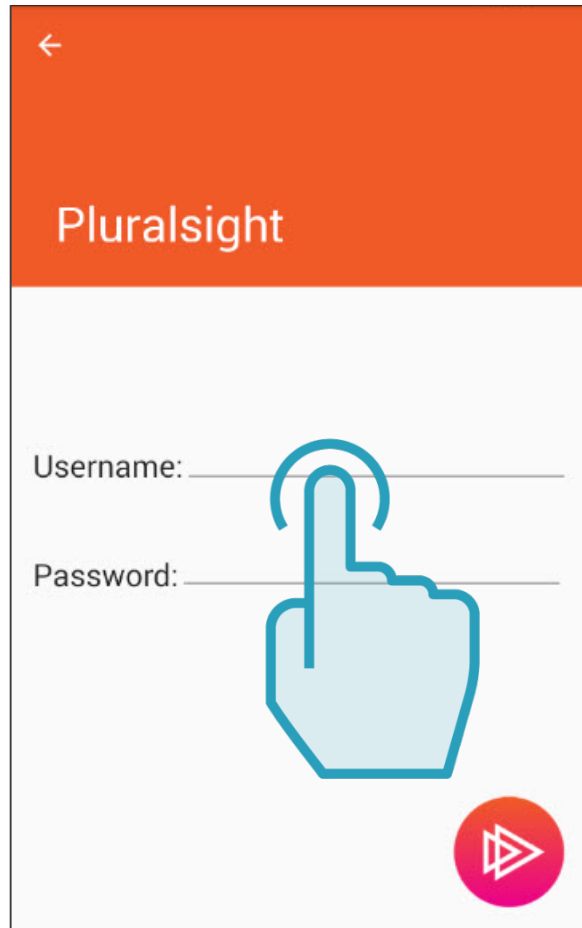


Refreshable
braille
display

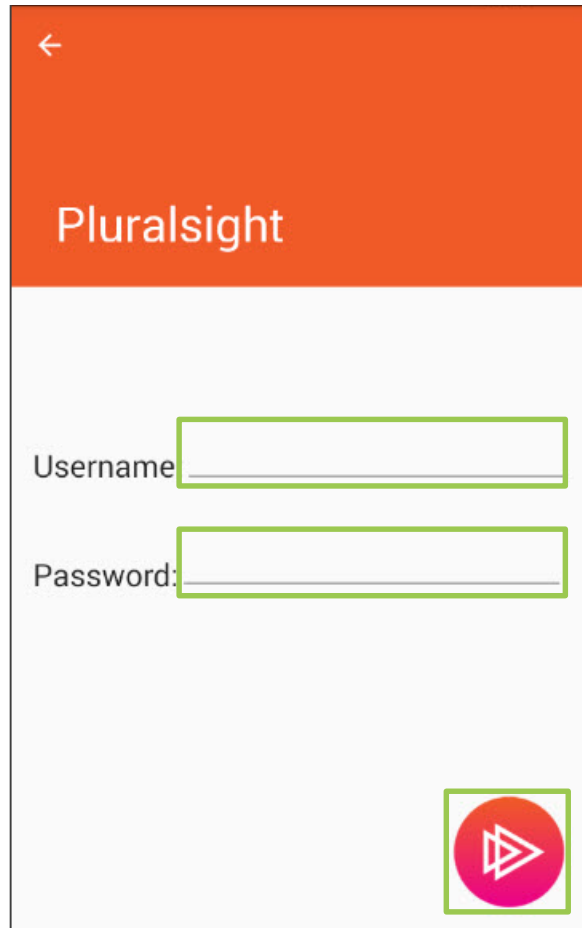


BrailleBack

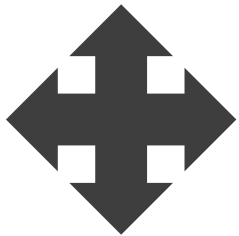
Navigation



Navigation



A mockup of a mobile application login screen. The top section is an orange header bar containing a white back arrow icon on the left and the text "Pluralsight" in white. Below the header is a light gray area with two text input fields. The first field is labeled "Username:" and the second is labeled "Password:". Both fields have a green border. In the bottom right corner of the light gray area, there is a circular logo with a red-to-purple gradient and a white play button icon. The entire mockup is enclosed in a thin black border.



Navigation : D-pad



Assistive Devices



**TV, Video Device,
Gaming System, et. al.**

Achieving the Goal of Accessibility

Accessible apps supported interactions

- Present information with screen readers
- Navigate with d-pad



Screen Reader View Descriptions

Reader will attempt to infer description

- Uses displayed content
- Works well for TextView, Spinner, etc.

View can specify description

- Use view's contentDescription attribute
- Works well for non-text view types



Screen Reader View Descriptions

EditText description

- Any entered text is always included
- Can also include purpose of the EditText

Identifying purpose with hint attribute

- Included in description only when no entered text

Identifying purpose with a TextView

- Can link a TextView to the EditText
- Use TextView labelFor attribute
- Always included in EditText description



Providing Directional Navigation

Android infers directional relationship

- Android default behavior is pretty good
- Does need help sometimes

Layout attributes can specify next view

- Set to ID value of desired view
- `nextFocusUp`
- `nextFocusDown`
- `nextFocusLeft`
- `nextFocusRight`



Automating Accessibility Checks

Make accessibility part of your UI testing

- Automates checks for common issues
- Content description or equivalent
- Duplicate content descriptions
- And more...



Automating Accessibility Checks

Enable AccessibilityChecks before tests

- Creates global view assertions
- Runs check for all ViewAction actions
- Verifies action target and descendants



Automating Accessibility Checks

```
@RunWith(AndroidJUnit4.class)
public class MyTestClass {

    @BeforeClass
    public static void turnOnAccessibility() {

    }

    // Setup rules and tests as normal
    // Accessibility checks run on each ViewAction action
}
```



Summary



Goal of accessibility

- Usability beyond traditional expectations
- Empower users with physical limitations
- Support non-phone/non-tablet devices

Accessible apps supported interactions

- Present information with screen readers
- Navigate with d-pad



Summary



Screen reader descriptions

- Reader can infer most text content
- Use `contentDescription` to explicitly set

D-pad navigation

- Android can infer in many cases
- Use `nextFocusXXX` to explicitly set



Summary



Accessibility testing

- Can incorporate in UI tests
- Use AccessibilityChecks class

