LE/EECS 4443: Mobile User Interfaces (LAB)

Week 3: User Studies & Designing Robust Applications

S.Toyonaga[1]

[1]Lassonde School of Engineering
York University

January 26, 2024

Table of Contents

Motivation & Problem Statement

By the end of this tutorial, you will be able to...

1. Design an effective user study
2. Write a research proposal and/or paper
3. Implement robust applications that can save and restore user states.
4. Design applications that cater to device size and/or screen orientation

User Studies: Scientific Method

- Empirical research requires that we follow the **scientific method:**
  1. What problem or question are you trying to answer?
  2. What are your hypotheses?
  3. How are you going to collect your data?
  4. How are you going to test your hypotheses, given the data?

User Studies: Research Questions

- **First Pass:** Ask **broad** questions

  Is X viable? What are its limits? How intuitive is it?
- **Second Pass:** Reframe the first-pass with testable elements

  Is X (measured in $\cdots$) more Y efficient than Z at performing task α?
- **Note:** There is a tug-of-war between internal and external validity. The more constrained a question, the less generalizable it is to a particular population.

User Studies: Design Phase

Assuming that you have a well-defined research question, we now identify the following:

1. **Independent Variables (Factors)**
   - What are their levels (test conditions) of circumstance ($x \geq 2$)?
   - Typically, these are some characteristic of the apparatus in question
2. **Dependent Variables**
   - A measured human behaviour that relates to the independent variable in some way.
   - **Gamer Move:** Have your application automatically record the dependent variable!

Make sure you identify the units of each variable!

User Studies: Experimental Design

1. Control Variables?
   - These are properties of the study you will be fixing at a particular constant setting.
2. Random Variables?
   - Factors allowed to vary at random.
3. Subject Design?
   - **Within-Subjects:** Tested on $\forall$ levels
   - **Between-Subjects:** Tested on one level
   - **Mixed:** The independent variable(s) are composed of both between-subjects and within-subjects

User Studies: Within-Subjects

- (+) Requires less participants
- (-) Order of Effect / Learning Effect require groups to be counterbalanced (Balanced Latin Square $\vee$ Permutations (Sequences))

User Studies: Between-Subjects

- (+) Avoid interference effects
- (-) Groups must be balanced to ensure equivalent skill levels

User Studies: Hypothesis Testing

Generally, ANOVA (Analysis of Variance) will be sufficient. See the below resources to learn about statistical hypothesis testing and how to do it:

1. Tutorial: Scott MacKenzie's GoStats Application
2. IBM SPSS for York Undergraduate Students

TLDR: $p < 0.05$ means that there is a statistically significant effect! This means that a condition performed much better than the rest!
For post-hoc analysis, use a Scheffe test to determine the significant pairs!

User Studies: Writing the Paper

1. Catchy Titles Are Good: But Avoid Being Cute (Jacob O. Wobbrock)
2. Human-Computer Interaction: An Empirical Research Perspective MacKenzie, I. Scott
3. Mechanics of Style (Scott MacKenzie)

Closing Remarks

Here are some wonderful resources that can streamline parts of the paper writing process.

1. Zotero: An easy-to-use citation manager with a Browser Connector. It makes citing references and generating bibliographies extremely straightforward.
2. How to Read a Search Paper: A very insightful paper outlining how to optimize reading research papers. 10/10!

Activity States

- Recall the Android Activity Lifecycle:
  - Applications are *destroyed* and need to be *recreated* during orientation changes, crashes, leaving-and-returning, etc.,
- We need to override and implement the following methods to support a robust application:
  1. onSaveInstanceState(···)
  2. onRestoreInstanceState(···)

Saving an Instance

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putString("primaryColour", "black");
    outState.putString("secondaryColour", "green");
}
```

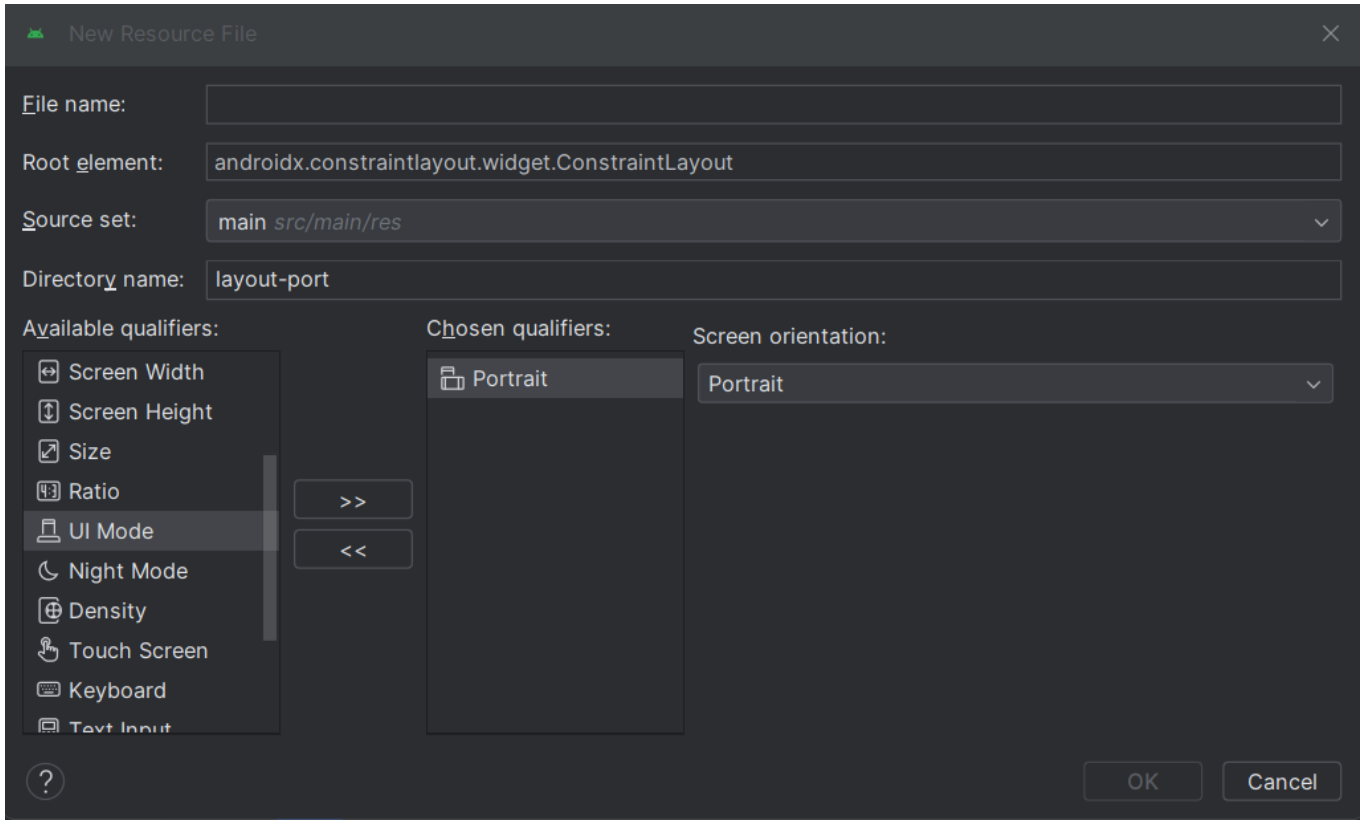
Restoring an Instance

```
@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    primaryColState = savedInstanceState.getString("primaryColour");
}
```

Designing for Different Layouts

- Sometimes it is better to design multiple layouts (xml) for device configurations:
  1. Portrait
  2. Landscape
  3. Square
- Layout → New → Layout Resource File

Designing for Different Layouts



Designing for Different Layouts

- In the layout design manager, you can work accordingly with the **References Devices** list to meet your goals!
- At a minimum, you should ensure that transitioning between **landscape** & **portrait** mode does not lose any critical information.

Conclusion

Remark

Thank you for your time!
Questions?