Due this soon

1. Lo-Fi Prototype Test Report (this Friday)

2. IPA 3 (Monday 3/31 before class)
New Assignment

Interactive Prototype: First steps towards making your app real!

Functional, interactive app written in Android SDK

Due on April 7
Presentations in class on April 7, April 9
Midterm on 3/19

In class. 75 minutes.
Closed book & notes.

We sent out special DSP instructions.
## Office Hours

<table>
<thead>
<tr>
<th>Name</th>
<th>Availability</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maneesh</td>
<td>Tuesday 4-5pm, 535 Soda</td>
<td></td>
</tr>
<tr>
<td>Brittany</td>
<td>Tuesday 3-4pm, 510 Soda</td>
<td></td>
</tr>
<tr>
<td>Steve</td>
<td>Tuesday, 11-noon, 510 Soda</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wednesday 1-2pm, 510 Soda</td>
<td></td>
</tr>
<tr>
<td>Steve</td>
<td>Thursday 11am-1pm, 405 Soda</td>
<td></td>
</tr>
<tr>
<td>Eric</td>
<td>Thursday 4-6pm, 405 Soda</td>
<td></td>
</tr>
</tbody>
</table>

After Midterm – for IPA3

Steve: Thursday 11am-1pm, 405 Soda
Eric: Thursday 4-6pm, 405 Soda
Data Analysis
Bubble Cursor Online Experiment
UC Berkeley CS160

In this short experiment, you will be asked to click on a sequence of targets on screen. You will do this both with a regular mouse cursor and with a different type of cursor that dynamically expands in size to always select the closest target. This Bubble Cursor was introduced by Tovi Grossman and Ravin Balakrishnan at CHI 2005 [1].

This experiment does not work in Google Chrome. Firefox is preferred. Safari might work.

Warm-Up

First, get familiar with the two tasks. Here is the normal cursor. Your job is to click on the red targets - quickly but accurately. Click on a few of them.
Effect Sizes: Time

Normal vs. Bubble cursor at target size 10:
1123ms vs. 852ms: Bubble cursor 31% faster

Normal vs. Bubble cursor at target size 25:
826ms vs. 766ms: Bubble cursor 8% faster

Target size for normal cursor:
1123ms vs 826ms: Larger targets 35% faster

Target size for Bubble cursor:
852ms vs. 766ms: Larger targets 11% faster
Effect Sizes: Error

Normal vs. Bubble cursor, target size 10:
3.4 vs. 0.3 Errors per 20 trials: 1033% fewer errors

Normal vs. Bubble cursor, target size 25:
1.7 vs. 0.3 Errors per 20 trials: 466% fewer errors
Interaction Effects

Relationship between one IV and DV depends on the level of another IV
Interaction: Time

![Graph showing the relationship between size and movement time for normal and bubble interactions. The graph indicates a decrease in movement time as size increases.]

- Size 10
  - Normal: [Data points]
  - Bubble: [Data points]
- Size 25
  - Normal: [Data points]
  - Bubble: [Data points]
Population versus Sample
Are the Results Meaningful?

Hypothesis testing
Hypothesis: Manipulation of IV effects DV in some way
Null hypothesis: Manipulation of IV has no effect on DV
Null hypothesis assumed true unless statistics allow us to reject it

Statistical significance (p value)
Likelihood that results are due to chance variation
p < 0.05 usually considered significant (Sometimes p < 0.01)
Means that < 5% chance that null hypothesis is true

Statistical tests
T-test (1 factor, 2 levels)
Correlation
ANOVA (1 factor, > 2 levels, multiple factors)
MANOVA (> 1 dependent variable)
T-test

**Compare means of 2 groups**

Null hypothesis: No difference between means

**Assumptions**

Samples are normally distributed

Very robust in practice

Population variances are equal (between subjects tests)

Reasonably robust for differing variances

Individual observations in samples are independent

Important!
Single factor analysis of variance (ANOVA)
Compare means for 3 or more levels of a single independent variable

Multi-Way Analysis of variance (n-Way ANOVA)
Compare more than one independent variable
Can find interactions between independent variables

Repeated measures analysis of variance (RM-ANOVA)
Use when > 1 observation per subject (within subjects expt.)

Multi-variate analysis of variance (MANOVA)
Compare between more than one dependent var.

ANOVA tests whether means differ, but does not tell us which means differ – for this we must perform pairwise t-tests

Which should we use for the menu selection example?
Correlation

- Measure extent to which two variables are related
- Does not imply cause and effect
- Example: Ice cream eating and drowning
- Need a large enough sample size

Regression

- Compute the “best fit”
- linear
- logistic
Our Example

Two-Way ANOVA (Cursor, Size) for time:
Main effect for cursor
$F(1,5676) = 424.9, p<0.001$ is statistically significant.

Main effect for size
$F(1,5676)=556.2, p<0.001$ is statistically significant.

Interaction cursor x size
$F(1,5676)=169.5, p<0.001$ is statistically significant.
Two-Way ANOVA (Cursor, Size) for errors:

Main effect for cursor
$F(1, 564) = 314.04$, $p < 0.001$ is statistically significant.

Main effect for size
$F(1, 564) = 44.65$, $p < 0.001$ is statistically significant.

Interaction cursor x size
$F(1, 564) = 43.40$, $p < 0.001$ is statistically significant.
What does p<0.05 mean?

We have a statistically significant difference (at 5% level)
   - In the population there is less than 5% chance that the null hypothesis is true
   - Manipulating IV has an effect on DV
What does $p > 0.05$ mean?

No statistically significant difference in means (at 5% level)
- We cannot reject the null hypothesis
- Manipulating IV may not have an effect on DV
Confidence Intervals

95% Confidence Interval: The range of values in which we’re 95% sure the true population mean falls.

Calculate with the help of the standard error SE.

Standard Deviation: measures variability of individual data points.
Standard Error: measures variability of means

\[ SE = \frac{SD}{\sqrt{N}} \]

\[ 95\% CI = M \pm 1.96 \times SE \]
What's missing from this bar chart?
Mean Errors per 20 trials (error bars show 95% CI)

- bubble25: 0.28 errors
- normal25: 1.70 errors
- normal10: 3.41 errors
- bubble10: 0.30 errors
Draw Conclusions

What is the scope of the finding?
Are there other parameters at play?
Internal validity
Does the experiment reflect real use?
External validity
Summary

Quantitative evaluations
Repeatable, reliable evaluation of interface elements
To control properly, usually limited to low-level issues
Menu selection method A faster than method B

Pros/Cons
Objective measurements
Good internal validity $\rightarrow$ repeatability
But, real-world implications may be difficult to foresee
Significant results doesn’t imply real-world importance
3.05s versus 3.00s for menu selection
Midterm Review
Interface Design Cycle

Design → Evaluate → Prototype → Design
Task Analysis & Contextual Inquiry

Observe existing practices

Create scenarios of actual use

Create models to gain insight into work processes

http://www-personal.umich.edu/~chrisli/m2.html

CS247, Stanford, 2006
Rapid Prototyping

Build a mock-up of design (or more!)

Low fidelity techniques
Paper sketches
Cut, copy, paste
Video segments

Interactive prototyping tools
HTML, Flash, Javascript,
Visual Basic, C#, etc.

http://www.balsamiq.com/products/mockups/examples#wiki

http://www.nngroup.com/reports/prototyping/video_stills.html

Mogridge, Designing Interactions, p.704
Evaluation

Evaluate analytically (no users)

Test with real target users

Low-cost techniques

expert evaluation

walkthroughs

Higher cost

Controlled usability study

http://www.laurasmith.info/UsabilityTest.jpg
IDEO’s Brainstorming Rules

1. Sharpen the Focus
2. Playful Rules
3. Number your Ideas
4. Build and Jump
5. The Space Remembers
6. Stretch Your Mental Muscles
7. Get Physical

Aim for quantity
Hope for quality
What is the point of a critique?

Show off how great your project is.

Get honest reactions, ask for input on open questions.

Q: How is a critique different from a brainstorm?

http://www.flickr.com/photos/crystiancruz/2353909834/
Observation techniques:

Ethnography / Ethnomethodology
Task Analysis
Contextual Inquiry
Cultural Probes
Diary Studies
Prompted “pager” studies
Main Point of Observation

Don’t just trust your intuition to make design decisions.

Observe target users in context to inform your design.
Task Analysis Questions

1. Who is going to use system?
2. What tasks do they now perform?
3. What tasks are desired?
4. How are the tasks learned?
5. Where are the tasks performed?
6. What’s the relationship between user & data?
7. What other tools does the user have?
8. How do users communicate with each other?
9. How often are the tasks performed?
10. What are the time constraints on the tasks?
11. What happens when things go wrong?
Goals of Contextual Inquiry

Method:
“Go where the customer works, observe the customer as she works, and talk to the customer about their work” [Holtzblatt]

Goals:
Get inside the user’s head
See their tasks the way they do
A middle ground between pure observation and pure interview
Guideline: Master-Apprentice Model

Allows user to teach us what they do
- Skill knowledge is usually tacit (can’t put it in books)
- Sometimes literal apprenticeship is best

Matsushita Home Bakery – First automatic bread maker to have twist/stretch motion [Nonaka 95]
“Hypothetical Archetypes”
Archetype: (American Heritage)
An original model or type after which other similar things are patterned; a prototype
An ideal example of a type; quintessence

A precise description of user in terms
Capabilities, inclinations, background
Goals (not tasks)
Why Personas?

It’s hard to reason about users in aggregate, and impossible to please everyone.

General users have too many conflicting goals.
The term **affordance** refers to the *relationship between properties of a physical object and capabilities of a person*, that determine *how the object could be used*.

**Signifiers** help people figure out the affordances of objects without labels or instructions.

What are the signifiers?
- Chair affords sitting (flat surface held by legs)
- Chair affords lifting (??)
- Knobs afford turning (??)
- Buttons afford pushing (??)
- Glass affords seeing through (??)
- Glass affords breaking (??)
Review Conceptual Models

- Designers model may not match user's model
- Users get model from experience & usage
- Users only work with system image, not with designer

What if the two models don’t match?
1. Make Controls Visible
2. Make Sure Mapping is Clear

Mapping: Relationship between controls and their result

Mercedes S500 Car Seat Controller
3. Provide Feedback

People press >> 1 time
Unclear if system has registered the button press
Gulfs of Execution & Evaluation

Mental Model

Physical System

Gulf of Execution

Goals

Gulf of Evaluation

Real World
The Action Cycle

- **Goals**
  - Evaluation of interpretations
  - Interpreting the perception
  - Perceiving the state of the world

- **Execution**
  - Intention to act
  - Sequence of actions
  - Execution of actions

- **The World**
  - start here
Semantic distance reflects the relationship between the user’s intentions and the *meaning of expressions* in the interface languages.

Articulatory distance reflects the relationship between the *physical form* of an expression in the interaction language and its *meaning*. 
Semantic & Articulatory Distance

Semantic
Is it possible to say what one wants to say?
Does the interaction match the user’s conceptual model?

Articulatory
Is form of expression similar to meaning of expression?
Direct Manipulation

An interface that behaves as though the interaction was with a real-world object rather than with an abstract system

Central ideas

1. Visibility of the objects of interest
2. Rapid, reversible, incremental actions
3. Manipulation by pointing and moving
4. Immediate and continuous display of results
Modes: Examples
Quasimodes

Set and hold a mode via conscious, continuous action
Shift key to capitalize (vs. Caps Lock)
Foot pedal that must remain pressed
Pull down menus
Muscle tension reminds users they are holding a mode

Also known as “spring-loaded modes”
Human Info. Processor

Processors:
- Perceptual
- Cognitive
- Motor

Memory:
- Working memory
- Long-term memory

Unified model

Probably inaccurate
Predicts perf. well
Very influential
Perceptual Processor

Cycle time
Quantum experience: 100ms
Percept fusion
Working Memory

Access in chunks
Task dependent construct
7 +/- 2 (Miller)

Decay
Content dependent
1 chunk 73 sec
3 chunks 7 sec

Attention span
Interruptions > decay time
Motor Processor

Receive input from the cognitive processor
Execute motor programs
Pianist: up to 16 finger movements per second
Point of no-return for muscle action
Power Law of Practice

Task time on the nth trial follows a power law

\[ T_n = T_1 n^{-a} + c \]

You get faster the more times you do something!
Hick’s Law

Cost of taking a decision: \[ T = a + b \log_2(n + 1) \]
Fitts’ Law

\[ T = a + b \log_2 \left( \frac{D}{S} + 1 \right) \]

- \( a, b \) = constants (empirically derived)
- \( D \) = distance
- \( S \) = size

ID is Index of Difficulty = \( \log_2 (D/S+1) \)

Models well-rehearsed selection task

\( T \) increases as the **distance** to the target increases
\( T \) decreases as the **size** of the target increases
Considers Distance and Target Size

\[ T = a + b \log_2 \left( \frac{D}{S} + 1 \right) \]

Target 1 \quad \text{Same ID} \rightarrow \text{Same Difficulty} \quad \text{Target 2}
Considers Distance and Target Size

\[ T = a + b \log_2 \left( \frac{D}{S} + 1 \right) \]
Considers Distance and Target Size

\[ T = a + b \log_2 (D/S + 1) \]
What are the assumptions of Fitts Law?
Extend Fitts’ Law to 2D Targets?
Prototyping

PURPOSE

Understand Existing Experience

“Inquiring Actions”

Explore

Experiment

Validate

Communicate

Anchor Discussion

Persuade
Fidelity in Prototyping

Fidelity = level of detail

High fidelity
Prototypes look like the final product

Low fidelity
Artists renditions with many details missing
Paper Prototypes are low-fidelity.
What about software?
Hi-Fi Disadvantages

Distort perceptions of the tester
Formal representation indicates “finished” nature
People comment on color, fonts, and alignment
Discourages major changes
Testers don’t want to change a “finished” design
Sunk-cost reasoning: Designers don’t want to lose effort put into creating hi-fi design
Engineering Interfaces
User Interface Components

Each component is an object with

Bounding box

Paint method for drawing itself

Drawn in the component’s coordinate system

Callbacks to process input events

Mouse clicks, typed keys

Java:
public void paint(Graphics g) {
    g.fillRect(...); // interior
    g.drawString(...); // label
    g.drawRect(...); // outline
}

Cocoa:
(void)drawRect:(NSRect)rect
Layout: Containment Hierarchy

Window

Panel

Label

TextArea

Panel

Button

Button

Enter Text:
Anatomy of an Event

Encapsulates info needed for handlers to react to input

Event Type (mouse moved, key down, etc)
Event Source (the input component)
Timestamp (when did event occur)
Modifiers (Ctrl, Shift, Alt, etc)
Event Content
Mouse: x,y coordinates, button pressed, # clicks
Keyboard: which key was pressed
Event Dispatch Loop

Mouse moved \( (t_0, x, y) \)

Event Queue
• Queue of input events

Event Loop (runs in dedicated thread)
• Remove next event from queue
• Determine event type
• Find proper component(s)
• Invoke callbacks on components
• Repeat, or wait until event arrives

Component
• Invoked callback method
• Update application state
• Request repaint, if needed
Model-View-Controller

**OO Architecture for interactive applications** introduced by Smalltalk developers at PARC ca. 1983
Why MVC?

Combining MVC into one class will not scale
model may have more than one view
each is different and needs update when model changes

Separation eases maintenance and extensibility
easy to add a new view later
model info can be extended, but old views still work
can change a view later, e.g., draw shapes in 3D
flexibility of changing input handling when using separate controllers
Changing the Display

Erase and redraw
using background color to erase fails
drawing shape in new position loses ordering

Better:
Move in model and then redraw view
change position of shapes in model
model keeps shapes in a desired order
tell all views to redraw themselves in order
slow for large / complex drawings
flashing! (can solve / double buffering)
Damage / Redraw Method

View informs windowing system of areas that are damaged does not redraw them right away…

Windowing system batches updates clips them to visible portions of window

Next time waiting for input windowing system calls Repaint() method passes region that needs to be updated
Why use multithreading for UIs?

Not all code can complete quickly inside an event handler. Examples?

- Network access
- File and Database IO
- Simulation

We need to decouple code for long-running computations from code for event handling and screen updates!
Updating the UI from another thread

All common UI frameworks have a single UI thread. You are only allowed to modify the UI from the main thread.

Two fundamental rules:
Do not block the UI thread
Background threads they must not modify the UI.

Solution: When worker thread completes, request update back in the UI thread.
Handler.sendMessage Example

**Main thread**

- Handle event
- Handle event
- Message queue
- Update GUI
- Handle event

**Helper thread**

- Start new thread
- Message queue
- Long computation
- sendMessage("done")

The diagram illustrates the process of handling events in a UI application, using the `Handler.sendMessage` method to communicate between the main thread and a helper thread for long computations.
Usability Testing Methods
<table>
<thead>
<tr>
<th>Genre</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automated</td>
<td>Usability measures computed by software</td>
</tr>
<tr>
<td>Inspection</td>
<td>Based on skills, and experience of evaluators</td>
</tr>
<tr>
<td>Formal</td>
<td>Models and formulas to calculate measures</td>
</tr>
<tr>
<td>Empirical</td>
<td>Usability assessed by testing with real users</td>
</tr>
</tbody>
</table>
Usability Heuristics

“Rules of thumb” describing features of usable systems
Can be used as design principles
Can be used to evaluate a design

Example: Minimize users’ memory load
Nielsen’s Ten Heuristics

H2-1: Visibility of system status
H2-2: Match system and real world
H2-3: User control and freedom
H2-4: Consistency and standards
H2-5: Error prevention
H2-6: Recognition rather than recall
H2-7: Flexibility and efficiency of use
H2-8: Aesthetic and minimalist design
H2-9: Help users recognize, diagnose, recover from errors
H2-10: Help and documentation
Steps in Designing an Experiment

1. State a lucid, testable hypothesis
2. Identify variables (independent, dependent, control, random)
3. Design the experimental protocol
4. Choose user population
5. Apply for human subjects protocol review
6. Run pilot studies
7. Run the experiment
8. Perform statistical analysis
9. Draw conclusions
Experiment Design

**Testable hypothesis**
Precise statement of expected outcome

**Independent variables (factors)**
Attributes we manipulate/vary in each condition
Levels – values for independent variables

**Dependent variables (response variables)**
Outcome of experiment (measurements)
Usually measure user performance
Experiment Design

Control variables
Attributes that will be fixed throughout experiment
Confound – attribute that varied and was not accounted for
Problem: Confound rather than IV could have caused change in DVs
Confounds make it difficult/impossible to draw conclusions

Random variables
Attributes that are randomly sampled
Increases generalizability
Common Metrics in HCI

Performance metrics:
• Task success (binary or multi-level)
• Task completion time
• Errors (slips, mistakes) per task
• Efficiency (cognitive & physical effort)
• Learnability

Satisfaction metrics:
• Self-report on ease of use, frustration, etc.
Goals

**Internal validity**
Manipulation of IV is cause of change in DV
Requires eliminating confounding variables (turn them into IVs or RVs)
Requires that experiment is replicable

**External validity**
Results are generalizable to other experimental settings
*Ecological validity* – results generalizable to real-world settings

**Confidence in results**
Statistics
Between vs. Within Subjects

**Between subjects**

Each participant uses one condition
+/- Participants cannot compare conditions
+ Can collect more data for a given condition
- Need more participants

**Within subjects**

All participants try all conditions
+ Compare one person across conditions to isolate effects of individual diffs
+ Requires fewer participants
- Fatigue effects
- Bias due to ordering/learning effects