CS160: User Interface Design

Usability Testing

03/10/14
Stop Motion Recorder

Make your funny movie
Group Assignments

1. Due Today: Lo-fi Video Prototype

2. New Assignment: Test Low-Fi Prototype
   Get 3 users to try it
   Due Fri March 21 (1.5 weeks)
   Make it short and sweet
Individual Programming Assignment 3

**Due before Spring Break**
(March 21, 11:59pm)

**Due right after Spring Break**
(March 31 before class)
Implement the clapper for taking photos
Learn to use sound API
Learn to interface with camera
Team Assessment
Due Wednesday

### Rate Yourself

<table>
<thead>
<tr>
<th></th>
<th>Outstanding</th>
<th>Very Good</th>
<th>Adequate</th>
<th>Problematic</th>
<th>Inadequate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributing to the team's work</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interacting with teammates</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keeping the team on track</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expecting quality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Having relevant knowledge, skills, and abilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Rate your first teammate

<table>
<thead>
<tr>
<th>First teammate's name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please rate your first teammate along these five team dimensions

<table>
<thead>
<tr>
<th></th>
<th>Outstanding</th>
<th>Very Good</th>
<th>Adequate</th>
<th>Problematic</th>
<th>Inadequate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributing to the team's work</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interacting with teammates</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keeping the team on track</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
No Reading Response
Due Wednesday –
Complete Online
Experiment Instead!

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.
Plan until Spring Break

1. Today:  
   **Designing Usability Studies**

2. Wednesday:  
   **Statistics & Analyzing Study Data**

3. Monday:  
   **Midterm Review**

4. Wednesday:  
   **In-class Midterm**
Loaner Device Checkout

If you’ve submitted a loan form we’ve responded on Piazza. Bring a $200 check (made to UC regents), and pick your device up during my office hours.

Tue 4-5pm, 535 Soda Hall

Nexus 7 tablets are now out. Some Nexus Ones, many Vizios remain.
Midterm on 3/19

In class. 75 minutes.
Closed book & notes.

If you are registered with the DSP office and have special needs, you should have received email from us about exam accommodations. If not, talk to us immediately.
Threading Review
Event Dispatch Loop

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

Launch Thread
- ...Compute...
- Update view
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.
How to properly update the UI

Almost all GUI frameworks offer some mechanism to notify the main thread from another thread.

Android has at least three:

1. Call View.post(Runnable) from worker thread
2. Subclass AsyncTask – creates threads behind the scenes
3. Send messages in one thread with Handler. sendMessage() – message is received in another thread (like IPC)
Handler.sendMessage Example

**Main thread**
- Handle event
- Handle event
  - btn.OnClick()
- Handle event
- Handle event
- Handle event

**Helper thread**
- Start new thread
  - Long computation
  - .
  - .
  - sendMessage("done")

**Message queue**
- update GUI
Working with Sound
Raw audio data – time domain

Pulse-code modulation (PCM) data
Sampled from signal at regular interval (sample rate)
Quantized depending on bit depth

http://en.wikipedia.org/wiki/Pulse-code_modulation
Raw audio data – bit depth

Bit depth is size per sample
As 16-bit mono (1 channel)
Signed short: -32768 to +32767
May need to convert to float: -1.0 to 1.0
Raw audio data – sampling rate

- Sampling rate: number of values (samples) per second
  - 8000 Hz: telephone (adequate for speech)
  - 16000 Hz: modern VoIP products
  - 44100 Hz: audio CD quality

Basic audio record/playback

// create AudioRecord and AudioTrack
int minBufferSize = AudioRecord.getMinBufferSize(SAMPLE_RATE, AudioFormat.CHANNEL_IN_MONO, AudioFormat.ENCODING_PCM_16BIT);
buffer = new byte[minBufferSize];
recorder = new AudioRecord(
    MediaRecorder.AudioSource.MIC, SAMPLE_RATE,
    AudioFormat.CHANNEL_IN_MONO,
    AudioFormat.ENCODING_PCM_16BIT, minBufferSize);
audioTrack = new AudioTrack(AudioManager.STREAM_MUSIC,
    SAMPLE_RATE, AudioFormat.CHANNEL_OUT_MONO,
    AudioFormat.ENCODING_PCM_16BIT, minBufferSize,
    AudioTrack.MODE_STREAM);

recorder.startRecording();
audioTrack.play();

// in another thread, while recording:
text
int bufferReadResult = recorder.read(buffer, 0, buffer.length);
audioTrack.write(buffer, 0, bufferReadResult);
Frequency representation

Fast Fourier Transform (FFT) converts signal to frequency domain
Pitch detection

Using TarsosDSP:

PitchProcessor mPitchProc = new PitchProcessor(
    PitchProcessor.PitchEstimationAlgorithm.AMDF,
    SAMPLE_RATE,bufSize, this);

// in audio processing thread
// after creating AudioEvent
mPitchProc.process(audioEvent);

// elsewhere, implement handlePitch
Low-pass filter

Original

After filter with 1000Hz stop band
Low-pass filter

**Using TarsosDSP:**

```java
LostPassFS mLowPassFilter = new LowPassFS(0, SAMPLE_RATE);

// in audio processing thread, after
// creating an AudioEvent:

mLowPassFilter.process(audioEvent);
byte[] filteredBuffer = audioEvent.getByteBuffer();
mAUDIOTrack.write(filteredBuffer, 0, filteredBuffer.length);
```
Speech to text

Need a recognizer installed (e.g., Google Voice Search)

SpeechRecognizer mSpeechRecognizer =
    SpeechRecognizer.createSpeechRecognizer(this);
Intent i =
    new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH)
    .putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
                RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    .putExtra(RecognizerIntent.EXTRA_CALLING_PACKAGE,
              "com.example.speechtotext");
    .putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak!");
mSpeechRecognizer.startListening(i);
Text to Speech

```java
TextToSpeech mTTS =
new TextToSpeech(this, this);

mTTS.speak("I am a computer",
TextToSpeech.QUEUE_ADD, null);
```
Usability Testing Methods
Iterative Design

Prototype
Low-fi, paper

Design
Brainstorming
Task analysis
Contextual inquiry

Evaluate
Low-fi testing,
Qualitative eval
Quantitative eval
Genres of assessment

**Automated**  Usability measures computed by software

**Inspection**  Based on skills, and experience of evaluators

**Formal**  Models and formulas to calculate measures

**Empirical**  Usability assessed by testing with real users
Empirical Testing is Costly

User studies are very expensive – you need to schedule (and normally pay) many subjects.

User studies may take many hours of the evaluation team’s time.

A user test can easily cost $10k’s
“Discount Usability” Techniques

Cheap
No special labs or equipment needed
The more careful you are, the better it gets

Fast
On order of 1 day to apply
(Standard usability testing may take a week)

Easy to use
Can be taught in 2-4 hours
“Discount Usability” Techniques

**Heuristic Evaluation**
Assess interface based on a predetermined list of criteria

**Cognitive Walkthroughs**
Put yourself in the shoes of a user
Like a code walkthrough

**Other, non-inspection techniques are on the rise**
e.g., online remote experiments with Mechanical Turk
Cognitive Walkthrough

Given an interface prototype or specification, need:

- A detailed task with a concrete goal, ideally motivated by a scenario
- Action sequences for user to complete the task

Ask the following questions for each step:

- Will the users know what to do?
- Will the user notice that the correct action is available?
- Will the user interpret the application feedback correctly?

Record: what would cause problems, and why?

From: Preece, Rogers, Sharp – Interaction Design
Empirical Assessment: Qualitative

**Qualitative**: What we’ve been doing so far

**Contextual Inquiry**: try to understand user’s tasks and conceptual model

**Usability Studies**: look for critical incidents in interface

**Qualitative methods help us:**
Understand what is going on
Look for problems
Roughly evaluate usability of interface
Empirical: Quantitative Studies

**Quantitative**
Use to reliably measure some aspect of interface
Compare two or more designs on a measurable aspect
Contribute to theory of Human-Computer Interaction

**Approaches**
Collect and analyze user events that occur in natural use
Controlled experiments

**Examples of measures**
Time to complete a task, Average number of errors on a task, Users’ ratings of an interface*

*You could argue that users’ perception of speed, error rates etc is more important than their actual values
Comparison

**Qualitative studies**
Faster, less expensive → esp. useful in early stages of design cycle

**Quantitative studies**
Reliable, repeatable result → scientific method
Best studies produce generalizable results
You will conduct a **qualitative** study

We don’t have enough time or subjects for quantitative studies

But you should do a little quantitative analysis

What are your measures?

Compute summary statistics (mean, stdev)

Do you have independent, dependent, and control variables?
Designing Controlled Experiments
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
Example: Bubble Cursor
Lucid, Testable Hypothesis

H1: Users will acquire targets faster with the Bubble cursor (their movement time will be lower).

H2: Users will have a lower error rate with the Bubble cursor.

Other hypotheses?
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
Confounders 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.
Performance Metric: Errors
Satisfaction Metric: Likert Scales

Respondents rate their level of agreement to a statement

Likert data is ordinal, not continuous (matters for analysis)!

“Overall, I am satisfied with the ease of completing the tasks in this scenario”

1: Strongly Disagree
2: Disagree
3: Neither agree nor disagree
4: Agree
5: Strongly agree
Variables for the Bubble Cursor

Independent variables

Dependent variables

Control variables

Random variables
Variables

Independent variables
Cursor type (bubble, normal)
Target Distance
Target Size (Effective vs. Actual?)

Dependent variables
Movement Time
Error Rate
User Satisfaction

Control variables
Color scheme, input device, screen size

Random variables
Location, environment, Attributes of subjects
Age, gender, handedness, …

Conducting studies online vs. in person strongly influences which variables are controlled and which are random.
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
Experimental Protocol

What is the task? (must reflect hypothesis!)
What are all the combinations of conditions?
How often to repeat each combination of conditions?
Between subjects or within subjects
Avoid bias (instructions, ordering, …)
Number of Conditions

Consider all combinations to isolate effects of each IV (factorial design)
(2 cursor types) * (4 distances) * (3 sizes) = 24 combinations

Adding levels or factors can yield lots of combinations!
Reducing Num. of Conditions

Vary only one independent variable leaving others fixed

Problem: ?
Reducing Num. of Conditions

Vary only one independent variable leaving others fixed

Problem: Will miss effects of interactions
Other Reduction Strategies

**Run a few independent variables at a time**
If strong effect, include variable in future studies
Otherwise pick fixed control value for it

**Fractional factorial design**
Procedures for choosing subset of independent variables to vary in each experiment
Choosing Subjects

Pick balanced sample reflecting intended user population
Novices, experts
Age group
Sex
....

Example
12 non-colorblind right-handed adults (male & female)

Population group can also be an IV or a controlled variable
What is the disadvantage of making population a controlled var?
Between Subjects Design

Wilma and Betty use one interface

Dino and Fred use the other
Within Subjects Design

Everyone uses both interfaces
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
Within Subjects: Ordering Effects

In within-subjects designs ordering of conditions is a variable that can confound results
Why?

Turn it into a random variable
Randomize order of conditions across subjects
Counterbalancing (ensure all orderings are covered)
Latin square (partial counterbalancing)
...
Run the Experiment

**Always pilot it first!**
Reveals unexpected problems
Can’t change experiment design after starting it

**Always follow same steps – use a checklist**

**Get consent from subjects**

**Debrief subjects afterwards**
Participate in Bubble Cursor Experiment for Wednesday’s class!