CS160: User Interface Design

Threads, Usability Testing 03/05/12

Berkeley
UNIVERSITY OF CALIFORNIA

Assignments

Due Today:
Group Video Prototype

New Assignment:
Test Low-Fi Prototype with 3 users. You have 1 week – make it short and sweet

Plan Through Midterm

Today 3/5:
Threads & Designing Usability Studies

Wednesday 3/7:
Statistics & Analyzing Study Data

Monday 3/12:
Midterm Review
Due: Lo-fi test with three users

Wednesday 3/14:
In-class Midterm

Midterm on 3/14

In class. 75 minutes.

Closed book & notes.

Review on Monday 3/12.

If you are registered with the DSP office and have special needs, we need to see your letter by **this Wednesday,** 3/7, 1pm to make accommodations.

Threading in User Interfaces

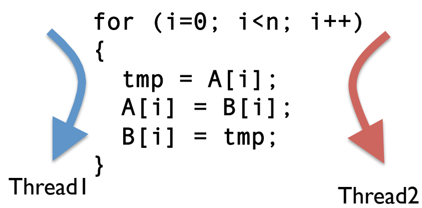
What is a thread?

A **thread** is a **partial virtual machine**.

Each thread has its own stack (and local variables) but shares its heap with other threads in the same application.

Threads can be independently scheduled by the OS/VM.

```
for (i=0; i<n; i++)
{
  tmp = A[i];
  A[i] = B[i];
  B[i] = tmp;
}
```



Threads vs. Processes

A **process** is a **complete virtual machine** with its own stack and heap.

Threads share memory – processes don't.

Threads can communicate through shared memory, processes need other mechanisms (IPC = inter-process communication).

Pros and Cons

Why use threads?

Useful model of concurrent execution, both on single processors (time-division multiplexing) and on multi processor/multi-core systems

Threads are relatively cheap to create, versatile because of shared memory

Why wouldn't one use threads?

Complicated programming model. Multithreaded programming is one of the biggest productivity killers of all time

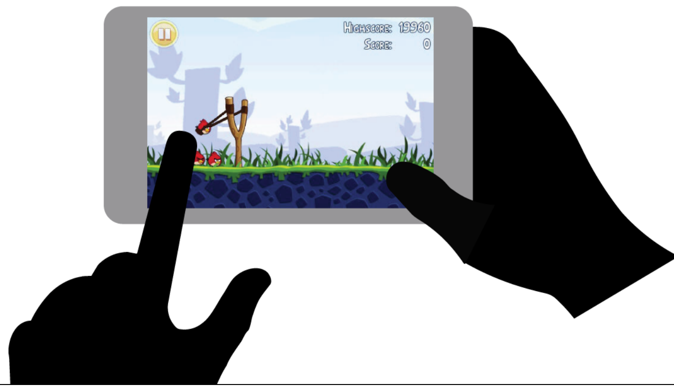
(locks, semaphores, monitors, mutexes, signals, spawn, fork, join,…)

"After a long and careful analysis the results are clear: 11 out of 10 people can't handle threads."

— Todd Hoff

Why use multithreading for UIs?

Interactive programs need to respond **quickly** to user input. Direct manipulation assumes that objects onscreen respond to user's touch/cursor.

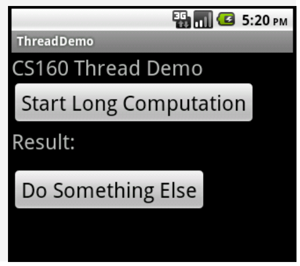


Why use multithreading for UIs?

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

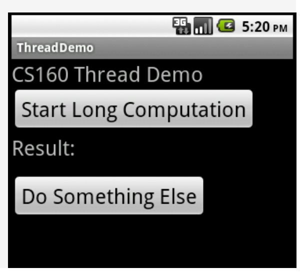
Android Demo: Long-running Task

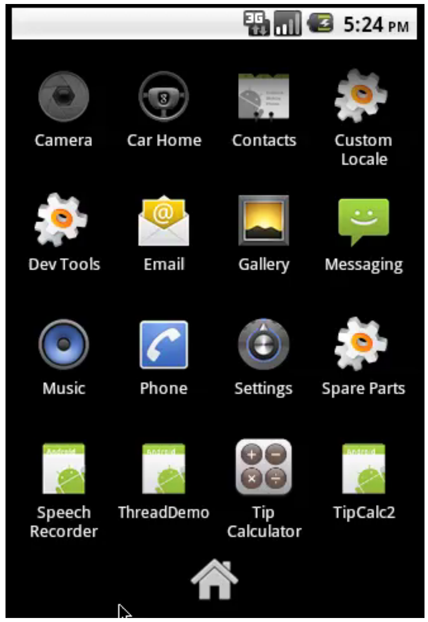
```
btnStart.setOnClickListener(
new OnClickListener() {
  public void onClick(View arg0) {
    // start long computation
    sleep(60000);
    // update UI when done
    txtResult.
    setText("Done.");
});
```



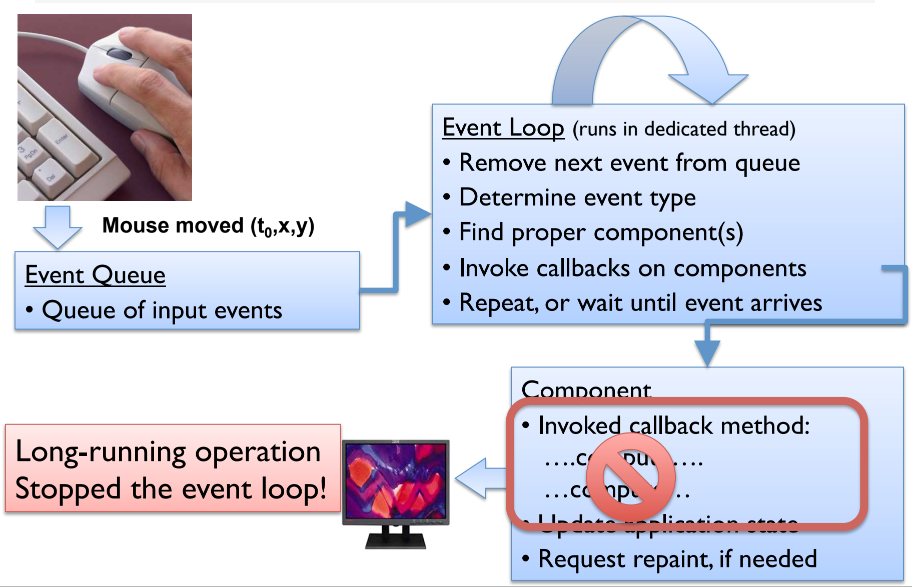
Android Demo: Long-running Task

```
btnStart.setOnClickListener(
new OnClickListener() {
  public void onClick(View arg0) {
    // start long computation
    Thread.sleep(60000);
    // update UI when done
    txtResult.
    setText("Done.");
});
```



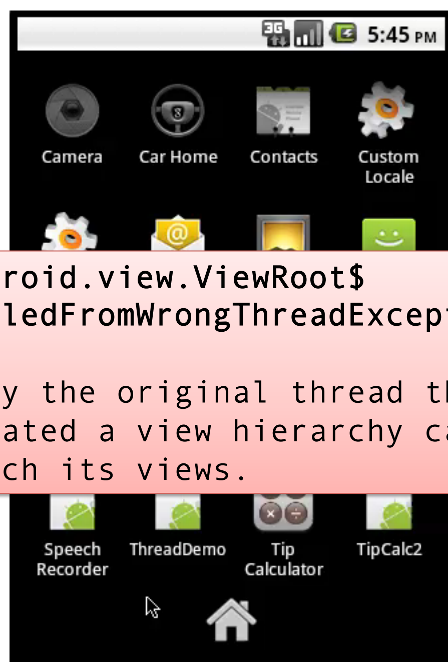


Event Dispatch Loop



Android Demo with Threads

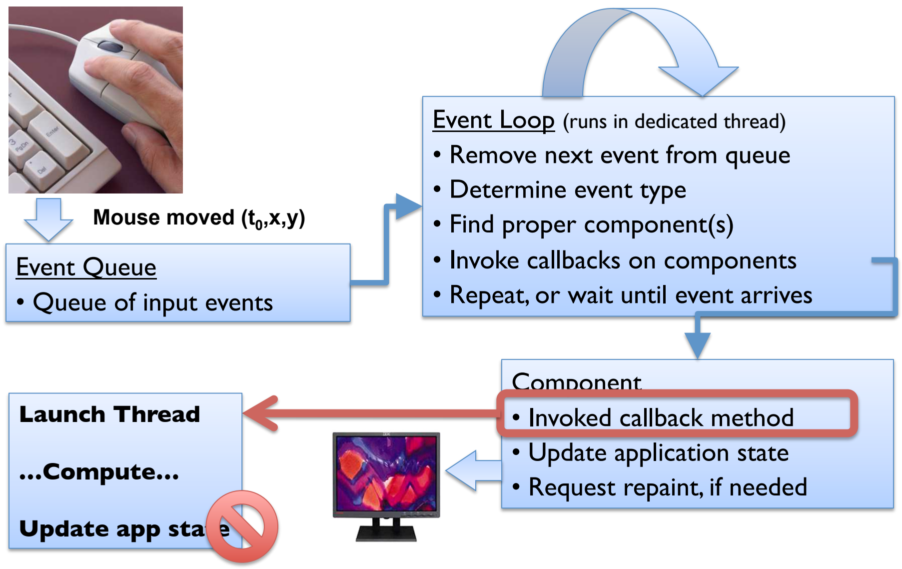
```
btnStart.setOnClickListener(
new OnClickListener() {
  public void onClick(View v) {
    new Thread( new Runnable() {
      public void run() {
        //start long computation
        Thread.sleep(10000);
        // update UI when done
        txtResult.setText("Done.");
    }}).start(); //start new thread
}});
```




```
android.view.ViewRoot$
CalledFromWrongThreadException:

Only the original thread that
created a view hierarchy can
touch its views.
```

Event Dispatch Loop



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 **must not modify the UI**

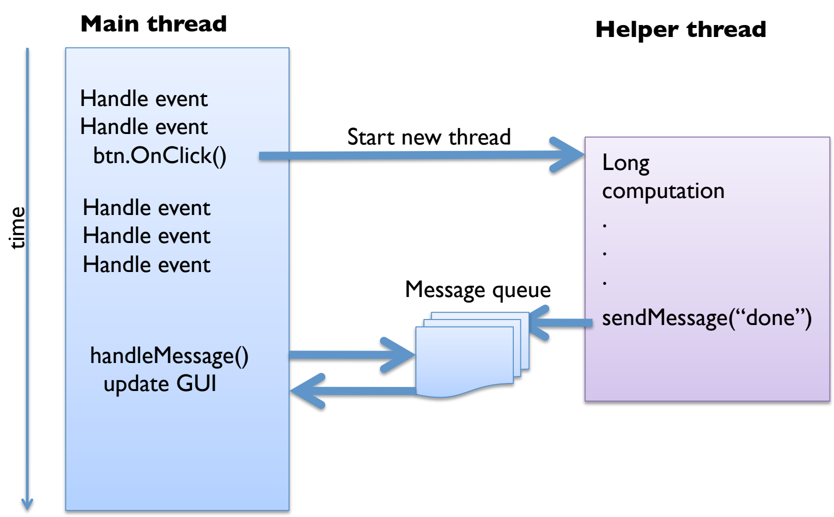
Solution: When worker thread completes, request update back in the UI thread.

How to properly update the UI

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

Notification commands are framework dependent

Handler.sendMessage Example

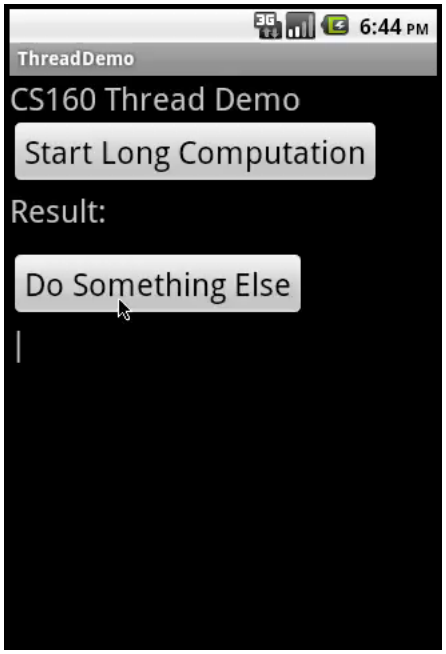


Android Code: Activity

```
public class ThreadDemo extends Activity {
final Handler handler = new Handler() {
  public void handleMessage (Message msg) {
     // update UI
    txtResult.setText((String)msg.obj);
  }
};
```

Android Code: Event Handler

```
public void onClick(View arg0) {
    new Thread(new Runnable() {
        public void run() {
            // long computation…
            Message msg = new Message();
            msg.obj = "Done.";
            handler.sendMessage(msg);
    }}).start();
}
```

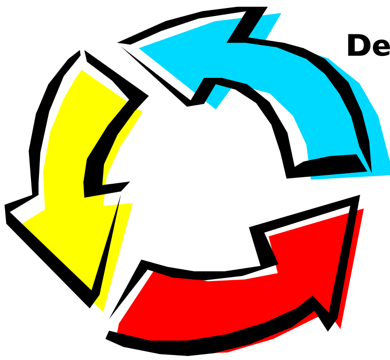


Usability Testing Methods

Iterative Design

Design
Brainstorming
Task analysis
Contextual inquiry

Prototype
Low-fi, paper



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:

- Write detailed task with a concrete goal, motivated by a scenario
- Write action sequence required 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

Pilot User Study Assignment (after midterm)

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

The Bubble Cursor:
Enhancing Target Acquisition by
Dynamic Resizing of the
Cursor's Activation Area

Tovi Grossman

Ravin Balakrishnan

Dynamic Graphics Project Lab
Department of Computer Science
University of Toronto
www.dgp.toronto.edu

Lucid, Testable Hypothesis

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

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

Confounds make it difficult/impossible to draw conclusions

Random variables

Attributes that are randomly sampled

Increases generalizability

Variable Types

Nominal: categories with labels, no order

Ordinal: categories with rank order

Continuous:
interval (w/o zero point), ratio (w/ zero point)

Common Metrics in HCI

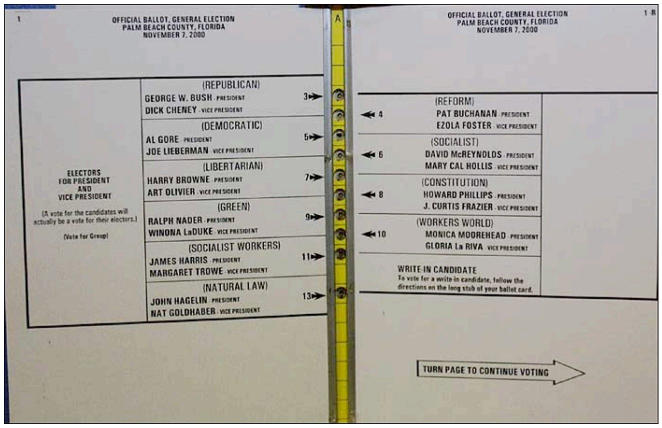
Performance metrics:

- 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



stcsig.org

media.tbo.com / AP

Performance Metric: Lostness

Smith 1996:

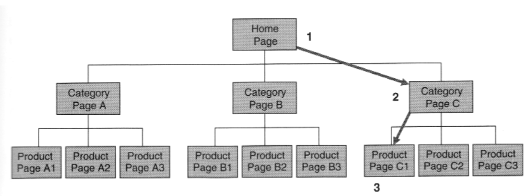
N: # of different pages visited

S: # of total pages visited, incl. revisits

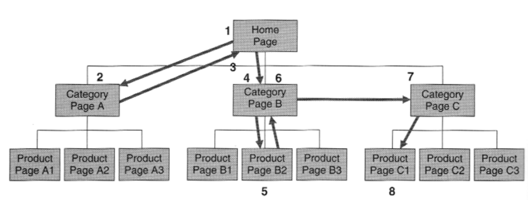
R: minimum # of pages to accomplish task

Lostness = $\sqrt{(N/S-1)^2+(R/N-1)^2}$

Smith 1996



Optimum number of steps (three) to accomplish a task that involves finding a target item on Product Page C1 starting from the homepage.



Actual number of steps a participant took in getting to the target item on Product Page C1. Note that each revisit to the same page is counted, giving a total of eight steps.

Satisfaction Metric: Likert Scales

Respondents rate their level of agreement to a statement

"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

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

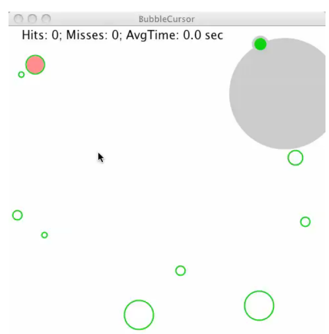
Variables for the Bubble Cursor

Independent variables

Dependent variables

Control variables

Random variables



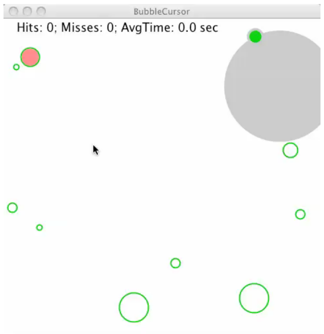
Variables

Independent variables
Cursor type (bubble, normal, area?)
Target Distance
Target Width (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)

(3 cursor types) * (3 distances) * (3 widths) = 27 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?