Low-Fidelity Prototyping

CS160: User Interfaces
Maneesh Agrawala
• Unlabeled button-like thing
• Uses familiar calculator metaphor (Takes it too far?!!)
  – Cryptic labels (CE vs. C) (M+, MS, MR, MS → why not show what is in memory)
  – Single line of display text
• Can use keypad but affordance is not visible

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

Mouse moved (t₀,x,y)
Review: Model-View-Controller

Architecture for interactive apps
- Model: Info the application manipulates
- View: Visual display of the model
- Controller: Receives input & decides what they do

![Diagram of Model-View-Controller](image)

Review: Storyboarding

Series of key frames depicting key steps in reaching a goal
- Can use a pin board for easy rearrangement/editing
- Describe the interaction in context
- Often useful to show user in at least 1st frame (establishing shot)
Assignment (Due Feb. 26)

- Android Intro Application
  - Build a simple application for searching and browsing Flickr photos using Android
  - Individual assignment
  - Requires significant work – get started early

- Emphasis on:
  - Designing a UI for searching / browsing
  - Creating appropriate Activity & Intent objects
  - Handling Activity lifecycle

Topics

- Creating the prototype
- Wizard of Oz prototype testing
Creating the Prototype

Why Do We Prototype?
Get feedback on our design faster
  – saves money
Experiment with alternative designs
Fix problems before code is written
Keep the design centered on the user
Fidelity in Prototyping

Fidelity refers to the level of detail

High fidelity
  – Prototypes look like the final product

Low fidelity
  – Artists renditions with many details missing
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
- Designers don’t want to lose effort put into creating hi-fi design
Materials

Large, heavy, white paper (11 x 17)
5x8 in. index cards
Post-it notes
Tape, stick glue, correction tape
Pens & markers (colors & sizes)
Transparencies (including colored)
Colorforms (toy stores)
Scissors, X-acto knives, etc.
Welcome to ESP.

Your Telebeers session is Tues. Sept. 21 @ 10am

Your current schedule is empty. Please click on Add a course to continue.

### ESP EECS Schedule Planner

<table>
<thead>
<tr>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-9</td>
<td>9-10</td>
<td>10-11</td>
<td>11-12</td>
<td>12-1</td>
</tr>
<tr>
<td>1-2</td>
<td>2-3</td>
<td>3-4</td>
<td>4-5</td>
<td>5-6</td>
</tr>
</tbody>
</table>
Interface Elements

Constructing the Prototype

Set a deadline
   – Don’t think too long - build it!

Draw a window frame on large paper
   – Draw at a large size, but use correct aspect ratio

Put different screen regions on cards
   – Anything that moves, changes, appears/disappears
   – Use greek-ing to indicate text if necessary

Ready response for any user action
   – e.g., Have those pull-down menus already made

Use photocopier to make many versions
Wizard of Oz Prototype Testing

Preparing for a Test

Select your participants
- Understand background of intended users
- Use a questionnaire to get the people you need
- Don’t use friends or family

Prepare scenarios that are
- Typical of the product during actual use
- Make prototype support these (small, yet broad)

Practice running the computer to avoid “bugs”
- You need every menu and dialog for the tasks
- All widgets the user might press
  - Remember “help” and “cancel” buttons
Conducting a Test

Three or Four testers (preferable)
- Greeter - Puts users at ease & gets data
- Facilitator - only team member who speaks
  - Gives instructions & encourages thoughts, opinions
- Computer - knows application logic & controls it
  - Always simulates the response, w/o explanation
- Observer(s) - Take notes & recommendations

Typical session should be approximately 1 hour
- Preparation, the test, debriefing
Conducting a Test (cont.)

Greet
– Get forms filled, assure confidentiality, etc.

Test
– Facilitator explains how test will work
  • Performs a simple task
– Facilitator hands written tasks to the user
  • Must be clear & detailed
– Facilitator keeps getting “output” from participant
  • “What are you thinking right now?”, “Think aloud”
– Observers record what happens
  • Avoid strong reactions: frowning, laughing, impatience – biases the test
– Designers should not lead participants
  • Let users figure things out themselves as much as possible
  • Only answer questions if user remains stuck for a long time

Conducting a Test (cont.)

Debrief
– Fill out post-evaluation questionnaire
– Ask questions about parts you saw problems on
– Gather impressions
– Give thanks
Wizard of Oz Tips

Rehearse your actions
- For a complicated UI, make a flowchart which is hidden from the user
- Make list of legal words for a speech interface

Stay “in role”
- You are a computer, and have no common sense, or ability to understand spoken English.

Facilitator can remind user of the rules if the user gets stuck

Think Aloud

Explain how to “think aloud.”
- Explain why you want participants to think aloud, and demonstrate how to do it. E.g.:
- We have found that we get a great deal of information from these informal tests if we ask people to think aloud. Would you like me to demonstrate?
Evaluating Results

Sort & prioritize observations
- What was important?
- Lots of problems in the same area?

Create a written report on findings
- Gives agenda for meeting on design changes

Make changes & iterate

Advantages of Low-Fi Prototyping

Takes only a few hours
- No expensive equipment needed

Can test multiple alternatives
- Fast iterations
  - Number of iterations is tied to final quality

Can change the design as you test
- If users are trying to use the interface in a way you didn’t design it – go with what they think! Adapt!

Especially useful for hard to implement features
- Speech and handwriting recognition
**Drawbacks of Lo-Fi Prototyping**

Evolving the prototype requires redrawing
   – Can be slow (but reprogramming usually slower)

Lack support for “design memory”

Force manual translation to electronic format

Do not allow real-time end-user interaction

**Caveats**

There is a down-side to the informal design approach:

Often hard to involve *paying* clients as subjects – they treat
the fidelity of the interface as a sign of development effort

Mitigators: involve them early and often, correspond with the
same people, explain the process up front (set expectations)
Exercise

Airport “wayfinder” for travelers

Problem statement:
- When passing through airports, people often have difficulty finding the services they need. The availability of these services, as well as how to get to them, is not obvious, which can result in missed flights, weary travelers, and a stressful customer experience.

Target persona: Angela, ~31, business traveler
- Wants to travel without hassle
- 30 minute layover in unfamiliar airport
- What might she want to do in this time?
- What kind of interface would support her tasks?
Constraints

- PDA/Smart phone class hardware
- Wireless infrastructure available
- Low resolution location information available

Exercise

Work with your project group

Your tasks
- Brainstorm about Angela’s goals
  - What does she want to do at the airport
  - What information does she need
  - What kind of interface will support her tasks
- Create an initial low fidelity prototype
- Debug the interface with users from another group
  - Does the interface meet Angela’s needs?
  - Is the interface hassle-free?
  - Is the interface confusing or difficult?
Summary
Informal prototypes allow you to design (and test!) before writing code.

Rapid evolution and elimination of many problems happens in this phase.

Paper+ink is the traditional tool

Next Time
Human Information Processing (Perception)