Lecture 9: Prototyping

Today's Topics

- Paper prototypes
- Computer prototypes
- Wizard of Oz prototypes

Why Prototype?

- Get feedback earlier, cheaper
- Experiment with alternatives
- Easier to change or throw away
**Prototype Fidelity**

- Low fidelity: omits details
- High fidelity: more like finished product

**Fidelity is Multidimensional**

- **Breadth**: % of features covered
  - Only enough features for certain tasks
- **Depth**: degree of functionality
  - Limited choices, canned responses, no error handling

**More Dimensions of Fidelity**

- **Look**: appearance, graphic design
  - Sketchy, hand-drawn
- **Feel**: input method
  - Pointing & writing feels very different from mouse & keyboard

**Paper Prototype**

- Interactive paper mockup
  - Sketches of screen appearance
  - Paper pieces show windows, menus, dialog boxes
- Interaction is natural
  - Pointing with a finger = mouse click
  - Writing = typing
- A person simulates the computer’s operation
  - Putting down & picking up pieces
  - Writing responses on the “screen”
  - Describing effects that are hard to show on paper
- Low fidelity in look & feel
- High fidelity in depth (person simulates the backend)
Why Paper Prototyping?

- Faster to build
  - Sketching is faster than programming
- Easier to change
  - Easy to make changes between user tests, or even during a user test
  - No code investment—everything will be thrown away (except the design)
- Focuses attention on big picture
  - Designer doesn’t waste time on details
  - Customer makes more creative suggestions, not nitpicking
- Nonprogrammers can help
  - Only kindergarten skills are required

Tools for Paper Prototyping

- White poster board (11”x14”)
  - For background, window frame
- Big (unlined) index cards (4”x6”, 5”x8”)
  - For menus, window contents, and dialog boxes
- Restickable glue
  - For keeping pieces fixed
- White correction tape
  - For text fields, checkboxes, short messages
- Overhead transparencies
  - For highlighting, user “typing”
- Photocopier
  - For making multiple blanks
- Pens & markers, scissors, tape

Tips for Good Paper Prototypes

- Make it larger than life
- Make it monochrome
- Replace tricky visual feedback with audible descriptions
  - Tooltips, drag & drop, animation, progress bar
- Keep pieces organized
  - Use folders & open envelopes

Hand-Drawn or Not?

- Make it larger than life
- Make it monochrome
- Replace tricky visual feedback with audible descriptions
  - Tooltips, drag & drop, animation, progress bar
- Keep pieces organized
  - Use folders & open envelopes
Roles for design team
- Computer
  - Simulates prototype
  - Doesn’t give any feedback that the computer wouldn’t
- Facilitator
  - Presents interface and tasks to the user
  - Encourages user to "think aloud" by asking questions
  - Keeps user test from getting off track
- Observer
  - Keeps mouth shut, sits on hands if necessary
  - Takes copious notes

How to Test a Paper Prototype
What You Can Learn from a Paper Prototype

- Conceptual model
  - Do users understand it?
- Functionality
  - Does it do what’s needed? Missing features?
- Navigation & task flow
  - Can users find their way around?
  - Are information preconditions met?
- Terminology
  - Do users understand labels?
- Screen contents
  - What needs to go on the screen?

What You Can't Learn

- Look: color, font, whitespace, etc
- Feel: Fitts’s Law issues
- Response time
- Are small changes noticed?
  - Even the tiniest change to a paper prototype is clearly visible to user
- Exploration vs. deliberation
  - Users are more deliberate with a paper prototype; they don’t explore or thrash as much

Computer Prototype

- Interactive software simulation
- High-fidelity in look & feel
- Low-fidelity in depth
  - Paper prototype had a human simulating the backend; computer prototype doesn’t
  - Computer prototype is typically horizontal: covers most features, but no backend

What You Can Learn From Computer Prototypes

- Everything you learn from a paper prototype, plus:
- Screen layout
  - Is it clear, overwhelming, distracting, complicated?
  - Can users find important elements?
- Colors, fonts, icons, other elements
  - Well-chosen?
- Interactive feedback
  - Do users notice & respond to status bar messages, cursor changes, other feedback
- Fitts’s Law issues
  - Controls big enough? Too close together? Scrolling list is too long?
Why Use Prototyping Tools?

- Faster than coding
- No debugging
- Easier to change or throw away
- Don’t let Java do your graphic design

Computer Prototyping Techniques

- Storyboard
  - Sequence of painted screenshots connected by hyperlinks (“hotspots”)
- Form builder
  - Real windows assembled from a palette of widgets (buttons, text fields, labels, etc.)
- Wizard of Oz
  - Computer frontend, human backend

Storyboarding Tools

- HTML
  - image maps
- Flash/Director
  - animation + actions
- PowerPoint
  - images + links + animation
- All these tools have scripting languages, too
  - Help orchestrate the transitions
- For high fidelity look, take screenshots of widgets from a form builder

Pros & Cons of Storyboarding

- Pros
  - You can draw anything
- Cons
  - No text entry
  - Widgets aren’t active
  - “Hunt for the hotspot”
Form Builders

- HTML pages and forms
  - Natural if you’re building a web application
  - May have low-fidelity look otherwise
- Java GUI builders
  - Sun NetBeans
  - Eclipse Visual Editor
  - Borland JBuilder
- Other GUI builders
  - Visual Basic, .NET Windows Forms
  - Mac Interface Builder
  - Qt Designer
- Tips
  - Use absolute positioning for now

Pros & Cons of Form Builders

- Pros
  - Actual controls, not just pictures of them
  - Can hook in some backend if you need it
    - But then you won’t want to throw it away
- Cons
  - Limits thinking to standard widgets
  - Useless for rich graphical interfaces

Wizard of Oz Prototype

- Software simulation with a human in the loop to help
- “Wizard of Oz” = “man behind the curtain”
  - Wizard is usually but not always hidden
- Often used to simulate future technology
  - Speech recognition
  - Learning
- Issues
  - Two UIs to worry about: user’s and wizard’s
  - Wizard has to be mechanical