Lecture 8: Computer Prototyping

Today's Topics

- Course evaluation results
- Quiz preview
- Paper prototype post-mortem
- Computer prototyping

Midterm Course Evaluation Results

- Overhead problems
- Hall of Fame & Shame
- Slides (but not notes) in advance

Quiz on Wednesday

- Topics
  - Usability
  - Iterative design
  - User & task analysis
  - Model human processor
  - Color
  - Conceptual models & metaphor
  - Affordance, constraint, visibility, feedback
  - Errors
  - Nielsen's heuristics
  - Heuristic evaluation
  - Prototyping
  - Graphic design principles
- Everything is fair game
  - Class discussion, lecture notes, readings, assignments
- Closed book exam, 80 minutes

Paper Prototyping Post-Mortem

- Time to make prototype
- Materials that worked well or badly
- Useful implementation tricks
- Parts of UI that are hard to prototype
- How it feels to be a user
- How it feels to watch a user
- Surprises learned from watching users

Paper Prototyping is Not Enough

- Low fidelity in:
  - Look
  - Feel
  - Dynamics
  - Response time
  - Context
- Users can't try it without a human to simulate computer

Computer Prototype

- Interactive software simulation
- High-fidelity in appearance & interaction
- 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?

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.)

Why Use Prototyping Tools?

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

Storyboarding Tools

- PowerPoint
  - drawings + hyperlinks
- Flash/Director
  - animation + actions
- HTML
  - image maps
- 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
  - “Hunt for the hotspot”
  - No text entry
  - Widgets aren’t active

Form Builders

- HTML/Dreamweaver
  - Natural if you're building a web application
  - May have low-fidelity look otherwise
- Visual Basic
- Java GUI builders
  - Sun NetBeans
  - IBM Visual Age/WebSphere
  - Borland JBuilder
- Tips
  - Use absolute positioning for now

Pros & Cons of Form Builder

- Pros
  - Direct manipulation editing, not coding
  - 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

Technical Challenges to Graphic Design

- Window resizing
- Platform differences
- Internationalization