Announcements

HW3 - status update?
Heads up: HW4 will be short.
No class on 10/19.
Design Tools
Report back from Wednesday

What was easy to express with the paper prototype? What was hard?

How did the conversation change between your pair (one prototype) and the group (two prototypes)?
Prototyping revisited

PURPOSE

Understand Existing Experience

“Inquiring Actions”

Communicate

Explore

Experiment

Validate

Anchor Discussion

Persuade
ASPECTS TESTED

Role / Context

Look & Feel

Looks Like
Behaves Like

Implementation
FUNCTIONALITY

Scope

- Partial
  - Horizontal slice (Breadth)
  - Vertical slice (Depth)

- Full

Method of Realization

- Low-Fidelity Mockup
- Simulation / Wizard of Oz

Working Implementation

- Created With Production Tools
- Created With Different Tools
Prototypes enable exploration and iteration around concrete artifacts.

Lawson, How Designers Think, Architectural Press
Cross, Designerly Ways of Knowing, Springer
Schrage, Serious Play, HBS Press
Mogridge, Designing Interactions, MIT Press
Buxton, Sketching User Experiences, Morgan Kaufmann
Prototyping != Development

Success criteria for interaction prototypes

- Rapid to construct
- Conveys user experience
- Uncovers design space
- Elicits the right feedback

(Some) success criteria for interactive products

- Robust
- Complete
- Polished
The first (successful) UI prototyping app?

Welcome to HyperCard

HyperCard is a unique software tool that allows you to do more with your computer.

With HyperCard, you use documents called stacks. Stacks can help you do many different things—for example, you could use a stack to keep track of your appointments, manage your expenses, learn a new language, or record and play sounds.

To learn more, click “What is HyperCard?,” to the right.

©1993-1995 Apple Computer, Inc. All Rights Reserved.
Professional Use of Prototyping Tools Reported by Myers

- Adobe Photoshop: 83%
- Adobe Dreamweaver: 66%
- Microsoft PowerPoint: 54%
- Adobe Illustrator: 52%
- Adobe Flash: 43%
- Microsoft Visio: 37%
- Adobe InDesign: 25%
- Omni Group Omnigraffle: 23%
- Microsoft Visual Studio: 19%
- Adobe Fireworks: 17%
- Adobe Director: 13%
- Microsoft Frontpage: 12%
- Adobe AfterEffects: 10%
- Axure RP: 8%
- Adobe GoLive: 6%
- Adobe Flex: 6%
- Microsoft Expression Blend: 4%
Design Tools @ Berkeley
Informal vs. Formal Representations

Informal visual representation
- communicates “unfinished”
- encourages creativity
- faster to create

Formal visual representation
- communicates “finished”
- inhibits creativity (detailing)
- slower to create
Informal User Interfaces

Take advantage of natural input modalities

- speaking
- writing
- gesturing
- sketching

Minimize recognition of the input

- allow users to work & communicate naturally
- document rather than transform
“Put me in a room with a pad & a pencil and set me up against a hundred people with a hundred computers -- I’ll out-create every goddamn sonofabitch in the room”

Ray Bradbury, Wired 6.10
Example: Web Design

Interviews with 11 professional web designers

Designers create representations of sites at multiple levels of detail:
Site maps, storyboards, schematics, mockups

All designers sketched - at all levels
Sketching enables “working through” ideas
But: hard to make incremental modifications
Implications for a Web Design Tool

- Support multiple levels of abstraction
- Use sketching as the input modality
- Support transformation of sketches to more formal/functional representations
Video: denim_talk.rm (VLC)
10 years later:

Microsoft SketchFlow:

Balsamiq Mockups:
http://balsamiq.com/products/mockups
(available for your projects)
SUEDE: Informal Prototyping for Speech-based UIs

Supports design practice
- example scripts
- Wizard of Oz
- error simulation
- iterative design (design-test-analysis)

Informal user interface
- no speech recognition or synthesis
- need not be programming expert
- fast & fluid design
SUEDE Video:

http://www.youtube.com/user/StanfordHCl#p/u/13/ULfAwZAUUbS
Informal Design Tools

Iterative design is the key to good UIs

Informal tools are the key to iterative design (?)

Berkeley built several informal design tools

- Web Design (Outpost & Denim)
- Speech UI Design (Suede)
- Multimodal, Cross device UI Design (CrossWeaver & Damask)
Exploring Alternatives
How can tools support the creation of multiple user interface alternatives?

Myers et al., VL/HCC 2008:
Comparing working UI alternatives “very important” or “crucial” for 43% of surveyed interaction designers (N=210)
Design Galleries

Marks et al,
SIGGRAPH 97
Parallel Pies
Juxtapose
Interaction Designers Write Code

Flash

Arduino / Processing

Dreamweaver
3 Requirements for Programmed Interactions

Manage parameter variations
Manage code alternatives
Access variations & alternatives at runtime

```java
public int ATTENUATION = 20;  // = 10;  // = 50;
public int BASE_ALPHA = 50;
public int POSITIVE_RESPONSE = 6;
```
3 Requirements for Programmed Interactions

- Manage parameter variations
- Manage code alternatives
- Access variations & alternatives at runtime

Processing code brought in by an interviewee

```java
int calculateNextSize(int[][] currentSizes, int i, int j) {
    float denominator = 0;
    int sumOfNeighbors = 0;
    int maxOfNeighbors = 0;
    if (i != 0) {
        sumOfNeighbors += currentSizes[i - 1][j]; denominator += 1;
        maxOfNeighbors = currentSizes[i - 1][j];
        // if (j != 0) sumOfNeighbors += currentSizes[i - 1][j - 1]/2; denominator += .5;
        // if (j != currentSizes[0].length - 1) sumOfNeighbors += currentSizes[i - 1][j + 1]/2; denominator += .5;
    }
    if (i != currentSizes.length - 1) {
        sumOfNeighbors += currentSizes[i + 1][j]; denominator += 1;
        if (currentSizes[i + 1][j] > maxOfNeighbors) maxOfNeighbors = currentSizes[i + 1][j];
        // if (j != 0) sumOfNeighbors += currentSizes[i + 1][j - 1]/2; denominator += .5;
        // if (j != currentSizes[0].length - 1) sumOfNeighbors += currentSizes[i + 1][j + 1]/2; denominator += .5;
    }
    return maxOfNeighbors; // Ratio
}
```
3 Requirements for Programmed Interactions

- Manage parameter variations
- Manage code alternatives
- Access variations & alternatives at runtime
Juxtapose: Source alternatives...
Juxtapose: Source alternatives...
...are executed in parallel,
and tuned through a generated UI.
Parameter Tuning

```javascript
var angmax: Number = 270;
var dimin: Number = 61;
var maxchild: Number = 3;
var maxrec: Number = 4;
```
Example: Tuning Phosphor

Baudisch, UIST 2006
Example: Tuning Phosphor
Tangible Control
Parallel Editing

Juxtapose editor for ActionScript 2. Generates Flash files.
Parallel Editing
Parallel Editing

```java
class FlashApplication {

```
Parallel Editing

```apl
FlashApplication { 
  var showLocalStreets:Boolean = false; 
  var showLocalNames:Boolean = true; 
  var showMajorStreets:Boolean = false; 
  var showMajorNames:Boolean = true; 
  var showHighways:Boolean = true;  // show highways and interstates? 
  var showPOI:Boolean = true;     // show points of interest? 
}
```
Example: Rethinking the Progress Bar

Harrison et al., UIST 2007

Progress bar
test application

Progress profiles
A structured approach to alternatives eliminates cruft and enables more exploration. Tool support requires connecting authoring and execution environments.

3 Techniques:
- Linked editing to manage code alternatives
- Execute set of programs side-by-side
- Auto-generate tuning interface