Announcements

Projects: ready, set, go!
Recall one more time

Implication:

To improve the process, encourage more rapid reflection, or improve the quality of the reflection

To improve the tools, create alternatives that make reflection easier to do or more informative
Goal: facilitate rapid iteration

[Hartmann, PhD thesis '09]

Prototypes enable exploration and iteration around concrete artifacts. The more fluid the prototyping process is, the more you can learn before you sink time into engineering.
Design tools should...

[Hartmann, PhD thesis ’09]

Decrease UI construction time

Isolate designers from implementation details

Enable designers to explore an interface technology previously reserved to engineers or other technology experts
Today

Rapid prototyping
Exploring alternatives: design galleries
Feedback
Tangible and physical interaction
Rapid prototyping
Recall: Sketching Interfaces Like Krazy
[Landay, CHI '96]

Combine the fluidity of paper-based sketching with the interactivity of tools

Technique: sketch recognition of basic UI components

Led to many projects on lower fidelity prototyping of interactive systems: let's tour some of them
DENIM: web storyboarding

[Lin et al., CHI ’00]

Enable fluid interactions for web site design

Work at a higher level of abstraction than HTML
Designer’s Outpost

[Klemmer et al., UIST ’01]

Fluid interactive brainstorming that bridges physical and digital artifacts
$1$ gesture recognizer

[Wobbrock, Wilson, and Li, UIST ’07]

Training an end-to-end ML system for gesture recognition would take thousands of examples and a lot of time—infeasible for prototyping.

The “$1$ recognizer”: quick 100 lines of code for 97% accuracy with only one example.

- Resample, rescale, rotate, & template match
Exploring alternatives
Design galleries

[Marks et al., SIGGRAPH '97]

Automatically generate perceptually-varying alternatives within a design space

Helps the designer explore other feasible approaches

Now a widely-adopted technique inside of design tools
// load asset file "task1-assets.swf", which defines movieclips "circle", "box", and "boxes"

//@SWF_ASSET_FILE task1-assets.swf

class FlashApplication {
    static var app:FlashApplication;

    // variables to be tuned
    var xNumber:Number = 12; //ORANGE 2..12
    var yNumber:Number = 12; //ORANGE 2..12
    var scale:Number = 100; //ORANGE 1..195

    //class constructor - all initialization code goes in here
    function FlashApplication() {
        var canvasWidth:Number = Stage.width;
        var canvasHeight:Number = Stage.height;

        var total:Number = xNumber*yNumber; //total number of atoms that will be created
        var gridSpacing:Number = 20; //spacing between atoms
        var counter:Number = 0;
        //_root.scale = 100;
        _root._x = 0;
        _root._y = 0;

        //create the parent of all our atoms.
Explore alternatives

Tighten the loop by allowing users to explore design spaces and alternatives on a live version

[Hartmann et al., UIST 2008]
DesignScape

[O’Donovan, Agarwala, and Hertzmann CHI ’15]

Model graphic design requirements such as alignment and hierarchical segmentation, then generate alternatives
Feedback
Learning Visual Importance

[Bylinskii et al., UIST '17]
Voyant: crowd feedback

[Xu, Huang, and Bailey CSCW ’13]
Physical and tangible interface prototyping
The challenge of physical prototyping

Prototype the bits, or prototype the atoms?

Goal: lower the threshold to prototype interactive systems that depend on electronics and physical materials
The Toastboard

Ubiquitous Instrumentation and Automated Checking of Breadboarded Circuits

Daniel Drew*, Julie Newcomb†, William McGrath‡, Filip Maksmovic*, David Mellis*, Bjoern Hartmann*
*UC Berkeley EECS, †University of Washington PLSE, ‡Stanford University HCI Group
ddrew73, fil, mellis, bjoern@berkeley.edu, newcombj@cs.washington.edu, wmcgrath@stanford.edu
Phidgets

The first physical prototyping platform

USB plug-and-programmable I/O

  servos, LEDs, buttons, sliders, etc.

Goal: program physical devices like you would a GUI widget
Led to: Arduino

Maker board for artists, programmers and hobbyists
Led to: Makey Makey

[Silver et al., TEI '12]

Alligator clips map onto keystrokes

Banana Space Bar

MaKey MaKey
d.tools: prototyping behavior

[Hartmann et al., UIST ’06]

Plug-and-play
HW, visual
statechart
behaviors

prototyping with d.tools
Sensor interaction by demonstration

[Hartmann et al., CHI '07]
Fabricating capacitive hardware
[Savage et al., UIST ’12]

Author behaviors, and the software does circuit layout
Replacing electronics with cameras
[Savage et al., UIST '13]
Create the physical prototype in CAD software, then use 3D printing and a camera to try out the interaction without building the electronics.
<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro to Interaction; Intro to Content Creation</td>
</tr>
<tr>
<td>2</td>
<td>Intro to Design Process; Interaction pt 1</td>
</tr>
<tr>
<td>3</td>
<td>Methods; Interaction pt 2</td>
</tr>
<tr>
<td>4</td>
<td>Content Creation</td>
</tr>
<tr>
<td>5</td>
<td>Design Process</td>
</tr>
<tr>
<td>6</td>
<td>Foundations; Cognition</td>
</tr>
<tr>
<td>7</td>
<td>Social Computing; Accessibility</td>
</tr>
<tr>
<td>8</td>
<td>Collaboration; AR/VR</td>
</tr>
<tr>
<td>9</td>
<td>AI + HCI; NLP + HCI</td>
</tr>
<tr>
<td>10</td>
<td>Final Presentations</td>
</tr>
</tbody>
</table>