Project Status

Looking Forward

3e: Digital Mockup Due Thursday 11/16
3f: Report Due Monday 11/27
3g: Presentation Due Wednesday 11/29

4a: Initial Website Due Monday 11/27
4b: Video Prototype Due Monday 12/4

Other Assignments

Reading 5 Due Saturday 12/2, Sooner is Better
Exam

Next Tuesday 11/21, in Denny 303

Mostly short answer, some long answer

Content drawn from lecture and readings

Compilation of the lecture slides is posted

Q&A scheduled Monday at 3:00 in CSE 403
Tools and Interfaces

Why Interface Tools?
Case Study of Model-View-Controller
Case Study of Animation
Sapir-Whorf Hypothesis
Thoughtfulness in Tools
Case Study in Self-Tracking
Objectives

Be able to:

Describe benefits of tools in interface implementation, why we use them

Describe the Model/View/Controller approach to organizing interface implementation

Describe why tools eventually limit design thinking
Sequential Programs

Program takes control, prompts for input

Person waits on the program

Program says when it is ready for more input, which the person then provides
Sequential Programs

while true {
    print "Prompt for Input"
    input = read_line_of_text()
    output = do_work()
    print output
}

Person is literally modeled as a file
Event-Driven Programming

A program waits for a person to provide input

All communication done via events

“mouse down”, “item drag”, “key up”

All events go to a queue

Ensures events handled in order

Hides specifics from applications

How many of these queues? How can you tell?
Basic Interactive Software Loop

do {
    e = read_event();
    dispatch_event(e);
    if (damage_exists())
        update_display();
} while (e.type != WM_QUIT);

All interactive software has this somewhere
Basic Interactive Software Loop

Have you ever written this loop?
Basic Interactive Software Loop

Have you ever written this loop?

Contrast with:

“One of the most complex aspects of Xlib programming is designing the event loop, which must take into account all of the possible events that can occur in a window.”

Understanding Tools

We use tools because they

- Identify common or important practices
- Package those practices in a framework
- Make it easy to follow those practices
- Make it easier to focus on our application

What are the benefits of this?
Understanding Tools

We use tools because they

- Identify common or important practices
- Package those practices in a framework
- Make it easy to follow those practices
- Make it easier to focus on our application

What are the benefits of this?

- Being faster allows more iterative design
- Implementation is generally better in the tool
- Consistency across applications using same tool
Understanding Tools

Why is designing tools difficult?

Need to understand the core practices and problems
Those are often evolving with technology and design

Example: Responsiveness in event-driven interface

Event-driven interaction is asynchronous

How to maintain responsiveness in the interface while executing some large computation?
Understanding Tools

Why is designing tools difficult?

- Need to understand the core practices and problems
- Those are often evolving with technology and design

Example: Responsiveness in event-driven interface

Cursor:
- \texttt{WaitCursor} vs. \texttt{CWaitCursor} vs. In Framework

Progress Bar:
- Data Races vs. Idle vs. Loop vs. Worker Objects
Fundamental Tools Terminology

Threshold vs. Ceiling

Threshold: How hard to get started
Ceiling: How much can be achieved
These depend on what is being implemented

Path of Least Resistance

Tools influence what interfaces are created

Moving Targets

Changing needs make tools incomplete or obsolete

Myers et al, 2000
http://dx.doi.org/10.1145/344949.344959
Tools and Interfaces

Why Interface Tools?
Case Study of Model-View-Controller
Case Study of Animation
Sapir-Whorf Hypothesis
Thoughtfulness in Tools
Case Study in Self-Tracking
Model-View-Controller

How to organize the code of an interface?

This is a surprisingly complicated question, with unstated assumptions requiring significant background to understand and resolve.
Seeheim Model

Results from 1985 workshop on user interface management systems, driven by goals of portability and modifiability, based in separating the interface from application functionality

Buxton, 1983
http://dx.doi.org/10.1145/988584.988586
Seeheim Model

Lexical - Presentation
External presentation of interface
Generates the display, receive input

Syntactic - Dialog Control
Parsing of tokens into syntax
Maintain state

Semantic - Application Interface Model
Defines interaction between interface and rest of software

e.g., “add” vs. “append” vs. “^a” vs.
e.g., how to make a “menu” or “button”
e.g., three-state model, interface modes

e.g., drag-and-drop target highlighting
Seeheim Model

Lexical

Presentation

Syntactic

Dialogue Control

Semantic

Application Interface Model

USER ↔ APPLICATION
Seeheim Model
Seeheim Model

Rapid Semantic Feedback

In practice, all of the code goes in here
Model-View-Controller

Introduced by Smalltalk developers at PARC
Partitions application to be scalable, maintainable
View / Controller Relationship

In theory:

Pattern of behavior in response to input events (i.e., concerns of the controller) are independent of visual geometry (i.e., concerns of the view)

Controller contacts view to interpret what input events mean in context of a view (e.g., selection)
View / Controller Relationship

In practice:

View and controller often tightly intertwined, almost always occur in matched pairs

Many architectures combine into a single class
Model-View-Controller

MVC separates concerns and scales better than global variables or putting everything together

Separation eases maintenance
  Can add new fields to model, new views can leverage, old views will still work
  Can replace model without changing views

Separation of “business logic” can require care
  May help to think of model as the client model
Model-View-Collection on the Web

Core ideas manifest differently according to needs

For example, backbone.js implements client views of models, with REST API calls to web server

Web tools often implement views as templates
Model View View-Model

Design to support data-binding by minimizing functionality in view

Also allows greater separation of expertise
Tools and Interfaces

Why Interface Tools?
Case Study of Model-View-Controller
Case Study of Animation
Sapir-Whorf Hypothesis
Thoughtfulness in Tools
Case Study in Self-Tracking
Luxor Jr.
Luxor Jr.
Animation Case Study

Principles of Traditional Animation Applied to 3D Computer Animation

Lasseter, 1987

http://dx.doi.org/10.1145/37402.37407
Squash and Stretch
Squash and Stretch
Squash and Stretch

FIGURE 4a. In slow action, an object's position overlaps from frame to frame which gives the action a smooth appearance to the eye.

FIGURE 4b. Strobing occurs in a faster action when the object's positions do not overlap and the eye perceives separate images.

FIGURE 4c. Stretching the object so that its positions overlap again will relieve the strobing effect.
Timing

Just two drawings of a head, the first showing it leaning toward the right shoulder and the second with it over on the left and its chin slightly raised, can be made to communicate a multitude of ideas, depending entirely on the Timing used. Each inbetween drawing added between these two "extremes" gives a new meaning to the action.

NO inbetweens........ The Character has been hit by a tremendous force. His head is nearly snapped off.

ONE inbetweens........ The Character has been hit by a brick, rolling pin, frying pan.

TWO inbetweens........ The Character has a nervous tic, a muscle spasm, an uncontrollable twitch.

THREE inbetweens..... The Character is dodging a brick, rolling pin, frying pan.
Timing

FOUR inbetweens.......... The Character is giving a crisp order, "Get going!" "Move it!"

FIVE inbetweens.......... The Character is more friendly, "Over here." "Come on-hurry!"

SIX inbetweens.......... The Character sees a good looking girl, or the sports car he has always wanted.

SEVEN inbetweens.......... The Character tries to get a better look at something.
Timing

EIGHT inbetweens........... The Character searches for the peanut butter on the kitchen shelf.

NINE inbetweens........... The Character appraises, considering thoughtfully.

TEN inbetweens.......... The Character stretches a sore muscle.
Anticipation
Staging

FIGURE 6. Andre's scratch was staged to the side (in "silhouette") for clarity and because that is where his itch was.
Staging

FIGURES 7-8. In Luxo Jr., all action was staged to the side for clarity.
Follow Through, Overlap, Secondary
Pose-to-Pose, Slow In, Slow Out

Objects with mass must accelerate and decelerate. Interesting frames are typically at ends, tweaks perception to emphasize these poses.

FIGURE 9. Timing chart for ball bounce.
Arcs
Luxor Jr.
Luxor Jr.
Animation: From Cartoons to the User Interface

Chang and Ungar, 1993

http://dx.doi.org/10.1145/168642.168647
Frames Three Principles

Solidity
- Desktop objects should appear to be solid objects

Exaggeration
- Exaggerate physical actions to enhance perception

Reinforcement
- Use effects to drive home feeling of reality
Solidity: Motion Blur
Solidity: Arrival and Departure
Exaggeration: Anticipation

Figure 7. Objects anticipate major actions with a quick contrary motion that draws the user eye to the object in preparation for the main motion to come.
Figure 8. Objects ease out of their beginning poses and ease into their final poses. Although these motions are slower than that during the main portion of the movement, they are still quite fast.
Reinforcement: Arcs

Figure 9. When objects travel under their own power (non-interactively), they move in arcs rather than straight lines.
Figure 10. When objects come to a stop after moving on their own, they exhibit follow through in the form of wiggling back and forth quickly. This is just suggested by the “wiggle lines” in the figure—in actuality, the object moves back and forth, with motion blur.
Animation Support in a User Interface Toolkit: Flexible, Robust, and Reusable Abstractions

Hudson and Stasko, 1993

http://dx.doi.org/10.1145/168642.168648
Events and Animation

Figure 5. Animation Event Translation and Dispatch
Not Just an Implementation

Provides tool abstractions for implementing previously presented styles of animation

Overcomes a fundamental clash of approaches

Event loop receives input, processes, repaints

Animations expect careful control of frames, but the event loop has variable timing
Events and Animation

Figure 5. Animation Event Translation and Dispatch
Transition Object

Figure 3. Parts of a Transition Object
Pacing Function

Figure 4. Two Example Pacing Functions
Computing a Frame

Figure 8. Translation from Time to Space
Animation Case Study

Based on increased understanding of how animation should be done in the interface, increasingly mature tools develop

Now built into major commercial toolkits (e.g., Microsoft’s WPF, JavaFX, jQuery)

Once mature, begins to be used as a building block in even more complex behaviors
Animation Case Study

The Kinetic Typography Engine: An Extensible System for Animating Expressive Text

Lee et al, 2002
http://dx.doi.org/10.1145/571985.571997
Kinetic Typography Engine
Kinetic Typography Engine

Kinetic Typography
Johnny Lee, Jodi Forlizzi, Scott Hudson
Carnegie Mellon University
Human-Computer Interaction Institute
2002
Kinetic Typography Engine

Goals of Kinetic Type

- Emotional content
- Creation of characters
- Direction of attention

Animation Composition

Figure 6. Waveform addition by chaining

Figure 7. Waveform scaling by functional composition with amplitude
Tools and Interfaces

Why Interface Tools?
Case Study of Model-View-Controller
Case Study of Animation
Sapir-Whorf Hypothesis
Thoughtfulness in Tools
Case Study in Self-Tracking
Sapir-Whorf Hypothesis

Roughly, some thoughts in one language cannot be stated or understood in another language.

Our tools define the language of interaction

Beyond the simple matter of code
Frame how we think about possibilities.

Sapir-Whorf Hypothesis

Roughly, some thoughts in one language cannot be stated or understood in another language.

Language is not simply a way of voicing ideas, but is the very thing which shapes those ideas.

Our tools define the language of interaction.

Beyond the simple matter of code.
Frame how we think about possibilities.

You must be aware of this when choosing tools, designing applications, and creating new tools.
Animation Case Study

Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects

Baudisch et al, 2006

http://dx.doi.org/10.1145/1166253.1166280
Phosphor

Animation can help people follow interface transitions

But the right speed is crucial
    Too fast increases error rate
    Too slow increases task time

The right speed depends on familiarity, distraction, etc.
    It cannot be determined
Phosphor

Phosphor shows the outcome immediately, then explains change in retrospect using a diagrammatic depiction.
Phosphor

phosphor
Phosphor

phosphor
Challenging Assumptions of Tools

Phosphor breaks from the assumptions that have evolved into current transition tools.
Prefab

Prefab uses pixel analysis to modify existing applications from the outside, using only pixels.

Prefab is informed by how toolkits work, but not linked to any particular toolkit implementation.

Allows trying and fielding new ideas that are not supported by existing applications or toolkits.
Prefab
Prefab
Understanding Tools

Tools promote and encapsulate proven practices
- Reduce expertise barriers
- Enable more rapid and iterative implementation

Codification eventually constrains design
- Inevitable consequence of codification versus evolving understanding of emerging technologies

Codification goes deeper than the code
- Frames how we think about our applications

Rebuilding the Language

We regularly rebuild the entire system

Command Line, Text Screens
Multiple Generations of Desktop
Multiple Generations of Web
Mobile Apps

We will do it again

Several near-term challenges require it
e.g., Touch, Cloud, Distributed Interfaces

Backward compatibility helps, but is not required

Informing the Next Language

Research explores the next generation of language, while being limited by the current

We therefore conflate:

- Ideas
- Proof of Concept
- Engineering
- Implementation
- Broken Metaphors
- Unspeakably Dirty Hacks
Informing the Next Language

Research explores the next generation of language, while being limited by the current

We therefore conflate:

- Ideas
- Proof of Concept
- Engineering
- Implementation
- Broken Metaphors
- Unspeakably Dirty Hacks

Prefab is not just about ‘do everything with pixels’, but about exploring new possibilities in the current ecosystem of interface tools.
Mobile Phones as Pagers

Our notion of technology design for journals / ESM / EMA has been anchored by papers journals and pager-based reminders.

Unlock Journaling for Self-Report

Zhang, Pina, Fogarty. Examining Unlock Journaling with Diaries and Reminders ... CHI 2016.
Unlock Journaling for Self-Report

Unlock Journaling for Self-Report

Pleasure and Accomplishment (e.g., self-monitoring depressive symptoms)

Lejuez, Hopko, Acierno, Daughters, Pagoto. ... Behavioral Activation Treatment for Depression ... Behav Modif 2011.
Unlock Journaling for Self-Report

Russell’s Affect Grid

Unlock Journaling vs. Notifications

Unlock journaling is:

rated less intrusive
(1.77 vs. 2.22 on a 5-point scale)

yields greater frequency
(15.0 vs. 9.8 per 12-hour day)

comparable timeliness
(8.6 vs. 9.3 minutes)

Instead of reminders to journal, unlock journaling makes the opportunity visible, easy, and optional

It should not have taken 10 years to get here
Mobile Food Journals

Origins in daily recall

Self-monitoring of food can support many goals

Weight Loss
Diabetes Management
Trigger Identification

High burdens detract from potential benefit, data is often wrong

Mobile Food Journals

Mobile devices provide real-time feedback

Search for each food in a large database, often breaking into components

Typically provide calorie-based feedback

High burdens detract from potential benefit, data is often wrong

Tsai, Lee, Raab, Norman, Sohn, Griswold, Patrick, K. Usability and Feasibility of PmEB ... Mobile Netw Appl 2007.
Perceptions of Healthy Eating

“What does healthy eating look like to you?”

Food types:
“vegetables”
“fruits”
“protein”

Food qualities:
“low processed”
“organic”
“fresh”

Diet qualities:
“balanced”
“variety”
“portion”

Cordeiro, Bales, Cherry, Fogarty. Rethinking the Mobile Food Journal … CHI 2015.
Difficulty as a Negative Nudge

“I just avoided eating things that were hard to log” – SP132

“Prepackaged meals were the easiest because of bar codes but those aren’t healthy” – SP123

“I could make life easier by eating the same things regularly” – SP97

“It discourages you from eating out or at a friend’s, even if it is healthy” – SP42
Deploying a Photo-Based Journal

Mobile capture and review

Web review and annotation

Cordeiro, Bales, Cherry, Fogarty. Rethinking the Mobile Food Journal … CHI 2015.
Leveling the Difficulty of Journaling

With prior techniques:

60% report not journaling because it was too difficult
65% report not journaling because they did not know

With photo-based capture:

22% report not journaling because it was too difficult
None report not journaling due to food knowledge

“For some meals, it’s just really easy to take a picture … than sit there and type in every ingredient” – FP20
Journaling without Judgment

With prior journals, participants report choosing not to journal because they would exceed a calorie budget or because a food was unhealthy

- 13% of survey participants
- 45% of field participants

Photos enable mindfulness while avoiding judgment

“[it was] easier because there were no calorie counts, no judgments, but still makes you aware” – FP14

“So I really want to eat this? I’m capturing this” – FP17
Triggers and Trends

“I eat too much pizza” – FP10

“I’m surprised at how many times I’m seeing things that I consider an exception to my diet!” – FP4

“I don’t branch out as much as I thought I did, even when I go somewhere new, I kind of get what I always get somewhere else” – FP10
Food Journals as Daily Recall

“it should be noted that much of the use of food journaling is in a more clinical setting with the purpose being sharing and evaluating the journal with nutritionists and care providers …

it’s not relevant if photos are more or less easily understood by the user if a nutritionist is the eventual consumer of the data”

– Actual Anonymous Grumpy R3
Tools and Interfaces

Tools embody expertise and assumptions

Tools evolve based on emerging understanding of how to address categories of problems

Be conscious of your tool decisions

Try to think about designs before tying to a tool
Choose good and appropriate tools
Understand what you are getting in a tool
Push yourself to think outside the tool
CSE 440: Introduction to HCI
User Interface Design, Prototyping, and Evaluation

Lecture 13:
Interface Implementation

Tuesday / Thursday
12:00 to 1:20

James Fogarty
Kailey Chan
Dhruv Jain
Nigini Oliveira
Chris Seeds
Jihoon Suh