

Review: User Interface Components

Each component is an object with

Bounding box

Paint method for drawing itself

Drawn in the component's coordinate system

Callbacks to process input events

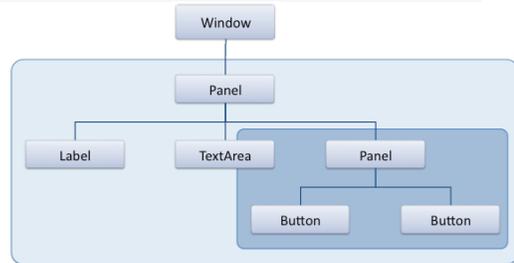
Mouse clicks, typed keys



```
Java:
public void paint(Graphics g) {
    g.fillRect(...); // interior
    g.drawString(...); // label
    g.drawRect(...); // outline
}
```

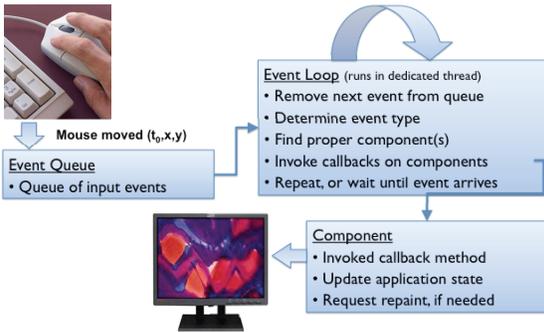
```
Cocoa:
(void)drawRect:(NSRect)rect
```

Review: Layout & Containment



Principle: Each container is responsible for allocating space and positioning its contents.

Review: Event Dispatch Loop



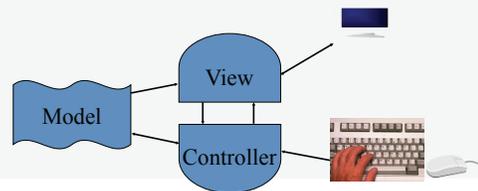
Review: Model-View-Controller

Architecture for interactive apps

Model: Info the application manipulates

View: Visual display of the model

Controller: Receives input & decides what they do



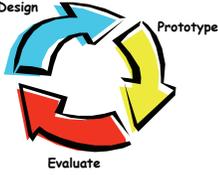
Topics

- Paper Prototyping
- Video Prototyping
- Wizard of Oz Testing

Paper Prototyping

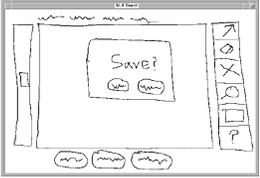
Why Do We Prototype?

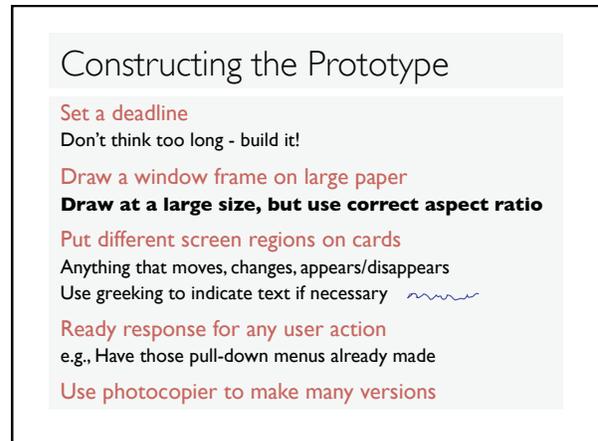
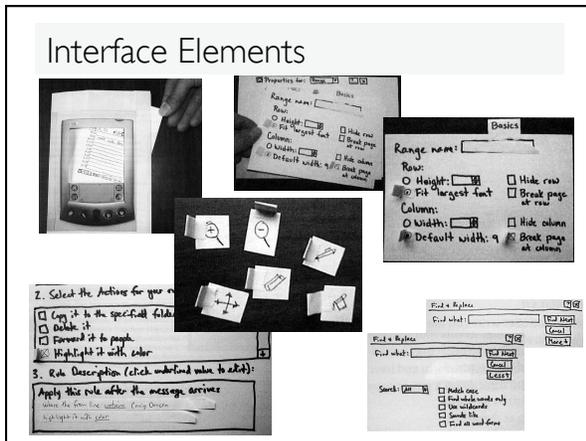
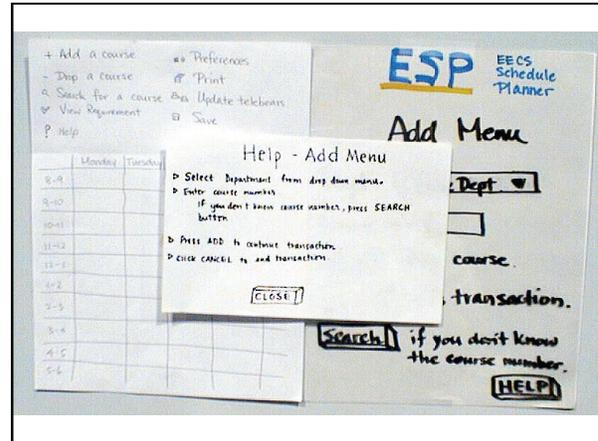
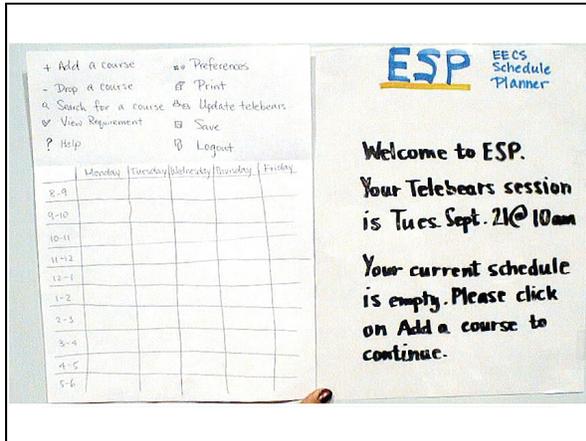
- Get feedback on our design faster
 - Saves money
- Experiment with alternative designs
- Fix problems before code is written
- Keep the design centered on the user



Fidelity in Prototyping

- Fidelity refers to the level of detail
- High fidelity:
 - Prototypes look like the final product
- Low fidelity:
 - Artists renditions with many details missing



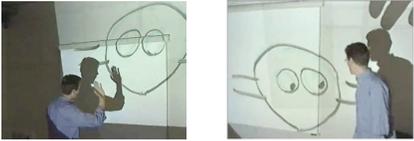


Video Prototyping

Video Brainstorming

Participants act ideas out in front of a video camera
 Goal is to create as many new ideas as possible
 each should take 2-5 minutes to generate & capture
 run standard brainstorming session first for ideas

Advantages
 video easier to understand later than notes
 participants actively experience interaction & preserve record of the idea

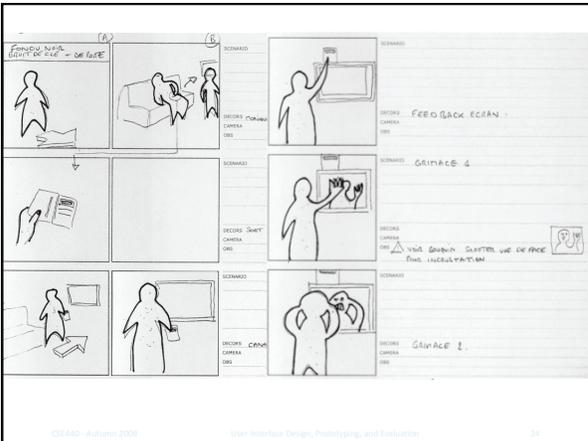


Video brainstorming of an animated character in *Prototyping Tools & Techniques* by Beaudouin-Lafon & Mackay. Character follows user with its eyes.

Forms of Video Prototypes

Build on paper prototypes
 Use existing software & images of real settings
 Narration optional (but required for your assignment!)
 Explain events while others move images/illustrate interaction

With good storyboards, should be able to create video prototype in 1 hour



Creating a Video Prototype

- 1) Review field data about users & work practices
- 2) Review ideas from video brainstorm
- 3) Create use scenario in words
- 4) Develop storyboard of each action/event with annotations explaining the scene. Put each element on a card.
- 5) Shoot a video clip for each storyboard card
 - Avoid editing in the camera – just shoot in storyboard order
 - Hold last frame of a section/shot for 1s
- 6) Use title cards to separate clips (keep it onscreen for 3s)
 - if you make an error; rewind to last title card & reshoot

CHI 440 - Autumn 2009

User Interface Design: Prototyping and Evaluation

25

Cluster

Andy Hou and Kevin Chiu: Univ. of Washington



Energy Usage Information

Lisa Seeman: Stanford



Tips & Tricks

Add structure to better explain context

- begin with a title
- follow with an "establishing shot"
- create series of closeup & mid-range shots, interspersed with title cards
- place a final card with credits at the end

Use colored paper title cards -- aids edit/search of video

Stop-motion lets images appear & disappear based on interaction

- e.g., illustrate pop-up menu by recording clip of user pressing button,

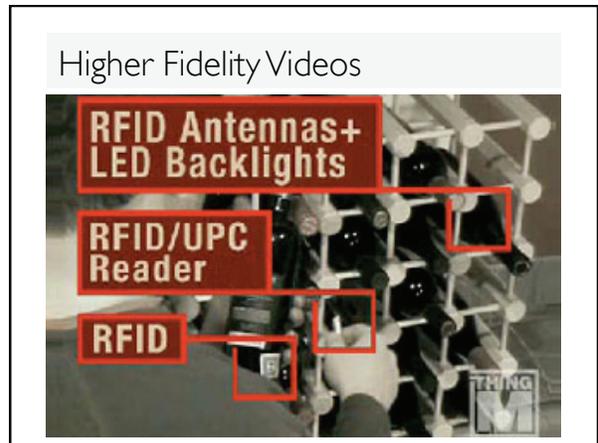
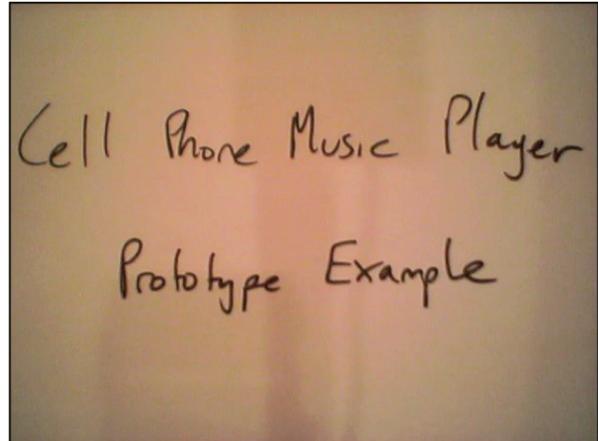
pause camera, add menu, restart camera

Be careful about taking video out of the original design setting for ethical reasons (context matters)

CHI 440 - Autumn 2009

User Interface Design: Prototyping and Evaluation

26





Wizard of Oz Prototype Testing

SIMS 213 Project: Telebeats redesign

Observer (or video camera)
User
Interface
"Computer"
Interface elements

A bit slow for a computer - but it works!

Conducting a Test

Three or Four testers (preferable)

- Greeter** - Puts users at ease & gets data
- Facilitator** - only team member who speaks
Gives instructions & encourages thoughts, opinions
- Computer** - knows application logic & controls it
Always simulates the response, w/o explanation
- Observer(s)** - Take notes & recommendations

Typical session should be approximately 1 hour
Preparation, the test, debriefing

Conducting a Test (cont.)

Greet

Get forms filled, assure confidentiality, etc.

Test

Facilitator explains how test will work

Performs a simple task

Facilitator hands written tasks to the user

Must be clear & detailed

Facilitator keeps getting "output" from participant

"What are you thinking right now?", "Think aloud"

Observers record what happens

Avoid strong reactions, frowning, laughing, impatience – biases the test

Designers should not lead participants

Let users figure things out themselves as much as possible

Only answer questions if user remains stuck for a long time

Conducting a Test (cont.)

Debrief

Fill out post-evaluation questionnaire

Ask questions about parts you saw problems on

Gather impressions

Give thanks

Preparing for a Test

Select your participants

Understand background of intended users

Use a questionnaire to get the people you need

Don't use friends or family

Prepare scenarios that are

Typical of the product during actual use

Make prototype support these (small, yet broad)

Practice running the computer to avoid "bugs"

You need every menu and dialog for the tasks

All widgets the user might press

Remember "help" and "cancel" buttons

WOZ is different from pre-built/canned functionality

Wizard of Oz Tips

Rehearse your actions

Make a flowchart which is hidden from the user

Make list of legal words for a speech interface

Stay "in role"

You are a computer, and have no common sense, or ability to understand spoken English.

Facilitator can remind user of the rules/think-aloud approach if the user gets stuck

Record Critical Incidents

Critical incidents are any unusual/interesting events

Most of them are usability problems.

They may also be moments when the user

- Got stuck

- Suddenly understood something

- Said "that's cool" etc.

Using the Results

Update task analysis and rethink design

- Rate severity & ease of fixing problems

- Fix both severe problems & make the easy fixes

Will thinking aloud give the right answers?

- Not always

- If you ask a question, people will always give an answer, even if it has nothing to do with the facts

- Try to avoid leading questions