

Designing Interactive Systems I

Prototyping

Prof. Dr. Jan Borchers
Media Computing Group
RWTH Aachen University

Winter Semester '25/'26

<https://hci.rwth-aachen.de/dis>



RWTHAACHEN
UNIVERSITY

Paper Prototypes

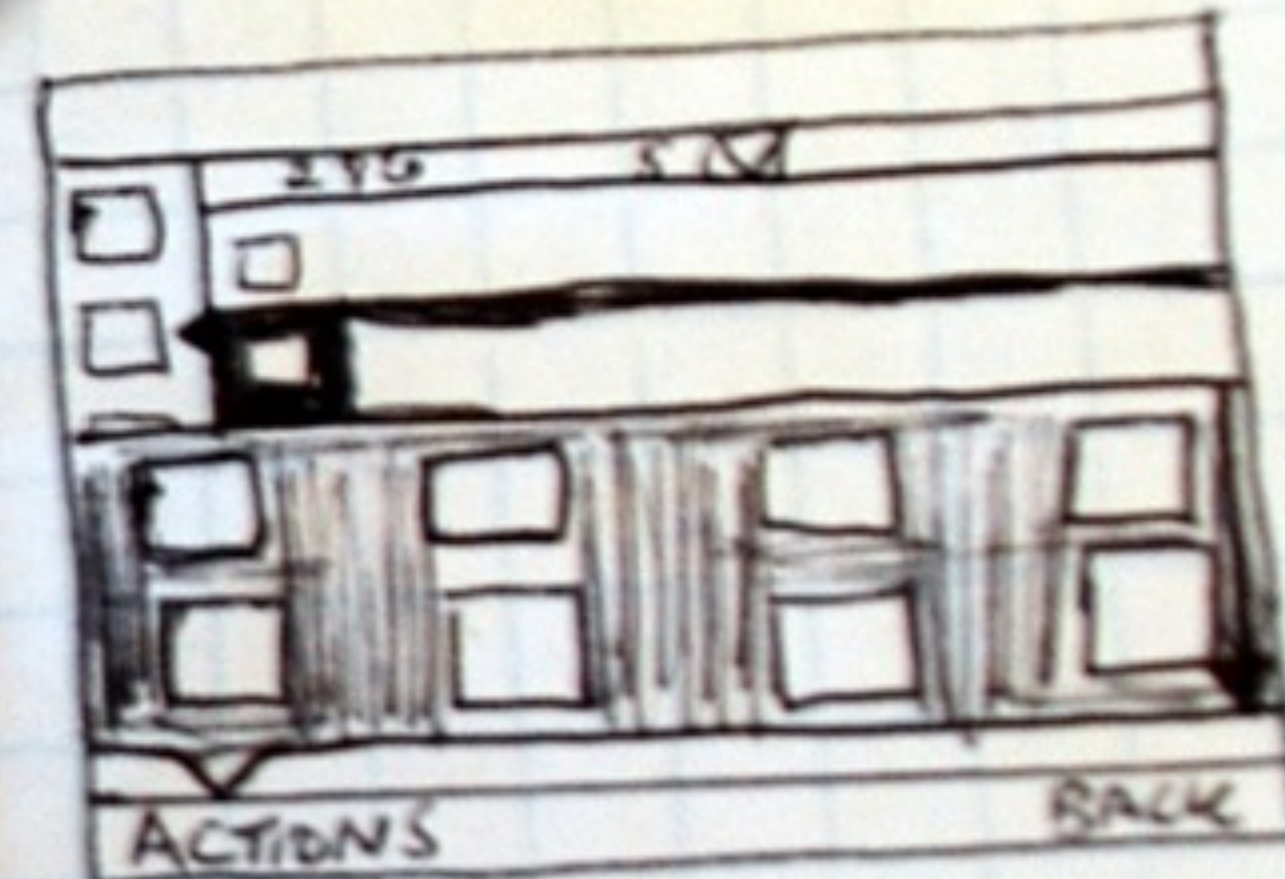
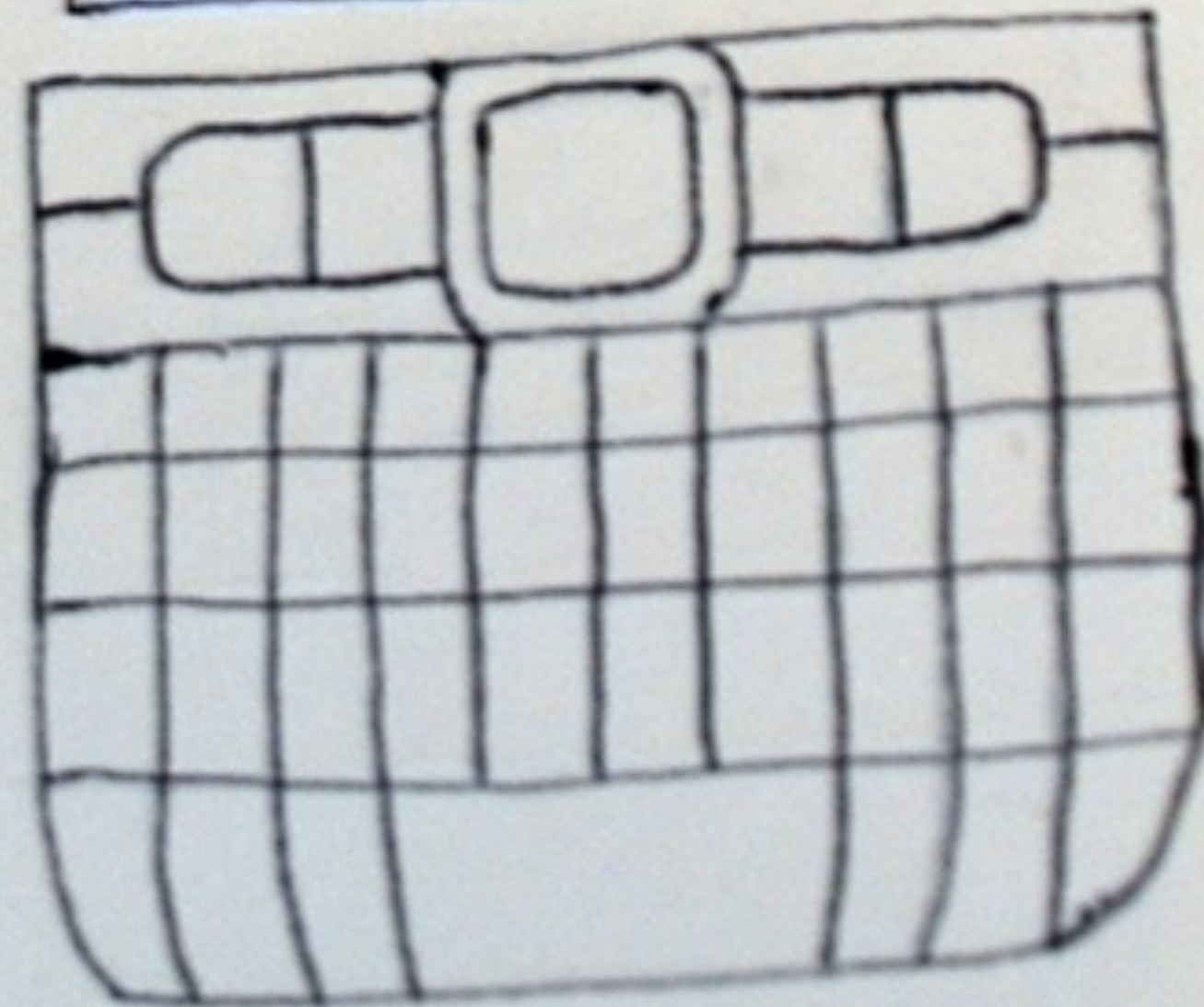
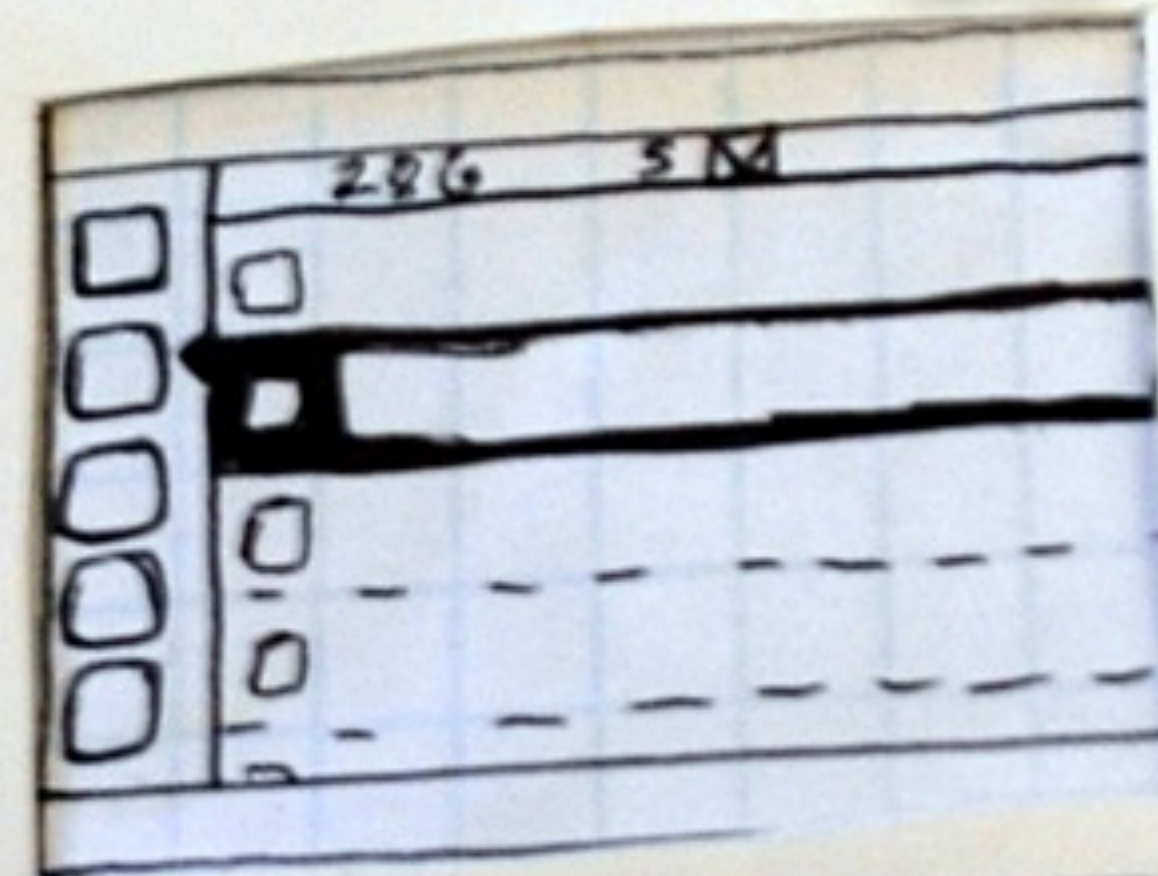
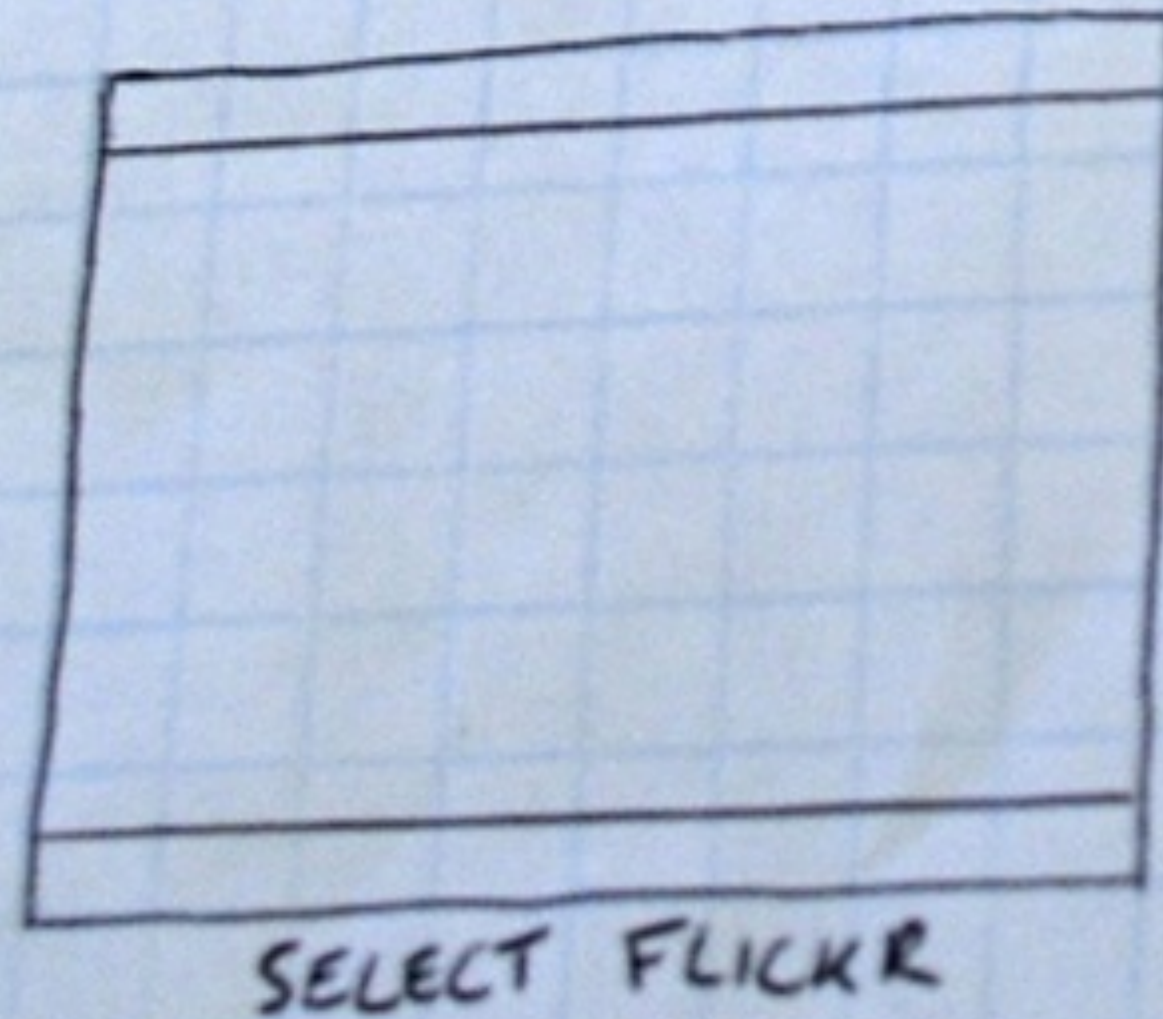
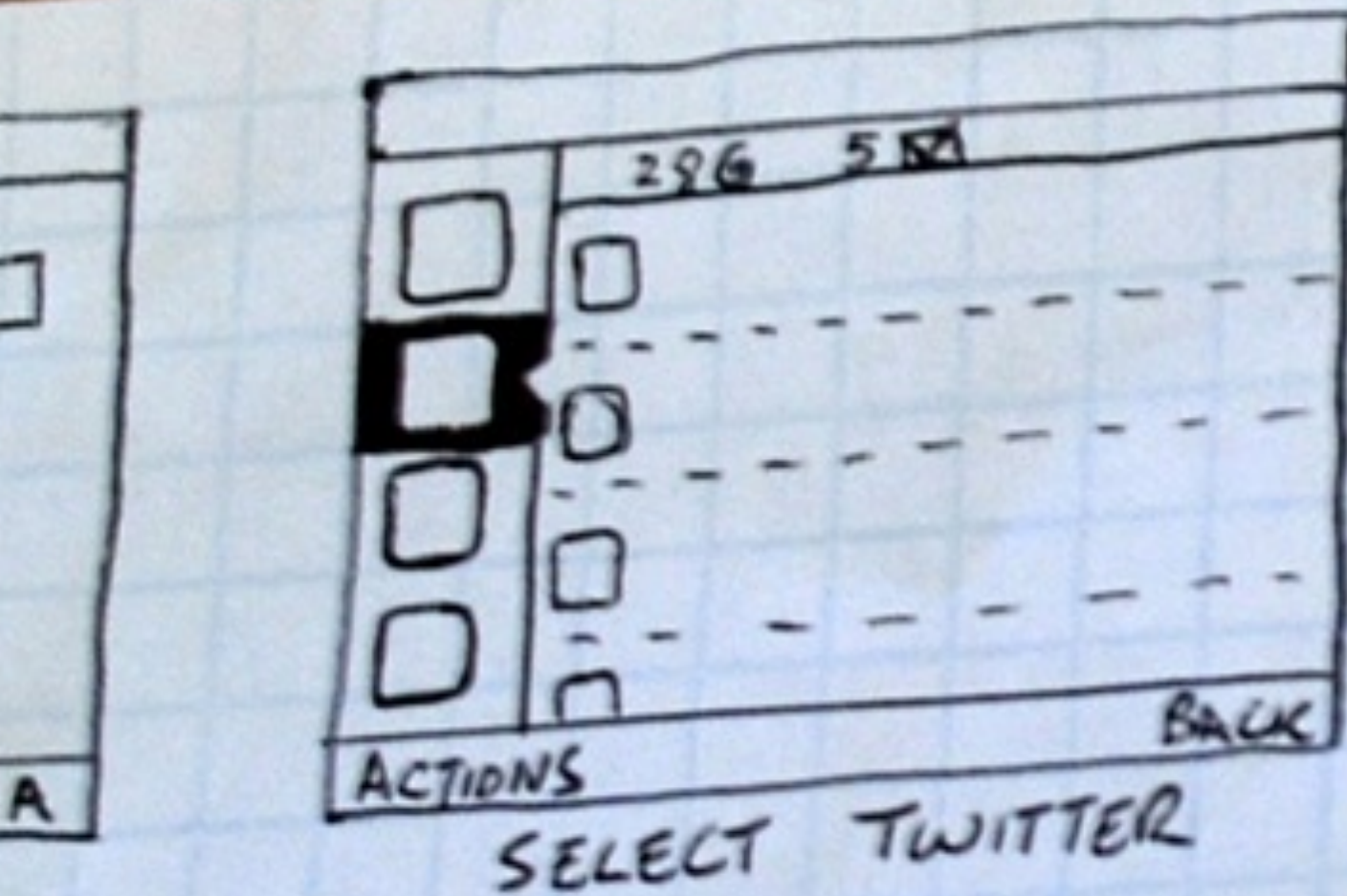
Paper Prototypes

- First prototypes, quick and cheap
- Rough paper & pencil sketches of interface or central UI dialogs
- Hand-drawn, no ruler, no computer!
- Pro: Not detailed, so designer and user focus on important **high-level** UI design
- Con:
 - Dialog sequence hard to convey unless you drive it yourself
 - Drawing many screens is a lot of work
- A storyboard can be your first paper prototype

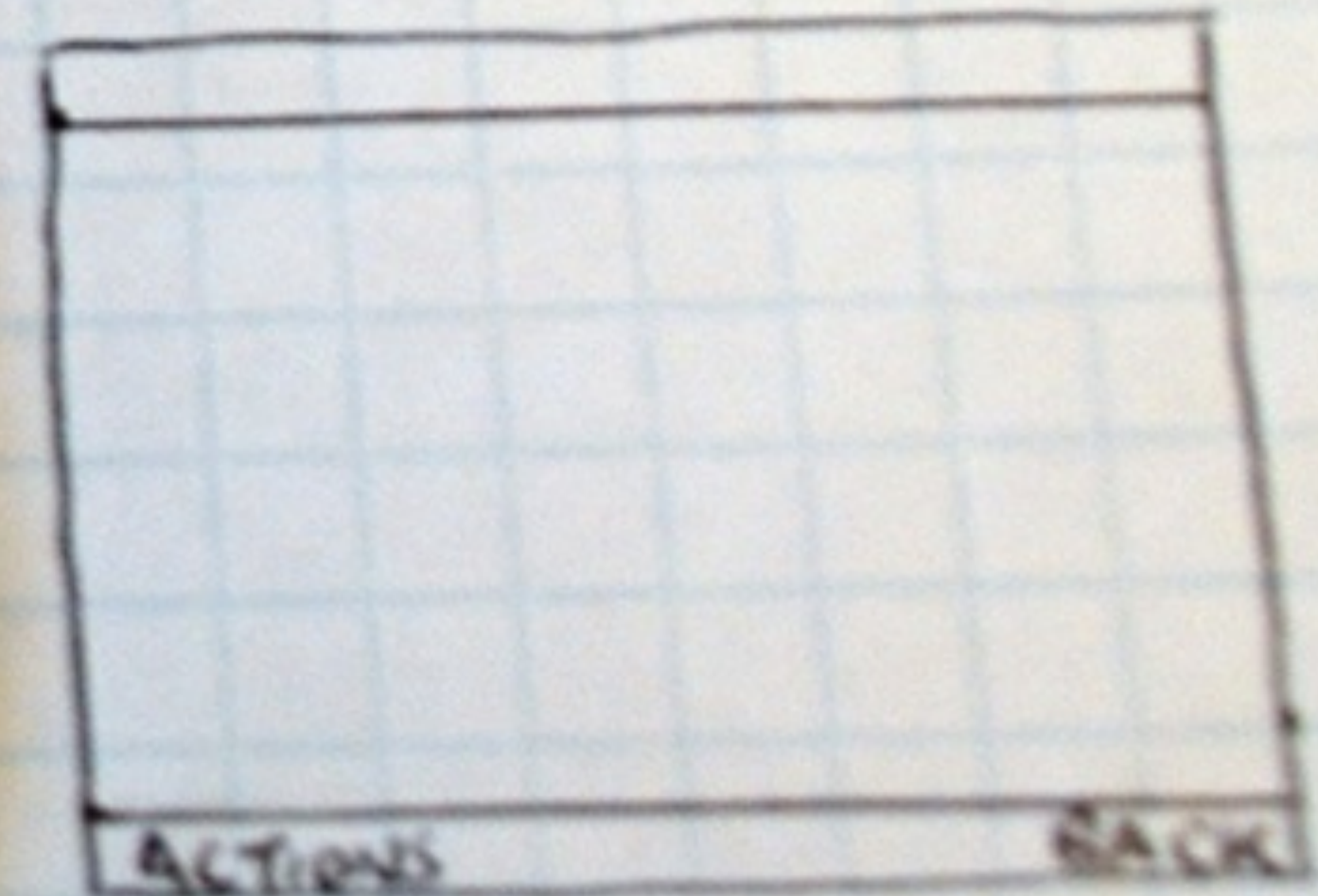
Paper Prototypes

- Type A: Storyboard-like
 - Put several frames with sketched snapshots of the UI on one page
 - Label each frame and each connection
 - Only allows you to show one fixed interaction sequence (scenario)
 - Like a storyboard, but only shows the UI (and maybe the user's hand), not the entire environment of the task
- Type B: Flipbook
 - Sketch each UI snapshot frame on separate page
 - Collect in a loosely bound flipbook that flips over easily
 - Usage: Show start screen page to user—he selects an action—turn to the resulting page from your flipbook, etc.
 - Allows you to simulate the UI for a user

Storyboard-like Prototypes



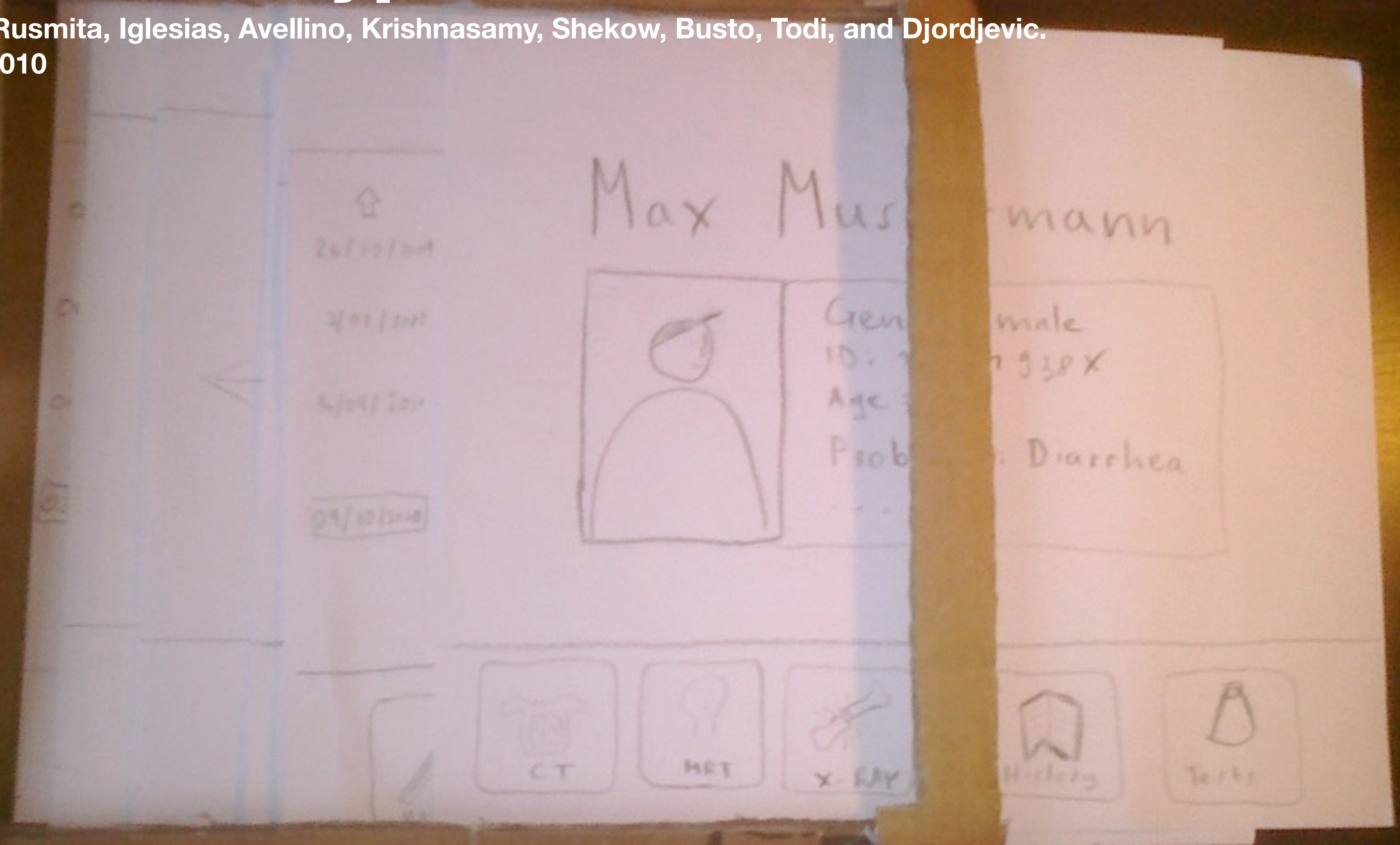
ACTIONS ON A TWEET



ACTIONS ON A FILTER

Flipbook Prototypes

Franzen, Ahmad, Rusmita, Iglesias, Avellino, Krishnasamy, Shekow, Busto, Todi, and Djordjevic.
DIS1 students in 2010



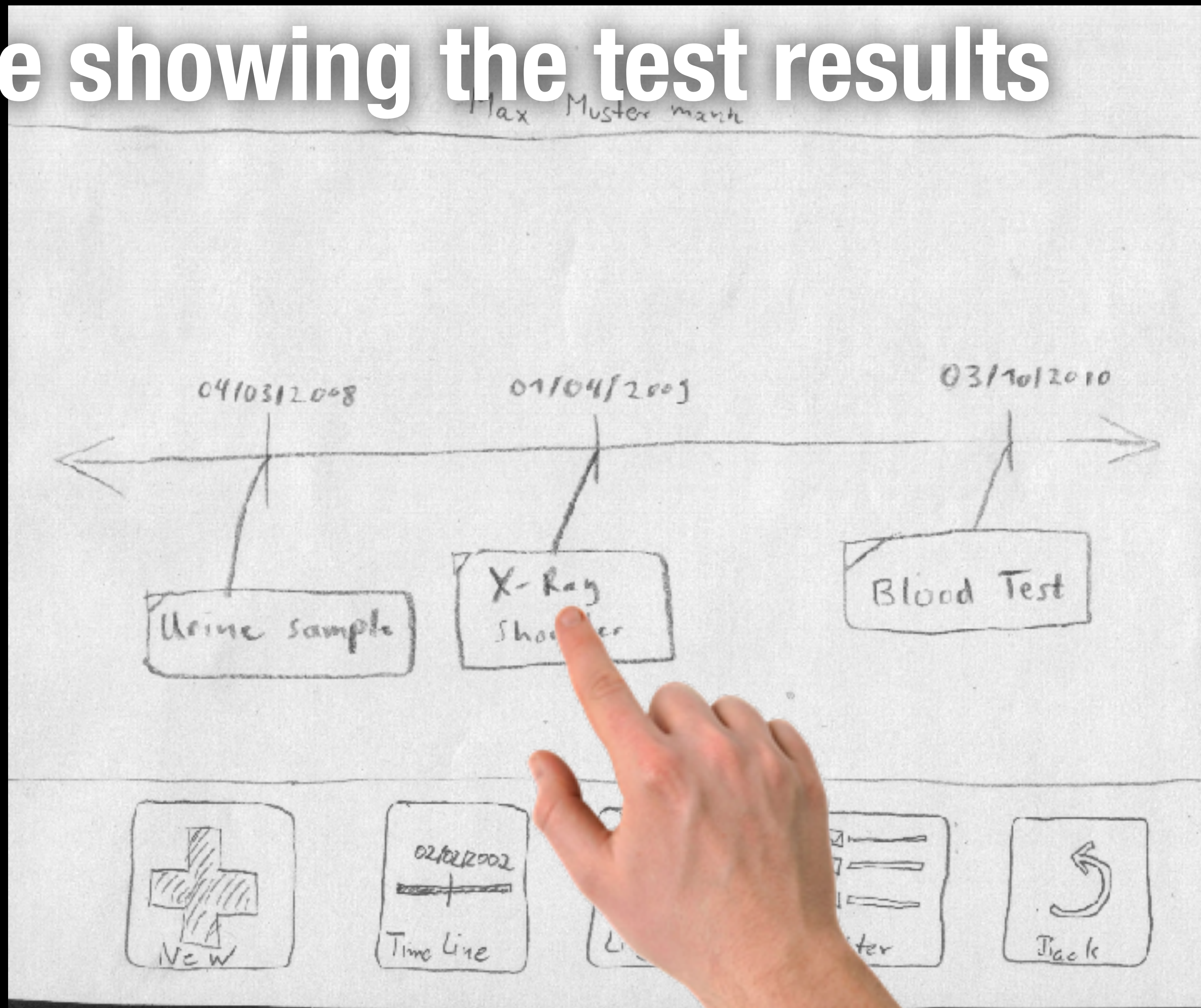
Patient overview

Max Mustermann

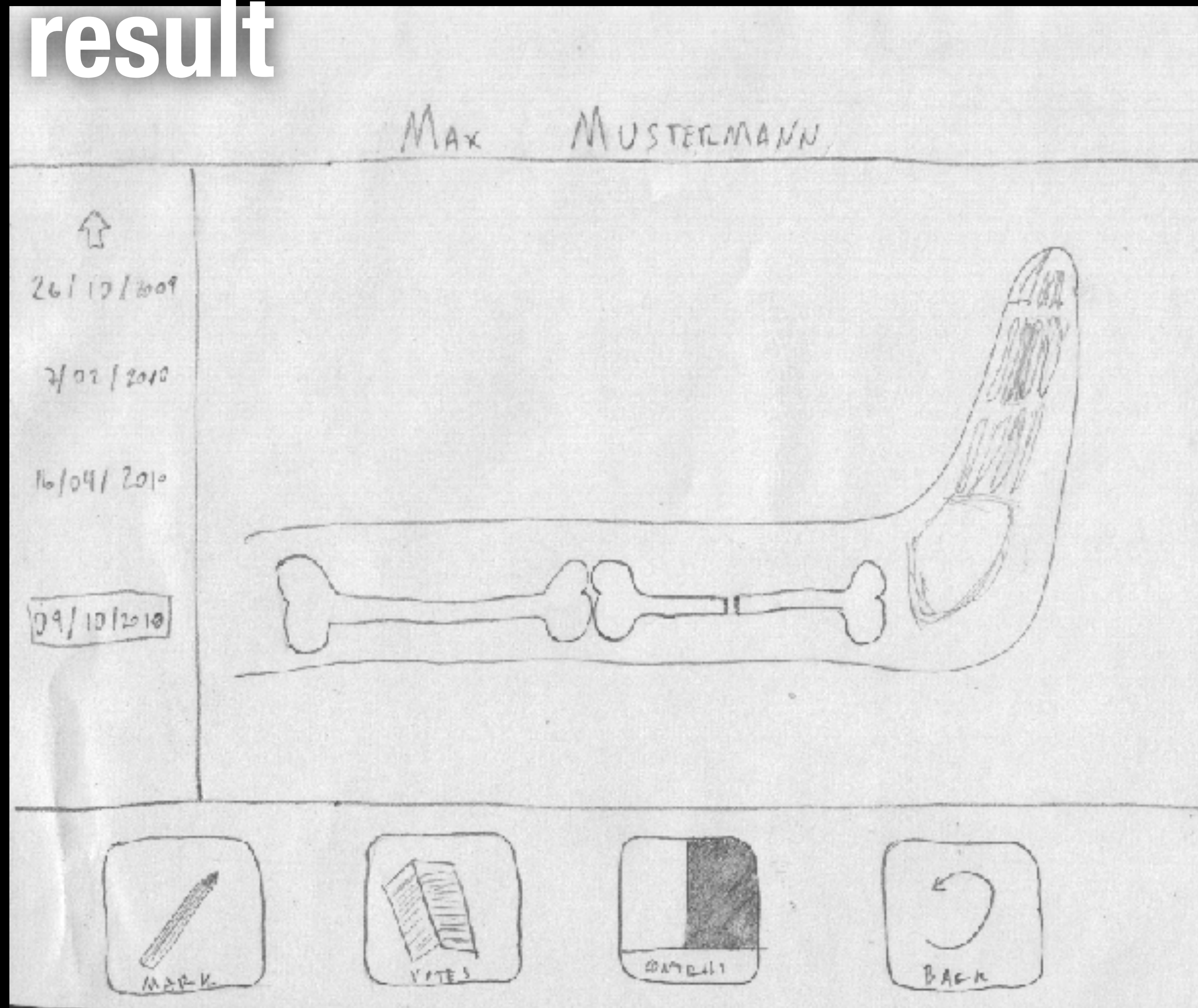
	Gender: male ID: 13527938X Age: 35 Problems: Diarrhea ...
---	--



Timeline showing the test results



Detailed result



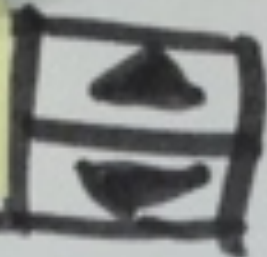
Post-It Prototypes

- More interactive paper prototype
- Dialogs, menus, windows on post-it notes in multiple layers
- Allows simulating opening dialogs, etc., by manipulating notes
- Quick to change by making new notes
- Tip: Create empty templates for dialog objects, then fill in
- Tip: Videotape user session for later analysis

Radio Buttons/Checkboxes

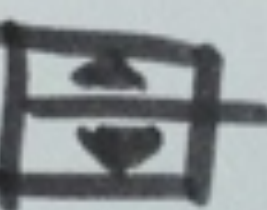
Range Name

Row:

☐ Height: 

☐ Fit largest font

Column:

☐ Width: 

☒ Default width: 9

☐ Hide row

☐ Break page
at row

☐ Hide column

☒ Break
page at
column

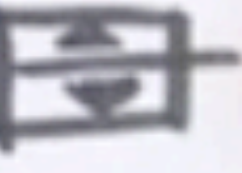
Range Name [PrototypeRange]

Row:

○ Height: [5] []

○ Fit largest font

Column:

○ Width: [] []

● Default width: 9

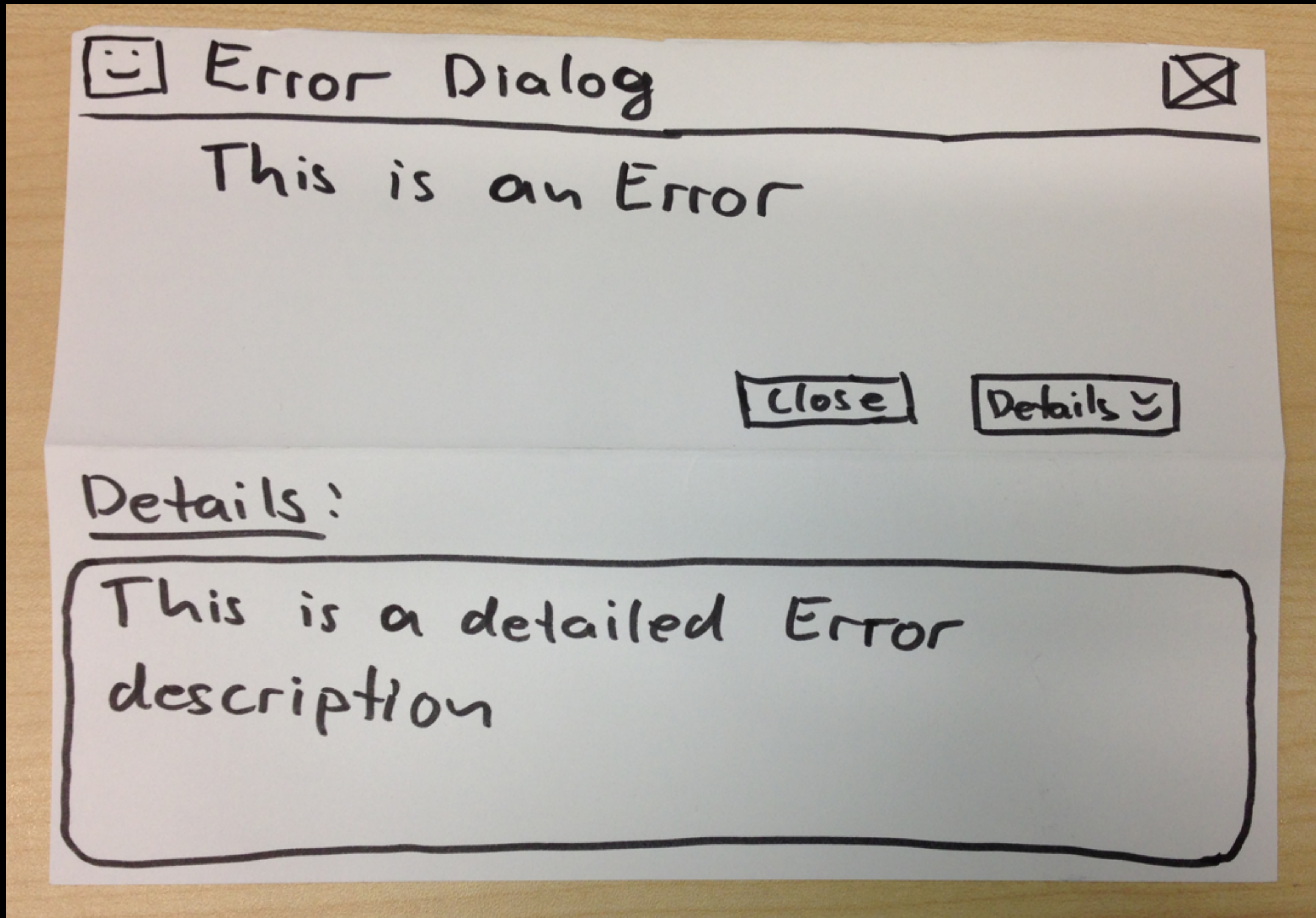
☐ Hide row

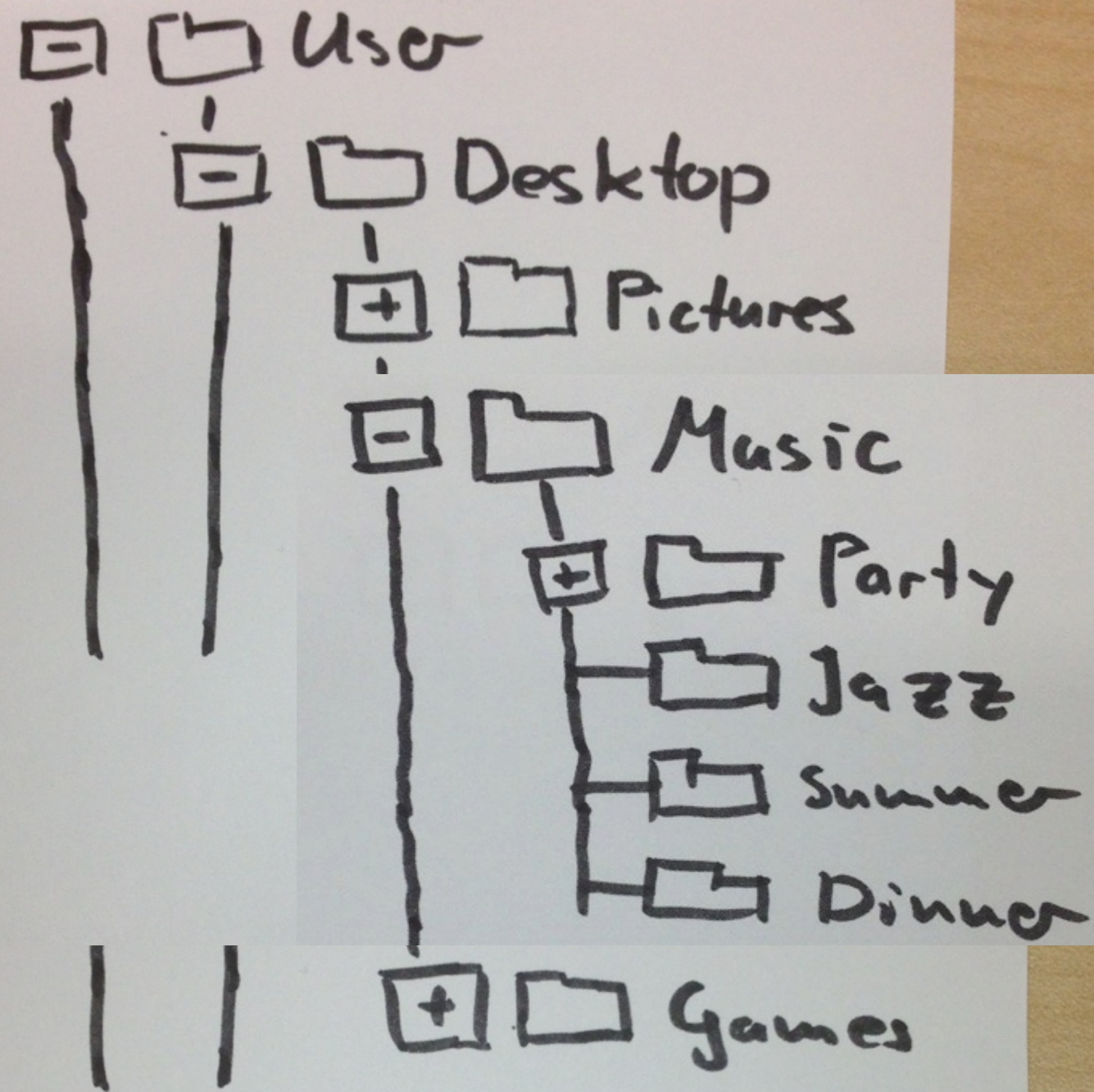
☐ Break page
at row

☒ Hide column

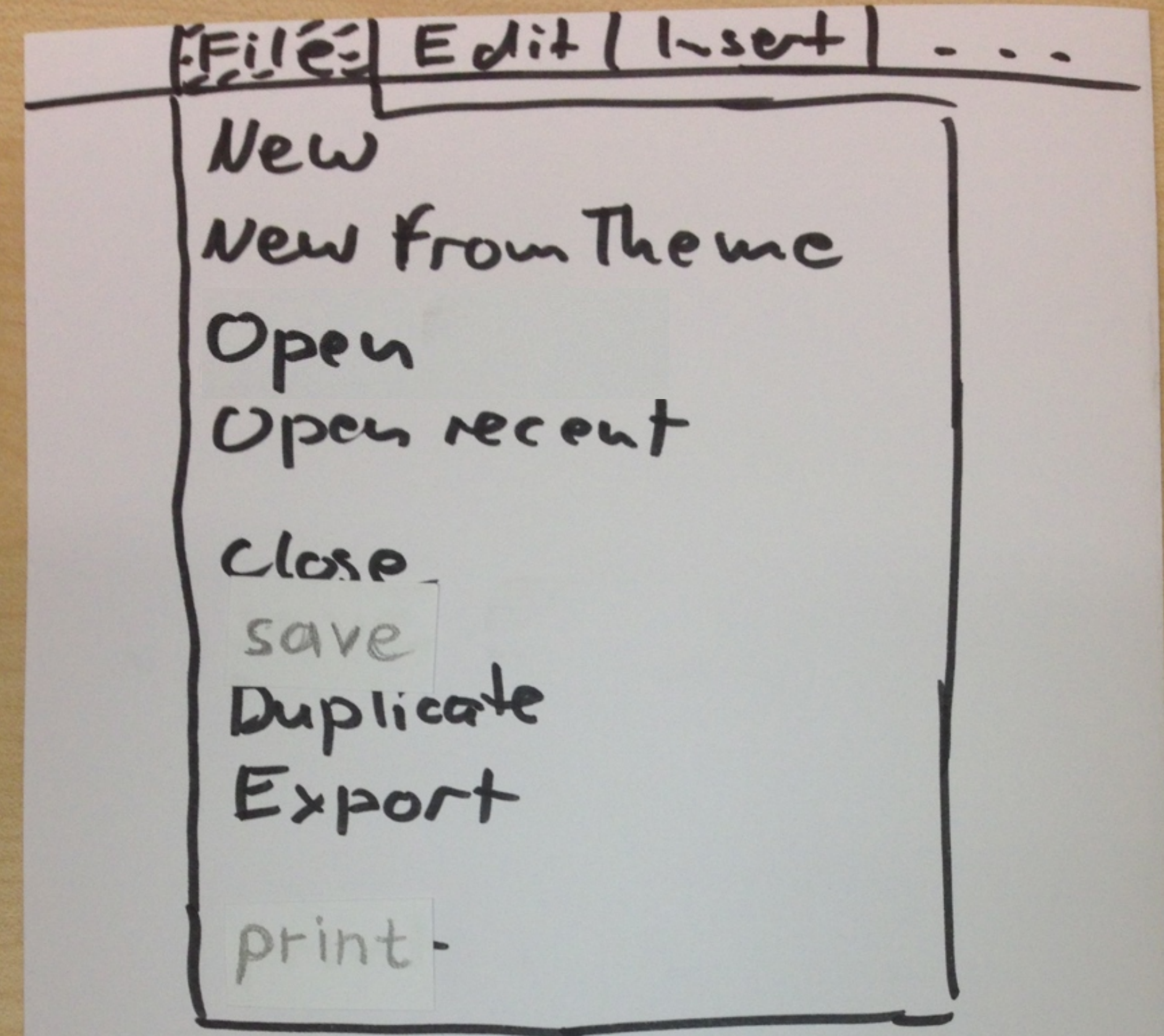
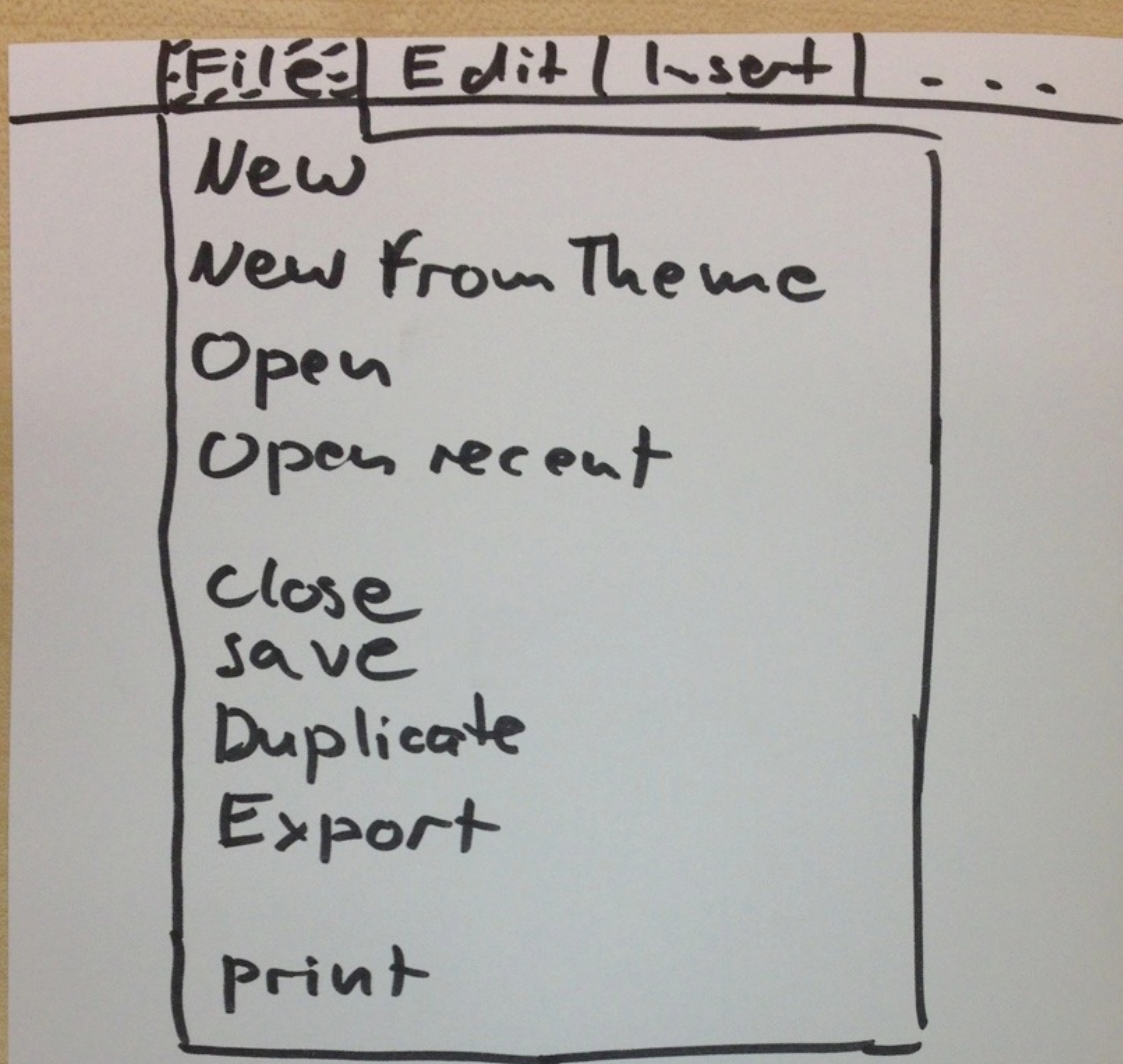
☐ Break
page at
column

Expandable Dialog Boxes





Expandable Lists



Disabled (“grayed-out”) Controls

Images: Paper Prototyping by Carolyn Snyder, 2003

You are logged
in as

JOE

Username: -----

Password: -----

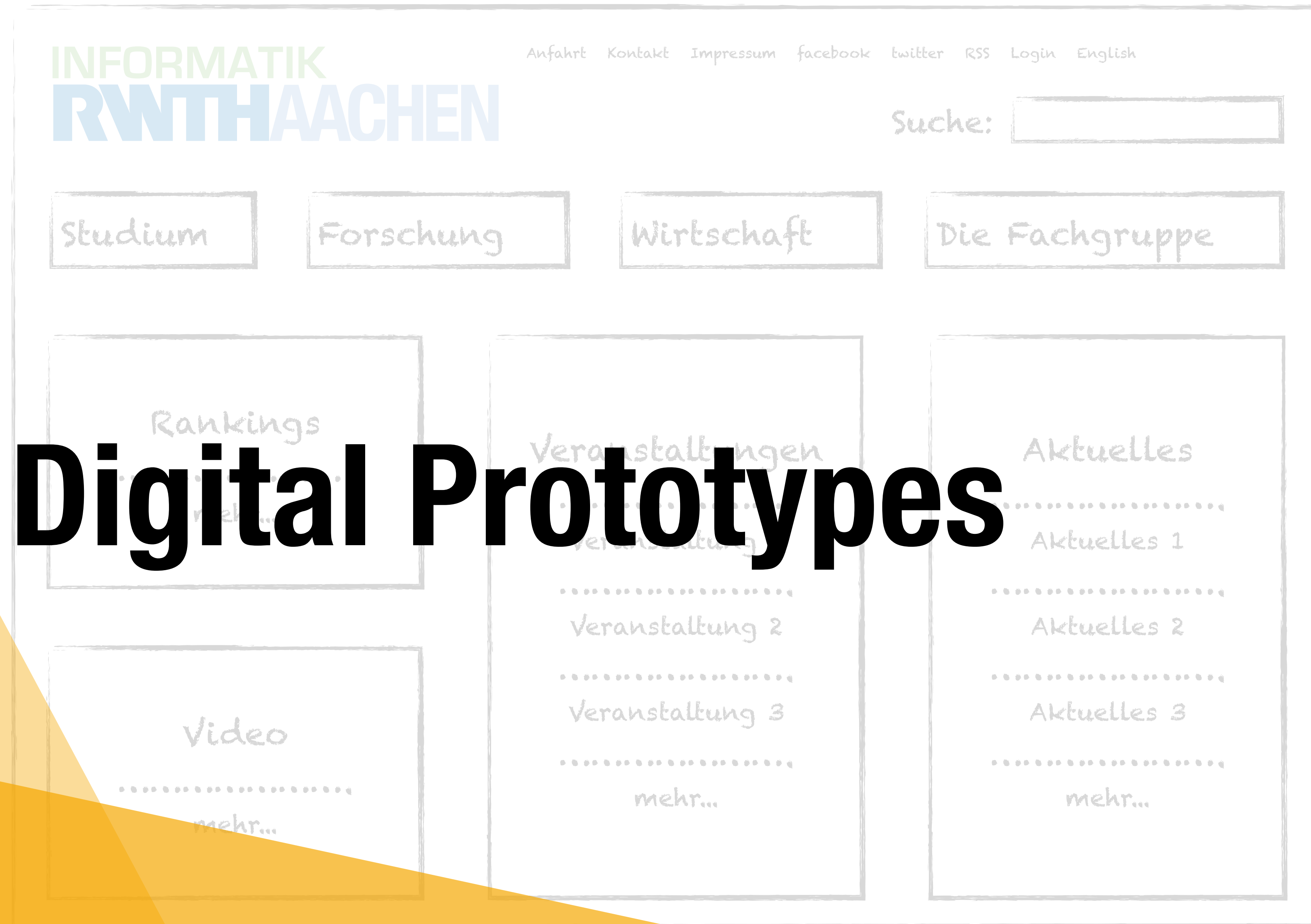
ENTER

I am a ☒ Rightie ☐ Leftie



Simulating Touchscreen UI with Paper Prototype

Kaiser, Dieckert. DIS1 students in 2010

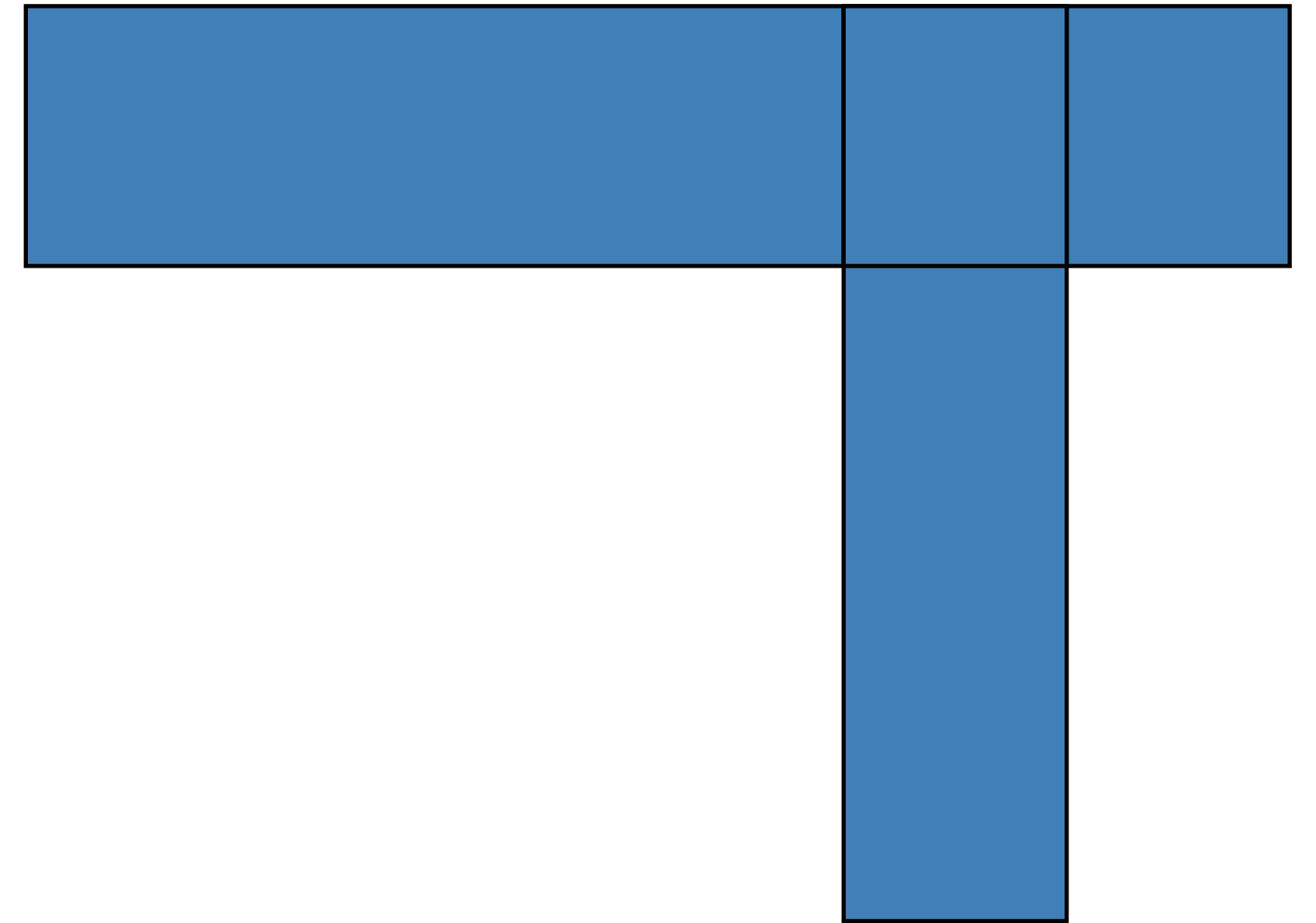


Digital Prototypes

- Medium fidelity prototype
 - More detailed, more precise, interactive
 - Create only after initial, simpler (paper) prototypes!
- Mock-up (model, illusion) of some (but not all) aspects of the final UI
- Example: Powerpoint prototype
- Important: UI, not functionality, is key!
- Pro: More engaging for user to try, user can play with it without designer around

How to Limit Prototypes

- Horizontal prototype
 - Entire UI visible, but no functionality
 - Simulate each interaction step (nothing “works”)
- Vertical prototype
 - Few functions, but those implemented in detail
 - Allows testing general design ideas by example
- Scenario
 - Combination of horizontal and vertical prototype
 - Script simulates only fixed interaction paths



Digital Prototyping: Screenshots

- Photoshop, PowerPoint, etc.
- Draw screens / UI storyboards
- Thin horizontal prototype
- Easier to change than hand drawings
- Allows for visual detail and quality
- Designs can become part of actual UI
 - Useful for non-standard GUIs
- Easy to distribute electronically



What to do

Find the item you want in the catalog and scan the bar code next to it.



What you selected

Item

Style

Cost

tax:

Total: \$ 0.00

All done?

Place your order

Print this list

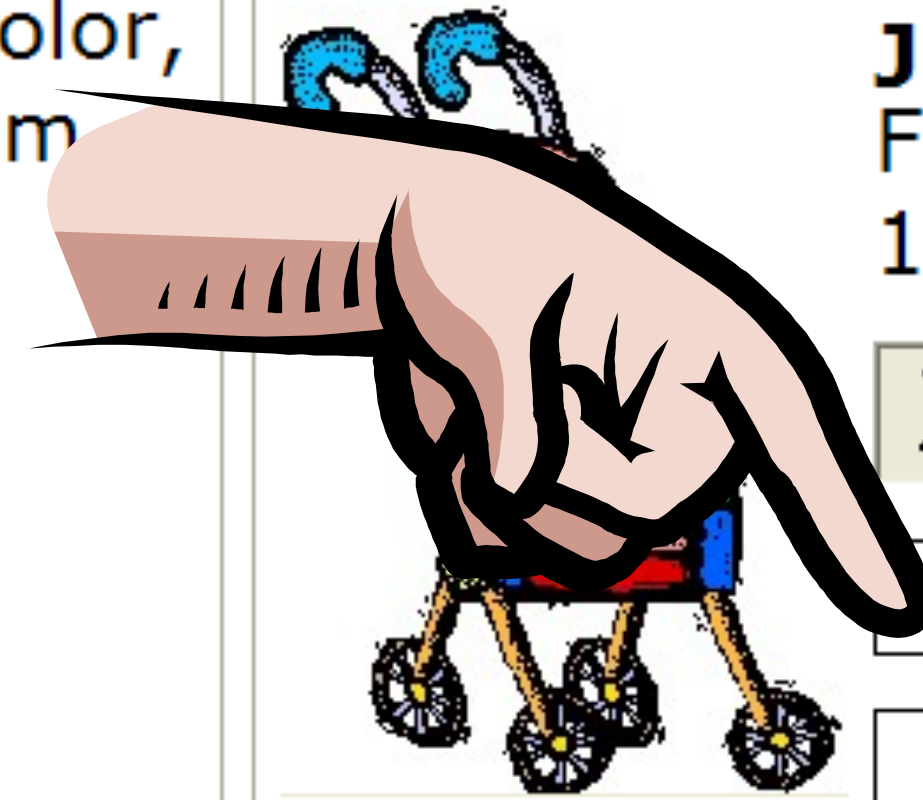
Throw this list away

What to do

Touch a different color,
or scan another item



What you selected



JPG Stroller

For children between
1-3 years old ...\$98.

- ☒ Green
☐ Blue
☐ Red (out of stock)

Item

JPG Stroller

Style

Green

Cost

98.00

Delete

tax: 6.98

Total: \$104.98

All done?

Place your order

Print this list

Throw this list away

What to do

Touch a different color,
or scan another item.



What you selected



JPG Stroller

For children between
1-3 years old ...\$98.

☐ Green

☒ Blue

☐ Red (out of stock)

Item

JPG Stroller

Style

Green

Cost

98.00

Delete

tax: 6.98

Total: \$104.98

All done?

Place your order

Print this list

Throw this list away

What to do

Touch a different color,
or scan another item.



What you selected



JPG Stroller

For children between
1-3 years old ...\$98.

☐ Green

☒ Blue

☐ Red (out of stock)

Item

JPG Stroller

Style

Green

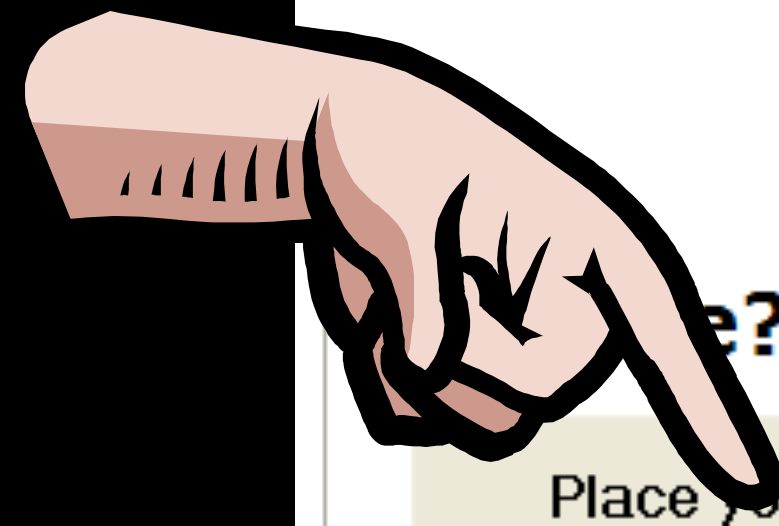
Cost

98.00

Delete

tax: 6.98

Total: \$104.98



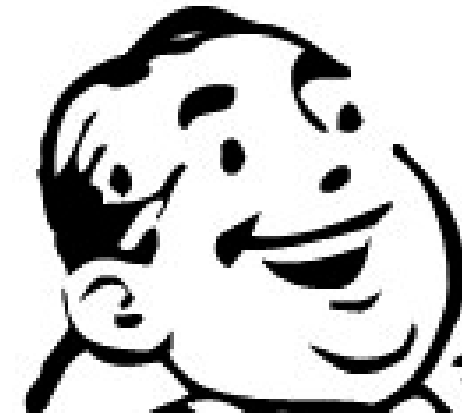
Place your order

Print this list

Throw this list away

What to do

To get your items,
bring your printout to
the front counter.



What you selected

Item

Style

Cost

JPG Stroller

Green

98.00

tax: 6.98

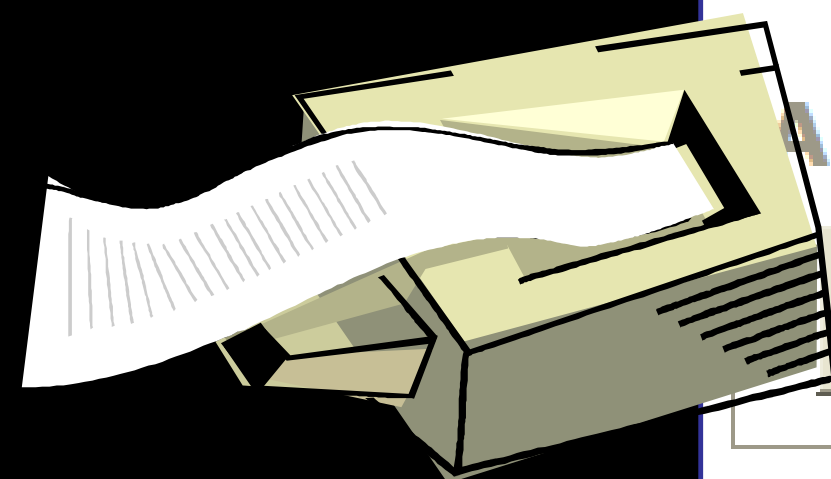
Total: \$104.98

All done?

Place your order

Print this list

Throw this list away



Screenshot Prototypes: Problems

- No interaction, does not capture any dynamic behavior or “feel” of the UI
- Danger of looking too polished, limits feedback, suggests the interface is “done”
- Missing physical aspects of devices

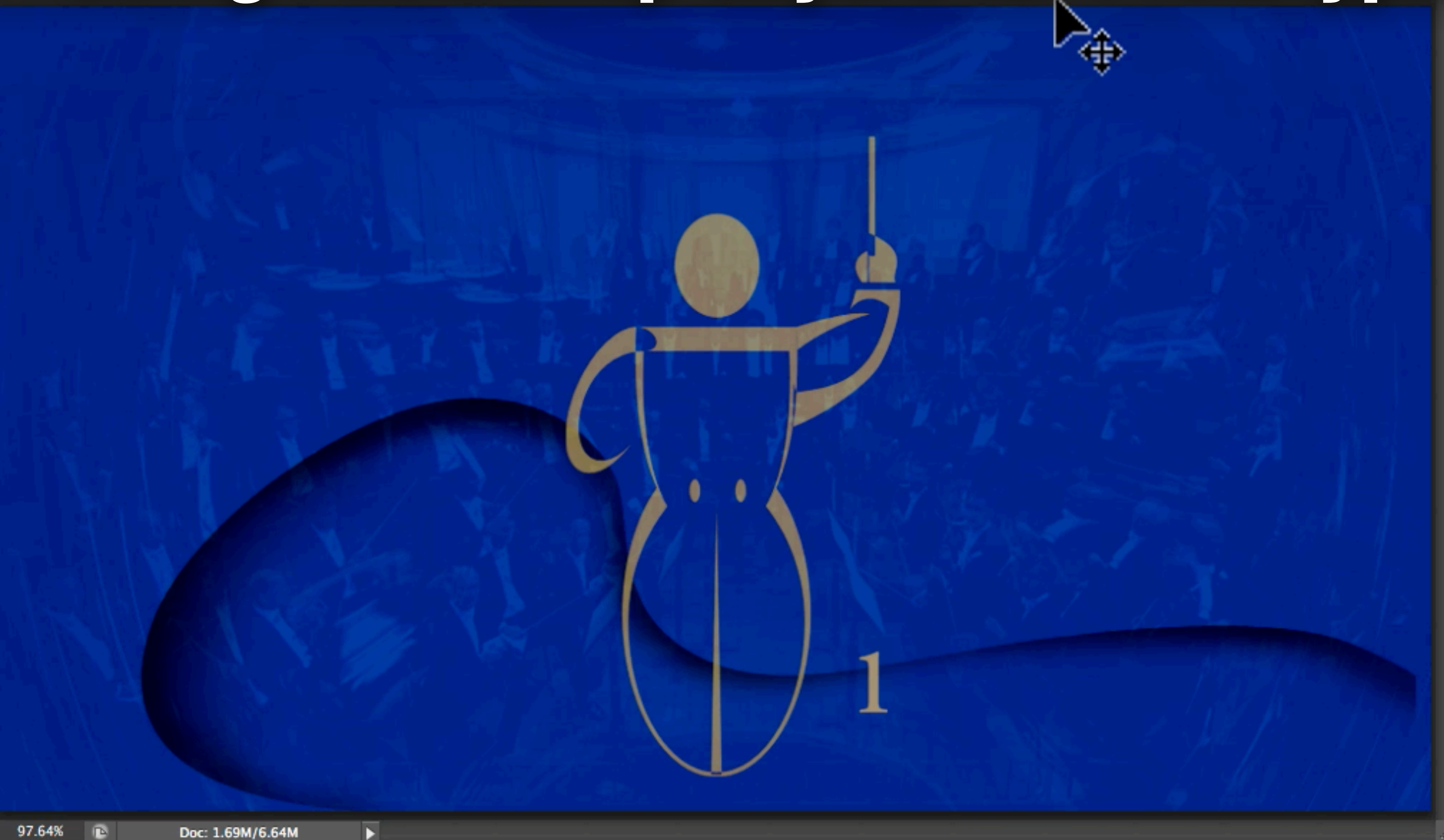
Screenshot Prototypes: Adding Effects

- Scripted simulations
- Using media tools such as PowerPoint or Photoshop layers
- More potential for interactivity:
 - Scene transition by simple input, timing, animation
- Prototype with slightly more vertical depth
- Use as click-through prototype for pitching
- Pro: looks real, good for non-standard UIs, no programming
- Con: still linear—simulation fails when script is not followed

Demonstration: Personal Orchestra Prototype

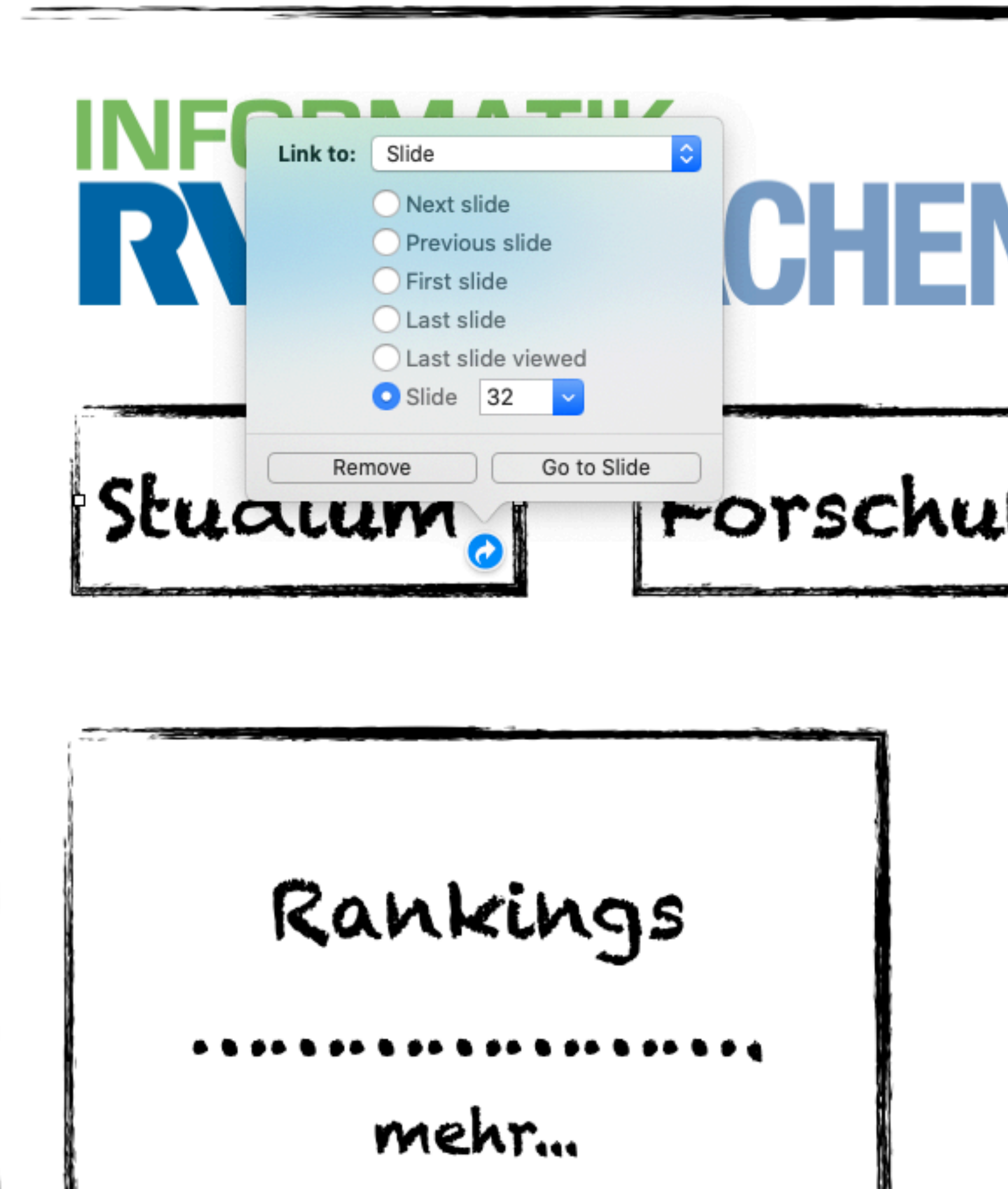
- Alternative to sequential interaction scripts
- Using Photoshop layers to simulate
 - Highlighting menu options
 - Moving to different screens
- Photoshop layers can do some magic
- Normally your Screenshot Prototype will look less polished
 - This example turned out to also become our final graphical layout

Using Photoshop Layers for Prototyping



Non-Linear Digital Prototypes

- Connect UI elements to a specific screenshot
- Prototypes with more horizontal and vertical depth
- No predefined sequence of actions
 - Users can decide what to do next and **do it themselves**



Suche:

Studium

Forschung

Wirtschaft

Die Fachgruppe

Rankings

.....
mehr...

Video

.....
mehr...

Veranstaltungen

.....
Veranstaltung 1

.....
Veranstaltung 2

.....
Veranstaltung 3

.....
mehr...

Aktuelles

.....
Aktuelles 1

.....
Aktuelles 2

.....
Aktuelles 3

.....
mehr...

Suche:

Studium

Forschung

Wirtschaft

Die Fachgruppe

Aktuelles

Studieren
bei uns?

Studiengänge

Ansprechpartner

Service

Absolventen

nn.nn.2011 Abend der Offenen Tür

nn.nn.2011 Girls' Day

nn.nn.2011 Tag Der Informatik

Suche:

Studium

Forschung

Wirtschaft

Die Fachgruppe

Aktuelles

Studieren
bei uns?

Studiengänge

Ansprechpartner

Service

Absolventen

Warum Informatik an der RWTH?

Girls' Day 2011

Schülerinformationstag 2011

Helle Köpfe 2011 für Grundschüler

5 vor 12: Die Wissenschaftsnacht 2011

Fit für Informatik? Mach' den Test!

Vorkurs Informatik

Suche:

Studium

Forschung

Wirtschaft

Die Fachgruppe

Aktuelles

Studieren
bei uns?

Studiengänge

Ansprechpartner

Service

Absolventen

Bachelor Informatik

Master Informatik

Master of Science in Software Systems Engineering

Master of Science in Media Informatics

Promotionsstudium Informatik

Diplomstudiengang Informatik

Lehramtsstudiengang Informatik - Gymnasium und Gesamtschule

Schwerpunkt Informatik im Studiengang Technik-Kommunikation

Suche:

Studium

Forschung

Wirtschaft

Die Fachgruppe

Aktuelles

Studieren
bei uns?

Studiengänge

Ansprechpartner

Service

Absolventen

Fachschaft Mathematik/Physik/Informatik

Studienberatung Bachelor Informatik

Studienberatung Master Informatik

Studienberatung Master Software Systems Engineering

Studienberatung Master Media Informatics

Studienberatung Lehramt Informatik

Studienberatung Technik-Kommunikation

Auslandsstudienberatung

Prüfungsausschüsse...

Suche:

Studium

Forschung

Wirtschaft

Die Fachgruppe

Aktuelles

Studieren
bei uns?

Studiengänge

Ansprechpartner

Service

Absolventen

Informatik-Bibliothek

Rechnerbetrieb Informatik (RBI)

RWTH-Rechenzentrum

Alle Lehrveranstaltungen in CAMPUS

Das RWTH-Lernportal L2P

Vorlesungen auf iTunes U

Prüfungsordnungen

Formulare

Semestertermine

Suche:

Studium

Forschung

Wirtschaft

Die Fachgruppe

Aktuelles

Studieren
bei uns?

Studiengänge

Ansprechpartner

Service

Absolventen

Master

Promotion

Berufseinstieg

Career Center

Alumni Office

Suche:

Studium

Forschung

Wirtschaft

Die Fachgruppe

Aktuelles

...

Personen

...

Gruppen

...

Projekte

...

Publikationen

Presse...

Auszeichnungen

...

Forschen an
der RWTH

Stellenangebote

...

Suche:

Studium

Forschung

Wirtschaft

Die Fachgruppe

Aktuelles

Die Informatik als Partner Angebote Recruiting Sponsoring & Fundraising

...

...

Weiterbildung... ...

...

Suche:

Studium

Forschung

Wirtschaft

Die Fachgruppe

Aktuelles

Über die Fachgruppe Fachgruppen-Sprecher Kooperationen

Struktur

Mitgliedschaften

Jahresberichte

Geschichte

Digital Prototyping Software



**Apple
Keynote**



miro



**Microsoft
PowerPoint**



Figma



**Origami
Studio**

More examples:

- DIS 2
- <https://www.interaction-design.org>

Prototyping Tools: Animation Apps

- Timeline metaphor
- Good for intricate animations
- Powerful when extended with scripts
 - But: Scripting languages are clumsy by CS standards
- May allow for integration of non-standard hardware and other OS features
- E.g., Adobe Animate
- Can even become final product
- But: Large designs become hard to manage



Image adopted from tutorial video: <https://creativecommons.adobe.com/cc/discover/learn/animate/beginner/graphic-design/animate-basics/vector-pattern-brush-animation>

Prototyping Tools: Web

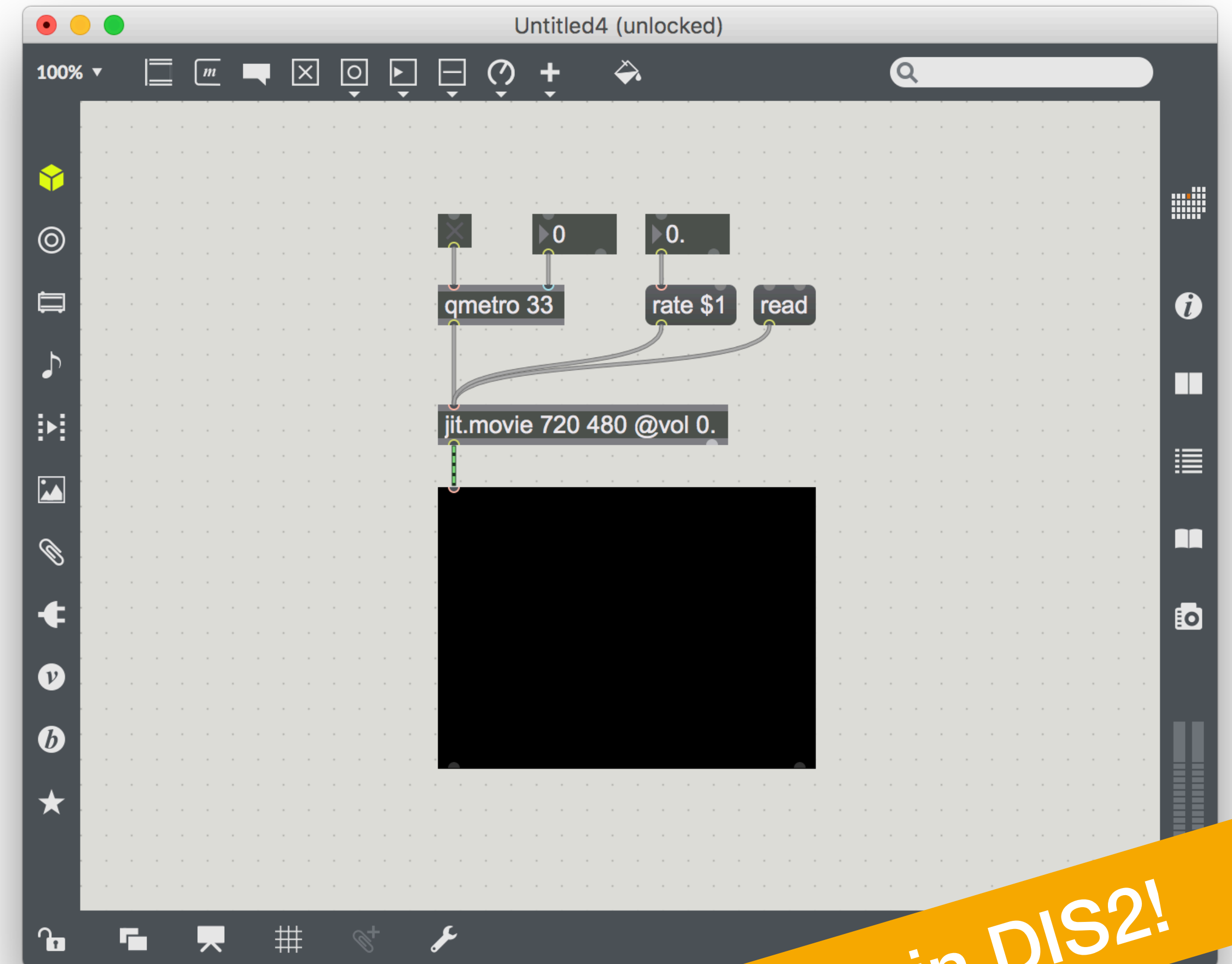
- HTML + JavaScript, etc.
- Natural choice for web interface design
 - Can become final product
- Ubiquitous
 - Many tools (Electron, Cappuccino, ...)
 - Cleartext format
 - Viewable in any browser (in theory...), over the net
 - But: No precise look & feel (nature of the web)

Prototyping Tools: Rapid Development Environments

- Visual Basic .NET, Tcl/Tk, etc.
- Good for standard GUIs (create standard look & feel)
- Often become final product
- Partly interpreted
 - Quick development cycle, but potential performance issues
- Distribution: OK
 - Not always cross-platform
 - May require specific runtime environment

Prototyping Tools: Special-Purpose

- Example: MAX/MSP
 - Multimedia development environment
 - Originally for MIDI applications
 - Extended to handle graphics, audio, and video
 - Build applications by connecting “patches” that process incoming data
 - Very helpful for specific type of applications
 - MIDI/audio/video processing, interactive music systems
 - Can be used for end products (WorldBeat)
 - Distribution: Mac and Windows, free player
 - pd for Linux



More in DIS2!

User Interface Builders

- Graphical/textual tools to define UI of real software application
- Usually part of integrated development environment (IDE)
- Pro:
 - Finished design can be used for final implementation
 - Real look & feel
 - Vertical functionality can be added easily
- Con:
 - Often limited to one UI toolkit
- Evolving into declarative coding environments (code and UI preview are the same, e.g., SwiftUI)

TemperatureConverter > My Mac
TemperatureConverter: Ready | Today at 12:13

Main.storyboard
TemperatureConverter > Temp...verter > Main....board > Main....(Base) > View...Scene > View...ntroller > View > Convert

TemperatureConverter > TemperatureConverter > AppDelegate.swift > ViewController.swift > Assets.xcassets > Main.storyboard > Info.plist > TemperatureConverter.entitle... > Products

Application Scene
Window Controller Scene
Window Controller
Window
First Responder
Storyboard Entry Point
Relationship "window..."
View Controller Scene
View Controller
View
Fahrenheit:
Celsius:
FahrenheitField
CelsiusField
Convert
Convert
First Responder

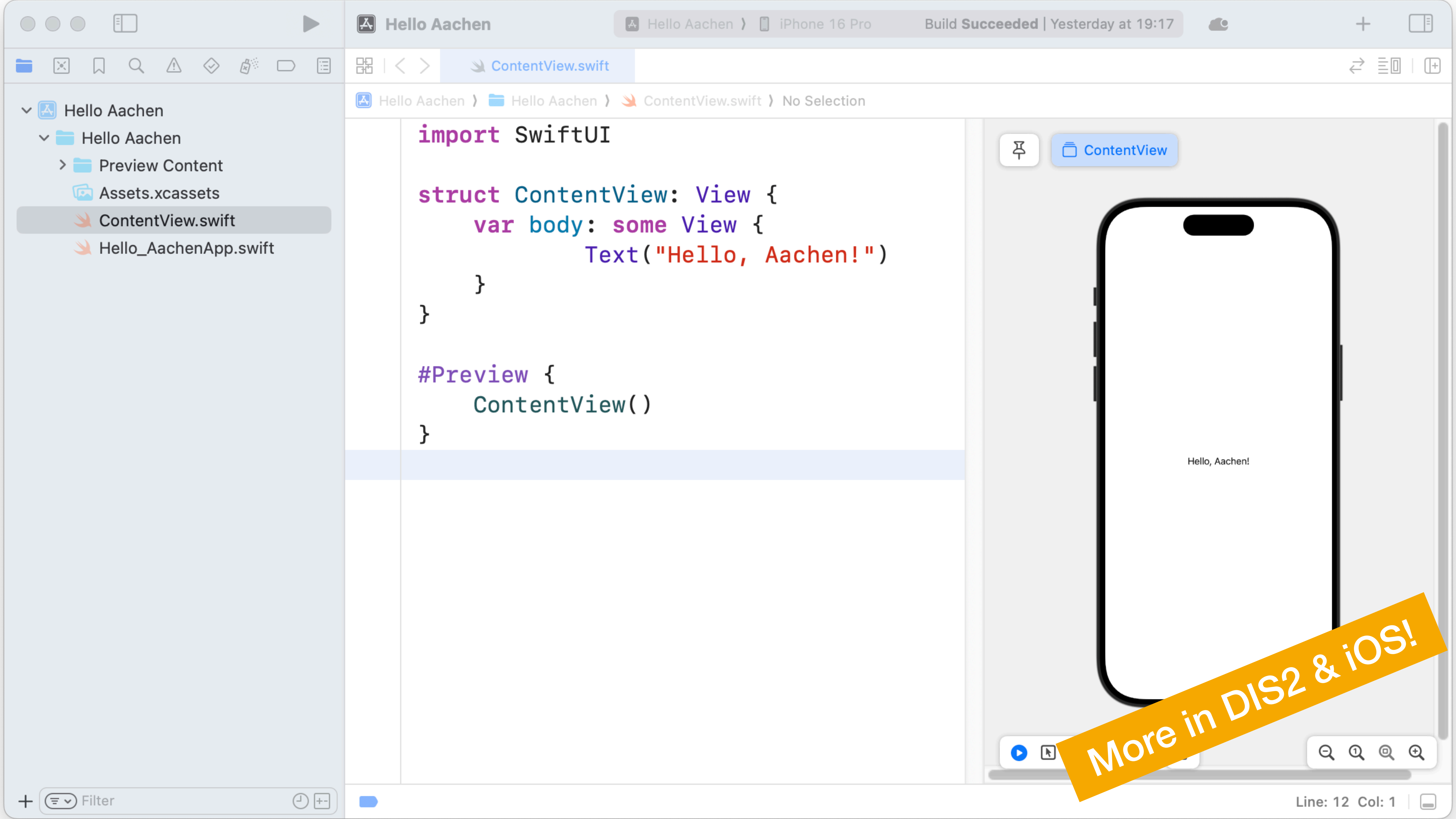
Main Menu
TemperatureConverter File Edit Format View Window Help
Window Controller
Window
View Controller
Fahrenheit:
Celsius:
Convert

ViewController.swift
TemperatureConverter > Temper...onverter > ViewController.swift > No Selection

```
1 //  
2 //  
3 //  
4 //  
5 //  
6 //  
7 //  
8 import Cocoa  
9  
10 class ViewController: NSViewController {  
11  
12     @IBOutlet weak var fahrenheitField: NSTextField!  
13     @IBOutlet weak var celsiusField: NSTextField!  
14  
15     override func viewDidLoad() { ... }  
16  
17  
18     @IBAction func convert(_ sender: NSButton) { ... }  
19  
20  
21  
22  
23  
24 }  
25  
26
```

Connect Action

View as: Light Appearance



Digital Prototypes: Dangers

- Users focus on design details and overlook larger problems
- Users afraid to criticize or suggest changes to “nice” UI design
 - Looks like it was so much work...
- Management may think it's real 😊
 - Looks like the software is almost done
 - Reason: Users think the interface is the system! (“Conceptual models”)

Hardware Prototypes

Hardware Prototypes

- For systems that are hard to imagine by software alone
 - Example: Palm's wooden blocks
- Physical interaction is important
 - E.g., new 3-D mouse
- Design in wood, foam core, plastics, styrofoam, cardboard, ...
- Problem: high effort to build and change

Prototypes of Microsoft Touch Mouse, Cut From Foam



3D Printed Prototype of Pebble Watch





HELLO!
Please
Select
Device :)



For more freedom of play use the
Virtual Airguitar Simulator™

Wizard of Oz

- Human ‘wizard’ simulates system response
 - Interprets user input according to an algorithm
 - Controls computer to simulate appropriate output
 - Uses real or mock interface
 - Wizard sometimes visible, sometimes hidden
 - “Pay no attention to the man behind the curtain!”
- Good for:
 - Adding simulated and complex vertical functionality
 - Testing futuristic ideas
 - Example: 1984 IBM voice recognition editor

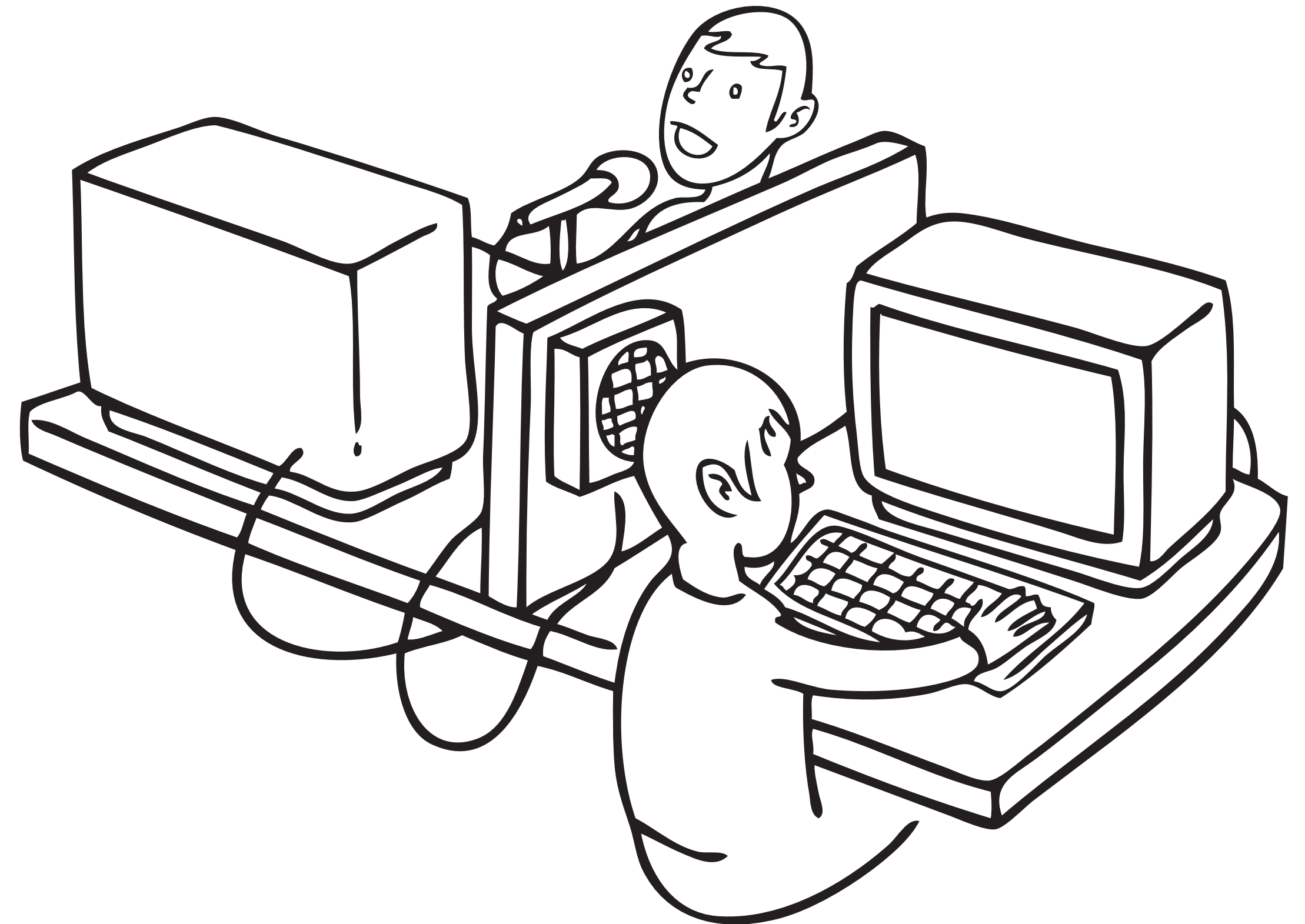


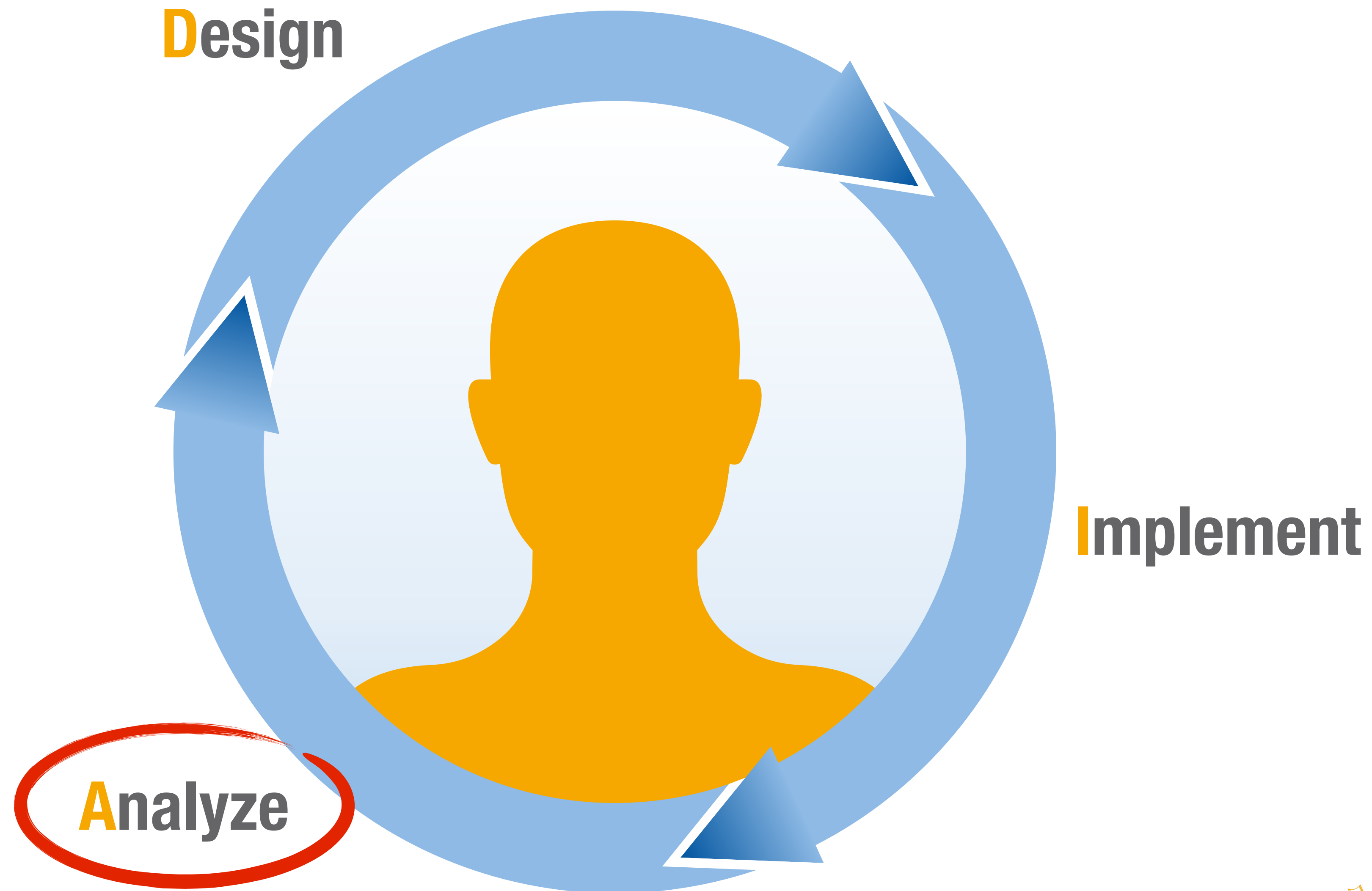
Image: Buxton 2007, *Sketching User Interfaces*

What to Do with a Prototype?

- Throw away
 - If creation was quick and cheap
- Continue to develop
 - Prototype improved incrementally
 - Becomes final product
 - Problem: Has to use production-strength technology

Evaluation

When to Evaluate



When to Evaluate

- Evaluation should happen throughout the entire software development process
 - Early designs are more often evaluated by the design team, analytically and informally
 - Later implementations are more often evaluated by users, experimentally and formally

Why Evaluate?

- To ensure that system matches user needs
 - Necessary even if design was already user-centered (interviews, ...)!
- To judge system features
 - Does it facilitate users' tasks?
 - Does it offer the right features, easy to reach, and presented as expected?
- To judge effects on users
 - How easy is it to learn and use the system?
 - How do users feel about the system?
 - Are there areas that overload users?
- To discover specific problems
 - Do unexpected/confusing situations come up?

Where to Evaluate: Lab

- + Equipment (A/V, see-through mirrors, special computers)
- + No disruptions
- + Quiet
- Natural environment missing (shelves, wall calendar, ...)
- Unnatural situation (relevance?)
- Preferable if
 - the real location is dangerous
 - remote (ISS), or
 - a controlled situation is needed



Where to Evaluate: In The Field

- Studies in the users' natural environment
 - + More realistic (also *because* of disruptions)
 - + Situations and behavior more natural
 - + Better suited to long-term studies
 - Noise, task interruptions
 - Will still feel like a test situation

Participatory Design

- Involve users as part of design team throughout entire software process
- Originated in Scandinavia where it is the law for certain products
- Techniques for team communication
 - Brainstorming, storyboarding, workshops, interviews, role plays, paper prototypes
- Problems
 - High effort, conflicts with client hierarchies, user conversion

Evaluation Techniques

Evaluating Without Users

- E1** Literature Review
- E2** Cognitive Walkthrough
- E3** Heuristic Evaluation
- E4** Model-based Evaluation
 - GOMS, HCI Design Patterns, ...

Evaluating With Users

Qualitative

- E5** Model Extraction
- E6** Silent Observation
- E7** Think Aloud
- E8** Constructive Interaction
- E9** Retrospective Testing

Quantitative

- E10** Controlled Experiments

+ Interviews, questionnaires,...

E1: Literature Review

- Many research results about user interface design have been published
- Idea: Search literature for evidence for (or against) aspects of your design
- Saves own experiments
- Results only carry over reliably if the context (users, assumptions) is very similar



E2: Cognitive Walkthrough

- Goal: Judge learnability and ease of use — without users
- Analytical method for early design or existing systems
- Requires an HCI **expert** (designer, cognitive psychologist), interface description, task description, user profile, and context description; takes time
- For each task, derive goal—intention—action sequence, and ask
 - Does system help the user to get from goals to intentions and actions?
 - What knowledge and cognitive processes will the user need for this decision process?
 - What problems could learning/doing this step have?
- Question forms can capture psychological knowledge to guide the user

E3: Heuristic Evaluation

- Variant of Cognitive Walkthrough
- Choose usability heuristics
 - General guidelines, e.g., Ten Golden Rules
- Step through tasks and check whether guidelines are followed
 - + Quick and cheap
 - Subjective
 - Better done by several independent designers



E4: Model-based Evaluation

- Some models exist that offer a framework for design and evaluation
- Examples:
 - GOMS, KLM
 - Information efficiency
 - Design Rationale (History of design decisions with reasons and alternatives)
 - HCI Design Patterns

} next lecture



Figure 17: Passing on a mouse for a group display.

...you have picked your hardware to control the room and its services—ROOM CONTROLLER (15), and now need to decide how the technology is operated by the users.



Interactive technology likes to be told when something happens or when it is supposed to do something. But people easily forget that extra step, especially when in the middle of a high-energy brainstorming session.

A research video by MIT once showed a group of researchers having a meeting around the table, and the room was “listening in” on the conversation going on. Whenever a certain point was reached, such as deciding to add a new item to the agenda, or delegating a task to a member in the room, everybody had to shut up, and the moderator would speak the corresponding commands for the computer to keep up with what was going on. It was the worst group support interface imaginable. Good group support software follows what’s going on in the room as good as it can, trying to detect from a variety of sensors, models, and other input what the current activity and actors are, and then takes initiative on a simple, reliable level to help the actors, without presuming to understand more than it can.

Computer scientists will argue that deriving this information from sensor values is not reliable, so the computer needs clear commands in order not to do something wrong. This is perfectly true in distributed settings with low bandwidth for human communication: If user A decides to pass control over the shared mouse cursor to remote user B in a shared application, he usually has to click a button to do so. In a collocated setting of an AE, an enormous advantage comes to the help of the system: social protocol. The people in the room can see and hear each other. If one person is controlling the mouse cursor using their laptop, and someone else wants to

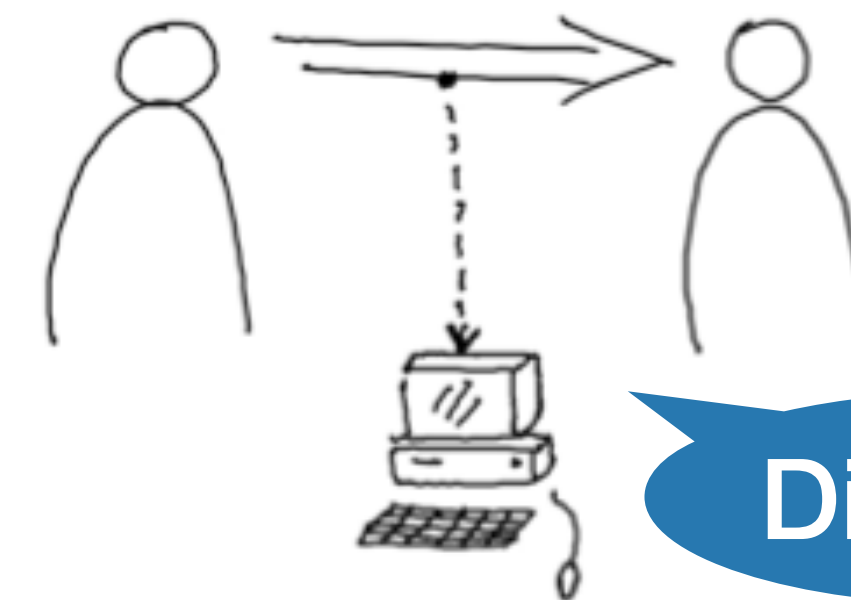
take over with their own laptop, they will just say so. The computer does not need to understand this verbal command, nor does he need to lock the cursor for everybody else but one user at a time: It can simply accept cursor movement from everybody in the room; if there’s a conflict of concurrent access, the users will quickly and easily notice and resolve it among themselves. This approach, on the other hand, saves the users having to send explicit messages each time they wish to pass control of that cursor to someone else, making the interaction much more fluid.

Examples include the design of the interaction for the iRoom’s remote cursor control that allows “mouse fights” to occur, simply always using the last coordinate received; or its iClipboard feature that lets people cut and paste in a single shared clipboard for the room.

grad et al., in their chapter elsewhere in this book, reflect on this concept by suggesting room infrastructure in which “...users and social conventions in an environment take responsibility for actions, and the system infrastructure is responsible for providing a fluid means to execute those actions.”

Therefore:

Do not put unnecessary protocols into place that are aimed at avoiding overlapping access to technology, if that collision can be easily noticed and fixed by the users through social interaction. If a user issues a social protocol act, such as passing a wireless mouse to someone else, it is an additional repetitive step from the user to tell the room what he just did for everyone else to clearly see.



This is a basic pattern with no further references within this language.

Problem

Examples

Solution

Diagram

References

**Name,
Ranking**

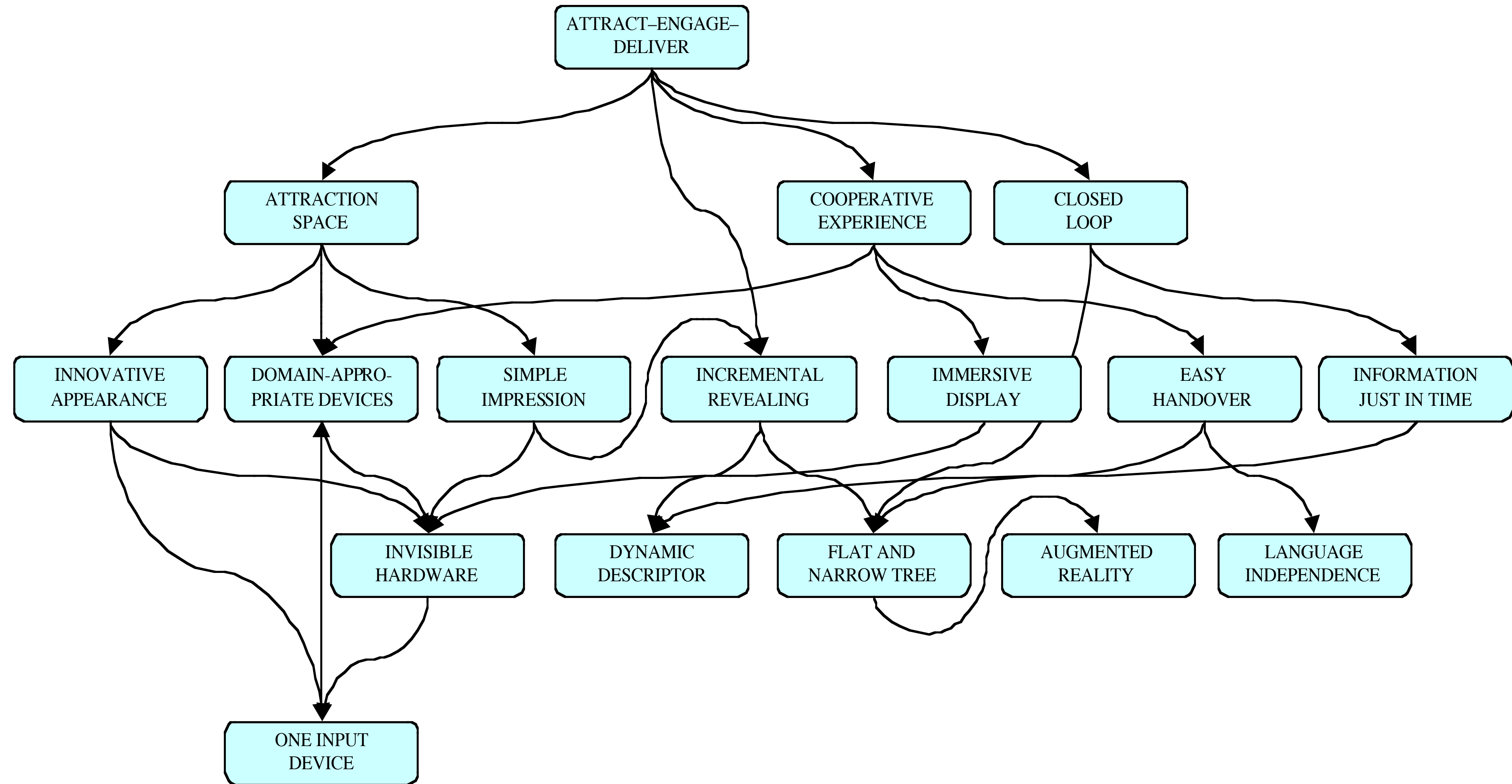
Sensitizer

Context

Pattern Languages in HCI

- Early references
 - Norman & Draper (1986): User-Centered System Design
 - Earlier than in SW-Eng!
 - Norman (1988): The Psychology (Design) of Everyday Things
 - "Fascinating to skim, frustrating to read" :)
 - Apple Macintosh Human Interface Guidelines (1992)
 - "seminal in the field of environmental design"

Interactive Exhibits: A Pattern Language



Summary

- Many iterations of prototypes necessary
 - Paper prototypes for high-level, rough UI design
 - Types: Storyboard-like & Flipbook
 - Post-it prototypes to create first interactive versions
 - Digital prototypes for more detailed, interactive walkthroughs
 - Can be vertical and/or horizontal
 - Types: Screenshot, Non-Linear
 - Tools: Presentation, Graphics, Animation, Wireframing, Web Design, Interface Builders
 - Hardware prototypes if physical interaction is important
- Wizard of Oz technique to add simulated and complex vertical functionality
- Throw it away or keep it?
- Evaluation: When, Why, and Where