

# Review

- What are the differences between mistakes and slips?
- What are the different types of slips?
- How do we tend to correct slips?
- What are forcing functions?
- How can UI design help to avoid errors?
- What are Norman's Seven Principles of Design?



# Design Principles



# Perception

Our brains are wired to make sense of what we perceive.





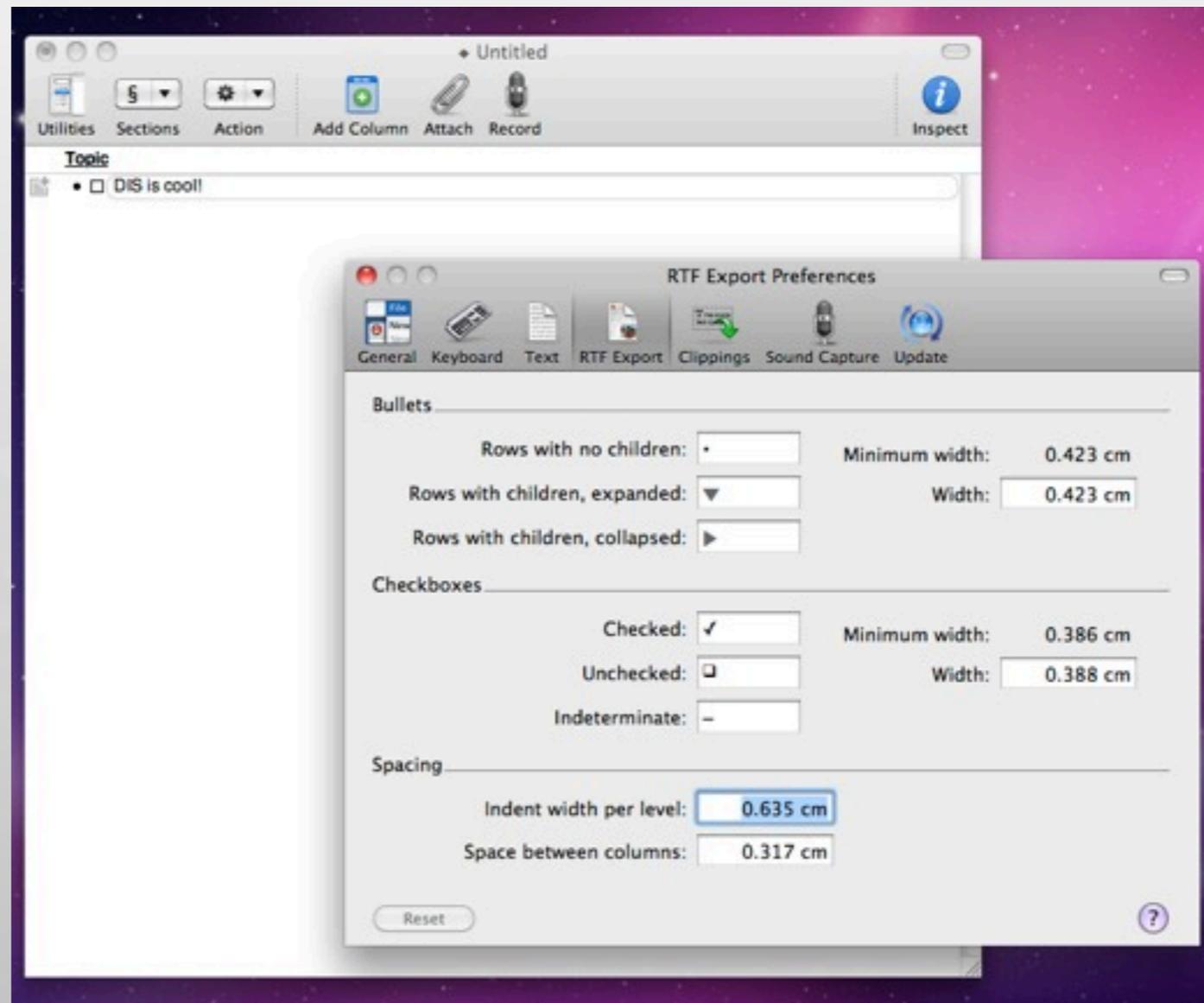
# Gestalt Theory

- Köhler, Koffka, Wertheimer (Berliner Schule):  
“*Gestaltpsychologie*”, 1912
- What do humans perceive as belonging together spatially or temporally?
- Basis of order in perception, movement, memory, thinking, learning, and acting
- Overall 100+ Gestalt laws



# Why Should I Care?

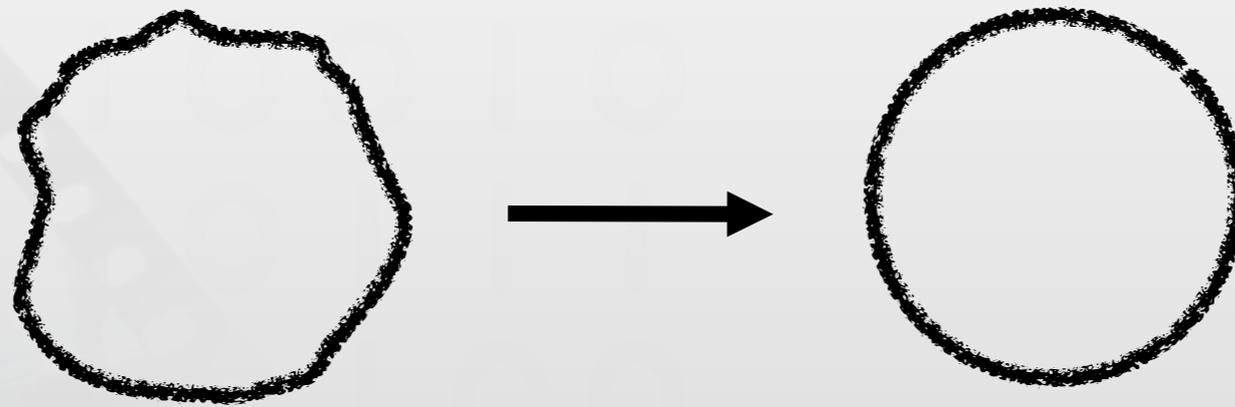
- Simple rules for visual (and auditory) UI design
- Hints how users will react to spatial and temporal order
- Good UIs respect and use Gestalt laws for understandability and intuitiveness



# In-class Experiment



# Law 1: Good Shape



- Perception has tendency towards remembering things as “good” / clear / simple shapes
- “Cognitive compression algorithm”!
- Easier shape  $\Rightarrow$  easier to remember



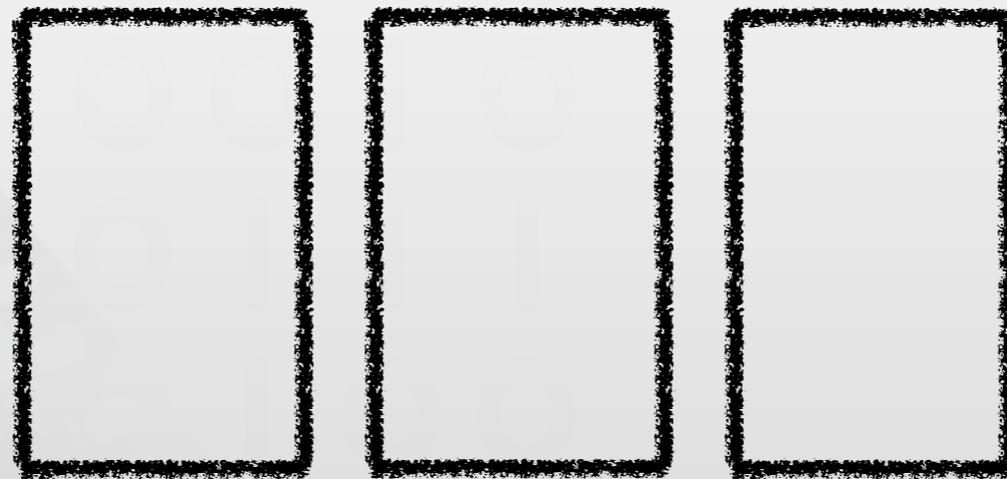
# Law 2: Proximity



- Spatially (or temporally!) close objects (events) are perceived as belonging together.
- Advantage: allows for order by position only, without other aides
- Helps to keep the interface simple



# Law 3: Closure



- Closed shapes appear as belonging together
  - Temporally?
- Foundation of window metaphor
- But: Don't Overdo It.

Contact Info

Name

First:

Last:

Address

Number:

Street:

City:

State:  ▼

Zip Code:

Assets

Salary

<=20K

>20-40K

>40-60K

>60-80K

>80K

Real Estate

Home

Rental

Farm

Other

Bank

Name:

Accounts

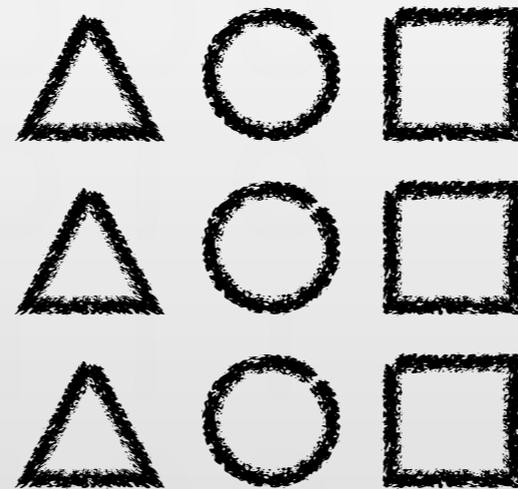
Checking:

Savings:

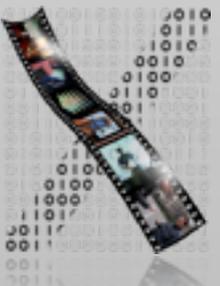
Too many boxes. (From Johnson: *GUI Bloopers*)



# Law 4: Similarity



- Similar shapes appear as belonging together
  - Temporally?
- Different objects have higher information content (i.e., cognitive effort)
  - This can be A Good Thing or A Bad Thing
- **Similar** is not necessarily **constant**
  - Linearity, “elegant curve”,...



Unread news in rec.humor.funny	1 article	+	5 old
Unread news in rec.humor.funny.reruns	1 article	+	5 old
Unread news in clari.living.columns.miss_manners	1 article	+	1 old
Unread news in misc.taxes.moderated	98 articles	+	383 old
Unread news in comp.dcom.telecom	35 articles	+	74 old
Unread news in comp.dcom.modems	240 articles	+	969 old
Unread news in alt.security	18 articles	+	91 old
Unread news in comp.os.linux.announce	9 articles	+	24 old
Unread news in comp.os.linux.development.apps	92 articles	+	175 old
Unread news in comp.os.linux.development.system	115 articles	+	187 old
Unread news in comp.os.linux.misc	400 articles	+	924 old
Unread news in comp.os.linux.networking	301 articles	+	560 old
Unread news in comp.os.linux.setup	264 articles	+	1711 old
Unread news in comp.periphs.printers	5 articles	+	839 old
Unread news in comp.protocols.kerberos	16 articles	+	29 old
Unread news in comp.security.announce	2 articles	+	0 old
Unread news in comp.security.gss-api	2 articles	+	2 old
Unread news in comp.security.misc	36 articles	+	60 old
Unread news in comp.security.unix	94 articles	+	105 old
Unread news in comp.windows.x.announce	2 articles	+	2 old
Unread news in comp.windows.x.apps	4 articles	+	22 old
Unread news in gnu.emacs.bug	15 articles	+	31 old
Unread news in news.announce.newgroups	18 articles	+	5 old
Unread news in news.software.b	2 articles	+	7 old
Unread news in news.software.nntp	90 articles	+	90 old
Unread news in news.software.readers	42 articles	+	163 old
Unread news in shore.sys	1 article	+	5 old
Unread news in shore.news	1 article	+	2 old
Unread news in alt.sources	1 article	+	5 old
Unread news in alt.source-code	1 article	+	1 old

Operations apply to current selection or cursor position

Quit	Read	Next	Prev	Catch up	Subscribe	Unsubscribe	Goto group	All groups
Rescan	Prev group	List old	Select groups	Move	Exit	Checkpoint	Gripe	Post
Post & Mail								

Bad button design in xrn

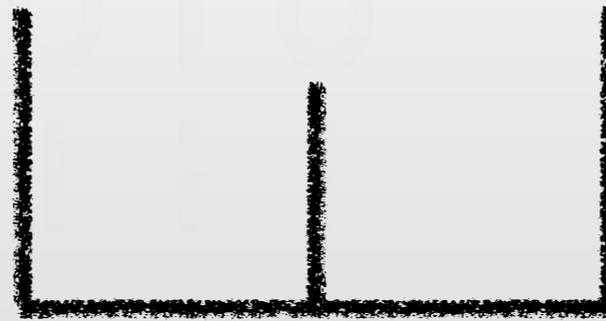
# Law 5: Continuity



- A.k.a. “Law of the Good Curve”
- Continuous shapes appear as belonging together
  - Temporally?
- Example: music



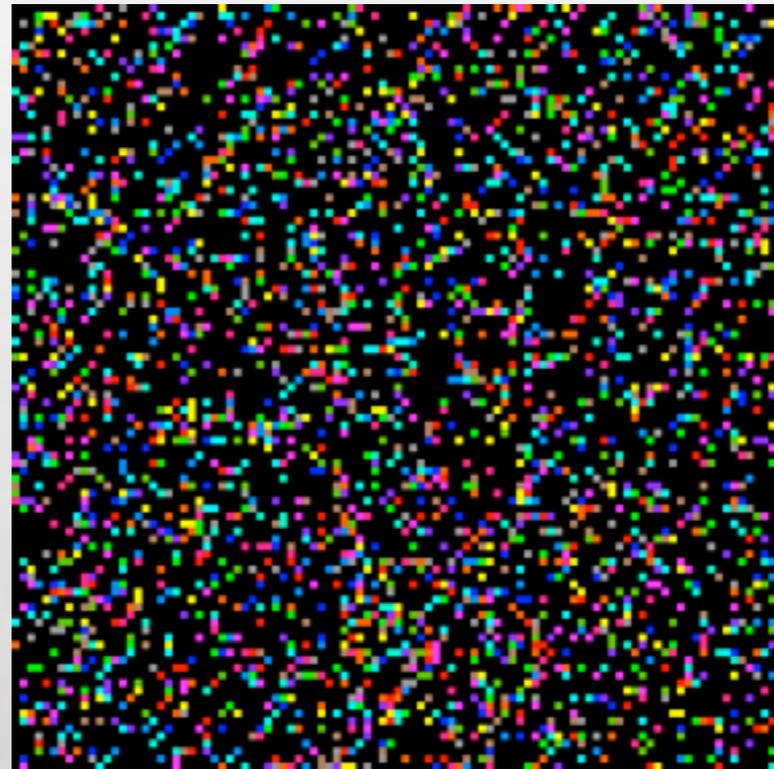
# Law 6: Experience



- We tend to “file” new things into categories we already know
- Uses existing knowledge, thereby saving learning effort and memory
- Foundation for the success of **metaphors** in UI design
  - Analog to real-world models
  - E.g., desktop metaphor



# Law 7: Common Fate



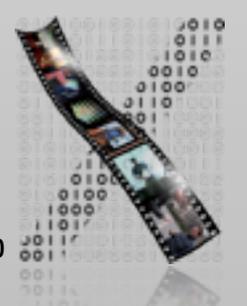
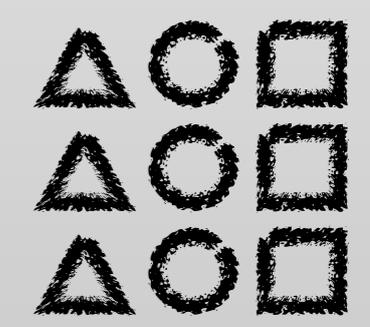
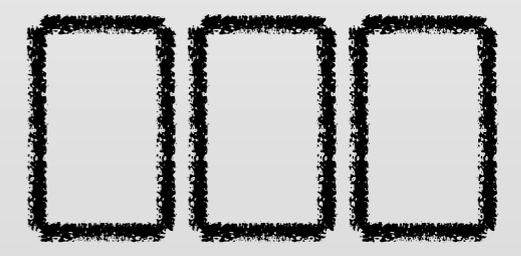
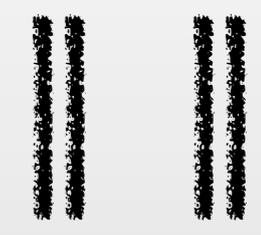
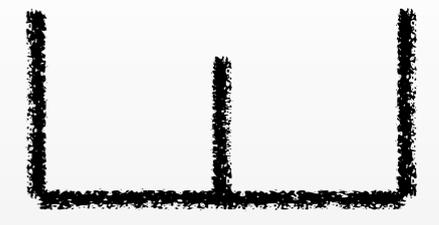
- A.k.a. “Law of Common Movement”
- Animated objects within a static environment appear as a group
- By-Law: Animation has a very strong effect in UI design
  - Here: Blinking in sync groups the items

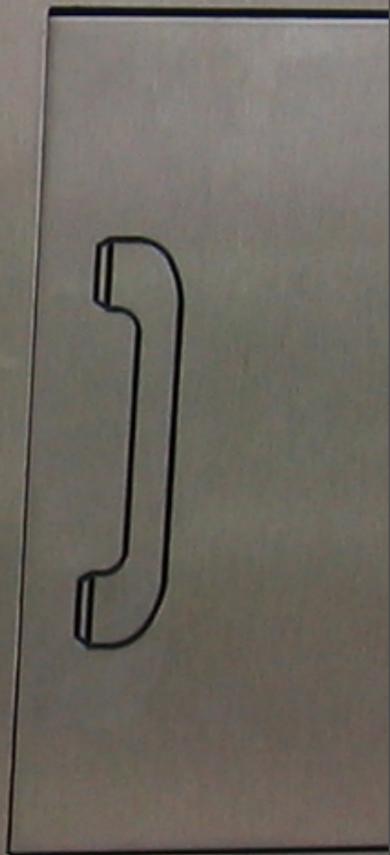
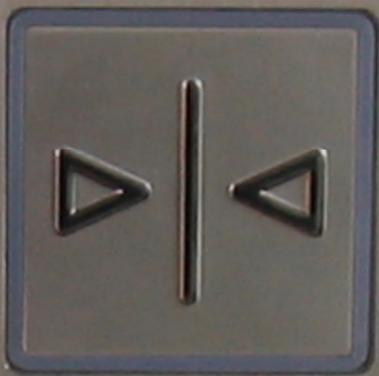
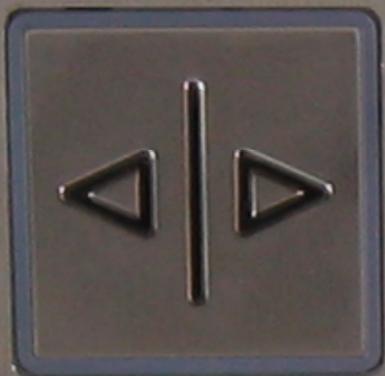


sample web page

This is the *HOTTEST* place ON THE Web

Blinking text is perceived as a group





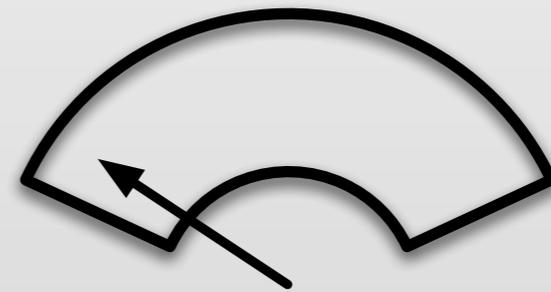
# Information Content in UIs

- Basic unit of information: **bit**
- Toggle button:
  - 2 states:    $\rightarrow \log_2(2) = 1$  bit
- Single digit
  - 10 states: 0...9  $\rightarrow \log_2(10) = 3.3$  bits
- Single letter, upper- and lowercase, U.S.:
  - 52 states: a...z | A...Z  $\rightarrow \log_2(52) = 5.7$  bits



# Information Content in UIs

- Analog scales (reading = estimate)
  - Unmarked scale (experiment)  
3 bits (8 different positions differentiable)
  - Audio pitch, volume, salt content  
Pitch 2.5 bits (But: with perfect pitch 5–6 bits)  
Volume 2 bits  
Saltiness 1.8 bits



# Analog or Digital?

- Example: speedometer in the car
- Analog displays (scales, ...)
  - Quick estimate possible, range limits visible
  - Easy to detect trends
  - But: reading time increases linearly with number of significant digits
- Digital displays (digits, ...)
  - Reading time ~ constant up to 3 – 4 digits
  - But: hard to estimate quickly, trends hard to detect, limits invisible without external labeling



## Theory

- ✓ Models of interaction
  - ✓ Affordances, mappings, constraints, types of knowledge, errors
- ⇒ Design principles
  - Human cognition and performance
  - History and vision of HCI

## Practice

- ✓ Sketching
- ✓ User observation
- ✓ Iterative design
- ⇒ Prototyping
- ⇒ Ideation
- User study and evaluation



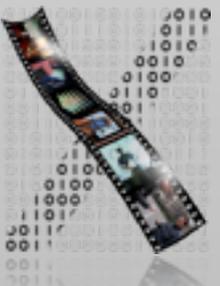
# Software Prototyping: On-Screen Storyboards

- 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 or for pitching
- Pro: looks real, good for non-standard UIs, no programming
- Con: simulation fails when script is not followed



# In-Class 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





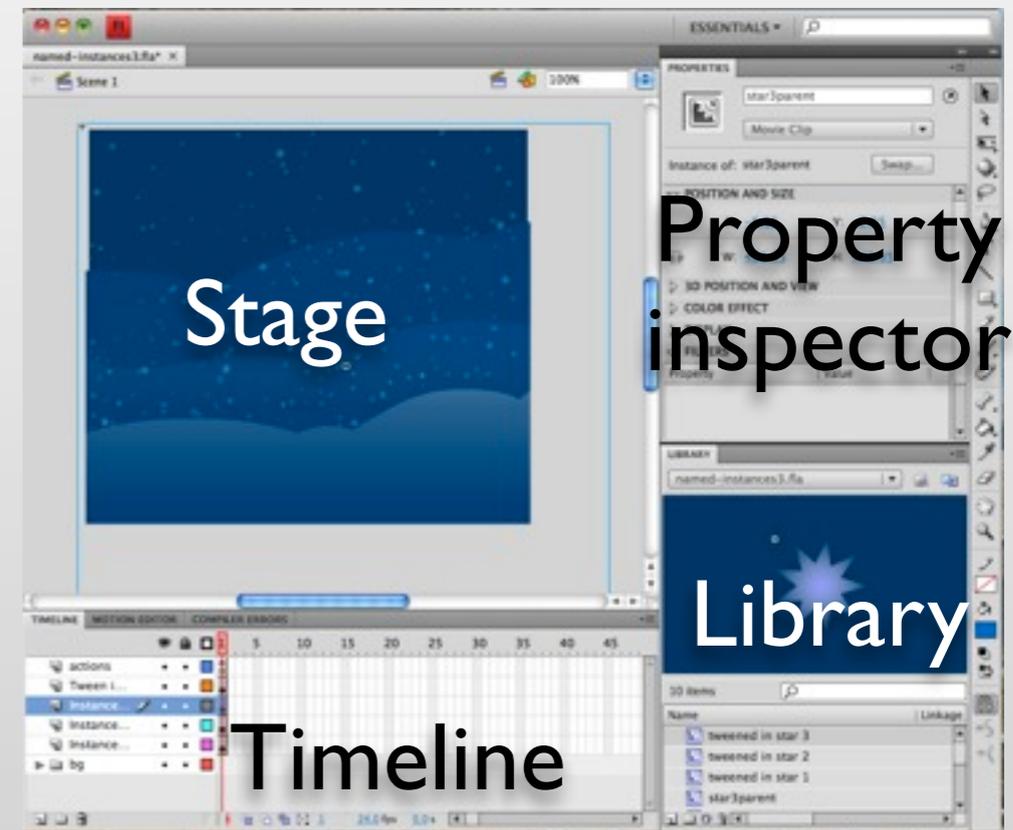
Annen Polka  
Can-Can  
Donauwalzer  
Eine kleine Nachtmusik  
Radetzky Marsch  
Ungarischer Tanz

WIENER PHILHARMONIKER



# Prototyping Tools: Animation Apps

- Usually implement timeline metaphor
- Good for intricate animations
  - Pixel-based (Adobe Director)
    - Maximum control over appearance
  - Vector-based (Flash)
    - Smaller files, editable objects
- 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 (Director Xtras,...)
  - Example: Virtual Vienna



# Prototyping Tools: Animation Apps

- Can even become final product
  - Virtual Vienna, Flash web content,...
- Distribution usually fairly easy
  - Free player apps
- But: Large designs become hard to manage
  - Virtual Vienna example



# Prototyping Tools: Web

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



- FIY
- LumiNet
- Multitouch
- Personal Orchestra
- SLAP
- Snowboard
- TWEND

Publications

- Books
- Papers
- Diploma & Master's Theses
- Upcoming Conferences
- Upcoming Journals

Cooperations

- Apple Training Center
- CocoaHeads
- DorkBot
- hcipatterns.org
- Humtec
- Mac-Beratung (de)
- RUF AE
- UMIC

Login

Login as...

User:

Password:

# Federal President Horst Köhler at the Silhouettes Interactive Exper



# Demo: Prototyping Interaction with Javascript

- Modern Javascript library allows prototyping the user interaction quickly
- [script.aculo.us](http://script.aculo.us)
  - Implementation of common animations and user interactions
  - Convenience `$()` function to access DOM elements (Prototype framework)
- Use your web browser as the IDE



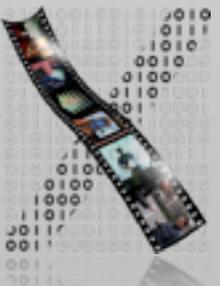


```
> var p = $('puzzle'), info = $('puzzleinfo'), moves = 0;

Sortable.create('puzzle', {
  tag: 'img', overlap: 'horizontal', constraint: false,
  onUpdate: function(){
    info.update('You\'ve made ' + (++moves) + ' move' + (moves>1 ? 's' : ''));
    if(Sortable.sequence('puzzle').join('')== '123456789')
      info.update('You\'ve solved the puzzle in ' + moves + ' moves!').morph('congrats');
  }
});
```

# Prototyping Tools: Rapid Development Environments

- VisualBasic, Tcl/Tk, etc.
- Good for standard GUIs (create standard look & feel)
- Often become final product
- Partly interpreted
  - Quick development cycle, but potential performance issues



# Prototyping Tools: Rapid Development Environments

- Distribution: OK
  - Not always cross-platform
  - May require specific runtime environment
- “Programming for the rest of us”
  - End-user programming
  - Empowers users
  - E.g., Automator in Mac OS X



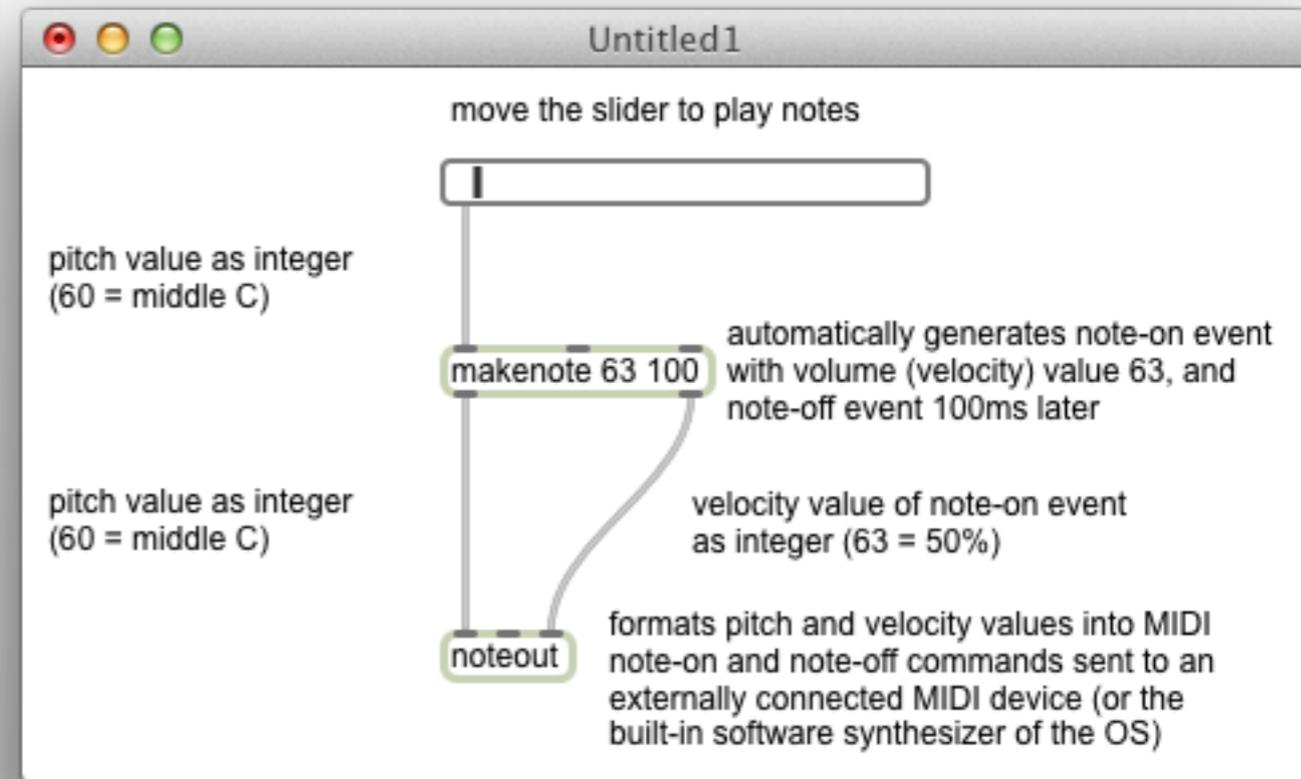
# Prototyping Tools: Special-Purpose

- 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

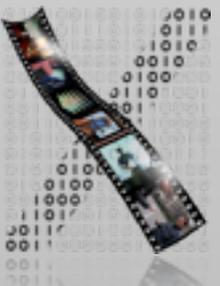


**More in DIS2!**



# 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:
  - Limited to 1 window system and its toolkit (windows, buttons, ...)

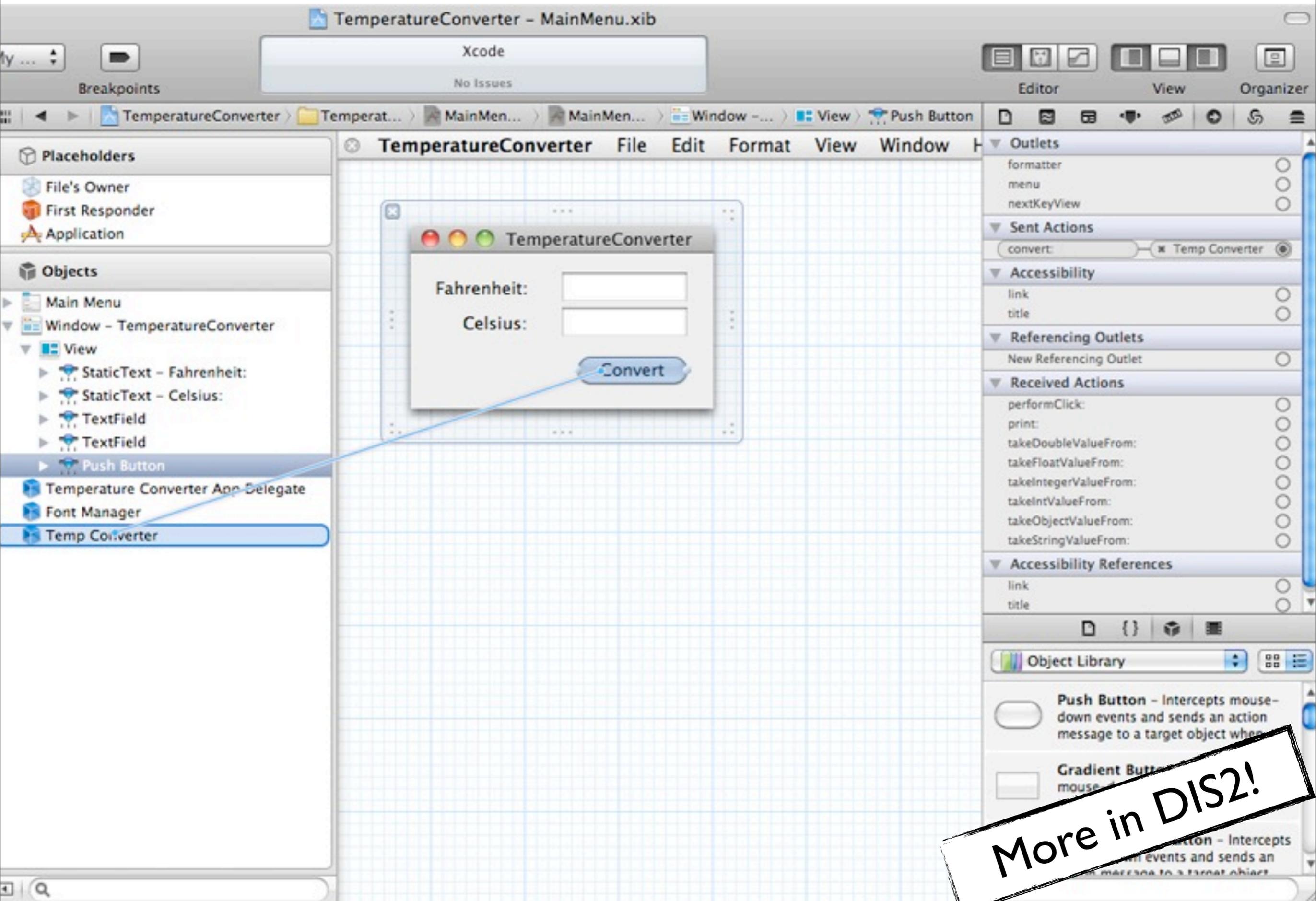


# Example: Interface Builder in Xcode



- Create UIs for Mac OS X and iOS applications
  - Design **static** layout, e.g., position of a button in a window
  - Connect **dynamic** behavior, e.g., connect a button to an action method in a class
- UI can be **tested** without compiling or writing any code
- Suggests a more user-centered implementation process
  - Start with the UI, not the application functionality
  - IB generates source code skeleton that can then be filled in
  - IB uses special constants to include hints about outlets and actions in the source code



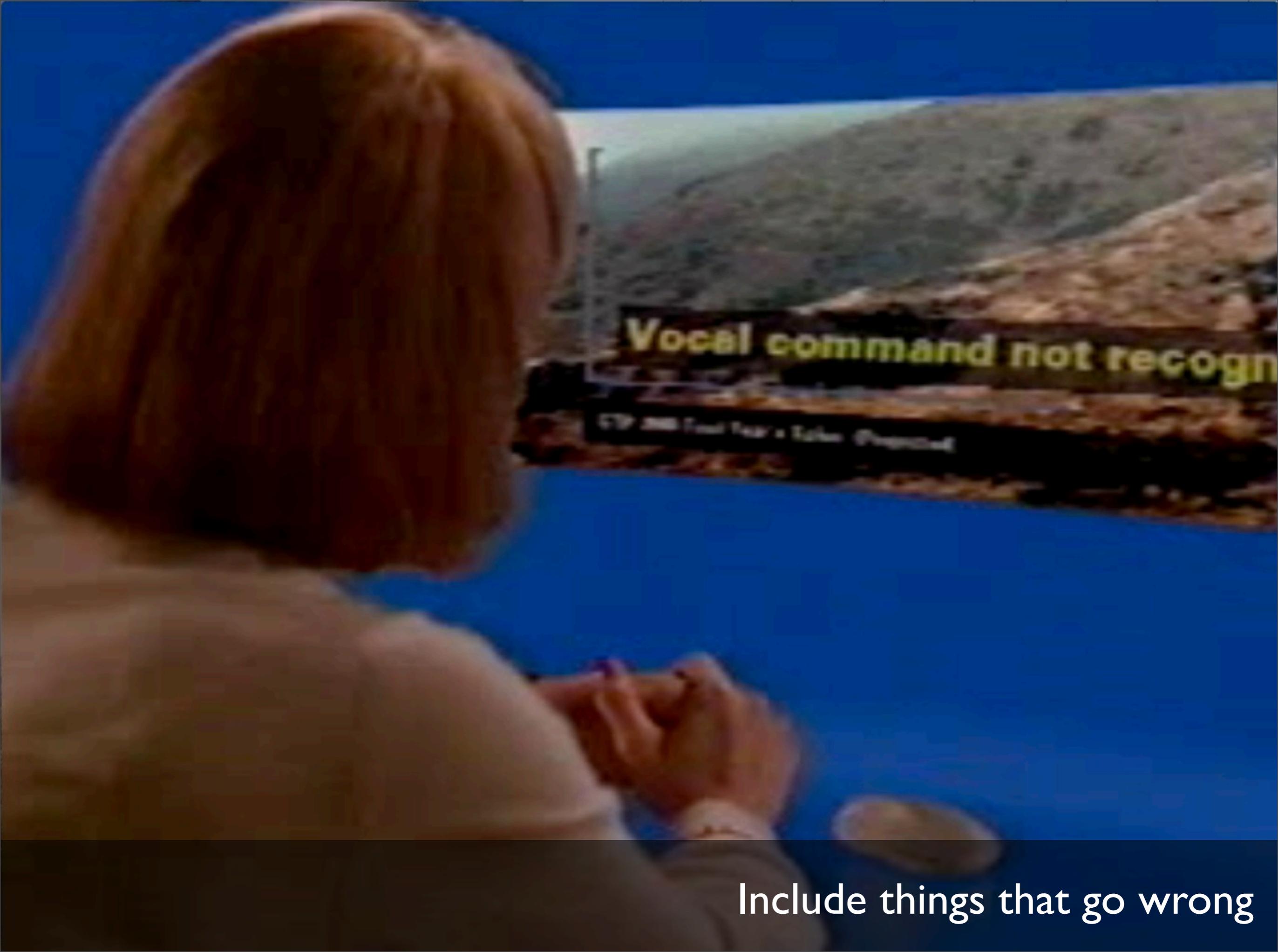


**More in DIS2!**

# Video Prototyping

- Visualize the behavior of a system
- Videotape short scenes of the user interacting with the system
- Cut together to tell the story
- Great for envisioning futuristic system
- Example: Sun's [Starfire](#), Apple's [Knowledge Navigator](#)





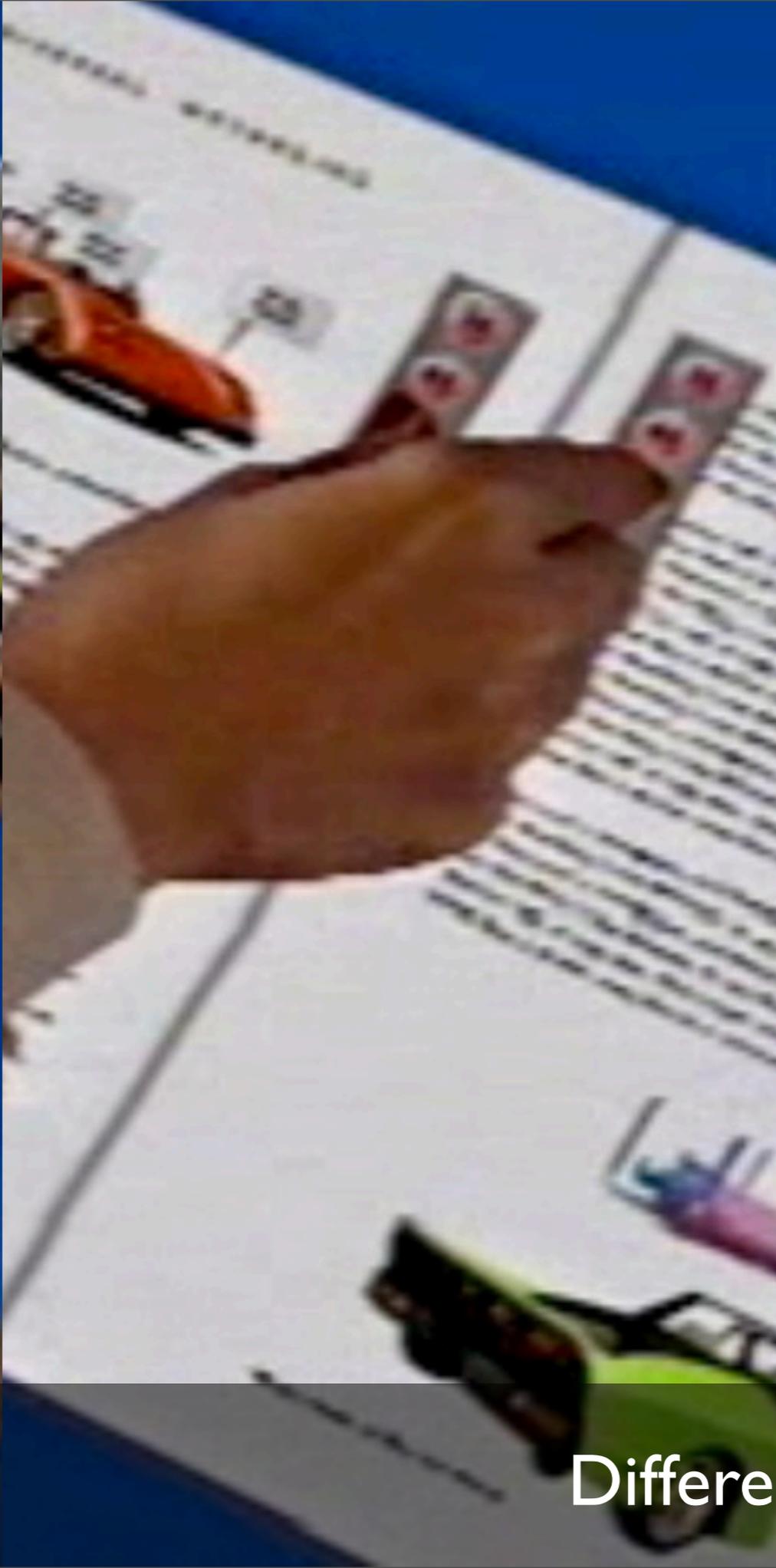
Include things that go wrong



Include things that go wrong



Use camera angle instead of implementing difficult interaction



Different input devices



*Away from my desk*



Discovering social issues during prototyping

# Starfire Prototyping Guideline

- Continuously question if assumptions are realistic within 10-year timeframe
- Iterative nature, like any other prototype
- Include things that go wrong
- Avoid impossible hardware designs
- Design interface first, then decide film scenes based on budget
  - E.g., Mouse, voice, reverse angle much cheaper than gesture and pen



# Ideation



# Styles of Thinking

- When thinking about a problem, we try to do too much at once
  - Emotion, information, logic, hope, creativity,...

⇒ Instead: Think in *one* style at a time!

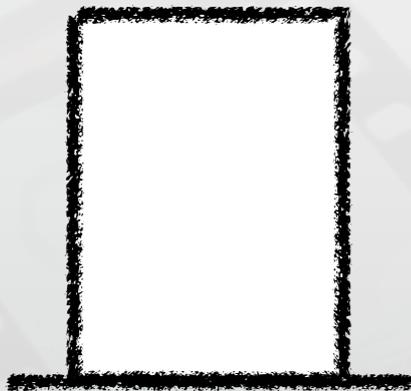
- Maximizes sensitivity of the brain in that direction
- Everybody has their own preferred styles of thinking
  - Correlated with personality, training, professional background, role, situation,...
- When people think in different styles, they argue

⇒ *Parallel thinking*:

- Let everybody think in the same style for a while
- Then move to the next style of thinking, to cover all styles

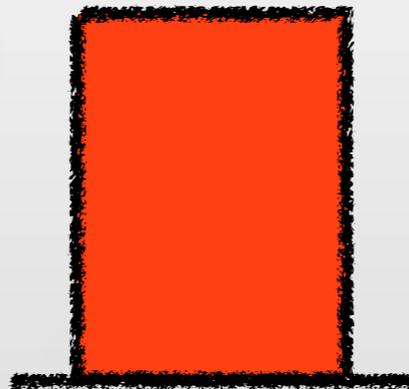


# Six Thinking “Hats” (Styles)



*Paper*

Objective, facts and figure



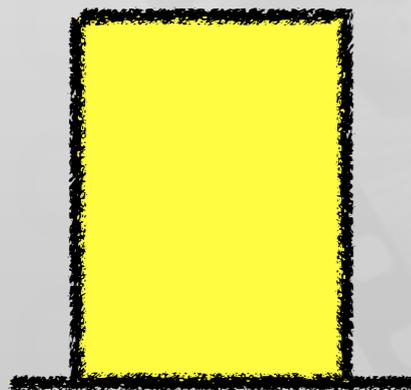
*Blood*

Intuition, gut feeling, emotion



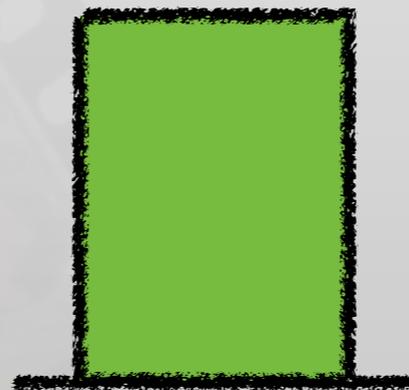
*Serious*

Cautious, critical



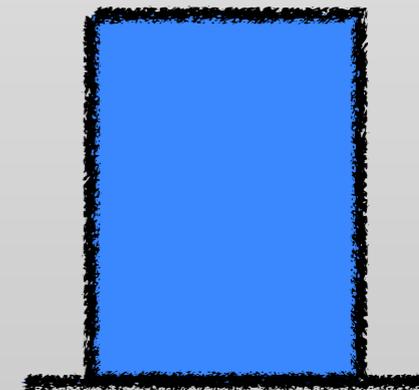
*Sun*

Hope, benefits, positive thinking



*Growing Grass*

Creativity, new ideas



*The Sky Above*

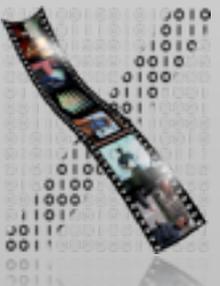
Organize other hats

[de Bono, 2001]



# Six Thinking Hats

- Use hats to refer to thinking styles instead of people
  - ✓ “That was good black hat thinking; now let’s put the yellow hat on.”
  - ✗ “You are too critical. You should see the benefits of this.”
  - ✗ “You are a black hat!”
- When to use which hat?
  - Preset: Determine hat sequence before meeting
  - Evolving: Determine next hat on-the-fly (not for beginners)



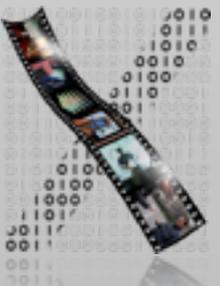
# Six Thinking Hats Guideline

- Only moderator can trigger hat changes
- Short time per hat (1 min per participant)
  - Extend when new things come up — do not limit creativity
  - Red hat: Keep time short. Make statements as definite as possible.
- Example sequence
  - Blue: organize the meeting and hats
  - Red (if there is a strong preexisting feeling): let people lay down emotional burden
  - White: bring everyone up-to-date with information
  - First Yellow, then Green, and then Black (benefits motivate people to overcome difficulties, get the ideas, criticize the ideas)
  - White: assess the idea against existing information
  - Blue: conclude and summarize
  - Red: reflect on thinking performance



# Your DISI Project

- Theme: “You make me want to be a better person”
  - Interactive system  $\Rightarrow$  persuades users  $\Rightarrow$  behavior change  $\Rightarrow$  improve quality of life
- Three directions
  - Go green
  - Go healthy
  - Go social
- Challenge: target users must not be university students between 20–30 years age group
  - Maximum grade without accepting the challenge: 2.0
  - Maximum grade with the challenge: 1.0
  - Groups of 4–6





Go green: use sustainable energy source

Image: <http://www.flickr.com/photos/30588268@N03/3576840442/>



Go green: use public transport

Image: <http://www.flickr.com/photos/mescon/3893805827>



Go green: use energy-efficient lightbulbs

Image: <http://www.flickr.com/photos/antonfomkin/5243218781>



Go healthy: exercise

Image: <http://www.flickr.com/photos/sbh/5127834263>



Go healthy: regular health checks

Image: <http://www.flickr.com/photos/seattlemunicipalarchives/4058808950/>



Go healthy: eat veggies

Image: <http://www.flickr.com/photos/vinothchandar/5612099123>



Go social: help others

Image: <http://www.flickr.com/photos/yourdon/2906764434>



Go social: be physically together

Image: <http://www.flickr.com/photos/pocketwiley/2910495143>



Go social: bridge age gap

Image:<http://www.flickr.com/photos/sashapo/5547805558>

# Brainstorming

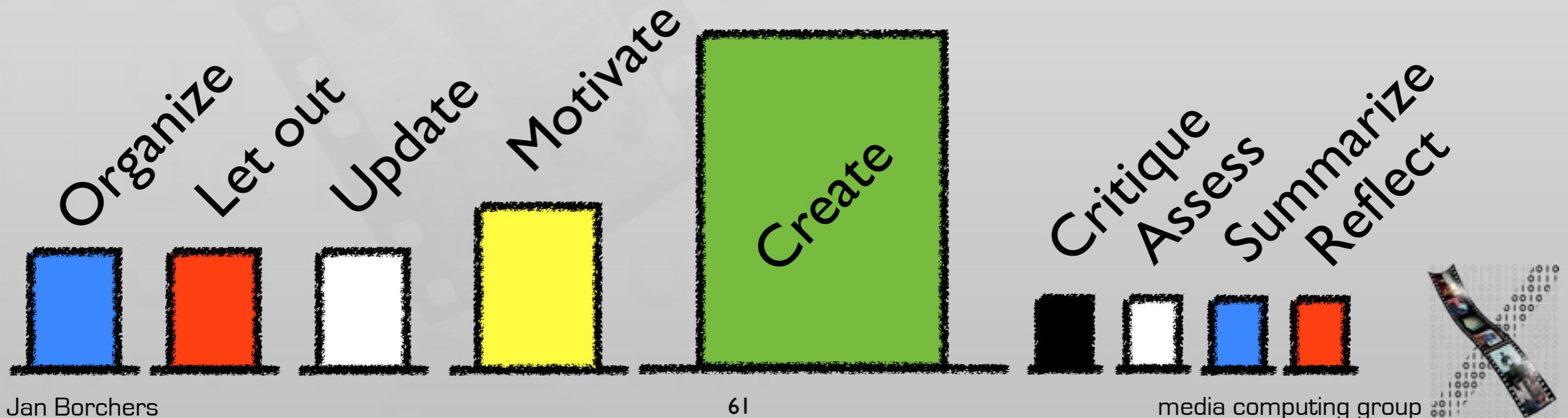
## An Initial Design Technique

- Goal: Collect as many ideas on a given topic as possible
  - Quantity, not quality; include crazy ideas
  - Go for a certain number of ideas, say, 100
- Defer judgment, don't criticize or argue (no black hat)
  - Instead, leapfrog on each other's ideas (green hat)
- Limit to 5–10 minutes
- Relax, have fun, invite good brainstormers
- Scribe collects ideas visible for all
- Trick: Cross-pollination who–what–where



# In-Class Exercise: Brainstorming

- Project Theme: “You make me want to be a better person”
- Brainstorm on
  - What behaviors could change to improve quality of life?
  - How to persuade users to change?



# Structuring Brainstorms: Concept Mapping

- Used since 1500s by Spanish monks
  - **Mind Mapping** trademarked by Tony Buzan in 70's
- Uses both brain sides, structures note-taking for overview, planning, learning... with a visual “gestalt”
  - Use A3 landscape, subject in middle, aspects on branches, subtopics on subbranches (software?)
  - Connect additional relationships with arrows
  - Use images/icons for keywords where they work
  - Use color for branches & connections (after pencil version becomes stable)
- Grows over time, combine individual maps





# Summary

- Gestalt laws allow us to leverage human perception in visual layout design
- Different software prototyping tools support different purpose of prototyping
- Six Thinking Hats and brainstorming allow early design ideas to be explored effectively

