CS160

USER INTERFACE DESIGN

SPRING 2016

HEURISTIC EVALUATION

10 FEB 2016

ERIC PAULOS
www.paulos.net

UNIVERSITY OF CALIFORNIA Berkeley

ANNOUNCEMENTS

Fill out FEED 01 (Due Fri)

Please don’t remove chairs from side meeting rooms

Back to Google OnAir Hangouts

DSP Emails

Brainstorming Class MANDATORY ATTENDANCE — 22 Feb

SECTION: Android Wear Intro + Figma + Heuristic Evaluation

DESIGN 02 - Heuristic Evaluation

PROG 02

PROG 02: REPRESENT!

Caucus, Primary, Debate, Delegates, Issues! How can we find the signal in all the noise and hype as we approach the 2016 US Presidential Election? You have been tasked to help design a groundbreaking mobile and smartwatch application to deliver facts to voters on the go.

This is a three phase assignment. As such it is staged across three different deliverables. In brief, the first phase will allow you to explore a broad design space for your application, delivering your design as wireframes. The second phase moves from these wireframes to actual code focusing on the phone to watch communication. The final phase uses APIs to tie the whole application together.

PHASE A - DUE 17 FEB @ 2:30PM (ONE WEEK)

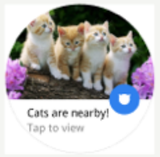
In the first phase of this assignment, you will need to generate wireframes for each of the screens you will need. These should be at a reasonable fidelity that we can get a sense of your visual design choices and interaction flow. You are not locked into this exact design: if you later decide to make some changes to your design, that's ok.

We want you to use Figma to quickly layout your design as you envision it now based on the description and your design ideas. The election is approaching fast and there is, unfortunately, less time than usual to iterate. Therefore, you should start to make some initial design considerations for your app in terms of text, images, colors, background, layout, etc. While you will want to start sketching with pen and paper, what you deliver to the client (i.e. us), should be somewhere between a low-fidelity mock up and a high-fidelity design. Basically, not just boxes, but clear design choices on some of the other visual and interaction design elements.

Key deliverables:

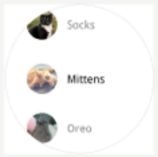
- Mock up of the interaction flow of the app on watch and phone using Figma showing all of the primary screens as a PDF handed in via bCourses.
- Confirmation that your watch and phone emulator are working by taking a video of you sending a notification to the watch from the Android Wear phone app. See the image below.Phase A Grading
- Are all of the screens rendered? (2 points)
- Are the designs shown using adequate fidelity? (3 points)
- Does the video demonstrate that your watch and phone emulator are properly setup and running? (5 points)
- Does your design make good use of visual elements (fonts, layout, colors, icons, etc? (5 points)
- Does your design have an intuitive, easy-to-use interaction flow? (5 points)

Notification when at location



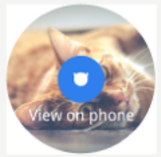
Tap

View list of nearby cats (populated based on location data)



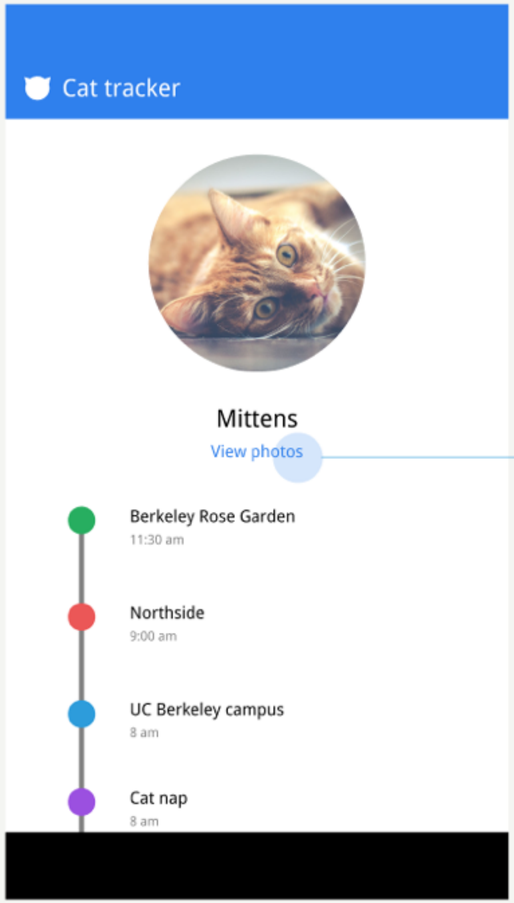
Tap

Single view for selected cat that prompts for photo info



Tap

Phone detail view of selected cat that shows where cat has been over time



Tap

Photo gallery view

Tap

Single photo detail view

PHASE B - DUE 2 MAR @ 2:30PM (TWO WEEKS)

Code the wireframes from Phase A into functional Android code.
Implement the code to render the screens you designed in Phase A.
They may not be exactly the same visually but they should be reasonable approximations. In Phase B you will focus on the communication between the watch and the phone. You will be expected to hand in code and a video that demonstrates correct communication and a callback between the watch and phone. For example, selecting a Senator on the watch should call up the corresponding Senator's detailed view on the phone.

Key deliverables:

- Code and Video demonstrating correct functionality of watch and phone interaction as well as basic screen layouts in the Android emulator.

PHASE C - DUE 11 MAR @ 11:59PM (9 DAYS)

Phase C - In the final phase you will add code to properly interface with APIs that will bring the application to life with real live data. This includes the ability to lookup the members of congress based on zip code or the phone's current location, the committees they serve on, the bills the sponsor, their last Tweet, etc. You will also code the functionality to capture the 2012 election data by location.

Key deliverables:

- Code and Video demonstrating correct functionality of watch and phone interaction as well as APIs in action in the Android emulator.

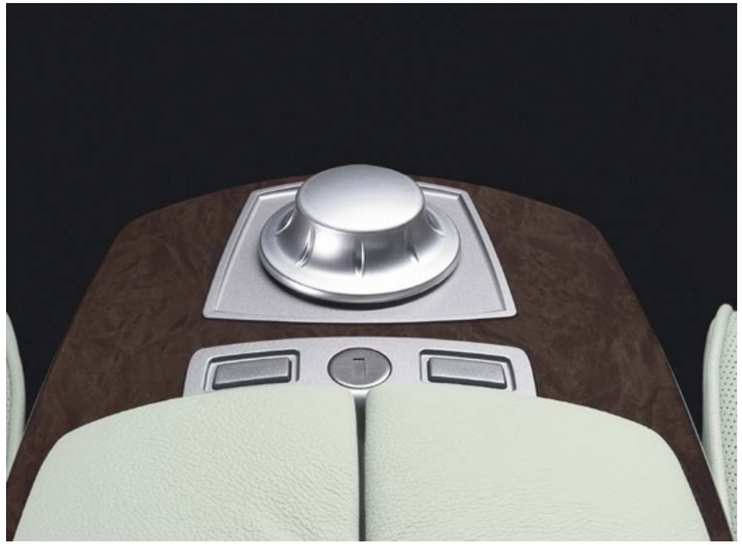
Phase A Grading

- Are all of the screens rendered? (2 points)
- Are the designs shown using adequate fidelity? (3 points)
- Does the video demonstrate that your watch and phone emulator are properly setup and running? (5 points)
- Does your design make good use of visual elements (fonts, layout, colors, icons, etc? (5 points)
- Does your design have an intuitive, easy-to-use interaction flow? (5 points)

DESIGN PRINCIPLES

1. MAKE CONTROLS VISIBLE

POOR VISIBILITY (BMW'S IDRIVE)



How do you put someone on hold?

How do you set the alarm?


SONY
FM II
ST
LCL
LOUD
102.7
3
FM/AM CASSETTE CAR STEREO
DIR
DSPL
ATA
1
2
3
4
5
6
LCL
RELEASE
35W×4
XR-1890
A MEM
–
SEL
+
–
SEEK
MANU
+
LOUD
TUNER
OFF

Primary controls are visible

But how to set a radio station preset?

TOO MUCH VISIBILITY?

6 remote controls for “modest” home theater

2. MAKE SURE MAPPING IS CLEAR

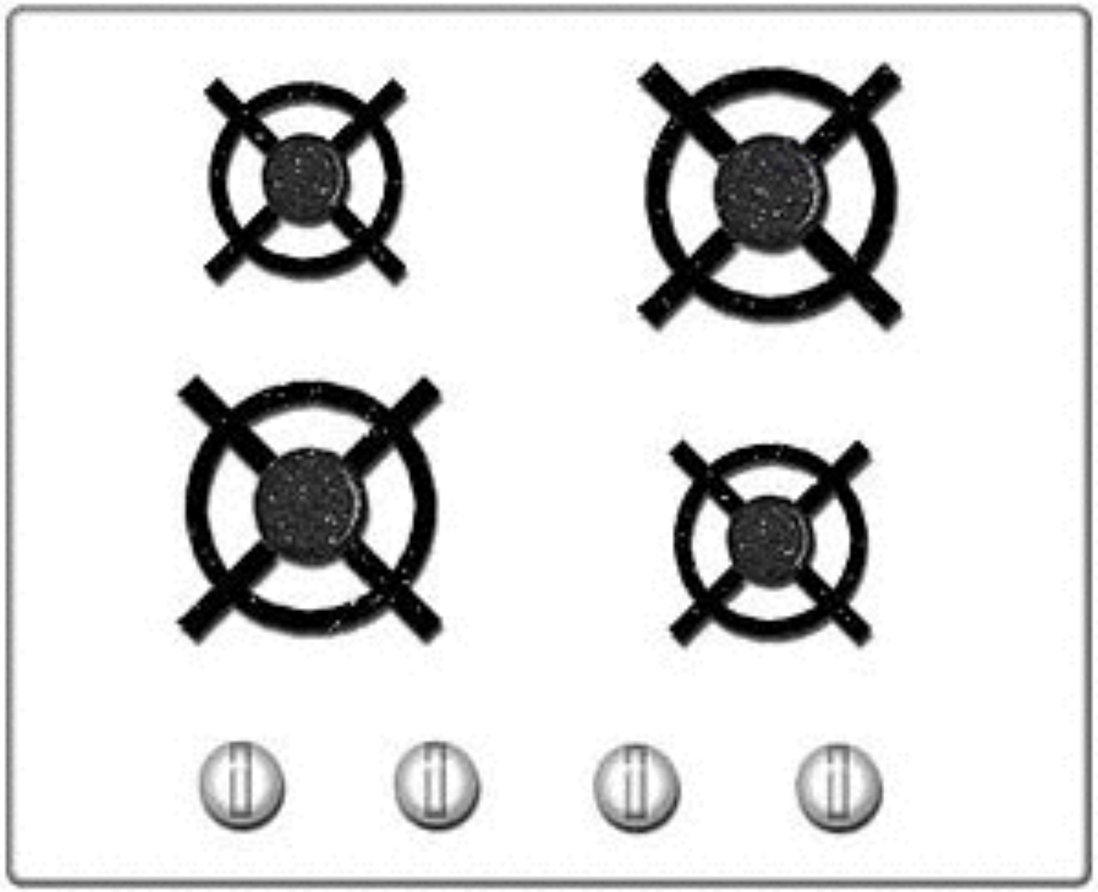
Mapping: Relationship between controls and their result

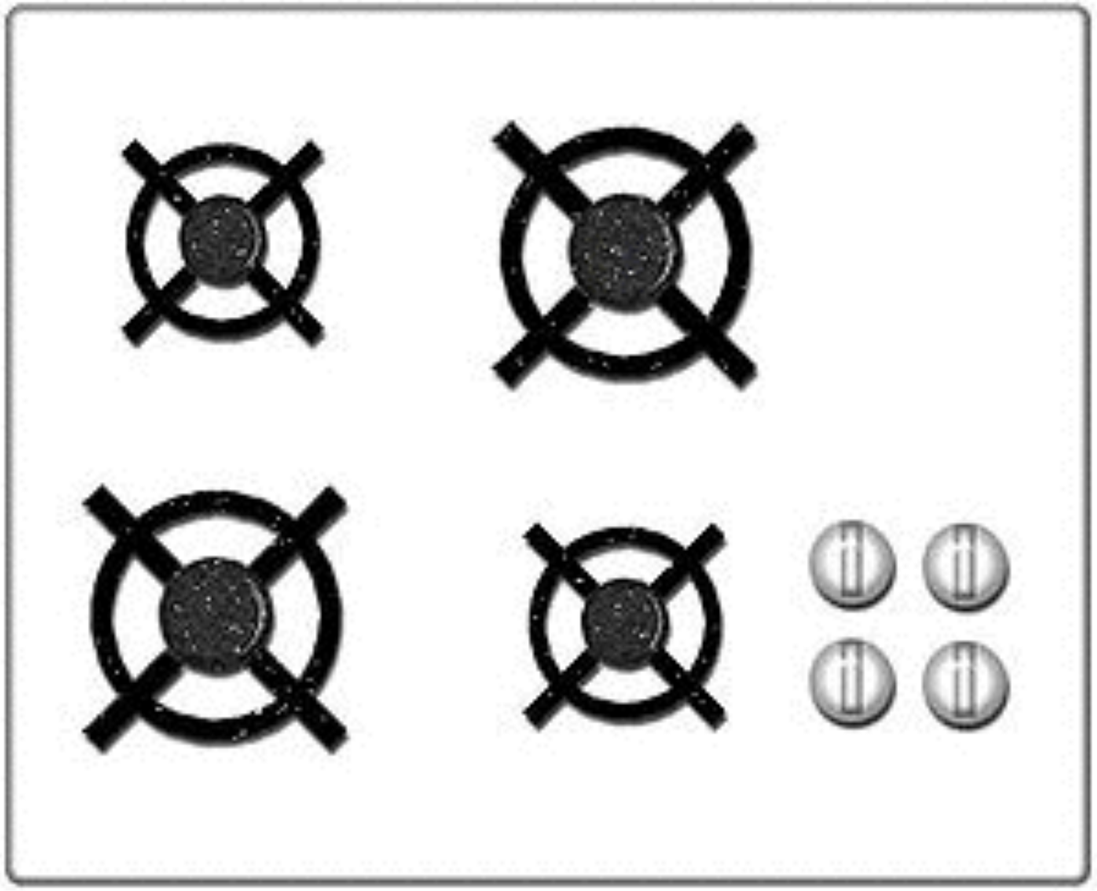
Mercedes S500 Car Seat Controller


LOAD
mp3
EJECT
FM
AM
CD
SAT
CAT FLDR
SEEK TRACK
ON
VOL
BASS
MIDDLE
TREBLE
BALANCE
FADER
SCAN
TUNE
AC
72
64
80
0
1
2
3
4

Which way will the sound be moved when you turn this knob?

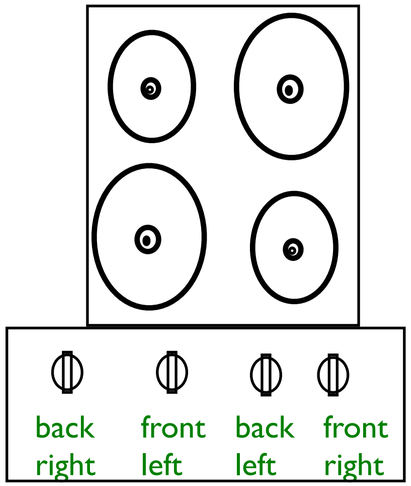
STOVETOP CONTROLS





STOVETOP CONTROLS

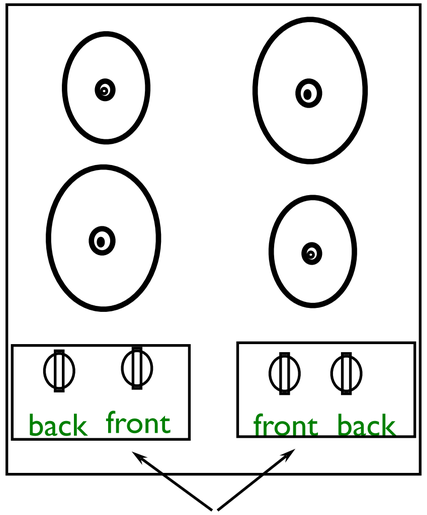
arbitrary



24 possibilities, requires:

visible labels
memory

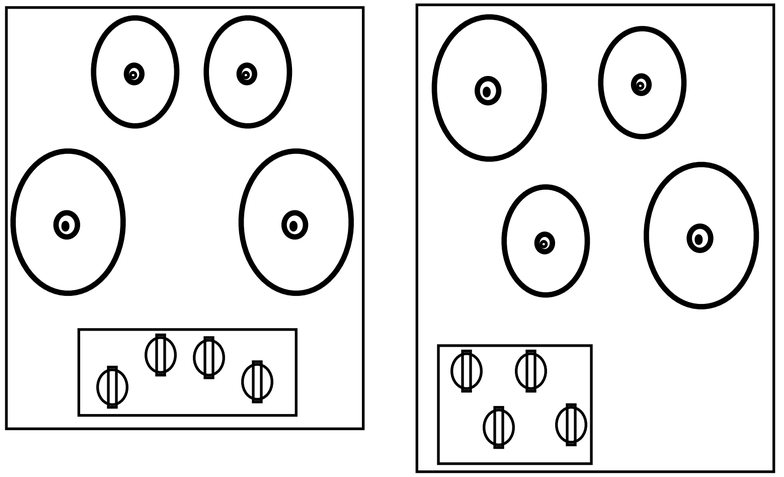
paired



2 possibilities per side

=4 total possibilities

full mapping



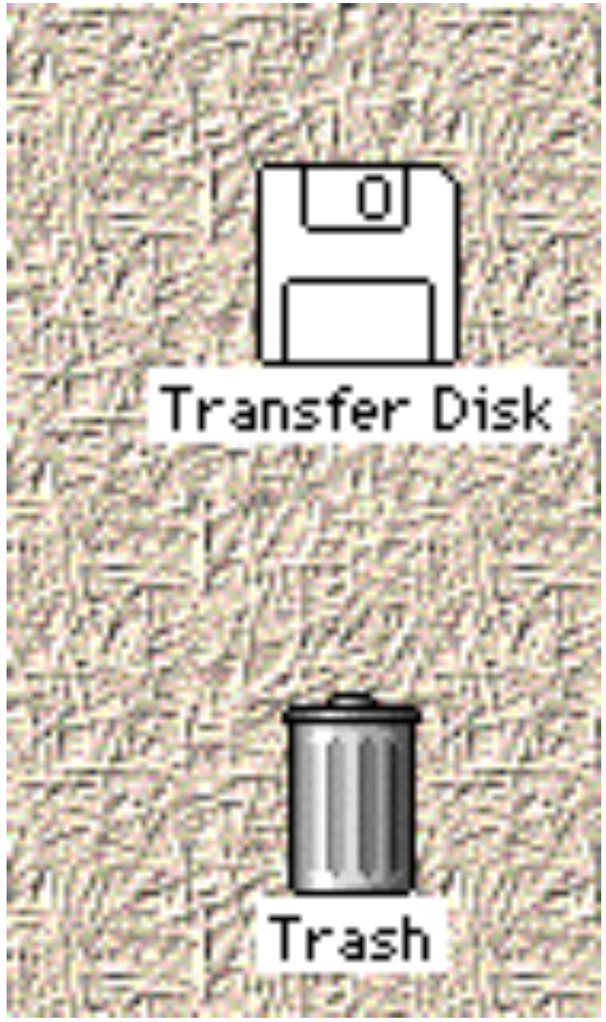
TRANSFER EXPECTATIONS

From known objects to similar new ones

Positive: previous experience applies to new situation

Negative: previous experience conflicts with new situation

What happens when disk is dragged onto trash can?



Crown in
Apple Watch

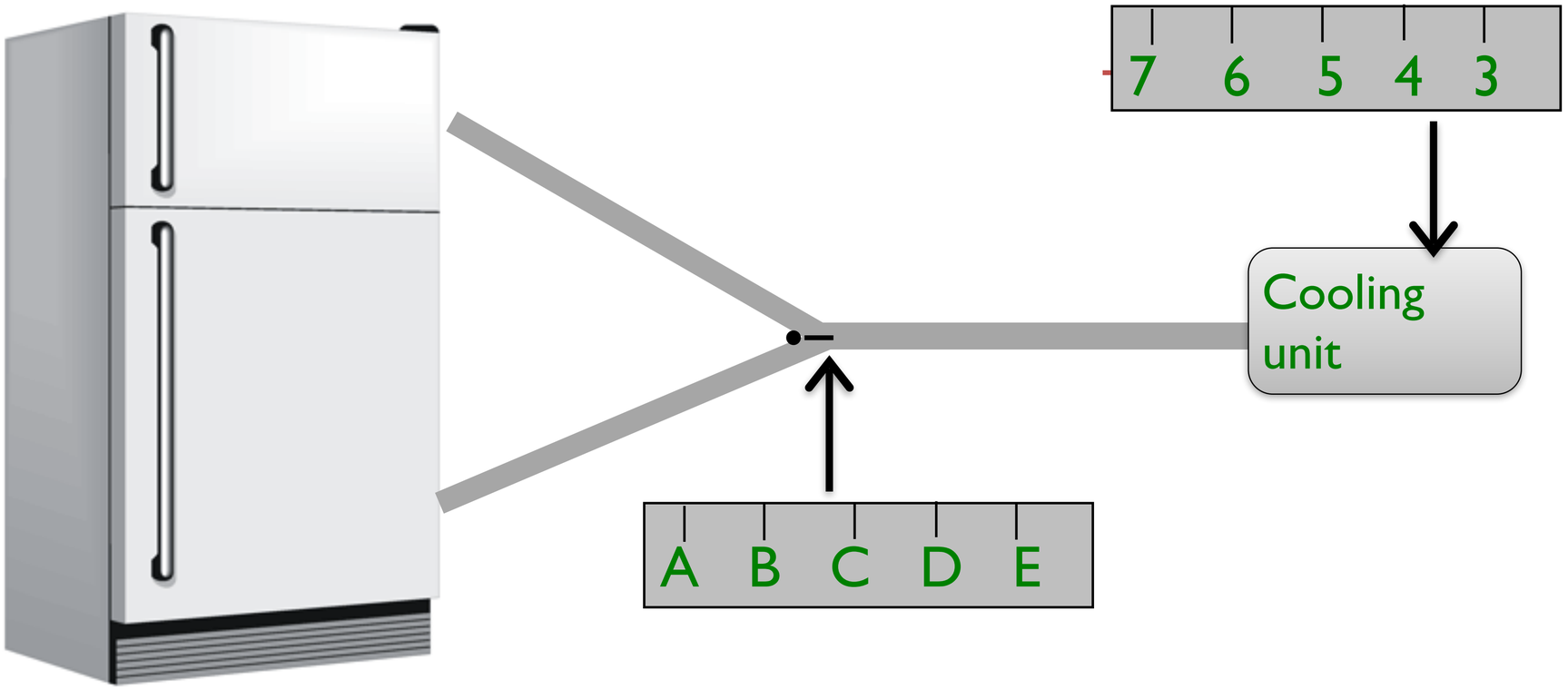
3. PROVIDE FEEDBACK

People press >> 1 time

Unclear if system has registered the button press

Elevator buttons light up → reduces multiple presses

POOR FEEDBACK



Took a day for refrigerator to adjust to new settings

HEURISTIC EVALUATION

USABILITY HEURISTICS

"Rules of thumb" describing features of usable systems

Can be used as design principles

Can be used to evaluate a design

Example: Minimize users' memory load

HEURISTIC EVALUATION

Developed by Jakob Nielsen (1994)

Can be performed on working
UI or on sketches

Small set (3-5) of evaluators (experts) examine UI

Evaluators check compliance with usability heuristics

Different evaluators will find different problems

Evaluators only communicate afterwards to aggregate findings

Designers use violations to redesign/fix problems

NIELSEN'S TEN HEURISTICS

H1: Visibility of system status

H2: Match system and real world

H3: User control and freedom

H4: Consistency and standards

H5: Error prevention

H6: Recognition rather than recall

H7: Flexibility and efficiency of use

H8: Aesthetic and minimalist design

H9: Help users recognize, diagnose, recover from errors

H10: Help and documentation

H-1: VISIBILITY OF SYSTEM STATUS

Keep users informed about what is going on. Example: response time

0.1 sec: no special indicators needed

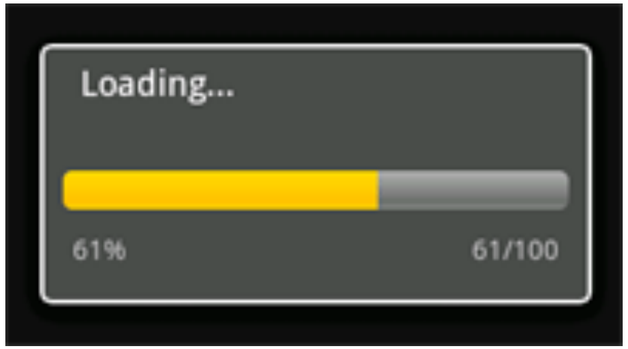
1.0 sec: user tends to lose track of data

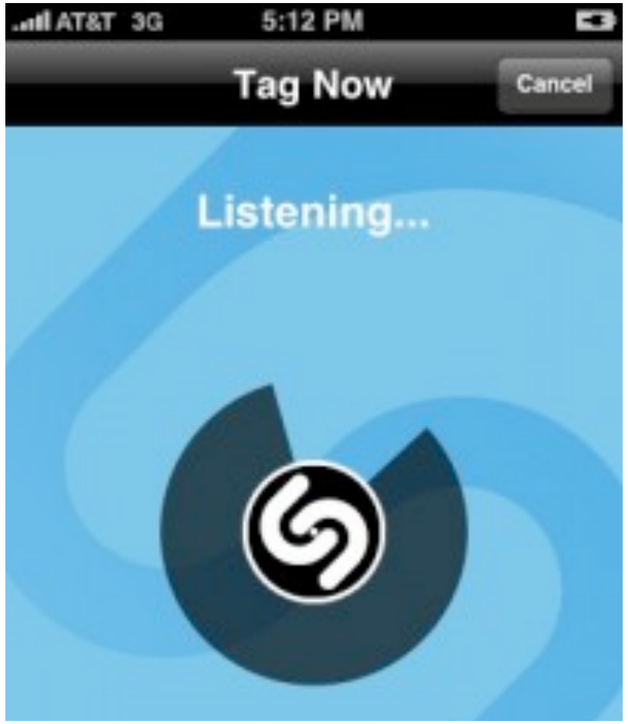
10 sec: max. duration if user to stay focused on action

Short delays: Hourglass

Long delays: Use percent-done progress bars

Overestimate usually better



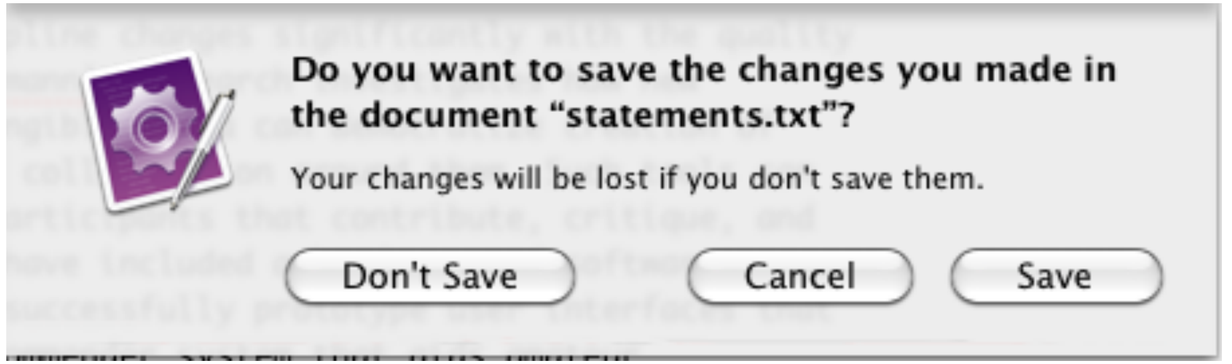


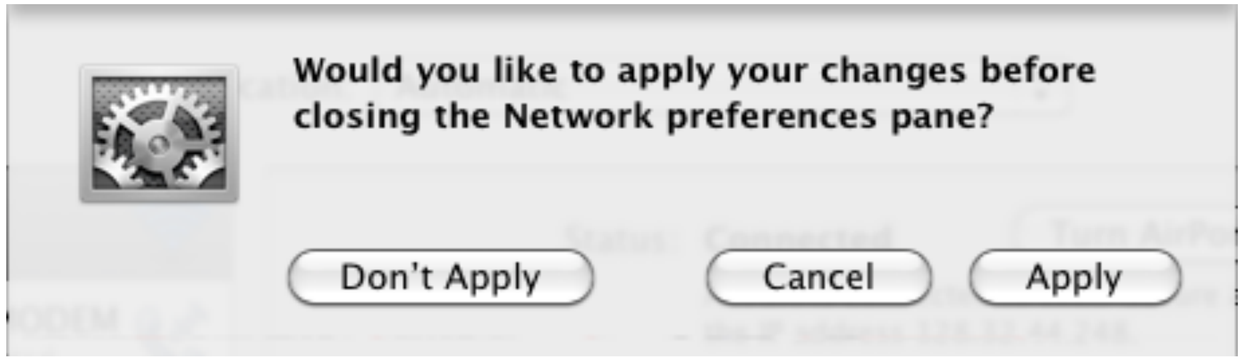
H-1: VISIBILITY OF SYSTEM STATUS

Users should always be aware of what is going on

So that they can make informed decision

Provide redundant information





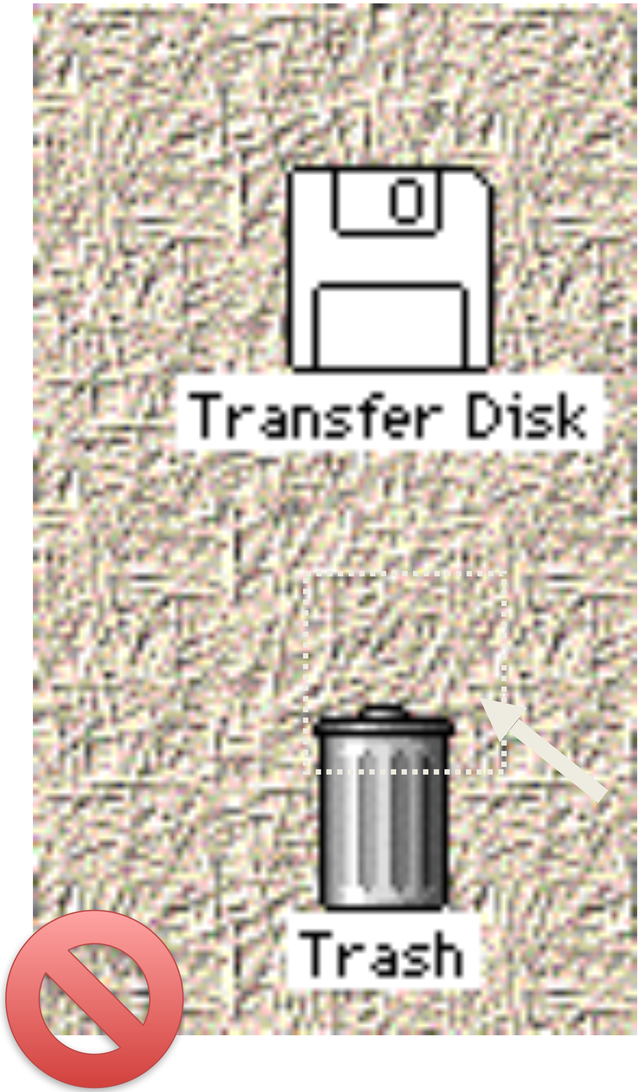
H-2: MATCH SYSTEM & WORLD

Speak the users' language

Follow real world conventions

Pay attention to metaphors

Bad example: Mac desktop



H2-2: MATCH SYSTEM & WORLD

H-3: USER CONTROL & FREEDOM

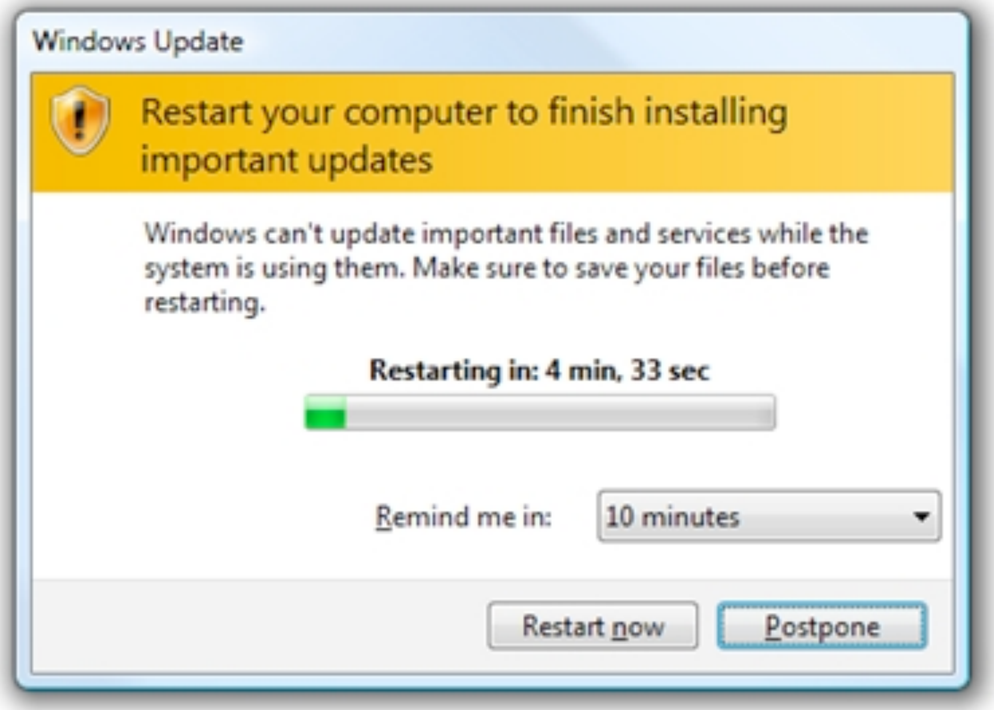
Users don’t like to be trapped!

Strategies

Cancel button
(or Esc key) for dialog

Make the cancel button responsive!

Universal undo

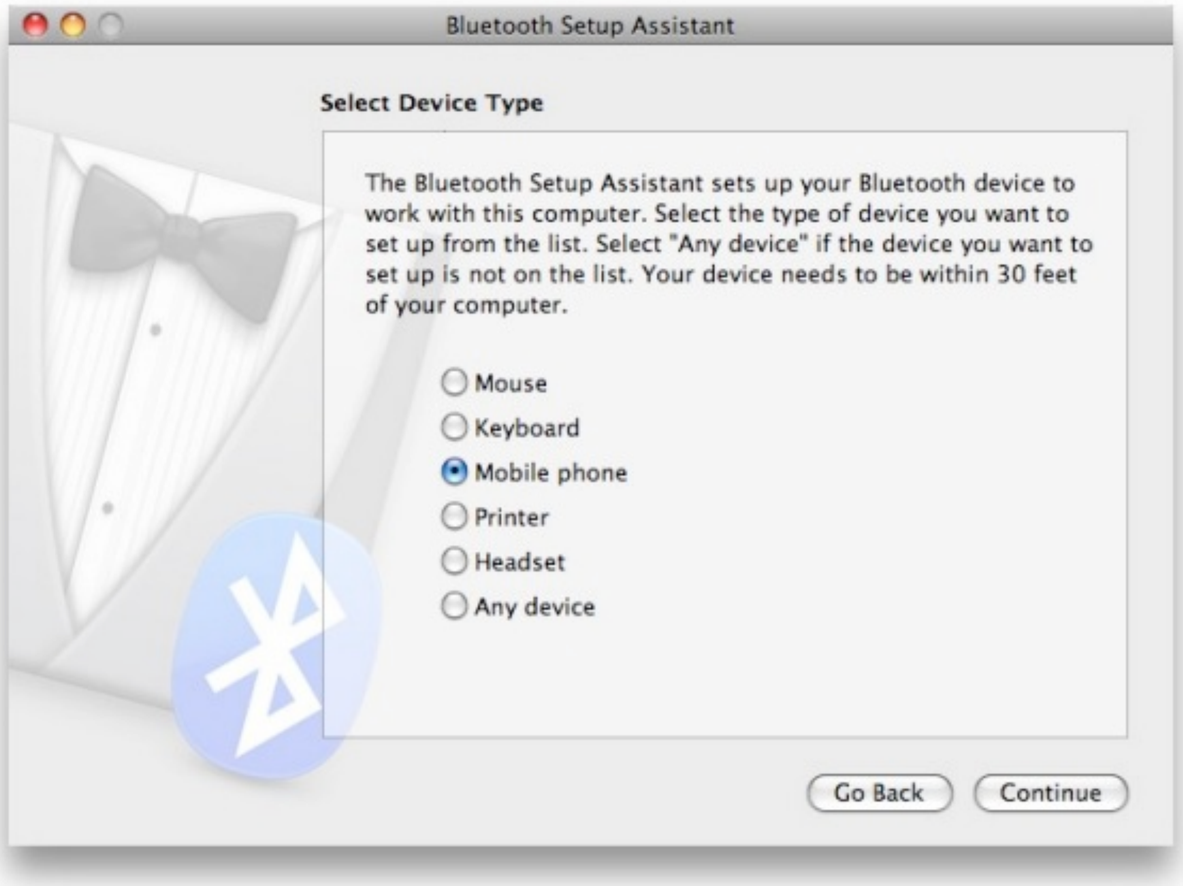


H-3: USER CONTROL & FREEDOM

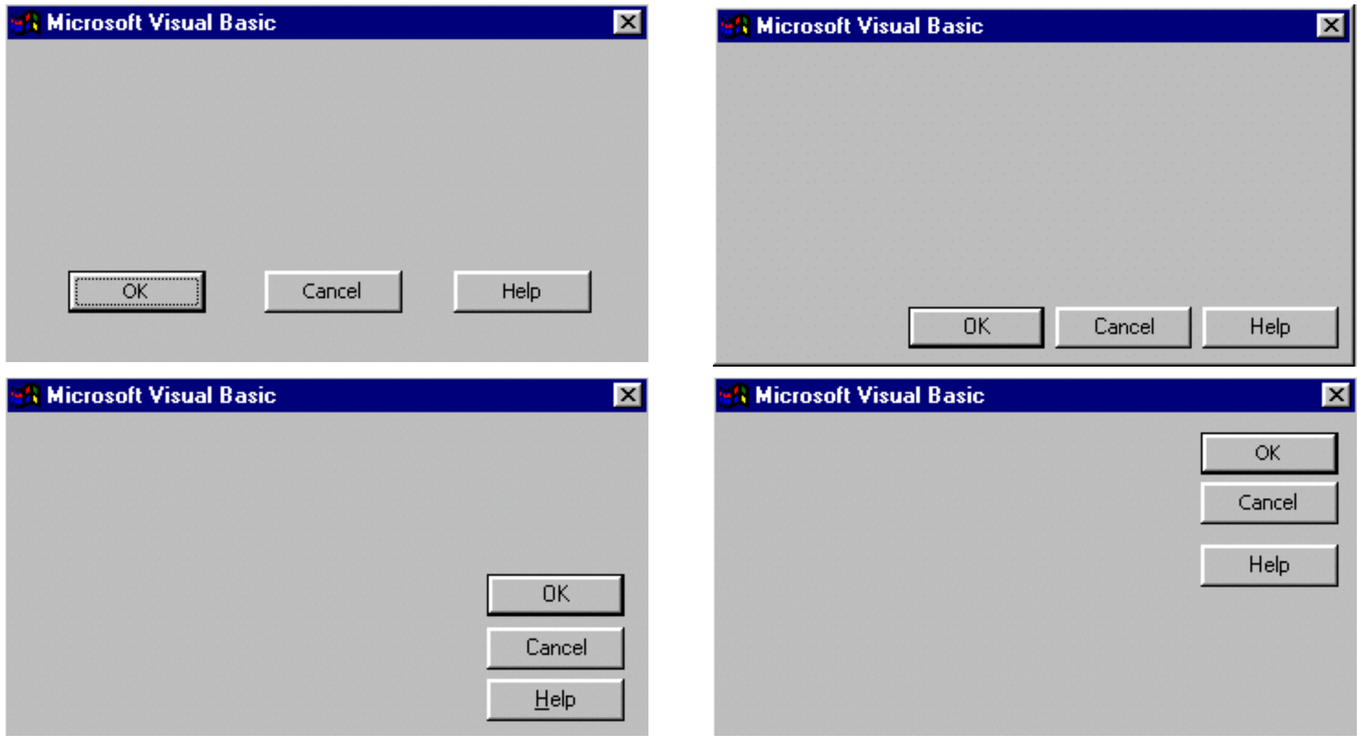
Offer “Exits” for mistaken choices, undo, redo
Don’t force the user down fixed paths

Wizards

Must respond to Q before going to next step
Good for infrequent tasks (e.g., network setup) & beginr
Not good for common tasks (zip/unzip)

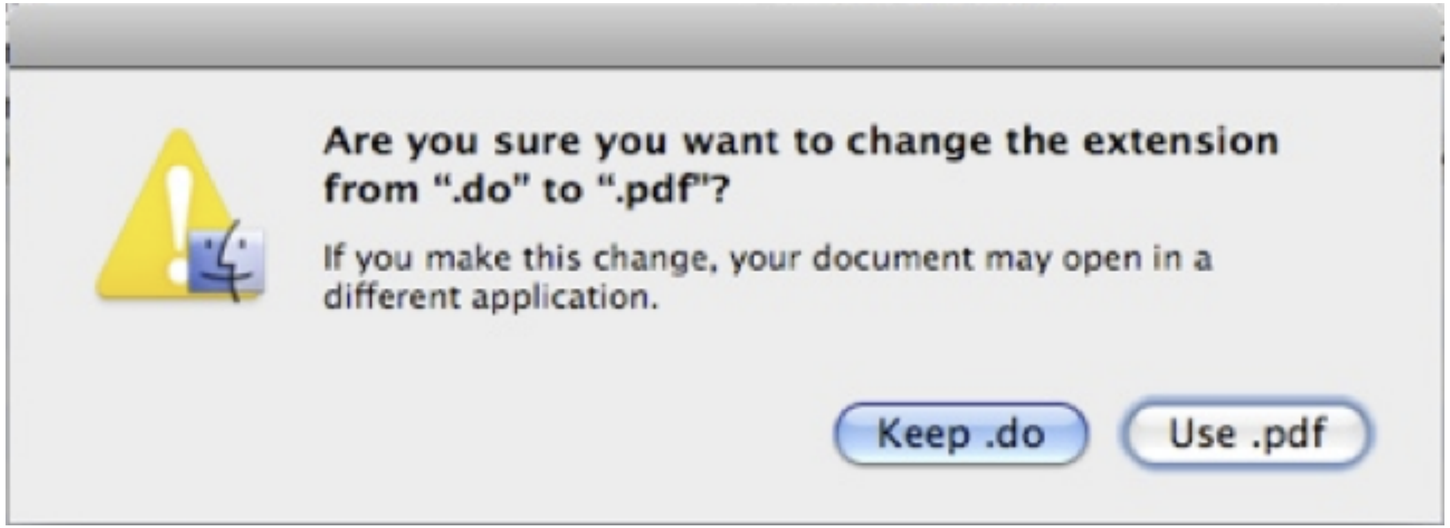


H-4: CONSISTENCY AND STANDARDS



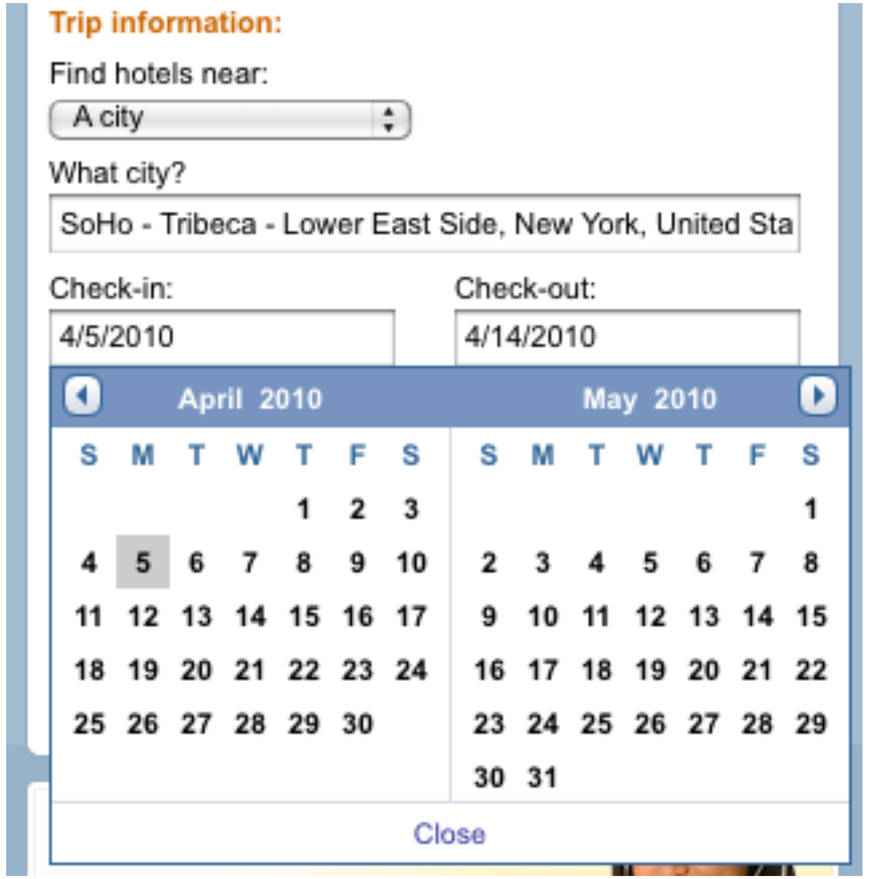
H-5: ERROR PREVENTION

Eliminate error-prone conditions or check for them and ask for confirmation

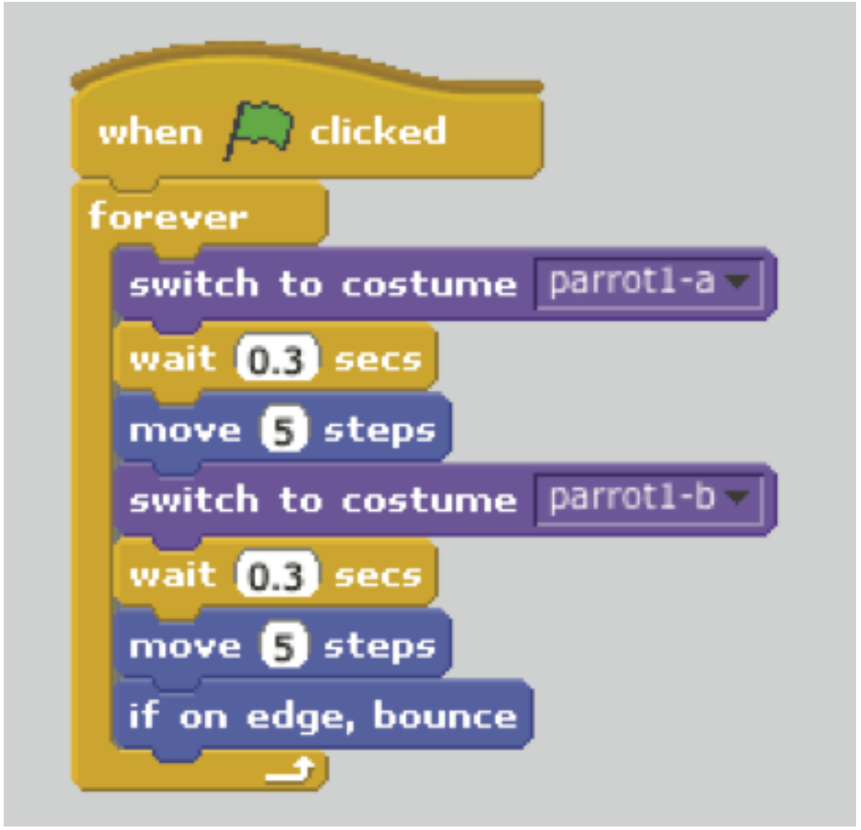


H-5: ERROR PREVENTION

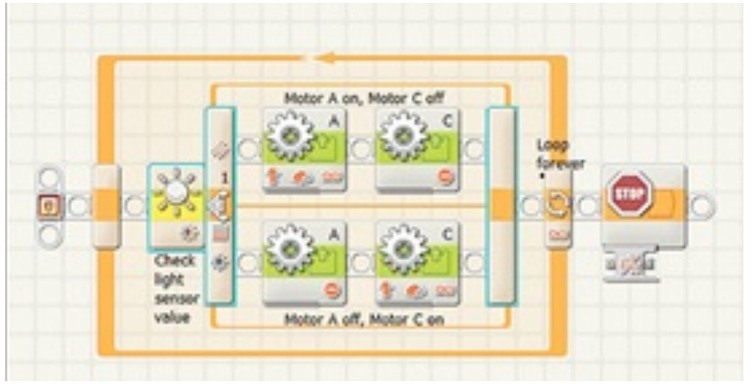
Aid users with specifying correct input



H2-5: ERROR PREVENTION



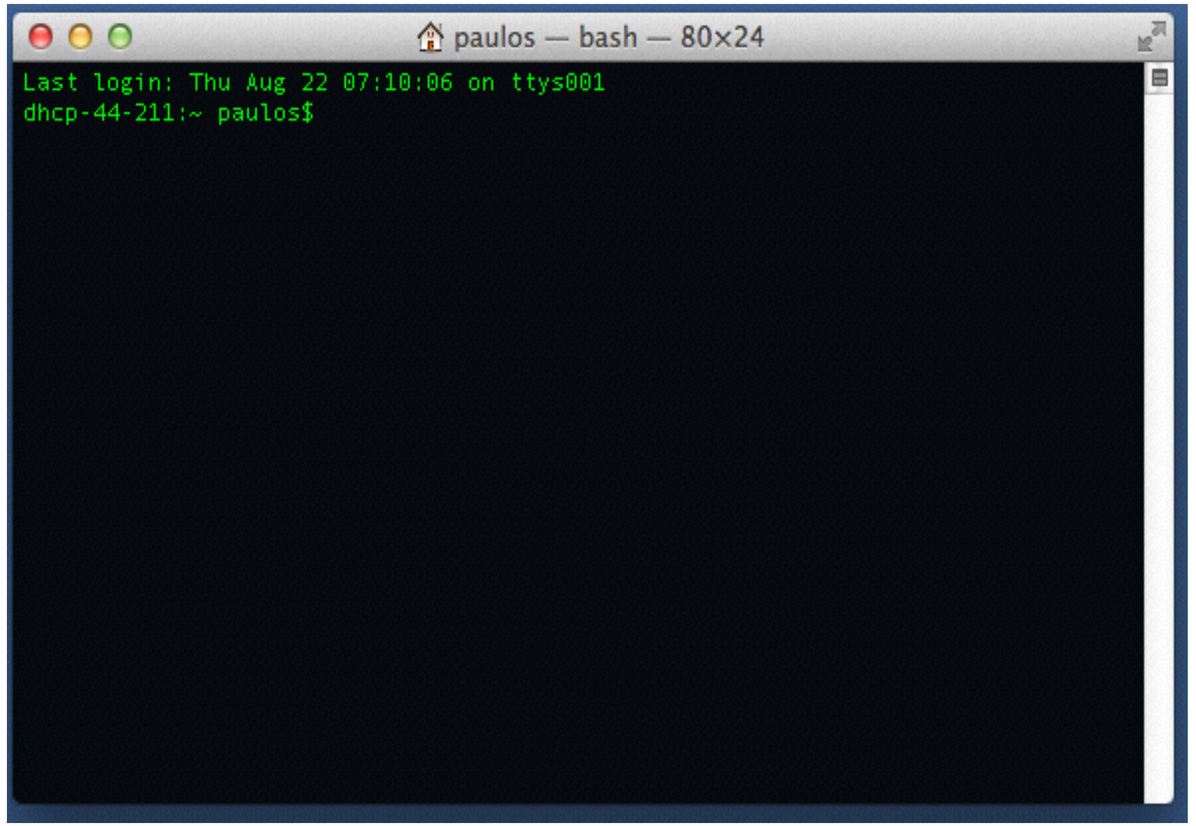
MIT Scratch



Lego Mindstorms

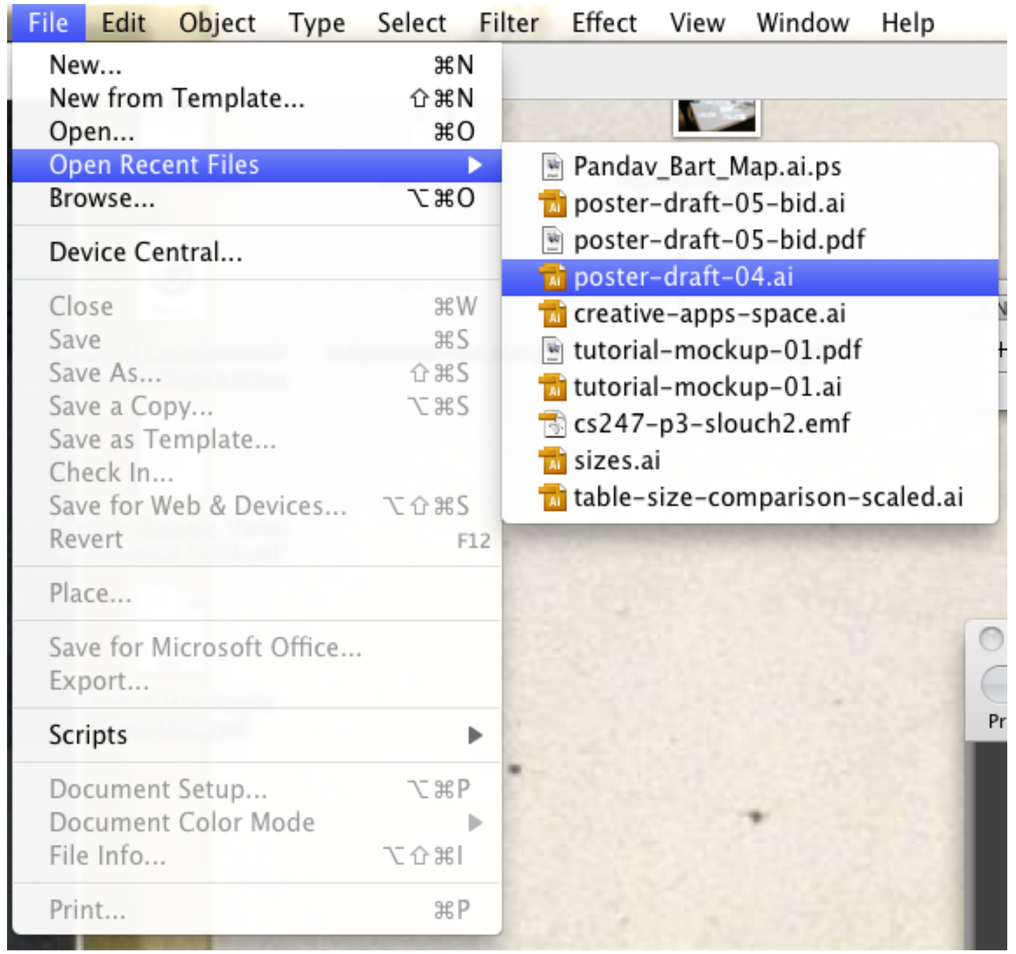
Don’t allow incorrect input

H-6: RECOGNITION OVER RECALL

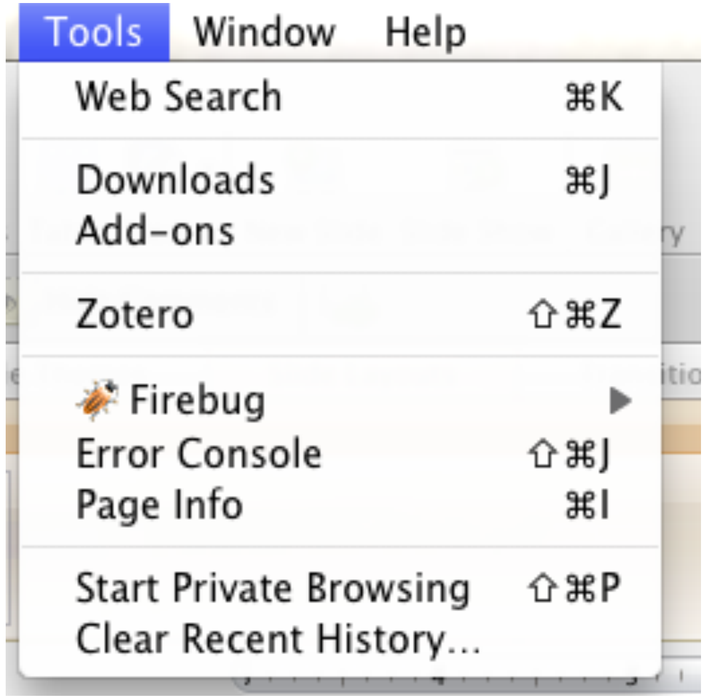


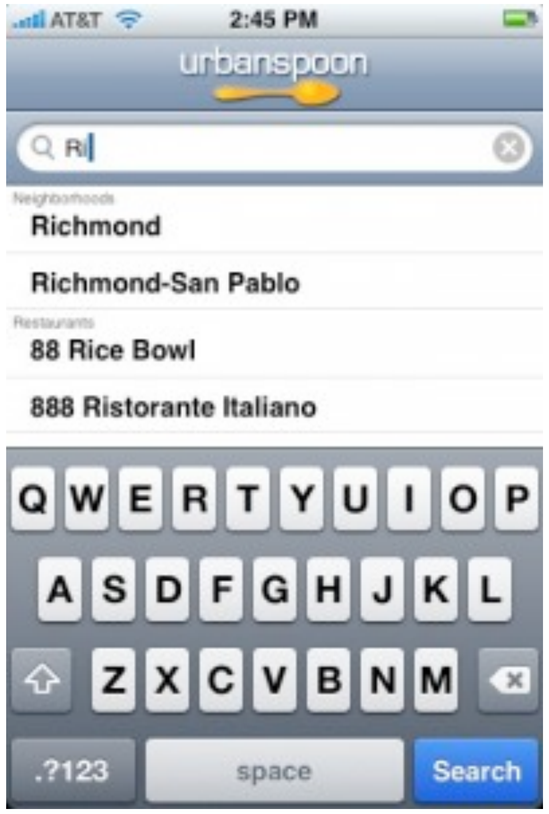
H-6: RECOGNITION OVER RECALL

Minimize the user's memory load by making objects, actions, and options visible.



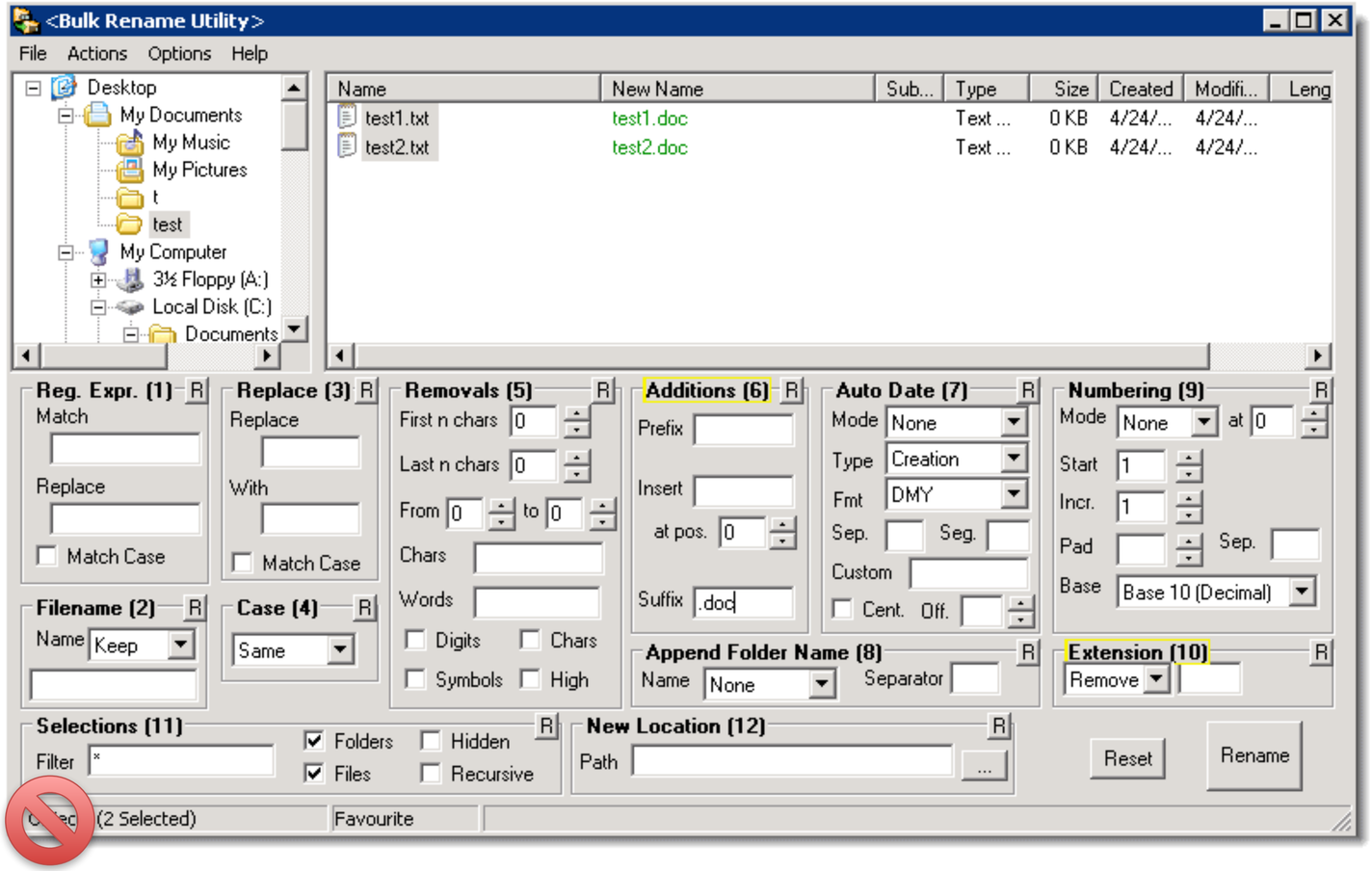
H2-7: FLEXIBILITY AND EFFICIENCY OF USE





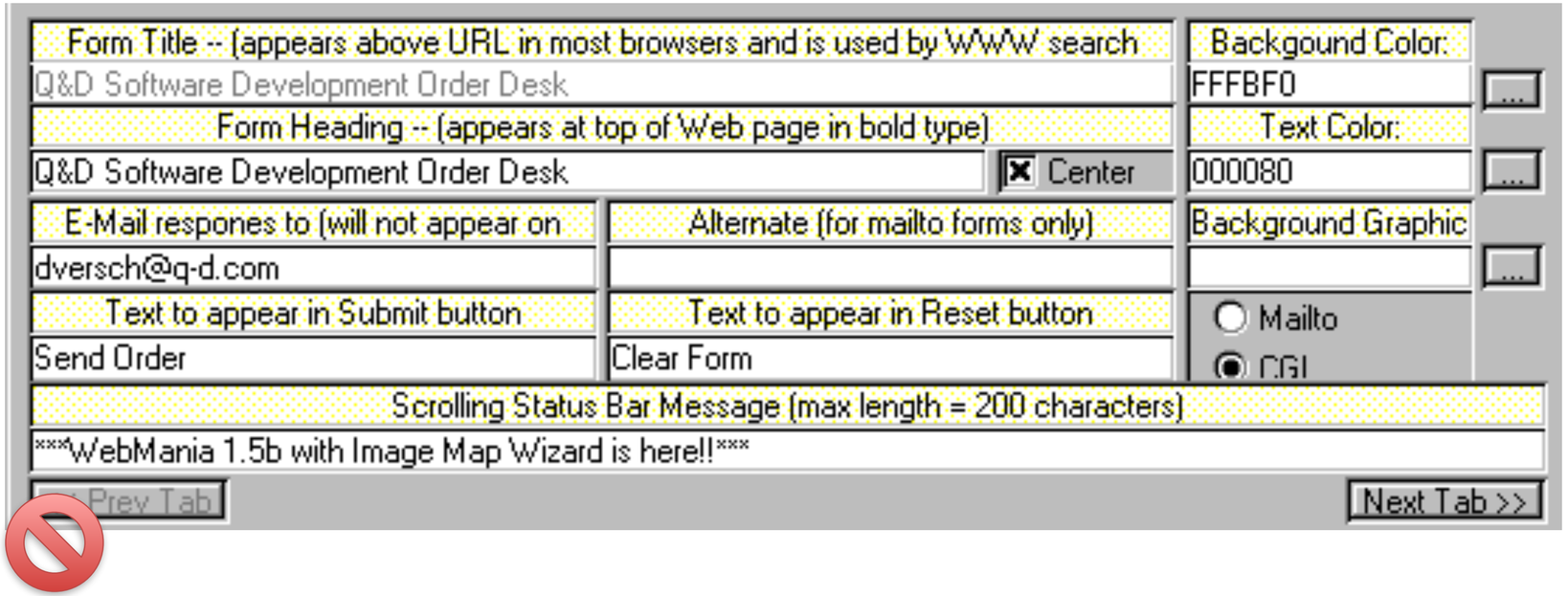
http://www.iphoneuxreviews.com/?p=114

H-8: AESTHETIC AND MINIMALIST DESIGN



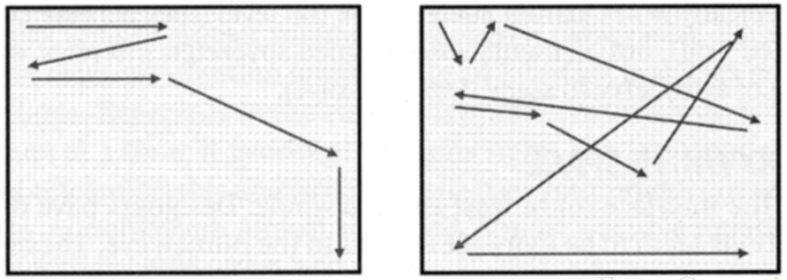
H-8: AESTHETIC AND MINIMALIST DESIGN

No irrelevant information in dialogues



H-8: AESTHETIC AND MINIMALIST DESIGN

Present information in natural order



From Cooper's "About face 2.0"

Occam's razor

Remove or hide irrelevant or rarely needed information – They compete with important information on screen

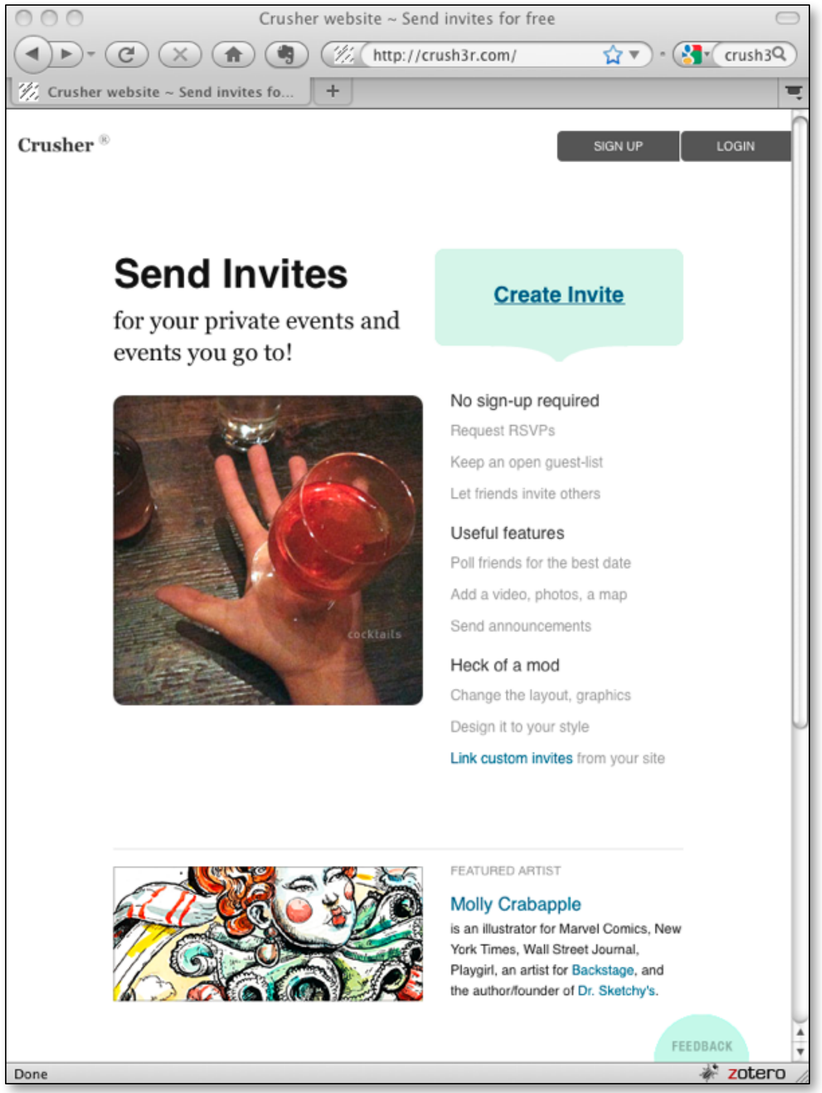
Pro: Palm Pilot

Against: Dynamic menus

Use windows frugally

Avoid complex window management

H-8: AESTHETIC AND MINIMALIST DESIGN



phpMyAdmin

Database

_mt_db (49)

bjoern_mt_db (49)

- b_book
- member
- mt_asset
- mt_asset_meta
- mt_association
- mt_author
- mt_author_meta
- mt_blog
- mt_blog_meta
- mt_category
- mt_category_meta
- mt_comment
- mt_comment_meta
- mt_config
- mt_entry
- mt_entry_meta
- mt_fileinfo
- mt_ipbanlist
- mt_log
- mt_notification
- mt_objectasset
- mt_objectscore
- mt_objecttag
- mt_permission
- mt_placement
- mt_plugindata
- mt_role
- mt_session
- mt_tag
- mt_tbping
- mt_tbping_meta
- mt_template
- mt_templatemap
- mt_template_meta
- mt_touch
- mt_trackback
- mt_ts_error
- mt_ts_exitstatus
- mt_ts_funcmap
- mt_ts_job
- phpwiki_link
- phpwiki_nonempty
- phpwiki_page
- phpwiki_rating
- phpwiki_recent
- phpwiki_session
- phpwiki_version
- pref
- user

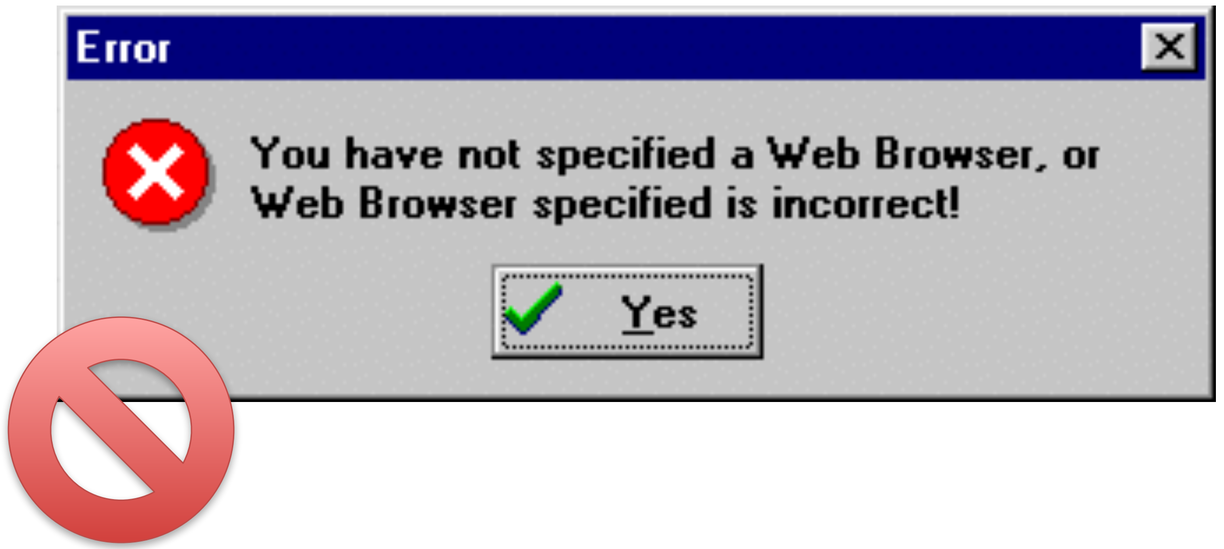
localhost ▸ bjoern_mt_db ▸ mt_blog

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Empty | Drop

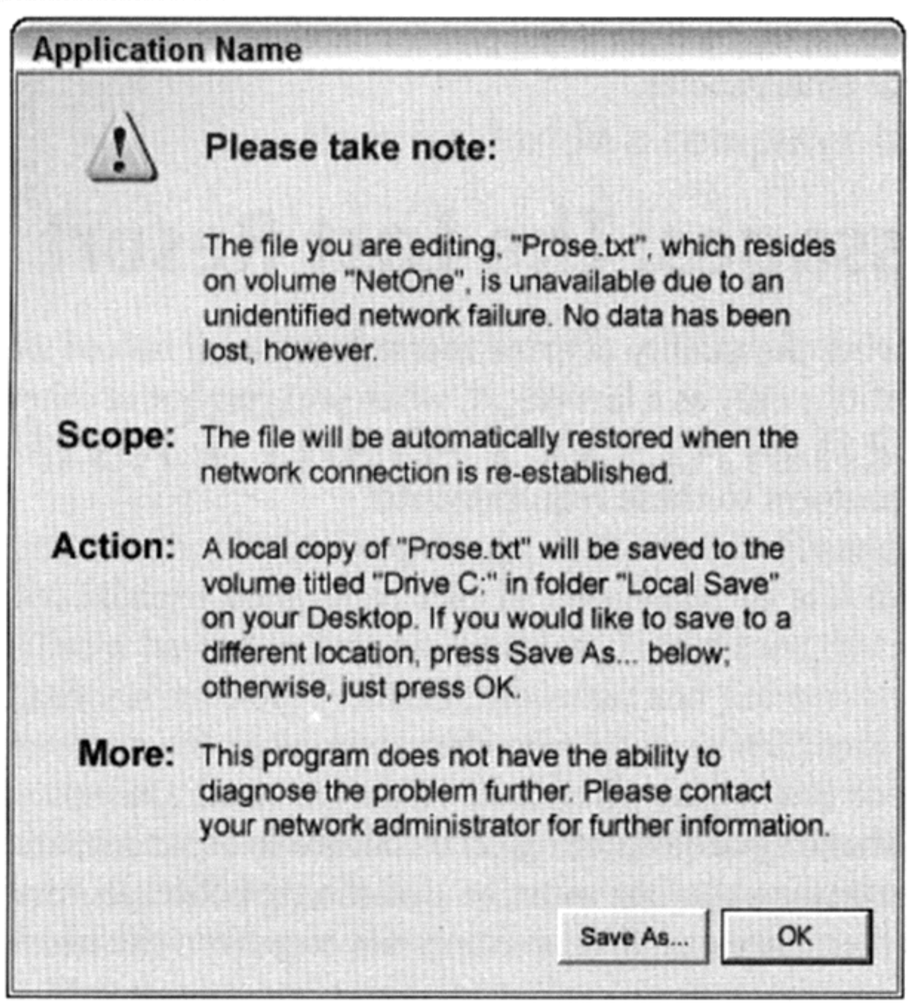
Do a "query by example" (wildcard: "%")

Field	Type	Collation	Operator	Value
blog_id	int(11)		=	
blog_name	varchar(255)	latin1_general_ci	LIKE	
blog_description	text	latin1_general_ci	LIKE	
blog_site_path	varchar(255)	latin1_general_ci	LIKE	
blog_site_url	varchar(255)	latin1_general_ci	LIKE	
blog_archive_path	varchar(255)	latin1_general_ci	LIKE	
blog_archive_url	varchar(255)	latin1_general_ci	LIKE	
blog_archive_type	varchar(255)	latin1_general_ci	LIKE	
blog_archive_type_preferred	varchar(25)	latin1_general_ci	LIKE	
blog_days_on_index	int(11)		=	
blog_language	varchar(5)	latin1_general_ci	LIKE	
blog_file_extension	varchar(10)	latin1_general_ci	LIKE	
blog_email_new_comments	tinyint(4)		=	
blog_email_new_pings	tinyint(4)		=	
blog_allow_comment_html	tinyint(4)		=	
blog_autolink_urls	tinyint(4)		=	
blog_sort_order_posts	varchar(8)	latin1_general_ci	LIKE	
blog_sort_order_comments	varchar(8)	latin1_general_ci	LIKE	
blog_allow_comments_default	tinyint(4)		=	
blog_allow_pings_default	tinyint(4)		=	
blog_server_offset	float		=	
blog_convert_paras	varchar(30)	latin1_general_ci	LIKE	
blog_convert_paras_comments	varchar(30)	latin1_general_ci	LIKE	
blog_status_default	smallint(6)		=	
blog_allow_anon_comments	tinyint(4)		=	
blog_words_in_excerpt	smallint(6)		=	
blog_ping_weblogs	tinyint(4)		=	
blog_ping_blogs	tinyint(4)		=	
blog_ping_others	text	latin1_general_ci	LIKE	
blog_mt_update_key	varchar(30)	latin1_general_ci	LIKE	
blog_autodiscover_links	tinyint(4)		=	
blog_welcome_msg	text	latin1_general_ci	LIKE	
blog_archive_tmpl_monthly	varchar(255)	latin1_general_ci	LIKE	
blog_archive_tmpl_weekly	varchar(255)	latin1_general_ci	LIKE	
blog_archive_tmpl_daily	varchar(255)	latin1_general_ci	LIKE	
blog_archive_tmpl_individual	varchar(255)	latin1_general_ci	LIKE	
blog_archive_tmpl_category	varchar(255)	latin1_general_ci	LIKE	
blog_google_api_key	varchar(32)	latin1_general_ci	LIKE	
blog_sanitize_spec	varchar(255)	latin1_general_ci	LIKE	
blog_cc_license	varchar(255)	latin1_general_ci	LIKE	
blog_is_dynamic	tinyint(4)		=	
blog_require_comment_emails	tinyint(4)		=	
blog_allow_reg_comments	tinyint(4)		=	
blog_junk_score_threshold	float		=	
blog_moderate_pings	tinyint(4)		=	
blog_moderate_unreg_comments	tinyint(4)		=	
blog_created_by	int(11)		=	

H-9: HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS

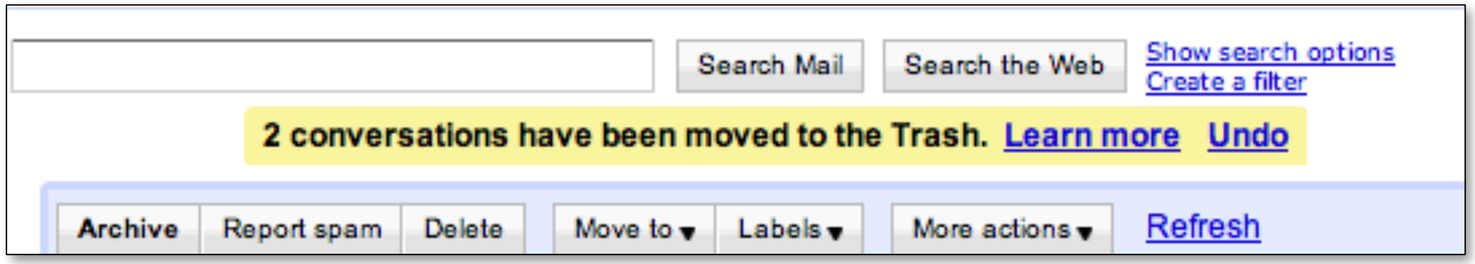


BETTER ERROR MESSAGES



From Cooper's "About Face 2.0"

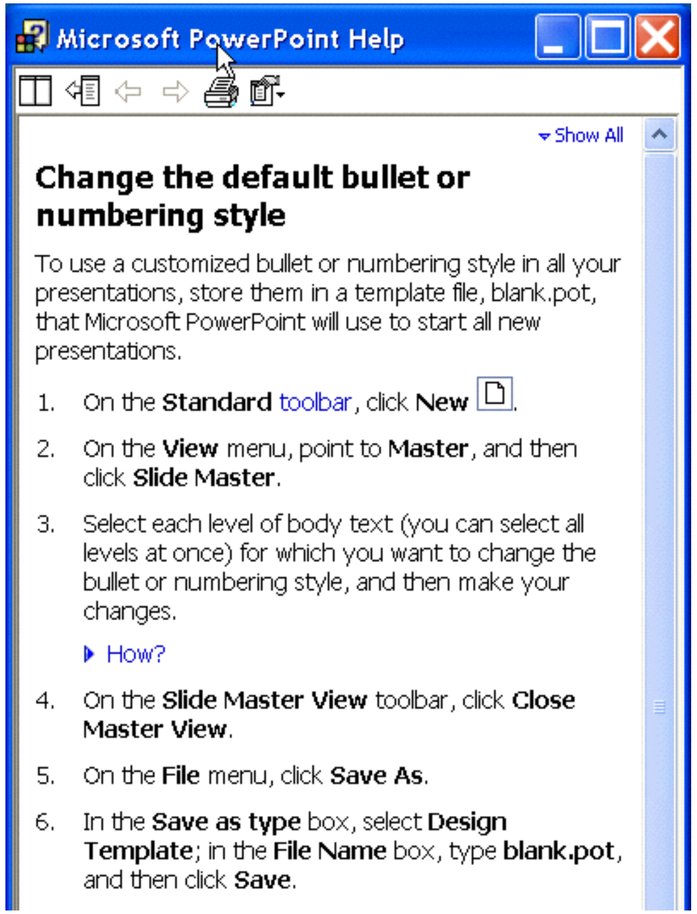
H2-9: HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS



H-10: HELP AND DOCUMENTATION

Help should be:

- Easy to search
- Focused on the user's task
- List concrete steps to carry out
- Not too long



TYPES OF HELP

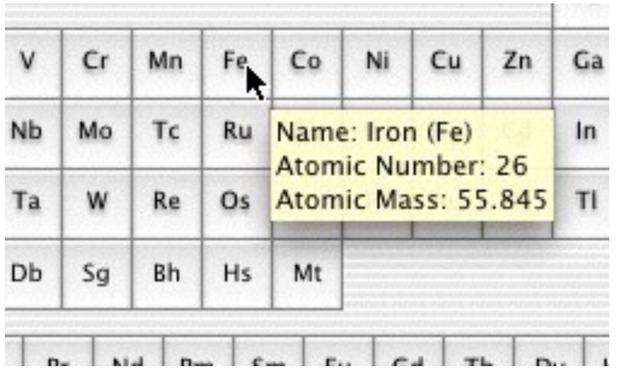
Tutorial and/or getting started manuals

Presents the system conceptual model

Basis for successful explorations

Provides on-line tours and demos

Demonstrates basic features



Reference manuals

Designed with experts in mind

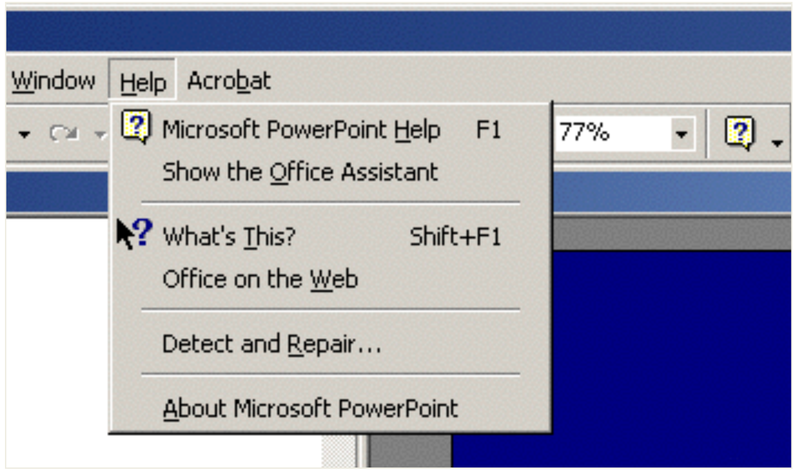
Reminders

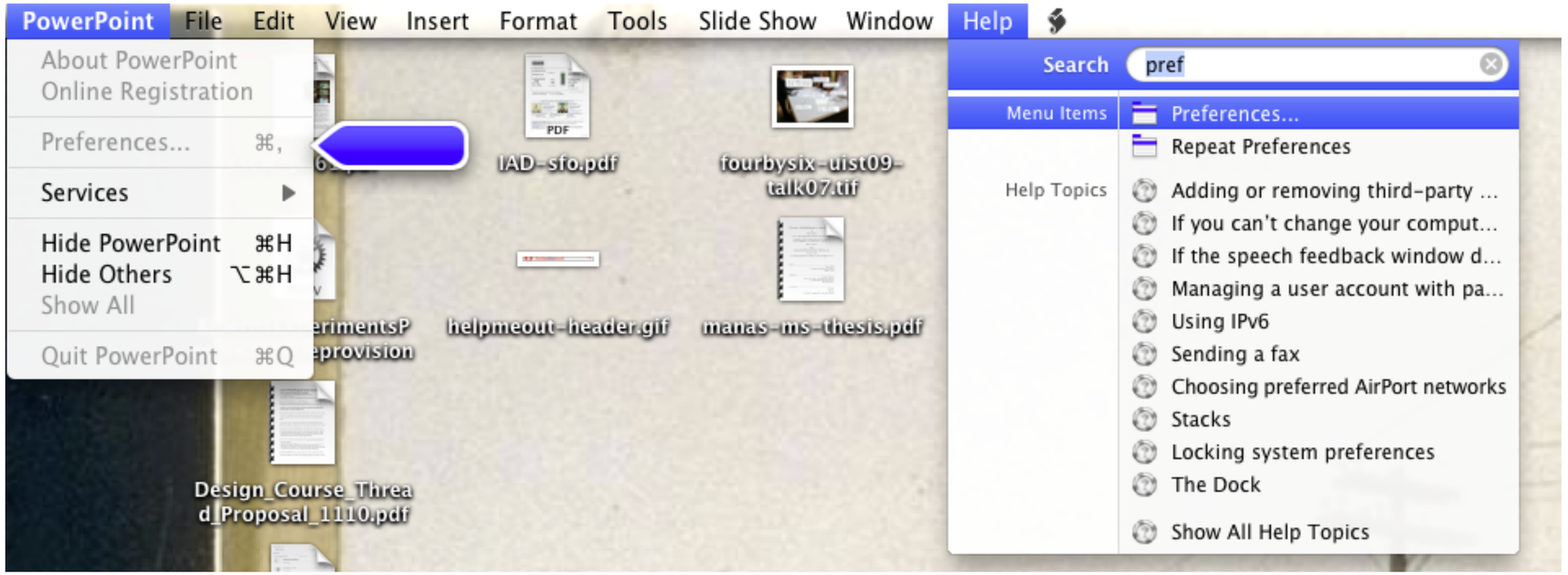
Short reference cards, keyboard templates, tooltips...

TYPES OF HELP

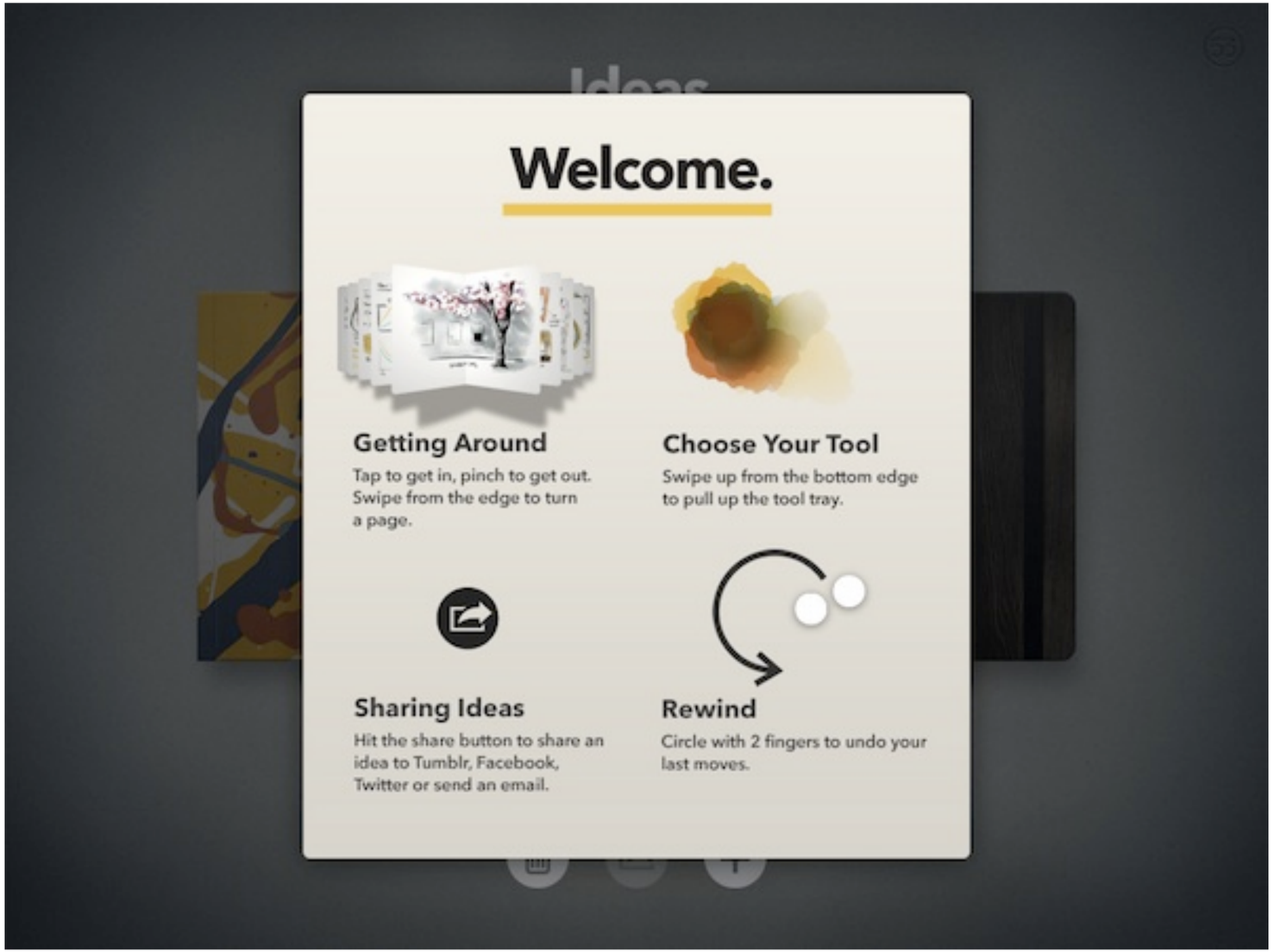
Context sensitive help

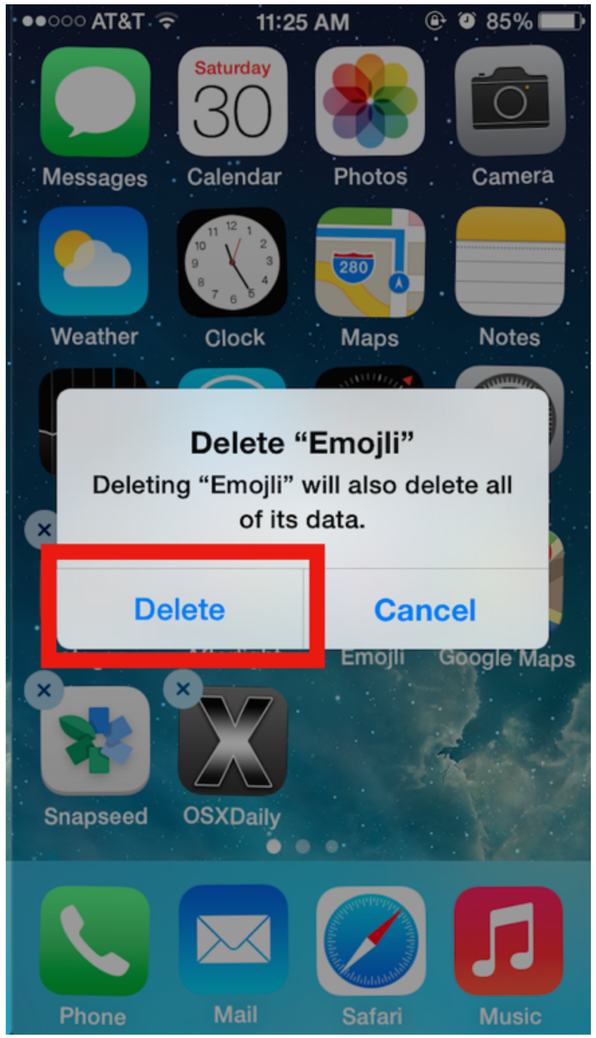
Search

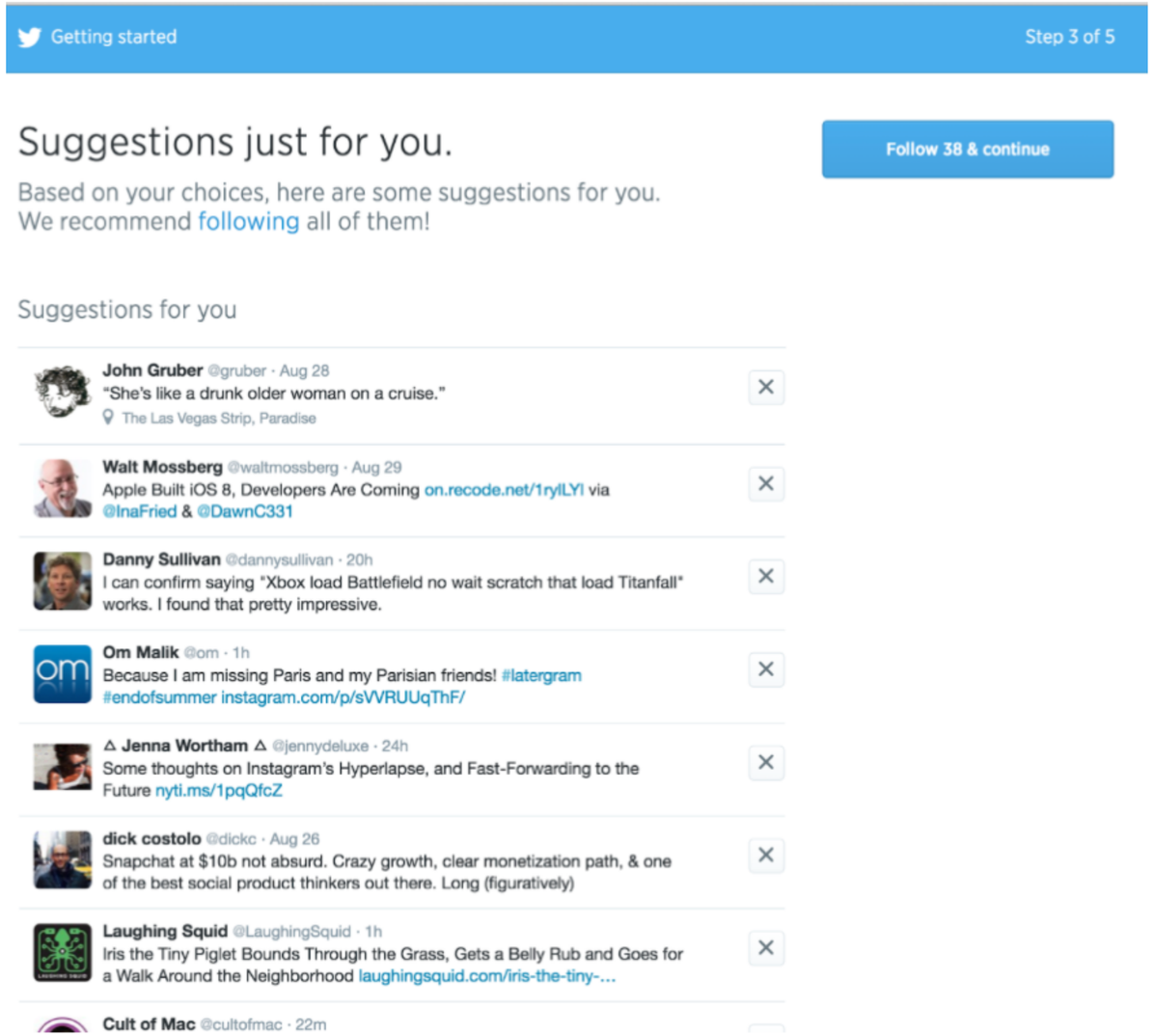




NEW USER GUIDES




AT&T
11:25 AM
85%
Messages
Saturday
30
Calendar
Photos
Camera
Weather
Clock
280
Maps
Notes
Delete “Emojli”
Deleting “Emojli” will also delete all of its data.
Delete
Cancel
Emojli
Google Maps
Snapseed
OSXDaily
Phone
Mail
Safari
Music


Getting started
Step 3 of 5
Suggestions just for you.
Based on your choices, here are some suggestions for you.
We recommend following all of them!
Follow 38 & continue
Suggestions for you
John Gruber @gruber · Aug 28
"She's like a drunk older woman on a cruise."
The Las Vegas Strip, Paradise
Walt Mossberg @waltmossberg · Aug 29
Apple Built iOS 8, Developers Are Coming on.recode.net/1rylLYl via @InaFried & @DawnC331
Danny Sullivan @dannysullivan · 20h
I can confirm saying "Xbox load Battlefield no wait scratch that load Titanfall" works. I found that pretty impressive.
Om Malik @om · 1h
Because I am missing Paris and my Parisian friends! #latergram #endofsummer instagram.com/p/sVVRUUqThF/
Jenna Wortham @jennydeluxe · 24h
Some thoughts on Instagram's Hyperlapse, and Fast-Forwarding to the Future nyti.ms/1pqQfcZ
dick costolo @dickc · Aug 26
Snapchat at $10b not absurd. Crazy growth, clear monetization path, & one of the best social product thinkers out there. Long (figuratively)
Laughing Squid @LaughingSquid · 1h
Iris the Tiny Piglet Bounds Through the Grass, Gets a Belly Rub and Goes for a Walk Around the Neighborhood laughingsquid.com/iris-the-tiny-...
Cult of Mac @cultofmac · 22m

THE PROCESS OF HEURISTIC EVALUATION

PHASES OF HEURISTIC EVAL. (1-2)

1) Pre-evaluation training

Provide the evaluator with domain knowledge if needed

2) Evaluation

Individuals evaluate interface then aggregate results
Compare interface elements with heuristics

Work in 2 passes
First pass: get a feel for flow and scope
Second pass: focus on specific elements

Each evaluator produces list of problems
Explain why with reference to heuristic or other information
Be specific and list each problem separately

PHASES OF HEURISTIC EVAL. (3-4)

3) Severity rating

Establishes a ranking between problems

Cosmetic, minor, major and catastrophic

First rate individually, then as a group

4) Debriefing

Discuss outcome with design team

Suggest potential solutions

Assess how hard things are to fix

EXAMPLES

Typography uses mix of upper/lower case formats and fonts

Violates “Consistency and standards” (H-4)

Slows users down

Fix: pick a single format for entire interface

Probably wouldn’t be found by user testing

LEVELS OF SEVERITY

0 - don’t agree that this is a usability problem

1 - cosmetic problem

2 - minor usability problem

3 - major usability problem; important to fix

4 - usability catastrophe; imperative to fix

SEVERITY RATINGS EXAMPLE

1. [H-4 Consistency] [Severity 3]

The interface used the string "Save" on the first screen for saving the user's file, but used the string "Write file" on the second screen. Users may be confused by this different terminology for the same function.

DEBRIEFING

Conduct with evaluators, observers, and development team members

Discuss general characteristics of UI

Suggest improvements to address major usability problems

Development team rates how hard things are to fix

Make it a brainstorming session

PROS AND CONS OF HEURISTIC EVALUATION

HE VS. USER TESTING

HE is much faster

1-2 hours each evaluator vs. days-weeks

HE doesn't require interpreting user's actions

User testing is far more accurate

Takes into account actual users and tasks

HE may miss problems & find "false positives"

Good to alternate between HE & user-based testing

Find different problems

Don't waste participants

NUMBER OF EVALUATORS

Single evaluator achieves poor results

Only finds 35% of usability problems

5 evaluators find ~ 75% of usability problems

Why not more evaluators???? 10? 20?

Adding evaluators costs more

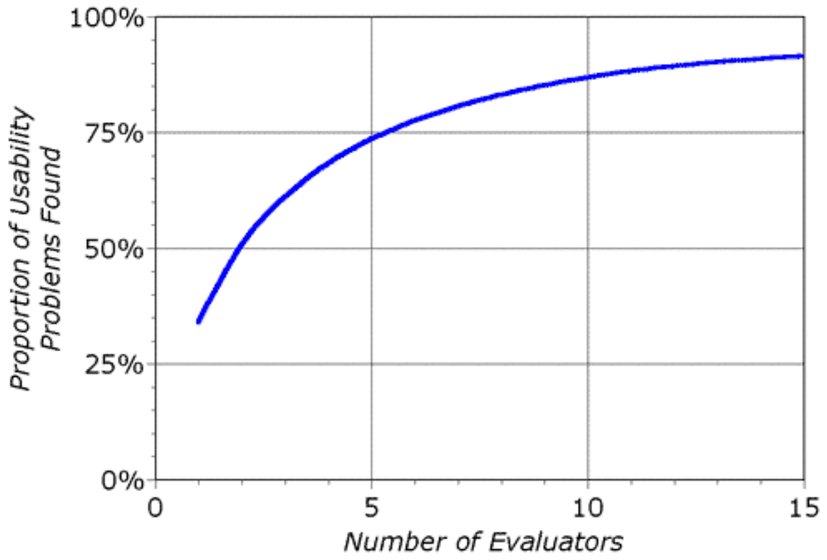
Many evaluators won't find many more problems

But always depends on market for product:

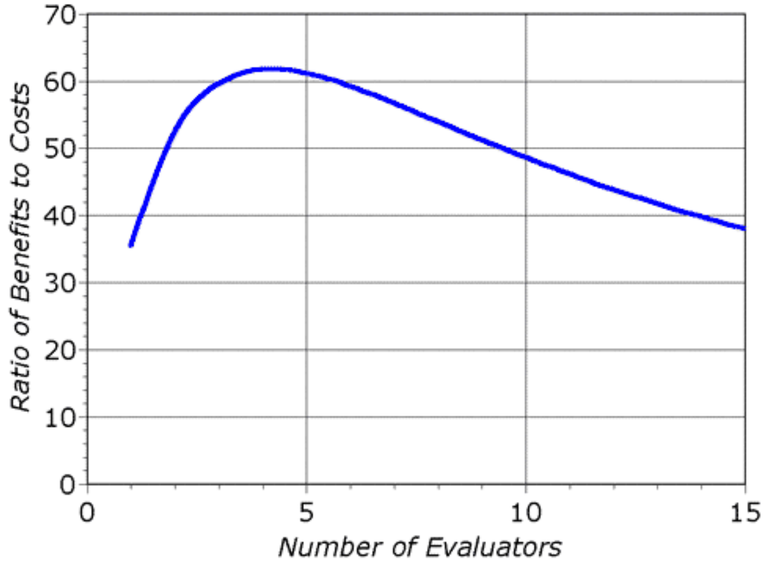
popular products → high support cost for small bugs

DECREASING RETURNS

Problems Found



Benefits / Cost



Caveat: graphs are for one specific example!

SUMMARY

Heuristic evaluation is a discount method

Have evaluators go through the UI twice
Ask them to see if it complies with heuristics
Note where it doesn't and say why

Have evaluators independently rate severity

Combine the findings from 3 to 5 evaluators
Discuss problems with design team

Cheaper alternative to user testing
Finds different problems, so good to alternate