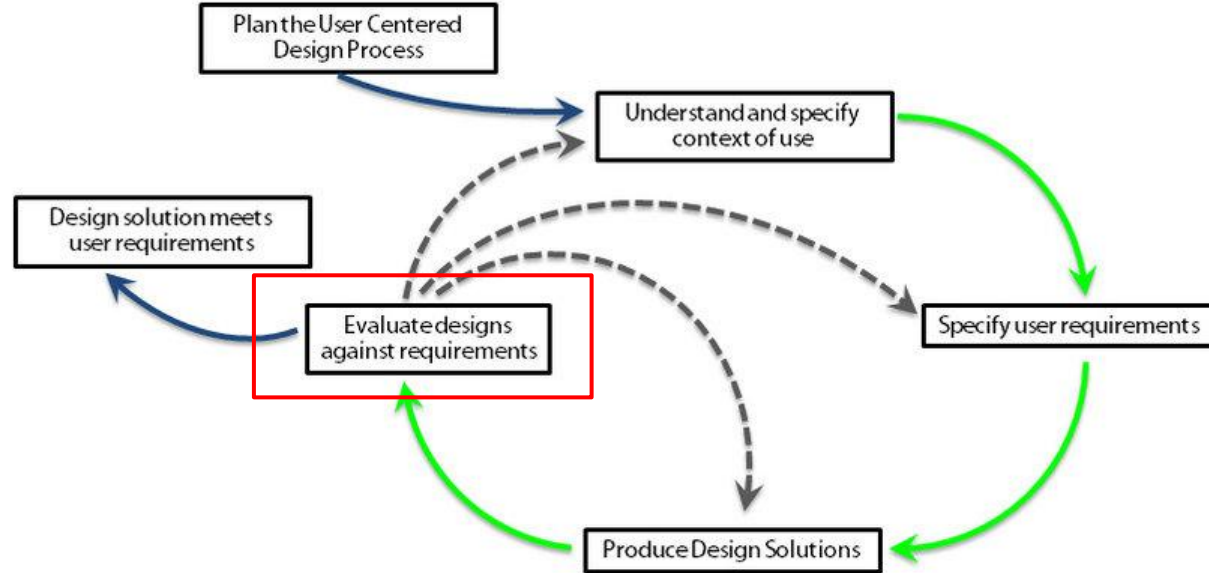


SENG 310

Lecture 12 - June 15th, 2023

RECAP - HEURISTIC EVALUATION

HUMAN-CENTERED DESIGN PROCESS



HEURISTIC EVALUATION

Systematic inspection of an interface design to see if an interface complies with a set of usability heuristics, or usability guidelines.

Generally:

- 3-5 inspectors (usability engineers, end users, experts...)

- inspect interface in isolation (~1-2 hr for simple interfaces)

- results are aggregated afterwards

 - single evaluator catches ~35% usability problems

 - 5 evaluators catch ~75%

HEURISTIC

A heuristic is a rule of thumb—a principle for solving a problem or making decisions

- Never chase after a bus, another one is coming...
- Stuck in traffic: car in the lane next to me passed me
□ that lane must be moving faster

Not always right/true, but cognitive shortcuts

DESIGN HEURISTICS

Broad usability statements that can guide a developer's design efforts

Derived from common design problems across many systems and several researchers and practitioners have developed different sets of heuristics (e.g. domain specific)

CONDUCTING A HEURISTIC EVALUATION

Pre-evaluation training

Provide evaluators with domain knowledge and information on scenario

Evaluation

Individuals evaluate **and then** aggregate results

Severity ratings

Determine how severe each problem is (priority)

Perform individually **and then** as a group

Debriefing

Discuss outcome with design team

EVALUATION

Each evaluator performs at least two passes

First: get a feel for flow and scope of system

Second: focus on specific elements

Assistance

For walk-up and use interfaces, no need; otherwise, supply evaluators with scenarios

Each evaluator produces a list of problems

Explain why with respect to the heuristics or other information

Be specific and list each problem separately

EVALUATION

Why list each violation?

Where problems may be found

- Single location in UI

- Two or more locations that need to be compared

- Overall structure of UI

- Something that is missing

 - Hard w/ paper prototypes

SEVERITY RATINGS

0 – don't think this is a usability problem

1 – cosmetic problem

2 – minor usability problem

3 – major usability problem; important to fix

4 – usability catastrophe; must fix

SEVERITY RATING

Used to allocate resources to fix problems

Combination of:

- Frequency

- Impact

- Persistence (one time or repeating)

Should be estimated after all problems have been seen

Independently first is good

HEURISTIC EVALUATION: ADVANTAGES

Minimalist approach

- Few guidelines identify many common usability problems

“Discount” usability engineering

- End users not required

- Cheap and fast

- Can be done by usability experts as well as end users

- provides common evaluation template (to evaluate across systems)

HEURISTIC EVALUATION: DISADVANTAGES

Principles are general

Not a simple checklist

You may actually have the wrong design altogether

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

NIELSEN'S 10 HEURISTICS

1. **VISIBILITY OF SYSTEM STATUS**
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

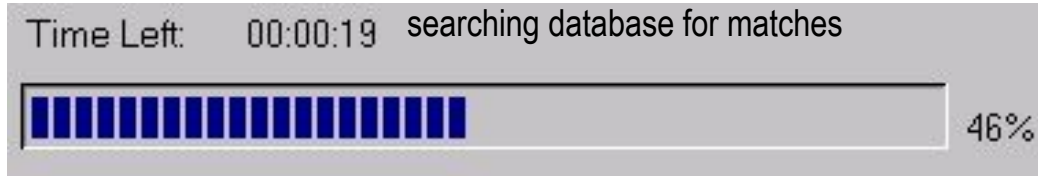
1. VISIBILITY OF SYSTEM STATUS

Keep users informed about what is going on

Feedback **should be appropriate** (from the perspective of the user)

Feedback **should be timely**

1. VISIBILITY OF SYSTEM STATUS



Example: pay attention to response time

- 0.1 sec: no special indicators needed, why?

- 1.0 sec: user tends to lose track of data

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

for longer delays, use percent-done progress bars

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. **MATCH BETWEEN SYSTEM AND REAL-WORLD**
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

2. MATCH BETWEEN SYSTEM AND REAL-WORLD

System should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.

Follow real-world conventions: information should appear in natural and logical order based on user's expectations.

Remove modes.

2. MATCH BETWEEN SYSTEM AND REAL-WORLD

Operating Statement

Balance Sheet

Additional Reports

Help

Home

Application Info

Quick Launch

Dashboard

Pinned Reports

Help & Resources

Finance Reporting

Please input your desired report parameters.
You can click on the magnifying glass to lookup a field.

Chart(s):

☒ 1 - University Of Victoria

Index:

Fund:

Orgn:

Acct:

Prog:

Actv:

Course Info

Registration

Student Grades

Help

Student Reporting

Course Summary

Summary listing of all Courses Offered

(Data as of 11/06/2020 01:09:48 AM)

Filter Options

Help

+ More

Term

Click to add values

Campus

Click to add values

Schedule Desc

Click to add values

Instructor

Click to add values

Subject

Click to add values

Number

Click to add values

CRN

Click to add values

Capacity

Click to add values

Enrolled

Click to add values

Waitlist

Click to add values

Avail Seats

Click to add values

Faculty

Click to add values

Start Date

Click to add values

End Date

Click to add values

Attributes

Click to add values

Cross List

Click to add values

Level

Click to add values

Building

Click to add values

or Subj and Course

Click to add values

Room

Click to add values

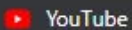
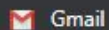
MODES: NO ONE REMEMBERS WHAT MODE THEY'RE IN

Universal remote controls can control a variety of devices

Buttons are overloaded (e.g. “up”, “down”, “left”, “right”, “OK”), so that pressing it will control a device depending on which mode the device is in.

This one indicates (by a blinking light) which device is being activated, but you may still not notice it.





You've gone incognito

Now you can browse privately, and other people who use this device won't see your activity. However, downloads and bookmarks will be saved. [Learn more](#)

Chrome won't save the following information:

- Your browsing history
- Cookies and site data
- Information entered in forms

Your activity might still be visible to:

- Websites you visit
- Your employer or school
- Your internet service provider

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. **USER CONTROL AND FREEDOM**
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

3. USER CONTROL AND FREEDOM

Users should be provided with clearly marked “emergency exits” to leave unwanted states.

Support undo and redo.

Cancel long operations

3. USER CONTROL AND FREEDOM

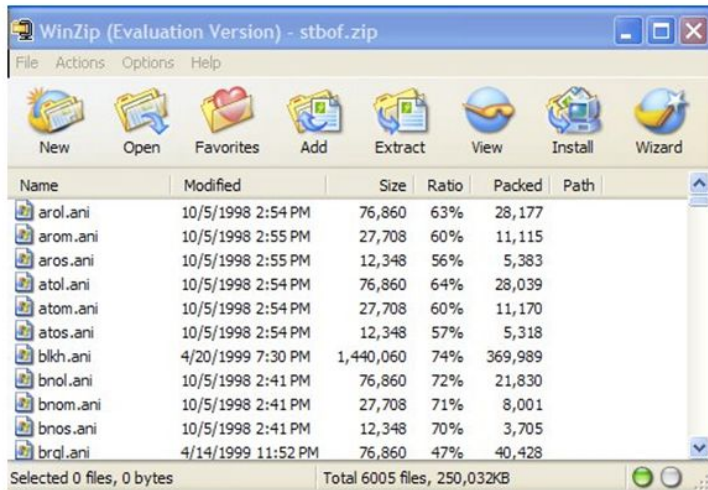


3. USER CONTROL AND FREEDOM



Wizard

Center Stage



NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. **CONSISTENCY AND STANDARDS**
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

4. CONSISTENCY AND STANDARDS

Principle of least surprise

similar things should act similarly

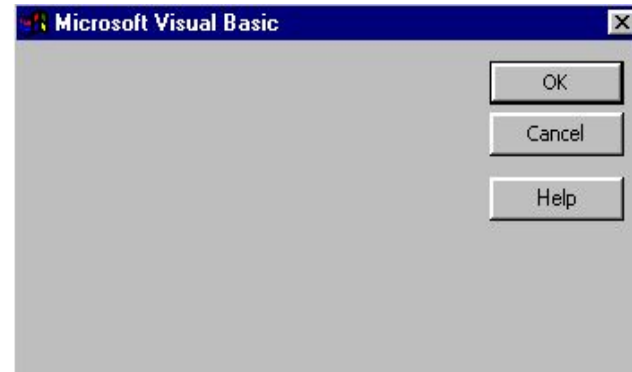
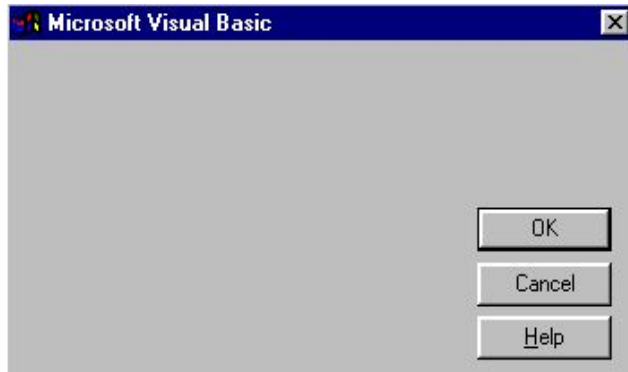
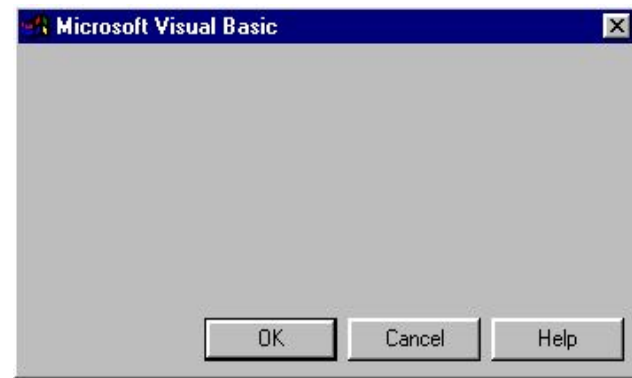
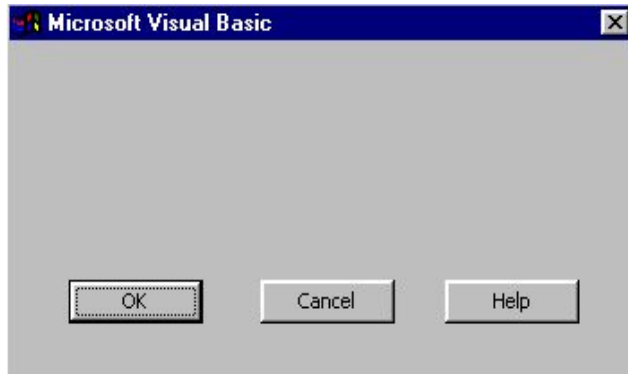
different things should look different

Adhere to platform guidelines

Consistent language, colour, wording, ordering

Consistent use of input syntax

GOAL: PREDICTABILITY



THREE TYPES OF CONSISTENCY

Internal consistency: is the interface consistent with itself?

External consistency: is the design consistent with similar types of applications/applications on the platform?

Metaphorical consistency: is the design consistent with the similar real-world entity/object?

NIELSEN'S 10 HEURISTICS

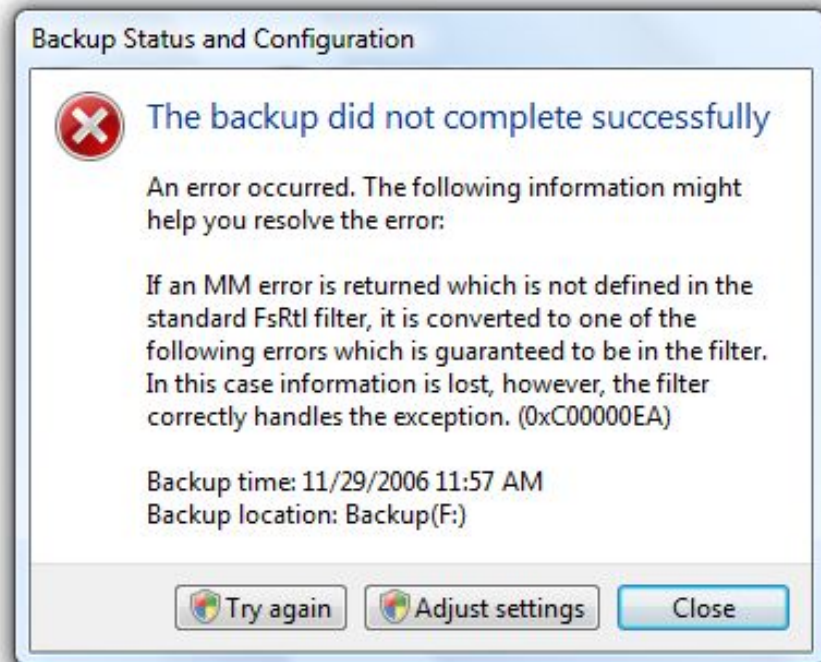
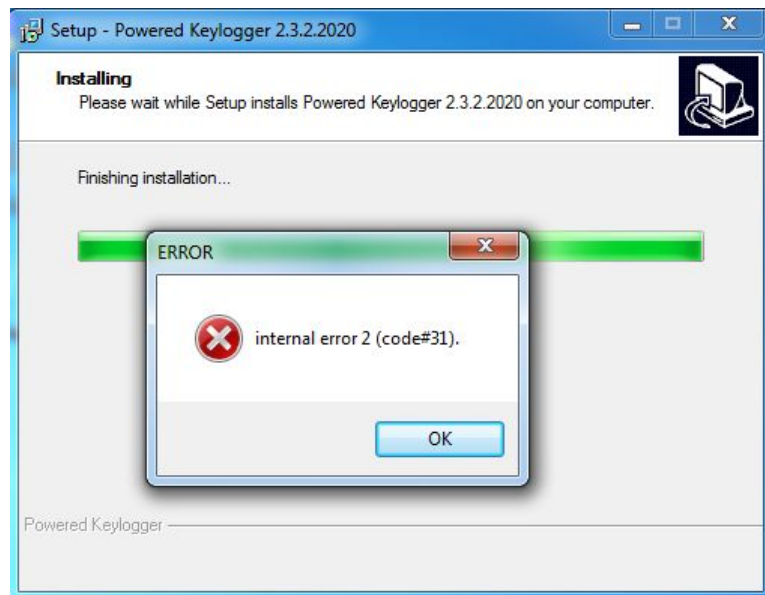
1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. **HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS**
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS

Use plain language

Identify the problem

Constructively suggest a solution



- Invalid email address entered
- Billing zipcode format is invalid.
- Missing a value for the required property: billzipcode
- Missing a value for the required property: billcity
- Missing a value for the required property: billaddress1
- Missing a value for the required property: login
- Missing a value for the required property: billstate

registration

1. Personal information

First *	M	Last *
<input type="text" value="Stuart"/>	<input type="text"/>	<input type="text" value="Jones"/>
Email (Login) *		
<input type="text"/>		
<input checked="" type="checkbox"/> Email me promotions?		
Gender <input type="button" value="v"/>		

2. Billing information

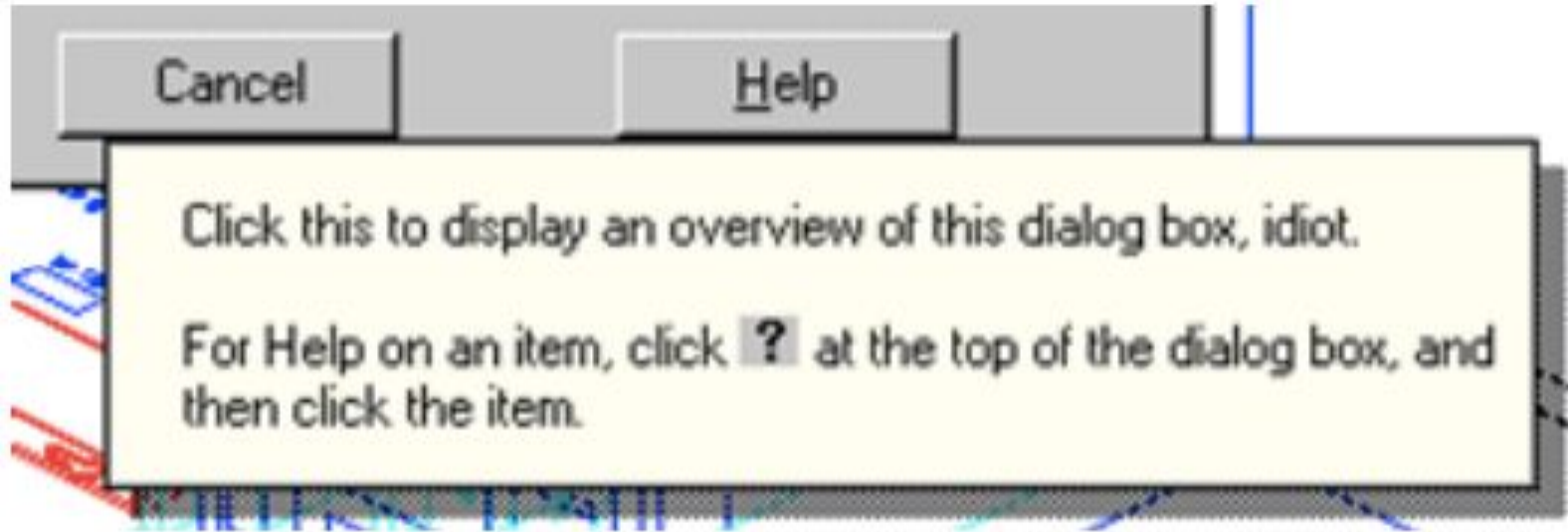
(Credit Card Mailing Address)

Care of
<input type="text"/>
Address *
<input type="text"/>
City *
<input type="text"/>

Be polite » don't blame



BE POLITE EVEN WHEN YOU'RE CODING FOR YOURSELF...



Source Interface Hall of Shame

Provide
suggestions/examples

 Please enter your email address in this format: "youremail@domain.com".

Restate the user's input
not "cannot open file", but "cannot open file
named paper.doc"

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. **ERROR PREVENTION**
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

6. ERROR PREVENTION

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

CART RETURN LANE

Constraining width of the lane makes it possible to avoid “get it wrong”



PREVENTION TECHNIQUES

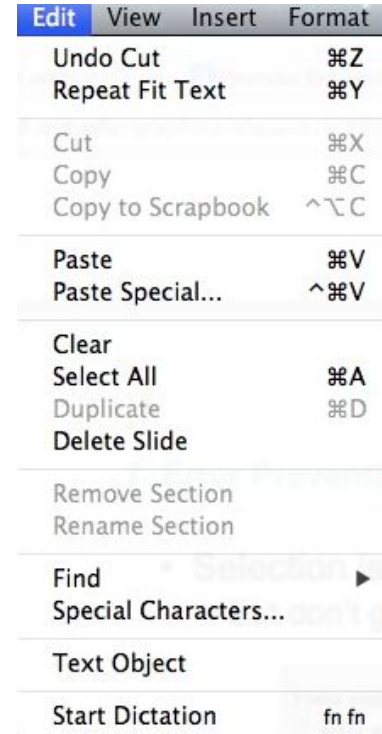
Grey out commands

Select rather than type

Avoid typing errors
through selection



Source: Interface Hall of Shame



More Error Prevention

Provides auto-complete suggestions for city

Date picker for dates (while still allowing for text-entry)

Greys out inappropriate “check out” dates given “check in” dates



The image shows a screenshot of the Hotwire website's 'Hot Rate Locator' form. At the top, the Hotwire logo is on the left, and 'Hotels' and 'Cars' tabs are on the right. The main form area has a yellow background and is titled 'Hotwire Hot Rate Locator'. It contains several sections: 1. 'Bundle + Save' with radio button options for 'Hotels' (selected), 'Cars', and 'Flights', and a 'Bundle + Save' section with options for 'Flight + Hotel', 'Flight + Hotel + Car', 'Flight + Car NEW!', and 'Hotel + Car'. 2. 'Where are you going?' with a text input field for 'Enter address, zip, city, or airport' and a 'See all cities' link. 3. 'When?' section with 'Check in' and 'Check out' date pickers. 4. 'How many people?' section with dropdown menus for 'Rooms' (set to 1), 'Adults' (set to 2, with '(18+ yrs)' below), and 'Children' (set to 0, with '(2-17 yrs)' below). At the bottom, there is a red 'Find a hotel' button with a play icon and a grey 'Feedback' button.

LAPSES

Loss of intention » forgetting the goal partway
walking into a room, forgetting why you went there

Omissions due to interruption
get coat out, interrupted by phone call; then go out without coat

Omissions due to already satisfied goal
walking away from ATM w/o card
walking away from copier without originals

AVOIDING LAPSES

Keep procedures short and have dialog closure

Use forcing functions

automatic transmission: you must hold down brake to shift out of Park

must take card out of ATM before you get your money

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. **RECOGNITION RATHER THAN RECALL**
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

7. RECOGNITION RATHER THAN RECALL

Making objects, actions, and options visible to minimize the user's memory load

The user should not have to remember information from one part of the dialogue to another

Instructions for use of the system should be visible or easily retrievable whenever appropriate



Tell us which artists you like

We'll recommend you new music and calculate your music compatibility with your friends on iLike. The more artists you rate, the better.



Thievery



Carrie Underwood



Keith Urban



Leona Lewis



Radiohead



The Beatles



 **Hi uipatens**
Your profile | Settings | Inbox

Welcome to Last.fm

To get started, type in some of your favourite artists, separated with commas.

Save

Get music recommendations based on the music in your library.

Download the [Scrobbler](#) to fill your library automatically from your media player or iPod.

Arno Pro

Ayuthaya

Baghdad

✓ **BANK GOTHIC**

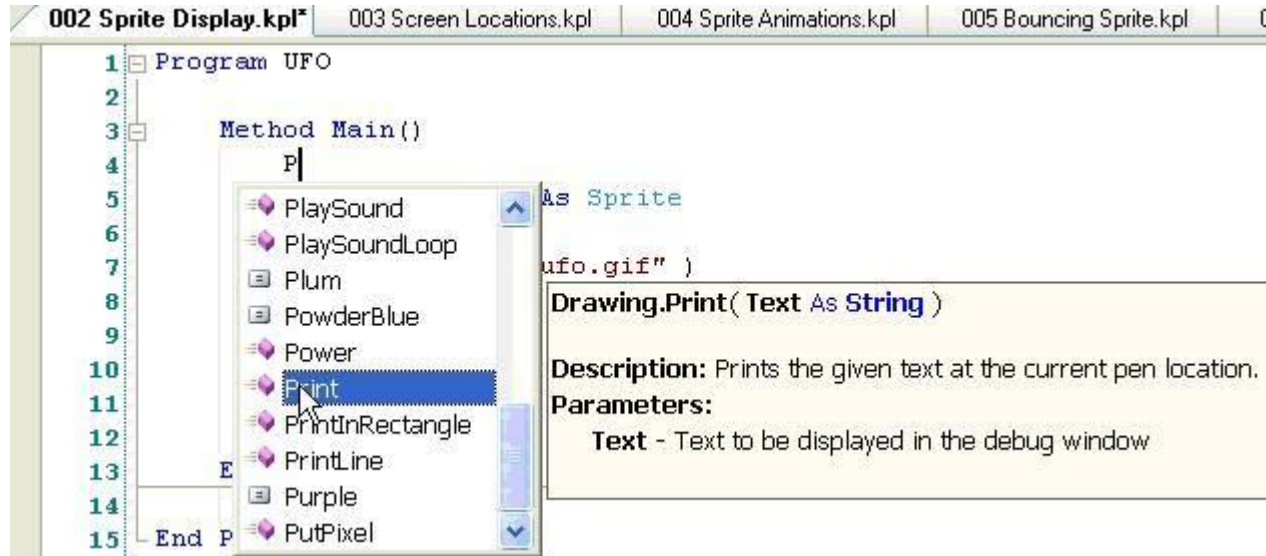
Baskerville

Baskerville Old Face

Bastion

Batang

Autocomplete is a nice example of how the system aids you. It is easier to recognize symbols rather than recall them from scratch.



NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. **AESTHETIC AND MINIMALIST DESIGN**
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION

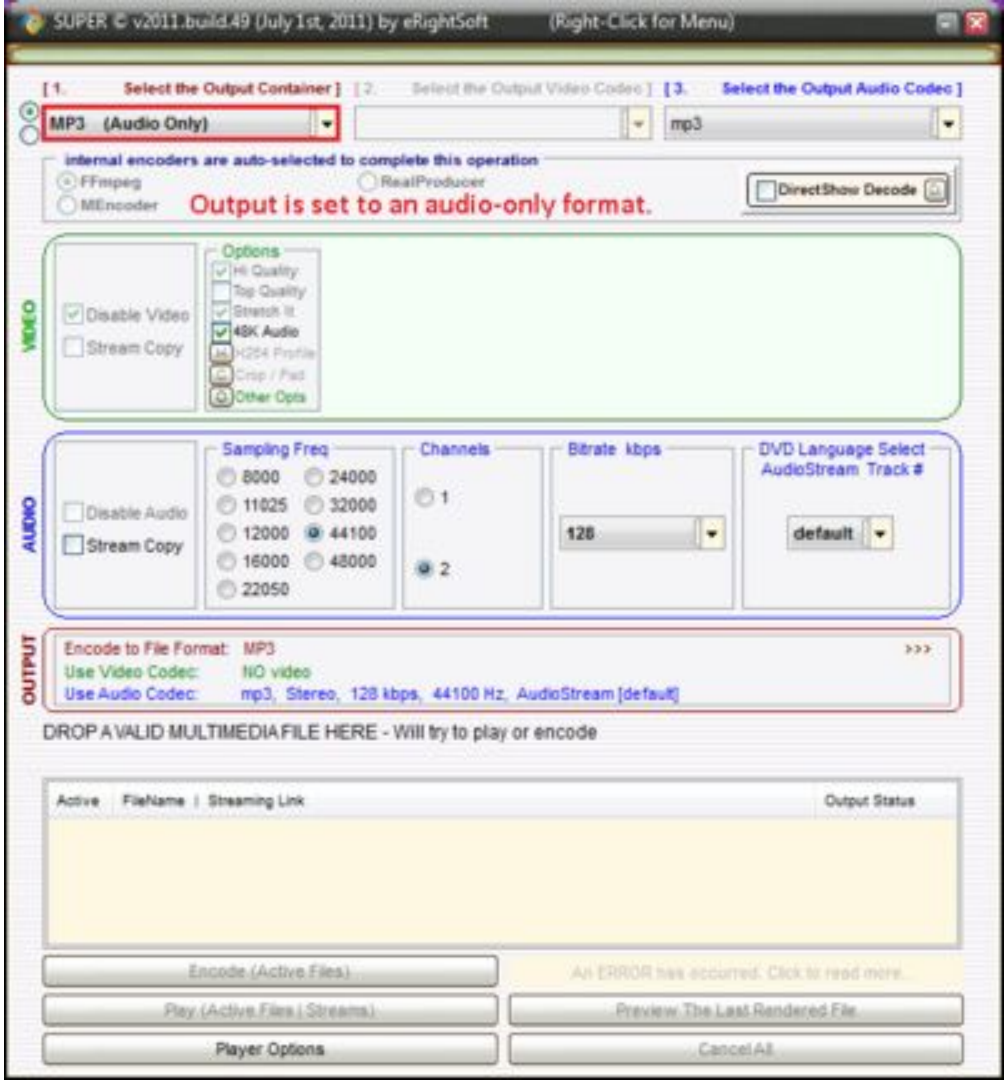
8. AESTHETIC AND MINIMALIST DESIGN

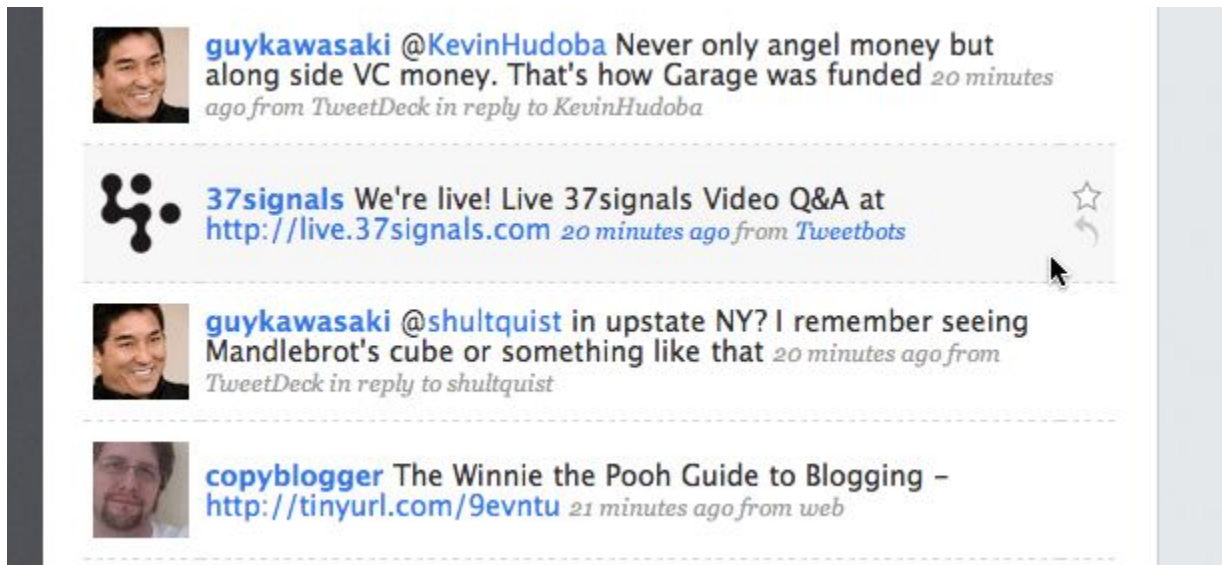
Dialogues should not contain information which is irrelevant or rarely needed

Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

SUPER is an audio/video converter tool

Although I have selected “audio only”, all of the video stuff is still there.





Hover controls only appear when they are likely to be used (i.e. when the mouse is hovering over the activation area).

What is the downside of this approach? Discoverability can be compromised

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. **FLEXIBILITY AND EFFICIENCY OF USE**
10. HELP AND DOCUMENTATION

9. FLEXIBILITY AND EFFICIENCY OF USE

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users

- Macros

- Double clicking to like in Instagram

- Keyboard shortcuts

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. **HELP AND DOCUMENTATION**

10. HELP AND DOCUMENTATION

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation.

Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

WHAT CONSTITUTES HELP?

Tutorials / get started manuals

Reference manuals

Reminders

Wizards

Tips

NIELSEN'S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION