

Design Principles And Usability Heuristics

You can avoid common design pitfalls by following these rules-of-thumb for design

You can inspect an interface for usability problems by applying these same principles

James Tam

Design Principles And Usability Heuristics

Heuristics are broad “rules of thumb” that describe features of “usable” systems:

- 1) Heuristics can be used as a design guide before the system is built.
- 2) Heuristics can be used as an evaluation mechanism after the system has been completed. The same principles can be used to “evaluate” a system for usability problems

Becoming quite popular:

1. User involvement is not required.
2. Can be applied without a great deal of prior training or time (another example of discount usability) yet it catches many design flaws.

James Tam

Design Principles And Usability Heuristics **(Advantages)**

- A few general guidelines can correct for the majority of usability problems
- Easily remembered, easily applied with modest effort
- A cheap and fast way to inspect a system
- It can be conducted by people who are not usability experts

James Tam

Design Principles And Usability Heuristics **(Weaknesses)**

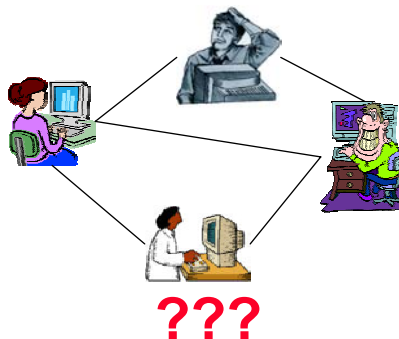
Principles are more or less at the motherhood level

- Can't be treated as a simple checklist
- Subtleties involved in their use
- A very broad approach that may not cover specific situations
 - Example:

Heuristics for a
single user



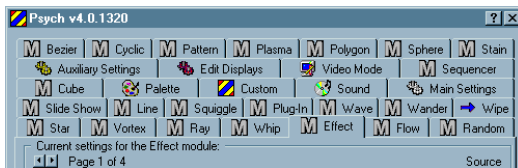
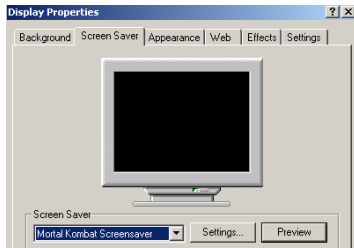
Heuristics for many
collaborating users



James Tam

1. Simple And Natural Dialogue

Avoid making the interaction unnecessarily complex



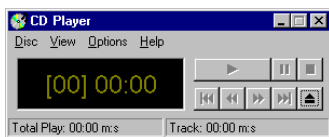
James Tam

1. Simple And Natural Dialogue

Match the user's conceptual model

Match the users' task in as natural a way as possible

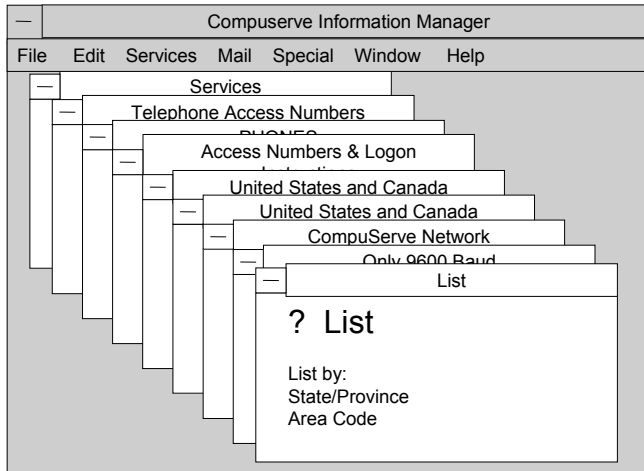
- Minimize mapping between interface and task semantics



James Tam

1. Simple And Natural Dialogue

Menu or window? Which window?



James Tam

1. Simple And Natural Dialogue

Present exactly the information the user needs when it is needed

- Less is more
 - Less to learn, to get wrong, to distract...
- Remove or hide irrelevant or rarely needed information
 - Competes with important information on screen
- Information should appear in natural order
 - Order of accessing the information matches the user's expectations
 - Related information is graphically clustered
- Minimize or mitigate modes
- Use windows frugally
 - Don't make navigation and window management excessively complex

James Tam

Dr. Drugs Pharmacy Master

Davison, Thomas

First Name: Thomas

Last Name: Davison

Address: 4 BRADEN CRS NW

City: Prudhoe

Province: Alberta

Postal Code: F4T6H8

Phone Number: 403 282-9759

Alberta Health Care #: 45688-3357

Doctor: Jane Yang

Insurer: FARMERS INSURANCE AGE

Policy #: 4215487-56

Notes:

Prescriptions

Medication	Start Date	End Date
Alprazolam	Jan 2001	Feb 2001
Citalopram	Feb 2001	Mar 2001
Ranitidine	Mar 2001	May 2001
Folic Acid	Jun 2001	Jul 2001
Levothyroxine	Aug 2001	Sep 2001
Celecoxib	Oct 2001	Nov 2001
Omeprazole	Nov 2001	Dec 2001

1999 < Year > 2001 | Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Good: Info in the same place

By previous 481 students Brant LeClercq, Lloyd Yoon, Amy Yang James Tam

Dr. Drugs Pharmacy Master

First Name:

Last Name:

Address:

City:

Province:

Postal Code:

Phone Number:

Alberta Health Care #:

Doctor:

Insurer:

Policy #:

Notes:

Prescriptions

Medication	Start Date	End Date
Alprazolam	Jan 2001	Feb 2001
Citalopram	Feb 2001	Mar 2001
Ranitidine	Mar 2001	May 2001
Folic Acid	Jun 2001	Jul 2001
Levothyroxine	Aug 2001	Sep 2001
Celecoxib	Oct 2001	Nov 2001
Omeprazole	Nov 2001	Dec 2001

1999 < Year > 2001 | Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Good: Info in the same place
Bad: Special edit mode

By previous 481 students Brant LeClercq, Lloyd Yoon, Amy Yang (with permission) James Tam

Dr. Drugs Pharmacy Master

Davison, Thomas Add Undo New Customer

First Name: Thomas
Last Name: Davison
Address: 4 BRADEN CRS NW
City: Prudhoe
Province: Alberta
Postal Code: F4T6H8
Phone Number: 403 282-9759

Health Care #: 45688-3357
Doctor: Jane Yang
Insurer: FARMERS INSURANCE AGE
Policy #: 4215487-56
Notes:

Prescriptions See List View

Drug Name
 Manufacturer
 Quantity
 Price/Unit
 Doctor
 Units
 Refill Quantity

collapsed onto one screen (needs formatting)

Click to get info

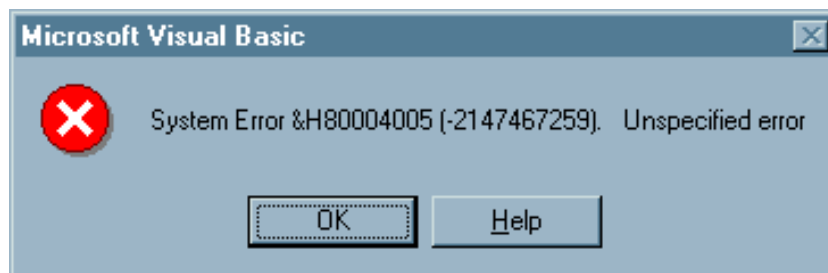
Double click to edit (mode buttons gone)

Drug Name	Start Date	End Date
Alprazolam	Jan 2001	Feb 2001
Citalopram	Feb 2001	Mar 2001
Ranitidine	Mar 2001	Apr 2001
Folic Acid	Apr 2001	May 2001
Levothyroxine	May 2001	Jun 2001
Celecoxib	Jun 2001	Jul 2001
Omeprazole	Jul 2001	Aug 2001
	Aug 2001	Sep 2001
	Sep 2001	Oct 2001
	Oct 2001	Nov 2001
	Nov 2001	Dec 2001

1999 <Year> 2001 Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

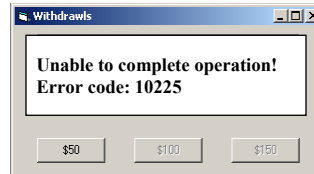
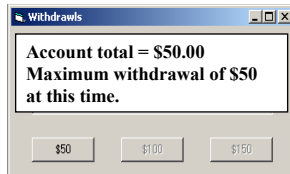
By previous 481 students Brant LeClercq, Lloyd Yoon, Amy Yang (with permission) James Tam

2. Speak The User's Language



2. Speak The User's Language

Terminology based on users' language for task



Use meaningful mnemonics, icons, and abbreviations

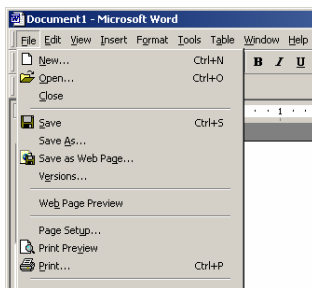
- eg File / Save
 - Ctrl + S (abbreviation)
 - Alt FS (mnemonic for menu action)
 - Open folder (tooltip icon)



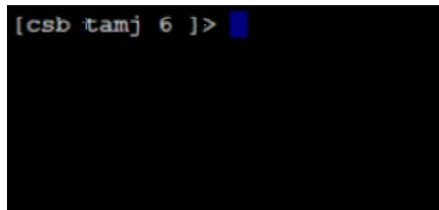
James Tam

3. Minimize The User's Memory Load

- Describe required the input format, use examples, provide default inputs
- Promote recognition over recall



✓ Recognition



✗ Recall

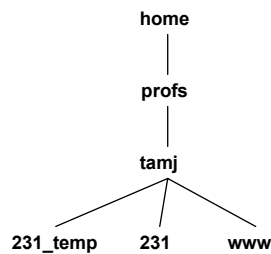
James Tam

3. Minimize The User's Memory Load

Small number of rules applied universally

Remove
what?

File
11110000
01010101



James Tam

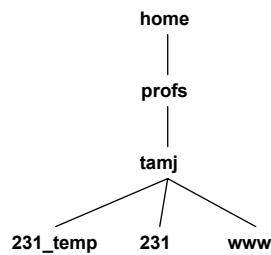
3. Minimize The User's Memory Load

Small number of rules applied universally

Remove
what?

rm file name

File
11110000
01010101



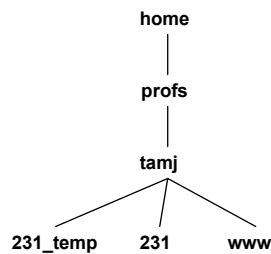
James Tam

3. Minimize The User's Memory Load

Small number of rules applied universally

File
11110000
01010101

Remove
what?



James Tam

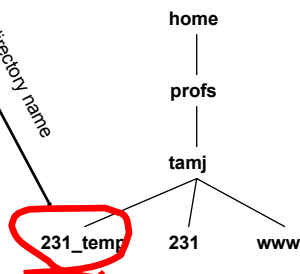
3. Minimize The User's Memory Load

Small number of rules applied universally

File
11110000
01010101

Remove
what?

mdir directory name



James Tam

4. Be Consistent

Consistency of effects

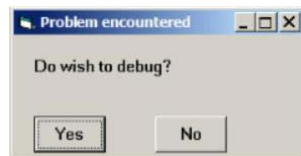
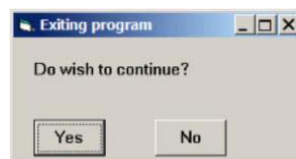
- Same words, commands, actions will always have the same effect in equivalent situations
- Makes the system more predictable
- Reduces memory load

James Tam

4. Be Consistent

Consistency of language and graphics

- Same information/controls in same location on all screens / dialog boxes forms follow boiler plate.
- Same visual appearance across the system (e.g. widgets).



James Tam

4. Be Consistent

These are labels with a raised appearance.

Is it any surprise that people try and click on them?

Subscriber

Name: Tech. Re

Account #: Status:

Contact

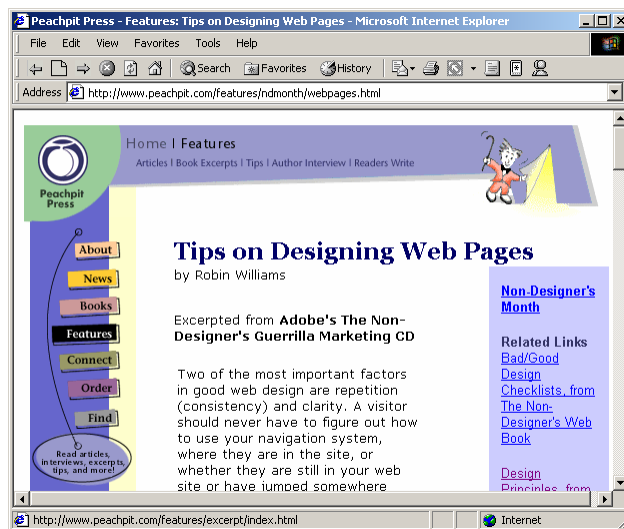
Telephone: E-Mail:

Address: St

Save Cancel

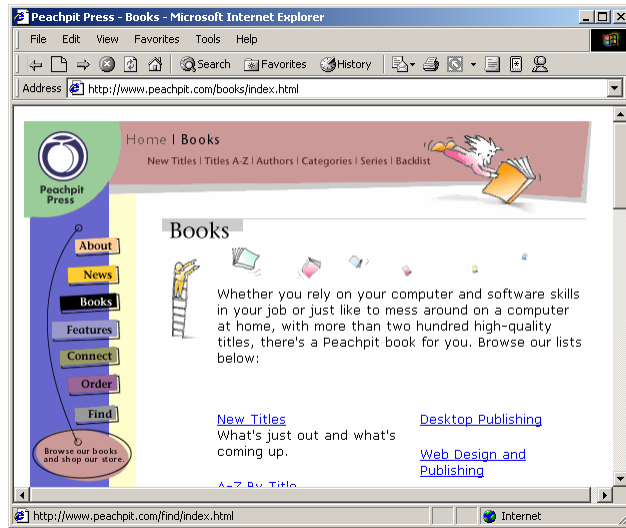
James Tam

4. Be Consistent



James Tam

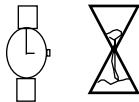
4. Be Consistent



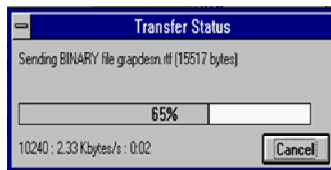
James Tam

5. Provide Feedback

What is the program doing?



Cursor



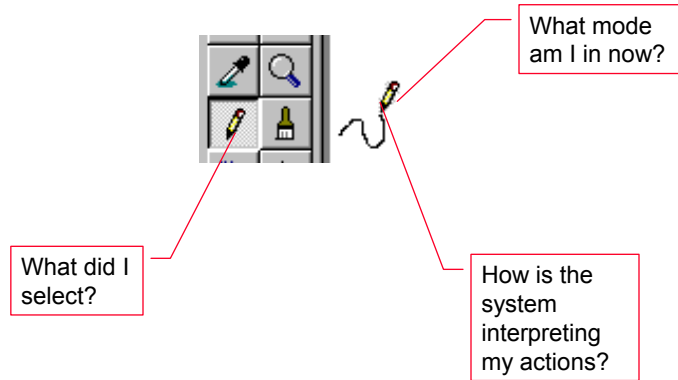
Progress bar



Random graphic

James Tam

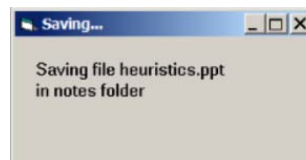
5. Provide Feedback



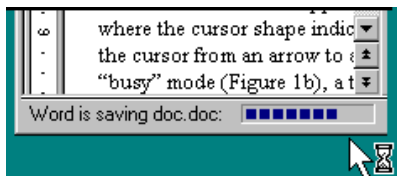
James Tam

5. Provide feedback

Should be as specific as possible, based on user's input



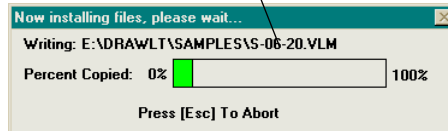
Best within the context of the action



James Tam

5. Provide feedback

Multiple files being copied,
but feedback is file by file.



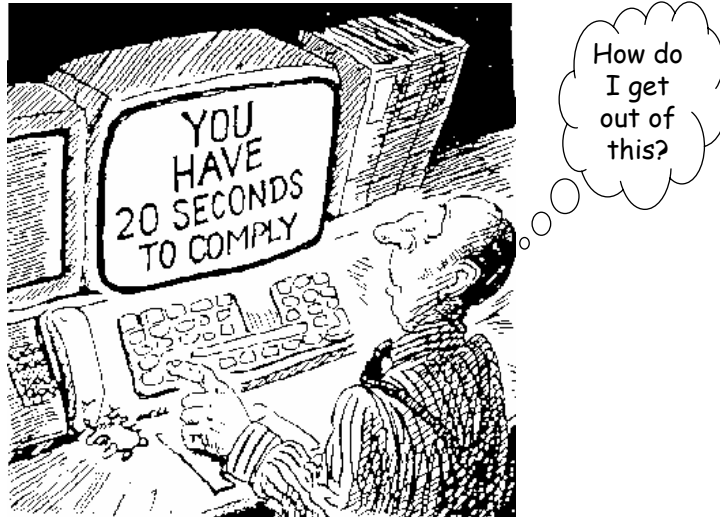
5. Provide feedback

Feedback response time

- How users perceive delays

Delay	Effect
$0 \leq \text{delay} \leq 0.1$ seconds	Delay not noticed (system appears to operate instantaneously).
$0.1 < \text{delay} \leq 1$ second	Delay noticed but train of thought remains uninterrupted.
$1 < \text{delay} \leq 10$ seconds	Train of thought interrupted but person can still remain focused on the system. (Cursor)
Delay > 10 seconds	Person wants to do something else while waiting for the system. (Progress bar)

6. Provide Clearly Marked Exits

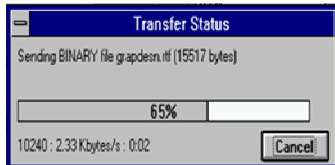


James Tam

6. Provide Clearly Marked Exits

This doesn't just mean providing an exit from the program but the ability to 'exit' (take back) the current action.

- Universal Undo/Redo
 - e.g., <Ctrl>-<Z> and <Ctrl> <Y>
- Progress indicator & Interrupt
- Length operations

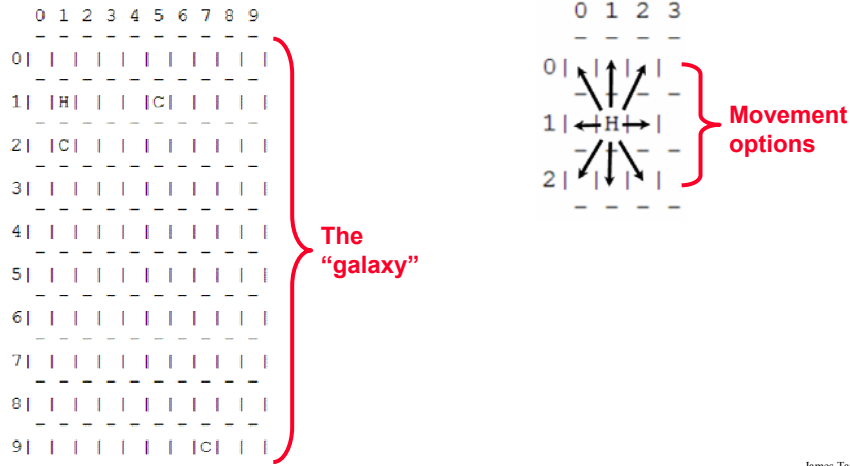


James Tam

6. Provide Clearly Marked Exits

This can apply even with systems that are written for ‘experts’

- Example: A text-based space simulation game



James Tam

6. Provide Clearly Marked Exits

```
do
{
    insideGalaxy = isEmpty = withinRange = true;
    System.out.print("Enter the destination row (0 - 9): ");
    row = Console.in.readInt();
    System.out.print("Enter the destination column(0 - 9): ");
    column = Console.in.readInt();
    if ((row < 0) || (row > (Galaxy.SIZE-1)) || (column < 0) || (column > (Galaxy.SIZE-1)))
    {
        insideGalaxy = false;
        System.out.println("You cannot move outside the galaxy");
    }
    while ((insideGalaxy == false) || (isEmpty == false) || (withinRange == false));
```

Loop
while
input is
invalid

James Tam

6. Provide Clearly Marked Exits

```
do
{
    insideGalaxy = isEmpty = withinRange = true;

    System.out.print("Enter the destination row (0 - 9, -1 to pass): ");
    row = Console.in.readInt();

    if (row == -1)
    {
        System.out.println("You have chosen not to move your ship.");
        break;
    }

    System.out.print("Enter the destination column(0 - 9, -1 to pass): ");
    column = Console.in.readInt();

    :
    :

} while ((insideGalaxy == false) || (isEmpty == false) || (withinRange == false));
```

Allow the user an option to quit

James Tam

6. Provide Clearly Marked Exits

Restoring defaults

- Getting back original settings



7. Provide Shortcuts

Experienced (power) users should be able to perform frequently used operations quickly

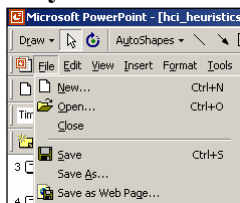
- Type-ahead (entering input before the system is ready for it)
- Navigation jumps
 - e.g., going to window/location directly, and avoiding intermediate nodes
- History systems
 - WWW: ~60% of pages are revisits
 - General: shortcuts for frequently used actions, shortcuts for the user's most recently used functions.



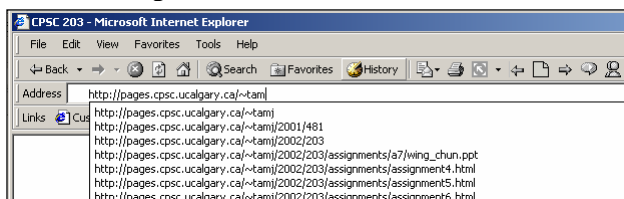
James Tam

7. Provide Shortcuts

Keyboard accelerators



Name completion



James Tam

The screenshot shows the Microsoft PowerPoint 2003 interface. The main slide content is as follows:

7. Provide shortcuts

Experienced users should be able to perform frequently used operations quickly

Strategies:

- keyboard and mouse accelerators
 - abbreviations
 - command completion
 - menu shortcuts
 - function keys
 - double clicking vs menu selection
- type-ahead (entering input before the system is ready for it)
- navigation jumps
 - e.g., going to window/location directly, and avoiding intermediate nodes
- history systems
 - WWW: ~60% of pages are revisits

Annotations on the left side of the screenshot:

- Keyboard accelerators for menus (points to the Alt key icon in the menu bar)
- Customizable toolbars and palettes for frequent actions (points to the toolbar)
- Split menu, with recently used fonts on top (points to the Font menu)
- Double-click raises toolbar dialog box (points to the toolbar)
- Double-click raises object-specific menu (points to the context menu icon in the toolbar)

Annotation on the right side of the screenshot:

- Scrolling controls for page-sized increments (points to the scroll bar on the right)

PowerPoint © Microsoft

James Tam

8. Deal With Errors In A Helpful And Positive Manner

People will make errors!

Errors we make

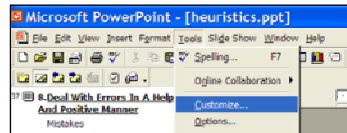
- Mistakes
- Slips



8. Deal With Errors In A Helpful And Positive Manner

Mistakes

- Arise from *conscious deliberations* that lead to an error instead of the correct solution



Slips

- *Unconscious behavior* that gets misdirected en route to satisfying goal
- Shows up frequently in skilled behavior
 - Usually due to inattention
- Often arises from similarities of actions

```
[csb 481 16 ]> rm *  
rm: remove file1 (yes/no)? y  
rm: remove file2 (yes/no)? y  
rm: remove file3 (yes/no)? y  
rm: remove file4 (yes/no)? y  
rm: remove file5 (yes/no)?
```

James Tam

Types Of Slips

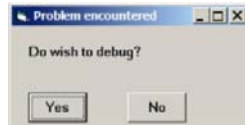
- 1) Capture error
- 2) Description error
- 3) Loss of activation
- 4) Mode error

James Tam

Types Of Slips: Capture Error

The frequently done activity “captures” the intended activity

- Occurs when common and rarer actions have same initial sequence
- “Force of habit results in the slip”



James Tam

Types Of Slips: Description Error

Intended action has much in common with others that are possible

- Usually occurs when right and wrong objects physically near each other
- “Perform the right action but on the wrong object”
 - e.g., Pour juice into coffee cup instead of glass



www.baddesigns.com

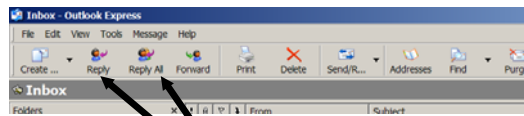
James Tam

Types Of Slips: Description Error (2)



PowerPoint

Print
preview??



Outlook Express

Oops! I only meant for that
letter to only go to one person.

James Tam

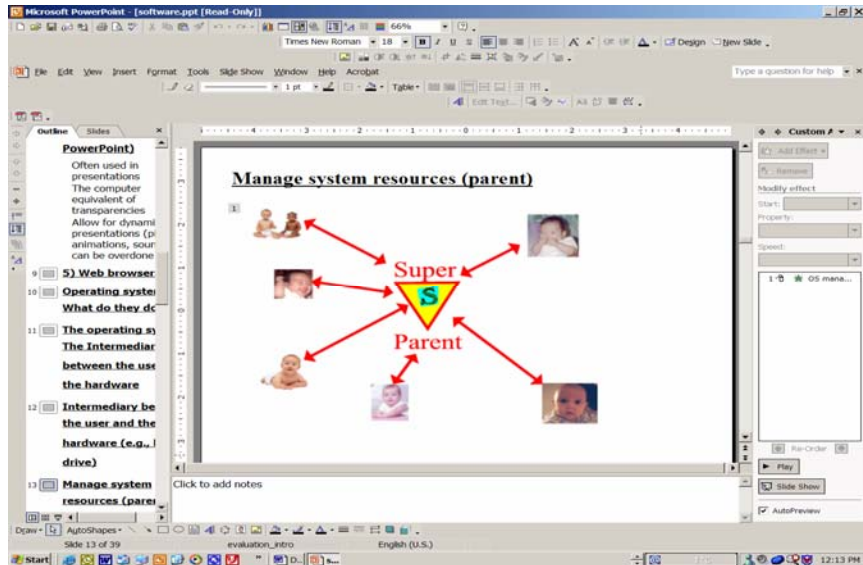
Types Of Slips: Loss Of Activation

Loss of activation

- Forgetting what the goal is while undergoing the sequence of actions
 - e.g., Start going to room and forget why you are going there
- “What the heck was I doing again?”
- Misordering a sequence

James Tam

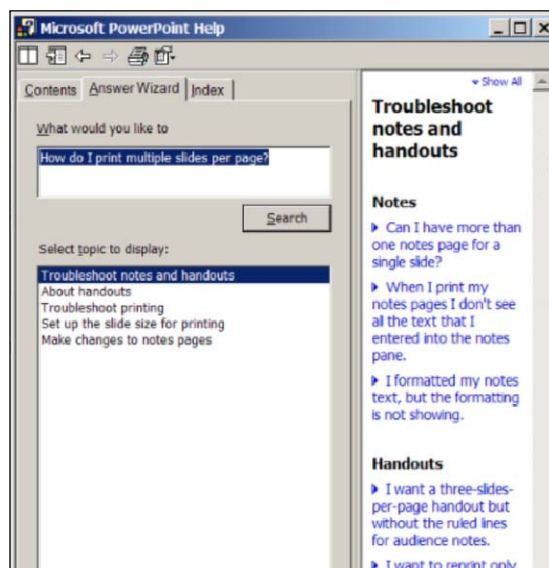
Loss Of Activation



PowerPoint © Microsoft

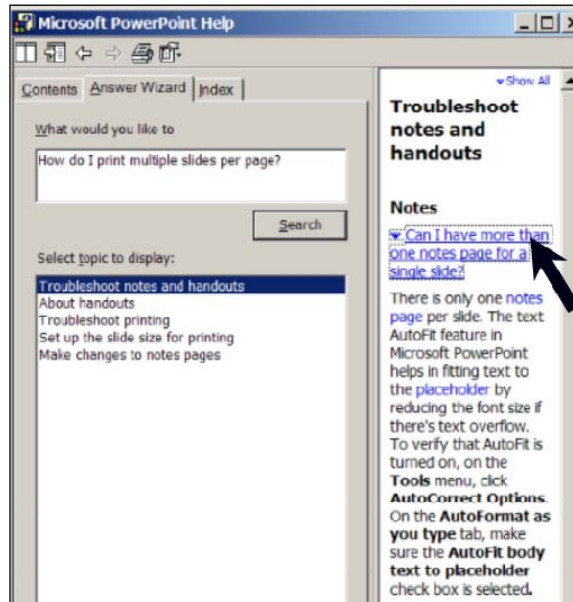
James Tam

Loss Of Activation (2)



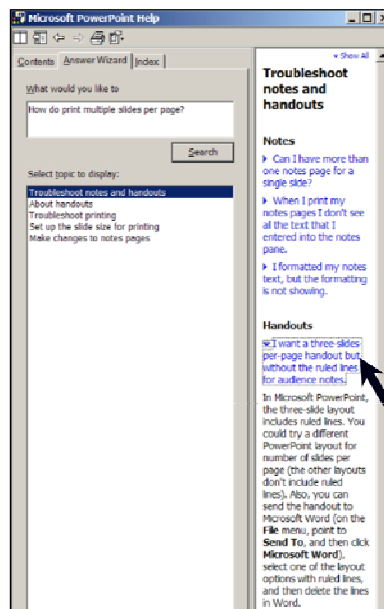
James Tam

Loss Of Activation (3)



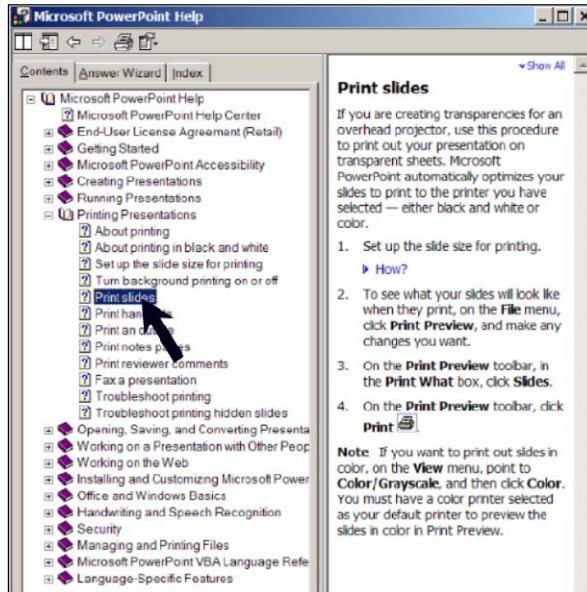
James Tam

Loss Of Activation (4)



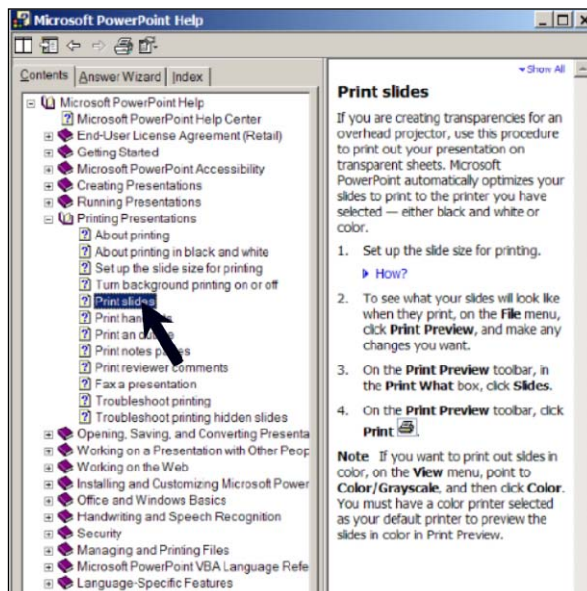
James Tam

...Loss Of Activation (???)



James Tam

...Loss Of Activation (???)



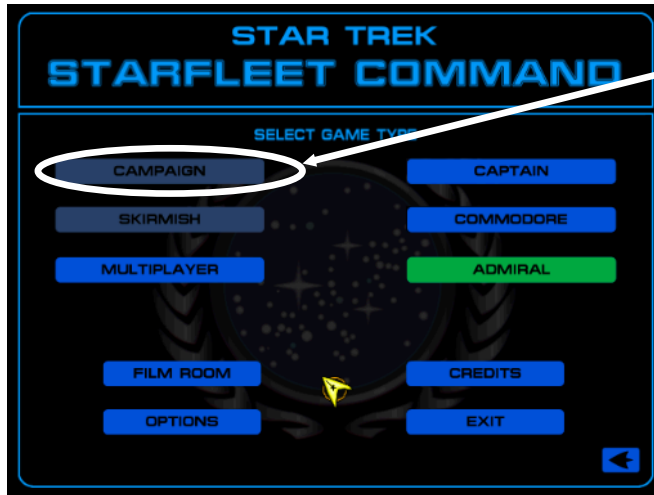
What was I trying to do again???

James Tam

Types Of Slips: Mode Error

Mode errors

- People do actions in one mode thinking they are in another mode



Game is in multiplayer mode only (cannot load saved games)

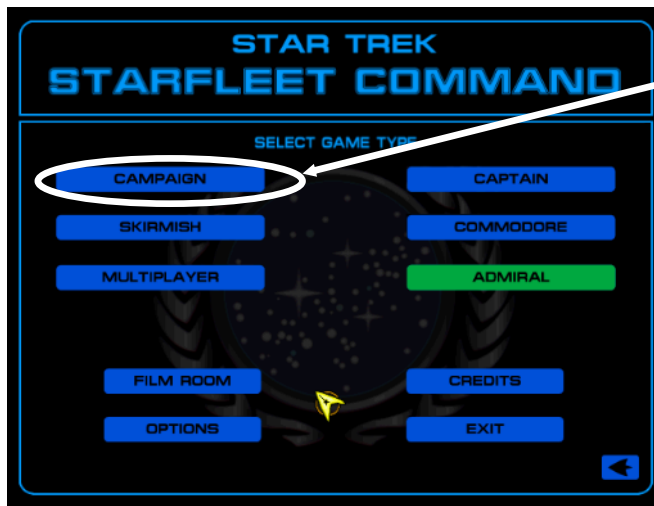
Star Fleet Command
© Interplay
Productions

James Tam

Types Of Slips: Mode Error (2)

Mode errors

- People do actions in one mode thinking they are in another mode



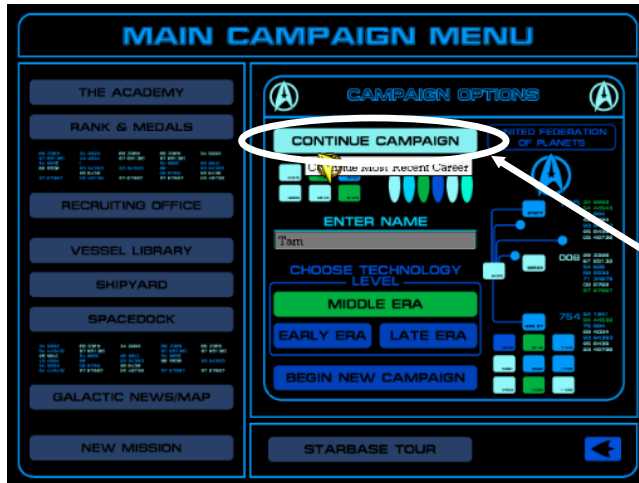
Game is in single and multiplayer mode (can start new campaigns or load existing games)

James Tam

Types Of Slips: Mode Error (3)

Mode errors

- People do actions in one mode thinking they are in another mode



In this mode it is possible to load a game

James Tam

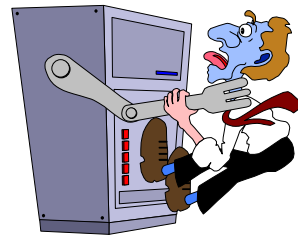
Designing For Slips

General rules

- Prevent slips before they occur
- Detect and correct slips when they do occur
- User correction through feedback and undo

Examples

- Capture errors
 - One action captures another
 - Allow actions to be undone rather than asking for confirmations
- Description errors
 - Correct action on the wrong object
 - Make objects look physically distinctive and/or far apart
- Loss of activation
 - Forgot goal
 - If system knows goal, make it explicit
 - If not, allow person to see path taken or shorten steps
- Mode errors
 - Mistake modes
 - Have as few modes as possible (possibly none)
 - Make modes highly visible



James Tam

Generic System Responses For Errors

General idea: Forcing functions

- Prevent / mitigate continuation of wrongful action
 - e.g., range selection for dynamic queries

Gag

- Deals with errors by preventing the user from continuing
 - e.g., cannot get past login screen until correct password entered

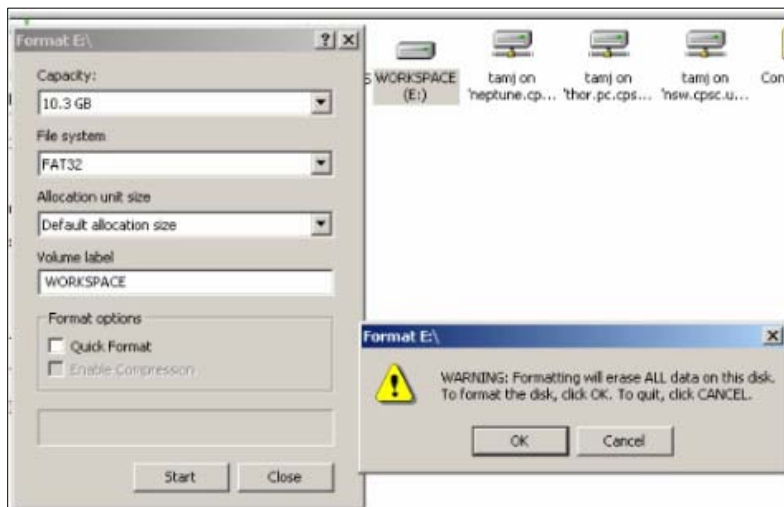
Warn

- Warn people that an unusual situation is occurring
- Better than nothing but when overused, becomes an irritant
 - e.g., audible bell, alert box

James Tam

Generic System Responses For Errors (2)

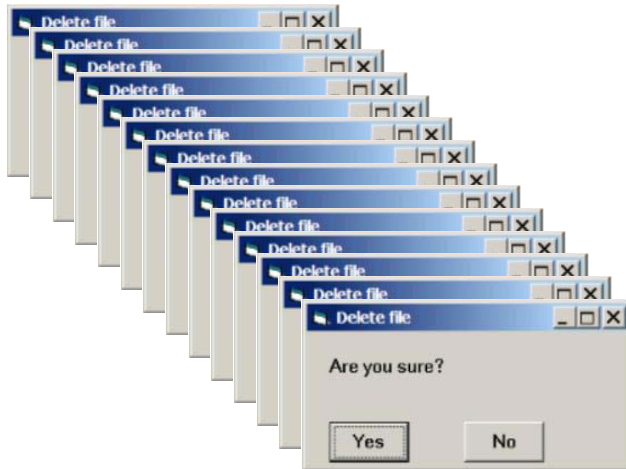
Warning (probably needed)



James Tam

Generic System Responses For Errors (3)

Warning (can get annoying, eventually the warning is ignored)



James Tam

Generic System Responses For Errors (4)

Do nothing

- Illegal action just doesn't do anything
- User must infer what happened
 - e.g., enter letter into a numeric-only field (non-numeric key pressed are ignored)
 - e.g., put a file icon on top of another file icon (returns it to original position)

Self-correct

- System guesses legal action and does it instead - but leads to a problem of trust
 - e.g., spelling checkers

Lets talk about it

- System initiates dialog with user with solution to the problem
 - e.g., compile error brings up offending line in source code

James Tam

Generic System Responses For Errors (5)

Teach me

- System asks user what the action was supposed to have meant
- Action then becomes a legal one

James Tam

Error Handling In “The Good Ole Days”



What is “error 15762”?

James Tam

Examples Of Error Handling From The “The Good Ole Days”

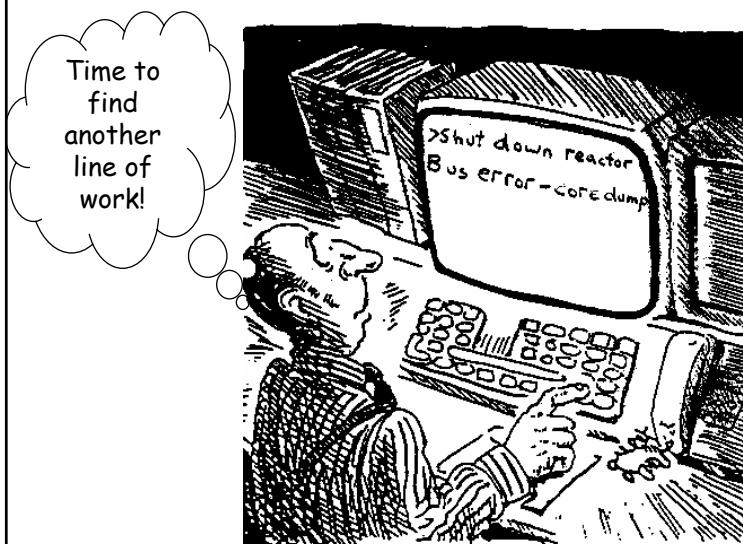
Below is the full list of error codes with[out] a suitable explanation to their meaning.

- 103: Insufficient Free store
- 105: Task Table Full
- 120: Argument line invalid or too long
- 121: File is not an object module
- 122: Invalid resident library during load
- 202: Object in use
- 203: Object already exists
- 204: Directory not found

List of error codes from the AmigaDos OS

James Tam

Errors: Don't Make Things Seem Worse Than They Really Are



James Tam

Rules Of Thumb For Error Messages

1. Polite and non-intimidating

- Don't make people feel stupid
 - Try again, bonehead!

2. Understandable

- Error 25

3. Specific

- Cannot open this document
- Cannot open "chapter 5" because the application "Microsoft Word" is not on your system

4. Helpful

- Cannot open "chapter 5" because the application "Microsoft Word" is not on your system. Open it with "WordPad" instead?

James Tam

Examples Of Dealing With Errors In A Positive And Helpful Manner

Prevent errors

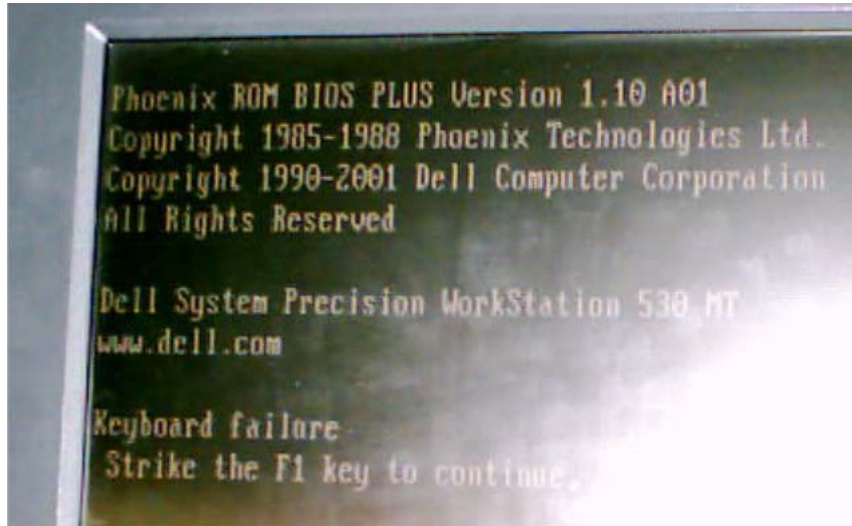
- Try to make errors impossible
- Modern widgets: only "legal commands" selected, or "legal data" entered

Provide reasonableness checks on input data

- On entering order for office supplies
 - 5000 pencils is an unusually large order. Do you really want to order that many?

James Tam

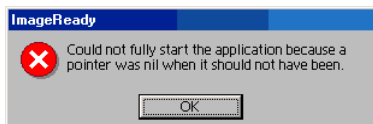
I'd Rather Deal With The Any Key Instead!



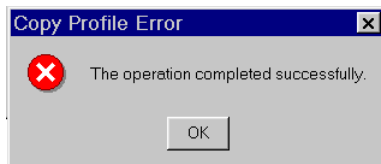
Picture courtesy of James Tam: An error message from a Dell desktop computer

James Tam

Examples Of Bad Error Messages (2)



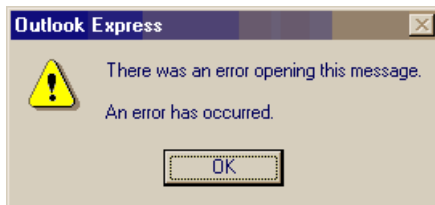
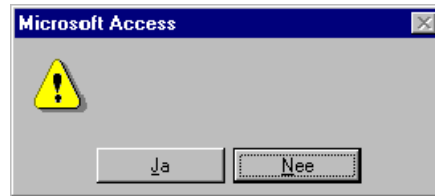
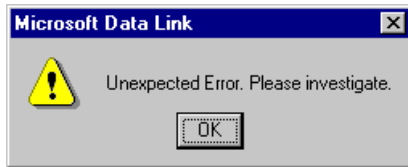
Adobe's ImageReady



Microsoft's NT Operating System

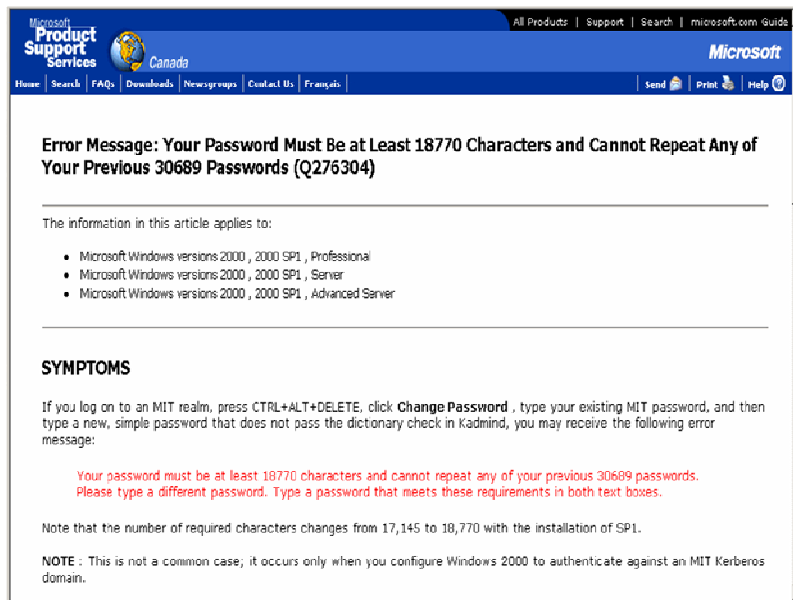
James Tam

Examples Of Bad Error Messages (3)



James Tam

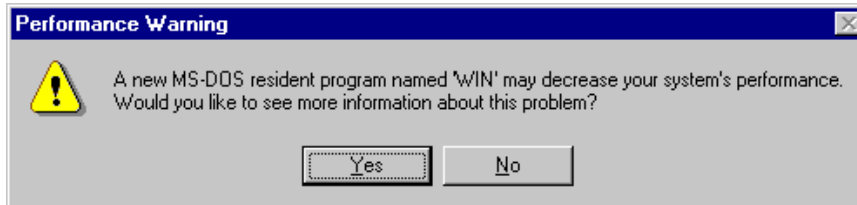
Examples Of Bad Error Messages (4)



Microsoft help page?

James Tam

Errors: Giving Away Too Much Information?



Windows 95 dialog box

James Tam

9. Provide Help

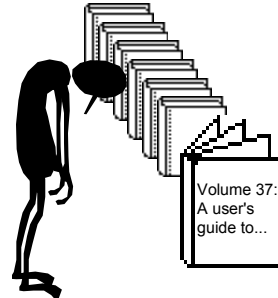
Help is not a replacement for bad design!

Simple systems:

- Walk up and use; minimal instructions

Most other systems:

- Feature rich
- Some users will want to become “experts” rather than “casual” users
- Intermediate users need reminding, plus a learning path (novice to advanced)



James Tam

Documentation And How It's Used

Many users do not read manuals

- Prefer to spend their time pursuing their task
- Paper manuals unavailable in many businesses!
 - e.g. single copy locked away in system administrator's office

Typical usage scenarios:

- Usually used when users are in some kind of panic, need immediate help
 - Indicates need for online documentation, good search/lookup tools
 - Online help can be specific to current context
- Sometimes used for quick reference
 - Syntax of actions, possibilities...
 - List of shortcuts ...

James Tam

Types Of Help

Reference Manuals

Reminders

- Reference cards
- Keyboard templates

Context-sensitive help

- Tool tips
- What's this (Balloon help)

Wizards

Tips

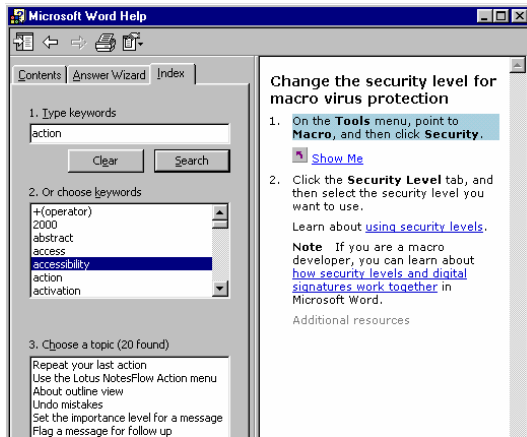
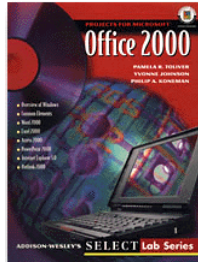
James Tam

Reference Manuals

Traditional form of help

Detailed

Paper or online



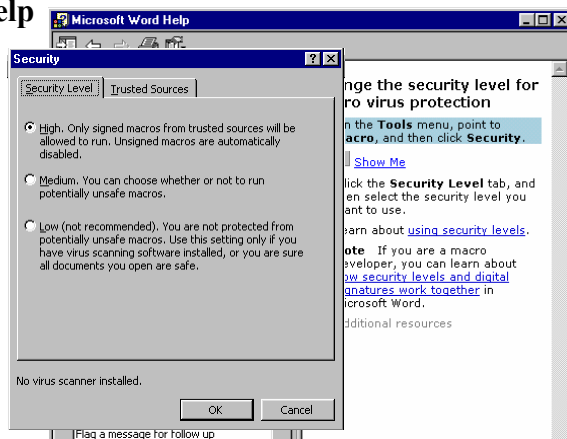
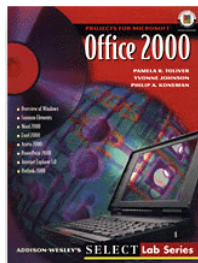
James Tam

Reference Manuals

Traditional form of help

Detailed

Paper or online



James Tam

Reminders

Short reference cards

- Expert user who just wants to check facts
- Novice who wants to get overview of system's capabilities

WEAPONS	
Fire	Spacebar
Select Weapon/Group	Enter
Chain-Fire/Group-Fire Toggle	\
Alpha Strike	P
Add Weapon to Group 1-5	Shift+1-5
Fire Group 1, 2, 3	Num Lock, /, * (Keypad)
Fire Current Group	.
Jettison Ammunition	K
TARGETING	
Cycle Through All Targets	T
Previous Target	R
Target Nearest Enemy	E
Target Friendly (Lancemate)	F
Target Object Underneath Reticle	Q
Inspect Targeted Object	I
Targeting OFF	Ctrl+T

HUD SYSTEMS	
Cycle through Multi-Function Display	F1
DISPLAYS	
Radar Display (Normal/Full Screen/OFF)	F2
Damage Display	F5
HTAL (Head/Torso/Arms/Legs)	
Armor Damage Report	F6
Enable/Disable/Save/HUD	F11
Remove Cockpit	U
Objectives/Briefing Summary	F12
CAMERAS	
Target View	F4
Rear View	F7
Down View	F8
Weapon View	F9
Weapon View (Full Screen, After Launch)	F10

From "Mechwarrior 2: Mercenaries" © Microprose

James Tam

Reminders (2)

Keyboard templates

- Shortcuts/syntactic meanings of keys: recognition vs. recall.

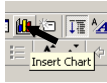


From "Gunship" © Microprose

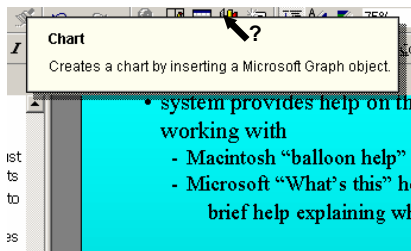
James Tam

Context-Sensitive Help

Tool tips

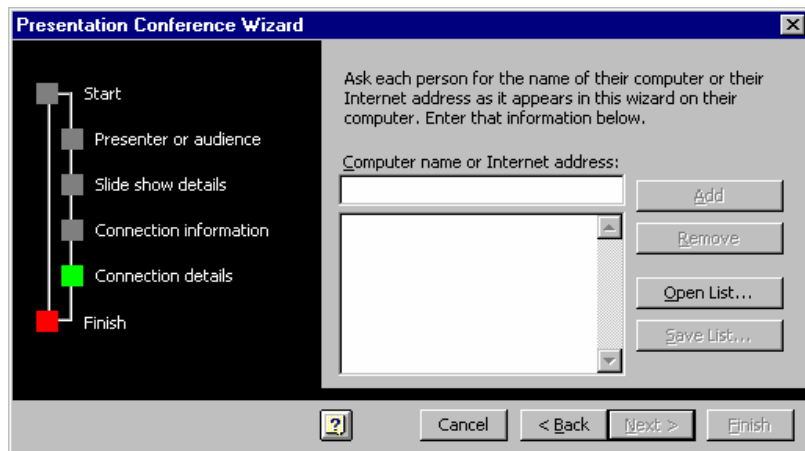


What's this (Windows) or Balloon help (Mac)



James Tam

Wizards



James Tam

Tips

- **Introduce advanced features**
- **Point out incorrect use**



Change Awareness

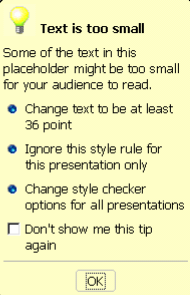
- What is change awareness
- Why is it an important area of study
- Issues in change awareness
- Future work



James Tam

Tips

- **Advanced features**
- **Point out incorrect use**



James Tam

Conducting A Heuristic Evaluation

It's a compromise between extensive style guides and intuition-based inspections



Style guides



Heuristic
evaluation

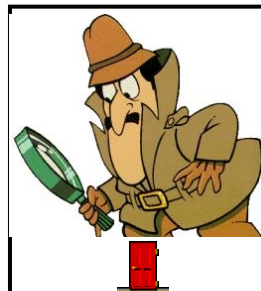
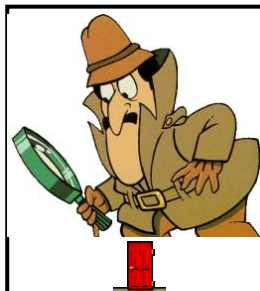
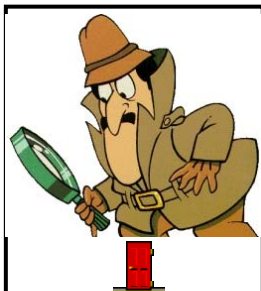


Inspections

James Tam

Conducting A Heuristic Evaluation (2)

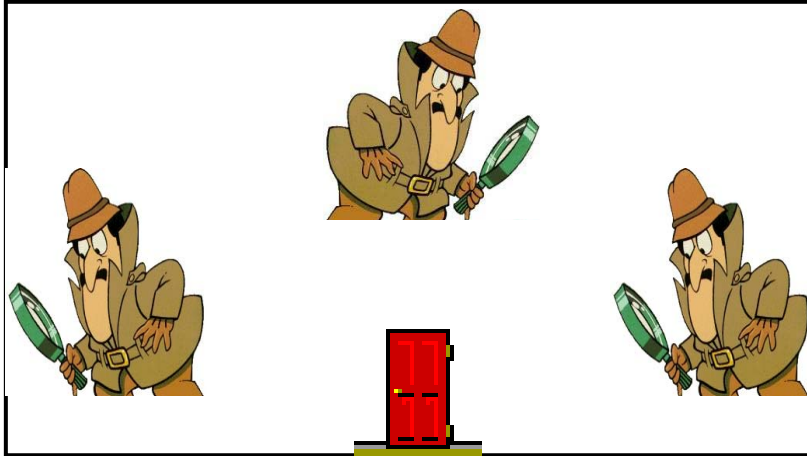
- **Employ a small set of evaluators (3–5) examine interface using heuristics as a structure**
- **Each person evaluates the system according to the heuristics individually (~1 – 2 hours)**



James Tam

Conducting A Heuristic Evaluation (3)

- After this bring the evaluators together to pool/compare their results

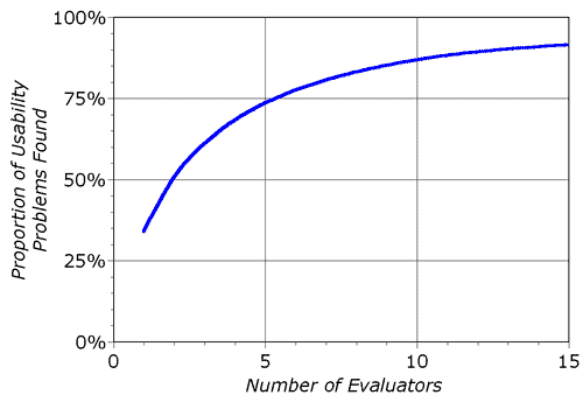


James Tam

Conducting A Heuristic Evaluation (4)

Benefits of group evaluations

- Single evaluator only catches ~35% of usability problems
- 5 evaluators catch ~75%



Statistics from "Usability Engineering" by Jakob Nielsen

James Tam

Heuristic Evaluations: Who Should/Could Conduct Them

Interface experts



End users



Software development team



Double experts



James Tam

Heuristic Evaluations: Types Of Systems To Evaluate

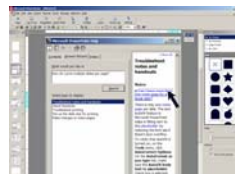
Low fidelity paper prototypes



Medium fidelity prototypes



Completed systems



James Tam

You Now Know

Nine principles of design

- Simple and natural dialog
- Speak the user's language
- Minimize user's memory load
- Be consistent
- Provide feedback
- Provide clearly marked exits
- Provide shortcuts
- Deal with errors in a positive manner
- Provide help

James Tam

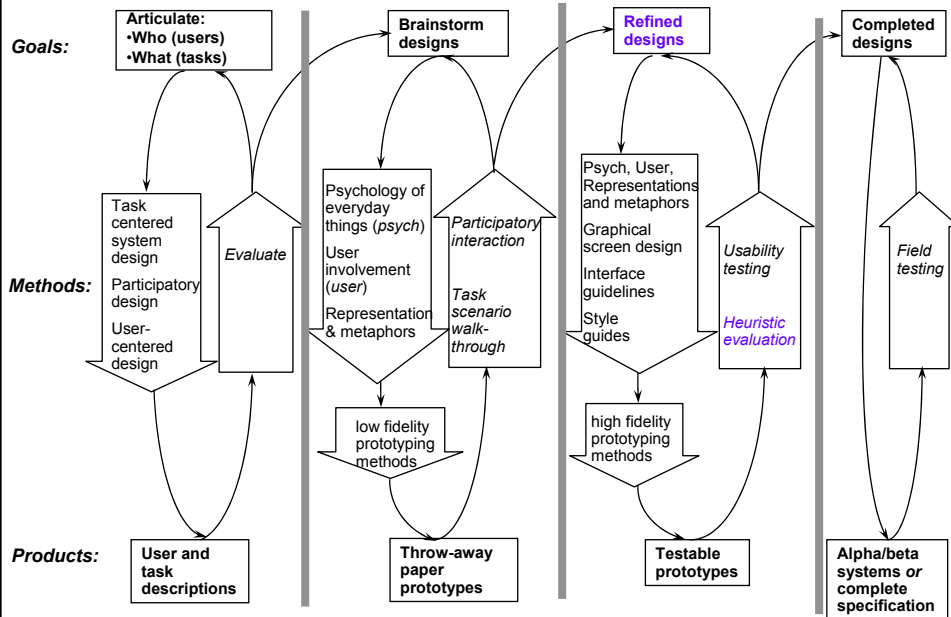
You Now Know (2)

Heuristic evaluation

- Principles can be used to systematically inspect the interface for usability problems

James Tam

Interface Design And Usability Engineering



This diagram is a variation of the one presented by Saul Greenberg

James Tam