

CS325

USER INTERFACE AND USER EXPERIENCE DESIGN

Week 10

Dr Frank Guan

Agenda for Today's Lecture

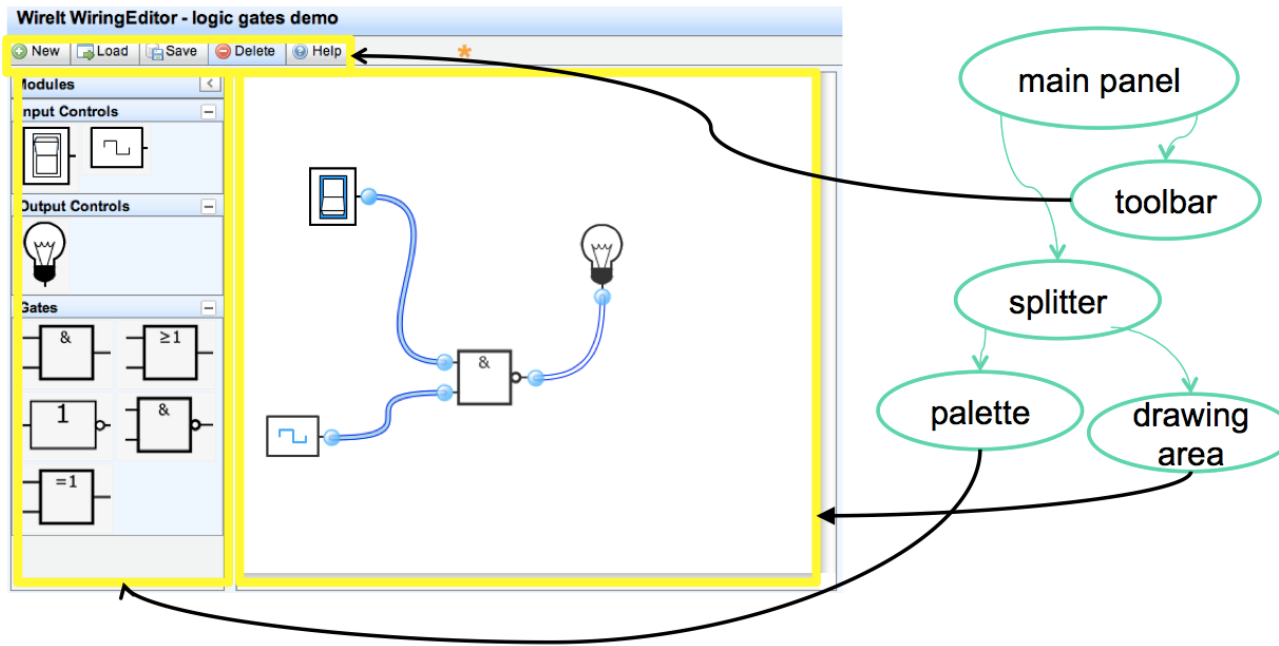
- GUI Patterns
- Widgets
- Evaluation

GUI Patterns

Patterns

- View Tree
- Listeners
- Model-View-Controller (MVC)

View tree

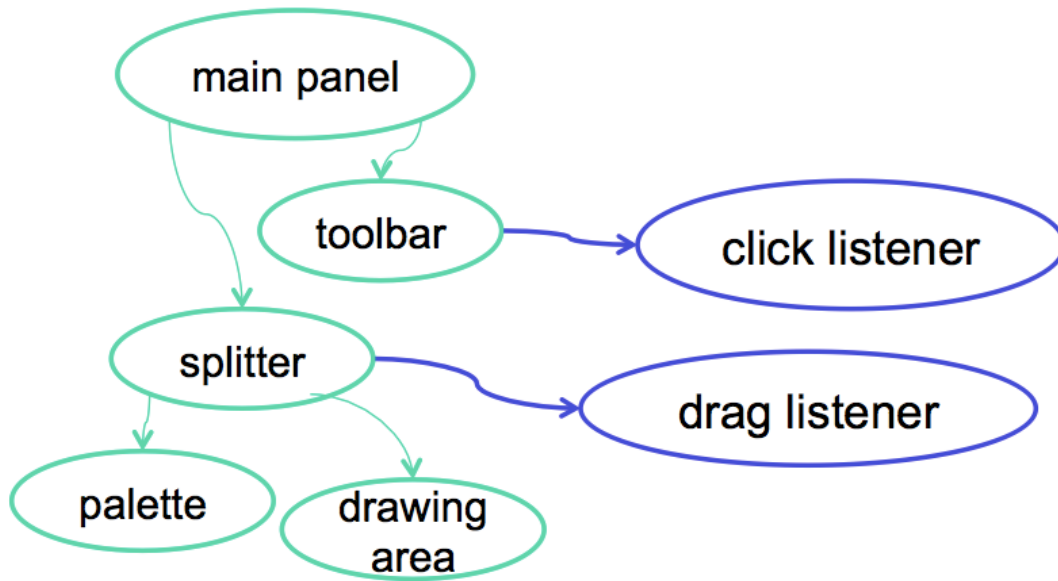


- A view is an object that covers a certain area of the screen, generally a rectangular area called its bounding box.
- Views are arranged into a hierarchy of containment, in which some views contain other views.

View tree

- Virtually every GUI system has some kind of **view tree**
- Input
 - GUIs receive keyboard and mouse input by attaching listeners to views (more on this in a bit)
- Output
 - GUIs change their output by mutating the view tree
 - A redraw algorithm automatically redraws the affected views
- Layout
 - Automatic layout algorithm traverses the tree to calculate positions and sizes of views

Listeners



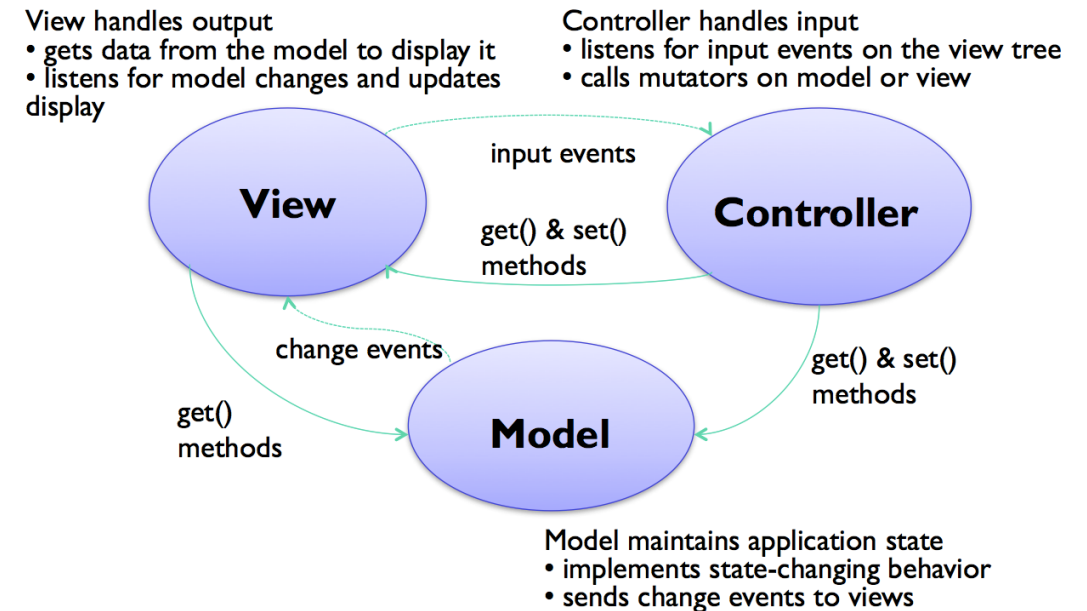
- To handle mouse input, for example, we can attach a handler to the view that is called when the mouse is clicked on it.
- Handlers are variously called **listeners**, event handlers, subscribers, and observers.

Listeners

- GUI input handling is an example of the Listener pattern
 - aka Publish-Subscribe, Event, Observer
- An event source generates a stream of discrete events
 - e.g., mouse events
- Listeners register interest in events from the source
 - Can often register only for specific events - e.g., only want mouse events occurring inside a view's bounds
 - Listeners can unsubscribe when they no longer want events
- When an event occurs, the event source distributes it to all interested listeners

Model-View-Controller

- Architecture for developing programs with modular, reusable components
- Allows separation of application state (model) from application display (view) and application logic (controller)



MVC

Model

- Maintains application data
- Provides methods to access and modify data
- Notifies observers when data changes

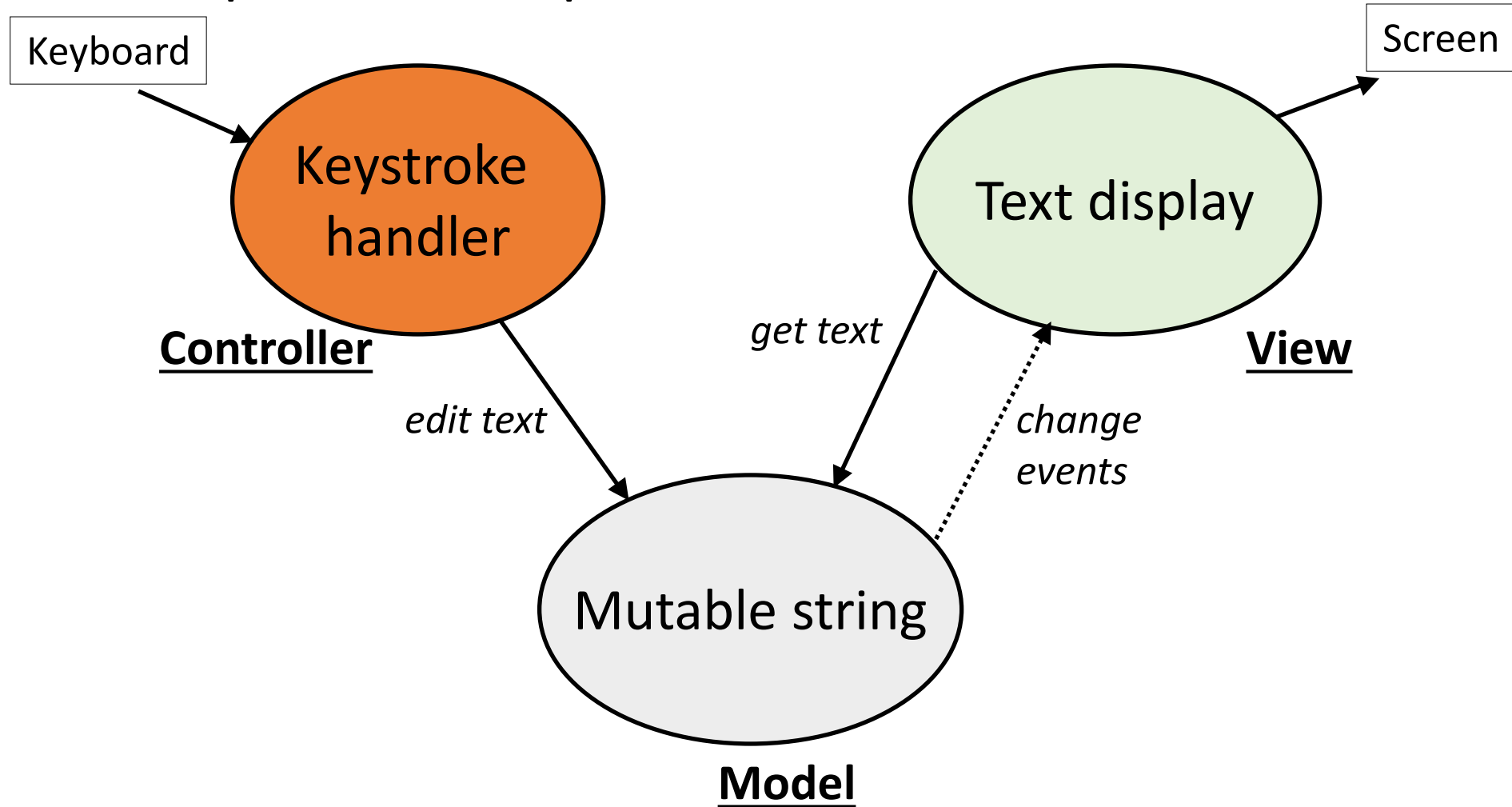
View

- Maintains application display
- Updates view by listening to and querying model
- Can have multiple views for the same model

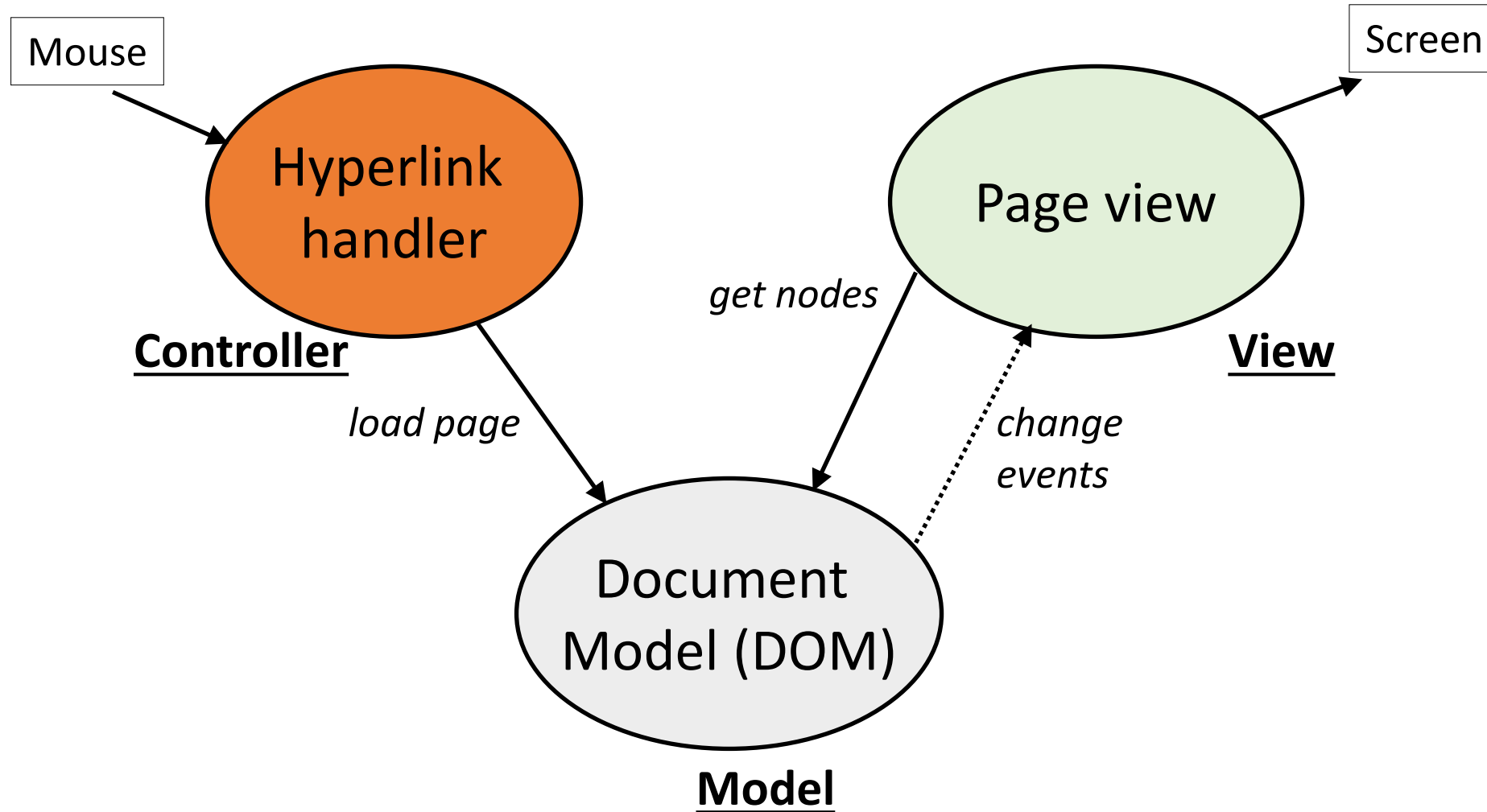
Controller

- Handles user input
- Mediates between view and model
- Manages lifecycle of other objects
- Views and controllers can be reusable across applications

One simple example



Another example



Widget (combined View and Controller)

- Theoretically, it's good to separate input and output into separate, reusable classes.
- Practically, it isn't always feasible because input and output are tightly coupled in graphical user interfaces.
- As a result, the view and controllers are fused together into a single class, often called a **component** or a **widget**.
- The MVC pattern is therefore changed to Model-View.

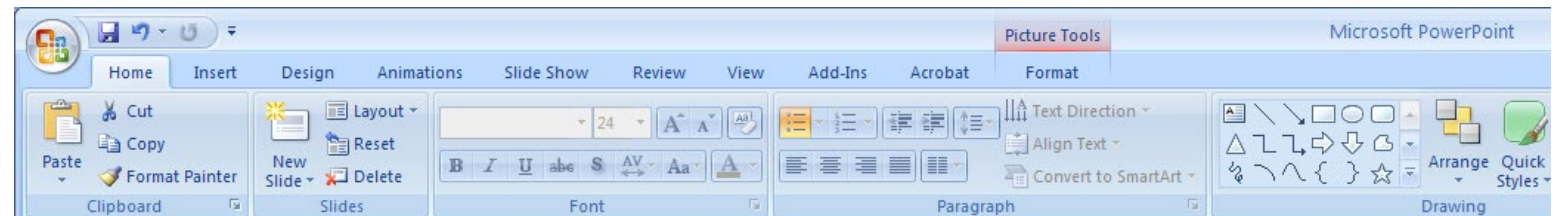
Widgets

Widgets

- Reusable user interface components
 - Also called components, controls, interactors, etc.
 - Handle both input and output
- Includes a view, a controller, and possibly a model
 - Embedded model - data included in widget; needs to be copied in and out
 - Linked model - data stored in model object, which provides interface for accessing and editing
 - Allows *binding* of widgets to data objects

Interface elements

- menu
- dialog boxes
- buttons
- radio buttons
- check boxes
- sliders
- toolbars
- tabs
- progress indicators
- palettes



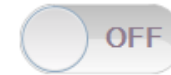
Widgets for Boolean Options

☒ I agree to the terms of service for this site.

Style = Metro



Style = IOS5



Style = Android



Style = BrushedMetal



Style = Iphone



Style = Modern



Style = Carbon



Style = OSX



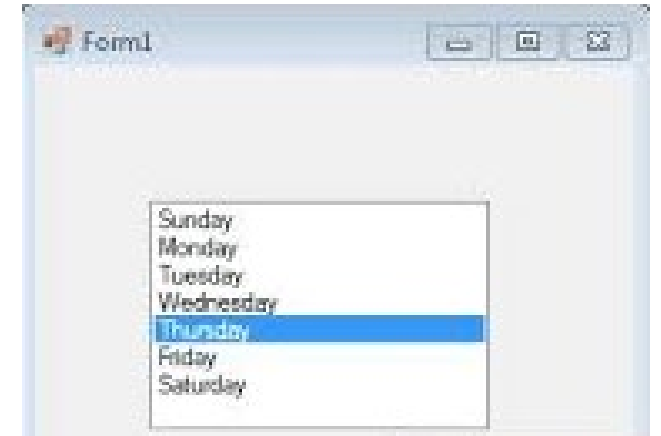
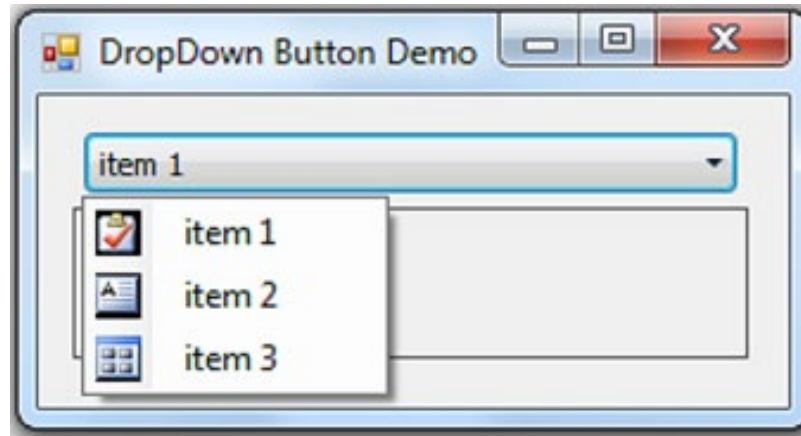
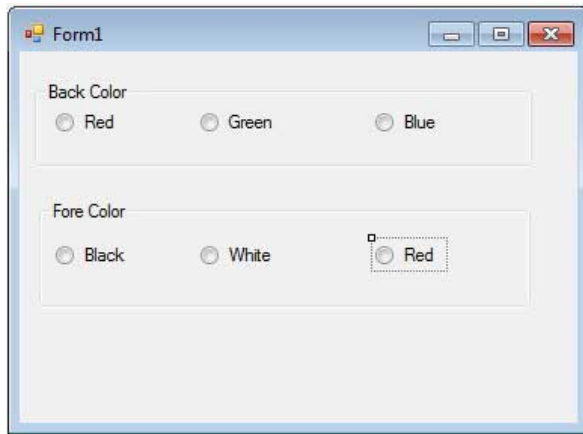
Style = Fancy



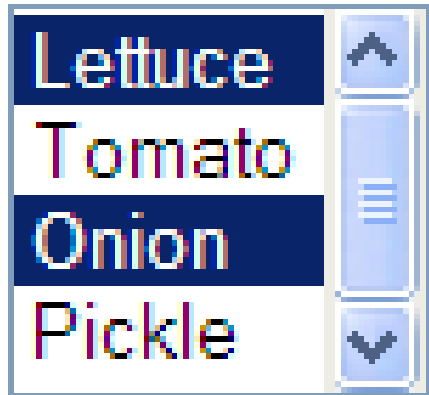
Style = PlainAndSimple



Widget for 1-of-N choices



Widgets for multiple selections

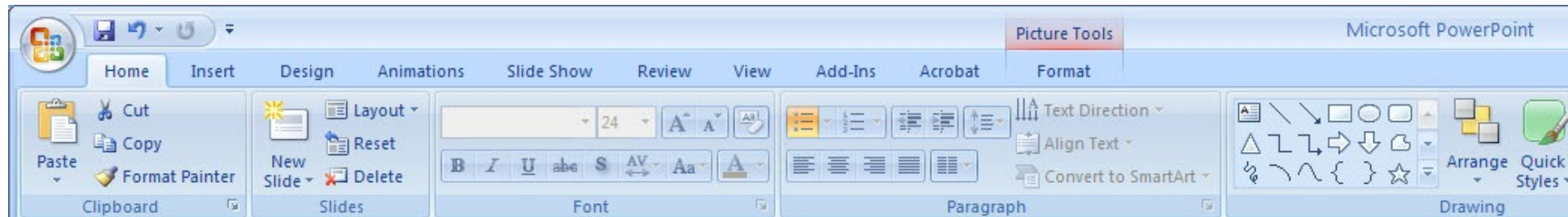


- ☒ Lettuce
- ☐ Tomato
- ☒ Onion
- ☐ Pickle

Menus

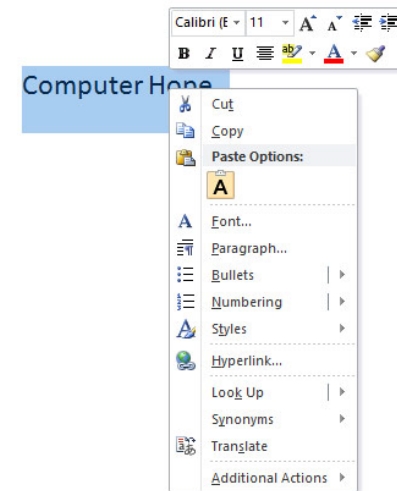
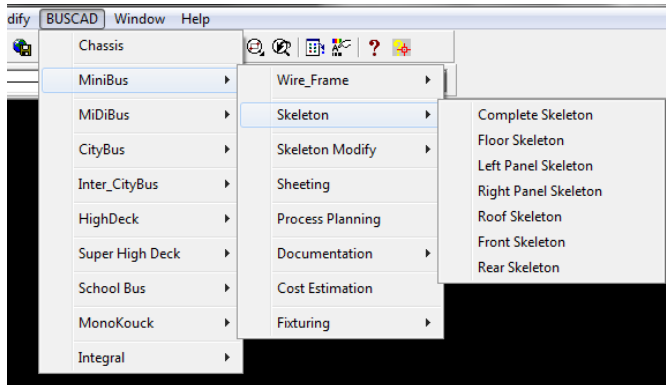
Menus

- Key advantages:
 - 1 keystroke or mouse operation vs. many
 - No memorization of commands
 - Limited input set



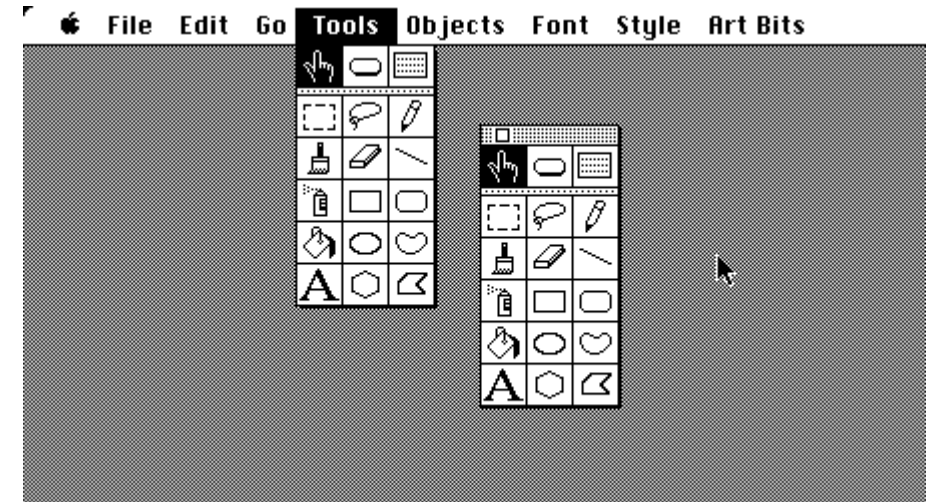
Types of menus

- pull-down
- pop-up

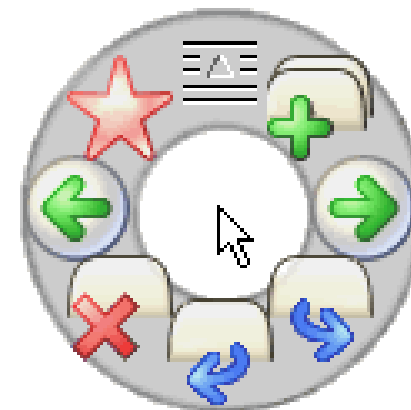


Types of menus

- tear-off (stay on the screen until explicitly requested to go away)



- pie
(menu options arranged in a circle).



Menu

- Usually hierarchical
 - each menu option opens up another menu
- Menus used in many types of system
 - with GUIs
 - with form fill-in
 - with most other software systems
- Design of menu structure is crucial
- Access to menu
 - GUI – mouse point/click
 - non-GUI – F keys
 - joy-stick, cursor keys

Advantages

- Advantages
 - Recognize options instead of recalling
 - Structures learning (logical grouping and meaningful naming)
 - Decomposes a complex interaction into a series of smaller steps
 - Simplified the tasks
 - reduces error rates
 - reduces keystrokes
 - reduces training

Disadvantages

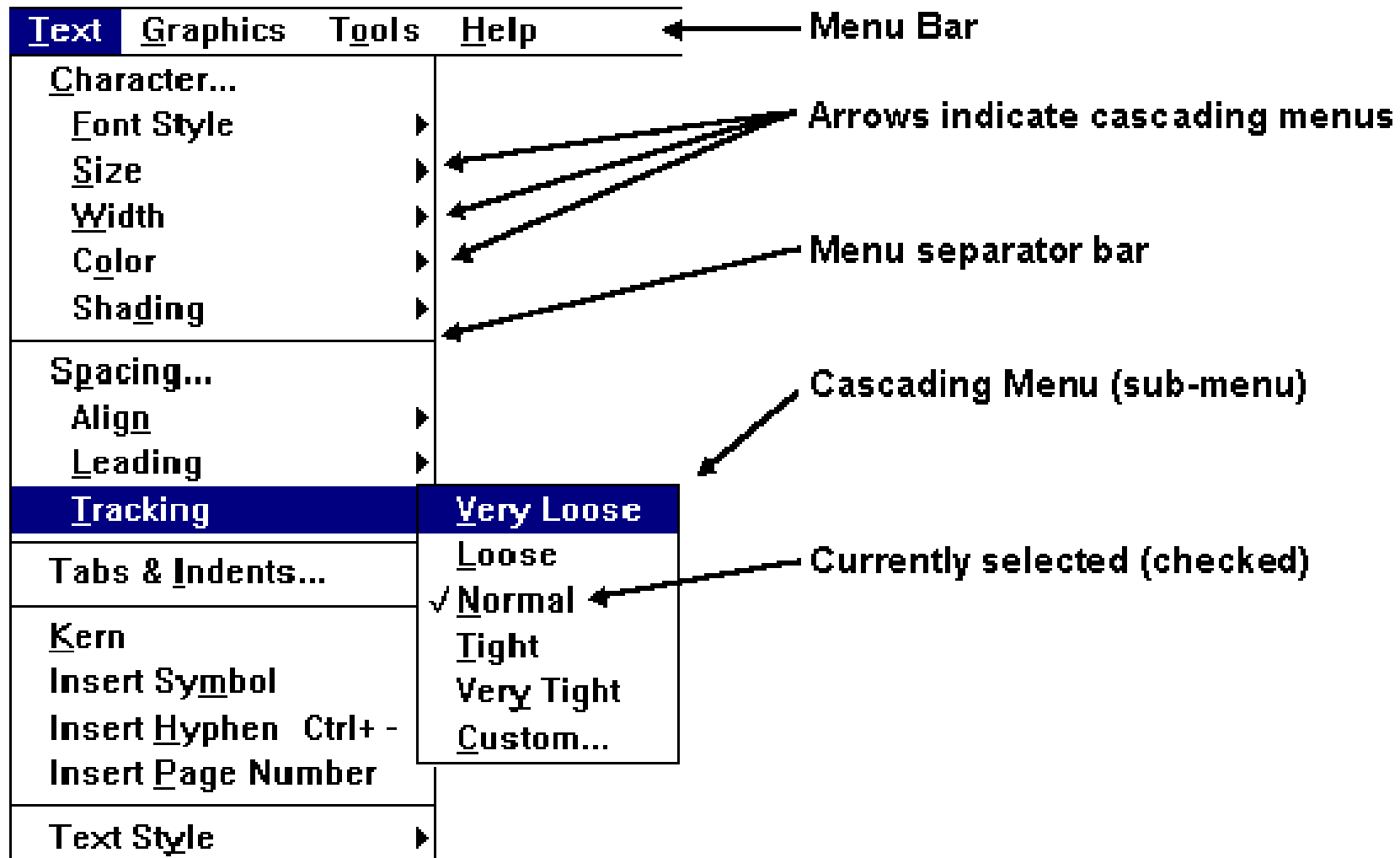
- Dangers of too many menus
- Can consume lots screen space
 - Consider country selection
- The user needs to follow the designer's logic of organization
- Sometimes less efficient
 - Especially with deep structures
 - Slow for frequent users

Choose Your Country or Region

Africa, Middle East, and India

Armenia	Israel	Oman
Bahrain	Jordan	عُمان
البحرين	الأردن	Qatar
Botswana	Kenya	قطر
Cameroun	Kuwait	Saudi Arabia
République Centrafricaine	الكويت	المملكة العربية السعودية
Côte d'Ivoire	Madagascar	Sénégal
Egypt	Mali	South Africa
مصر	Maroc	Tunisie
Guinea-Bissau	Maurice	Uganda
Guinée	Mozambique	United Arab Emirates
Guinée Equatoriale	Niger	الإمارات العربية المتحدة
India	Nigeria	

Pull-down menu

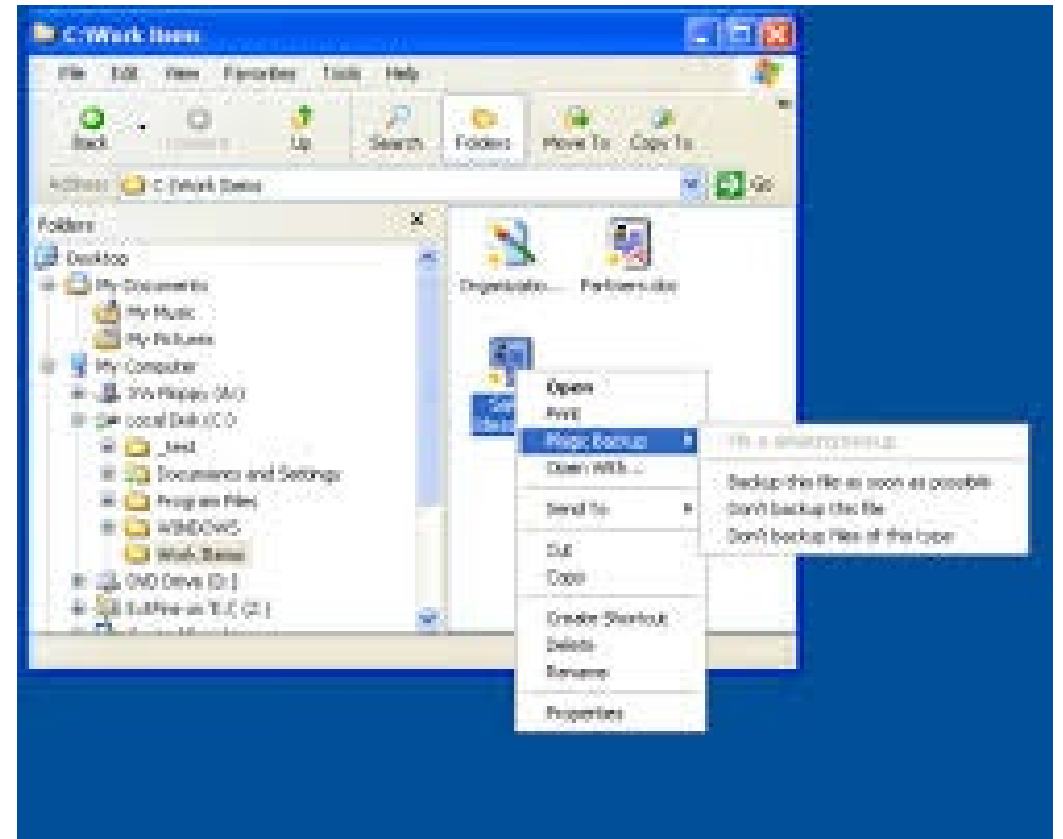


Pull-down menu

- **Pull down menus are used extensively in Windows applications.**
- **Advantages:**
 - The menu names are visible textual items on the menu bar, so the user can see the main functions of the software at all times.
 - Menu titles, or menu items (such as Open Project on the File menu) should be unambiguous if they are well chosen, whereas, it can be difficult to design satisfactory icons for abstract functions is the standard icon for open.
 - The menu bar takes up very little space on the screen.
 - The menu options are generally available by means of access keys (holding down the Ctrl key then typing underlined letters of the menu title and menu item required, so Ctrl+O for 'Open Project' on the 'File' menu).
 - Functions are grouped together to form a set of menu items that are related, this helps users to know where to look for facilities.

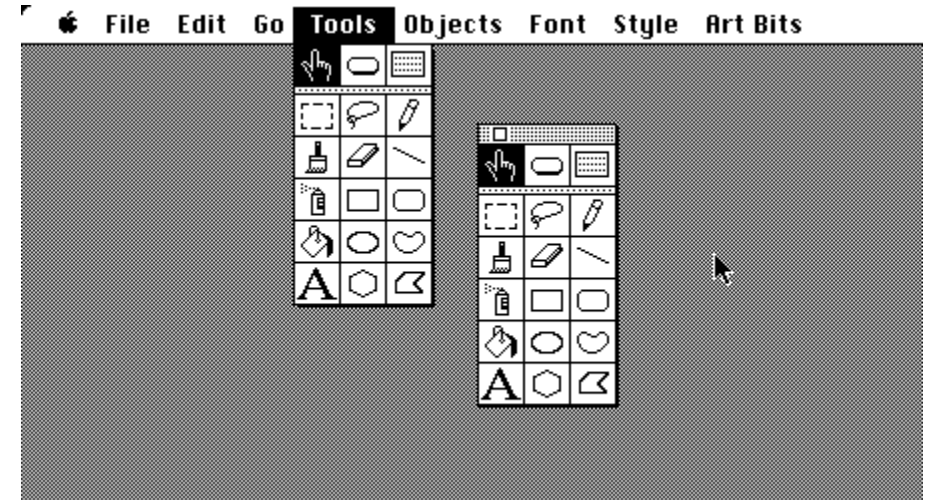
Pop-up menu

- A pop-up menu is one that is displayed on top of a form or at the current cursor location, and is not attached to the menu bar.
- You can create a pop-up form to display information to a user or to prompt a user for data.
- When a pop-up form is modeless, you can access other objects and menu commands while the form is open.



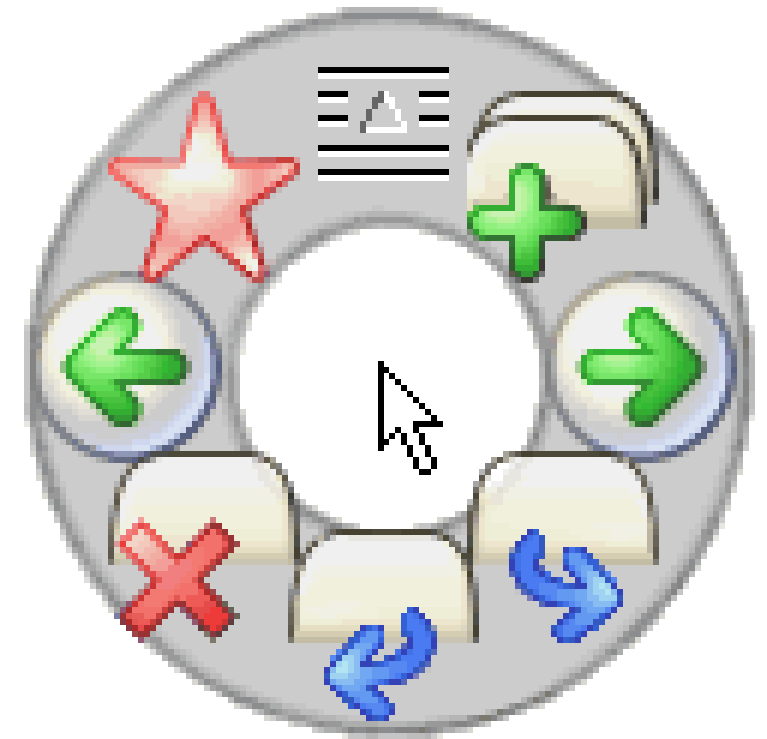
Tear-off menu

- Allow the user to tear off a menu when the user is likely to want to make a number of choices from a pull-down menu.
- By default, allow a user to tear off a menu when the user will frequently access the menu.
- Allow the user to configure any pop-up or cascaded menu, other than an option menu, to be torn off.



Pie menu

- Pie Menus are a naturally efficient user interface technique: directional selection of pie slice shaped targets.
- The cursor starts out in the center of the pie, so all targets are large, nearby, and in different directions.
- Fitts' Law explains the advantages of pie menus, relating their **fast selection speed** and **low error rate** to their large target size and small distance.
- Pie menus are easy for novice users, who just follow the directions, and efficient for experienced users, who can quickly "mouse ahead" once they know the way.



Dialog Box

Example

The image shows a 'Text Field Properties' dialog box with a title bar and a close button. It contains several tabs: 'General', 'Appearance', 'Options', 'Actions', 'Format', 'Validate', and 'Calculate'. The 'General' tab is active. Inside this tab, there are two text input fields: 'Name' with the value 'First name' and 'Tooltip' with the value 'First name (required)'. Below these is a section titled 'Common Properties' which contains a 'Form Field' dropdown set to 'Visible', an 'Orientation' spinner set to '0' degrees, and two checkboxes: 'Read Only' (unchecked) and 'Required' (checked). At the bottom left is a 'Locked' checkbox (unchecked), and at the bottom right is a 'Close' button.

Text Field Properties

General Appearance Options Actions Format Validate Calculate

Name First name

Tooltip First name (required)

Common Properties

Form Field: Visible

Orientation: 0 degrees

☐ Read Only

☒ Required

☐ Locked

Close

In general about dialog box

- Principle
 - User's main task is to provide data in labeled fields clustered in one or more screens
- Data
 - Binary choice (Yes/No, Female/Male)
 - Selection from brief list (days of week, set of colours)
 - Large domain (personal names, chemical formulae)
 - Essential unbounded (exploratory paragraphs, meteorological data)
- Can be replaced with a series of menu choices, however, may become extremely tedious

Guidelines (1/2)

- Meaningful title
- Comprehensible instructions
- Logical grouping and sequencing of fields
- Visually appealing layout of the form
- Familiar field labels

Guidelines (2/2)

- Consistent terminology and abbreviations
- Visible space and boundaries for data entry fields
- Error correction for individual character
- Error messages for unacceptable values
- Optional fields should be marked
- Explanatory messages for fields

Evaluation

Overview

- What is evaluation?
- Goals of evaluation
- Formative vs. Summative evaluation
- Evaluation methods
 - Predictive
 - Heuristic Evaluation
 - Cognitive Walkthrough
 - Empirical

Evaluation

Gather data about the usability of a design for a particular activity by a specified group of users within a specified environment

Goals of Evaluation

- Assess extent of system's functionality
- Assess effect of interface on user
- Identify specific problems with system

Formative vs. Summative

- Formative evaluation
 - As project is forming
 - All through the lifecycle
 - Early, continuous and iterative
 - “Evaluating the design”
- Summative evaluation
 - After a system has been finished
 - Make judgments about final item
 - “Evaluating the implementation”

Multiple ways to evaluate

Empirical Assess with real users

Formal Models and formulas to calculate measures

Automated Software measures

Predictive/Critique Expertise and heuristic feedback

Some Evaluation Methods

- Predictive evaluation
 - Heuristic Evaluation
 - Cognitive Walkthrough
 - Pluralistic Walkthrough (not covered)
- Empirical evaluation
 - Lab studies
 - Field studies
 - “Think Aloud”

Predictive Evaluation

- Conducted with expert reviewers in the field
- HCI experts (not real users) interact with the system, find potential problems, and give prescriptive feedback
- Works best if they:
 - Haven't used earlier prototype
 - Are familiar with domain or task
 - Understand user perspectives

Predictive Evaluation Methods

- Heuristic Evaluation
- Cognitive Walkthrough
- Other methods

A portrait of Jakob Nielsen, a man with thinning grey hair and glasses, wearing a blue striped shirt and a patterned tie under a grey jacket. He is smiling slightly. The background is a blurred office interior.

Jakob Nielsen's 10 Usability Heuristics

Heuristic Evaluation

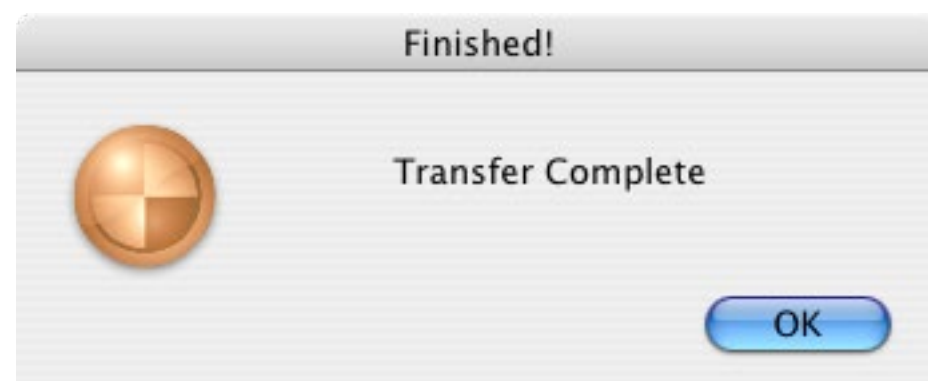
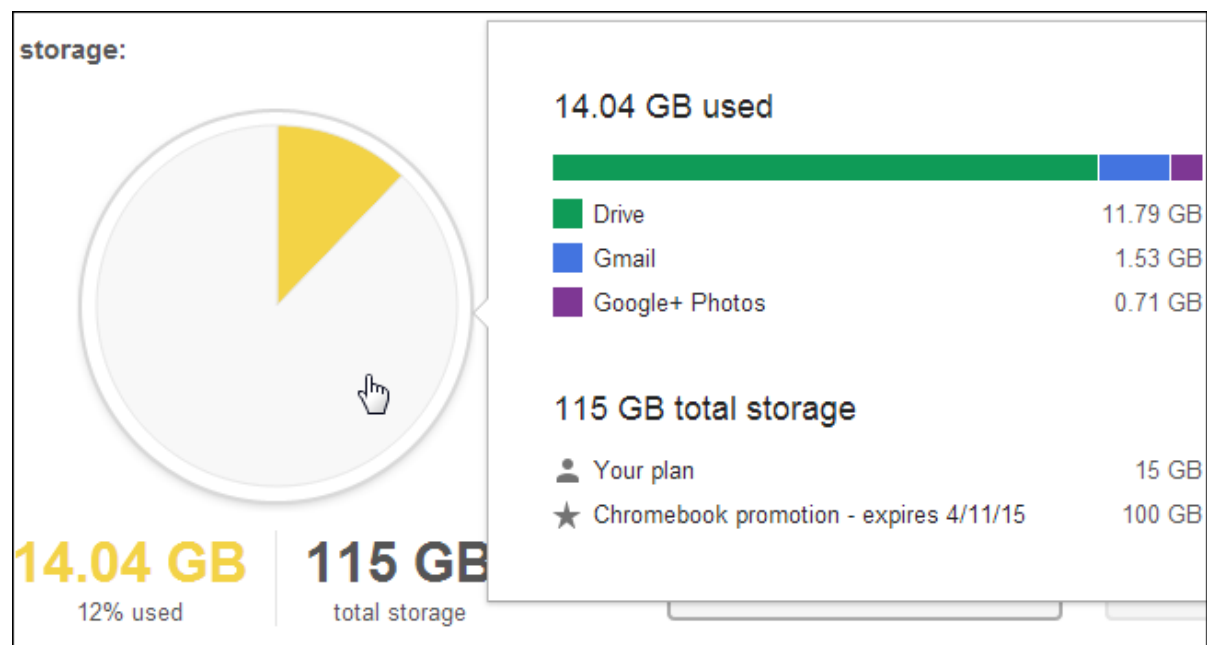
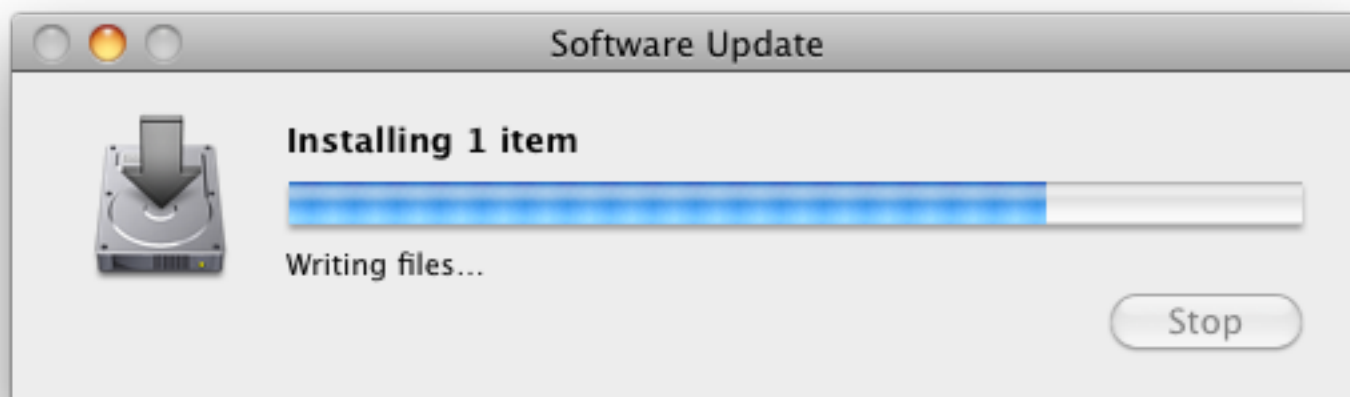
- Developed by Jakob Nielsen in the early 1990s
- Heuristics are developed based on broad “rules of thumb”
- Small set (3-5) expert usability evaluators assess system based on simple and general heuristics (principles or rules of thumb)
 - independently check for compliance with usability principles (“heuristics”)
 - different evaluators will find different problems
 - evaluators only communicate afterwards
 - findings are then aggregated
 - Can perform on working UI or sketches
- <http://www.nngroup.com/articles/ten-usability-heuristics/>

Ten Usability Heuristics

- Visibility of system status
- Familiar metaphors & language
- User Control & freedom
- Consistency
- Error prevention
- Recognition over recall
- Flexibility & efficiency
- Aesthetic & minimalist design
- Recognize, diagnose, & recover from errors
- Help & documentation

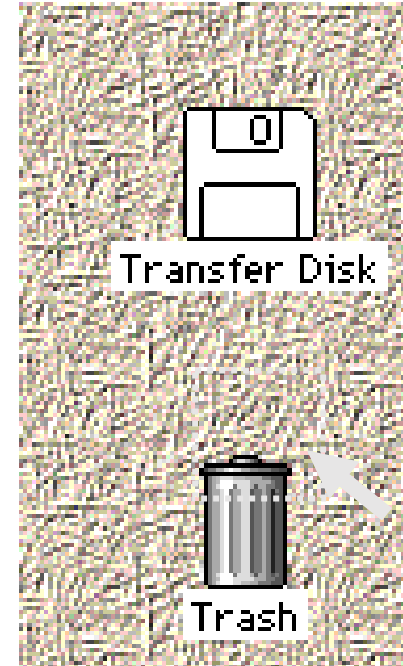
1/10 Visibility of system status

- Feedback of system's current state
- Page is loading, error loading page, something saved or deleted
- Show completion and what to do next



2/10 Familiar metaphors & language

- Match between system and real world
- Follow real world conventions
- Speak the user's language
- Familiar categories, language and choices
- Familiar components and widgets

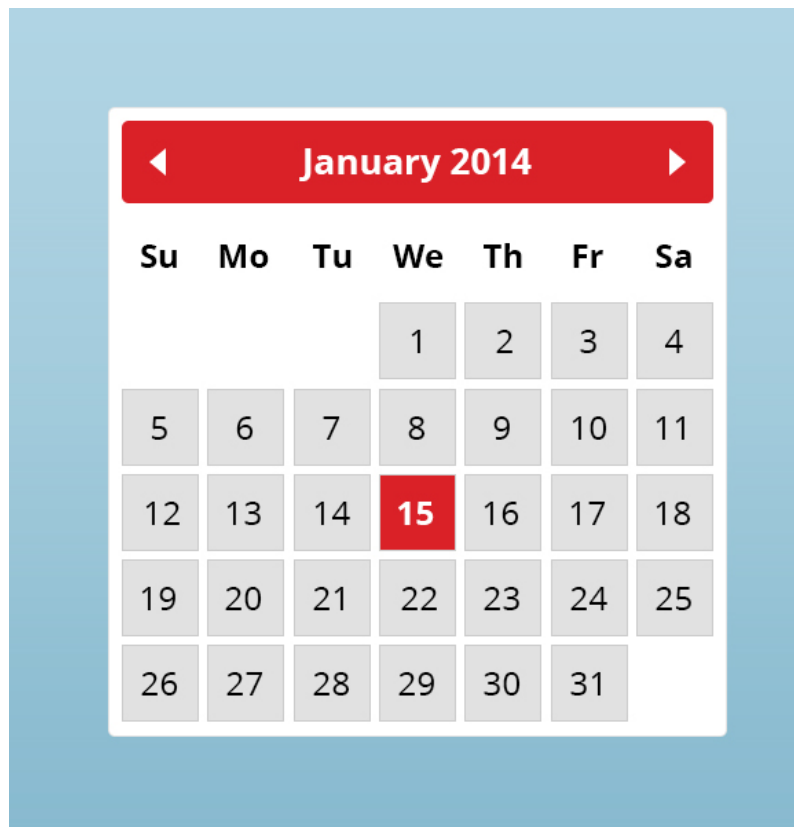


Old Mac Desktop: Dragging disk to trash should delete it, not *eject* it

State

FIND STORES

- Select a State
- Alabama
- Alaska
- Arizona
- Arkansas**
- California
- Colorado
- Connecticut
- D.C.
- Delaware
- Florida
- Georgia
- Hawaii
- Idaho
- Illinois
- Indiana
- Iowa
- Kansas
- Kentucky
- Louisiana



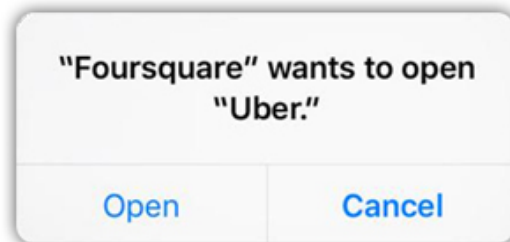
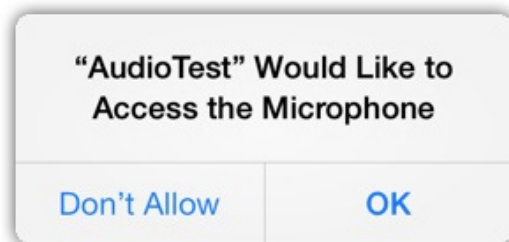
3/10 User control and freedom

- Freedom to undo, explore and change choices
- “Exits” for mistaken choices
- Exploration of data or options
- Multiple methods of doing the same thing
- Do tasks the way they like

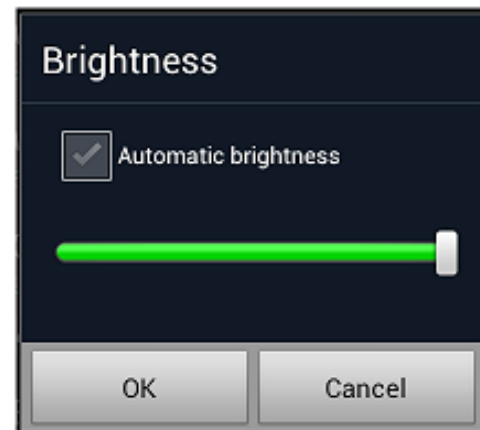
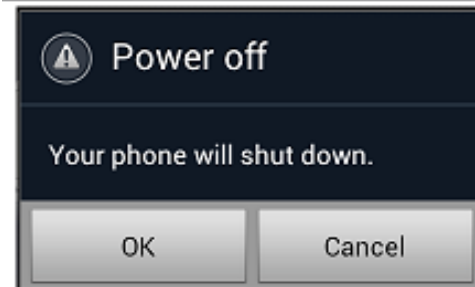
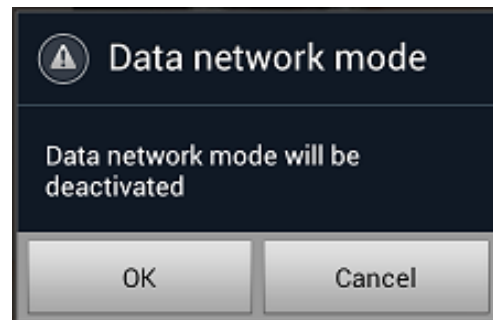
4/10 Consistency

- Consistent layout
- Standard locations and positions
 - Same language, placement etc. everywhere
 - Follow platform conventions
- Same terminology is used
- Same structure on all webpages
 - E.g. submit button at the bottom after filling out information



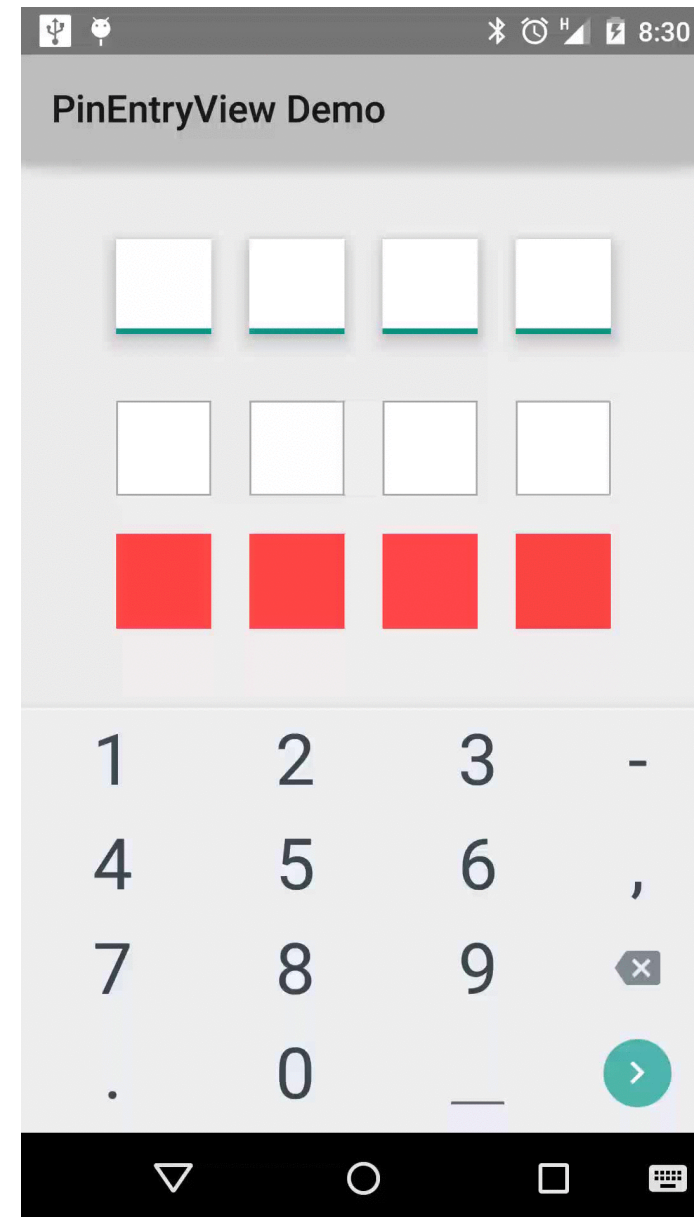
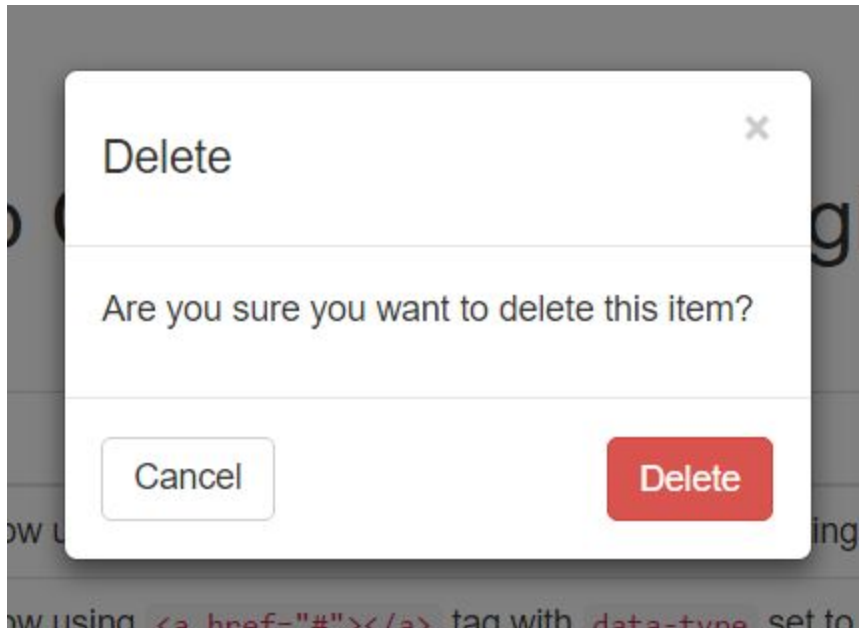


iOS uses action priority rather than static button order



5/10 Error prevention

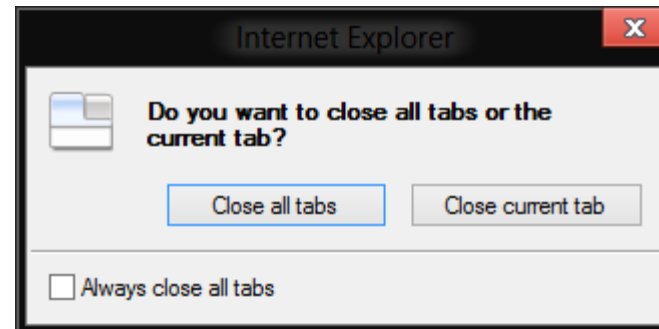
- Prevent data loss
 - e.g. Confirmation dialog before deleting an item
- Prevent bad input
 - e.g. restrict to numerals if asked to key in mobile number
- Prevent misinterpretation of the message
 - e.g. User entered text in the field when entering their mobile number. The system also displays errors about compulsory fields not completed, but the problem is with the mobile number input

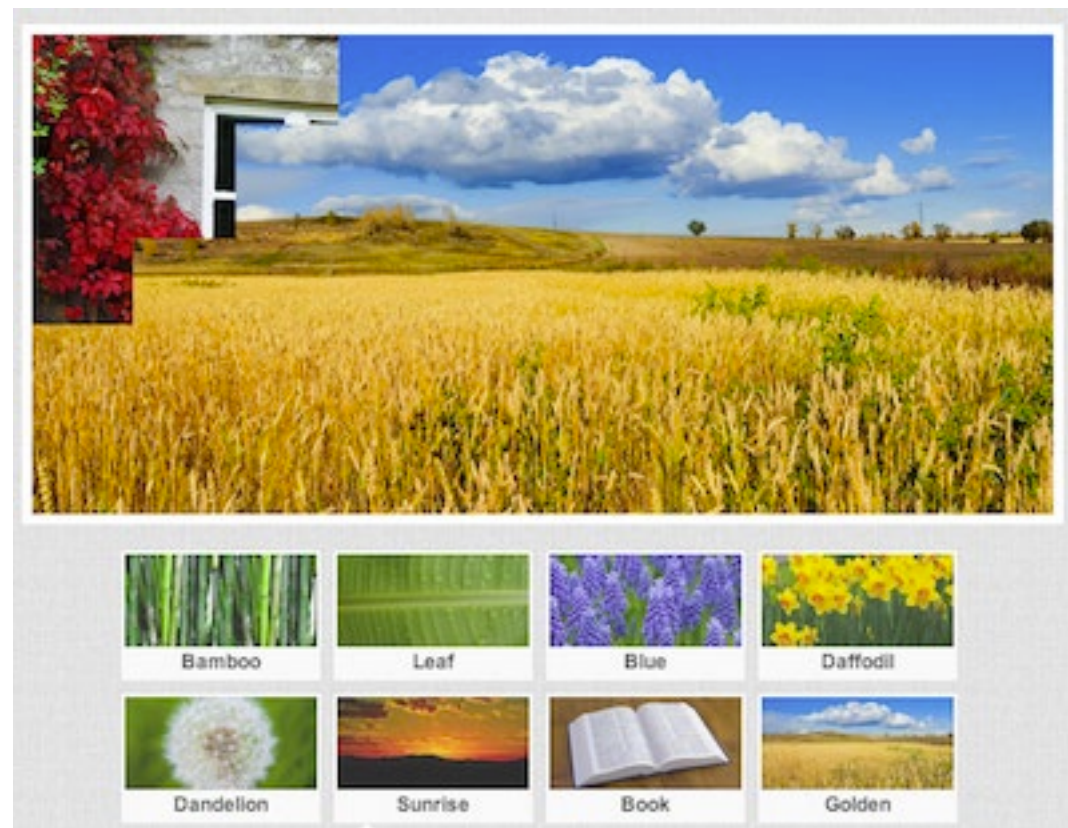
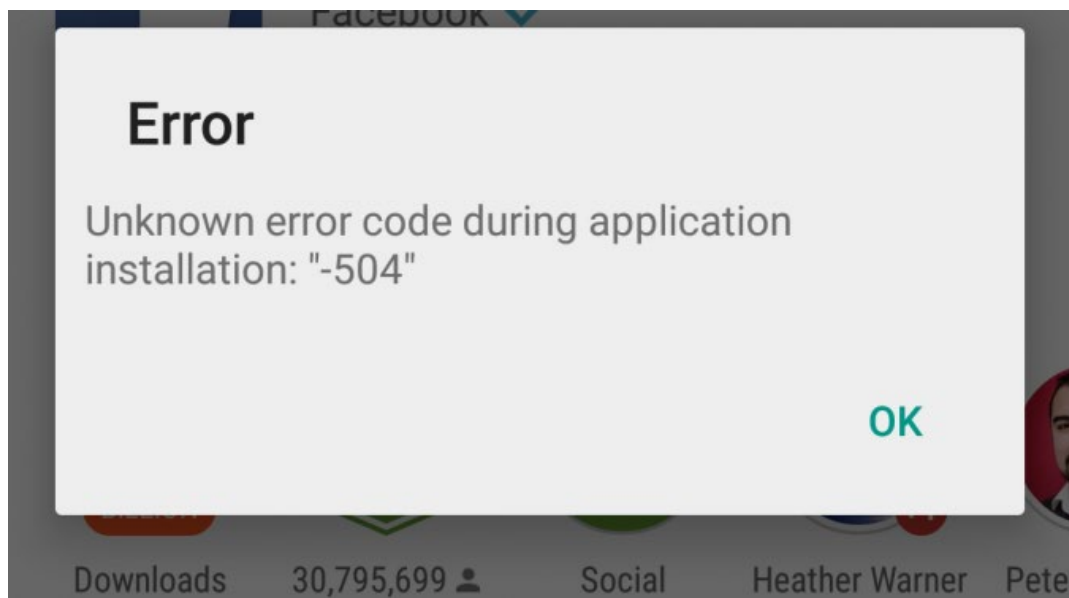


6/10 Recognition over recall

- Make objects, actions, options, & directions visible/easily retrievable
- Show users previews e.g. images, results
- Do not use codes
- Icons to click rather than type the text

```
% rm -rf *  
%
```





7/10 Flexibility & efficiency

- Flexible shortcuts
 - accelerators for experts (e.g., gestures, kb shortcuts)
 - allow users to tailor frequent actions (e.g., macros)
- Defaults with options
- Recommendations and relevant items

Google

Keyboard shortcuts

Mail Keyboard shortcuts are enabled. [Disable](#) More

COMPOSE

Inbox Starred Important Sent Mail Drafts All Mail Spam (3) Trash

Jumping

- g** then **i** : Go to Inbox
- g** then **s** : Go to Starred conversations
- g** then **t** : Go to Sent messages
- g** then **d** : Go to Drafts
- g** then **a** : Go to All mail
- g** then **c** : Go to Contacts
- g** then **k** : Go to Tasks
- g** then **l** : Go to Label

Using 1.5 GB of your 10.1 GB

Threadlist selection

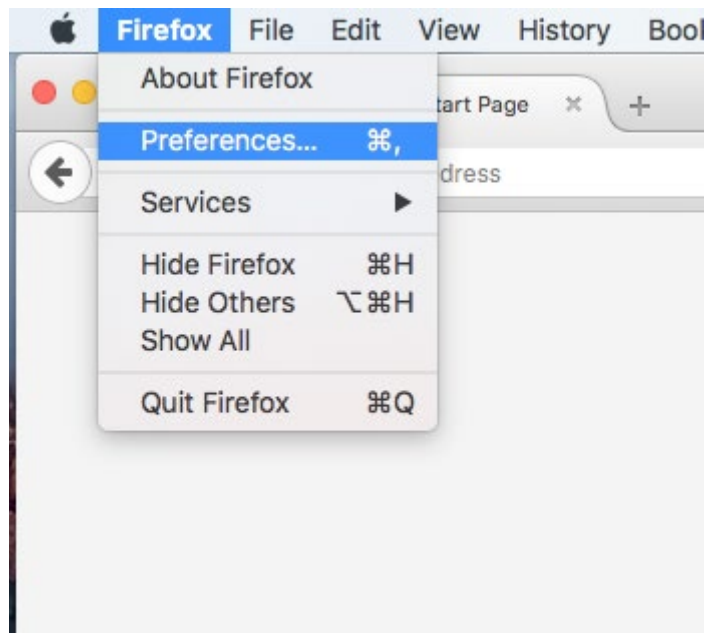
- ★** then **a** : Select all conversations

No new mail!

Actions

- ,** : Move focus to toolbar
- x** : Select conversation
- s** : Rotate superstar
- y** : Remove label
- e** : Archive
- m** : Ignore conversation
- !** : Report as spam
- #** : Move to trash
- r** : Reply
- <Shift> + r** : Reply in a new window

©2012 Google - [Terms of Service](#) - [Privacy Policy](#) - [Program Policies](#)



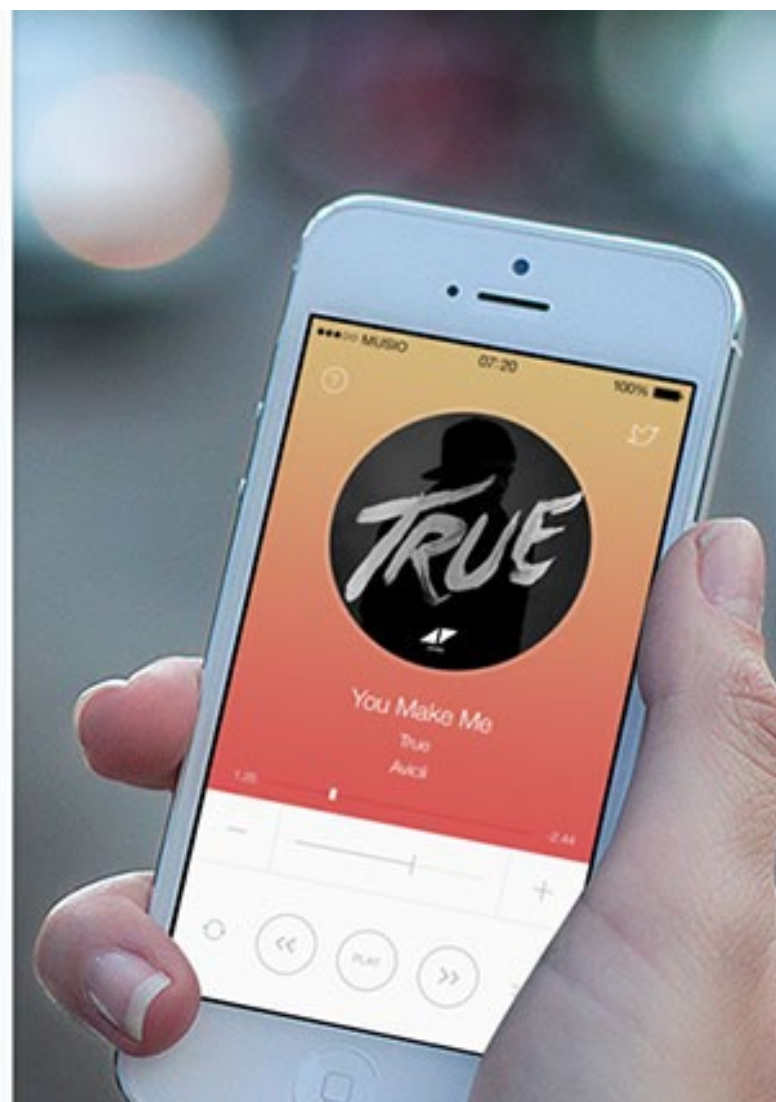
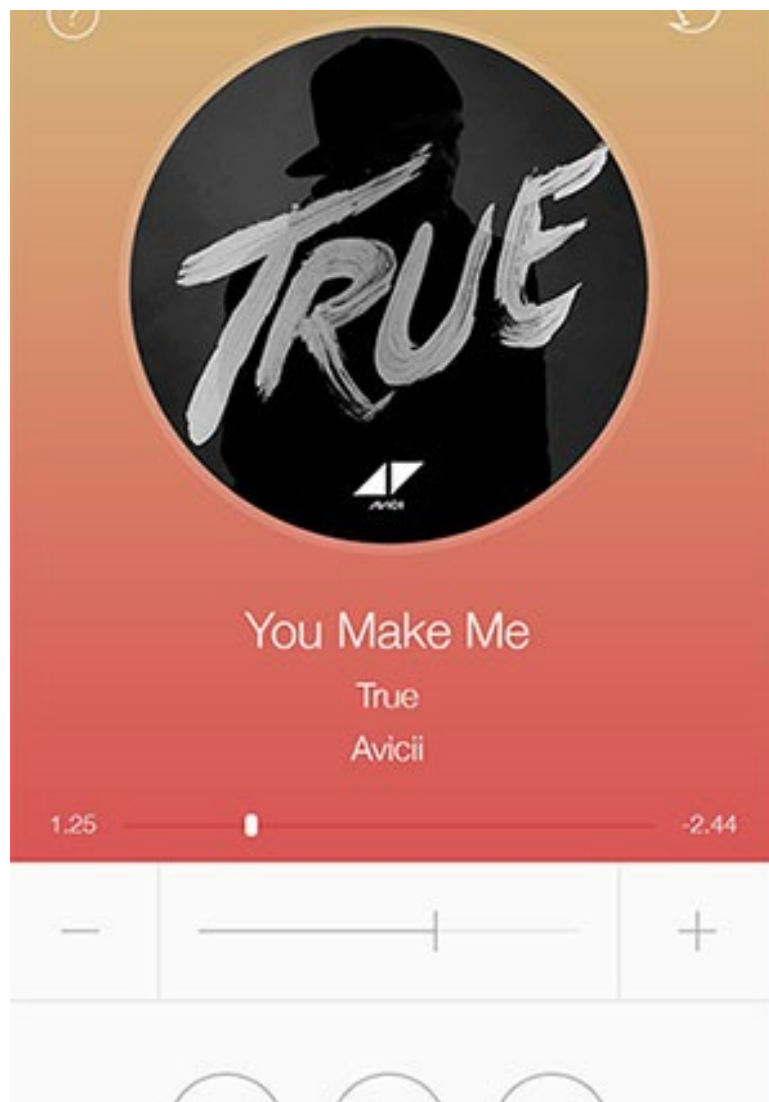
amazon.com

More Items to Consider

You viewed	Customers who viewed this also viewed
<div><p>The Appeal John Grisham Paperback \$14.00 \$11.20</p></div>	<div><p>The Innocent Man John Grisham Mass Market Paperback \$7.99</p></div> <div><p>The Associate: A Novel John Grisham Mass Market Paperback \$9.99</p></div> <div><p>Ford County: Stories John Grisham Paperback \$15.00 \$8.19</p></div>

8/10 Aesthetic & minimalist design

- Clean and uncluttered interface
- Eliminate redundant information, only leave what has to be there
- No irrelevant information in dialogs





Welcome to
Mail App

Enter your name



Password



Login

[Lost your password?](#)

Note: I live inside this website Monday to Friday 9am-6pm, to give you the very best service and make your experience a happy one! - I am Ling, accept no substitutes



LINGsCARS.com

UK CONTRACT HIRE CARS FROM LING VALENTINE




I prefer email to phone ☺

sales@LINGsCARS.com

T 0191 460 9444

F 0870 486 1130

World HQ, Vance Bus Pk

Gateshead, NE11 9NE

CAR and VAN PRICES - have a look...

INFORMATION - have a think

READ MY LETTERS

ACTION - have a go!

FULL

BUSINESS

CAR PAGE

PRINT LIST

FULL

PRIVATE

CAR PAGE

PRINT LIST

VANS AND PICKUPS

GET P/X PRICE

Step by step guide

HELLO! I am Ling

Ling's tips & hints!

1153 letters! Big page!

175 PRESS Items!

GREEN car info

Get unlisted car QUOTE!

ORDER a car now!

commended as

BEST

non-franchised WEBSITE 2007

EXTRA CHEAP

Chinese Car Websites!

CREDIT CHECK? DIY, free, HERE

my BLOG!

RSS/XML feed-Wah!

commended as

BEST

non-franchised 2008 ENVIRONMENTALIST

FREE CASH

blatant bribery from LING! CLICK!

DRAGONS' DEN

SUCCESS

READ!

NORTH-EAST ENTREPRENEUR OF THE YEAR

Winner of the Woman in Retail Award: LING VALENTINE

women into the network

WEAR & TEAR GUIDE HERE

Play with my customers!

...interactive map toy

Ling-oogle Maps

My Top 20 cars!

BEST-SELLING CARS!

VIZ

OFFICIAL Business Ambassador!



Office hours 1to1 Ling live chat:  [Live Chat is Closed](#)

CHEAP car insurance:  [Click HERE](#)

MY BUS HERE 

WIN! Ling's RACING tips!

Current Winnings: £-74.35

Less fuel = Less CO2

NEW CARS SAVE FUEL!

weekly fuel costs displayed for EVERY SINGLE CAR

CLICK HERE

NEW Ford Fusion

1.4 TDCi (67bhp) Style Hatchback 5dr 1399cc

MORE INFO **£169.99/month** (plus VAT)

World Famous CHOP-GEAR TV

CHOP Gear

CAR-U-LIKE-ATOR

PetroChina 中国石化

CAR-U-LIKE-ATOR cars by pot-luck!

2008 LINGsCARS.com RALLYCROSS OPEN CHAMPIONSHIP

PERSONAL CAR PRICE LIST

Automatic email updates

KEEP YOUR EYE ON A CAR

click here!

BE THE 1st TO KNOW knowledge is power - BACON & LING

CALLING ALL VISITORS!

I am human being, not robot!

Office hours: Mon-Fri 9 to 6

...after 6pm I making bean curd!

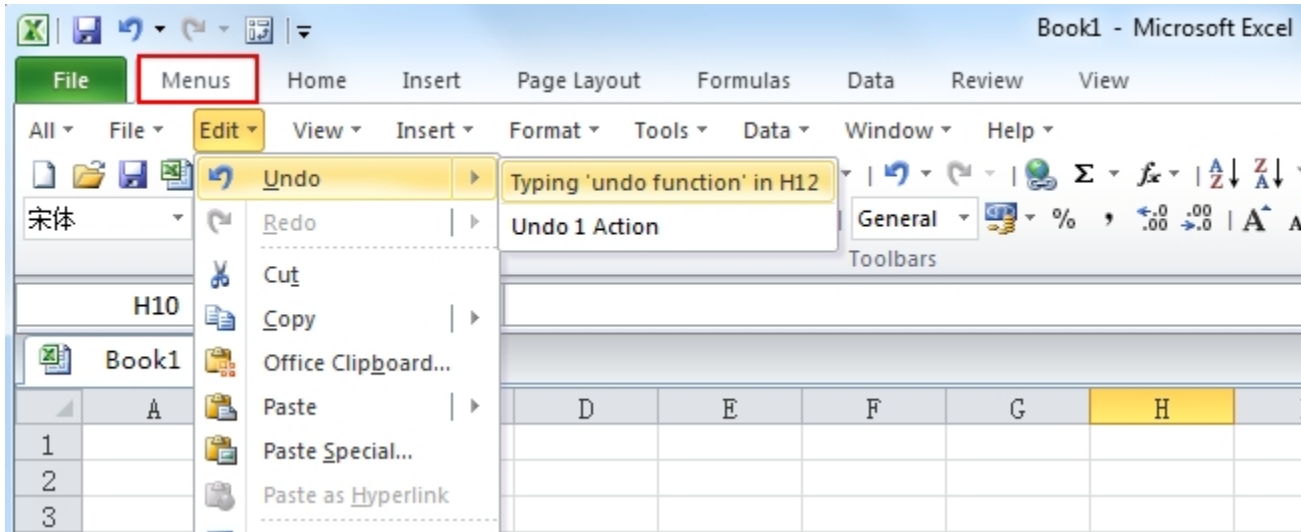
...enjoy my website...

Ling Tube

Broadcast Herself™

9/10 Recognize, diagnose, & recover from errors

- Clearly state the problem, provide a solution and how to address the problem
 - E.g. Created a password with 8 characters, but system requires at least 1 special character and 1 numeral. Error message should state the requirements, so the user can fix the error
 - E.g. Did not fill all the fields while registering for an account. Error message should highlight the fields which have not been filled



← Application ⋮

Personal info

First name

First name is required

Connors

01/01/90 Social Security

MM/DD/YYYY ### - ## - ####

Residence address

2000 Main Street, Apartment J

San Francisco CA

94112 United States

10/10 Help

- Provide more information and clarify, show steps and what to do next
 - easy to search
 - focused on the user's task
 - list concrete steps to carry out
- E.g. press (?) beside a field, popup will tell you what the field is about, and what you should enter
- E.g. Help text on social networking websites for users to understand the consequences of certain things – upload photos, change privacy settings, who can see your posts etc.

Phone Details

IMEI Code: ?

PAC Code: ?

Phone Details

IMEI Code: ?

Explanation of IMEI Code

The International Mobile Equipment Identity (IMEI) number is a unique 15-digit code used to identify an individual GSM mobile telephone. The number can be found on most mobiles by typing in *#06#. If this combination doesn't work on your mobile phone, please call our support centre on +44 (0) 1252 xxxx xxx.

[Back to IMEI input field.](#)

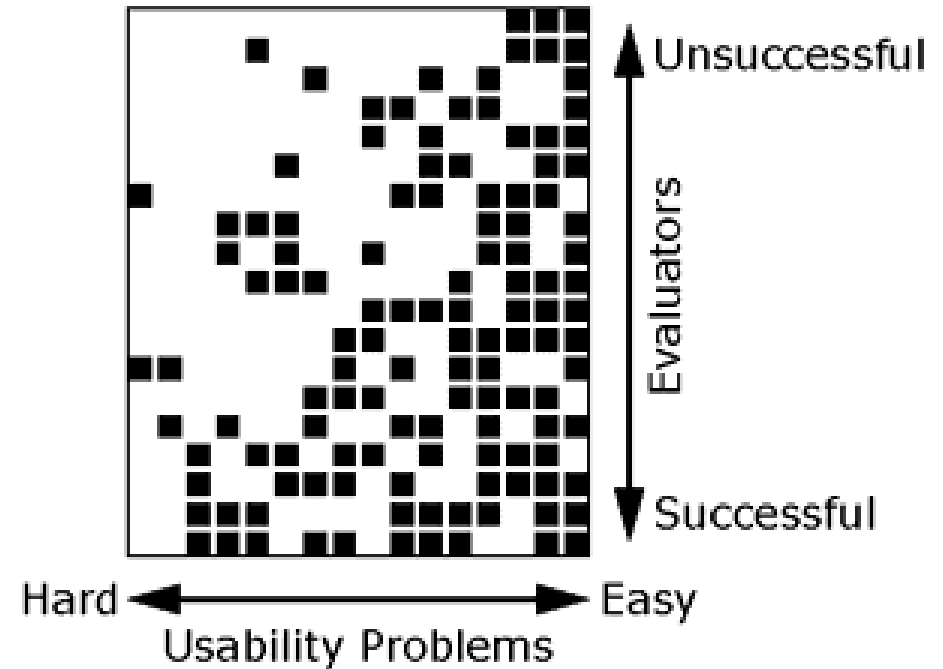
PAC Code: ?

Evaluators' Process

- Step through design several times
 - Examine details, flow, and architecture
 - Consult list of usability principles
 - ..and anything else that comes to mind
- Evaluate against heuristics
 - Nielsen's "heuristics"
 - Category-specific heuristics
 - e.g., design goals, competitive analysis, existing designs
- Use violations to redesign/fix problems

Multiple Evaluators

- No evaluator finds everything
- Some find more than others
- Single evaluator achieves poor results
 - Only finds 35% of usability problems
- 5 evaluators find ~ 75% of problems



Phases of Heuristic Evaluation

1. Pre-evaluation training
 - Give evaluators needed domain knowledge and information on the scenario
2. Evaluation
 - Individuals evaluate and then aggregate results
3. Severity rating
 - Determine how severe each problem is (priority). Can do first individually and then as a group
4. Debriefing
 - Review with design team

How to perform heuristic evaluation

- At least two passes for each evaluator
 - first to get feel for flow and scope of system
 - second to focus on specific elements
- If system is walk-up-and-use or evaluators are domain experts, no assistance needed
 - otherwise might supply evaluators with scenarios
- Each evaluator produces list of problems
 - explain why with reference to heuristic or other information
 - be specific & list each problem separately

How to perform heuristic evaluation

- Why separate listings for each violation?
 - risk of repeating problematic aspect
 - may not be possible to fix all problems
- Where problems may be found
 - single location in UI
 - two or more locations that need to be compared
 - problem with overall structure of UI
 - something that is missing
 - common problem with paper prototypes
 - note: sometimes features are implied by design docs and just haven't been "implemented" – relax on those

Severity Rating

- Independently estimate after review
- Allocate resources to fix problems
- Estimate need for more usability efforts
- Severity combines:
 - Frequency: how common?
 - Impact: how hard to overcome?
 - Persistence: how often to overcome?

Severity Ratings

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. [H1-4 Consistency] [Severity 3][Fix 0]

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 potential improvements to address major usability problems
- Development team rates effort to fix
- Brainstorm solutions

Evaluating prototypes

- Heuristic evaluation works on:
 - Sketches
 - Paper prototype
 - Buggy implementations
- “Missing-element” problems are harder to find on sketches
 - Because you are not actually using the interface
 - Tip: look harder for them

Writing Good Heuristic Evaluations

- Heuristic evaluations must communicate well to developers and managers
- Include positive comments and criticisms
 - “Good: toolbar icons are simple with good contrast and few colors”
- Be tactful (gentle)
 - Not: “the menu organization is a complete mess”
 - Better: “menus are not organized by function”
- Be specific
 - Not: “text is unreadable”
 - Better: “text is too small and has poor contrast”

Suggest Report Format

- Include:
 - Problem
 - Heuristic
 - Description
 - Severity
 - Recommendation (if any)
 - Screenshot (if helpful)

12. Severe: **User may close window without saving data** (error prevention)

If the user has made changes without saving, and then closes the window using the Close button, rather than File >> Exit, no confirmation dialog appears.

Recommendation: show a confirmation dialog or save automatically



© Microsoft. All rights reserved.

Example



Cost-effectiveness of Heuristic Evaluations

- In one case: benefit-cost ratio of 48
 - estimated benefit \$500,000; cost \$10,500
 - value of each problem ~\$15K
- How to calculate this value?
 - in-house -> productivity
 - open market -> sales
- Severe problems found more often

Heuristics vs. User Testing

- Heuristic evaluation often faster
 - 1-2 hours for each evaluator
- Heuristic evaluation results come pre-interpreted
- User testing is more accurate (by definition)
 - Takes into account actual users and tasks
 - Heuristic evaluation may miss problems & find “false positives”
- Valuable to alternate methods
 - Find different problems
 - Don't waste participants

Cognitive Walkthrough

- Assess learnability and usability through simulation of the way users explore and become familiar with interactive system
- “cognitive walkthrough” = expert inspection focused on learnability
- Input
 - Prototype
 - Task
 - Sequence of tasks to do in the prototype
 - User analysis
- Like code walkthrough in software engineering

Cognitive Walkthrough Process

- Construct carefully designed tasks from system spec or screen mock-up
- Walk through (cognitive & operational) activities required to go from one screen to another
- Review actions needed for task, attempt to predict how users would behave and what problems they'll encounter

Cognitive Walkthrough Requirements

- Description of users and their backgrounds
- Description of task user is to perform
- Complete list of the actions required to complete task
- Prototype or description of system

Cognitive Walkthrough Assumptions

- User has a rough plan (not always true)
- User explores system, looking for actions to contribute to performance
- User selects action that seems best for reaching desired goal
- User interprets response and assesses whether progress has been made toward completing task

Cognitive Walkthrough Methodology

- Step through action sequence
 - Action 1
 - Response A, B, ..
 - Action 2
 - Response A
 - ...
- For each one, ask **four questions** and try to construct a believability story

Cognitive Walkthrough 4 Questions

1. Will users be trying to produce whatever effect the given action has?
2. Will users be able to notice that the correct action is available?
3. Once found, will they know it's the right action for the desired effect?
4. Will users understand feedback after the action?

Question 1

- Will user be trying to produce effect?
- Typical supporting evidence
 - It is part of their original task
 - They have experience using the system
 - The system tells them to do it
- No evidence?
 - Construct a failure scenario
 - Explain, back up opinion

Question 2

- Will user notice the action is available?
- Typical supporting evidence
 - Experience
 - Visible device, such as a button
 - Perceivable representation of an action such as a menu item

Question 3

- Will user know it's the right action for the desired effect?
- Typical supporting evidence
 - Experience
 - Interface provides a visual item (such as prompt) to connect action to result effect
 - All other actions look wrong

Question 4

- Will user understand the feedback?
- Typical supporting evidence:
 - Experience
 - Recognize a connection between a system response and what user was trying to do

One example

- User tasks
 - Withdraw \$2,000 from ATM machine
 - Check the balance
 - Deposit a cheque
- Task #1: Withdraw \$2,000 from ATM machine
 - Step 1: Insert ATM card
 - Will user notice the action is available?
 - Will user notice the action is available?
 - Will user know it's the right action for the desired effect?
 - Will user understand the feedback?



Withdraw \$2,000 from ATM machine

- Step 2: Enter PIN number
 - Will user notice the action is available?
 - Will user notice the action is available?
 - Will user know it's the right action for the desired effect?
 - Will user understand the feedback?
- Step 3: Tap 2,000 on the screen
 - Will user notice the action is available?
 - Will user notice the action is available?
 - Will user know it's the right action for the desired effect?
 - Will user understand the feedback?

Empirical Evaluation

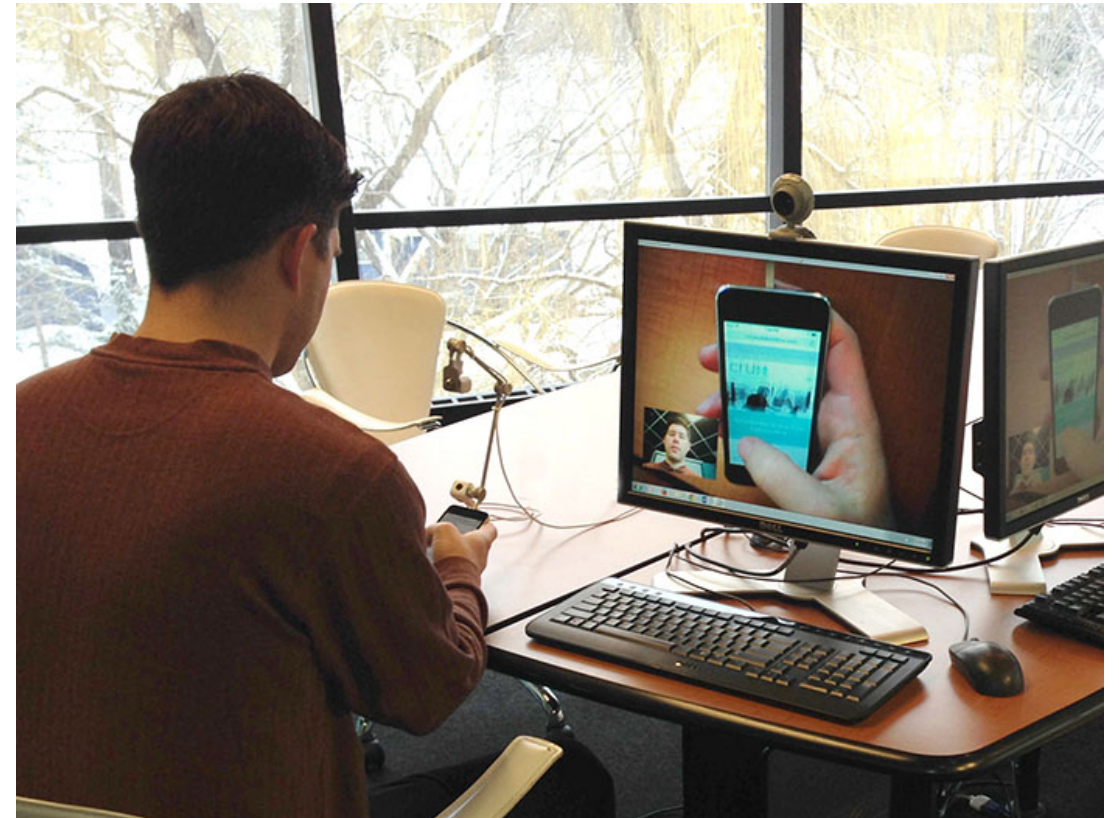
Empirical Evaluation

- Verifiable or provable by means of observation or experiment (based on empirical evidence)
- Qualitative or Quantitative
- Conducted with real users
- Earlier in the design process: testing paper prototypes in the lab
- Later in the design process: Field studies, out in the wild, usability testing

Empirical Evaluation

Lab studies (quantitative results)

- Typically in a closed, lab setting
- Manipulate independent variables to see effect on dependent variables
- Compare experimental group with the control group
- E.g. new UI vs. existing UI
- Pros: Replicable
- Cons: Expensive, requires real users and lab



Empirical Evaluation

Field studies (qualitative results)

- Observation occurs in “real life” setting
- Natural environment
- Watch process over time
- Pros:
 - “Ecologically valid”
 - Cheap, quick, less training required
- Cons:
 - Not reproducible; user-specific results
 - Not quantitative (how much better is it...?)



User Tasks

- Give user some tasks to complete
 - Tasks related to objectives/goals
 - Objective may have one or more tasks
- Tasks have a description
 - Task user needs to complete
 - Input and expected output
 - Normally included on an information sheet
- Decide on measures and questions
 - Measures to be made during the task: what to measure, how to measure?
 - Time taken to complete the task
 - Number of errors made
 - Number of users completing the task successfully
 - Number of time users had to see information sheet to complete the task

Think Aloud Protocol

- Ask users to think aloud while completing the tasks
 - Speak loud the steps to complete the tasks
 - Speak up any issues they are facing
 - Speak about confusion they are encountering
- Evaluator/observer role:
 - Conduct observation of the user
 - Take notes
 - E.g. of problems faced, confusion and types of errors made
 - Record video or audio
 - Analyze data later

Useful to know what users are thinking, while the task is being completed

Think Aloud Protocol

- May need to facilitate user responsiveness
- Possible prompts:
 - “Tell me what you are thinking.”
 - “Tell me what you are trying to do.”
 - “Are you looking for something? What?”
 - “What did you expect to happen just now?”
 - “What do you mean by that?”

Do not help users to do the tasks or give solutions. Want to understand the problems and issues users are facing.



...and, uh, I need to make sure
that the robots are working.

<https://www.youtube.com/watch?v=-h8hUtwkMCE>

Another Think Aloud Example



Summary

- GUI Patterns and Widgets
- Evaluation
 - Heuristic Evaluation
 - Cognitive Walkthrough
 - Empirical Evaluation