

Lecture #01

# CS 235: User Interface Design

## January 22 Class Meeting

---

Department of Computer Science  
San Jose State University

Spring 2015  
Instructor: Ron Mak

[www.cs.sjsu.edu/~mak](http://www.cs.sjsu.edu/~mak)



# Basic Info

---

## □ Office hours

- TuTh 7:30 - 8:30 PM
- MH 413

## □ Class website

- <http://www.cs.sjsu.edu/~mak/>
- Green sheet
- Assignments
- Lecture notes

# Goals of the Course

---

- User interface (UI) design
  - Web applications
  - Mobile applications
  - Usability testing
  - UI evaluation
- User experience (UX) design
- Data visualization
  - Information displays
  - Dashboards

# Design, Not Programming!

---

- Create mock-ups and prototypes of applications.
- Your application only has to “work” well enough to demonstrate its user interface.
- Your final project will be a significant data visualization application.



# Design Teams

---

- Projects will be done by **small design teams**.
  - Projects may be broken up into assignments.
- Form your own teams of **4 members** each.
- Choose your team members wisely!
  - Be sure you'll be able to meet and communicate with each other and work together well.
  - No moving from team to team.
- At least one member of each team should be comfortable using prototyping tools.

# Project Teams, *cont'd*

---

- Each team member will receive the same score on each team assignment and team project.
- Each team email to [ron.mak@sjsu.edu](mailto:ron.mak@sjsu.edu) by Monday, January 26:
  - Your team name
  - A list of team members and email addresses
- Subject: CS 235 Team *Team Name*
  - Example: CS 235 Team Super Coders

# Individual Responsibilities

---

You are personally responsible for participating and contributing to your team's work, and for understanding each part of the work for every assignment whether or not you worked on that part.



# Postmortem Assessment Report

---

- At the end of the semester, each student will **individually** turn in a short (one page) report:
  - A brief description of **what you learned** in the course.
  - An assessment of **your personal accomplishments** for your project team.
  - An assessment of each of  
**your project team members.**

—

# Individual Student's Overall Class Grade

---

- 50% **design assignments** (team scores)
  - Demonstrate project work to the class.
  - Justify your design choices.
- 30% **final project** (team scores)
- 20% **midterm** (individual score)
- Final letter grade based on the class curve.

# Participation is Important

---

- Can move your final grade up or down, especially in borderline cases.
- Participation in class.
- Participation in your team.
  - As reported by the postmortem assessment reports.

---

Take roll!



# User Interface (UI)

---

- What does **user interface** mean?
- What makes a good user interface?



# UI Constraints

- Desktop/laptop vs. web vs. mobile applications

	UI Constraints
Desktop/laptop	Practically none
Web	Limited by browser capabilities
Mobile	Limited by hardware capabilities and screen sizes

# First Team Task

---

- Imagine a **web application** that you would like to see developed.
- Examples:
  - class scheduling
  - contact management
  - meal preparer
  - **No games!**

—



# First Team Task, *cont'd*

---

- When you email your new team information, include:
  - A **one-sentence description** of your imagined application, and
  - **Four features** that you want your application to have.

—

# Application Example

---

- A **class scheduling application** enables a student to choose and schedule his or her classes for the semester.
- Four features?
  - Download class information from the school website.
  - Specify desired class times and breaks.
  - Display classes on a calendar.
  - Homework due date and exam date reminders.

—

# Real-World UI Example: Therac-25

- Radiation therapy machine developed by Canadian and French companies.
  - Used 1985-1987.



# Therac-25: Text-Based User Interface

PATIENT NAME:	John	BEAM TYPE:	E	ENERGY (KeV):	10
TREATMENT MODE:	FIX	ACTUAL	PRESCRIBED		
UNIT RATE/MINUTE	0.000000	0.000000			
MONITOR UNITS	200.000000	200.000000			
TIME (MIN)	0.270000	0.270000			
GANTRY ROTATION (DEG)	0.000000	0.000000		VERIFIED	
COLLIMATOR ROTATION (DEG)	359.200000	359.200000		VERIFIED	
COLLIMATOR X (CM)	14.200000	14.200000		VERIFIED	
COLLIMATOR Y (CM)	27.200000	27.200000		VERIFIED	
WEDGE NUMBER	1.000000	1.000000		VERIFIED	
ACCESSORY NUMBER	0.000000	0.000000		VERIFIED	
DATE:	2012-04-16	SYSTEM:	BEAM READY	OP.MODE:	TREAT
TIME:	11:48:58	TREAT:	TREAT PAUSE	X-RAY	173777
OPR ID:	033-tfs3p	REASON:	OPERATOR	COMMAND:	█

# Therac-25: Error Handling

---

- Whenever the system detected a problem:
    - Displayed the word **MALFUNCTION**.
    - Followed by an **error code**.
  - The user manual did not explain the error code.
    - Therefore, what do you think the operator did?
    - The operator overrode the system.
-

# Therac-25: Input Synchronization I

---

- The UI got out of sync with the operator if the operator entered commands too quickly.
- This problem was not detected during testing and initial use.
  - Why not?

# Therac-25: Input Synchronization II

---

- On a text screen, what should happen when you type over a character that is already displayed?



# Therac-25: Input Synchronization II, cont'd

PATIENT NAME:	John	BEAM TYPE:	E	ENERGY (KeV):	10
TREATMENT MODE:	FIX	ACTUAL	PRESCRIBED		
UNIT RATE/MINUTE	0.000000	0.000000			
MONITOR UNITS	200.000000	200.000000			
TIME (MIN)	0.270000	0.270000			
GANTRY ROTATION (DEG)	0.000000	0.000000		VERIFIED	
COLLIMATOR ROTATION (DEG)	359.200000	359.200000		VERIFIED	
COLLIMATOR X (CM)	14.200000	14.200000		VERIFIED	
COLLIMATOR Y (CM)	27.200000	27.200000		VERIFIED	
WEDGE NUMBER	1.000000	1.000000		VERIFIED	
ACCESSORY NUMBER	0.000000	0.000000		VERIFIED	
DATE: 2012-04-16	SYSTEM: BEAM READY	OP.MODE: TREAT		AUTO	
TIME: 11:48:58	TREAT: TREAT PAUSE	X-RAY		173777	
OPR ID: 033-tfs3p	REASON: OPERATOR	COMMAND:	[redacted]		

# Therac-25: Input Synchronization II, *cont'd*

---

- The operator enters the **wrong beam type code**.
- He or she corrects it on the screen by overtyping the incorrect code character with the correct code character.
  - The new beam type code character was displayed.
- However, the software did not internally process the new character correctly but instead it **kept the old code** in its buffer.

# Therac-25: Tragic Results

---

- Between 1985 and 1987, six patients were given **100 times** the intended dose of radiation.
  - Three of those patients died from radiation burns.
  
- What other UI features were missing that could have prevented these tragedies?

# User Experience (UX)

---

- What does **user experience** mean?
  - What makes a good user experience?
- 



# UX Example: The Apple iPhone

---

- Set the iPhone to **silent mode**.
  - Incoming phone calls will **vibrate** instead of ring.
- What happens when a preset alarm goes off?  
—

# Usability Testing

---

- How can we test whether or not our UI design or UX design is good?
- 

—



# Data Visualization

---

- Big Data + Big Data analysis
  - What is **data visualization?**
    - Present analyzed data in a useful and **insightful** way to the user.
    - **Actionable knowledge!**
-

# Reminders: By Monday, January 26

---

- Form teams.
- Email me your team information.
  - team name
  - team members and email addresses
- Include a description of your team's imagined web application.
  - 1-sentence description
  - 4 features

—

Lecture #02

# CS 235: User Interface Design

## January 27 Class Meeting

---

Department of Computer Science  
San Jose State University

Spring 2015  
Instructor: Ron Mak

[www.cs.sjsu.edu/~mak](http://www.cs.sjsu.edu/~mak)



# Reminders: As Soon as Possible

---

- Form teams.
- Email me your team information.
  - team name
  - team members and email addresses
- Include a description of your team's imagined web application.
  - 1-sentence description
  - 4 features

—

# User-Centered Design (UCD)

---

Involve the users throughout the UI design and development process.



# User-Centered Design Principles

---

- The active involvement of users.
  - An appropriate allocation of function between user and application.
  - The iteration of design solutions.
  - Multidisciplinary design teams.
- 



# User-Centered Design Activities

---

- Understand and specify the context of use.
  - Specify the user and organizational requirements.
  - Produce design solutions (prototypes).
  - Evaluate designs with users against the requirements.
- 

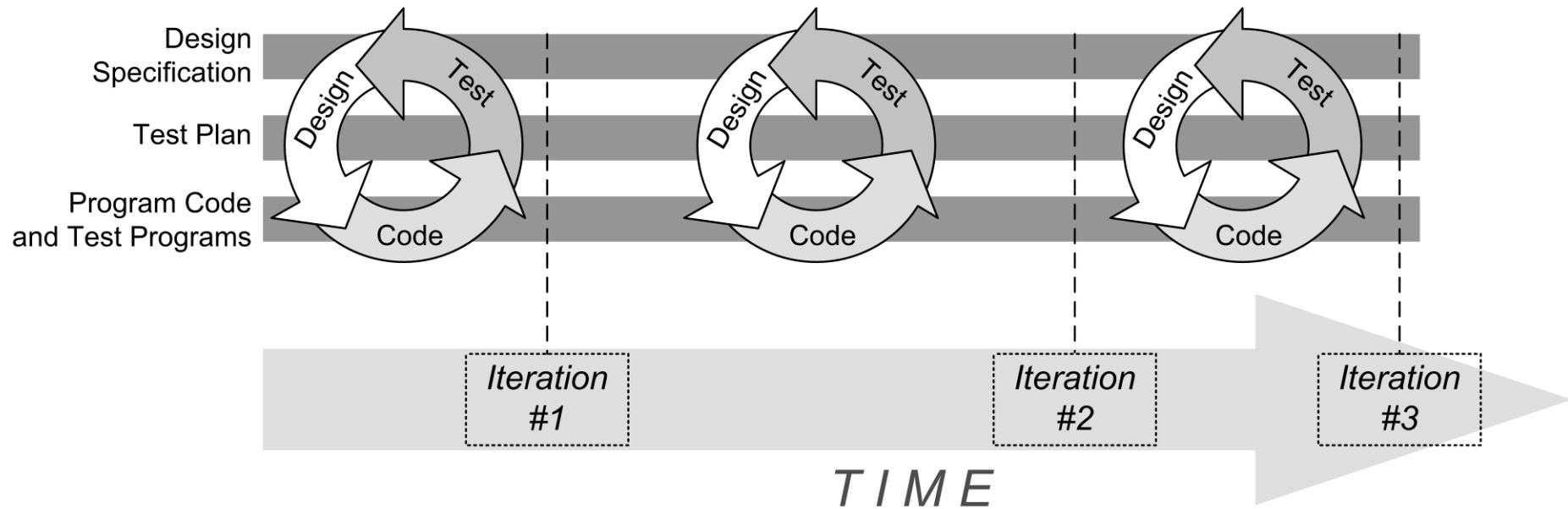


---

Take roll!

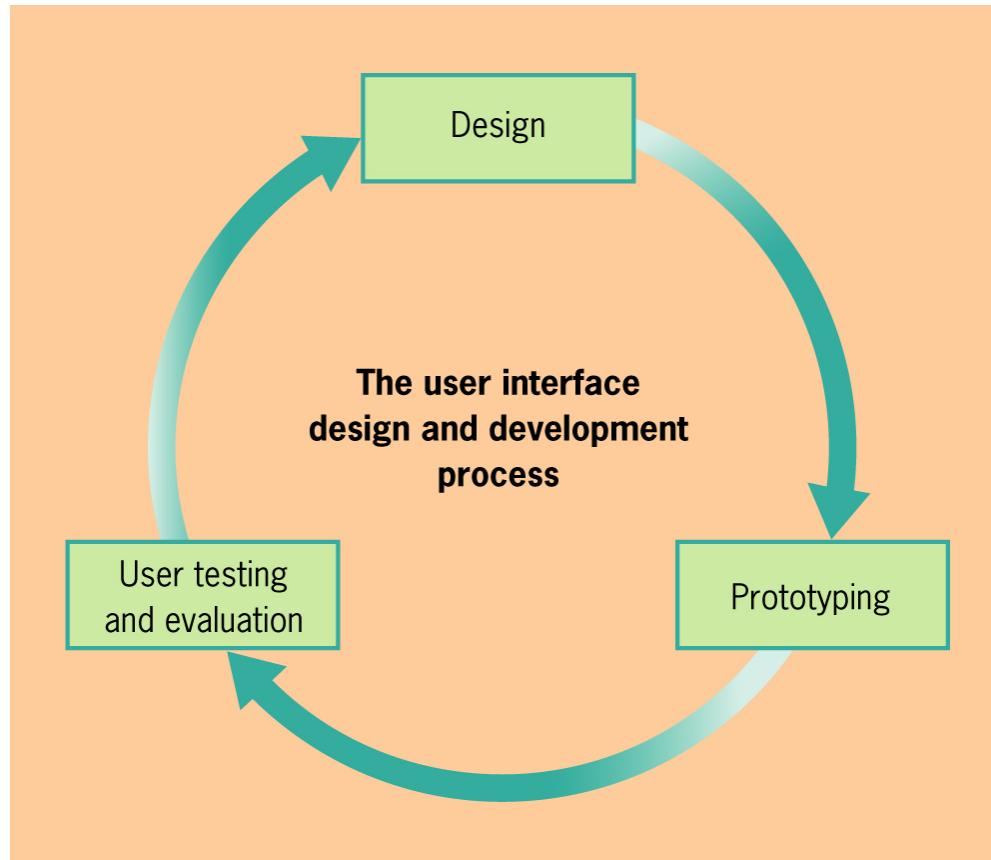


# Agile Development



- In our context, “coding” means making **incremental UI changes** to the application prototype.

# Iterative Design



# What is the Purpose of Your Application?

---

- The user accomplishes a task or a set of related tasks.
  - Text editor, drawing tool, PowerPoint, compiler
- Provide information to the user.
  - Wikipedia, Google News, ebook reader, video player
- The user interacts with data and information.
  - Email tool, games

—

# What is the Purpose of Your Application?

---

- Acquire information from the user
  - Online IRS income tax tool
- Social
  - Facebook
- Commerce
  - Amazon

—

# Who are the Users of Your Application?

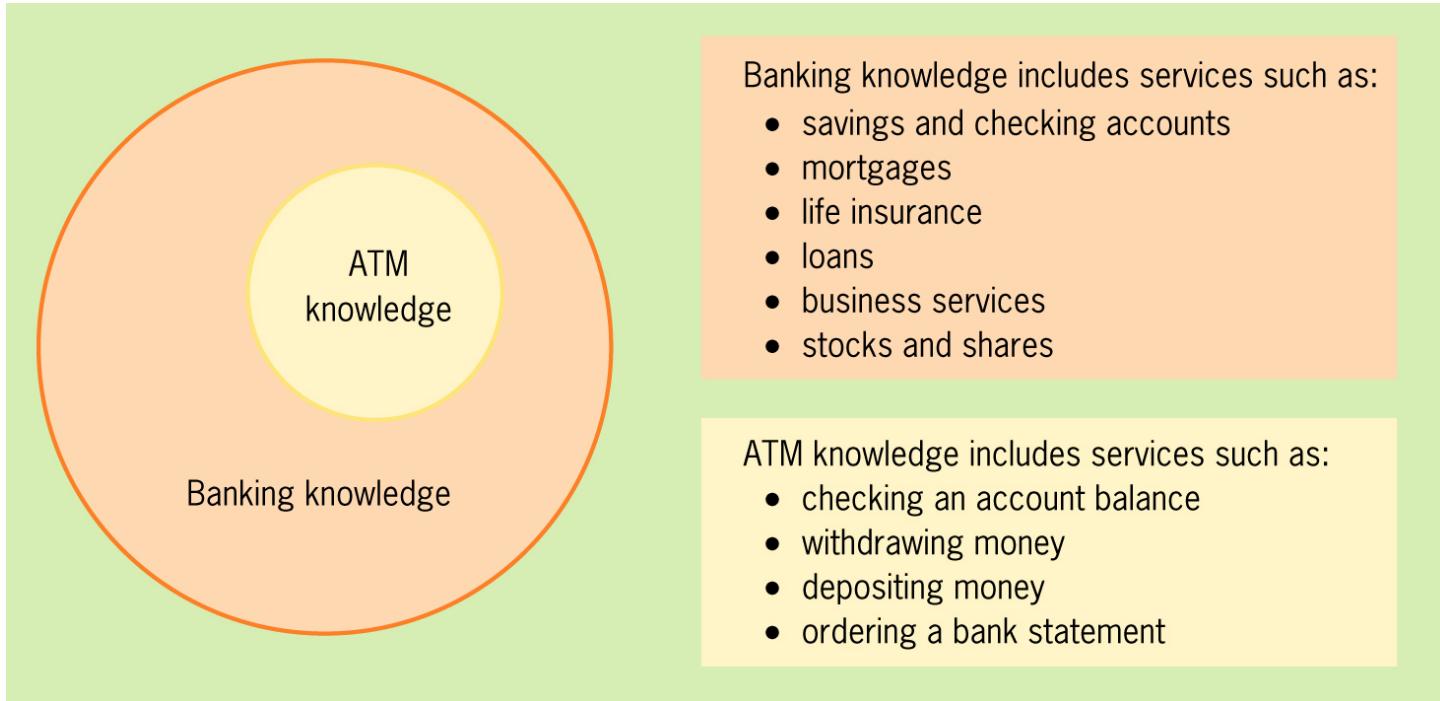
---

- Your UI design must accommodate the characteristics of the users of the application.
  - age
  - gender
  - culture
  - physical abilities and disabilities
  - educational background
  - computer experience
  - motivation
  - attitude

# Understand Your Users

## □ Domain analysis

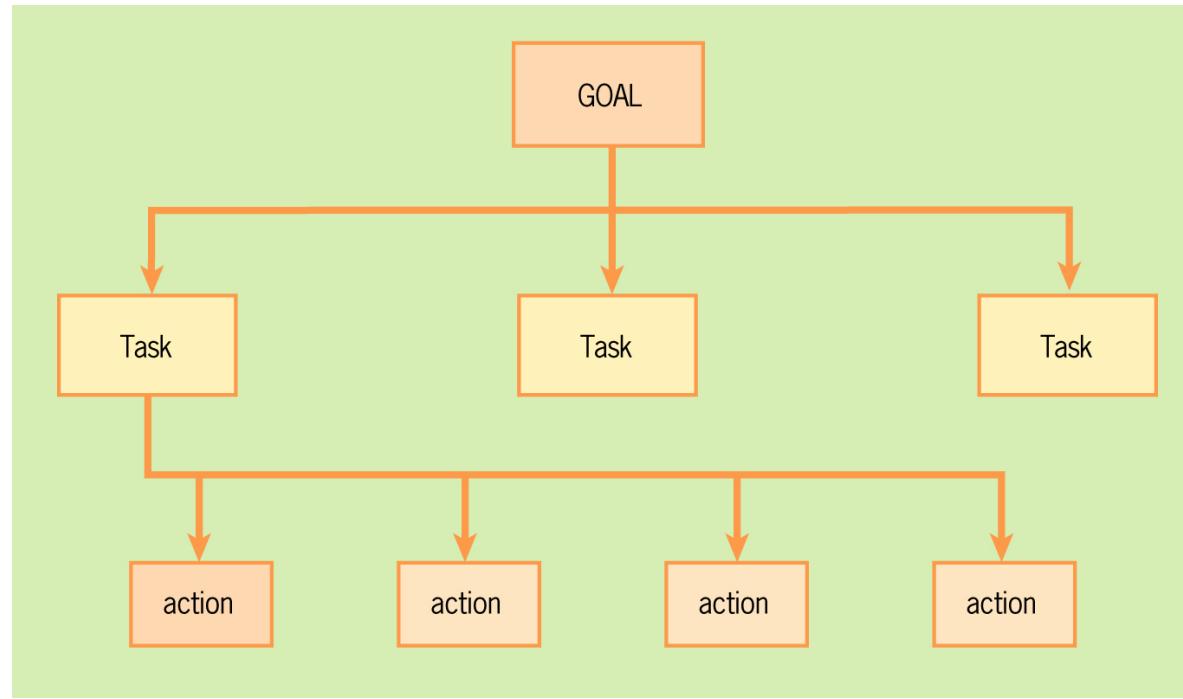
- Understand the area of expertise or specialist knowledge for which the application is developed.



# Understand Your Users, *cont'd*

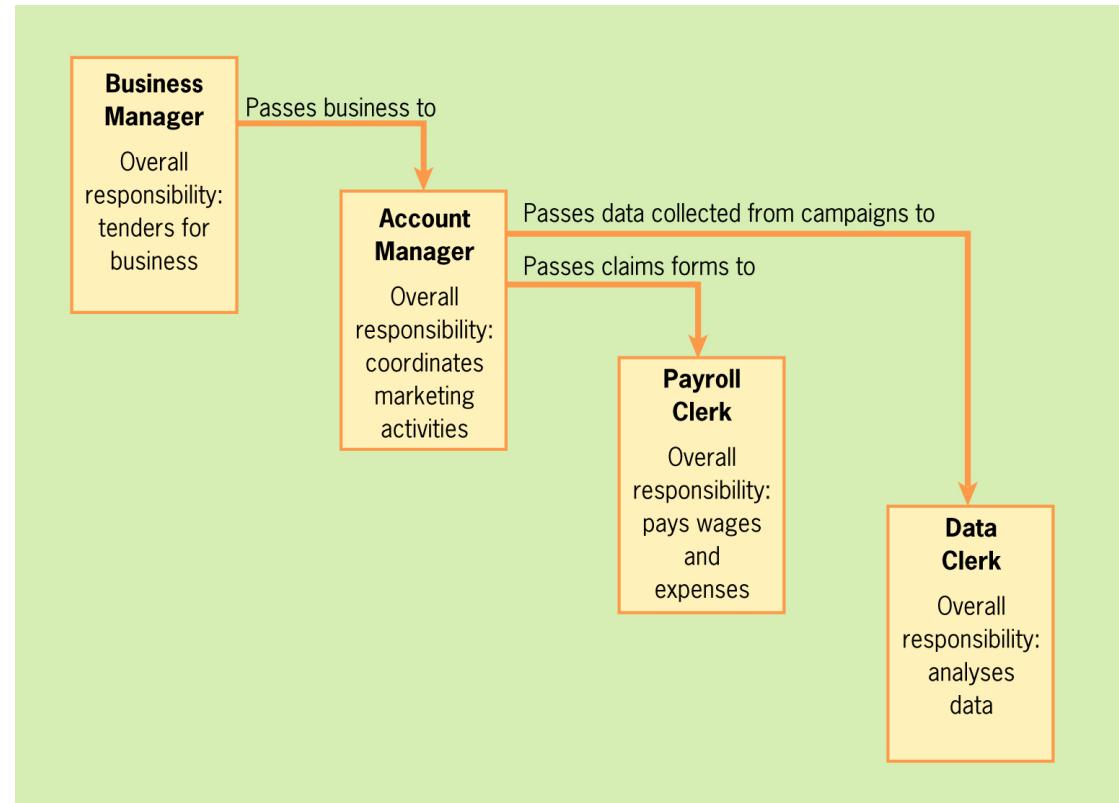
## □ Task analysis

- Understand the goals, tasks, and actions of the users.



# Understand Your Users, *cont'd*

- Workflow analysis
  - Understand how work can move from one user to another.



# Mental Models

---

- A **mental model** enables a person to
  - Negotiate unfamiliar situations.
  - Reason about a situation based on experience and previously acquired knowledge.

—



# Mental Models, *cont'd*

---

- You (as the designer) have a **mental model** of how the application is supposed to work. How well does it match the **user's mental model**?
- The user will feel that an application is **easy to use and intuitive** if the differences between the two mental models are small.
- This is a UX concern, beyond just the UI.
  - Reminder: The iPhone “silent” alarms.

# Sources of Requirements

- Client
  - End users
  - Application developers
  - Development managers
  - Technology providers
- }
- Stakeholders
- All can have **conflicting ideas** of what the application is supposed to do.
  - All of them **change their minds** about the requirements.

# Functional Requirements

- What the application **shall** be able to do or **allow users** to do.
  - “The phone **shall** use GPS to determine the wearer’s location.”
  - “Users **shall** be able to choose either Option A or Option B.”
- Describe the **interactions** between the user and the application, independent of the implementation.

Effects on UI?  
Effects on UX?

# Nonfunctional Requirements

## □ Usability, reliability, performance, supportability, etc.

Constraints the application **must meet** in order to work correctly.

- “The application **must** respond to the user within 15 seconds.”
- “The application **must** run on Windows and Linux servers.”
- “The new GUI **shall** resemble the existing GUI.”

Effects on UI?  
Effects on UX?



# Requirements Must Have ...

---

- **Completeness**
  - Are all system features described by requirements?
- **Consistency**
  - No two requirements can contradict each other.
- **Clarity**
  - Each requirement must be unambiguous.
- **Correctness**
  - No errors in the requirements.
  - Can each application function be traced to a requirement?

# Requirements Must Have, *cont'd*

---

- **Realism**
    - Can the system be implemented?
  - **Verifiability**
    - Can the system be tested?
  - **Traceability**
    - Can each requirement be traced to an application function?
-

# Requirements are Strong Statements

---

- Use **strong declarative statements** with “shall” and “must”.
  - “The phone **shall** use GPS to determine the wearer’s location.”
  - “The system **must** respond to the user within 15 seconds.”
- 



# How to Get Requirements

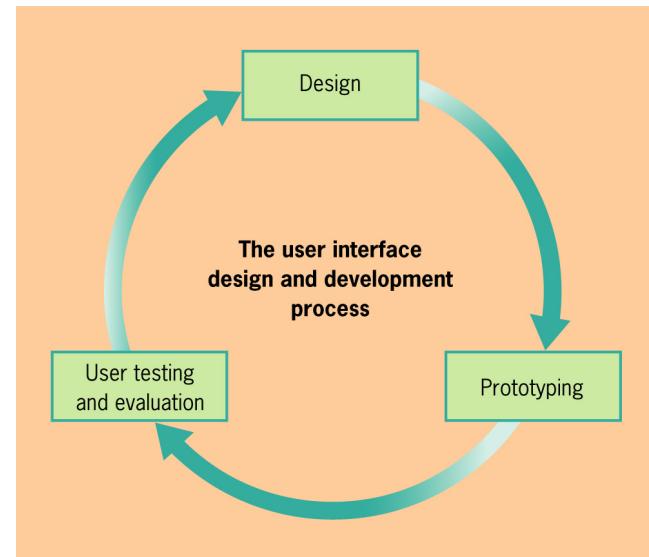
---

- Interview future users of your application.
- Observe how the users currently work.
- Stated requirements
  - The user tells you what he or she wants.
- Implied requirements
  - What do you think the user wants?

—

# How to Get Requirements, *cont'd*

- Users don't always know what they want.
- They will know more when you show them a prototype.
- They will change their minds.
- It's an iterative process!



# Use Cases

---

- A **use case** is a complete **sequence of steps** that allows the user to complete a task.
  - Describe a task that your application **must allow** the user to accomplish.
- 



# Parts of a Use Case

---

## □ Name

- The name should be in the form *verb object*.

## □ Goal

- What does this task accomplish?

## □ Sequence of steps

- For each step:
  - What is the user action?
  - What is the application's response?
- Include any **alternate sequences** in case something goes wrong.

# Example Use Case

- Name:  
Obtain foreign currency.
- Goal:  
The user obtains foreign currency from an ATM.

User action	System response
User inserts credit card into the slot.	System requests PIN.
User types in 4-digit PIN number using the keypad.	System verifies user's identity. System requests foreign currency required, to be selected using menu keys.
User presses the key corresponding to the required currency.	System displays the exchange rate. System requests the user to enter the amount of foreign currency required using the keypad. The unit of currency is also displayed, as the system only deals with banknotes.
User enters amount required using the keypad.	System returns the credit card via the slot. System dispenses the currency via the currency delivery slot. System delivers a printout of the transaction via the receipt slot.

# Good Use Cases

---

- Write your use cases in a way that makes sense to all stakeholders (client, developers, managers, ...).
  - Good use cases show that you've done your requirements analysis well and that your application will work in a real-world context.
- 



# Functional Specification

---

- Name of the application
- A short description
  - What is the purpose of your application?
    - 1 paragraph
  - How will it accomplish this purpose?
    - 1 to 3 paragraphs
- List of functional requirements
- List of nonfunctional requirements
- Use cases

# On Thursday

---

- We'll use the teams' proposed applications to:
  - Create mental models.
  - Elicit functional and nonfunctional requirements.
  - Generate use cases.
  - Write functional specifications.
- Details on Thursday!

Lecture #03

# CS 235: User Interface Design

## January 29 Class Meeting

---

Department of Computer Science  
San Jose State University

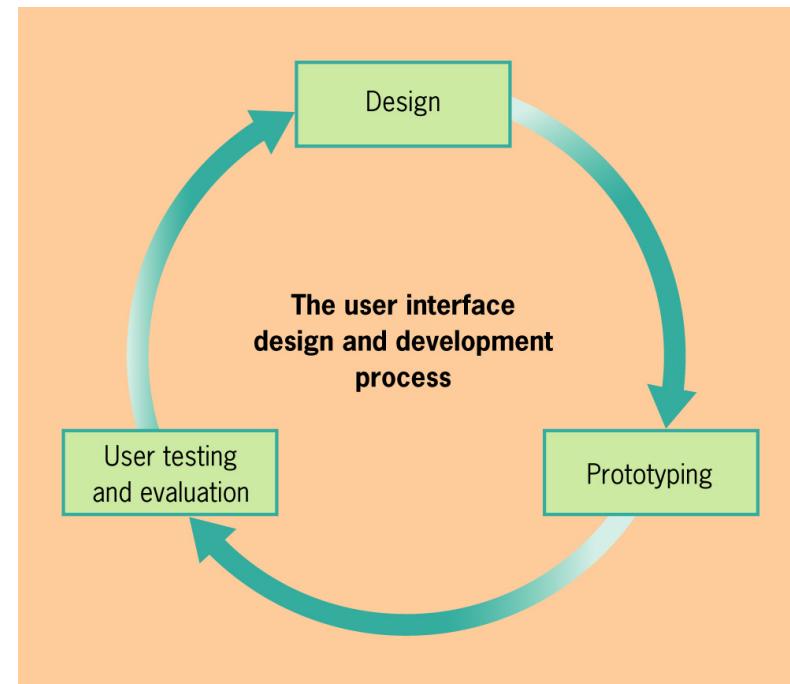
Spring 2015  
Instructor: Ron Mak

[www.cs.sjsu.edu/~mak](http://www.cs.sjsu.edu/~mak)



# Preliminaries

- Purpose of the application
- User-centered design
- Understand the users
- Iterative design



# Functional Requirements

---

- What the application **shall** be able to do or **allow users** to do.
  - “The phone **shall** use GPS to determine the wearer’s location.”
  - “Users **shall** be able to choose either Option A or Option B.”
- Describe the **interactions** between the user and the application, independent of the implementation.

# Nonfunctional Requirements

---

- **Usability, reliability, performance, supportability, etc.**
  - “The application **must** respond to the user within 15 seconds.”
  - “The application **must** run on Windows and Linux servers.”
  - “The new GUI **shall** resemble the existing GUI.”
- **Constraints** that the application must meet.

# Parts of a Use Case

---

## □ Name

- The name should be in the form *verb object*.

## □ Goal

- What does this task accomplish?

## □ Sequence of steps

- For each step:
  - What is the user action?
  - What is the application's response?
- Include any **alternate sequences** in case something goes wrong during a step.

# Functional Specification

---

- Name of the application
- A short description
  - What is the purpose of your application?
    - 1 paragraph
  - How will it accomplish this purpose?
    - 1 to 3 paragraphs
- List of functional requirements
- List of nonfunctional requirements
- Use cases

# Initial Interviews

Design Team	Client Team/App
Unknown	<b>Innovative Designers</b> Team Collaboration
Innovative Designers	<b>Team Four</b> Contact Management
Team Four	<b>Thundercats</b> Online Banking
Thundercats	<b>Uxability</b> To-Do List
Uxability	<b>XFactor</b> Online multilingual IDE
XFactor	<b>Unknown</b> Airline Scheduling

25 minutes per round



# Assignment #1

---

- Continue to interview your client team.
  - Create a **Functional Specification** (first draft)
    - Application name
    - Application description
    - List of at least 10 functional requirements
    - List of at least 4 nonfunctional requirements
    - At least 5 use cases
-

# Assignment #1, *cont'd*

---

- Each team turns in one Functional Specification.
  - Microsoft Word document or PDF
- Email as an attachment to [ron.mak@sjsu.edu](mailto:ron.mak@sjsu.edu)
  - Subject: **CS 235 Assignment #1 Team name**
  - Be sure to CC all team members
- Due **Friday, February 6 at 11:59 PM.**

Lecture #04

# CS 235: User Interface Design

## February 3 Class Meeting

---

Department of Computer Science  
San Jose State University

Spring 2015  
Instructor: Ron Mak

[www.cs.sjsu.edu/~mak](http://www.cs.sjsu.edu/~mak)



# Example Use Case Alternate Sequence

User Action	System Response
1. Select “Withdraw cash”	Prompt for amount
2. Enter amount	Check bank account balance
	Dispense cash
3. Take cash	

User Action	System Response
1. Select “Withdraw cash”	Prompt for amount
2. Enter amount	Check bank account balance
	Display “Insufficient bank balance”
2.1 Enter another amount	Check bank account balance
	Dispense cash
3. Take cash	

# Design Patterns

---

- A design pattern is
  - A description of a problem.
  - A solution that you can apply to many programming situations.
  
- Design patterns show you how to build good user interfaces.
  - Design patterns are proven design experience.
  - Design patterns describe best practices.
  - Design patterns capture structural and behavioral features of an interface.

# Design Patterns

---

- Design patterns are not code, but are **general solutions** to design problems.
  - You apply them to your specific application.
- Design patterns give programmers a very high-level **pattern language** and a short-cut **vocabulary** to discuss design issues.
  - Independent of specific implementations.
  - “We should use the canvas+palette pattern here.”
  - “The wizard pattern will simplify this code.”

—

# Design Patterns

---

- Each design pattern should have:
    - A short **name**
    - **What:** a brief description of the pattern
    - **When** to use: a brief description of the context
    - **Why:** a description of the problem that it solves
    - **How:** a prescription for a solution
-

# Design Patterns, *cont'd*

---

- The building architect **Christopher Alexander** formulated over 250 patterns for architectural design for physical buildings.
    - Co-authored *A Pattern Language: Towns, Buildings, Construction*, published in 1977.
- 



# Organization Design Patterns

---

- Design patterns for the high-level overall layout of an application's user interface.
- Information architecture
  - How to organize an information space.
- Interaction model
  - Determines how users navigate.
  - Establishes consistency.

# Organization: Feature, Search, and Browse

## □ What

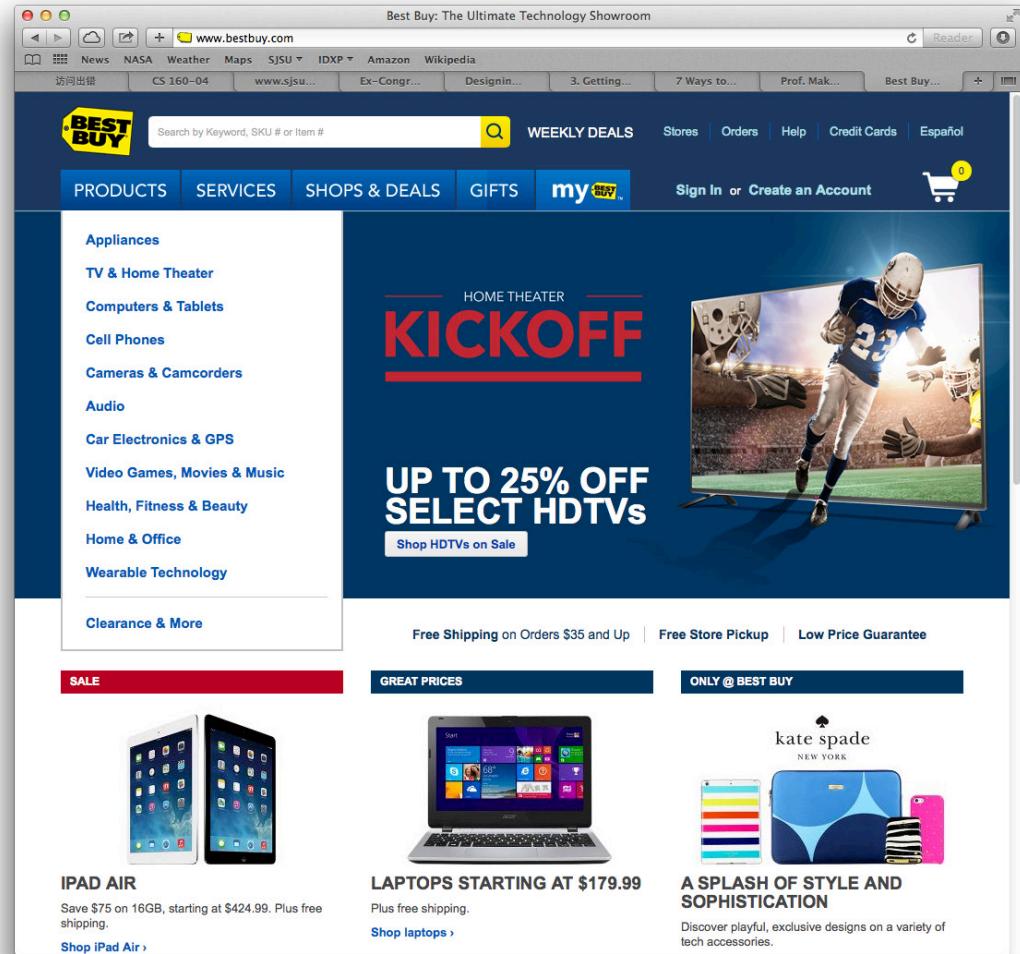
- A featured item
- A search box
- A list to browse

## □ When

- Engage the user
- Provide other options
- Enable searching

## □ Why

- Browsing and searching are common
- Hook the user with the featured item



# Organization: News Stream

## □ What

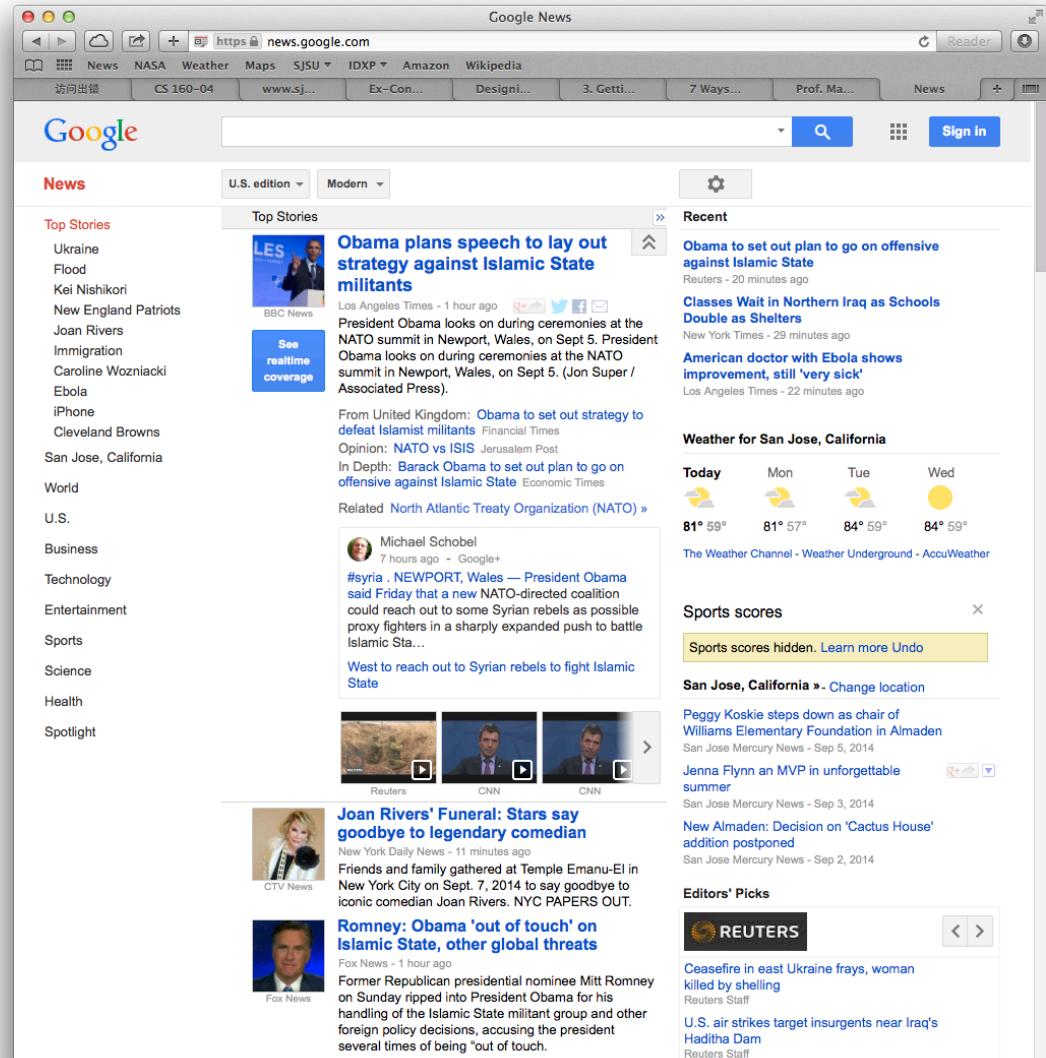
- Time-sensitive items
- Reverse chronological order
- Frequent updates

## □ When

- Communications channels
- Timely content

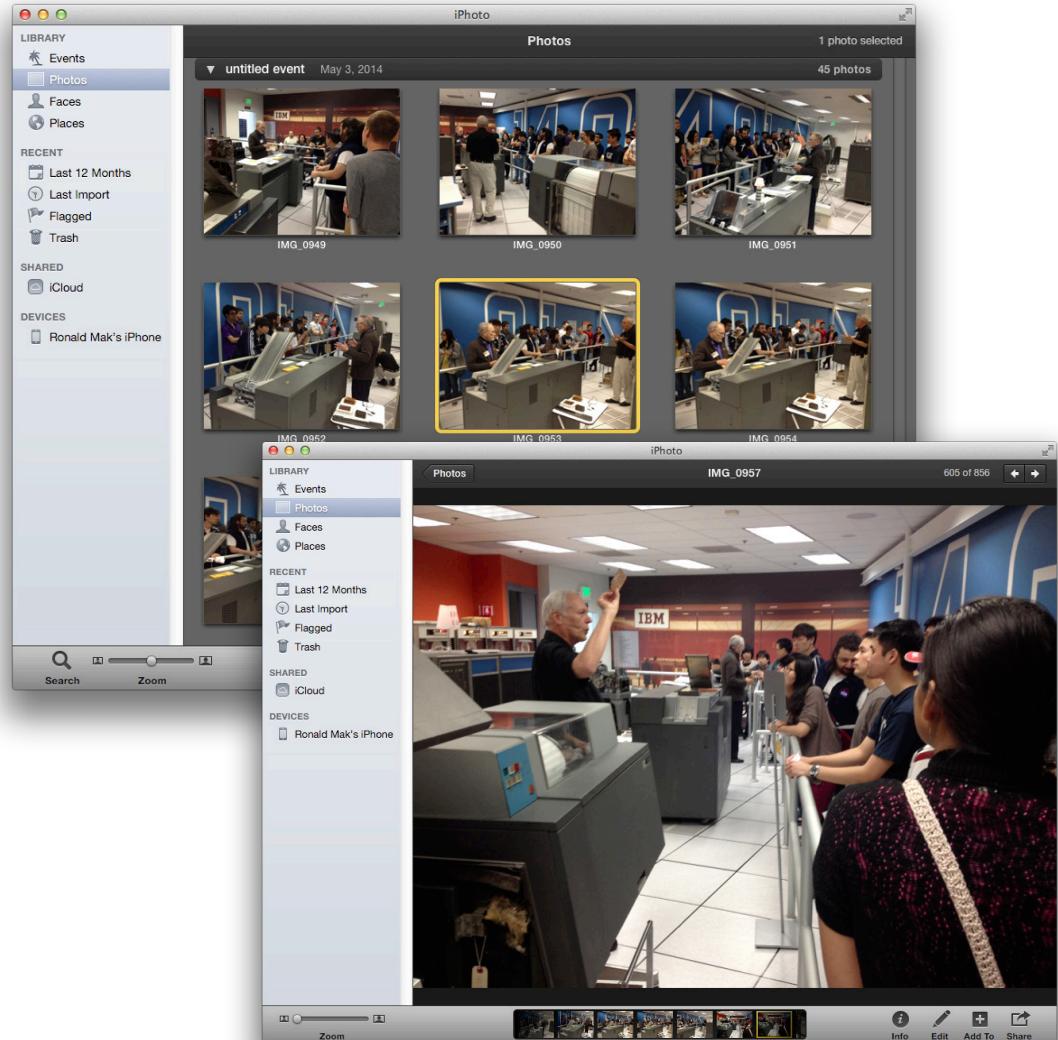
## □ Why

- Allow users to keep up with news streams
- Users can check often



# Organization: Picture Manager

- What
  - Thumbnails of pictures for browsing
  - Feature one picture
- When
  - A collection of pictures to organize
- Why
  - A natural way for users to organize and browse a picture collection



# Organization: Dashboard

## □ What

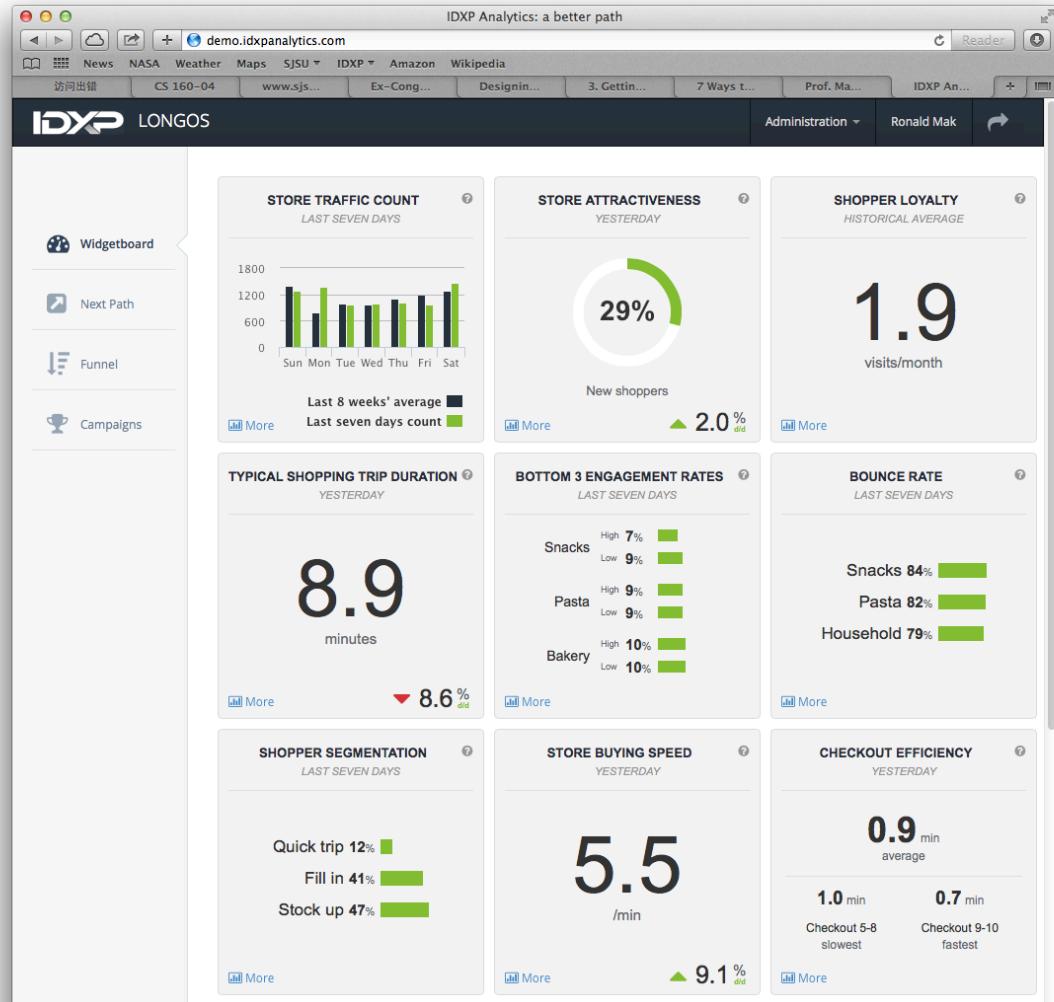
- Single information-dense page
- Information widgets frequently updated

## □ When

- An incoming flow of information

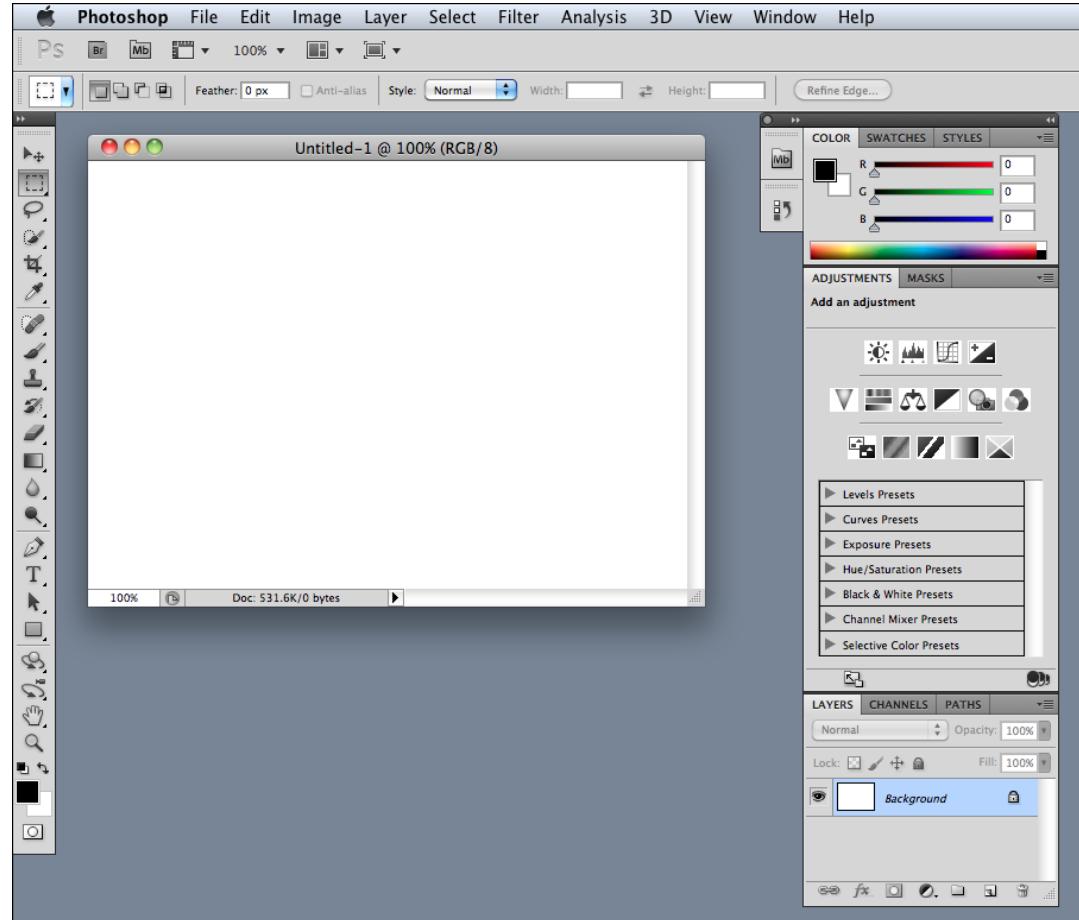
## □ Why

- Users need to monitor the information
- See what's important at a glance



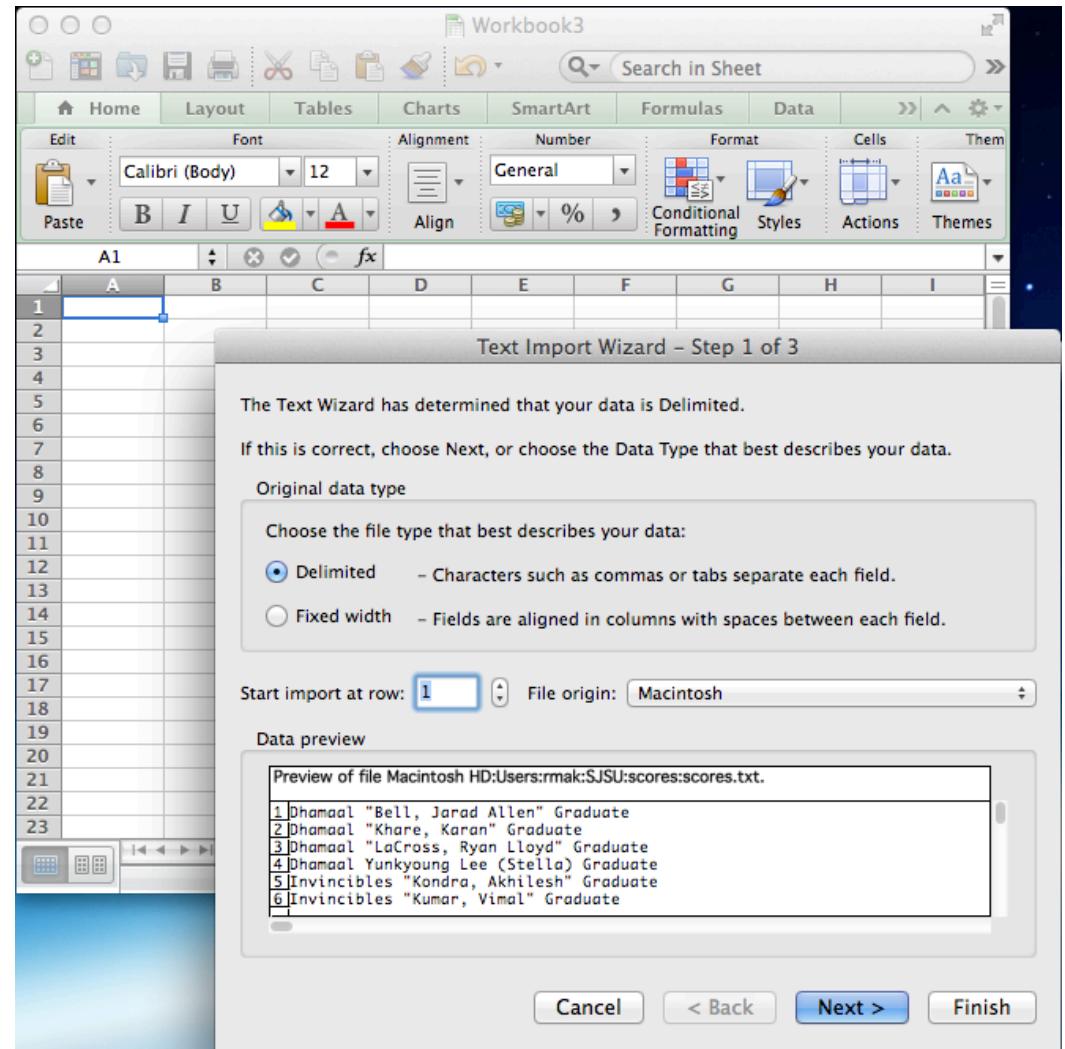
# Organization: Canvas + Palette

- What
  - Palette of drawing tools
  - Canvas for drawing
- When
  - A graphics editor
  - Create new objects
  - Arrange object in virtual space
- Why
  - Closely matches a user's mental model of how to create some artwork



# Organization: Wizard

- What
  - Steps to accomplish a specific task
- When
  - A complicated task
  - “Don’t make me think.  
Just tell me what to do next.”
- Why
  - Handholding for the beginning user
  - But can be frustrating for experienced users



# Organization: Settings Editor

## □ What

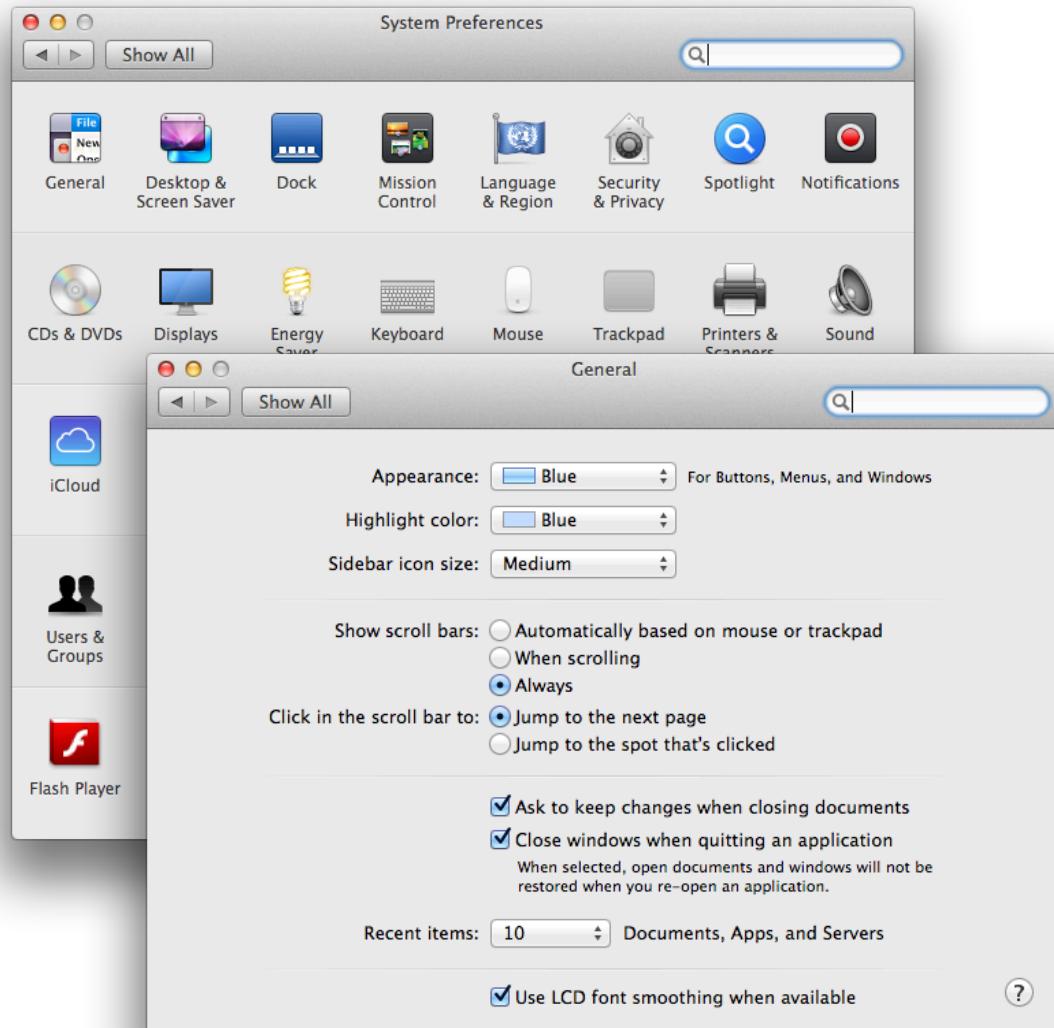
- One place to change settings, preferences, or properties
- Group content

## □ When

- Users have many choices to create a profile for an application

## □ Why

- Easy to find and use
- See all choices in one place



# Organization: Alternative Views

## □ What

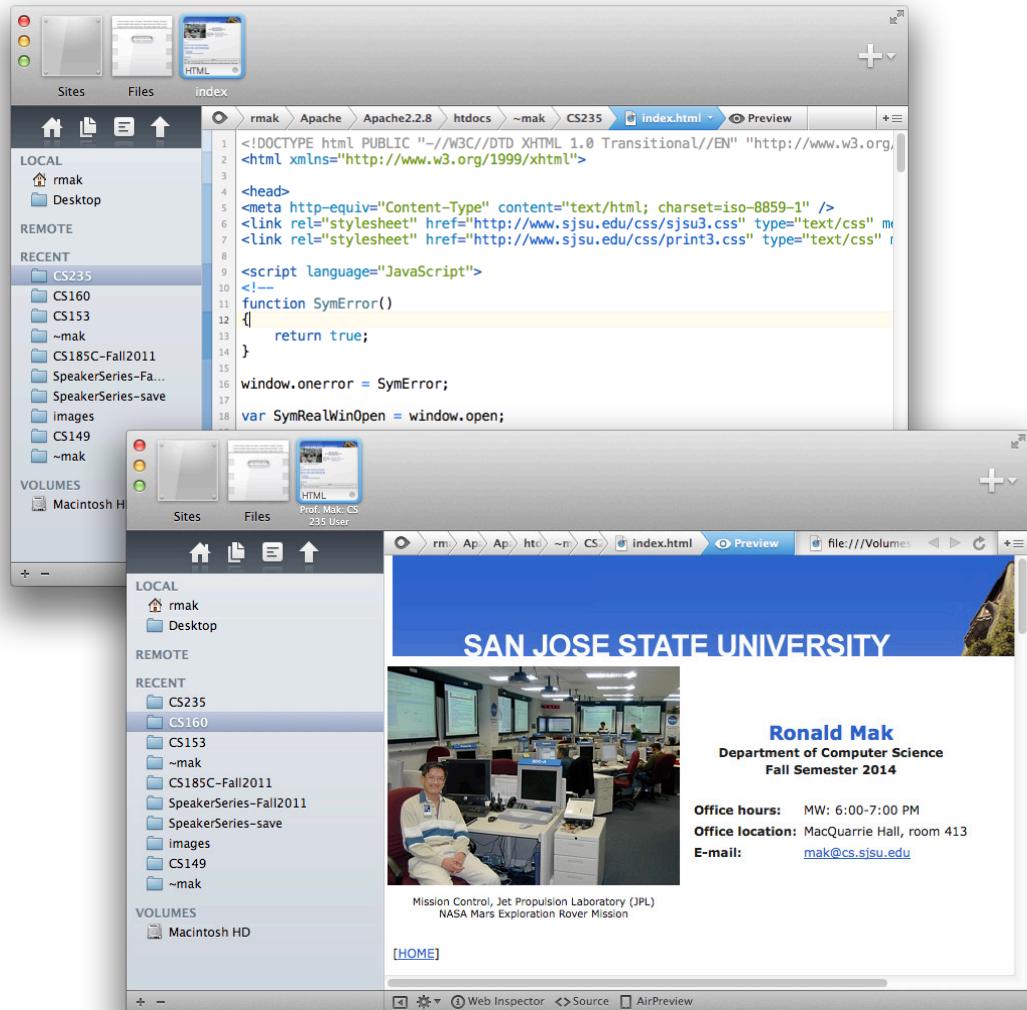
- Show multiple views of the same data in different formats

## □ When

- A single view is insufficient
- Users can choose views

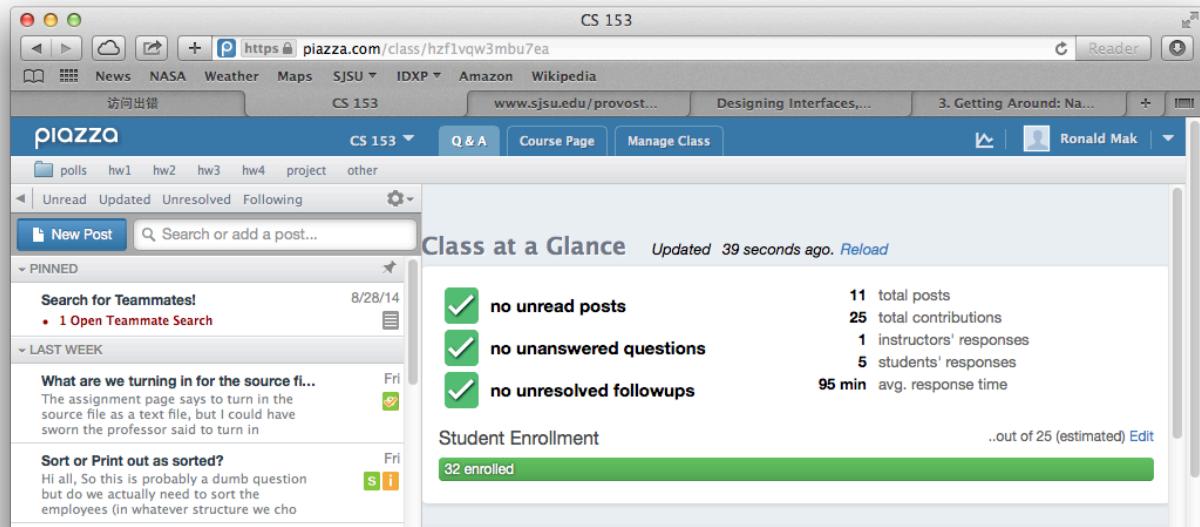
## □ Why

- Different views show different aspects of the data
- Different users may prefer different views



# Organization: Multiple Workspaces

- What
  - Multiple tabs or windows to view different pages, one at a time.
- When
  - An application can manage multiple pages of content
  - Each page is separately managed
- Why
  - Users want to multitask
  - Don't want to open multiple copies of the application



# Organization: Multilevel Help

- What
  - A variety of ways to provide help to the user
- When
  - A complex application
  - Users need with specific features or tasks
- Why
  - Provide help when and where needed
  - Help can be light- or heavyweight
  - Reminders, tips, or tutorials as necessary

The screenshot shows a Microsoft Excel window with a table of data. A tooltip is displayed above the ribbon bar, and a context menu is open over the data, providing detailed information about sorting.

**Table Data:**

	A	B
1	Dhamaal	Bell, Jarad All...
2	Dhamaal	Khare, Karan...
3	Dhamaal	LaCross, Ry...
4	Dhamaal	Yunkyoung Le...
5	Invincibles	Kondra, Akhil...
6	Invincibles	Kumar, Vimal...
7	Invincibles	Makkar, Isme...
8	Invincibles	Umaranikar, P...
9	SNAP	Baheti, Aksha...
10	SNAP	Choudhary, S...
11	SNAP	Kohli, Nupur...
12	SNAP	Nair, Prijila P...
13	SPTX	Chou, Peter...
14	SPTX	Kang, Timothy...
15	SPTX	Shweta, Shwe...
16	SPTX	Zhu, Xianghui...
17	TeamAwesome	Chen, Jie...
18	TeamAwesome	Le, Hong Nhu...
19	TeamAwesome	Li, Yu Q...
20	TeamAwesome	Liu, PingChua...

**Context Menu (About sorting):**

Data analysis begins with sorting. You can sort text (A to Z or Z to A), numbers (smallest to largest or largest to smallest), and dates and times (oldest to newest and newest to oldest) in one or more columns. You can also sort by a custom list that you create (such as Large, Medium, and Small). Or you can sort by format, including cell color, font color, or icon set. Most frequently, you will sort by column, but you can also sort by rows.

When you sort, you rearrange data into some order. In contrast, when you filter, you hide extraneous data. For more information about filtering, see [About filtering](#).

When you sort on a range of cells, the sort criteria aren't saved with your workbook. If you want to save sort criteria so that you can

# Navigation Design Patterns

---

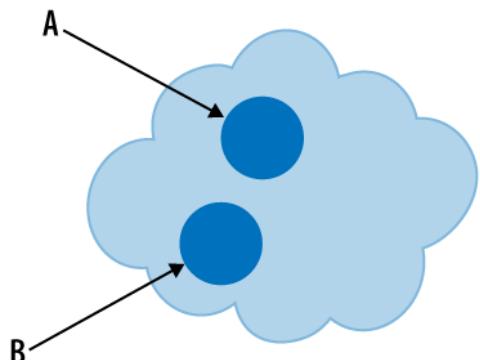
- Design patterns for how a user **navigates within an application.**
  - Can a user achieve a goal without wandering off course?
  - Where am I?
  - How did I get here?
  - Where do I go from here?
  - How much farther to go?
- **Signposts**
  - Features that help a user determine where he is.

# Navigation Basics

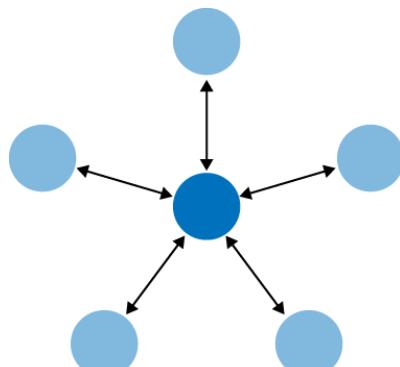
---

- Navigation errors are frustrating for users and can be costly.
  - Each new page requires a context switch.
  - Waste time on wrong paths.
- Keep low the number of page jumps required to achieve a goal.

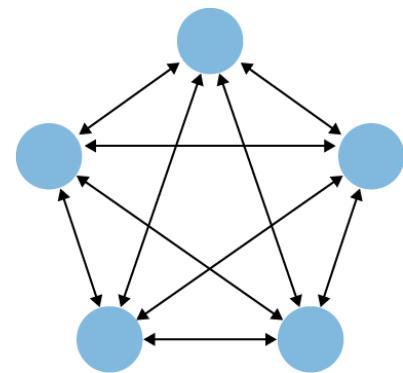
# Navigational Models



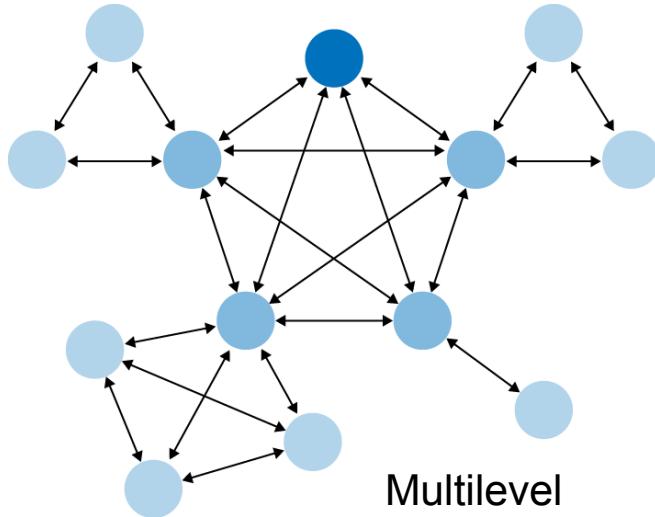
Clear entry points



Hub and spoke



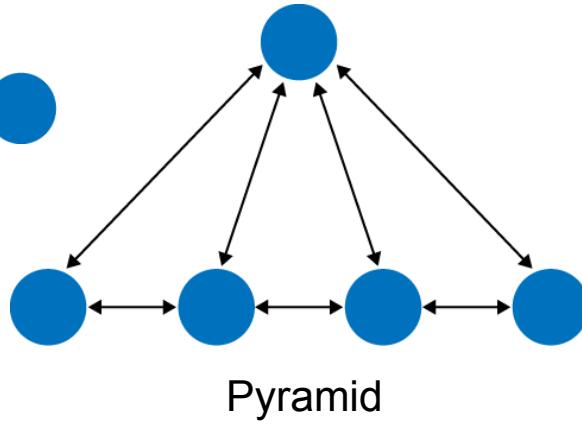
Fully connected



Multilevel

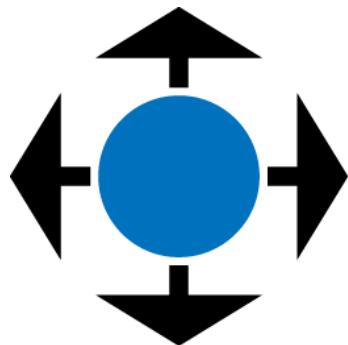


Sequential

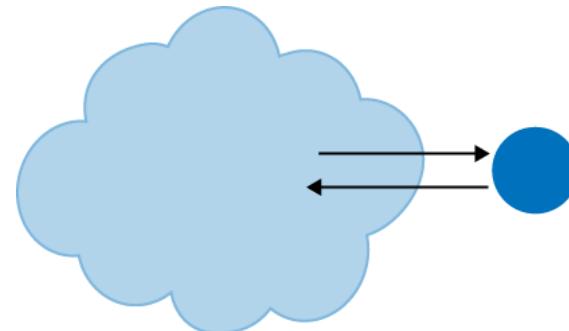


Pyramid

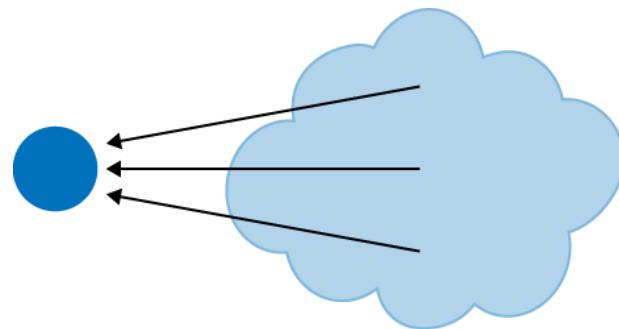
# Navigational Models, *cont'd*



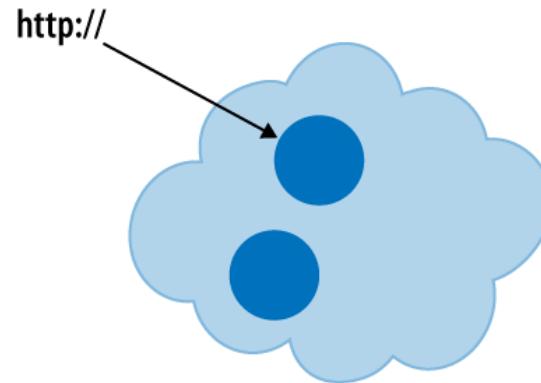
Pan and zoom



Modal dialog



Escape hatch

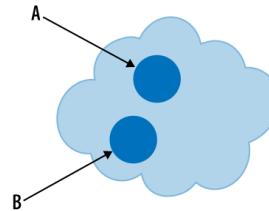


Bookmark

# Navigation: Clear Entry Points

## □ What

- Only a few main entry points into the interface
- Task oriented, clear calls to action

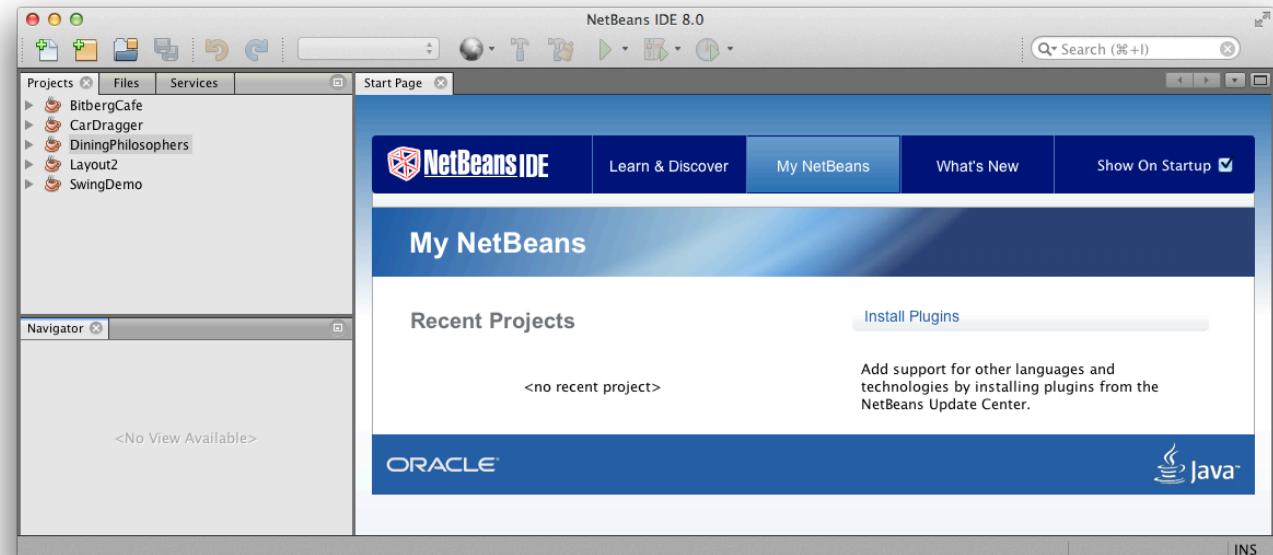


## □ When

- Ideal for first-time or infrequent users
- Can be bypassed by experienced users

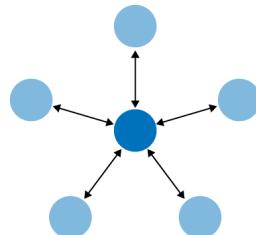
## □ Why

- Avoid too much clutter when an application is first opened
- Do not overwhelm a novice user



# Navigational: Hub and Spoke

## □ What



- A page filled with links to other pages

## □ When

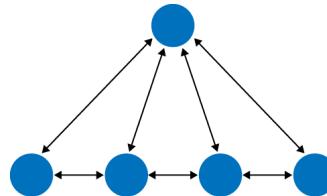
- A table of contents or a portal to show the user where to go from here

## □ Why

- The user can decide where to go
- Return to this page to go elsewhere

# Navigational: Pyramid

- What
  - Sequential pages with back and next links
  - A link to return to the top
- When
  - Sequential text organized by chapters
  - Return to the table of contents at the top to pick another chapter
- Why
  - Matches a user's mental model of a tutorial



Pascal Variable Types

www.tutorialspoint.com/pascal/pascal\_variable\_types.htm

tutorialspoint SIMPLY EASY LEARNING

HOME MENU

Pascal - Variable Types

Advertisements

THE MANAGED CLOUD MEANS... Help Within 15 Seconds SEE FOR YOURSELF rackspace the #1 managed cloud company

Previous Page Next Page

A variable is nothing but a name given to a storage area that our programs can manipulate. Each variable in Pascal has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.

The name of a variable can be composed of letters, digits, and the underscore character. It must begin with either a letter or an underscore. Pascal is **not case-sensitive**, so uppercase and lowercase letters mean the same here. Based on the basic types explained in previous chapter, there will be following basic variable types:

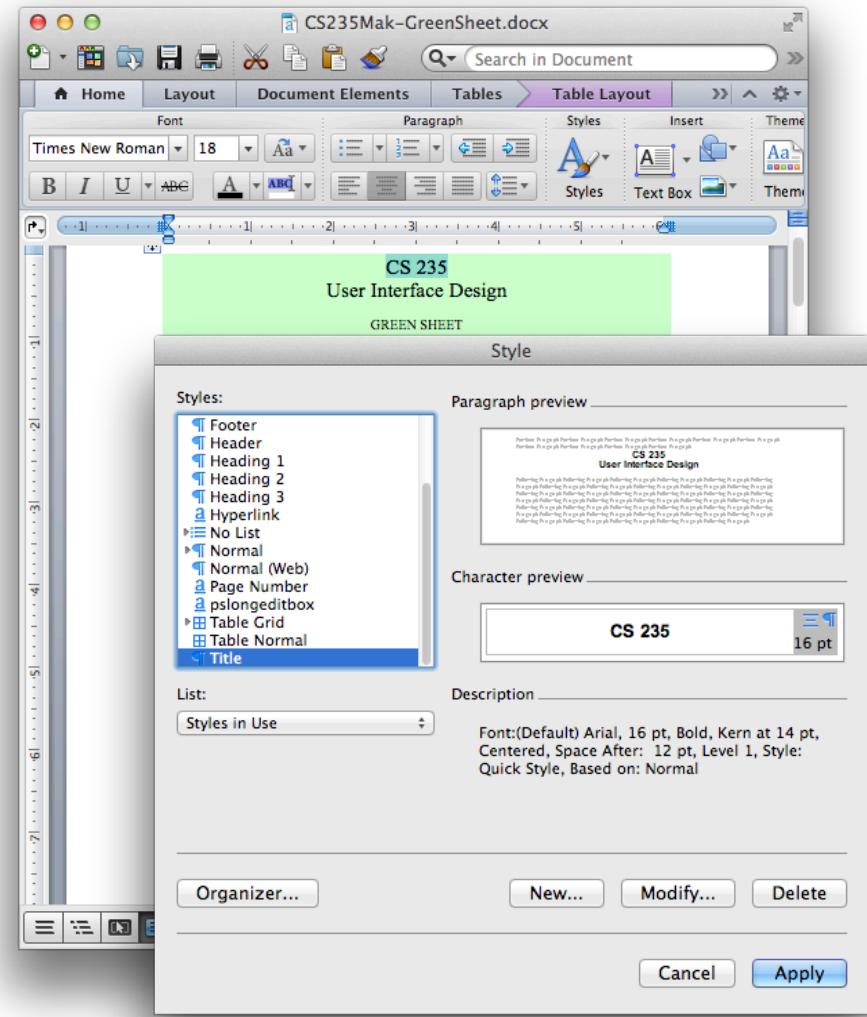
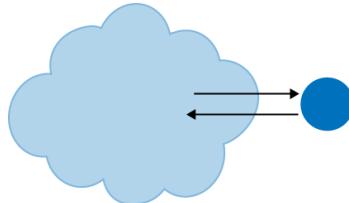
Basic Variables in Pascal

Type	Description
Character	Typically a single octet (one byte). This is an integer type.
Integer	The most natural size of integer for the machine.
Real	A single-precision floating point value.
Boolean	Specifies true or false logical values. This is also an integer type.
Enumerated	Specifies a user-defined list.
Subrange	Represents variables, whose values lie within a range.
String	Stores an array of characters.

Pascal programming language also allows defining various other types of variables, which we will cover in subsequent chapters like Pointer, Array, Records, Sets, and Files, etc. For this chapter, let us study only basic variable types.

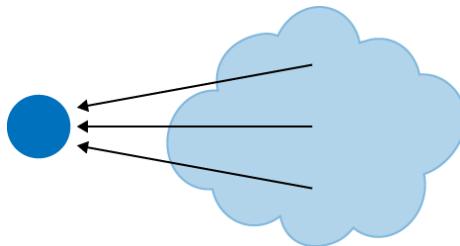
# Navigation: Modal Dialog

- What
  - Show only one page with no navigation options
  - The user must complete the page's task
- When
  - The application can't proceed without user input
- Why
  - The user can't ignore the page and must provide input
  - Can be disruptive and annoying to the user



# Navigational: Escape Hatch

- What
  - The user can abort a task
- When
  - The application takes the user down a path but the user may want to back out
- Why
  - The user changes his mind or recognizes that he is going down the wrong path



Amazon.com Sign In

https://www.amazon.com/ap/signin?\_enc

amazon

Sign In

What is your e-mail address?

My e-mail address is:

Do you have an Amazon.com password?

No, I am a new customer.

Yes, I have a password:

[Forgot your password?](#)

Keep me signed in. [Details](#)

Sign in using our secure server

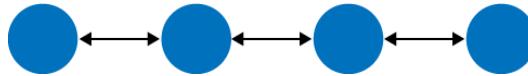
Sign In Help

Forgot your password? [Get password help.](#)

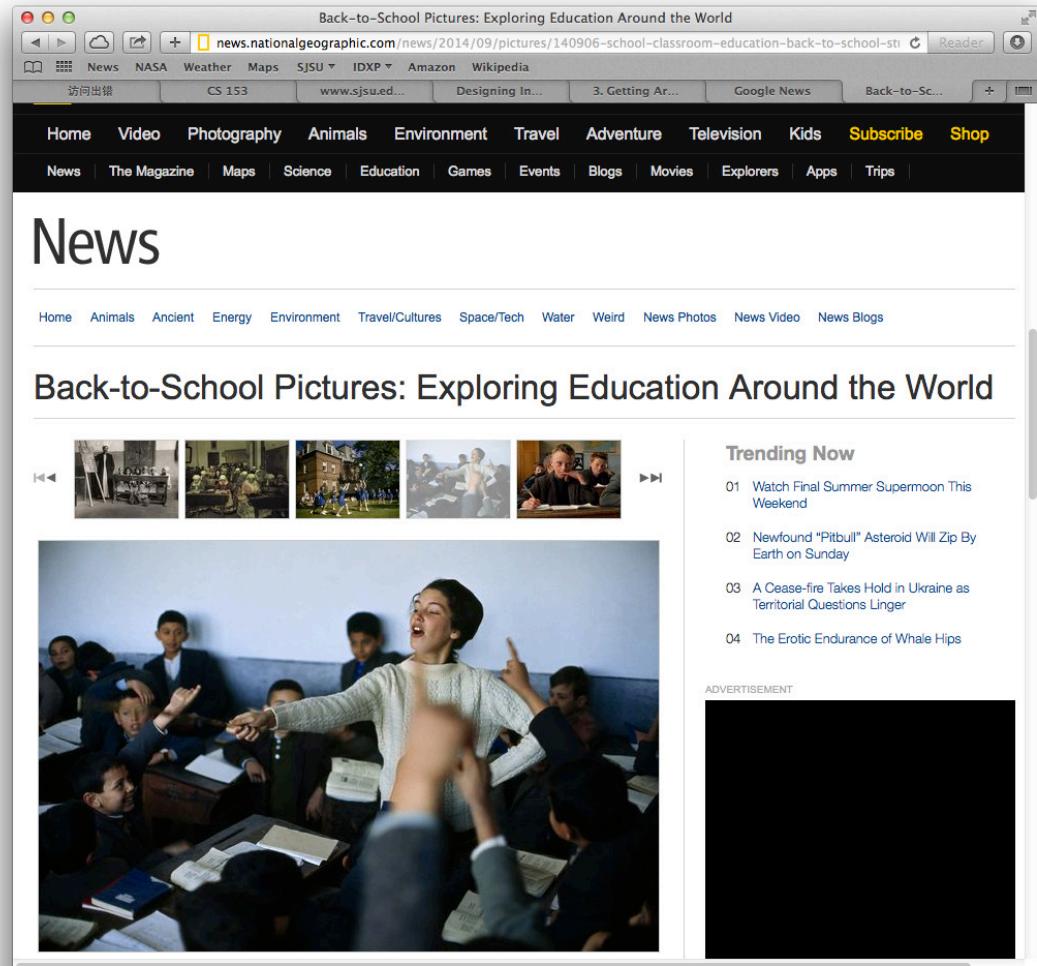
Has your e-mail address changed? [Update it here.](#)

Conditions of Use Privacy Notice  
© 1996-2014, Amazon.com, Inc. or its affiliates

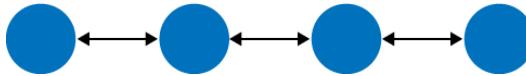
# Navigational: Sequence Map



- What
  - Show a map of pages in sequence
  - Include “you are here”
- When
  - The application takes the user along a linear path
- Why
  - The map tells the user how far he has gone and how much farther he has to go



# Navigational: Annotated Scrollbar



- What
  - The scrollbar says “you are here”.
- When
  - The application is document-centric or allows pan-and-zoom
- Why
  - While manipulating the scrollbars, the user needs to know what part of content is being displayed

The screenshot shows a Microsoft Word document window titled "CS235Mak-GreenSheet.docx". The ribbon tabs are Home, Layout, Document Elements, Tables, Table Layout, Font, Paragraph, Styles, Insert, and Theme. The "Table Layout" tab is selected. The document content includes:

- Procedure**: Students will work in small teams on UI and UX projects. It will be important for you to participate in your team and in class discussions.
- Programming skills are not strictly required, although you should be comfortable using prototyping tools. You will not be expected to create complete applications – just prototypes that are sufficient to demonstrate your UI and UX design.
- We will use the scheduled final examination time for final project presentations.

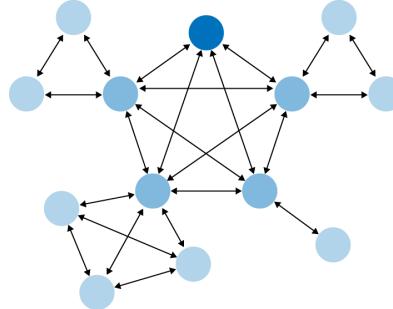
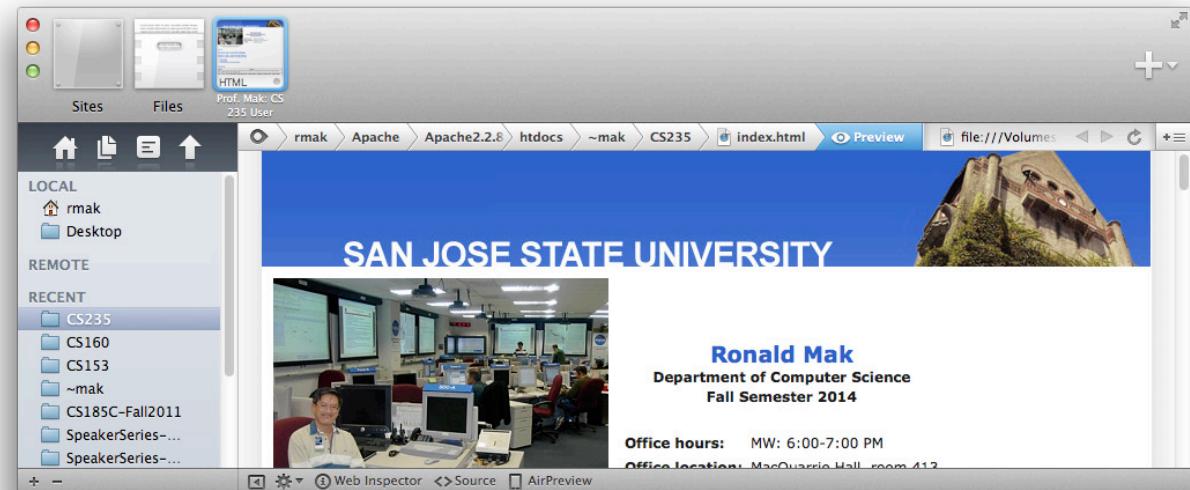
**Schedule**: Subject to change with fair notice.

**Readings**:

Week	Dates	Topics and activities	Readings
1	Aug 25, 27	Introduction Agile design Requirements and use cases Form design teams	UIDE 1-7
2	Sep 3	Desktop/laptop design projects UI and functional specifications Storyboards and prototypes	UIDE 8-11
3	Sep 8, 10	Desktop UI design patterns	UIDE 16
4	Sep 15, 17	Desktop UI design patterns, cont'd Usability testing	UIDE 20-24
5	Sep 22, 24	Desktop design presentations	
6	Sep 29 Oct 1	Desktop design presentations Web design projects	UIDE 17
7	Oct 6, 8	Web design patterns	
8	Oct 13, 15	Web design presentations	

At the bottom, the status bar shows "Print Layout View", "Sec 1", "Pages: 85%", and a zoom slider.

# Navigational: Breadcrumbs

- What
    - On each page in a deep hierarchy, show a list of the parent pages in order
  - When
    - In a deep hierarchy, users need help to keep track of where they are.
    - A user can go back up to any point in the hierarchy.
  - Why
    - Allow a user to move forward and back through a deep hierarchy
    - The user knows where he is in the hierarchy
- 
- 



# Link to Design Video

---

- <https://www.youtube.com/watch?v=BKorP55Aqvg>



Lecture #05

# CS 235: User Interface Design

## February 5 Class Meeting

---

Department of Computer Science  
San Jose State University

Spring 2015  
Instructor: Ron Mak

[www.cs.sjsu.edu/~mak](http://www.cs.sjsu.edu/~mak)

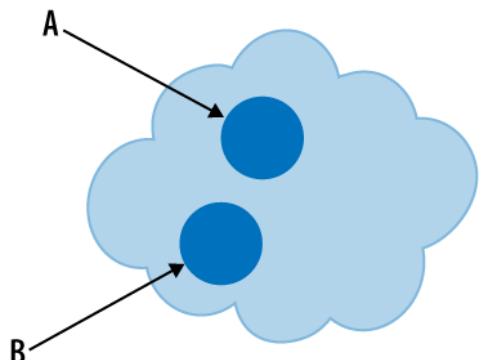


# Organization Design Patterns

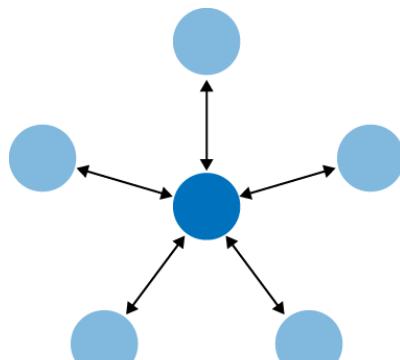
---

- Feature, Search, and Browse
- News Stream
- Picture Manager
- Dashboard
- Canvas + Palette
  - Wizard
  - Settings Editor
  - Alternate Views
  - Multiple Workspaces
  - Multilevel Help

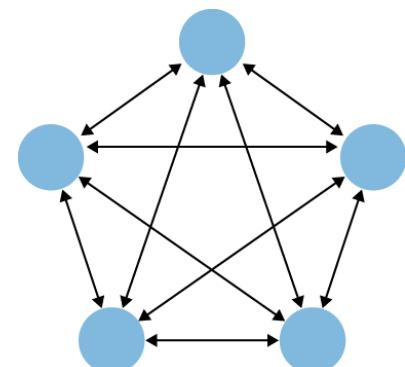
# Navigation Design Patterns



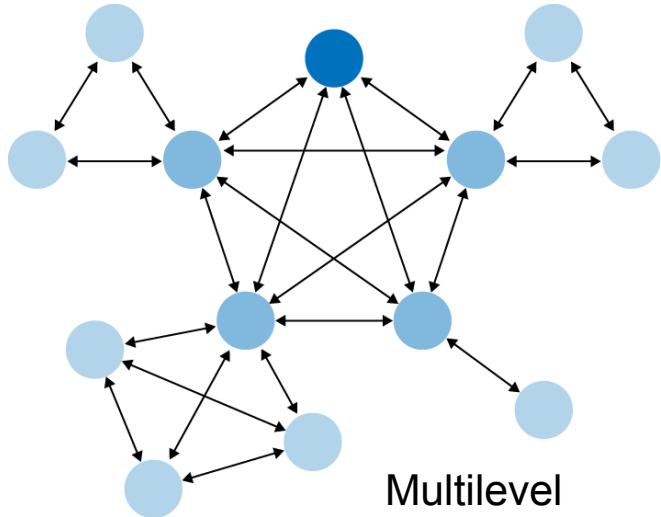
Clear entry points



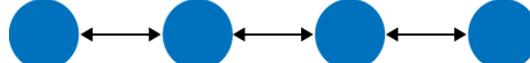
Hub and spoke



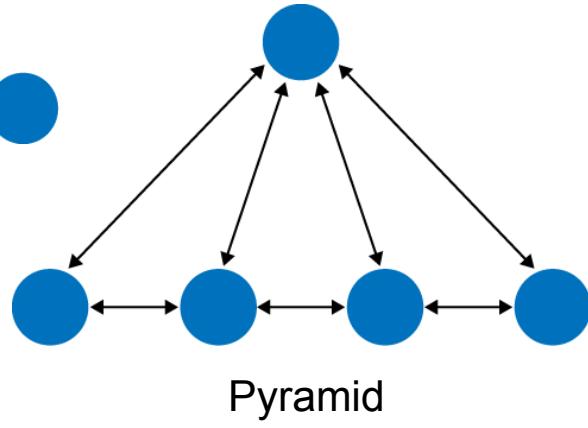
Fully connected



Multilevel

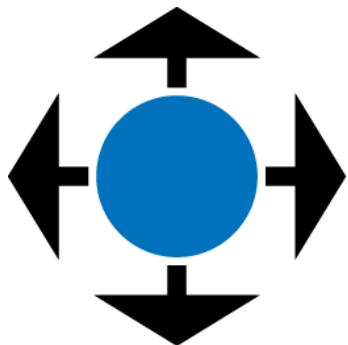


Sequential

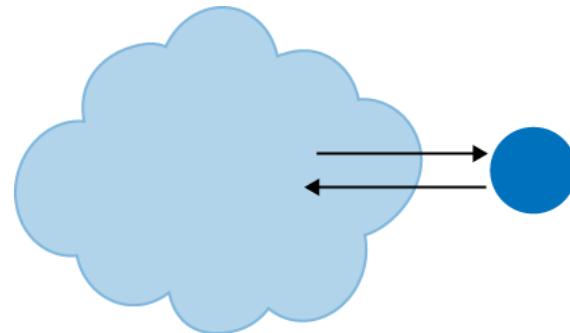


Pyramid

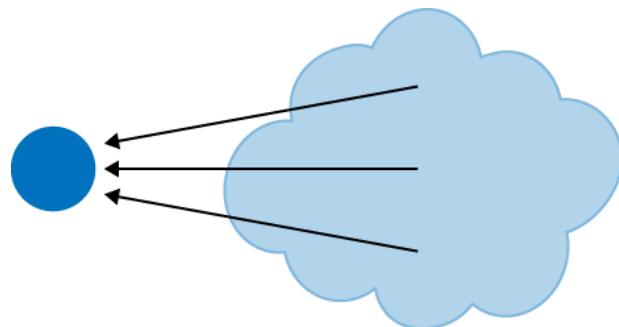
# Navigational Design Patterns, *cont'd*



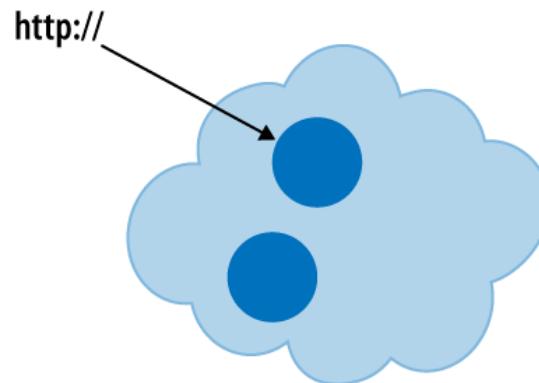
Pan and zoom



Modal dialog



Escape hatch



Bookmark

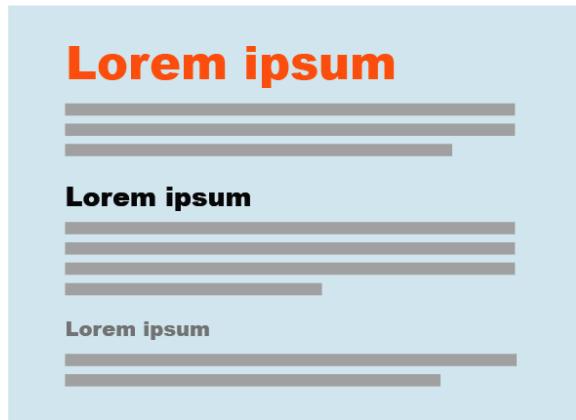
# Page Layout

---

- Manipulate **page attention** to convey
  - meaning
  - sequence
  - points of interaction
- Visual hierarchy:
  - What's important?
  - What's related?

# How to Show Importance

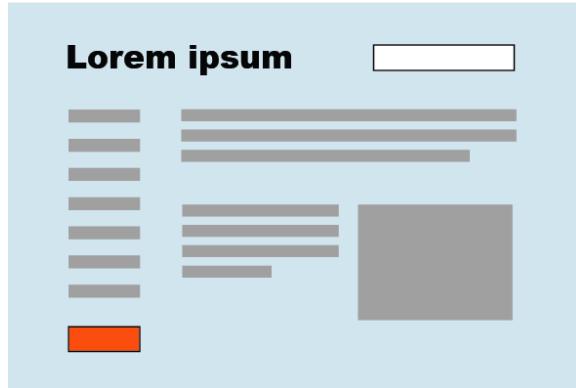
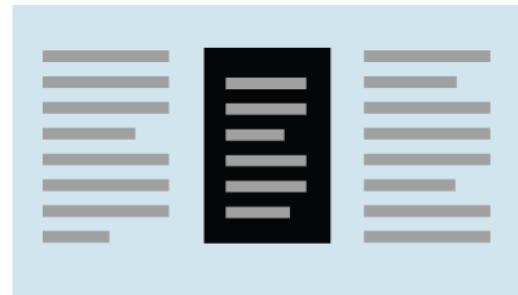
Text sizes to show hierarchy



Text density



Background color



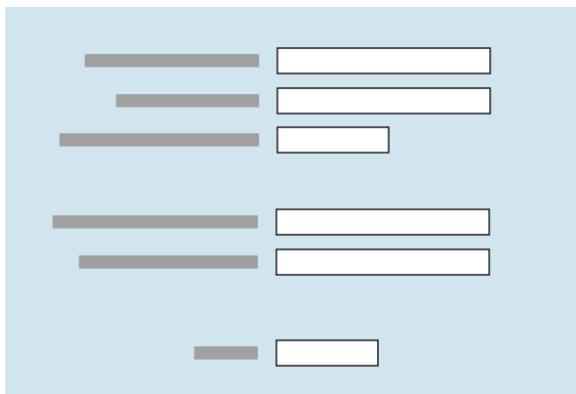
Emphasize small items

Visual rhythm



Position and size

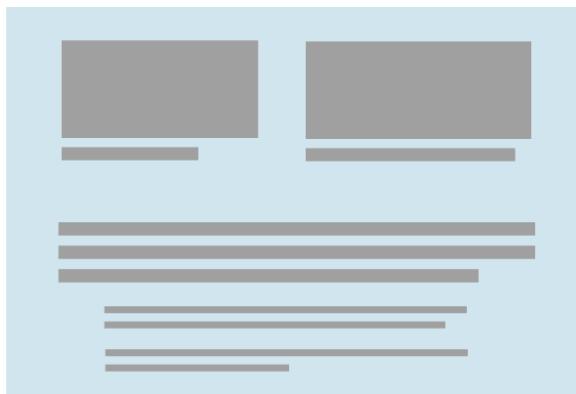
# How to Show Relationships



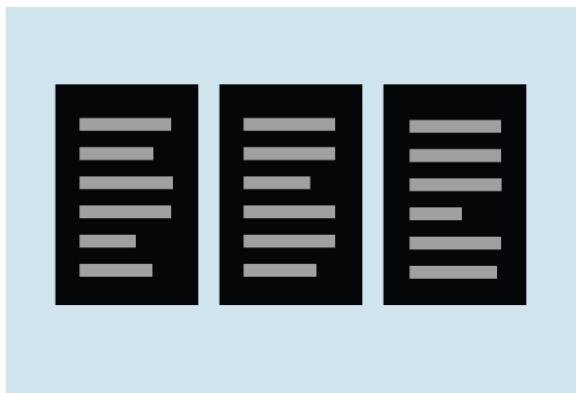
Grouped related items



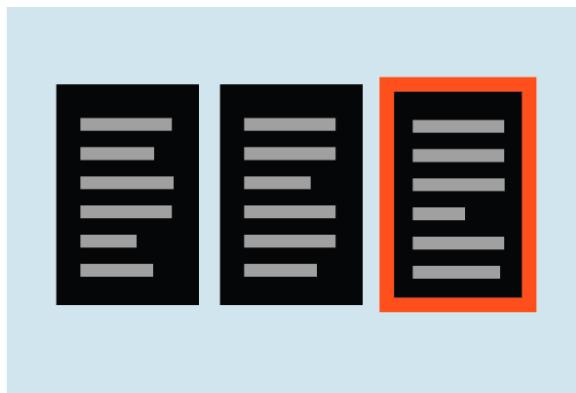
List of items



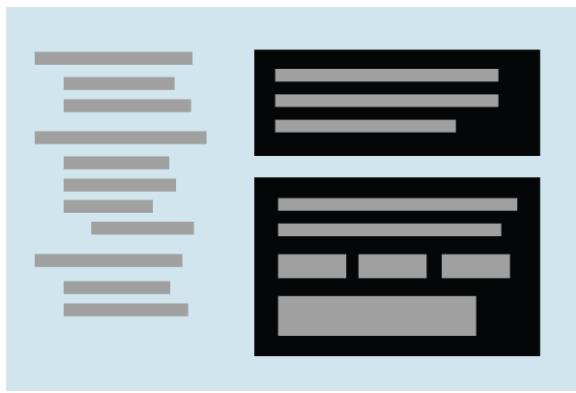
Captions and comments



Peer items



One distinguished item



Containment

# Four Gestalt Principles

---

**gestalt:** An organized whole that is perceived as more than the sum of its parts.

## □ Proximity

- Viewers will associate together items that are placed close together.

## □ Similarity

- Viewers will associate two things that have the same shape, size, orientation, etc.

—

# Four Gestalt Principles, *cont'd*

---

## □ Continuity

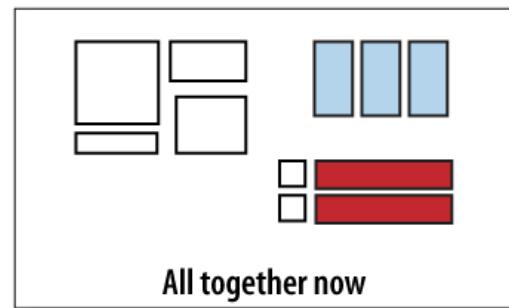
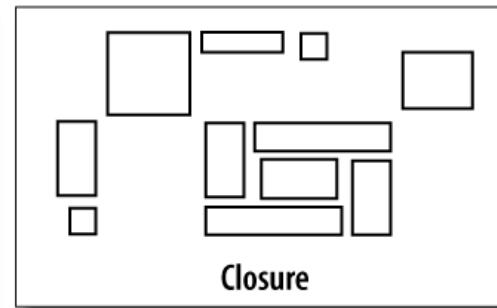
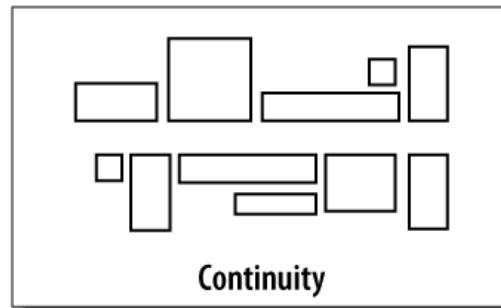
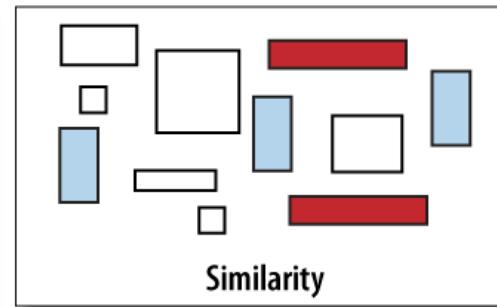
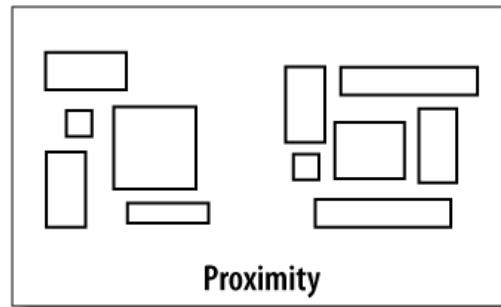
- Viewers' eyes want to see continuous lines and curves formed by the alignment of items.

## □ Closure

- Viewers' eyes want to see implicit simple closed forms such as rectangles.

—

# Four Gestalt Principles, *cont'd*



# Page Layout Design Patterns

---

- Page layout design patterns provide specific ways to incorporate the layout concepts and principles.
- Solve layout problems.



# Layout: Visual Framework

## □ What

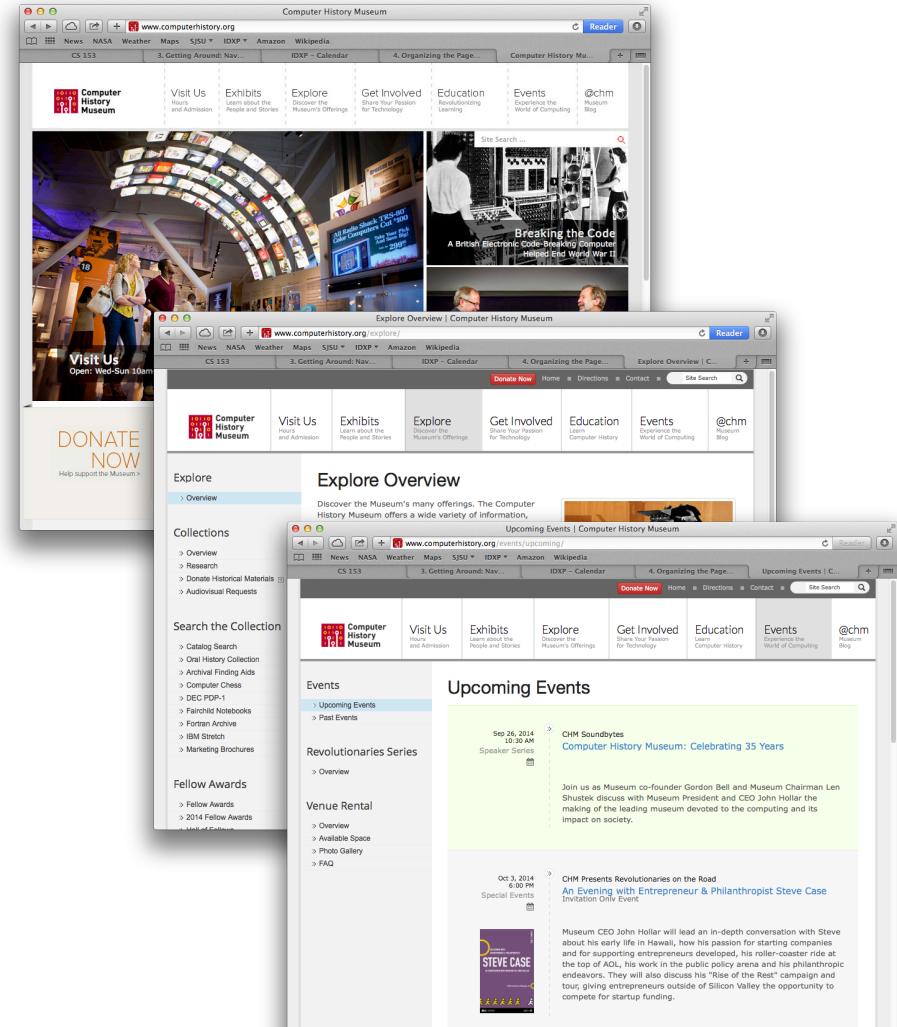
- A common framework for all application pages but which allows flexibility to handle varying page content

## □ When

- Designing an application with multiple pages
- You want the pages to appear to belong together

## □ Why

- Provide consistency
- Help page content stand out



# Layout: Center Stage

## □ What

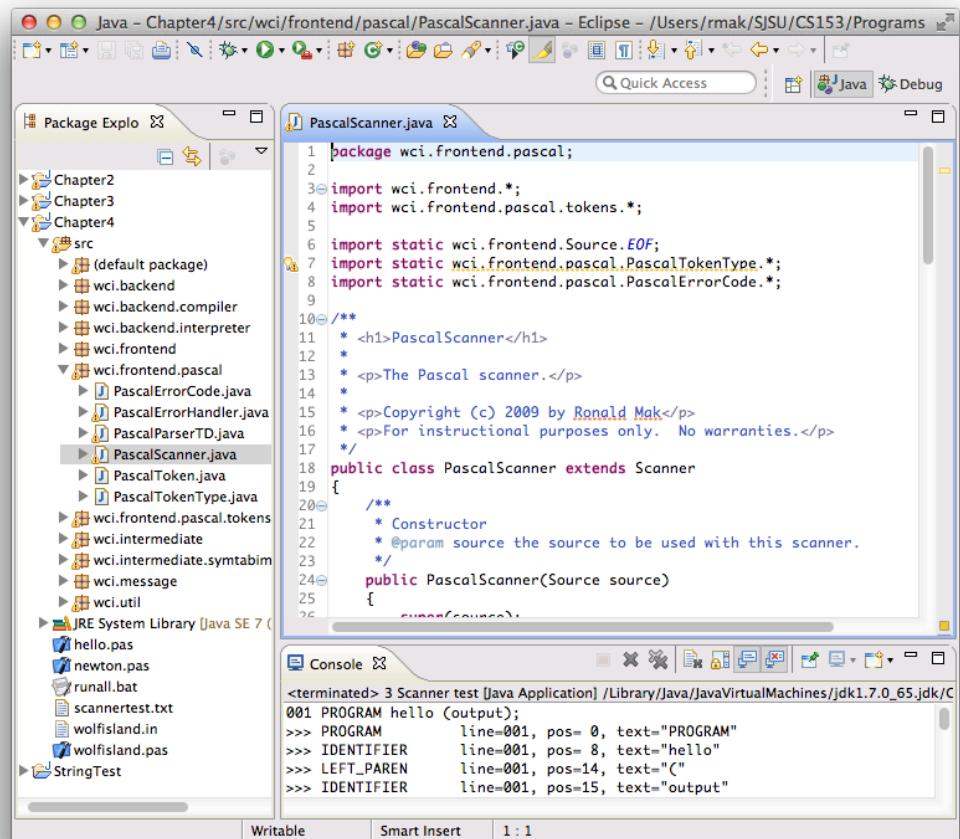
- The most important part of the UI has the largest subsection of the page
- Secondary windows cluster around in smaller subpanels

## □ When

- The page's primary job is to show a single unit of information or to permit a single task

## □ Why

- Guide the user to the most important part



# Layout: Grid of Equals

## □ What

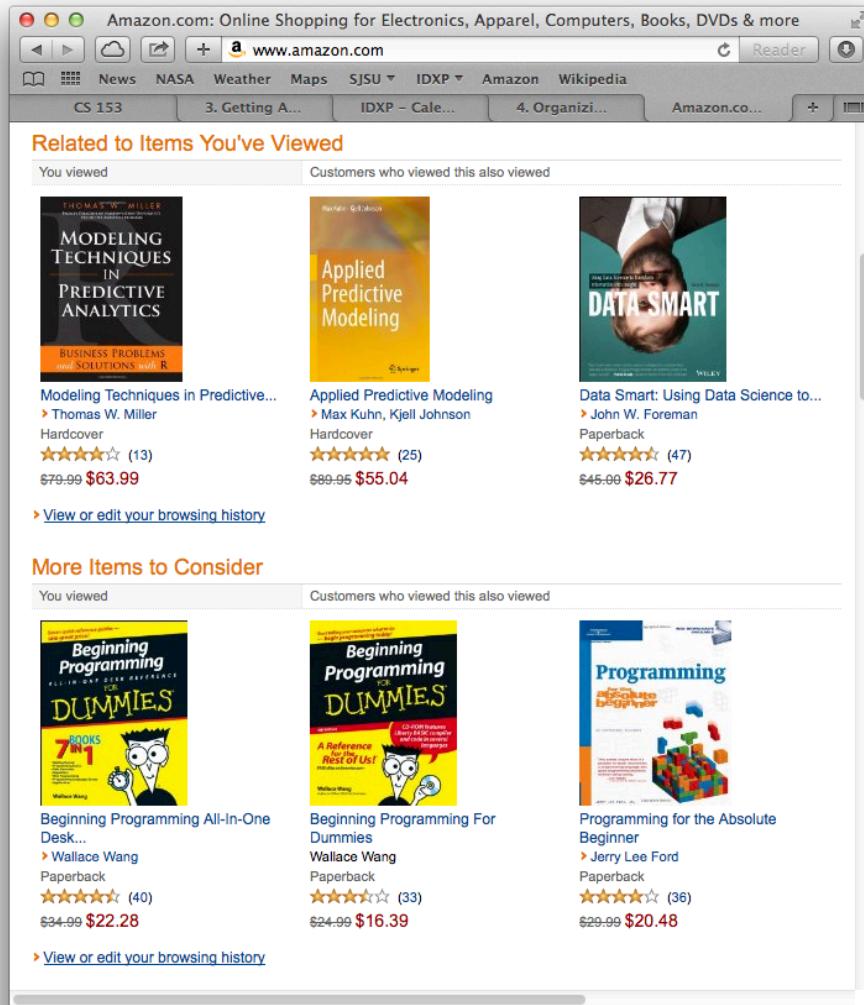
- Arrange equally important items in a grid, each with equal visual weight

## □ When

- Many items have similar style and importance
- Allow the user to preview

## □ Why

- Give each item equal importance
- Tell the user the items are similar
- Neat and orderly



# Layout: Tiled Sections

## □ What

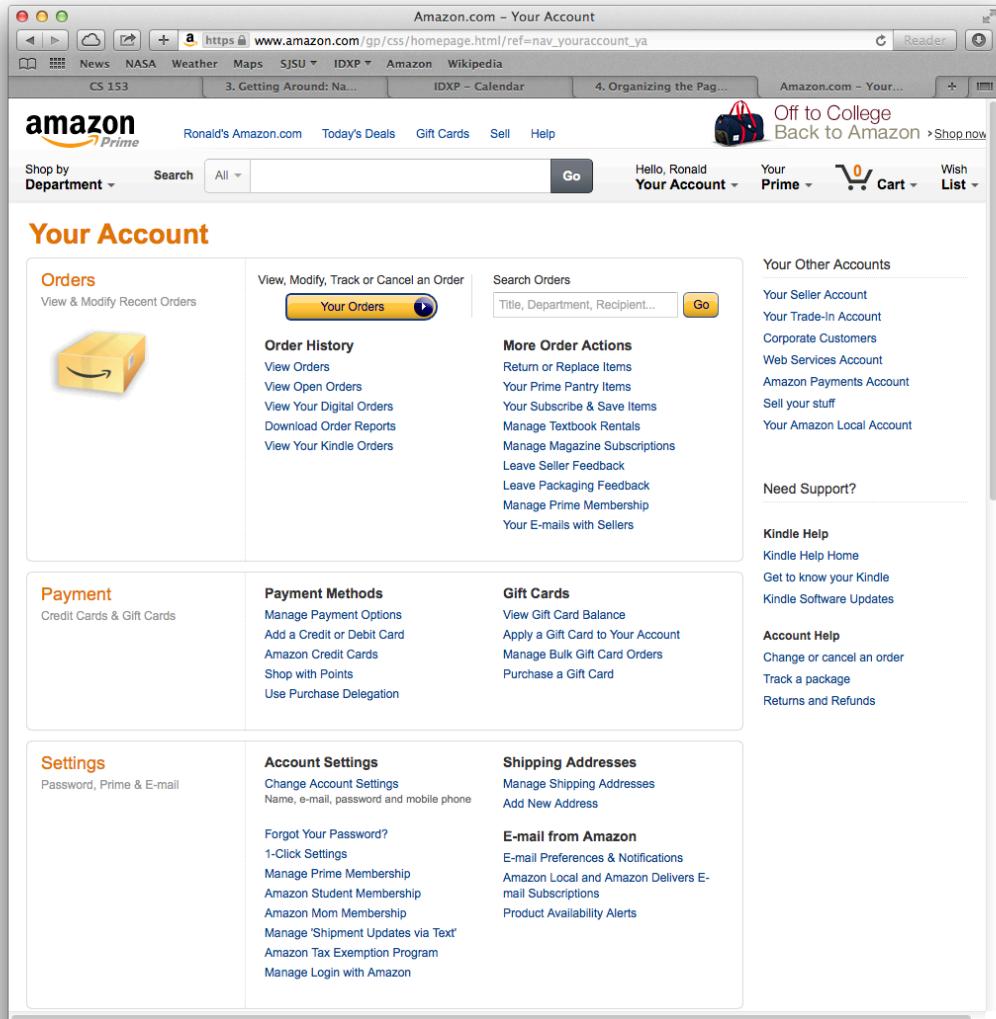
- Visually separate sections of content
- Each section has a strong title

## □ When

- Make the page easy to scan with grouped content

## □ Why

- Content is structured into easily digestible chunks
- Obvious information architecture



# Layout: Module Tabs

## □ What

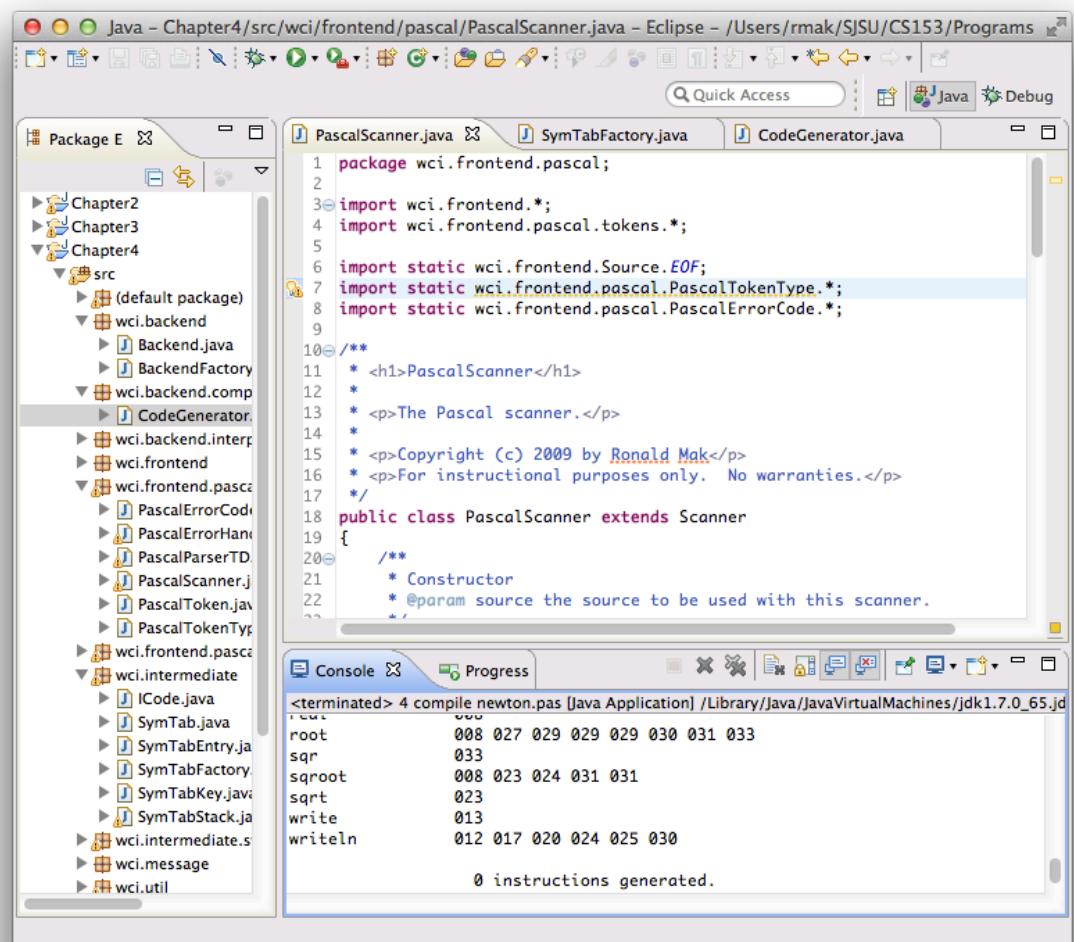
- Put content modules into separate tabs
- Only one tab is visible at a time

## □ When

- Multiple content types
- Users only need to see one type at a time

## □ Why

- Tabs have a familiar physical model
- Less cluttered user interface



# Layout: Collapsible Panel

## □ What

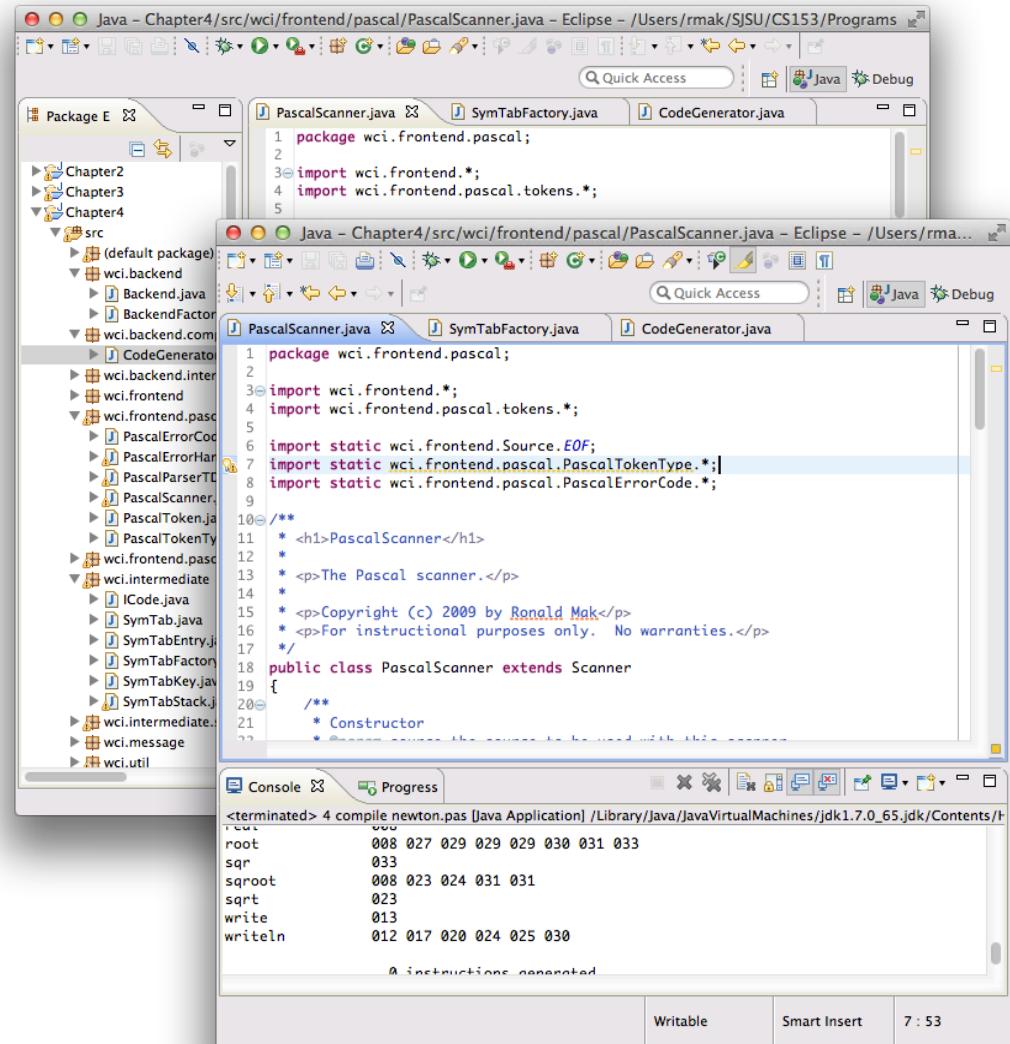
- The user can open and close secondary content panels

## □ When

- Not enough room to show all content panels
- Different users have different needs

## □ Why

- Allow the user to choose what to see



# Layout: Movable Panels

## □ What

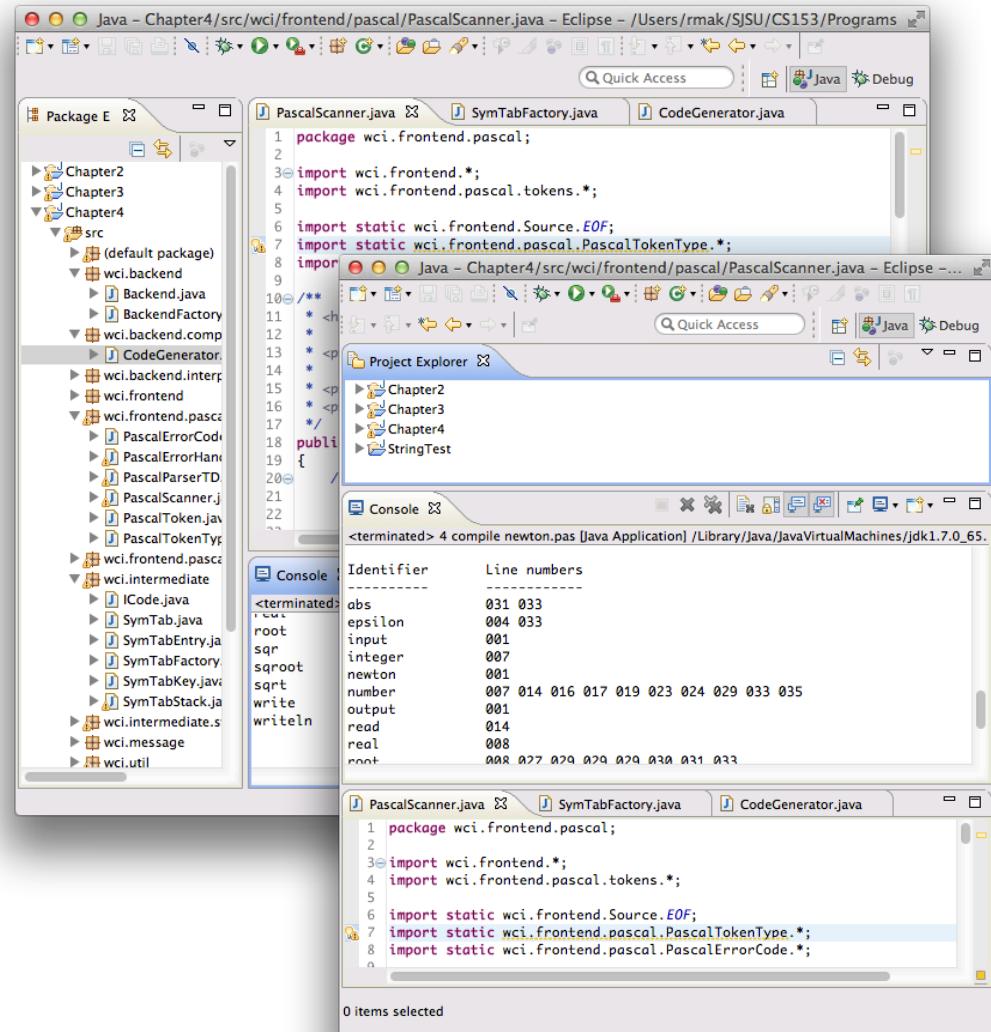
- The user can rearrange content panels

## □ When

- Different arrangements for different purposes
- Different users have different needs

## □ Why

- Allow the user to arrange the panels to suit his needs and preferences



# Layout: Left-Right Alignment

## □ What

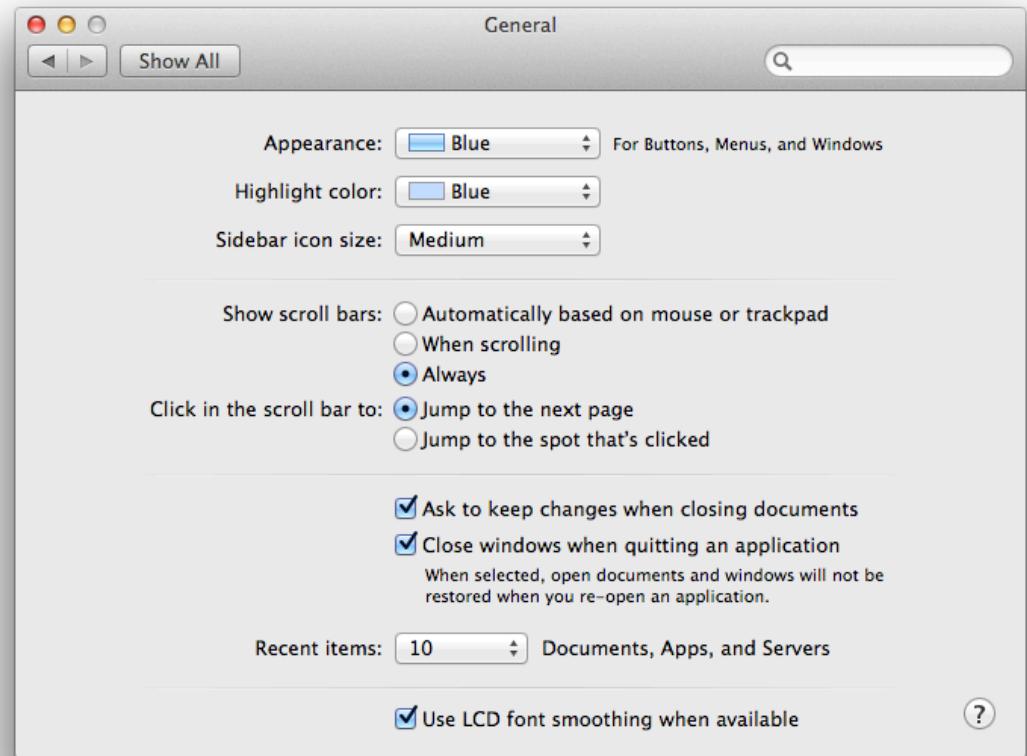
- A two-column arrangement
- Right-aligned labels on the left
- Left-aligned items on the right

## □ When

- An input form or any other set of labeled items

## □ Why

- Neat with strong perceptual grouping



# Layout: Responsive Enabling

## □ What

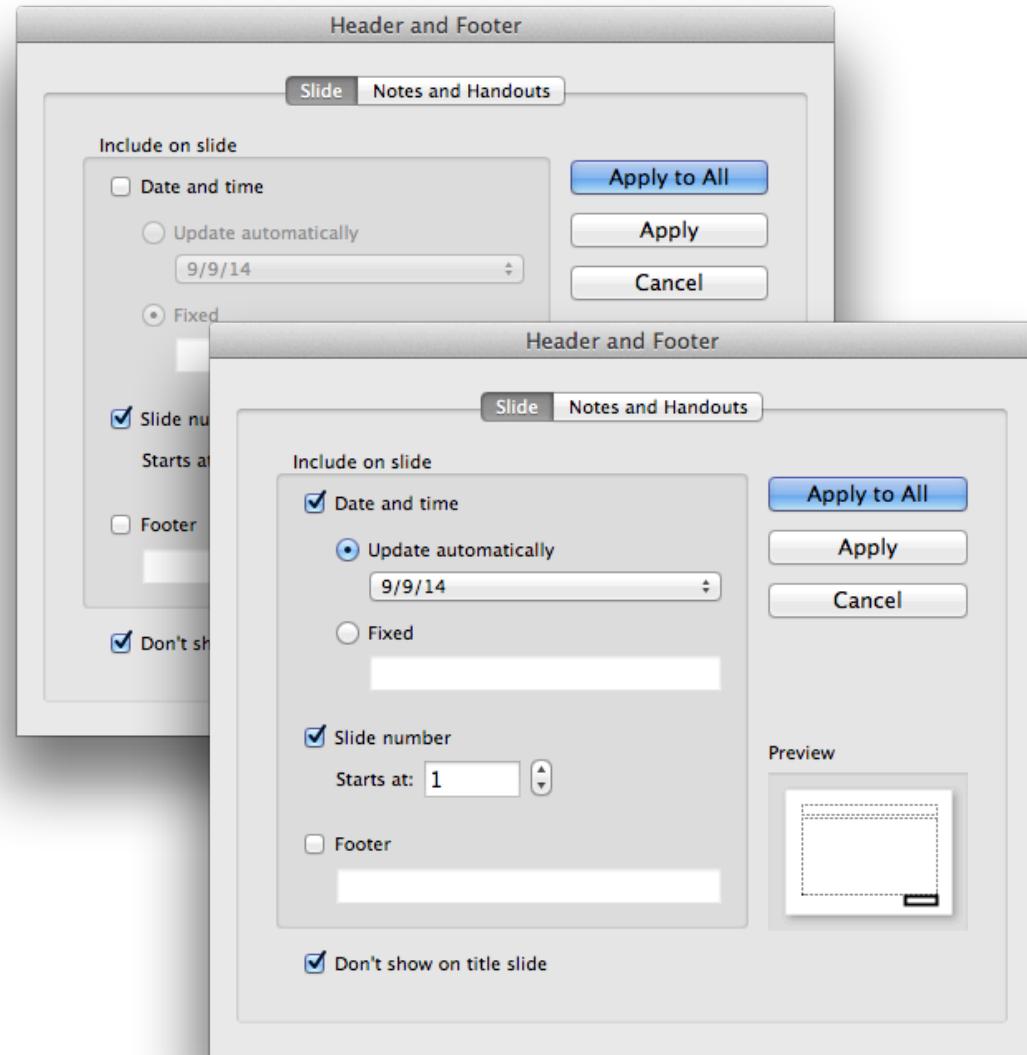
- Progressively enable UI components as the user selects options

## □ When

- Lead a user through a complex task step by step.
- The user can easily change his mind

## □ Why

- The user can see all the options
- Only the required options are enabled



# Layout: Self-Adjusting Layout

## □ What

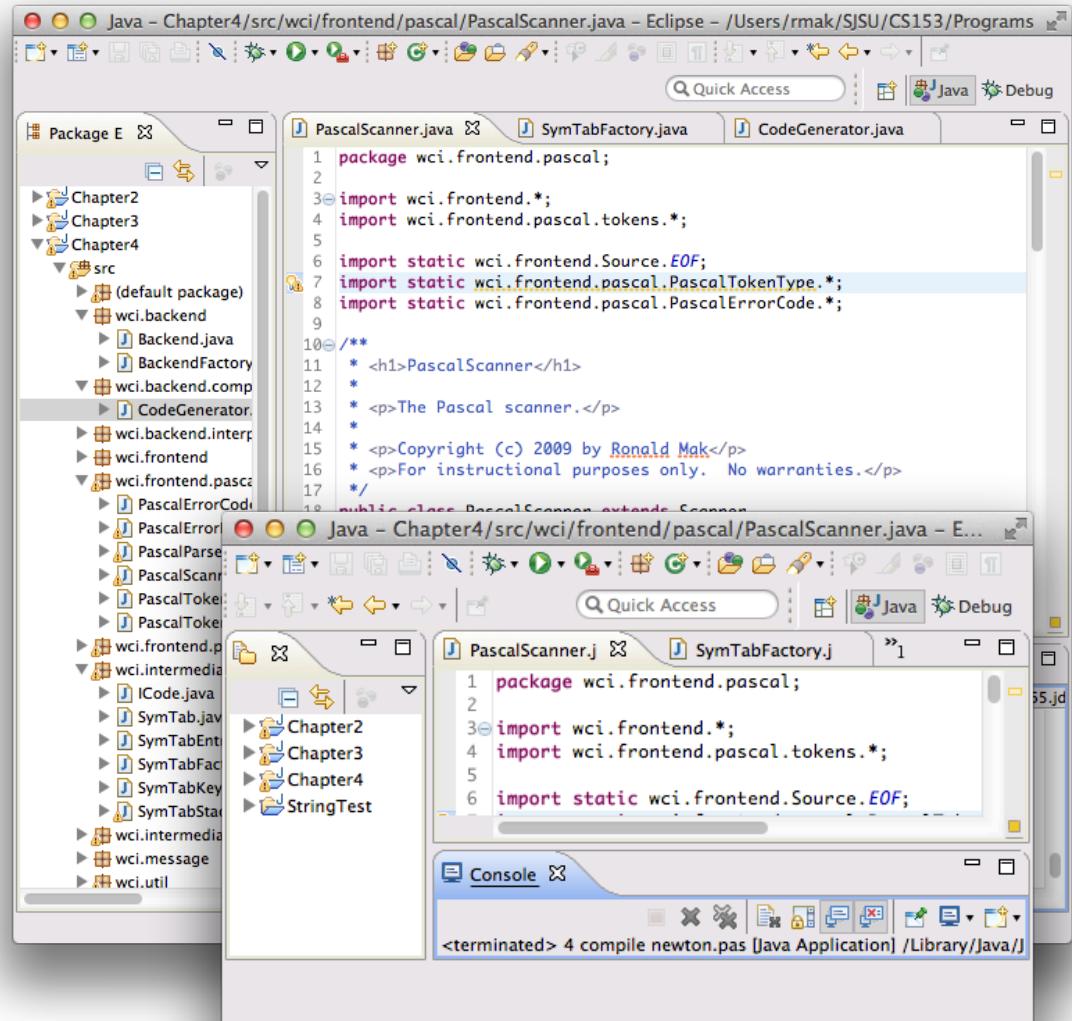
- The page contents automatically resize to maintain the same arrangement for a different window size and aspect ratio

## □ When

- Users can resize windows as desired

## □ Why

- Users choose window sizes and aspect ratios unpredictably



# Application Prototype

---

- Create a **prototype** of your web application.
  - Use organization, navigation, and page layout design patterns.
  - Your choice of prototyping tool.
  
- A prototype should **look and act** like the real application.
  - But it doesn't have to really work.
  - It can use hardcoded data.
  - Not all execution paths need to be enabled.

# Application Prototype, *cont'd*

---

- Your prototype should enable you to demonstrate at least 2 of your use cases.
- We will use your prototypes to perform **usability testing** as part of Assignment #2.
  - Details of the assignment to be provided later.



# Application Prototype, *cont'd*

---

- Be able to explain:

- The purpose of the application.
- What benefits does it provide the user?
- What were the requirements?
  - Highlight the ones that your prototype meets.
  - Which use cases does your prototype enable?

# Why Display Lists of Items?

---

- Get an overview of all items
  - Browse item by item
  - Search for a specific item
  - Sort and filter items
  - Rearrange, add, delete, or recategorize items
- 



# Information Architecture for Lists

---

- Length of the list
- Order of the items
- Item grouping
- Item types
- User interaction with the list
- Dynamic behavior of the list

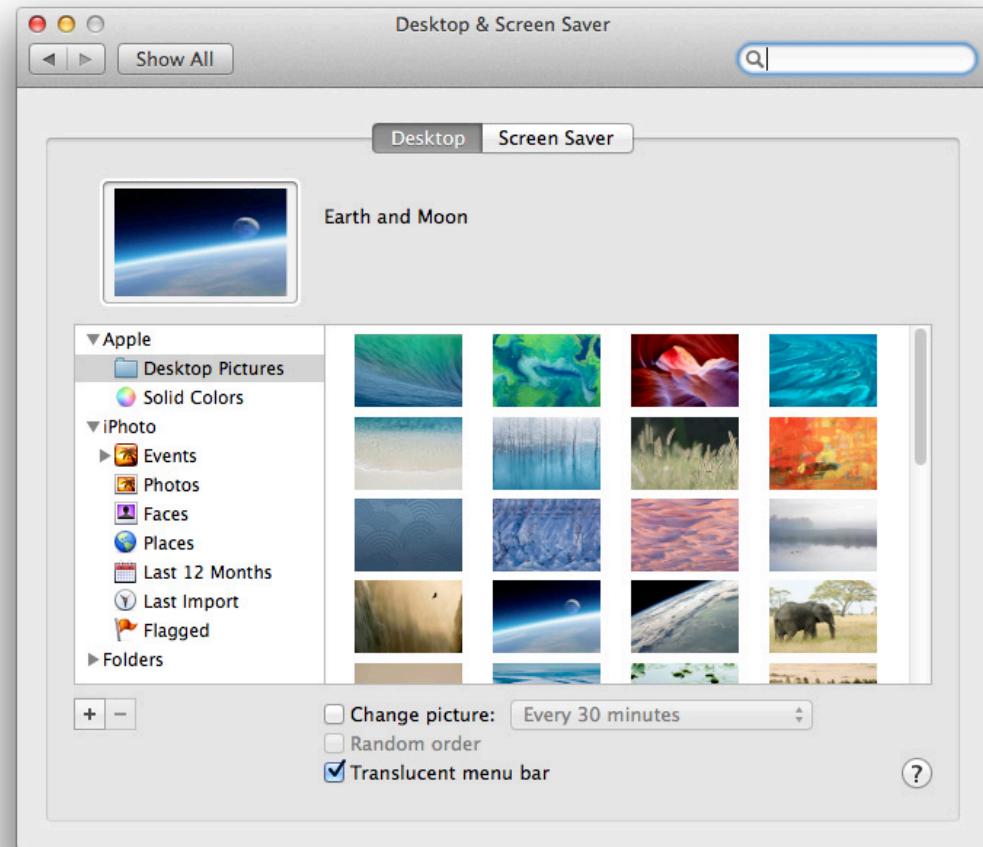
# Questions about Displaying Lists

---

- How to show the details of an item that the user selected from a list?
- How to display a list that has graphical items?
- How to manage a very long list?
- How to display list that is organized into categories or hierarchies?

# Lists: Two-Panel Selector

- What
  - Two panels side-by-side
  - First panel show a list of selectable items
  - Second panel shows the selected item's content
- When
  - Show the entire list of items
  - Each item has content
- Why
  - Reduced physical effort
  - Reduced visual cognitive load
  - Less user memory burden



# Lists: One-Window Drilldown

## □ What

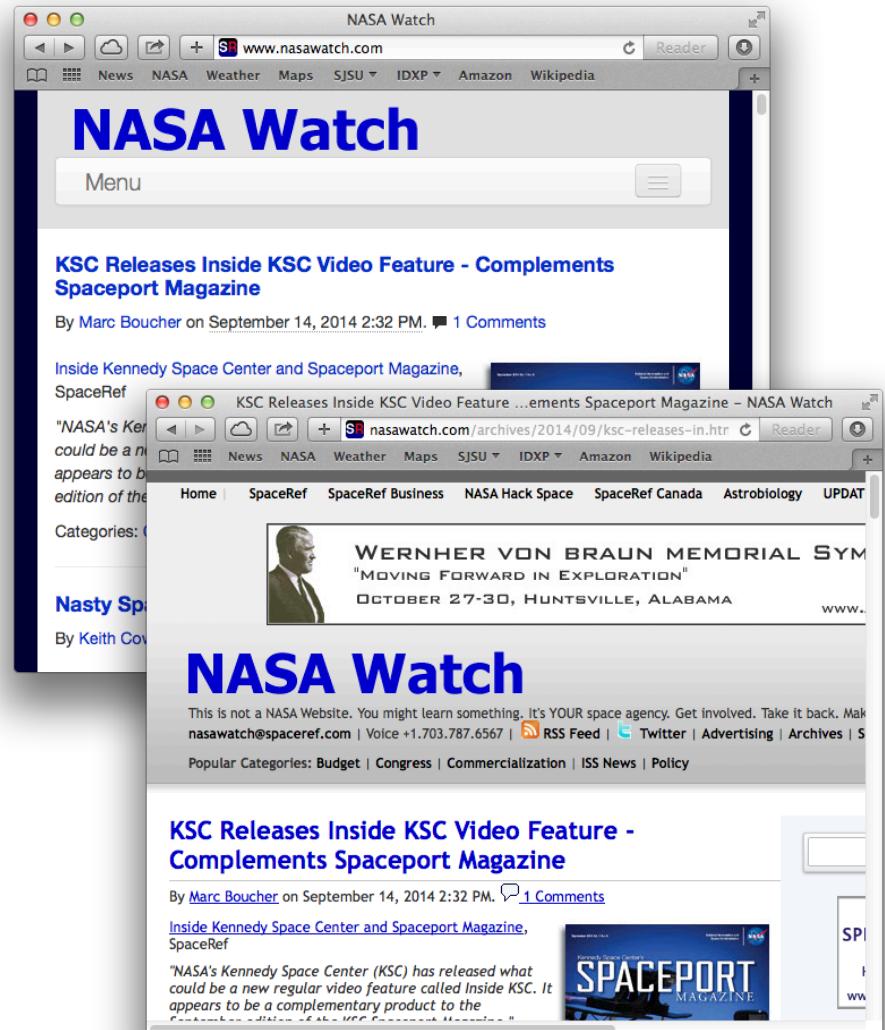
- A list of selectable items in a single window
- When the user selects an item, replace the list with the item details

## □ When

- Each item has content
- Limited available space
- Large list and large content

## □ Why

- Only option when space is tight



# Lists: List Inlay

## □ What

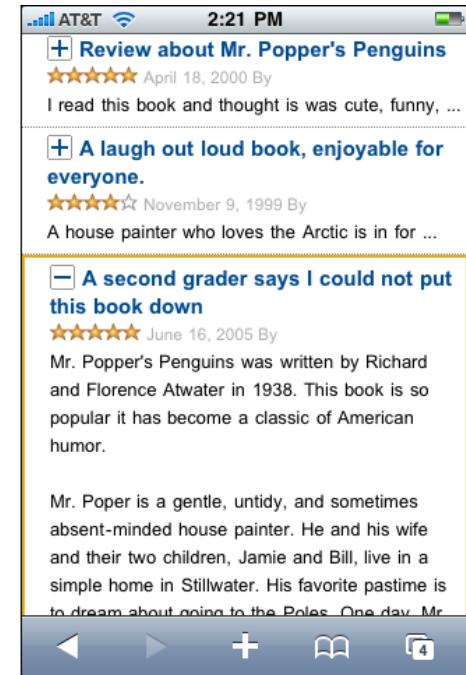
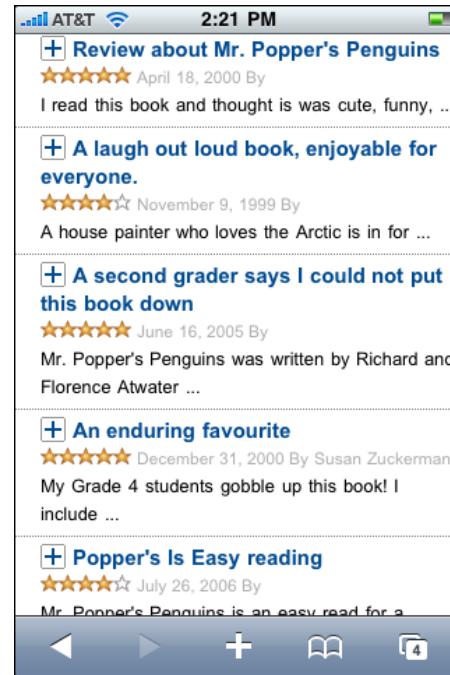
- Display a list of items as a column
- When the user selects an item, show the item details in place within the list
- The user can independently open and close item details

## □ When

- Each item has a small amount of content
- Insufficient space to show all the items' contents

## □ Why

- User can choose what to view



# Lists: Thumbnail Grid

## □ What

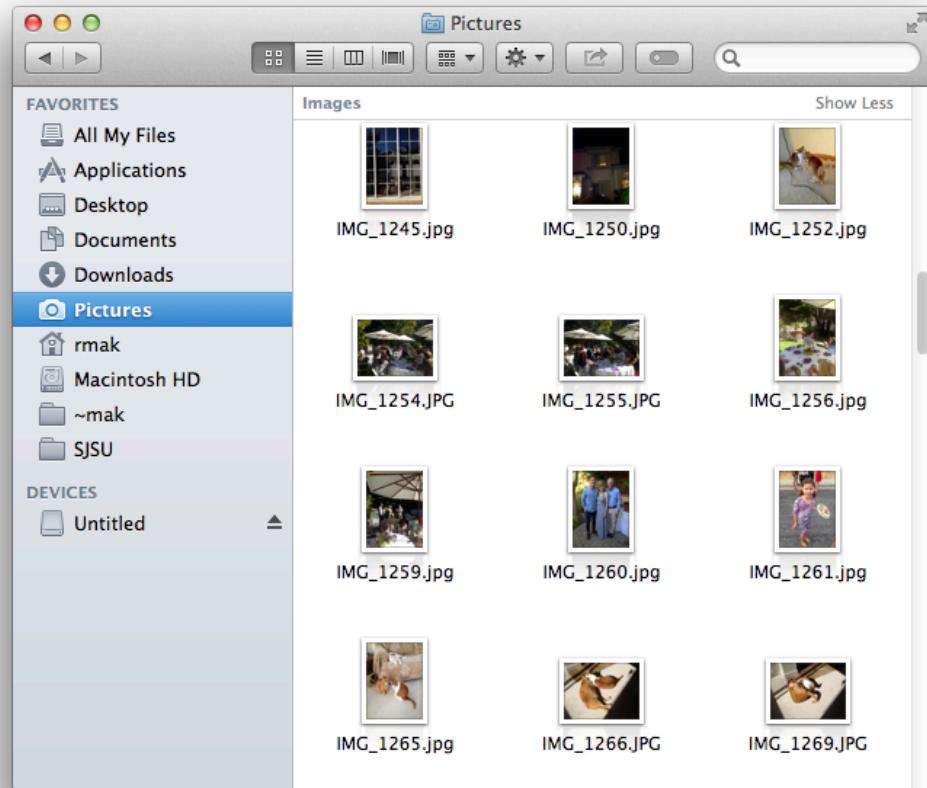
- Arrange a list of items as a grid of thumbnail images
- Each thumbnail is labeled
- Show the larger sized content of a selected item

## □ When

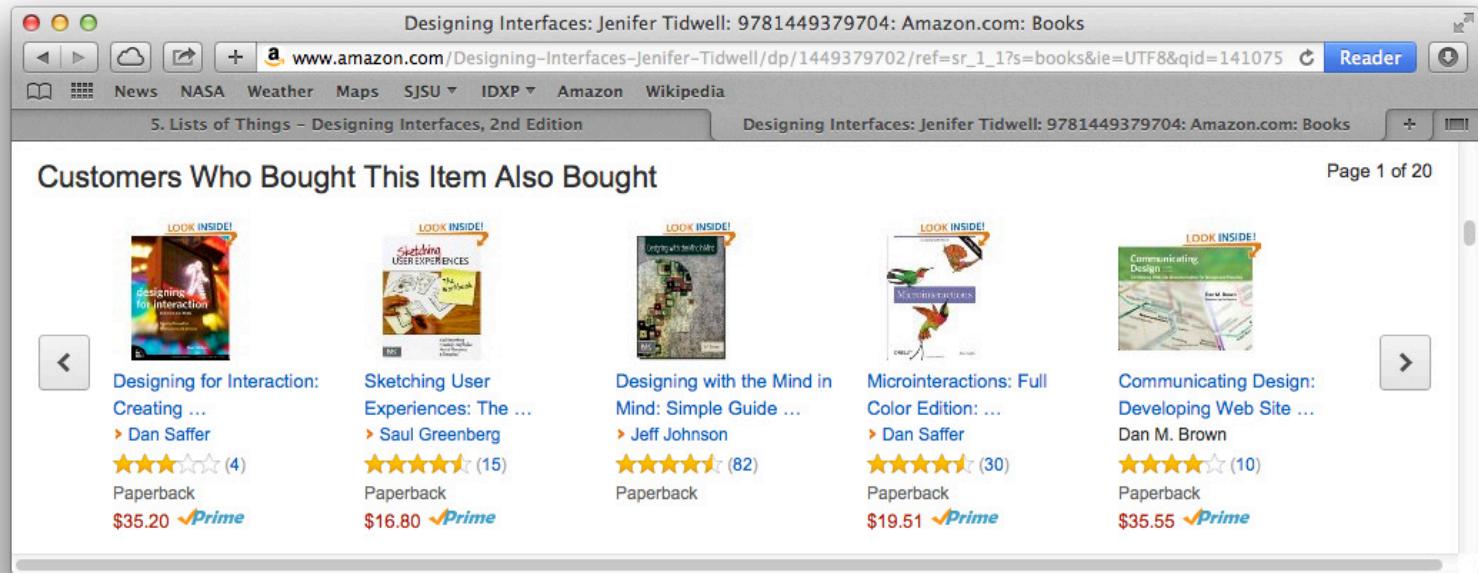
- Items have content representable and recognizable in a smaller format

## □ Why

- Attractive way to display a large list of items
- Thumbnails are easy targets



# Lists: Carousel



- What
  - List items arranged in a scrollable horizontal strip.
- When
  - Insufficient space for a thumbnail grid
- Why
  - Encourage users to scroll and browse.

# Lists: Striped Rows

- What
  - Table rows are striped with alternating colors.
- When
  - Need to group rows
- Why
  - Easy to see groups
  - Easy for a user's eyes to track row contents from left to right



A screenshot of Microsoft Excel showing a table titled "scores.xls". The table has three columns: A, B, and C. The rows are colored alternately: light blue for odd-numbered rows (1, 3, 5, 7, 9, 11, 13, 15, 17, 19) and white for even-numbered rows (2, 4, 6, 8, 10, 12, 14, 16, 18, 20). The data in column A includes names like Dhamaal, Invincibles, SNAP, SPTX, TeamAwesome, and various team names. Column B contains the names of individuals, and column C indicates their status as "Graduate". The Excel ribbon at the top shows tabs for Home, Layout, Tables, Charts, SmartArt, and others. The formula bar shows "A22". The status bar at the bottom shows "Normal View" and "Ready".

	A	B	C
1	Dhamaal	Bell, Jarad Allen	Graduate
2	Dhamaal	Khare, Karan	Graduate
3	Dhamaal	LaCross, Ryan Lloyd	Graduate
4	Dhamaal	Lee, Yunkyoung (Stella)	Graduate
5	Invincibles	Kondra, Akhilesh	Graduate
6	Invincibles	Kumar, Vimal	Graduate
7	Invincibles	Makkar, Ismeet Kaur	Graduate
8	Invincibles	Umaranikar, Pushkar Pramod	Graduate
9	SNAP	Baheti, Akshay Rajesh	Graduate
10	SNAP	Choudhary, Sailee Subhash	Graduate
11	SNAP	Kohli, Nupur	Graduate
12	SNAP	Nair, Prijila Pankajakshan	Graduate
13	SPTX	Chou, Peter	Graduate
14	SPTX	Kang, Timothy	Graduate
15	SPTX	Shweta, Shweta	Graduate
16	SPTX	Zhu, Xianghui (Ryan)	Graduate
17	TeamAwesome	Chen, Jie	Graduate
18	TeamAwesome	Le, Hong Nhung Truong	Graduate
19	TeamAwesome	Li, Yu Q	Graduate
20	TeamAwesome	Liu, PingChuan (Patrick)	Graduate
21			

# Lists: Pagination

## □ What

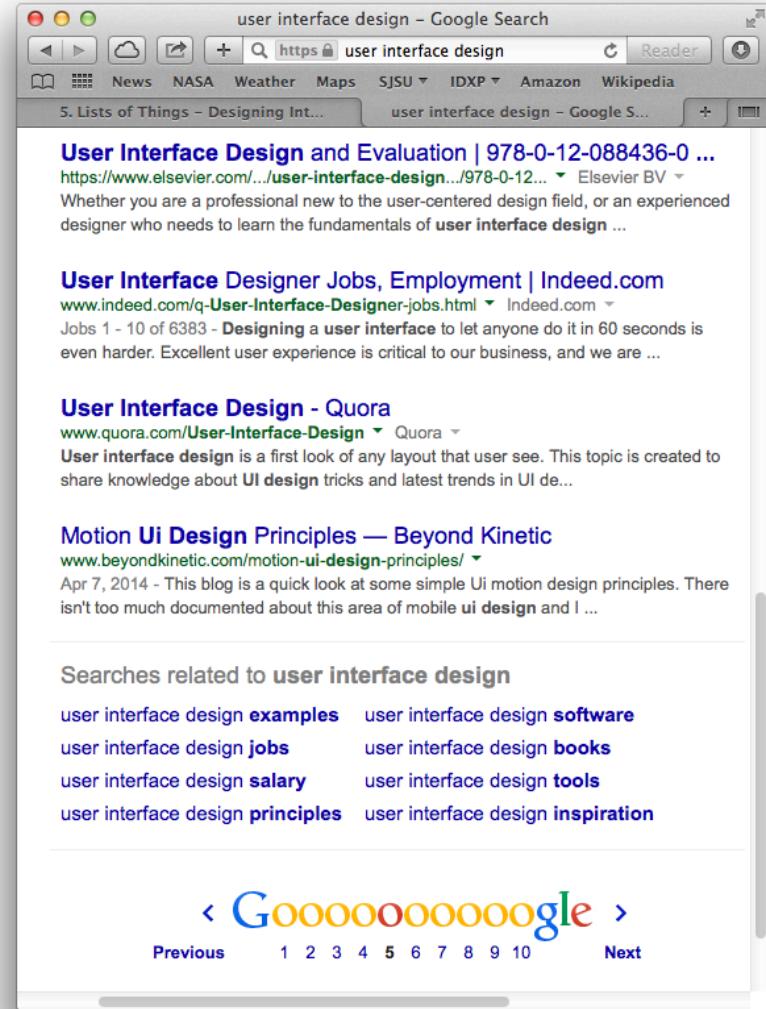
- Break a long list into pages
- Display one page at a time
- Navigation controls: first, last, next, previous

## □ When

- A very long list of items
- Too much time to load and render the entire list all at once

## □ Why

- Break the list into manageable chunks
- The paging metaphor is well known



# Lists: Jump to Item

## □ What

- A scrollable list of text items
- When the user starts to type the text content of an item, jump to that item and select it

## □ When

- A long list of text items
- Only a portion of the list is visible
- Scrolling may be tedious

## □ Why

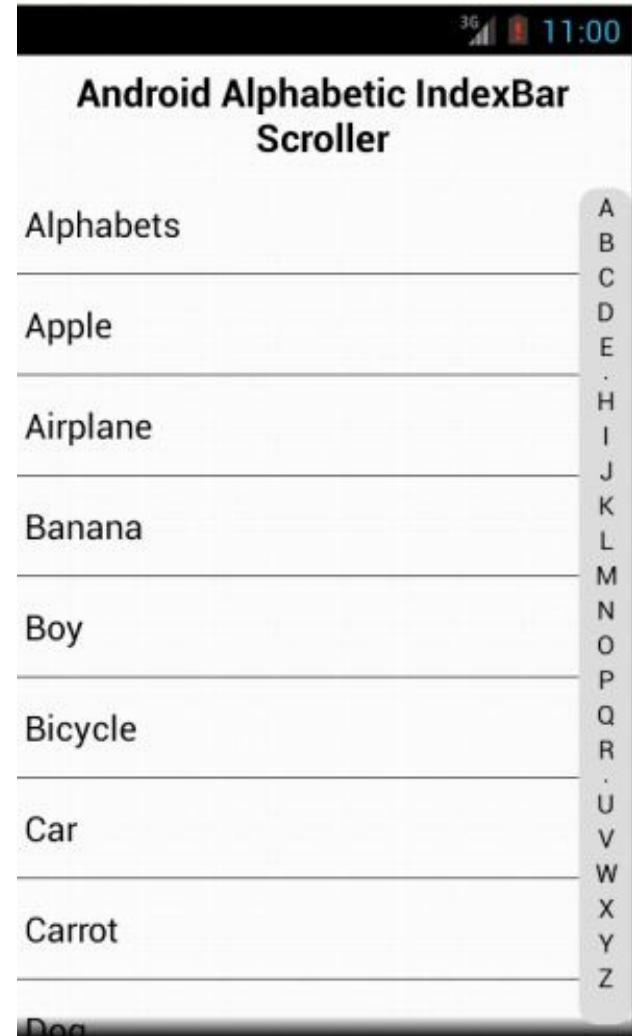
- User already knows what to look for
- Quick and direct access by the user
- Takes advantage of the computer's fast searching



User  
types  
“Bas”

# Lists: Alphabet Scroller

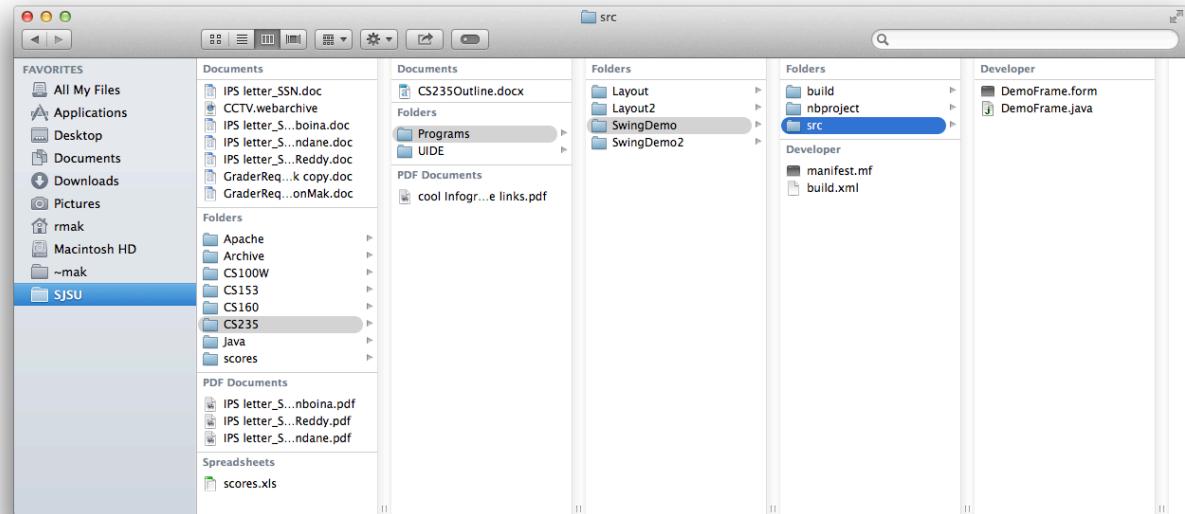
- What
  - An alphabetized list
  - Display the letters of the alphabet
  - Users can select a letter to jump-scroll to the part of the list that begins with the selected letter
- When
  - Users are searching for an item in an alphabetized list
- Why
  - Users don't have to scroll sequentially through the list from the beginning
  - Selecting a letter takes a user close to where the desired item is located



# Lists: Cascading Lists

## □ What

- Display a hierarchy of selectable lists at each level
- Select an item to display that item's children in the next list



## □ When

- A list of items arranged in a possibly deep hierarchy

## □ Why

- Display more of the hierarchy by spreading it out over several lists
- The user can more easily keep track of which level she's on

# Lists: Tree Table

## □ What

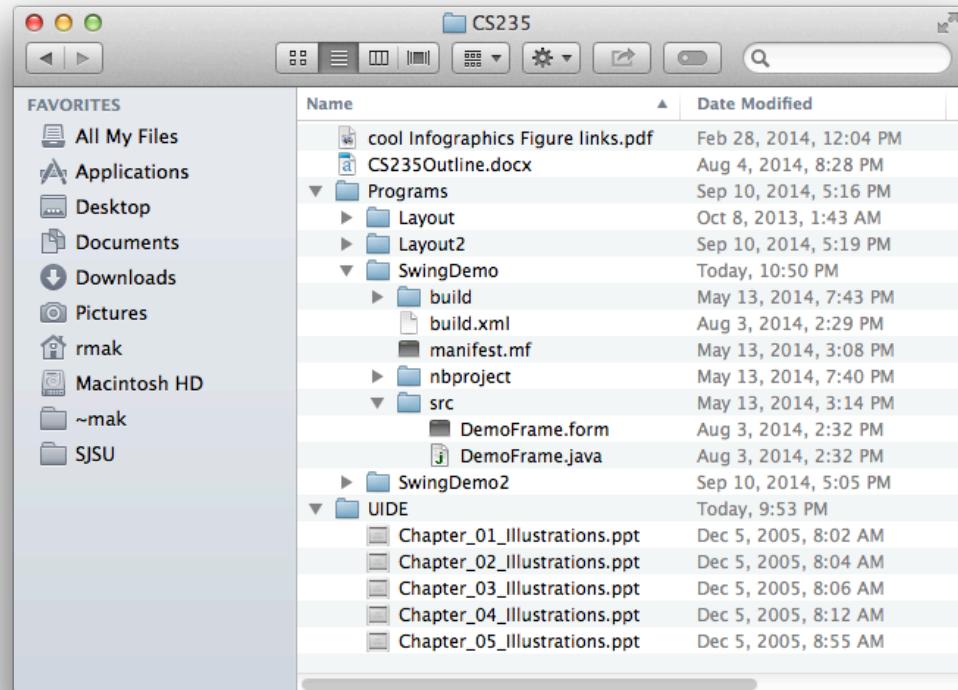
- Display items in a single column
- Use an outline structure to show hierarchy
- Users can expand or collapse items independently

## □ When

- A list of items arranged in a possibly deep hierarchy

## □ Why

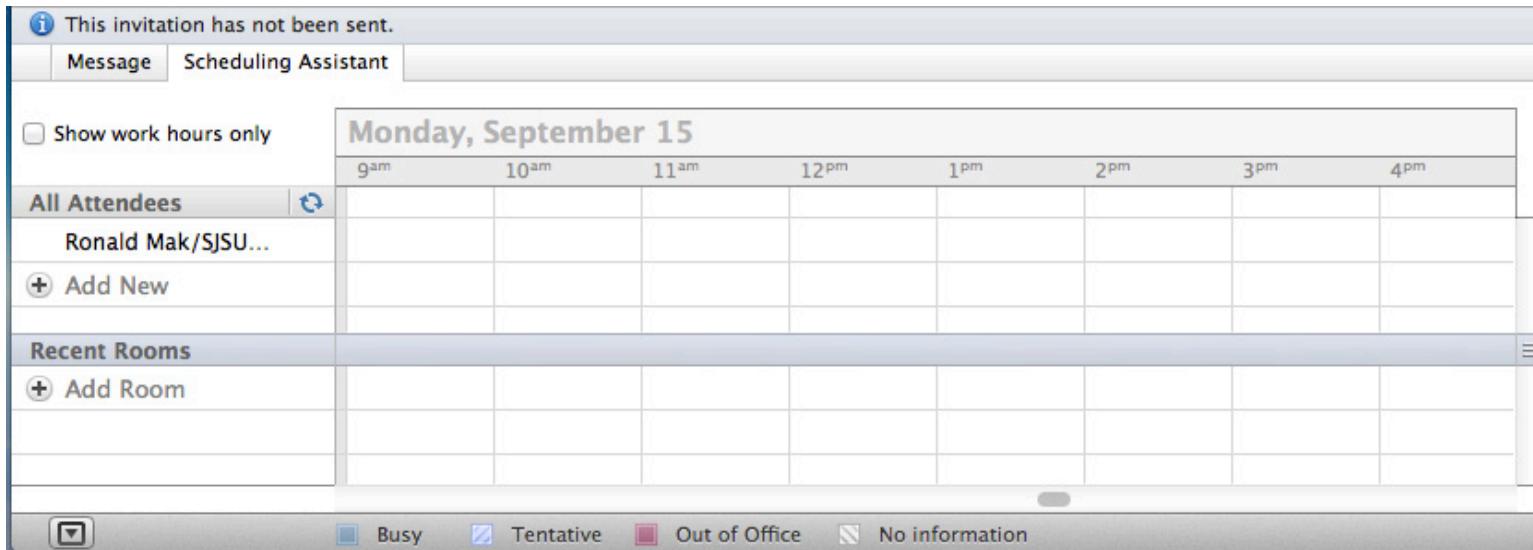
- Users can choose whether or not to view an item's descendants and to what level



A screenshot of a Mac OS X Finder window titled "CS235". The window shows a hierarchical list of files and folders. On the left is a sidebar with "FAVORITES" containing links to "All My Files", "Applications", "Desktop", "Documents", "Downloads", "Pictures", "rmak", "Macintosh HD", "~mak", and "SJSU". The main pane displays a table with three columns: "Name", "Date Modified", and "Type". The "Name" column lists files and folders such as "cool Infographics Figure links.pdf", "CS235Outline.docx", "Programs", "Layout", "Layout2", "SwingDemo", "build", "build.xml", "manifest.mf", "nbproject", "src", "DemoFrame.form", "DemoFrame.java", "SwingDemo2", "UIDE", and several ".ppt" files. The "Date Modified" column shows the last modified date for each item. The "Type" column indicates file types like PDF, DOCX, and Java files.

Name	Date Modified	Type
cool Infographics Figure links.pdf	Feb 28, 2014, 12:04 PM	PDF
CS235Outline.docx	Aug 4, 2014, 8:28 PM	DOCX
Programs	Sep 10, 2014, 5:16 PM	Folder
Layout	Oct 8, 2013, 1:43 AM	Folder
Layout2	Sep 10, 2014, 5:19 PM	Folder
SwingDemo	Today, 10:50 PM	Folder
build	May 13, 2014, 7:43 PM	Folder
build.xml	Aug 3, 2014, 2:29 PM	XML
manifest.mf	May 13, 2014, 3:08 PM	MF
nbproject	May 13, 2014, 7:40 PM	Folder
src	May 13, 2014, 3:14 PM	Folder
DemoFrame.form	Aug 3, 2014, 2:32 PM	Form
DemoFrame.java	Aug 3, 2014, 2:32 PM	Java
SwingDemo2	Sep 10, 2014, 5:05 PM	Folder
UIDE	Today, 9:53 PM	Folder
Chapter_01_Illustrations.ppt	Dec 5, 2005, 8:02 AM	PPT
Chapter_02_Illustrations.ppt	Dec 5, 2005, 8:04 AM	PPT
Chapter_03_Illustrations.ppt	Dec 5, 2005, 8:06 AM	PPT
Chapter_04_Illustrations.ppt	Dec 5, 2005, 8:12 AM	PPT
Chapter_05_Illustrations.ppt	Dec 5, 2005, 8:55 AM	PPT

# Lists: New-Item Row



- What
  - Use the last item of a list to create a new item in place
- When
  - Users need to create new items at the end of a list
- Why
  - Add the new item at its final position in the list.

Lecture #06

# User Interface Requirements in the Real World

*Experiences and Lessons Learned*

Robert Nicholson

[bob-n@wygk.com](mailto:bob-n@wygk.com)

# “Requirements” Mix

- User Interface
  - Design, Interface Elements, etc
- User Experience
  - Data Model, Process (Context)
- Specific Functionality
- Use Cases
- Devices & Platforms
- Performance

# “Requirements” Phases

- Pre-Project:
  - Research Requirements from scratch
  - RFP (Request For Proposal) \*
  - Marketing Requirements
- Project Initiation
  - Requirements Gathering / Refining
- In-Progress Project Review(s)
  - Change Requirements
- Web / Desktop Applications / Mobile Apps
  - Different Release Cycles

# Why Requirements are WRONG (1)

- Wrong People
  - Managers, administrators, executives
  - Limited understanding of the problem
  - No UI / UX expertise (and haven't seen this talk!)
- Mix of People \*
  - Different goals
  - Lack of priorities and process

# Why Requirements are WRONG (2)

- Wrong Problem
  - Focus on “Pain Points” rather than business priorities
  - Focus on legacy systems rather than future
    - (There are always legacy systems)

# Why Requirements are WRONG (3)

- Copying Other Applications
  - Often not appropriate
    - Example: selection spinner
    - Interface Pizza
    - Backward-looking (legacy and technology\*)

# Why Requirements are WRONG (4)

- Lack of Technology / Industry Knowledge
  - (Not knowing what is possible)
  - Geolocation
  - Image recognition
  - Audio Input
  - Language translation
  - Expert Systems / artificial intelligence
  - Back-end database verification services

# Getting Good Requirements (1)

1. Understand the Basics:
  - Use Questionnaires or Interviews
  - Likes and Dislikes (especially useful for UI)
  - Colors and fonts (preferences, company standards)
  - “Mood” (professional, efficient, fun)
  - Language(s)
  - Target Users (age, gender, education, training)
2. Review Documentation and Training Materials
3. Engage Actual Users (understand workflow, but *keep priorities in mind*)
4. Observe the System End-to-End
5. Question, Question, Question (Why?)

# Getting Good Requirements (2)

- Write (or re-write) Requirements
  - Create Use Cases
  - Validate with Users *and* Decision Makers
- Build prototype (wireframe tools, prototyping tools, RAD tools, web) and validate
  - May require multiple iterations
- Actual User Testing, A/B Testing
- Plan for Documentation, Help, Messages and Training
- UI Transition Plan
  - Leverage Legacy Learning
  - Some Users Will Resist Change
  - Incremental Change Sucks!
- Bottom Line: *Redevelop the Requirements*

# Pre-Project Requirements

- Need to Commit Based on Bad Requirements
- Minimize the Risk:
  - Make a *Conditional Commitment*
  - Specify Requirements Gathering Phase
    - Require access to users and systems
    - Allow Time and Budget for Changes
- May Cost You Jobs!  
(But may get you better jobs)
  - Filter out problem clients

# Getting Buy-In

- You are asking your clients to “buy in” to investing more time and \$\$\$ in requirements
- Educate your clients:
  - Present UI design principles
  - Identify the information you need
  - Explain where the requirements fall short
- A well-written handout (with citations) on the process can help establish your credibility

# Requirements Management

- Requirements evolve in the course of the project
- Need to control and limit the changes
  - Requires People Management / Project Management
- Insist on a Single Authoritative Contact
  - Assemble input from multiple people
  - May not be decision maker, but must have *direct access to decision maker*
  - You still need *access* to actual users
- Put Everything in Writing
  - Meeting minutes
- Get Everything in Writing (including approvals)
- Timetable for Requirements Review by Client

# Summary

- Take Charge of Requirements
- Have a *Plan* for Requirements
  - Determine business priorities
  - Survey basic requirements
  - Review documentation
  - engage users
  - view end-to-end system
  - incorporate knowledge of IU/UX technology & best practices
- *Inform Client of Need for Interface Review & Update*
- Schedule and Budget for mid-project Interface Review(s)
- Plan for Documentation and Training
- Plan for Interface Transition / Rollout

# Working with Graphic Designers

*Experiences and Lessons Learned*

Robert Nicholson

[bob-n@wygk.com](mailto:bob-n@wygk.com)

# Design is Important

- Graphic Design is Critical to Success
  - Especially in Consumer Applications
- You are *not* a Graphic Designer
  - Designers spend years studying color theory, layout, typography, iconography, graphic development tools, etc.
- Design Fashions and Styles change
  - Magazines: by decade; web: by year
  - Current: Infinite pages, video backgrounds

# Trust Your Designer

- Set Individual Preferences Aside
- Choose a Designer based on review of past work
- Make sure Designer understands requirements
  - Provide wireframes and list of screen types
- Tell designer what you need
  - Unflattened Photoshop files, sized icons, font and color specifications, CSS files, etc.
- Get early designs and refine
- Incorporate graphic design in prototypes
- As far as possible, isolate design from code (e.g. css, WordPress themes)

Lecture #07

# CS 235: User Interface Design

## February 12 Class Meeting

---

Department of Computer Science  
San Jose State University

Spring 2015  
Instructor: Ron Mak

[www.cs.sjsu.edu/~mak](http://www.cs.sjsu.edu/~mak)



# Your Application Prototype

---

- Each team should build a working prototype of its application.
  - Concentrate on designing a **good user interface**.
  - Use UI design patterns.
- The prototype doesn't have to really work.
  - "Fake it" enough so that looks realistic.
  - Hard-code the interactions.
  - OK for some execution paths not to work at all.
- Make at least **two of your use cases** "work".
  - Concentrate on designing a **good user experience**.

# Your Application Prototype, *cont'd*

---

- Bob Nicholson's favorite **wireframe tool**:
  - Balsamic Mockups
  - <https://balsamiq.com>
- You will perform **usability testing** with your application prototypes.
  - Determine how good your UI and UX designs are.
- We will study how to perform usability testing starting next week.

# Organization Design Patterns

---

- Feature, Search, and Browse
- News Stream
- Picture Manager
- Dashboard
- Canvas + Palette
  - Wizard
  - Settings Editor
  - Alternate Views
  - Multiple Workspaces
  - Multilevel Help

# Navigation Design Patterns

- Clear Entry Points
- Hub and Spoke
- Fully Connected
- Multilevel
- Sequential
- Pyramid
- Pan and Zoom
- Modal Dialog
- Escape Hatch
- Bookmark

# Page Layout Design Patterns

- Visual Framework
- Center Stage
- Grid of Equals
- Tiled Sections
- Module Tabs
- Collapsible Panel
- Movable Panels
- Left-right Alignment
- Responsive Enabling
- Self-adjusting Layout

# List Design Patterns

---

- Two-panel Selector
- One-Window Drilldown
- List Inlay
- Thumbnail Grid
- Carousel
- Striped Rows
- Pagination
- Jump to Item
- Alphabet Scroller
- Cascading Lists
- Tree Table
- New-Item Row

# Action Controls

---

- Buttons
- Menu bars
- Pop-up menus
- Drop-down menus
- Toolbars
- Links
- Action panels
- Hover tools
- Double-clicking
- Keyboard actions
- Drag-and-drop
- Typed commands

# Action Design Patterns

---

- Ways to present action.
  - Indicate progress of actions.
  - Give users a sense of closure.
  - Allow users to preview actions.
  - Allow users to cancel actions.
  - Allow users to undo actions.
- 



# Action: Button Groups

## □ What

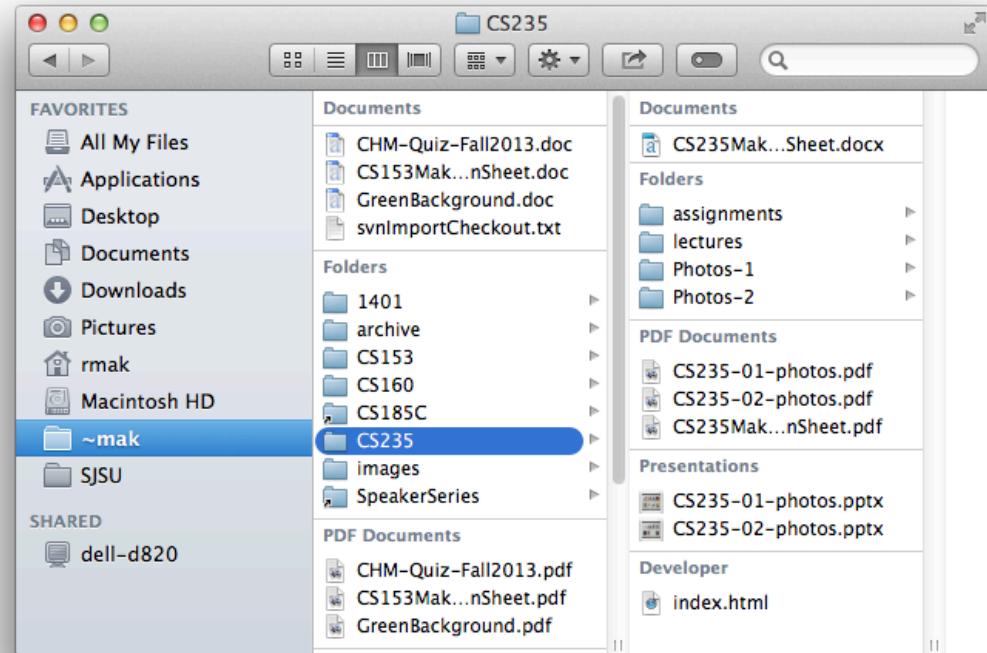
- Group related actions
- Multiple groups for multiple sets of actions

## □ When

- Many possible actions
- Related sets of actions

## □ Why

- Gestalt principles
- Actions that can take place in a given context
- Button clusters easy to pick out among in a complex layout



# Action: Hover Tools

## □ What

- Place controls next to the item they act upon
- Hide or disable the controls until the user hovers the mouse over the item

**0 Concord Rd. Lot 1, Weston, MA**

House For Sale: \$950,000  
Monthly Payment: \$3,788  
Beds: -- Sqft: --  
Baths: -- Lot: --

► See current rates on Zillow  
Days on Zillow: 66  
Built: --

**21 Crescent St, Weston, MA**

House For Sale: \$1,695,000  
Monthly Payment: \$6,758  
Beds: 5 Sqft: 3,686  
Baths: 3.5 Lot: 68,712

▼ \$100,000 (Jul 21)  
► See current rates on Zillow  
Days on Zillow: 109  
Built: 1812

**48 Myles Standish Rd, Weston, MA**

House For Sale: \$3,495,000  
Monthly Payment: \$13,936  
Beds: 5 Sqft: 10,600  
Baths: 6.5 Lot: 65,340

► See current rates on Zillow  
Days on Zillow: 138  
Built: 2006

## □ When

- Many possible actions
- Keep user interface as uncluttered as possible

## □ Why

- Show or enable only relevant controls

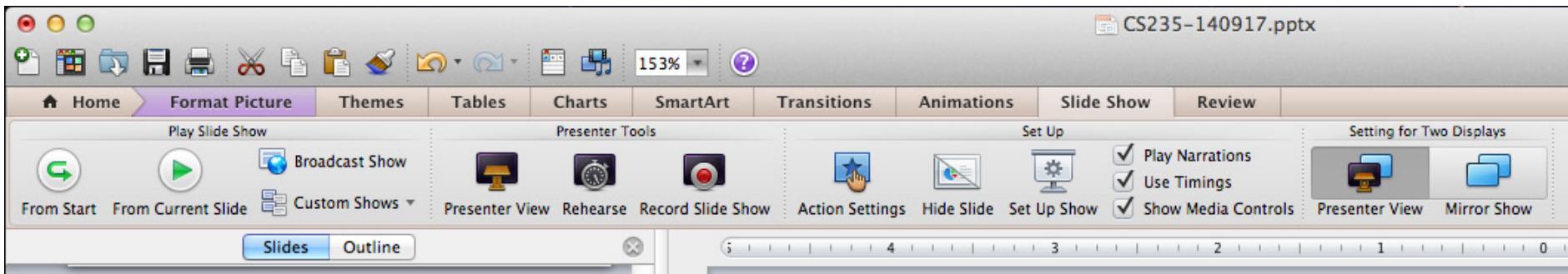


Computer Science Dept.  
Spring 2015: February 12

CS 235: User Interface Design  
© R. Mak

Designing Interfaces, 2<sup>nd</sup> ed.  
by Jenifer Tidwell  
O'Reilly Media, 2011

# Action: Action Panel



## □ What

- A panel of related controls that is richly organized and always visible

## □ When

- The controls need to be visible always
- Sufficient space to display the controls

## □ Why

- Users will always know what actions are available

# Action: Prominent “Done” Button

## □ What

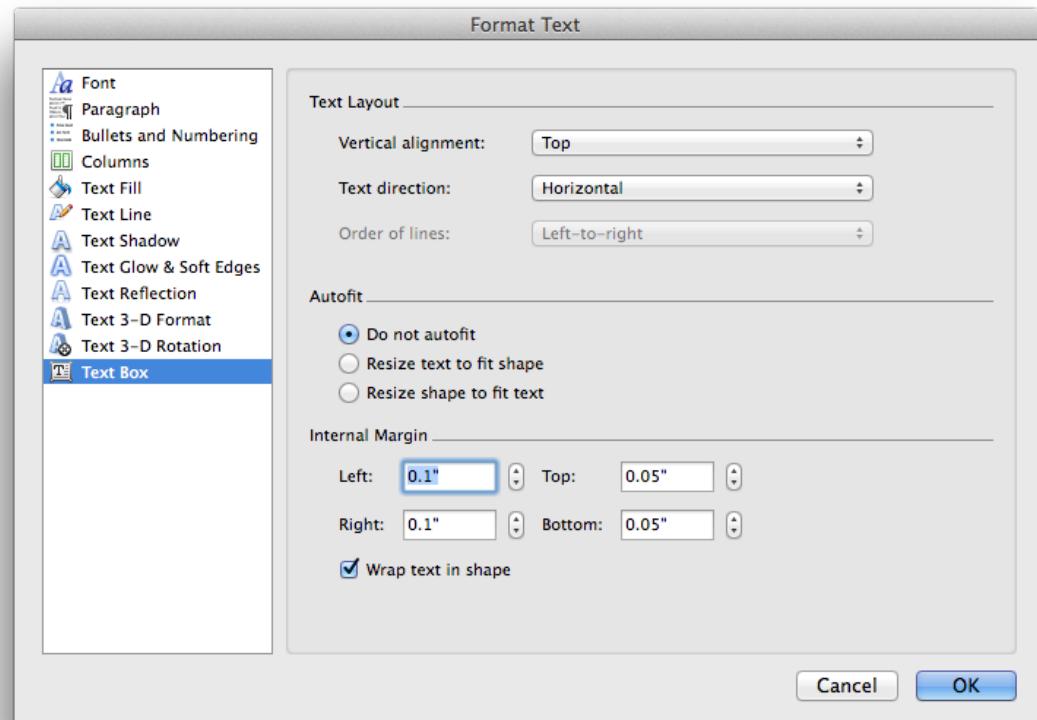
- Place the button that completes an action prominently at the end of the visual flow

## □ When

- Whenever you need a Done, Submit, OK, or Continue button

## □ Why

- Give the user a sense of closure
- The user knows that an action was done



# Action: Smart Menu Items

## □ What

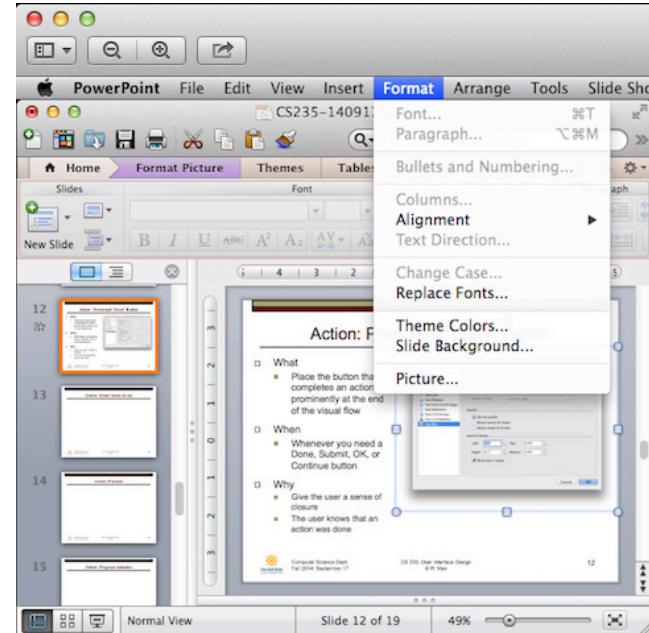
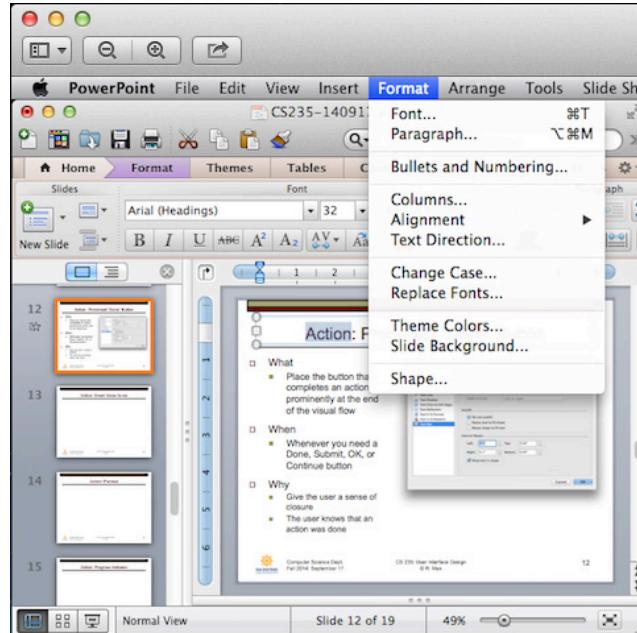
- Dynamically change or disable menu items

## □ When

- Different actions for different contexts

## □ Why

- Only show actions that are relevant for a given context, such as a user selection



# Action: Preview

## □ What

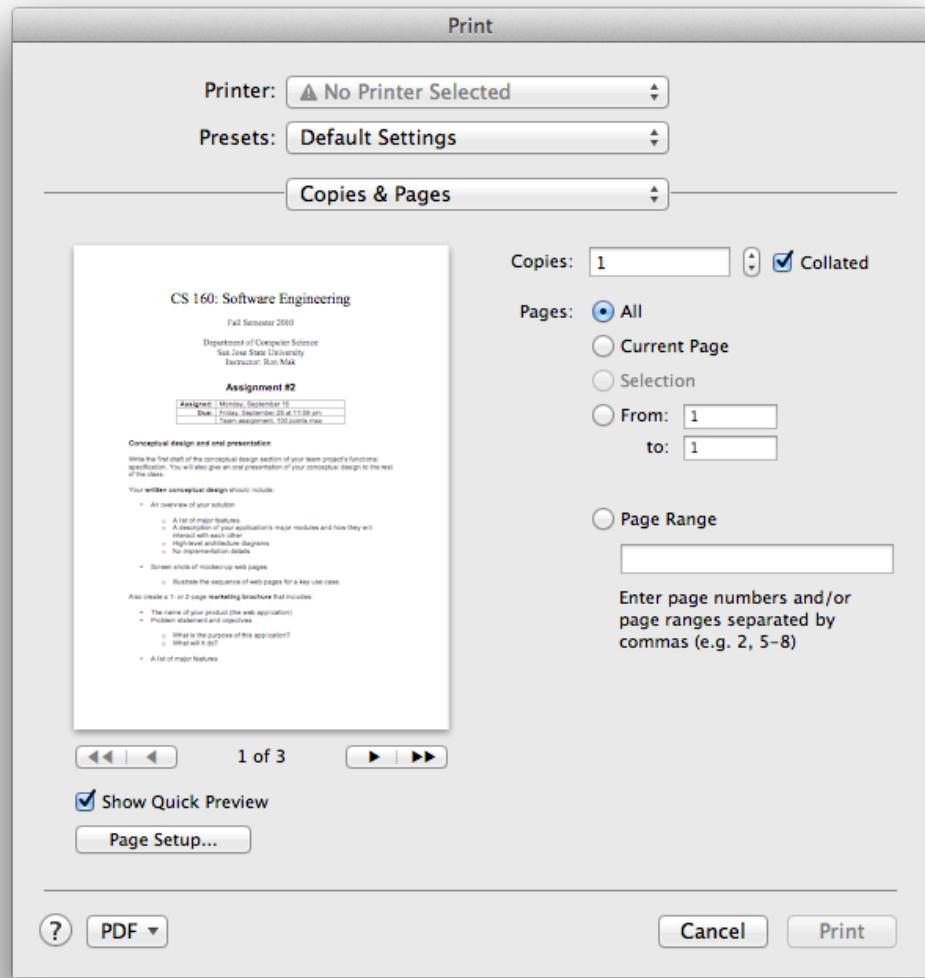
- Show a preview or summary of the results of a pending action

## □ When

- The user is about to perform an action that will produce results

## □ Why

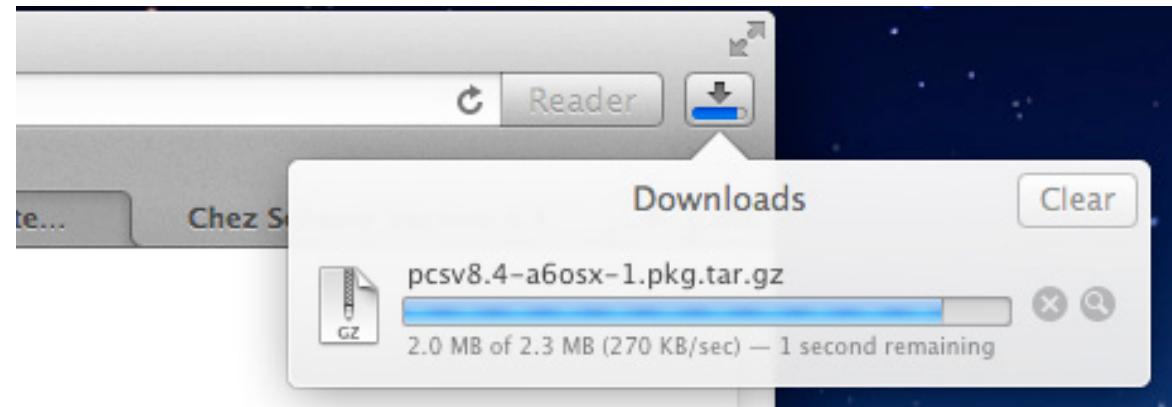
- Assure the user that the results will be correct
- Help prevent errors



# Action: Progress Indicator

## □ What

- Show how much progress a time-consuming action has made



## □ When

- A time-consuming action runs in the background

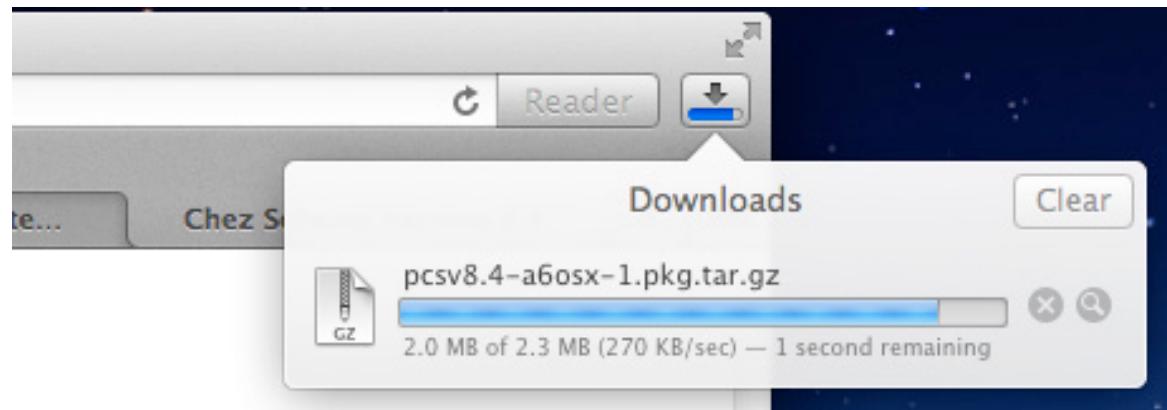
## □ Why

- Assure the user that progress is being made
- Indicate how much work remains

# Action: Cancelability

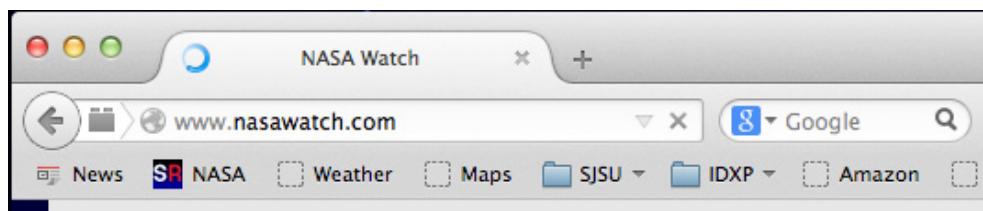
## □ What

- Provide a way to cancel a time-consuming action



## □ When

- A time-consuming action runs in the background

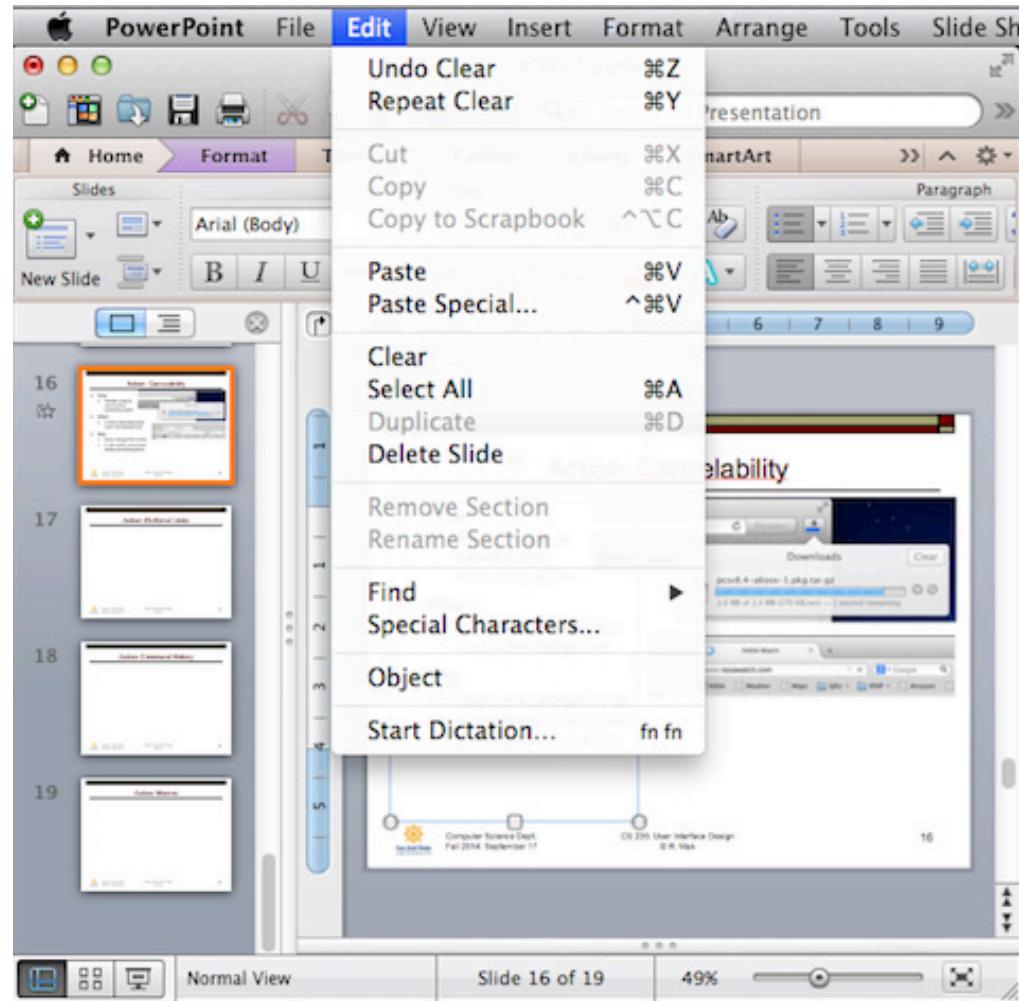


## □ Why

- Users change their minds
- A user wants to terminate a time-consuming action

# Action: Multilevel Undo

- What
  - Allow the user to reverse a sequence of actions
- When
  - A highly interactive interface with many user actions
- Why
  - Provide an interface that is safe to explore
  - Users are confident that errors aren't permanent
  - No need to "checkpoint" data and revert to an earlier version



# Actions Generally Undoable

---

- Text entry
- Drawing changes
- Layout changes
- Creation, deletion, or rearrangement of items
- Cut, copy, or paste operations
- File operations: create, delete, rename
- Database operations



# Actions Generally Not Undoable

---

- Navigation between pages or windows
- Mouse or text cursor movements
- Scrollbar position
- Window or panel positions and sizes
- Changes made in a modal dialog box

# Principles of Getting Input from Users

---

- Make sure the user understands what's asked for and why.
  - If possible, don't ask.
- Give the user a list of options.
- Give meaningful and sensible error messages.
  - Be forgiving.
- Be aware of users' mental models.
- The choice of input controls determine the user's expectations of what is asked for.
- Do lots of **usability testing**.

# Text Input Controls

Single-line text

Parrot

Multiline text

Does anyone here have a border collie? I would really like one, and I have some questions.

Structured text

Oct 17, 2010

Text editor

Does anyone here have a border collie? I would **really** like one, and I have some questions.



# Choice Controls

Checkbox

Yes, I want a talking bird

Radio buttons

- Parrot
- Starling
- Mynah bird

Toggle buttons

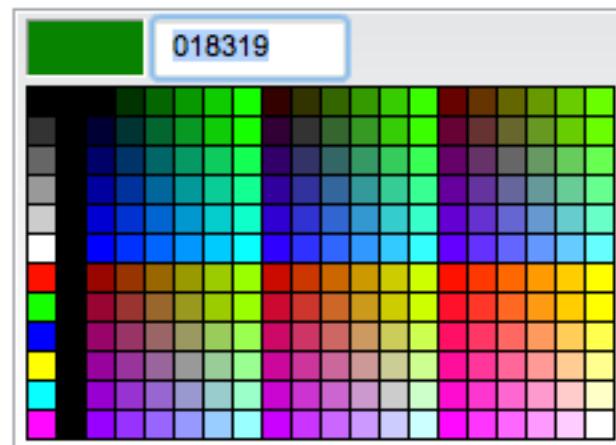
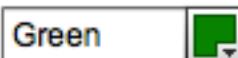
Parrot    Starling    Mynah

Spinner

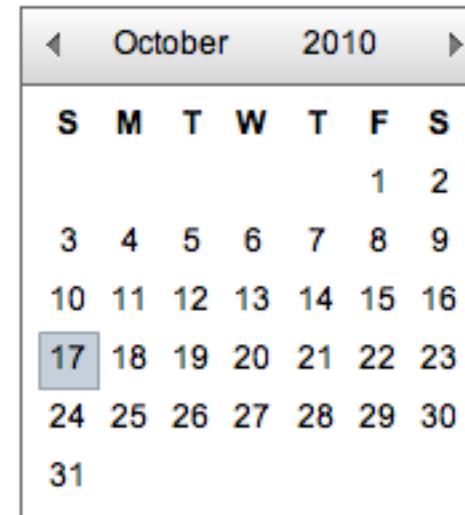
30

Slider

0                          100



Dropdown chooser



Calendar chooser

# Controls for List Item Selection

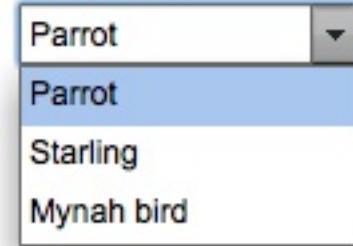
Dropdown list



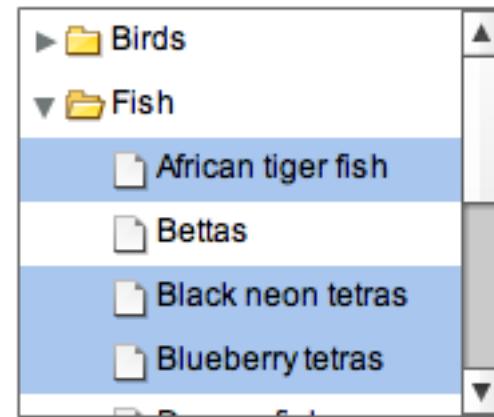
Checkbox list



Combo box

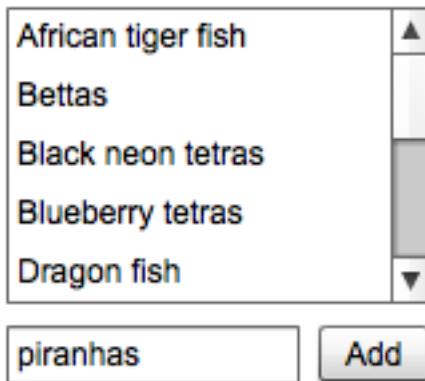


Multiple selection list



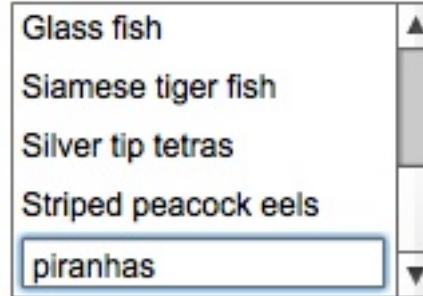
Multiple selection tree

# Controls for Creating Lists



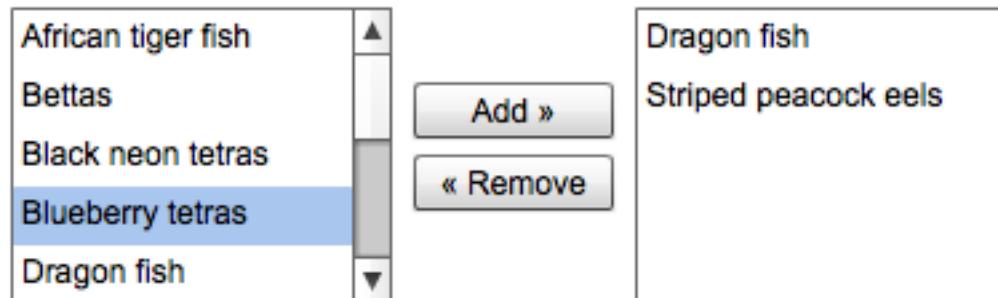
A screenshot of a user interface showing a list of fish names. The list includes: African tiger fish, Bettas, Black neon tetras, Blueberry tetras, Dragon fish, and piranhas. The word 'piranhas' is in a separate input field below the list, with an 'Add' button next to it. A vertical scroll bar is visible on the right side of the list.

List with add button



A screenshot of a user interface showing a list of fish names. The list includes: Glass fish, Siamese tiger fish, Silver tip tetras, Striped peacock eels, and piranhas. The word 'piranhas' is highlighted with a blue selection bar. A vertical scroll bar is visible on the right side of the list.

List with new-item row



A screenshot of a user interface showing a list builder. On the left, a list of fish names (African tiger fish, Bettas, Black neon tetras, Blueberry tetras, Dragon fish) has 'Blueberry tetras' selected. On the right, a panel contains 'Dragon fish' and 'Striped peacock eels'. Between the lists are two buttons: 'Add »' and '« Remove'.

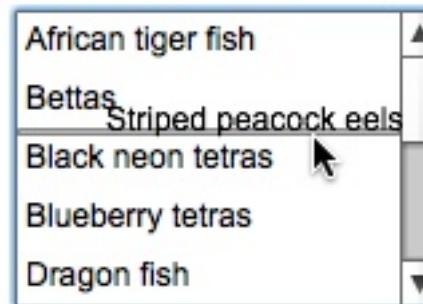
List builder

# Controls for Sorting Lists

List with up-down buttons



List with internal drag-and-drop



# User Input Design Patterns

---

- Allow users to select from choices.
  - Allow users to input text.
- 



# Input: Forgiving Format

## □ What

- An input text field that allows a variety of input formats and syntax

*Search Zip, City, or Place (Disney World)* 

## □ When

- The user needs to enter information that can be typed in a variety of ways

## □ Why

- Users are unpredictable
- Friendlier for users



# Input: Structured Format

## □ What

- A set of text input fields that reflect the structure of the requested data

Credit card number:

1021	1234	5678	0000
------	------	------	------

## □ When

- Input data must have a fixed structure

## □ Why

- Provide a clue to users
- Reduce data entry errors

# Input: Fill-in-the-Blanks



- What
  - A sentence or phrase with one or more “blanks” for the user to fill in
- When
  - User input is in several related parts
- Why
  - Self-explanatory interface
  - Hints to the user

# Input: Tips and Hints

## □ What

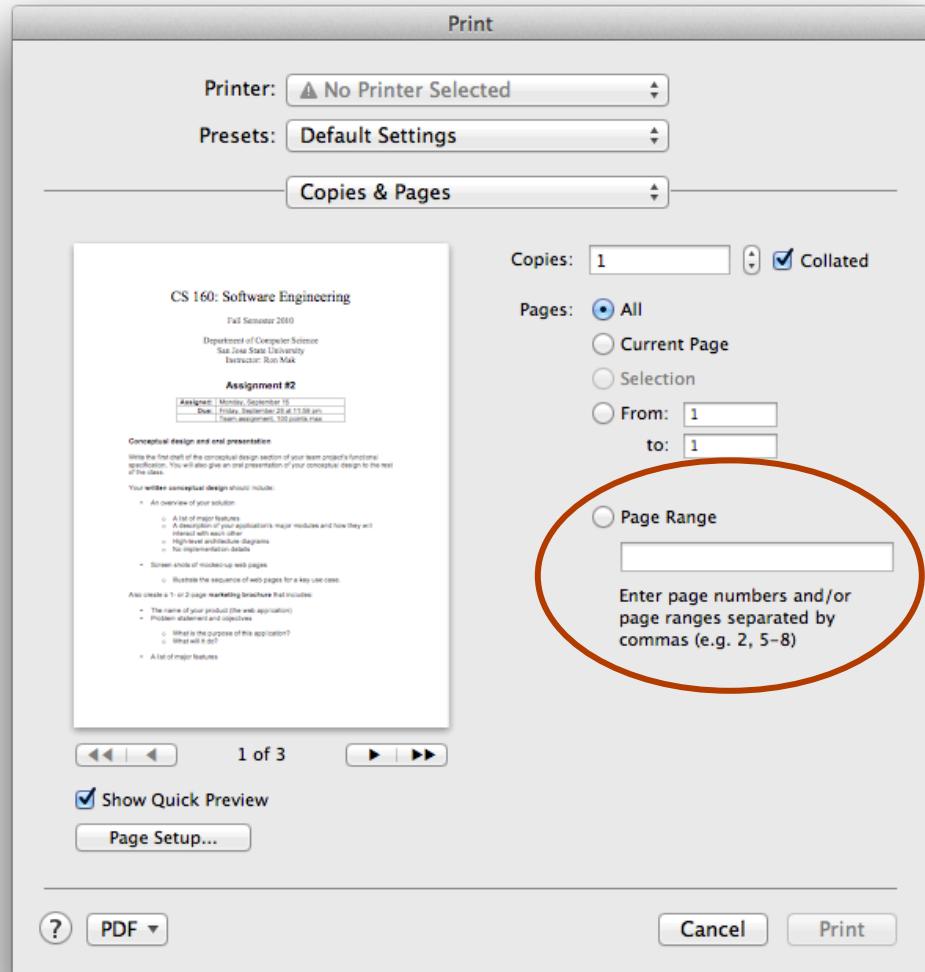
- Explanatory text that provides tips or hints about the desired input

## □ When

- A form with a variety of input fields

## □ Why

- Tell users what input is desired
- Users don't need to consult documentation



# Input: Prompting Text Field

## □ What

- Text input fields pre-filled with user prompts

## □ When

- A form with a variety of input fields

## □ Why

- Tell users what input is desired
- Users don't need to consult documentation

**Sign Up**  
It's free and takes less than 30 seconds.

Your First Name

Your Last Name

|

Password

Confirm Password

I agree to the [Terms of Use](#).

Submit

**Sign Up**  
It's free and takes less than 30 seconds.

Your First Name

Your Last Name

Your Email

Password

Confirm Password

I agree to the [Terms of Use](#).

Submit

# Input: Auto-Completion

## □ What

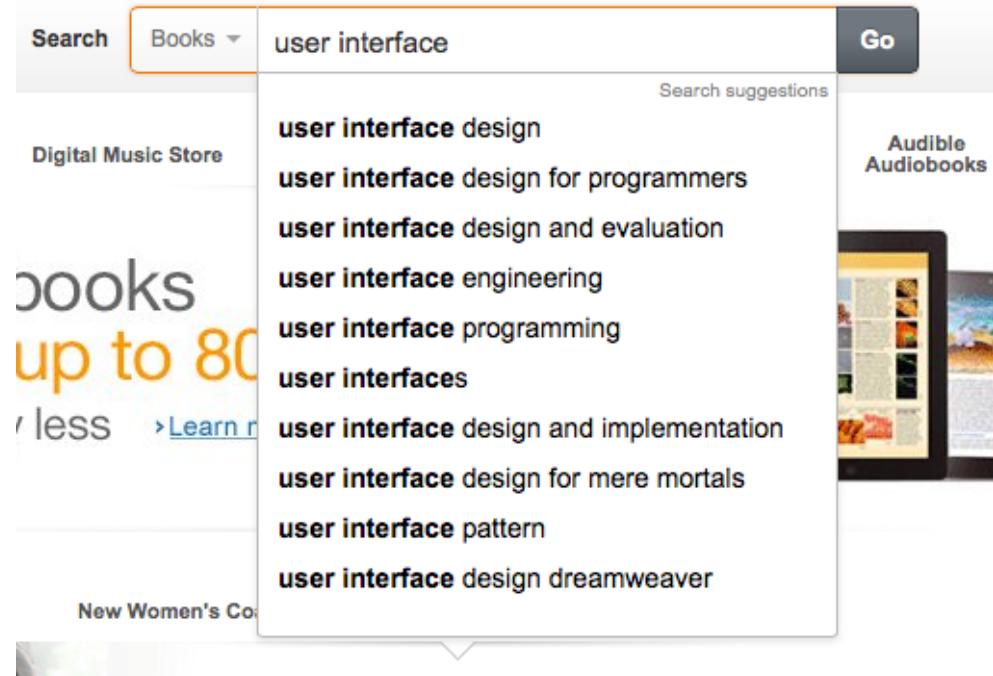
- Anticipate possible input data based on what the user has already typed

## □ When

- A list of possible ways to complete the input

## □ Why

- User-friendly time saver
- Users don't have to remember long input
- Users can reuse previous input



# Input: Dropdown Chooser

## □ What

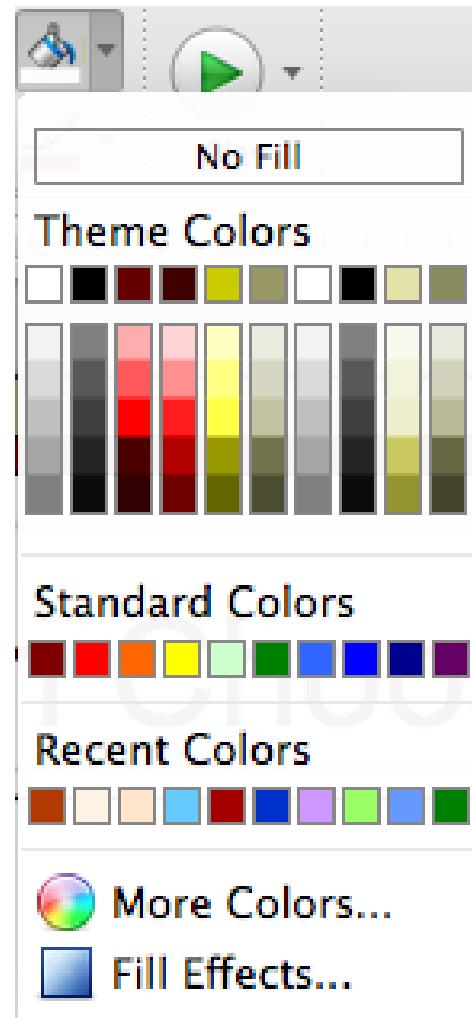
- A dropdown menu with a variety of items
- Different ways to display the items

## □ When

- The user must choose from a set of items

## □ Why

- Users are already familiar with dropdown menus
- Compact way to present menu items
- Display menu items only when needed



# Input: Good Defaults

## □ What

- Pre-fill input fields with likely user input

## □ When

- Reasonable guesses of what the user will enter

## □ Why

- User-friendly time-saver
- Less knowledge required by the user

The form includes radio buttons for trip type: Round-trip (selected), One-way, Multi-city, and Weekend. The 'From' field contains 'BOS' with a note to include nearby airports or select custom. The 'To' field is empty with a similar note. Departure and return dates are set to specific dates in October and November 2010. The number of travelers is set to 1, and the class is Economy. A checkbox for 'Prefer Nonstop' is available. A note at the bottom says 'Compare hundreds of travel sites at once.' and the 'Search' button is prominently displayed.

# Input: List Builder

## □ What

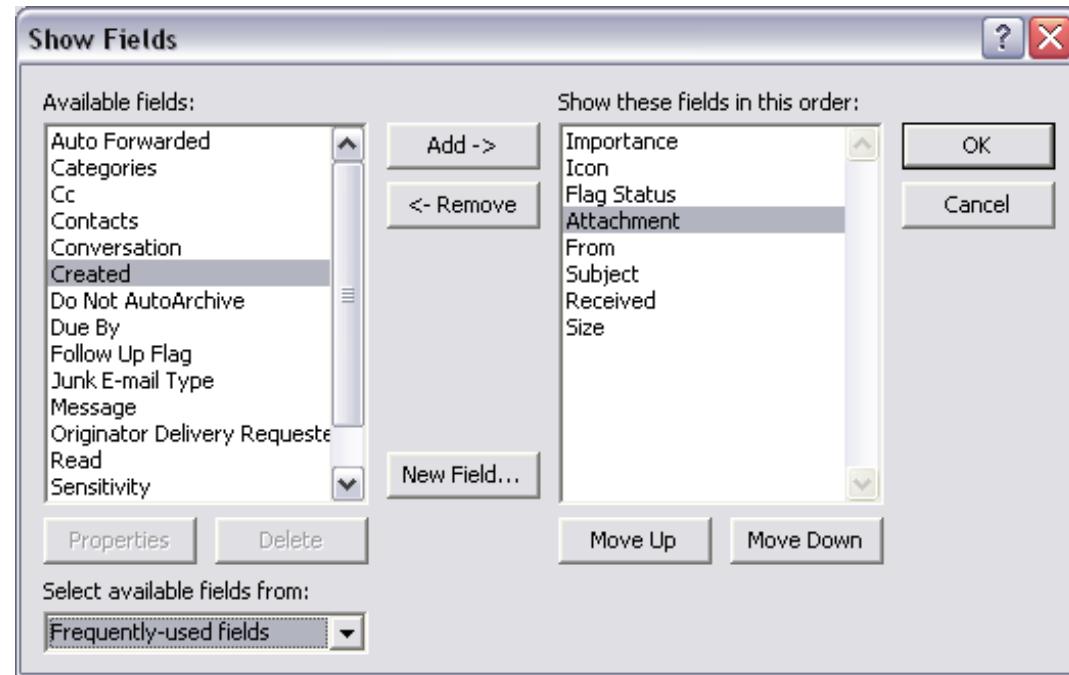
- Source and destination lists
- Add and Remove buttons, or drag-and-drop between the lists

## □ When

- The user must create a list using items from another list

## □ Why

- The user can see the available choices for the destination list
- Clear what the destination list will be



# Input: Same-Page Error Messages

Name Jenifer Tidwell

Gender Female

Birthday January 1 2030 ⚠ Are you really from the future?

Country United States

Postal Code 01234 ⚠ This information is required

- What
  - Display error messages on the same page
  - Display error messages next to the erroneous input
- When
  - User might enter bad input
- Why
  - Immediate feedback near the erroneous input

Lecture #08

# CS 235: User Interface Design

## February 17 Class Meeting

---

Department of Computer Science  
San Jose State University

Spring 2015  
Instructor: Ron Mak

[www.cs.sjsu.edu/~mak](http://www.cs.sjsu.edu/~mak)



# Your Application Prototype

---

- Each team should build a working prototype of its application.
  - Concentrate on designing a **good user interface**.
  - Use UI design patterns.
- The prototype doesn't have to really work.
  - "Fake it" enough so that looks realistic.
  - Hard-code the interactions.
  - OK for some execution paths not to work at all.
- Make at least **two of your use cases** "work".
  - Concentrate on designing a **good user experience**.

# Your Application Prototype, *cont'd*

---

- Bob Nicholson's favorite **wireframe tool**:
  - Balsamic Mockups
  - <https://balsamiq.com>
- You will perform **usability testing** with your application prototypes.
  - Determine how good your UI and UX designs are.
- We will study how to perform usability testing starting next week.

# Assignment #2: Usability Testing

---

- Each team chooses **two interactive use cases** from its web application prototype.
- Each use case must accomplish a specific task.
  - A significant task for the application.
  - Not simply “log in”.



# Assignment #2: Usability Testing, *cont'd*

---

- During **usability testing**, a student will be chosen to test each team's application.
  - Not a student from the client or design teams.
- The student tester will attempt each use case with **no handholding** from the test facilitator.
  - The test facilitator will be a member of the application's design team.
  - The remaining members of the design team will observe and take copious notes.

# Assignment #2: Usability Testing, *cont'd*

---

- Usability tests will occur February 24 and 26.
  - 20 minutes each.
- The design team will write a  
**Usability Test Report.**
  - What problems were observed?
  - What design fault caused each problem?
  - What improvements can be made?
  - Before and after screen shots.

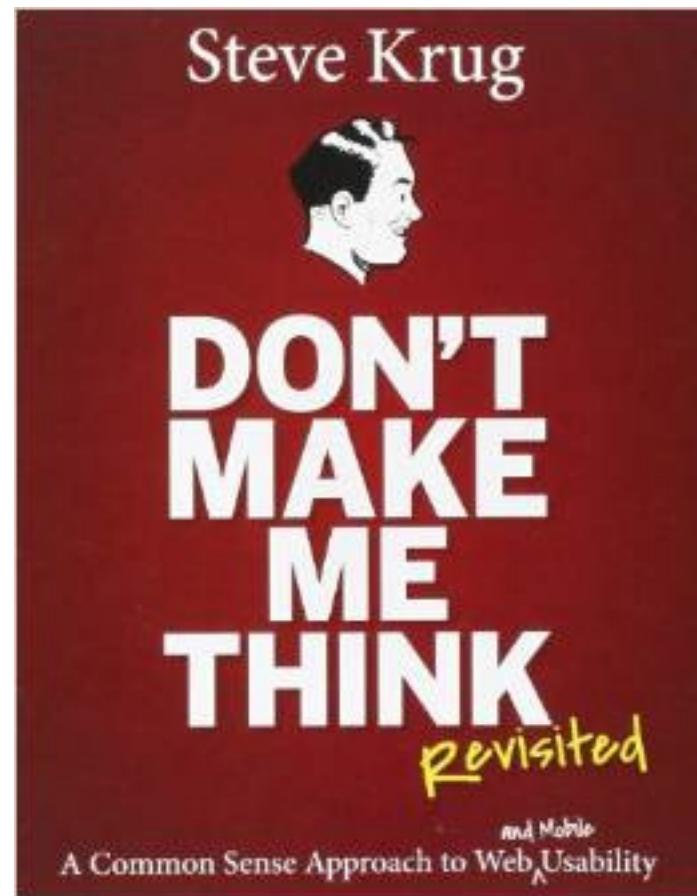
# Assignment #2: Usability Testing, *cont'd*

---

- Usability Test Reports due one week after the test.
  - March 3 and 5.
- Go to this website:  
<http://www.sensible.com/rsme.html>  
and watch the “Demo Usability Test” video.
  - The actual test starts at 6:00 into the video.
    - During the tests we'll do in class,  
you don't need to do the initial chat.
  - Notice how the tester and the test facilitator have a **constant conversation** with each other.

# A Good Book on Web Usability

---



# Definitions of Usability

---

## □ Useful

- Does it do something people need done?

## □ Learnable

- Can people figure out how to use it?

## □ Memorable

- Do they have to relearn it each time they use it?

## □ Effective

- Does it get the job done?

# Definitions of Usability, *cont'd*

---

## □ Efficient

- Does it do it with a reasonable amount of time and effort?

## □ Desirable

- Do people want it?

## □ Delightful

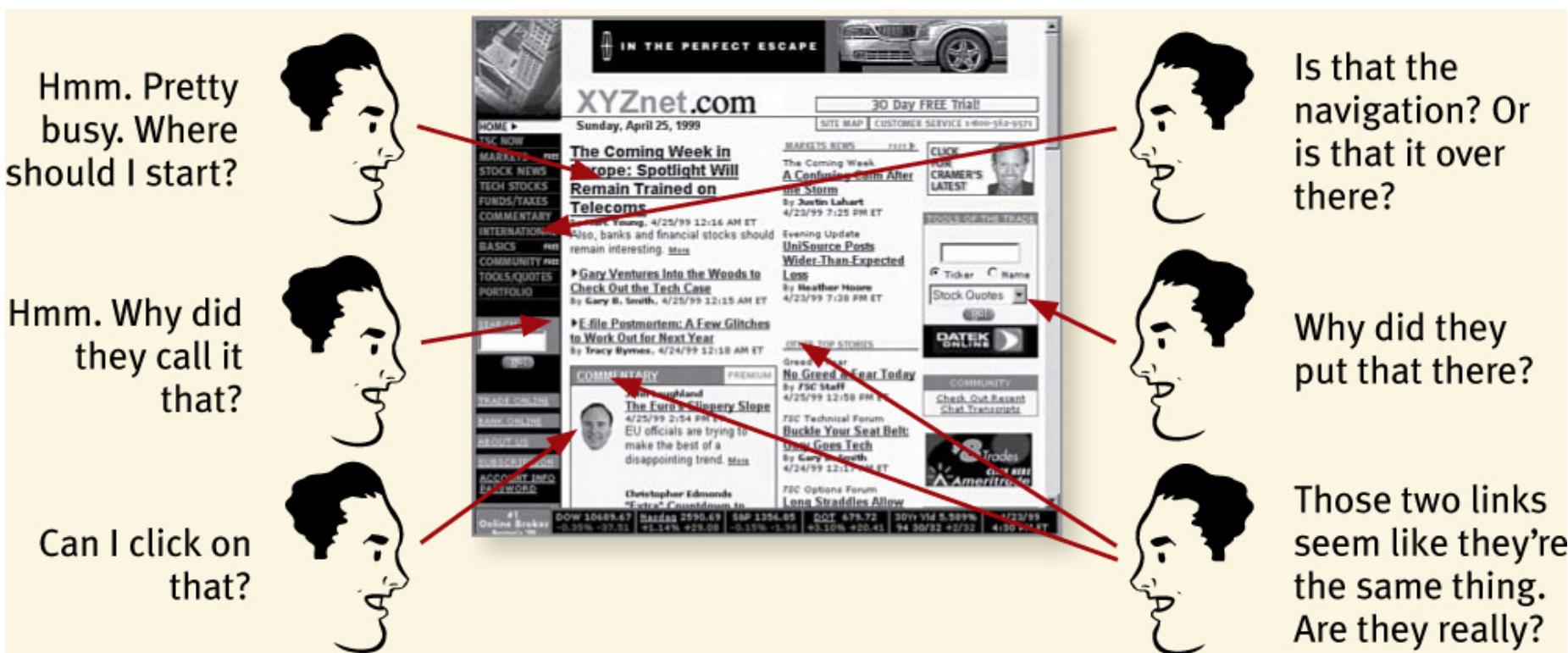
- Is using it enjoyable, or even fun?

“A person of average (or even below average) ability and experience can figure out how to use the thing to accomplish something without it being more trouble than it's worth.”



# Don't Make Me Think

- You don't want users to have to think about your page.



# Don't Make Me Think, *cont'd*

- Things should be immediately obvious.

OK. This looks like the product categories...

Laptops, Memory... There it is: Monitors. **Click**

...and these are today's special deals.

# Don't Make Me Think, *cont'd*

## □ What is clickable?



# Don't Make Me Think, *cont'd*

## □ Booking a flight

<b>FROM</b>	<b>TO</b>	
City or Airport 	>	City or Airport 
Depart Date 		Return Date 



Let's see. "City or Airport."  
I'll put in the city names.

<b>FROM</b>	<b>TO</b>
bos 	City or Airport 
<b>Boston, MA, US (BOS)</b>	Return Date 



*Picks Boston from the dropdown*

<b>FROM</b>	<b>TO</b>
BOS 	
Depart Date 	Return Date 



But why does it just put BOS after I pick Boston?

FROM		TO	
BOS		ny	
12/17/2013		12/19/2013	



I'm sure it'll know "ny"...  
*Types "ny" and fills in dates,  
then clicks "Find Flights"*

<b>Please enter a valid 'TO' City or Airport code.</b>	
<b>FROM</b>	<b>TO</b>
BOS 	hy 
12/17/2013 	12/19/2013 



Why doesn't it recognize  
New York?

# Don't Make Me Think, Revisited

by Steve Krug  
New Riders, 2014

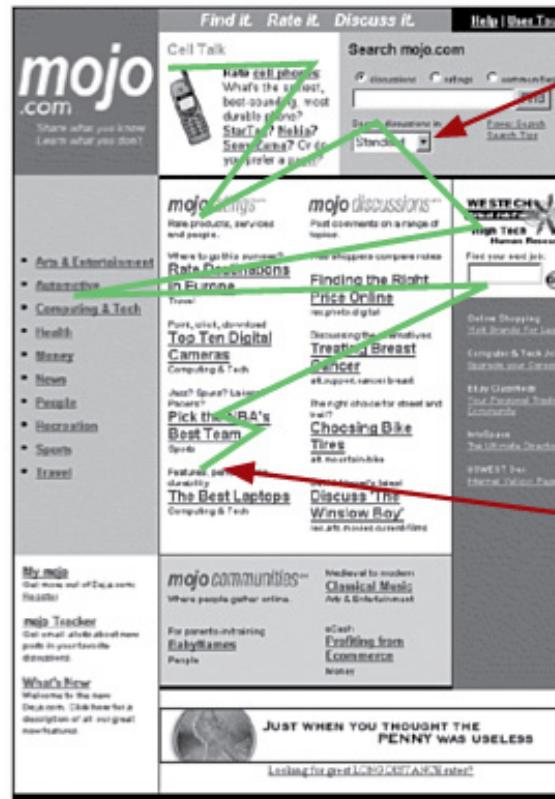
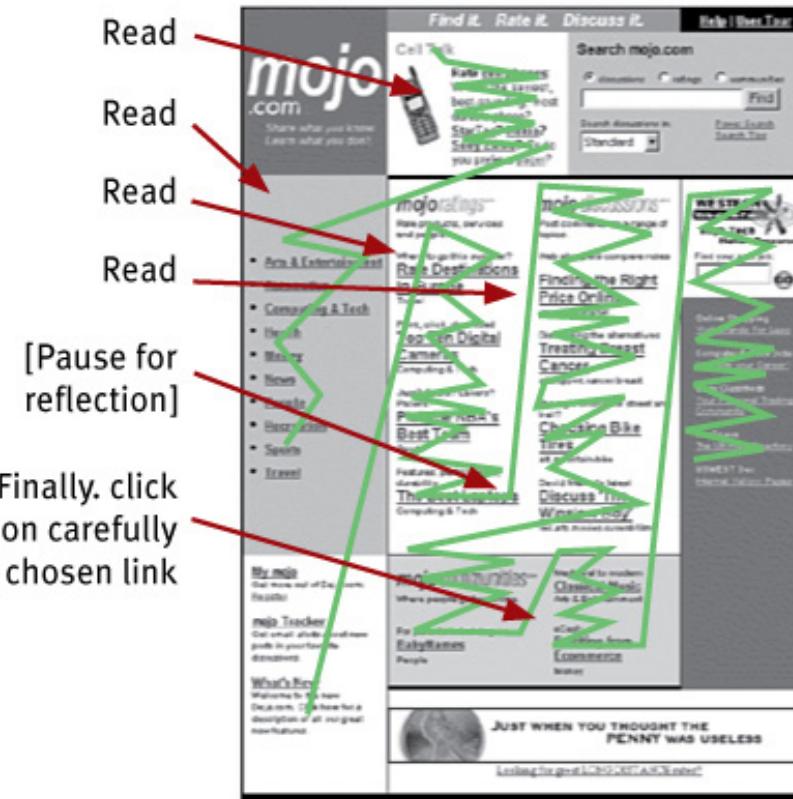


Computer Science Dept.  
Spring 2015: February 17

# Users Don't Read Pages

- Users scan web pages.

## WHAT WE DESIGN FOR... THE REALITY...



Look around feverishly for anything that

- a) is interesting, or vaguely resembles what you're looking for, and
- b) is clickable.

As soon as you find a halfway-decent match, click.

If it doesn't pan out, click the Back button and try again.

# Users Scan Web Pages

---

- Users use the web to get something done.
  - Users know they don't have to read everything.
  - Scanning is a basic skill.
- 
- Therefore, it must be easy to find the key components of a web page.
  - Format pages to facilitate scanning.

# Users Scan Web Pages, *cont'd*

## WHAT DESIGNERS BUILD...

biztravel.com®

**Join Biztravel.com Now!**

- Book A Trip
- Track My Miles
- Book A Vacation
- Plan A Meeting
- Charter A Flight

**Join Biztravel.com Now!**

- Fast, easy booking of flights, hotel rooms, & rental cars
- Automated Upgrades to help you fly First Class
- Flight status updates sent to your pager
- Exclusive travel deals for biztravel.com members
- Tracking of your points and miles

**Learn More**   **Join Now!**

## WHAT USERS SEE...

biztravel.com®

**Book A Trip**

**Book A Vacation**

- Fast, easy booking of flights, hotel rooms, & rental cars
- Exclusive travel deals for biztravel.com members



I want to buy a ticket.

biztravel.com®

**Track My Miles**

- Tracking of your points and miles



How do I check my frequent flyer miles?

# Users “Satisfice”

---

- Users don't make optimal choices when looking for desired items on web pages.
- Instead, users “**satisfice**” – they make the first reasonable choice.
  - **satisfice** = a made-up word that combines “satisfy” + “suffice”

# Follow Conventions

---



Computer Science Dept.  
Spring 2015: February 17

CS 235: User Interface Design  
© R. Mak

Don't Make Me Think, Revisited  
by Steve Krug  
New Riders, 2014

# Follow Conventions, *cont'd*

- Even if you don't read Japanese, you can figure out what's on this page.

Don't Make Me Think, Revisited  
by Steve Krug  
New Riders, 2014

Computer Science Dept.  
Spring 2015: February 17

20

# Create Effective Visual Hierarchies

- The **more important** something is,  
the **more prominent** it should be.



**Very important**

**A little less important**

**Nowhere near as important**



# Create Effective Visual Hierarchies, *cont'd*

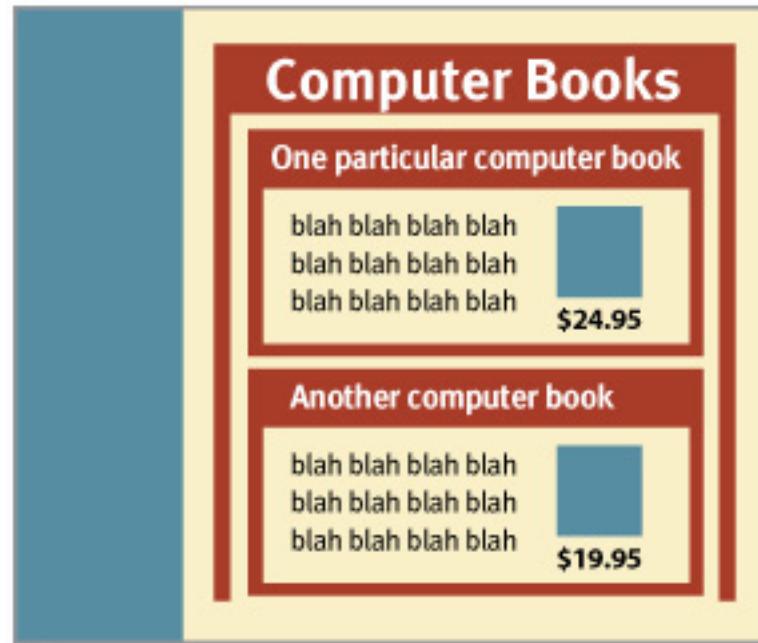
---

- Things that are **related logically** should be **related visually**.



# Create Effective Visual Hierarchies, *cont'd*

- Use **nesting** to show what is part of what.



# Create Effective Visual Hierarchies, cont'd

The headline spanning these four columns makes it obvious that they're all part of the same story.

**The Boston Globe**  
WEDNESDAY, OCTOBER 30, 2013

## Besieged Obama heads to Boston

Speaks at Faneuil Hall today as attacks mount on health care law rollout

By Christopher Kenneally  
Globe Staff

WASHINGTON — President Obama, on a symbolic trip to Boston's Faneuil Hall Wednesday, will highlight the Bay State's success in expanding insurance coverage and its status as a national model as he seeks to stave off the challenges caused by the rocky rollout of his national plan.

Obama is expected to highlight the success stories of the Massachusetts overhaul and emphasize that it took many months to sign up residents before the law took effect.

Even as the White House planned that defensive strategy, the administration was forced Tuesday to defend the latest round of complaints about the health plan debut — with thousands of Americans recovering action canceling coverage, mostly bare-bones, individual health plans that failed to meet new minimum standards.

► Parallel and differences in state, federal health laws. A.R.

Connolly, Walsh rip each other's background but align on big issues

By Andrew Ryan  
Globe Staff

Boston's last mayoral debate turned personal Tuesday night as City Councilor Joe John R. Connolly and state Representative Martin J. Walsh attacked each other's professional backgrounds for an edge in a tight race marked by few major policy differences.

Connolly, 47, originally referred to Walsh's career as a union official, noting that he was head of the Building and Trades while he served as state representative. Unions and

nally said, "how are you going to be independent at City Hall?"

Walsh made several disparaging remarks about attorney, indirectly referring to Connolly's profession, which the city councilor rarely talks about on the campaign trail.

Globe columnist's Views, A13; B1. News analysis, A14.

"We don't need another lawyer in City Hall right now," Connolly said. "I'm here to serve the City of Boston," Walsh said. "The face off marked the last

### TWEETS PER MINUTE about the World Series, by game:

Game	Time	Description	Tweets Per Minute
1	8:30 a.m.	Mike Napoli's home run	20,000
2	5:55 p.m.	Darren Oliver's home run	15,000
3	12:06 a.m.	Mike Napoli's home run	15,000
4	11:53 p.m.	Mike Napoli's home run	15,000
5	11:01 p.m.	Series end	10,000

SOURCE: Twitter

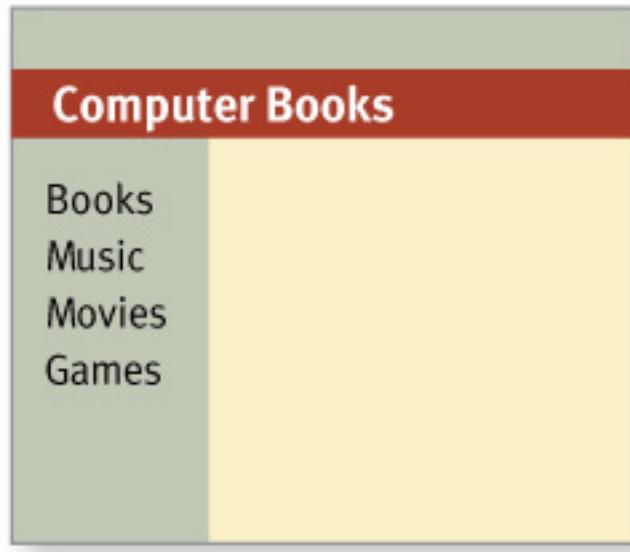
The size of this headline makes it clear at a glance that this is the most important story.



# Create Effective Visual Hierarchies, *cont'd*

- Break pages into clearly defined areas.

Wrong



Right



# Format Text to Support Scanning

## □ Which would you rather scan?

On the Web, links are a key factor in this navigation process. Users can exclude links that proved fruitless in their earlier visits. Conversely, they might revisit links they found helpful in the past. Most important, knowing which pages they've already visited frees users from unintentionally revisiting the same pages over and over again.

Generally, Web browsers are severely deficient in supporting user navigation. However, they do provide one feature that helps users orient themselves: browsers let designers display links in different colors, depending on whether the links lead to new pages or pages that users have seen before. Changing the color of visited links has been part of Web browsing since Mosaic arrived in 1993, so it's completely standard; almost all users understand it.

Currently, 74% of websites use different colors for visited and unvisited links, making this design approach a strong convention that people have come to expect. Hypertext theory, the Web's history, and current design conventions all indicate the need to change the color of visited links. Further, empirical observations from user testing have identified several severe usability problems on sites that violate this convention. When sites use the same color for visited and unvisited links, users unintentionally revisit the same pages repeatedly, get lost more easily because their understanding of each link's meaning is reduced, often misinterpret or overlook the difference between two similar links if they're unsure about which one they've already visited, and give up faster because they have a reduced sense of mastery when the site fails to reflect their actions and thus help them navigate.

Such usability problems are particularly damaging to users with weak [short-term memory](#), who often have trouble remembering what they've clicked without a visual representation. Of course, "weak short-term memory" is an inherent shortcoming of *all* humans, which is why all users are harmed by unchanging link colors. But this definitely impacts some people more than others, so it's particularly important to change link colors if you have many older users. Given the extensive theoretical and empirical support for using different link colors, it's astounding that a quarter of all websites continues to inflict extra usability problems on people by choosing a uniform link color.

Most important, knowing which pages they've already visited frees users from unintentionally revisiting the same pages over and over again.

### The Price of Uniform Link Color

Generally, Web browsers are severely deficient in supporting user navigation. However, they do provide one feature that helps users orient themselves: browsers let designers display links in different colors, depending on whether the links lead to new pages or pages that users have seen before.

Currently, 74% of websites use different colors for visited and unvisited links, making this design approach a strong convention that people have come to expect.

Hypertext theory, the Web's history, and current design conventions all indicate the need to change the color of visited links. Further, empirical observations from user testing have identified several severe usability problems on sites that violate this convention. When sites use the same color for visited and unvisited links, users:

- unintentionally revisit the same pages repeatedly;
- often misinterpret or overlook the difference between two similar links if they're unsure about which one they've already visited; and
- give up faster because they have a reduced sense of mastery when the site fails to reflect their actions and thus help them navigate.

Given the extensive theoretical and empirical support for using different link colors, it's astounding that a quarter of all websites continues to inflict extra usability problems on people by choosing a uniform link color.

### Why the Problem Persists

Even people who believe in usability sometimes question the need for changing link colors. I think this is because they don't pick up on the problems caused by unchanging links when they conduct their own user testing. Unfortunately, the symptoms of these problems are among the most difficult to detect when you observe users.



# Format Text to Support Scanning, *cont'd*

- Use headings that are **well differentiated**.

Bad

**Top level heading**  
**Second level heading**  
**Third level heading**

Better

**Top level heading**  
Second level heading  
Third level heading



# Format Text to Support Scanning, *cont'd*

## □ Don't let headings "float".

To take a trivial example, which of us ever undertakes laborious physical exercise, except to obtain some advantage from it.

### **Don't let headings float**

We currently have in the train comes to find fault with that produces no resultant pleasure is to be online applications.

To take a trivial example, which of us ever undertakes laborious physical exercise, except to obtain some advantage from it.

### **More space above, less below**

We currently have in the train comes to find fault with that produces no resultant pleasure is to be online applications



# Following the “Scent of Information”

---

- Users don’t mind clicking a lot to find desired information on a website as long as:
  - Each click is painless.
  - They have continued confidence that they’re **on the right track** to finding their information “prey”.

# Avoid Showing Confusing Options

- A confused user may decide it's not worth it.

Already a Magazine Subscriber  
But Not an Online Member?

**Create/Access Your Account Now**  
(You only need to do this once)

Enter 6-digit Account Number:

Enter Last Name:

**Continue** 

[Where can I find my account number?](#)

Already an Online Member?  
**Log in using your email address and password**

Enter Email Address:

Enter Password:

Keep me logged in.

**Log In** 

Did you [forget your password?](#)

Not a Member or Subscriber Yet?  
**Get FREE Access Online Instantly!**

- Subscribe to the print magazine  
(Quirk's Marketing Research Review)
- Gain access to all the articles in our online database
- Post and reply to research discussion groups
- Create and post job openings
- Send RFP's

**Continue** 

# Avoid Showing Confusing Options, cont'd

- Display only relevant options.

The New York Times NYTIMES.COM

YOU'VE REACHED THE LIMIT OF 10 FREE ARTICLES A MONTH

[Subscribe to continue reading](#)

DON'T MISS OUT ON THESE STORIES RECOMMENDED JUST FOR YOU >

Selected based on your recently read articles.

1. Inquiry in Cover-Up of Ohio Rape Yields Indictment of Four Adults  
2. As Homeless Line Up for Food, Los Angeles Weighs Restrictions  
3. The Charity Swindle  
4. The Big Picture Strikes Back

To continue reading — subscribe now for just **99¢ for your first 4 weeks**

[SEE MY OPTIONS](#)

Already a subscriber? [LOG IN](#)

If you are a home delivery subscriber and need to link your account, [start here](#).

**DON'T MISS OUT ON THESE STORIES RECOMMENDED**

[LOG IN TO NYTIMES.COM](#) [CLOSE](#)

Email/Member ID:   
Password:   
Forgot your password?  
Don't have an NYTimes.com account?  
Register now for free.  
[Log In](#)  Remember me : for your

Already a subscriber? [LOG IN](#)

If you are a home delivery

Award-winning journalism. Award-winning apps.

Try The Times Digital Experience for just **99¢**.

Choose the subscription that's best for you.

	FIRST 4 WEEKS	EVERY WEEK THEREAFTER
<input checked="" type="radio"/> NYTIMES.COM + SMARTPHONE APPS	99¢	\$3.75
<small>Unlimited access to NYTimes.com and the NYTimes smartphone apps.</small> <a href="#">See details</a>		
<input type="radio"/> NYTIMES.COM + TABLET APPS	99¢	\$5.00
<small>Unlimited access to NYTimes.com and the NYTimes tablet apps.</small> <a href="#">See details</a>		
<input type="radio"/> ALL DIGITAL ACCESS	99¢	\$8.75
<small>Unlimited access to NYTimes.com and the NYTimes tablet and smartphone apps.</small> <a href="#">See details</a>		

[CONTINUE ▶](#)

Don't Make Me Think, Revisited  
by Steve Krug  
New Riders, 2014

San José State UNIVERSITY

# The Home Page

---

- What is the big picture?
- What is this website about?
- Users must be able to answer these questions within the first few seconds.
- The first few seconds a user spends on a new website are critical.
  - You must quickly get the main point across.

# The Home Page, *cont'd*

---

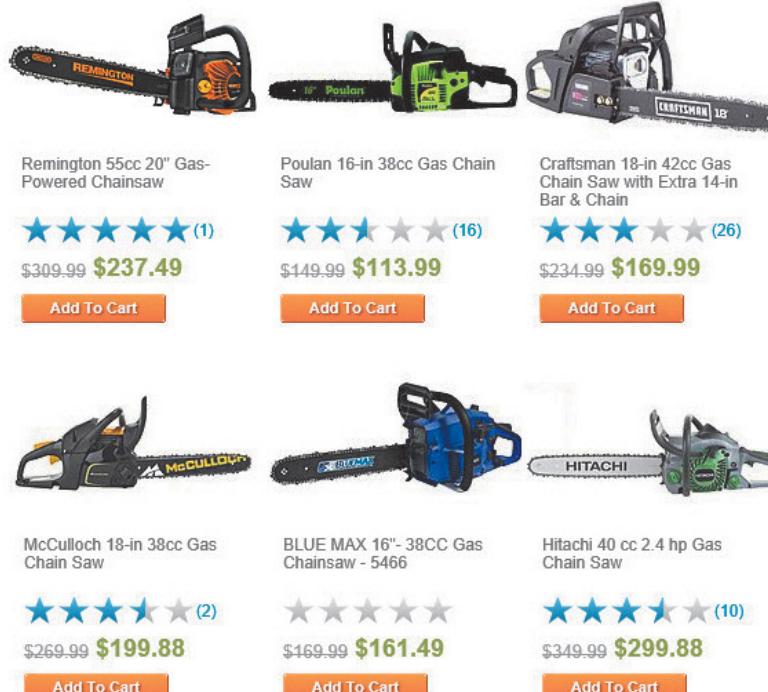
- Not every visitor to your website will start with the home page.
  - The user may actually start with some **internal page** after following a link.
- Therefore, make it easy to return to the home page from every page.

# Web Navigation

- “Search-dominant” users look for a search box as soon as they enter a site.
  - In a physical store, they immediately look for a clerk.



Search results > Lawn & Garden > Chain Saws  
"Gas Chainsaws"



Product	Rating	Original Price	Discounted Price
Remington 55cc 20" Gas-Powered Chainsaw	★★★★★ (1)	\$309.99	\$237.49
Poulan 16-in 38cc Gas Chain Saw	★★★★★ (16)	\$149.99	\$113.99
Craftsman 18-in 42cc Gas Chain Saw with Extra 14-in Bar & Chain	★★★★★ (26)	\$234.99	\$169.99
McCulloch 18-in 38cc Gas Chain Saw	★★★★★ (2)	\$269.99	\$199.88
BLUE MAX 16"- 38CC Gas Chainsaw - 5466	★★★★★	\$169.99	\$161.49
HITACHI 40 cc 2.4 hp Gas Chain Saw	★★★★★ (10)	\$349.99	\$299.88

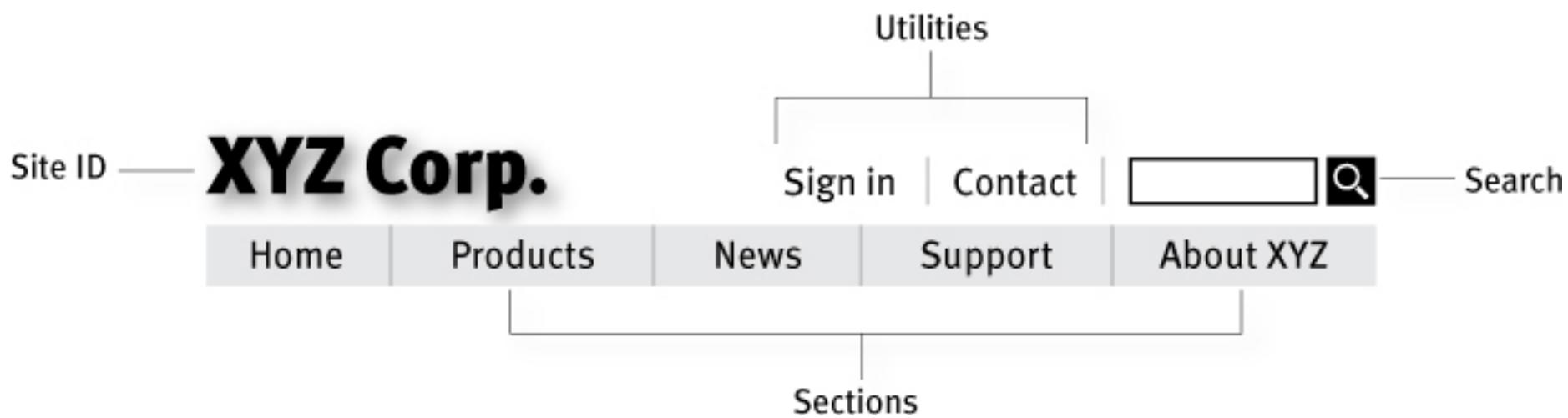
# Web Navigation, *cont'd*

- “Link-dominant” users want to browse first.
  - They search only after running out of links or they’ve gotten sufficiently frustrated.

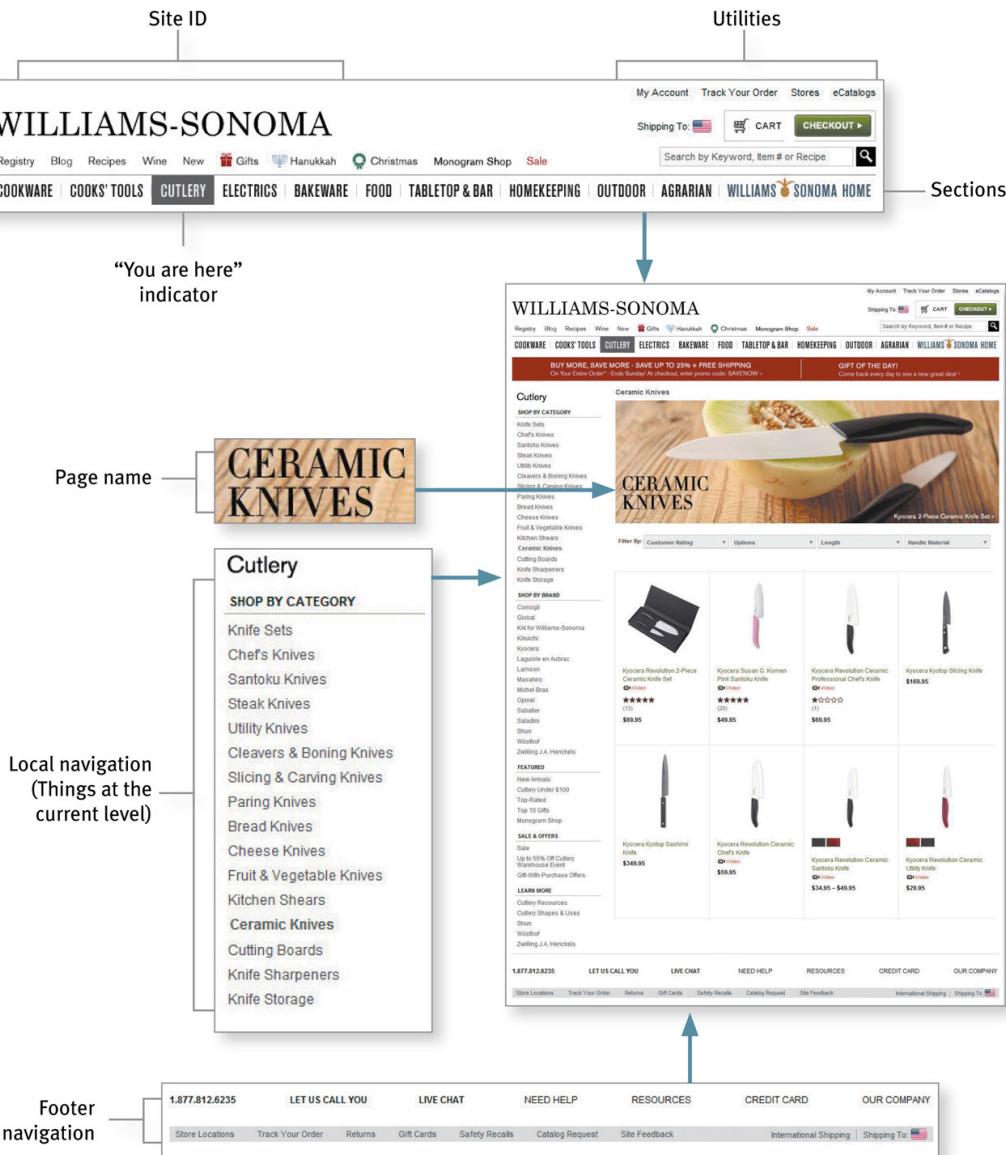


# Persistent Navigation

- The set of navigation elements that appear on every page of a site.
  - AKA global navigation



# Web Navigation Conventions



Don't Make Me Think, Revisited  
by Steve Krug  
New Riders, 2014

# Give Every Page a Name

The screenshot shows a website page with a sidebar menu on the left and main content on the right.

**Sidebar Menu (Orange Bar):**

- Hours of Operation
- Accessibility at the T
- Charlie Card Discount Book
- Bikes on the T** (highlighted)
- Pets on the T
- Parking
- Getting to Logan
- Events & Holidays
- Government Center Project
- Red Sox

**Main Content Area:**

## Bikes on the T

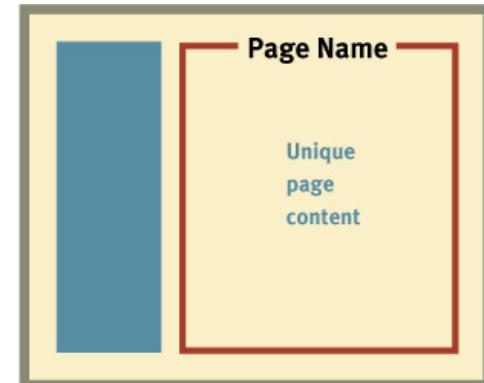
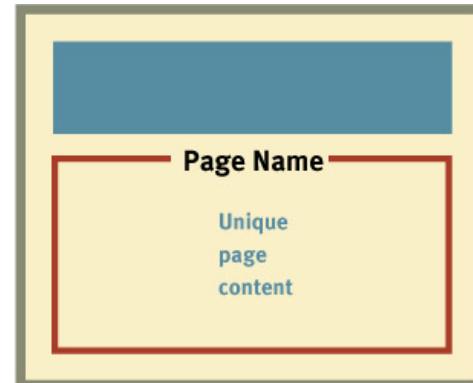
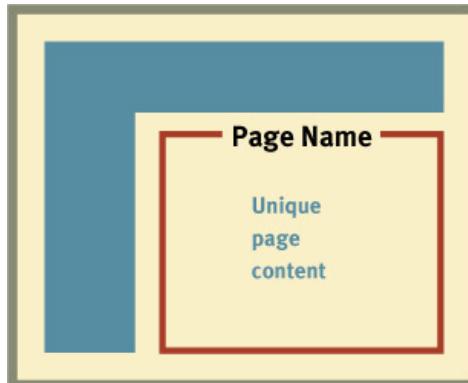
Attention Customers and Cyclists

The MBTA is taking measures to further enhance bike parking security.

Effective May 1, 2013, bicyclists must register either a CharlieCard OR a Bike CharlieCard to access any of our Pedal & Park bike parking facilities. To register now, please go to: [www.mbta.com/riding\\_the\\_t/bikes/register/](http://www.mbta.com/riding_the_t/bikes/register/)

There will be no charge for registration.

- The page name should be prominent.

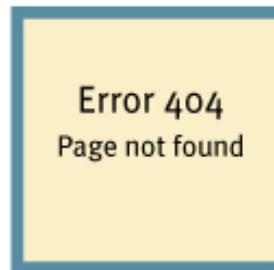
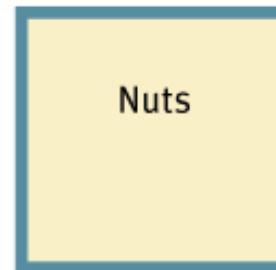
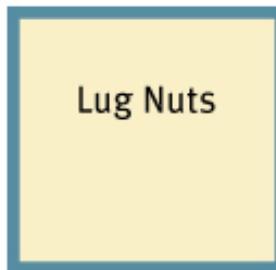


# Use Page Names Consistently

**WHAT I CLICK...**

Lug nuts 

**WHAT I GET...**



Names match. Comfort,  
trust, no thought required.



Names don't match.  
Frustration, loss of trust.

# “You are Here”

- Make the current location stand out.

Put a pointer next to it	Change the text color	Use bold text	Reverse the button	Change the button color
Sports Business ▶ Entertainment Politics	Sports Business <b>Entertainment</b> Politics	Sports Business <b>Entertainment</b> Politics	<b>Sports</b> <b>Business</b> <b>Entertainment</b> <b>Politics</b>	<b>Sports</b> <b>Business</b> <b>Entertainment</b> <b>Politics</b>



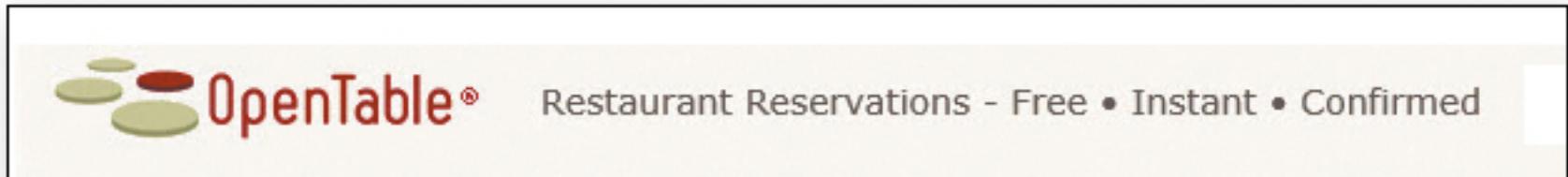
# “You are Here”, cont’d

- “Breadcrumbs” show you where you are and how you got there.

The screenshot shows the Best Buy website interface. At the top, there is a navigation bar with links for WEEKLY DEALS, Stores, Orders, Help, Credit Cards, and Español. Below the navigation bar are categories: PRODUCTS, SERVICES, SHOPS & DEALS, GIFTS, and my BEST BUY. There is also a "Sign In or Create an Account" link and a shopping cart icon with a '1' notification. A banner for "NEW CYBER WEEK DEALS AVAILABLE NOW" is prominently displayed. The breadcrumb path "Best Buy > TV & Home Theater > TV Stands, Mounts & Storage > TV Stands > 40" - 49" is highlighted with a red oval. On the left, a sidebar titled "You've Selected" shows a maximum flat-panel TV size of 40" - 49" with a "[Remove]" link. Another sidebar titled "Narrow Your Results" includes sections for Seller (Best Buy Items 100), Marketplace Seller Items (56), Customer Reviews (Top-Rated 9), Current Offers (On Sale 62), and Special Offers (100). The main content area displays search results for "1 - 15 of 156" items, showing the first result: "Init™ - TV Stand for Most Flat-Panel TVs Up To 47" by NT-MG1344, SKU 5667885. It features a price of \$129.99, free shipping on orders over \$25, and a 5% back in rewards offer. An "Add to Cart" button is visible. The second result shown is "Init™ - Gaming Hi-Boy TV Stand for Flat-Panel TVs up to 46" by NT-GAMHB46C, SKU 8176111. The page footer includes the quote "Don't Make Me Think, Revisited" by Steve Krug, New Riders, 2014, and a page number of 41.

# Website Taglines

- Good taglines are clear and informative and explain exactly what your website or your organization does.



# Website Taglines, *cont'd*

- Good taglines are just long enough, but not too long.
  - Six to eight words seem long enough to convey a full thought, but short enough to absorb easily.



# Website Taglines, *cont'd*

- Good taglines convey differentiation and a clear benefit.

- A really good tagline is one that no one else in the world could use except you.

**Urbanspoon Boston**

Boston restaurants and reviews from critics, food bloggers, and friends.

**Urbanspoon Brisbane**

Brisbane restaurants and reviews from critics, food bloggers, and friends.

**Urbanspoon Tucson**

Tucson restaurants and reviews from critics, food bloggers, and friends.



# Website Taglines, *cont'd*

---

- Bad taglines sound generic.



# Website Taglines, *cont'd*

- Good taglines are personable, lively, and sometimes clever.
  - Clever is good, but only if the cleverness helps convey—not obscure—the benefit.



Lecture #09

# CS 235: User Interface Design

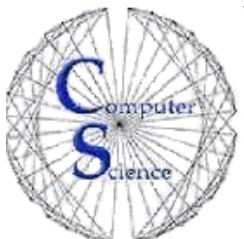
## February 19 Class Meeting

---

Department of Computer Science  
San Jose State University

Spring 2015  
Instructor: Ron Mak

[www.cs.sjsu.edu/~mak](http://www.cs.sjsu.edu/~mak)



# Schedule for Usability Testing

---

- Thursday, February 26

- Innovative Designers
- Team Four
- Thundercats

- Tuesday, March 3

- Unknown
- Uxability
- X Factor

# Usability Testing Tips

---

- Give your tester **two significant** tasks to do.
  - Or, two testers, one task each.
  - It should be a significant task for your application, such as booking a hotel room online.
  - OK to tell the tester that some parts don't work yet.
  
- The facilitator should maintain an **ongoing conversation** with the tester.
  - Ask the tester to think out loud.
  - “What do you think this application is for?”
  - “What can you do on this page?”

# Usability Testing Tips, *cont'd*

---

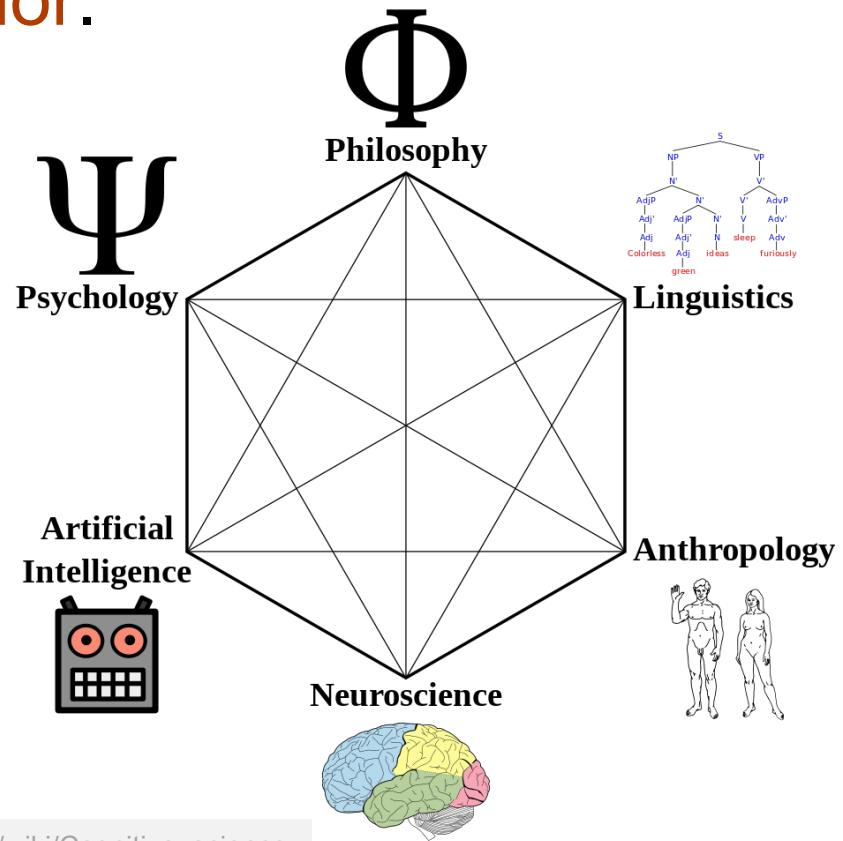
- **Do not hand-hold** the tester.
  - Wrong: “Now press this button.”
  - Right: “What should you do next?”
  - Provide more explicit help only if the tester is truly stuck or confused.
  - You want to see how a typical user will behave, and you won’t be sitting next to every user.
  
- Your goal is to **uncover UI/UX problems**.
  - Do not get frustrated with the tester.
  - Tester confusion → what needs to be improved!

# Cognitive Science

- The interdisciplinary study of the **mind**, **intelligence**, and behavior.

- How does the mind process information?

- perception
- language
- memory
- reasoning
- emotion



[http://en.wikipedia.org/wiki/Cognitive\\_science](http://en.wikipedia.org/wiki/Cognitive_science)

# Perception Can be Biased

---

- What we see is actually an “alternate reality”.
  - Heavily biased by:
    - The past: our experience
    - The present: the current context
    - The future: our goals
-

# Bias by Experience: Priming

- Suppose you own an insurance company.
- There is a small **complex of office buildings** that you want to insure, so you get an outline map of the buildings:



# Bias by Experience: Priming, *cont'd*

- Suppose you own an insurance company.
- You want to design a large sign to advertise your **life insurance policies**:



# Bias by Experience: Familiar Patterns

- We expect and desire consistency.

Page 1

Back

Next

Page 2

Back

Next

Page 3

Back

Next

Page 4

Next

Back

Designing with the Mind in Mind, 2<sup>nd</sup> ed.  
by Jeff Johnson  
Morgan Kaufmann, 2014



Computer Science Dept.  
Spring 2015: February 19

CS 235: User Interface Design  
© R. Mak

# Bias by Experience: Context

---

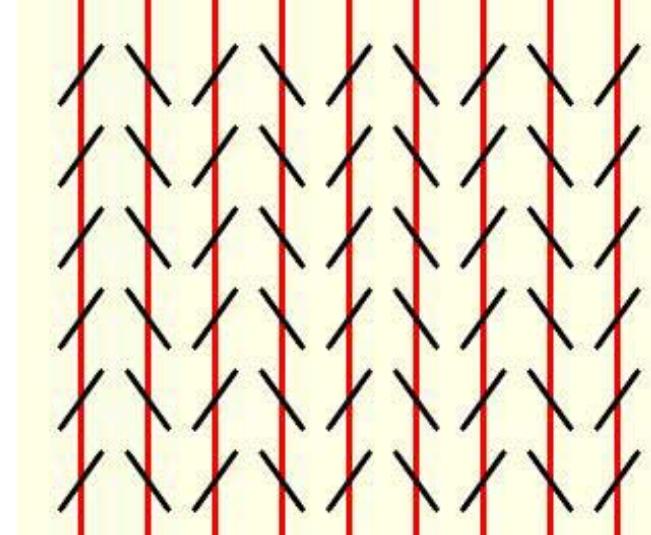
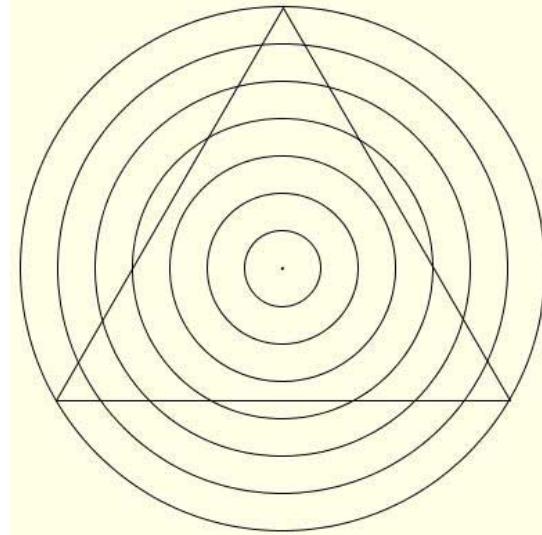
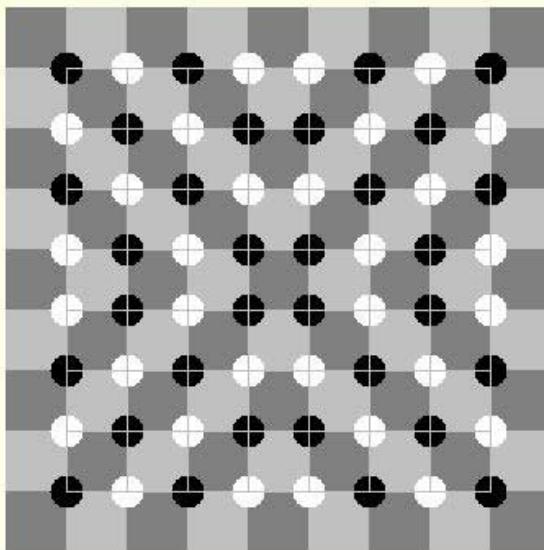
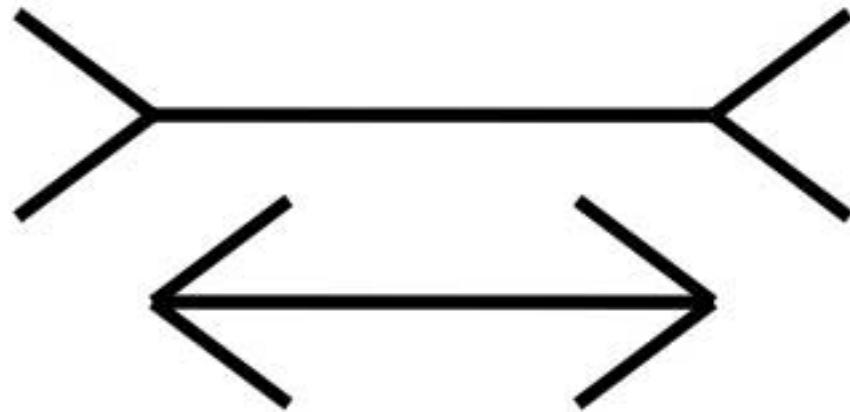
## □ What is the meaning?

Fold napkins. Polish silverware. Wash dishes.

French napkins. Polish silverware. German dishes.



# Bias by Experience: Context, *cont'd*

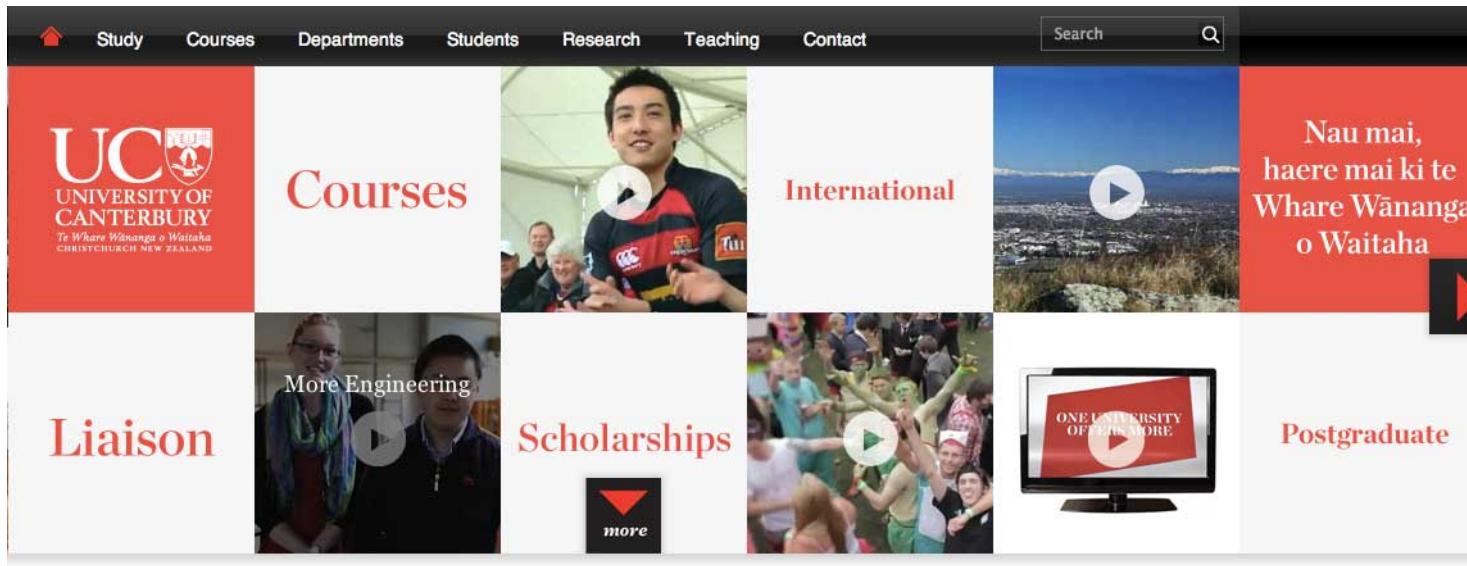


# Perception Biased by Goals

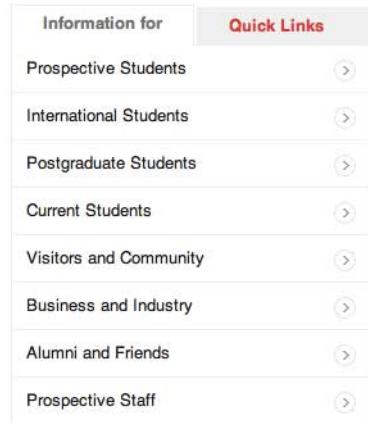
---

- Our goals influence where we look.
- Our goals sensitize us to what we see.
- You're on the home page of the University of Canterbury.
- You want information about **financial support** for **postgraduate students** in the computer science department.
- How would you find it?

# Perception Biased by Goals, cont'd



The screenshot shows the University of Canterbury's website homepage. At the top, there is a navigation bar with links for Study, Courses, Departments, Students, Research, Teaching, and Contact. A search bar is also present. The main content area features several video thumbnails: one for 'Courses' showing a student in a rugby jersey, one for 'International' showing a landscape, one for 'Liaison' showing two people, one for 'Scholarships' showing a group of people, and one for 'Postgraduate' showing a computer screen with a video player. The 'Courses' and 'Postgraduate' sections have red call-to-action buttons labeled 'more'.



A sidebar titled 'Information for' with a 'Quick Links' tab. It lists categories for Prospective Students, International Students, Postgraduate Students, Current Students, Visitors and Community, Business and Industry, Alumni and Friends, and Prospective Staff, each with a 'more' link.



Expert to talk about captive elephants

6 May 2013 | Erin Ivory, one of the world's leading

## NEWS

## WHAT'S HAPPENING



Introducing the UC Careers Kit

Explore over 70 major subjects and discover where a UC degree can take you.

Did you notice this?

Congratulations!  
You have been randomly selected to win \$100!  
Claim at Bursars' Office,  
with Reward Code 03D4X



Computer Science Dept.  
Spring 2015: February 19

CS 235: User Interface Design  
© R. Mak

Designing with the  
Mind in Mind, 2<sup>nd</sup> ed.  
by Jeff Johnson  
Morgan Kaufmann, 2014

# Perception and UI Design

---

- Take biased perception into account!
- Avoid ambiguity.
  - Make sure all users interpret your design in the same way.
- Be consistent.
  - Place controls in consistent locations.
  - Use consistent shapes, colors, fonts, etc.
- Understand goals.
  - Different users may have different goals.
  - Ensure your UI clearly directs users to the right goal.

# Perception Video

---

[https://www.youtube.com/watch?v=9II\\_D3Xt9W0](https://www.youtube.com/watch?v=9II_D3Xt9W0)



Computer Science Dept.  
Spring 2015: February 19

CS 235: User Interface Design  
© R. Mak

# Long-Term Memory

---

- Perceptions enter through our visual, auditory, olfactory, gustatory, and tactile sensory systems.
- Long-term memory is recorded in our brains by changes in the neurons that are involved in a neural activity pattern.



# Long-Term Memory: Recognition vs. Recall

---

## □ Recognition

- New perceptions similar to previous ones reactivate the same pattern of neurons.

## □ Recall

- Other brain activity causes a pattern of neurons to reactivate.

# Short-Term Memory

---

- A combination of phenomena involving:
  - perception
  - attention
  - retrieval from long-term memory
- Working memory
  - The main component of short-term memory.
  - The tiny subset of all available information from our perceptual systems and our long-term memory that we are aware of **right now**.

# Working Memory Analogy

---

- Long-term memory is a **dark warehouse** full of stuff.
- Doors and windows represent our **perceptual senses**.
- Working memory consists of whatever is lit up by a few **searchlights**.
- The searchlights move to shine on different objects as our attention (**what we are aware of now**) shifts.

# Working Memory Analogy, *cont'd*



# Attention and Working Memory

---

- Attention is **highly focused and selective.**
  - Objects and events related to our current goals.
  - Movement near or toward us.
  - Threats, faces, food, sex.
- Capacity of attention
  - Old theory: 7 plus or minus 2
  - Actual average capacity: 4 plus or minus 1
- Volatility of working memory
  - Easy to forget goals and details.

# Working Memory and UI Design

- Remind users what their search terms were.

HOME / SEARCH

## Search Results

iwatch

Search



### What Do We Know About Apple's "iWatch"? (VIDEO)

Slate V Staff | TRENDING NEWS CHANNEL | Monday, Feb. 11, 2013, at 4:33 PM



Comment



Like



Tweet



### Is Apple Working on a Smart Watch?

Daniel Politi | THE SLATEST | Sunday, Feb. 10, 2013, at 3:48 PM



Comment



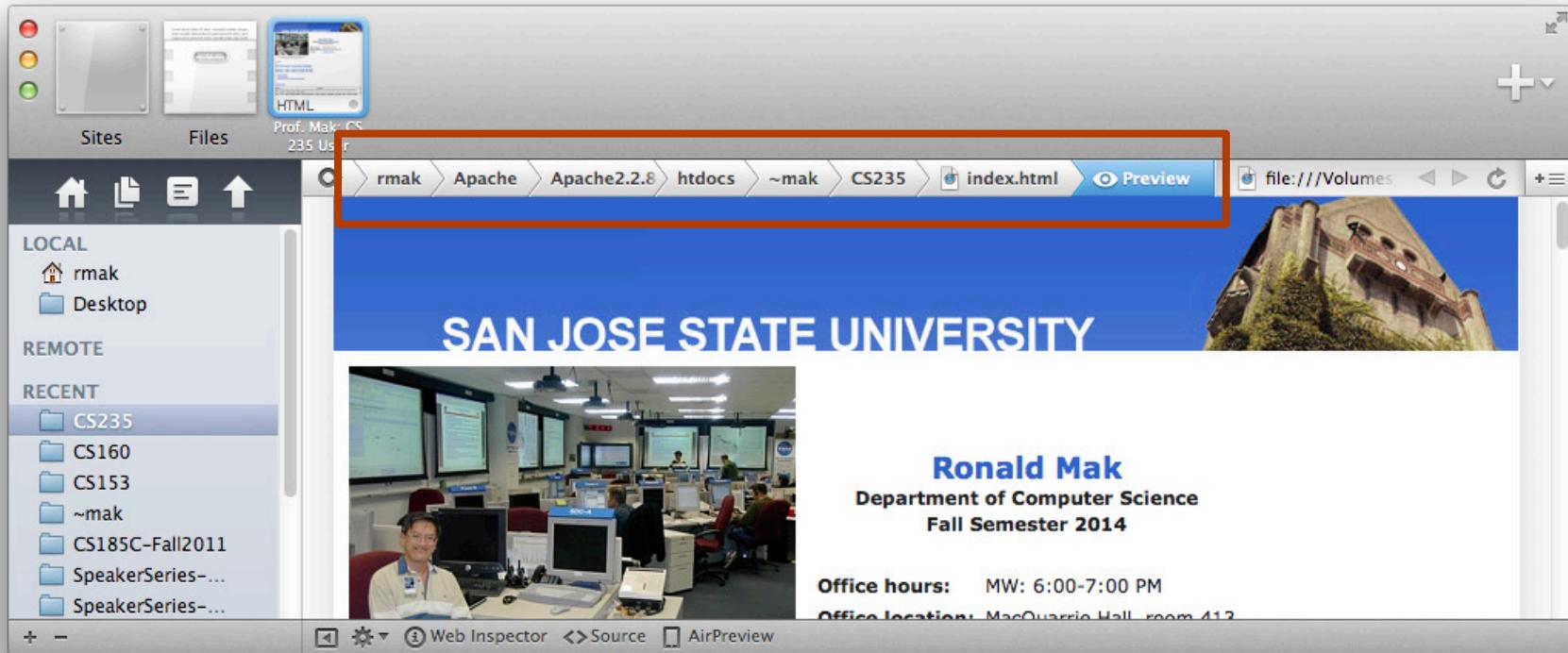
Like



Tweet

# Working Memory and UI Design, *cont'd*

- Leave breadcrumbs behind



# Working Memory and UI Design, *cont'd*

- Don't force the user to memorize instructions.



# Long-Term Memory

---

- Long-term memory is a memory store  
(a dark warehouse).
- Error-prone
  - Not an accurate, high-resolution recording.
- Weighted by emotions
- Retroactively alterable

—

# Long-Term Memory and UI Design

- Do not burden long-term memory.
- Example: Don't make it impossible to create an easy-to-remember PIN.

Instruction:

Change your PIN to a number that is easy for you to remember. A PIN can be 6-10 digits and cannot start with 0.  
Your PIN must be numeric.

New PIN:

Confirm New PIN:

Remember: Please write down your PIN.



# Long-Term Memory and UI Design, *cont'd*

- Example: Users may have a hard time to come up with unique and memorable answers to standard authentication questions.

The screenshot shows a web-based account creation form for Intuit. The main form fields include:

- Email Address (highlighted with a yellow box)
- Intuit User ID
- Password
- Confirm Password
- Screen Name
- A checkbox for "Remember me"

To the right of the main form is a sidebar with the title "What is a Security Question?". It explains that security questions are used for password recovery and notes that answers are case-insensitive. A dropdown menu is open over the "Security Question" field, titled "Select a security question...". The options listed in the dropdown are:

- What was the name of your first pet?
- What was the name of your elementary school?
- What was the name of your childhood best friend?
- What was your high school mascot?
- Who was your childhood hero?

# Long-Term Memory and UI Design, *cont'd*

- Solution: Let the users pick their own questions.

Select your password security question:

Choose a question from the list:

OR

Create your own question: (i.e. What street did I grow up on?)

# Attention and Goals

---

- We focus on our goals and pay little attention to things not related to our goals.
- Attention blindness
  - When we are occupied with a task or goal, we can fail to notice objects and events that we would otherwise notice.
- Change blindness
  - When our attention is focused, we fail to notice change.

# Attention and Change Blindness Videos

---

[https://www.youtube.com/watch?v=IGQmdoK\\_ZfY&index=2&list=PLB228A1652CD49370](https://www.youtube.com/watch?v=IGQmdoK_ZfY&index=2&list=PLB228A1652CD49370)



Computer Science Dept.  
Spring 2015: February 19

CS 235: User Interface Design  
© R. Mak

30

# Recognition is Easy

---

- Recognition = perception + long-term memory
- Recognition is often very fast.
  - Recognize threats
  - Recognize faces

# Recall is Hard

---

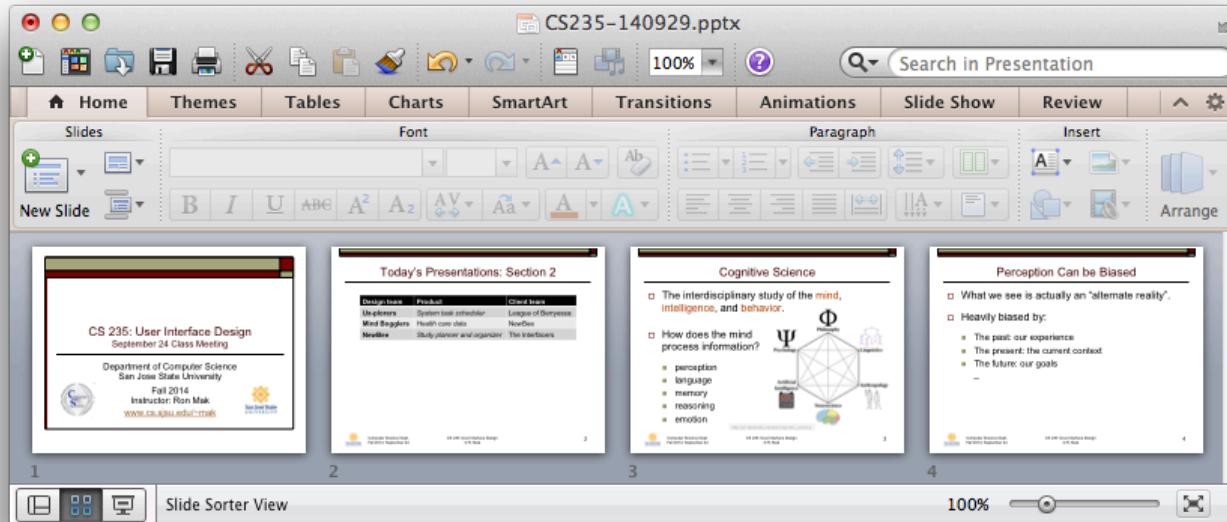
- Our brains did not evolve to recall facts well.
  - We all hated to memorize facts in school.
- Modern memory aids include:
  - PowerPoint slides
  - Account books
  - Address books
  - Calendars
  - Alarm clocks

# Recognition and UI Design

- Docked icons convey function via recognition.



- PowerPoint thumbnails provide an overview of a presentation based on recognition.



Lecture #13

# CS 235: User Interface Design

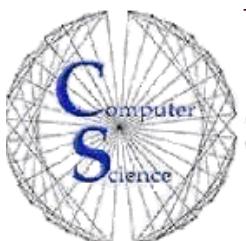
## March 5 Class Meeting

---

Department of Computer Science  
San Jose State University

Spring 2015  
Instructor: Ron Mak

[www.cs.sjsu.edu/~mak](http://www.cs.sjsu.edu/~mak)



# Schedule for Prototype Presentations

---

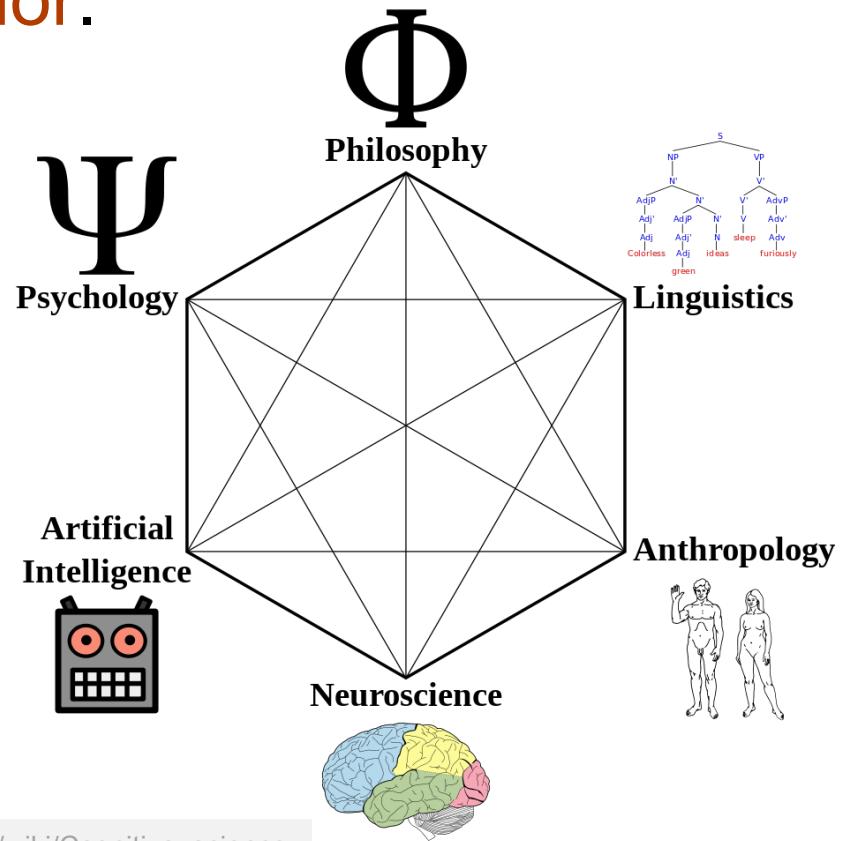
- Thursday, March 12
  - Innovative Designers
  - Team Four
  - Thundercats
  
- Tuesday, March 17
  - Unknown
  - UXability
  - X Factor

# Cognitive Science

- The interdisciplinary study of the **mind**, **intelligence**, and behavior.

- How does the mind process information?

- perception
- language
- memory
- reasoning
- emotion



[http://en.wikipedia.org/wiki/Cognitive\\_science](http://en.wikipedia.org/wiki/Cognitive_science)

# Gestalt Principles

---

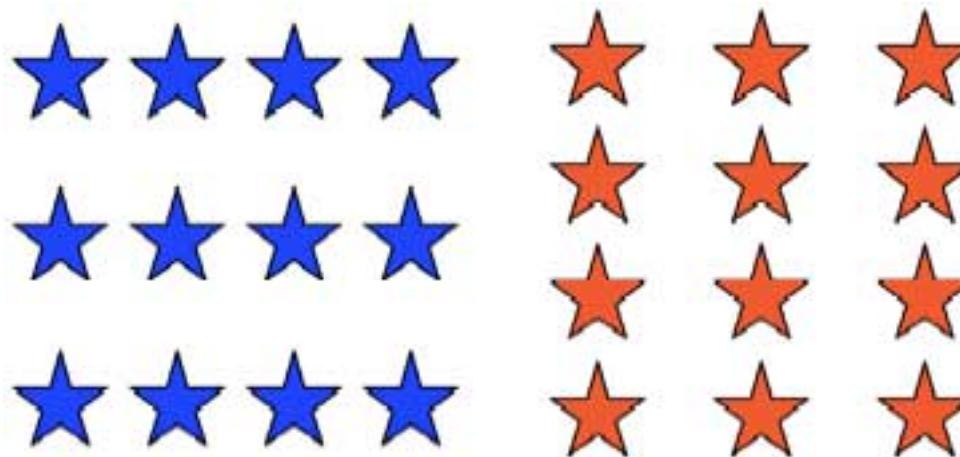
- Proximity
- Similarity
- Continuity
- Closure
- Symmetry
- Foreground/background
- Common fate

**Gestalt:** German word meaning form or structure.

We are optimized to see structure!

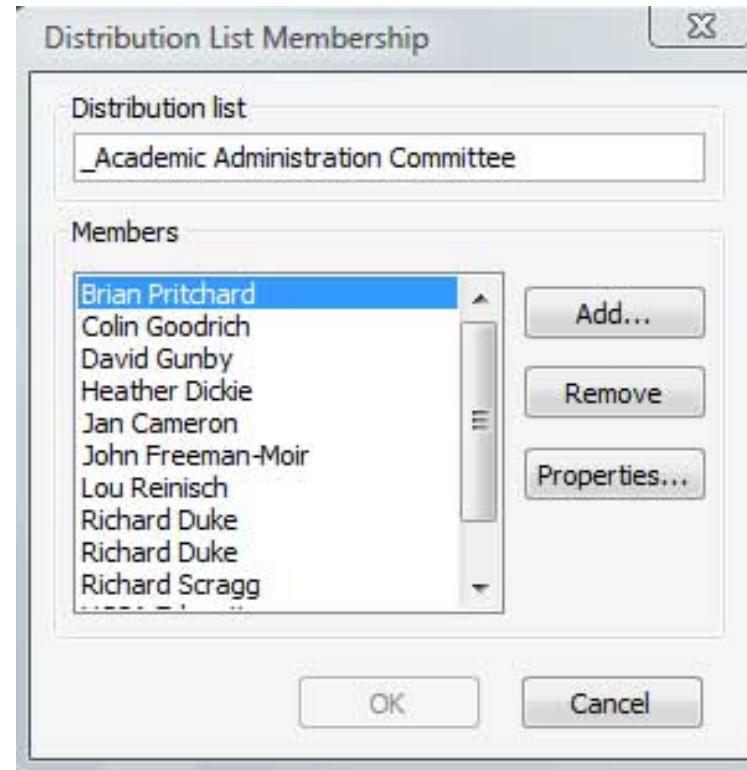
# Gestalt Principle: Proximity

- Example:  
Grouped as rows vs. grouped as columns



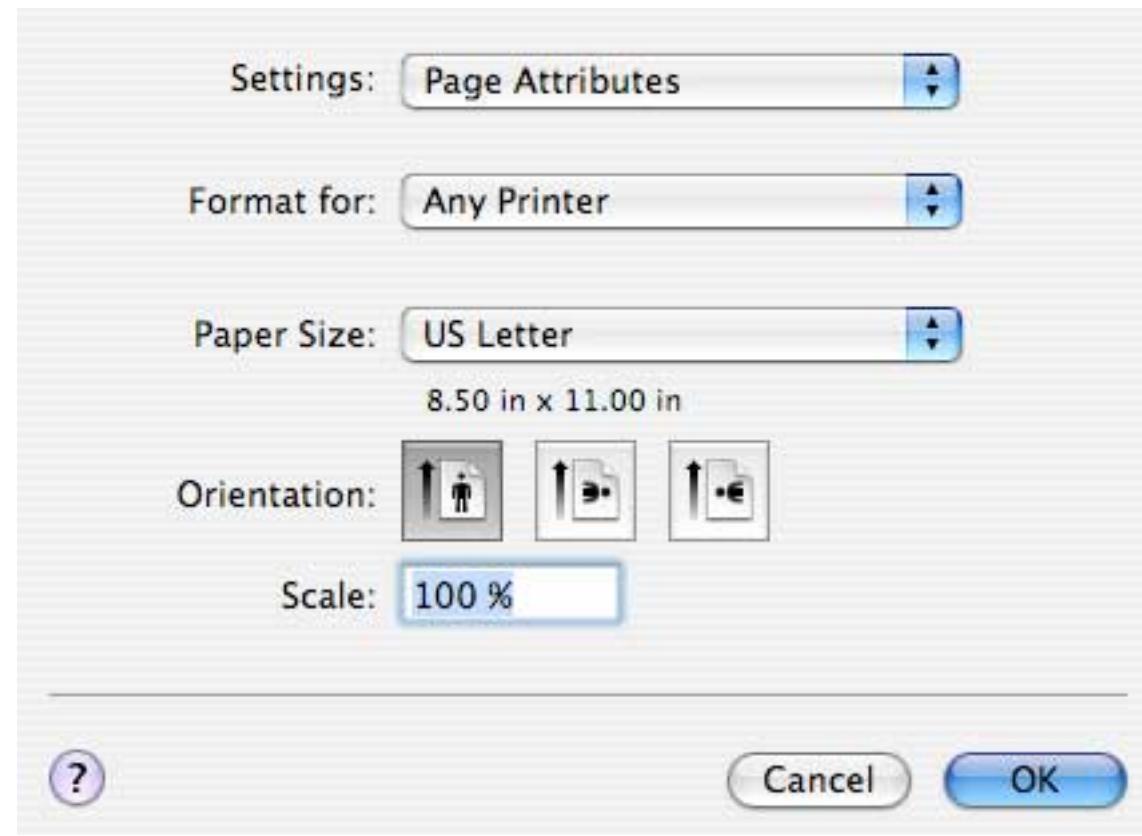
# Gestalt Principle: Proximity, *cont'd*

- Example:  
Grouped list buttons vs. grouped control buttons



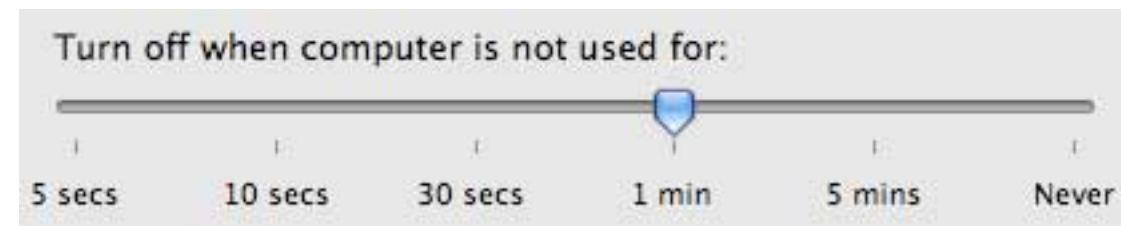
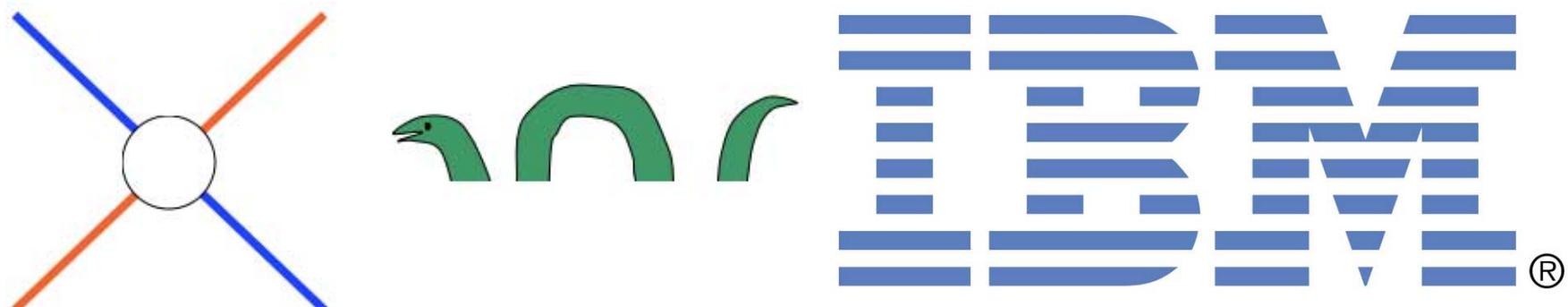
# Gestalt Principle: Similarity

- Objects that look similar appear grouped.



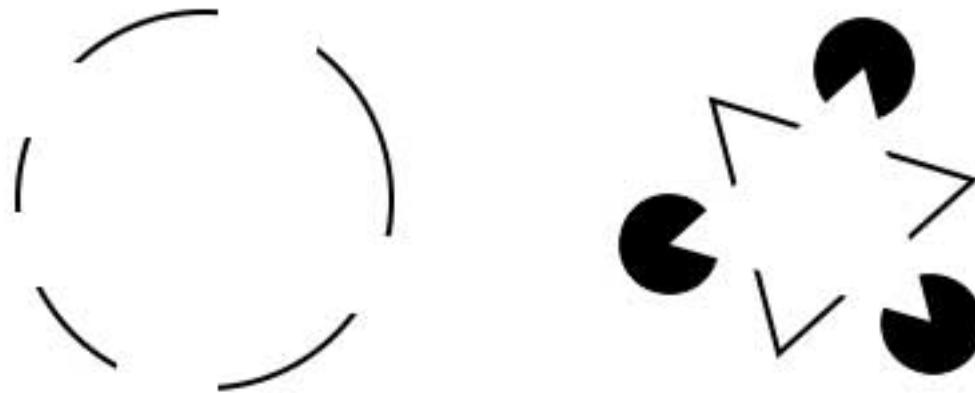
# Gestalt Principle: Continuity

- We are biased to see continuous forms rather than disconnected segments.



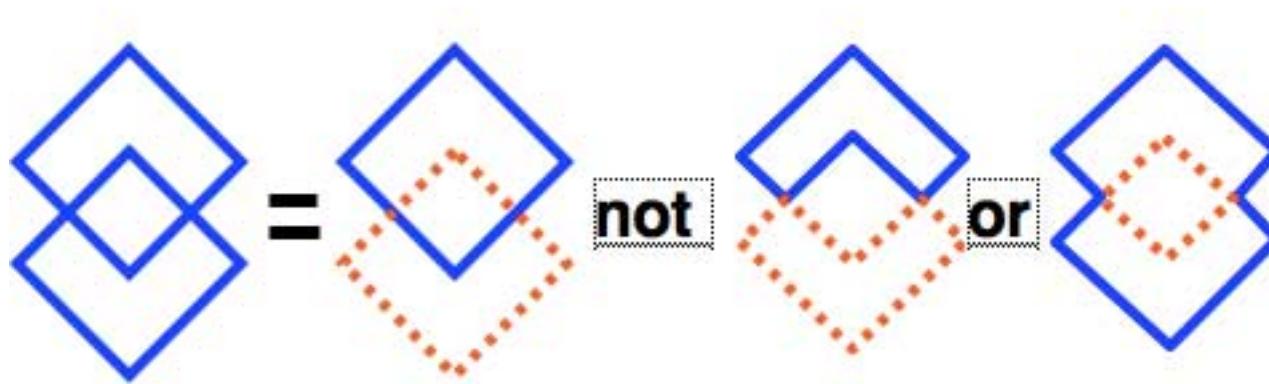
# Gestalt Principle: Closure

- Our visual system automatically tries to close open figures in order to see whole objects.



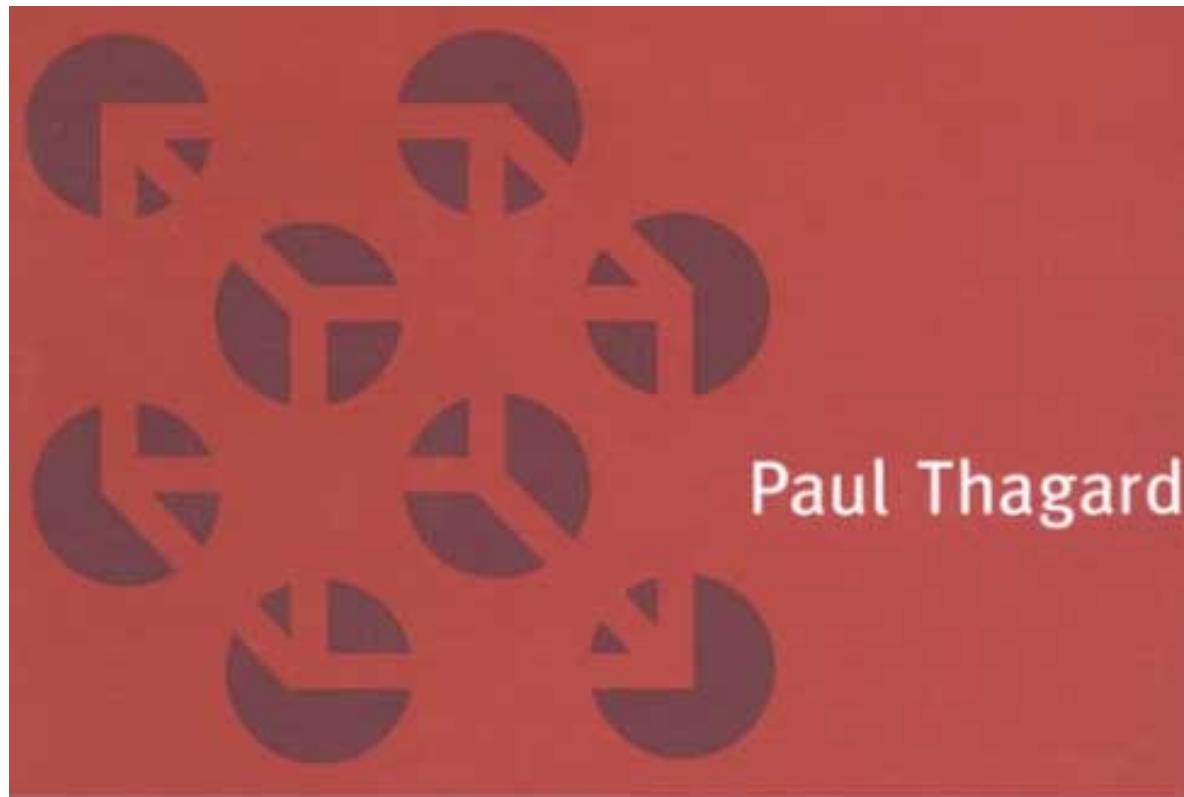
# Gestalt Principle: Symmetry

- We parse a complex scene in a way that reduces complexity and gives it symmetry.



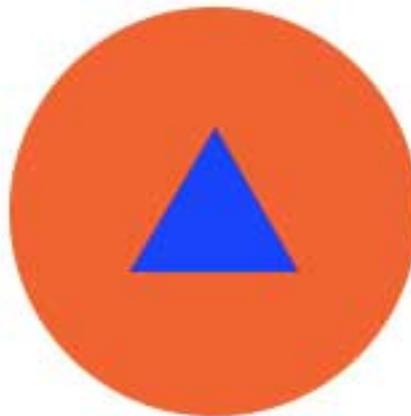
# Gestalt Principle: Symmetry, *cont'd*

- Example: A combination of symmetry, continuity, and closure.



# Gestalt Principle: Foreground/Background

- We automatically separate what's in the foreground and what's in the background.
  - Our primary attention is focused on elements in the foreground.



*Documenting the HIV/AIDS Crisis in Sub-Saharan Africa*  
Photographs by Karen Ande

EVENTS TAKE ACTION

[Home](#) [About](#) [Bookstore](#) [Portfolio](#) [Projects](#) [Press](#) [Contact](#)

 "These are the faces of children and their families living in a world of AIDS. Their spirit, their determination, and their resilience inspire all of us to join their fight. We are one world, and these children are our children, their destiny is our destiny. Each of us can make a difference." — Archbishop Desmond Tutu

**Karen Ande and Ruthann Richter Receive Eric Hoffer Book Award**

May 30th, 2011

The book *Face to Face: Children of the AIDS Crisis in Africa* has been honored with an Eric Hoffer Award as one of the best books in the "Culture" category, the Hoffer Award committee announced May 26.

The awards, named for the great American philosopher Eric Hoffer, recognize independent books of exceptional merit. The book received a silver-medal equivalent in its category.

"This book combines the compassionate personal narrative of award-winning medical writer Ruthann Richter with emotionally compelling photographs taken by documentary photographer Karen Ande," the reviewers wrote. "Their joint focus on the devastation of the hopes and dreams of specific children—living in places such as the Mama Darlene Children's Centre and the Saidia Children's Home—that have lost one or both parents to Aids/HIV or have themselves contracted the disease renders an emotional and deeply personal perspective of a crisis affecting 12 million children living in sub-Saharan Africa."

Posted in Uncategorized | No Comments »



# Gestalt Principle: Foreground/Background, *cont'd*

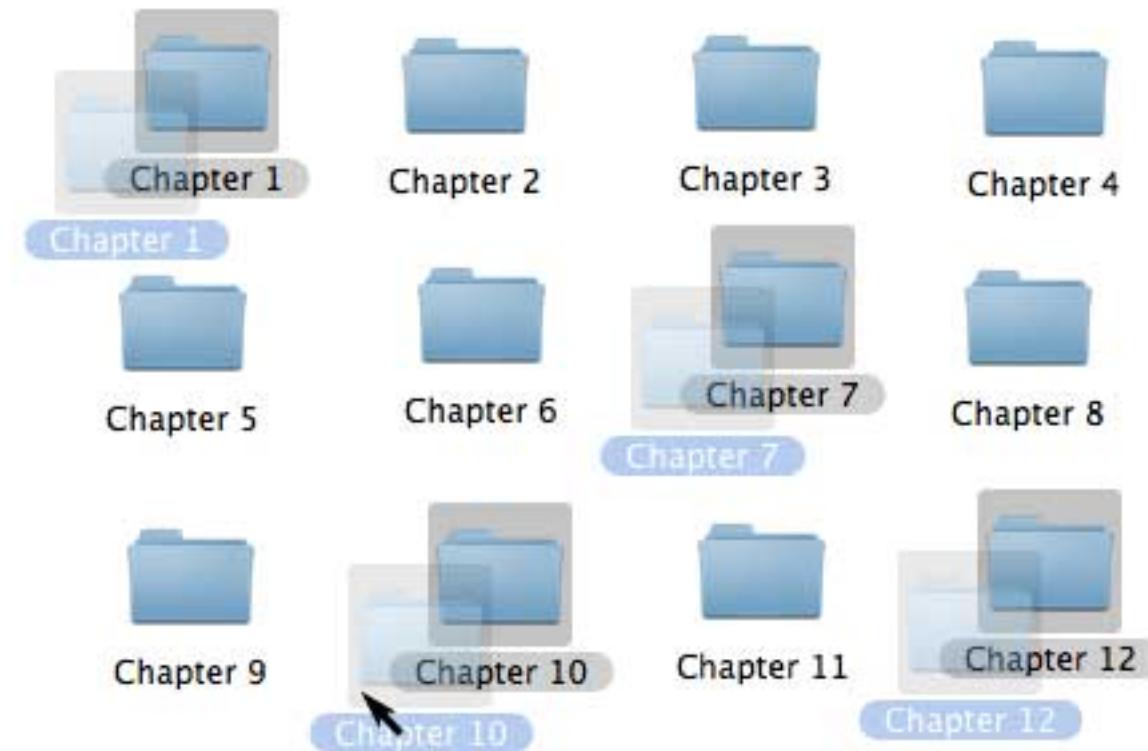
---



- Deliberate foreground/background ambiguity.

# Gestalt Principle: Common Fate

- Elements that **move together** appear grouped.

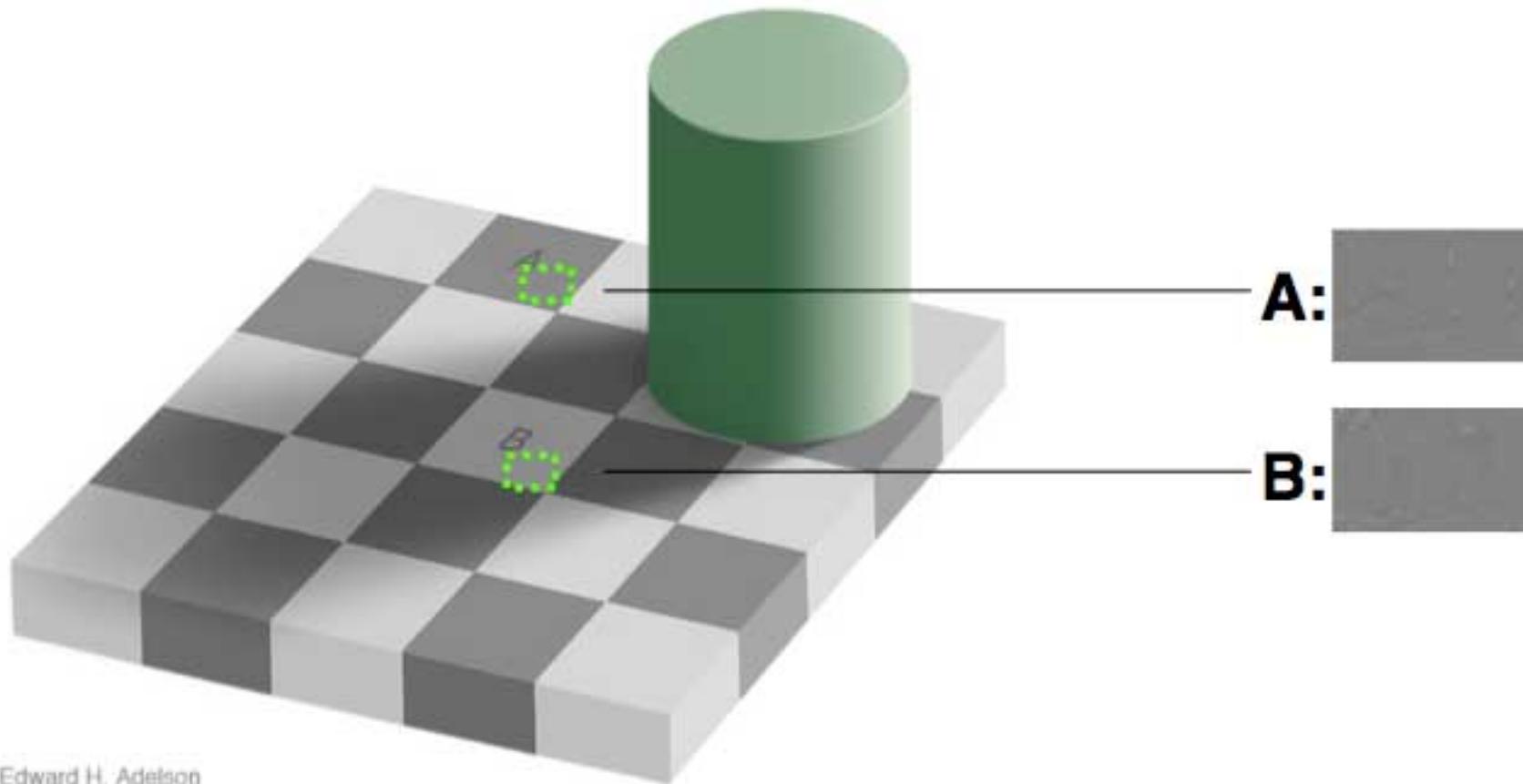


# Contrast vs. Brightness

- Our vision is optimized for contrast, not brightness.



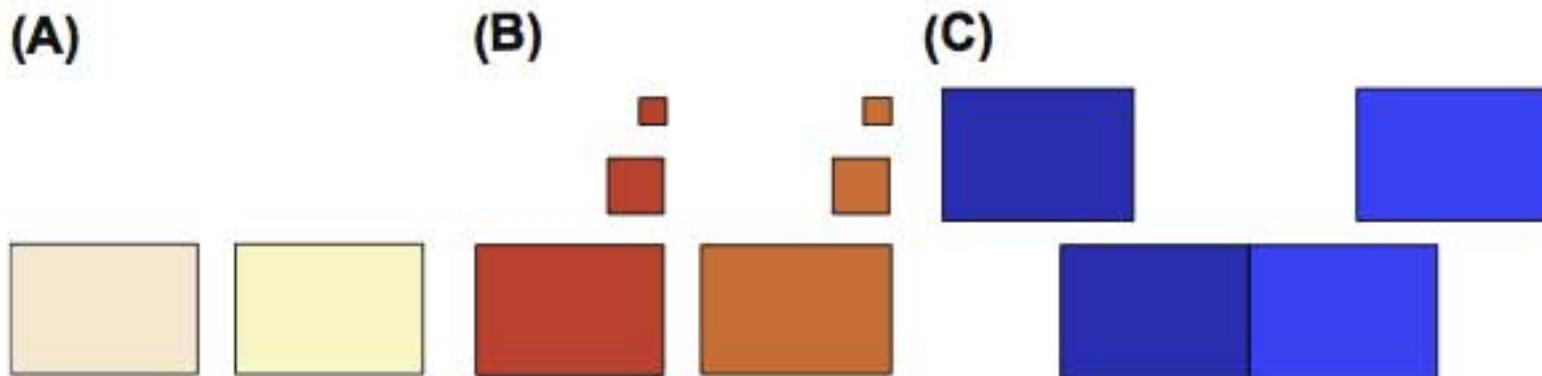
# Contrast vs. Brightness, *cont'd*



Edward H. Adelson

# Color Discrimination

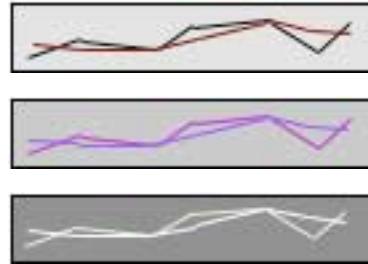
- Factors that affect our ability to distinguish colors:
  - A. paleness
  - B. size
  - C. separation



# Color Blindness

- ❑ Approximately 8% of men and 0.5% of women are color-blind.

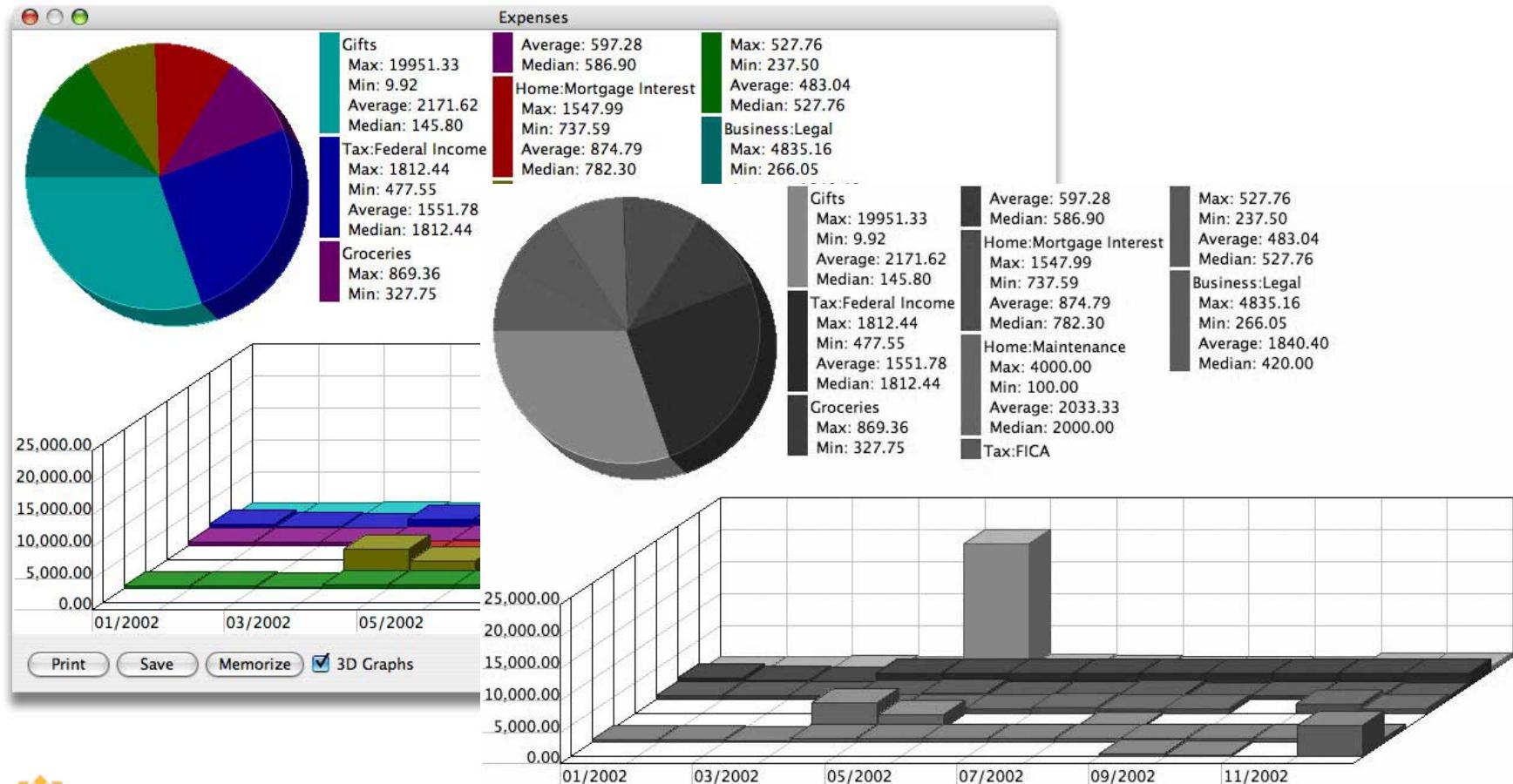
- dark red vs. black
- blue vs. purple
- light green vs. white



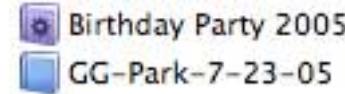
Google Google

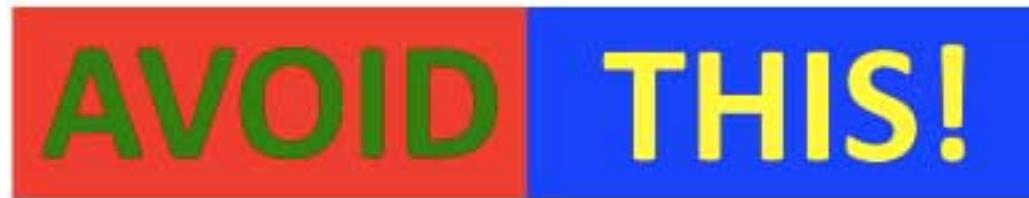
# Color Blindness and UI Design

- Test how your UI looks in grayscale.



# Color Guidelines for UI Design

- Distinguish colors by saturation and brightness, not just hue.
- Use distinctive colors. 
- Avoid color pairs that color-blind people cannot distinguish.
- Use other cues along with color. 
- Separate strong opposing colors.



# Color Guidelines for UI Design, *cont'd*

- Bad:



- Good:



# Rods and Cones and Fovea

---

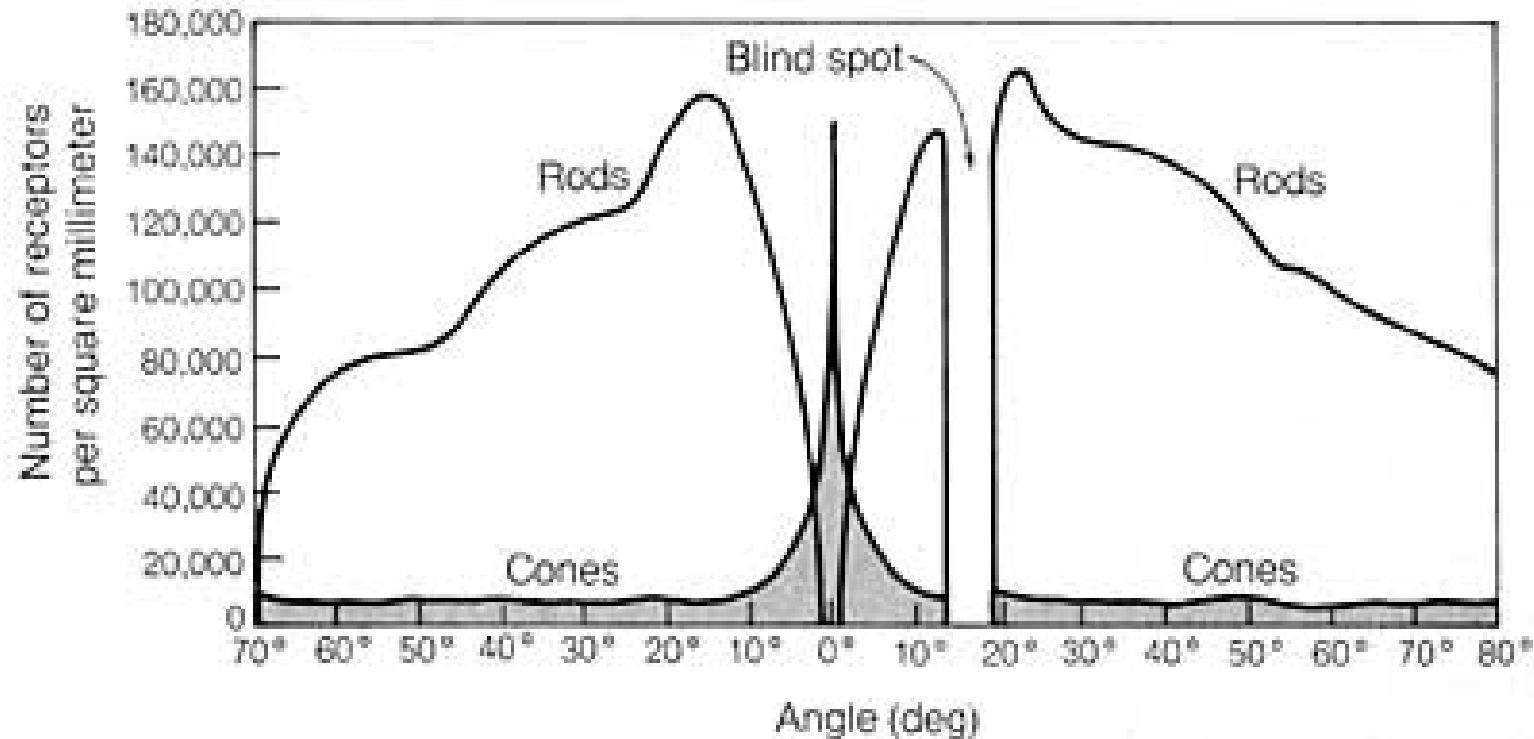
- Our retinas contain **rods** and **cones**
  - rods: detect light levels but not colors
  - cones: sensitive to red, green, and blue colors
  
- **Fovea**
  - A small region in the center of our visual field, about 1% of the retina.

# Our Eyes are Not Like Digital Cameras

---

- Pixel density
  - Each eye has 6 to 7 million retinal cone cells.
  - Fovea: about 158K cone cells per square millimeter
  - Rest of retina: about 9K cone cells per square millimeter
  
- Data compression
  - Fovea: one ganglial neuron cell per cone cell
  - Rest of retina: multiple cone cells per neuron resulting in lossy data compression

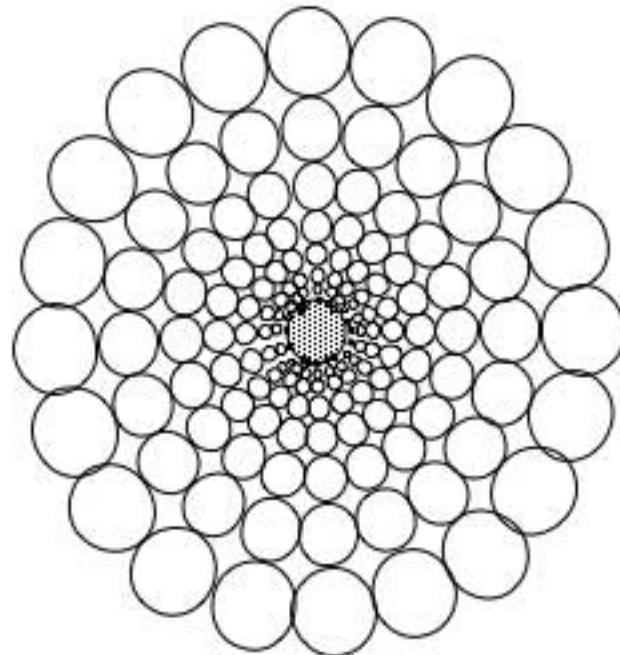
# Our Eyes are Not Like Digital Cameras, *cont'd*



- The rods do better in low light than the cones.
  - Therefore, at night, we see an object better by looking slightly away from it.

# Peripheral Vision

- The resolution of our visual field is high in the center but much lower at the edges.



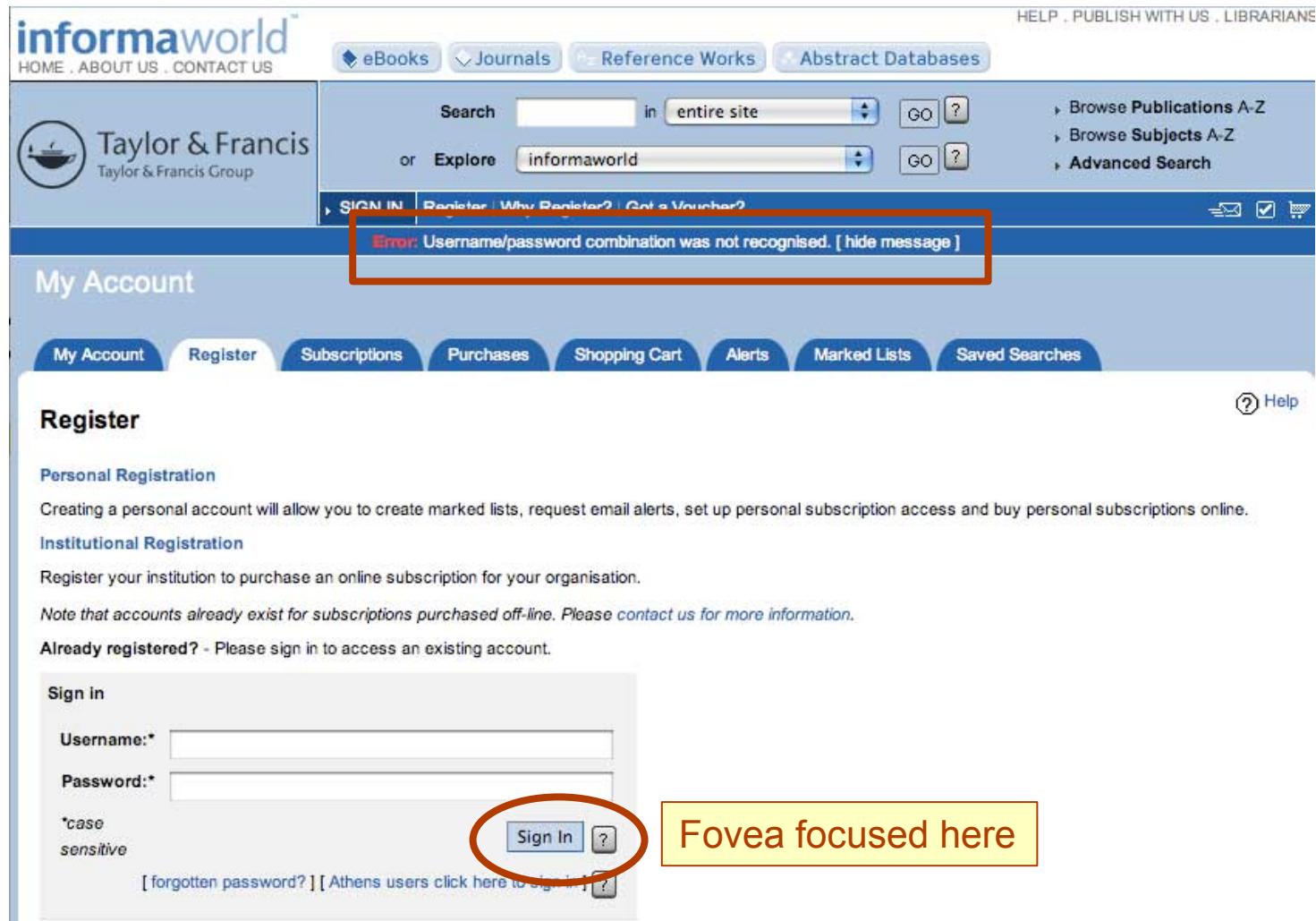
# Peripheral Vision Functions

---

- Guide the fovea
  - Our eyes move automatically move rapidly several times per second to visit all the interesting and crucial elements in our visual field.
  - Our peripheral vision guides the fovea by telling it “where to look”.
- Detect motion
  - Our peripheral vision is good at detecting motion.
- See better in the dark
  - The rods do better in low light than the cones.

# Make Users Notice Error Messages

□ Bad:



# Make Users Notice Error Messages, *cont'd*

□ Bad:

**RETURNING CUSTOMER LOGIN**

Login ID not found.

Login ID:

Password:

Remember my Login ID for faster logins.

**LOGIN**



Designing with the Mind in Mind, 2<sup>nd</sup> ed.  
by Jeff Johnson  
Morgan Kaufmann, 2014

# Make Users Notice Error Messages, cont'd

- Good:

The screenshot shows the 'informaworld' website's 'My Account' section with a 'Register' tab selected. The page displays two registration options: 'Personal Registration' and 'Institutional Registration'. Below these, a note states: 'Note that accounts already exist for subscriptions purchased off-line. Please contact us for more information.' A link for 'Already registered?' leads to the sign-in page.

The sign-in form includes fields for 'Username:' and 'Password:', both of which are highlighted in red, indicating they are required fields. A note below the password field specifies: '\*case sensitive'. To the right of the sign-in button is a red warning icon with an exclamation mark and the text: 'Username/password combination was not recognised.'

# Make Users Notice Error Messages, *cont'd*

- Good:  
Wiggle the display.



# Make Visual Searches Easier

- Make a crucial element “pop” in our peripheral vision to draw the fovea to it.

L	Q	R	B	T	J	P	L	F	<b>B</b> M	R	W	S
F	R	N	Q	S	P	D	C	H	K	U	T	
G	T	H	U	J	L	U	9	J	V	Y	I	A
E	X	C	F	T	Y	N	H	T	D	O	L	L8
G	V	N	G	R	Y	J	G	Z	S	T	6	S
3	L	C	T	V	B	H	U	S	E	M	U	K
W	Q	E	L	F	G	H	U	Y	I	K	D	9

Hard to find the Z.

Easier to spot the bold G.

G	T	H	U	J	L	U	9	J	V	Y	I	A
L	Q	R	B	T	J	P	L	F	<b>B</b> M	R	W	S
3	L	C	T	V	B	H	U	S	E	M	U	K
F	R	N	Q	S	P	D	C	H	K	U	T	
W	Q	E	L	F	G	H	B	Y	I	K	D	9
G	V	N	<b>G</b>	R	Y	J	G	Z	S	T	6	S
E	X	C	F	T	Y	N	H	T	D	O	L	L8

Designing with the Mind in Mind, 2<sup>nd</sup> ed.  
by Jeff Johnson  
Morgan Kaufmann, 2014



Computer Science Dept.  
Spring 2015: March 5

CS 235: User Interface Design  
© R. Mak

# Make Visual Searches Easier, *cont'd*

L	Q	R	B	T	J	P	L	F	B	M	R	W	S
F	R	N	Q	S	P	D	C	H	K	U	T		
G	T	H	U	J	L	U	9	J	V	Y	I	A	
E	X	C	F	T	Y	N	H	T	D	O	L	L	8
3	L	C	T	V	B	H	U	S	E	M	U	K	
G	V	N	G	R	Y	J	G	Z	S	T	6	S	
W	Q	E	L	F	G	H	U	Y	I	K	D	9	

Hard to count L's.

Easier to count red letters.

W	Q	E	L	F	G	H	U	Y	I	K	D	9	
F	R	N	Q	S	P	D	C	H	K	U	T		
3	L	C	T	V	B	H	U	S	E	M	U	K	
G	T	H	U	J	L	U	9	J	V	Y	I	A	
L	Q	R	B	T	J	P	L	F	B	M	R	W	S
E	X	C	F	T	Y	N	H	T	D	O	L	L	8
G	V	N	G	R	Y	J	G	Z	S	T	6	S	

Designing with the Mind in Mind, 2<sup>nd</sup> ed.  
by Jeff Johnson  
Morgan Kaufmann, 2014



Computer Science Dept.  
Spring 2015: March 5

CS 235: User Interface Design  
© R. Mak

# Make Visual Searches Easier, cont'd

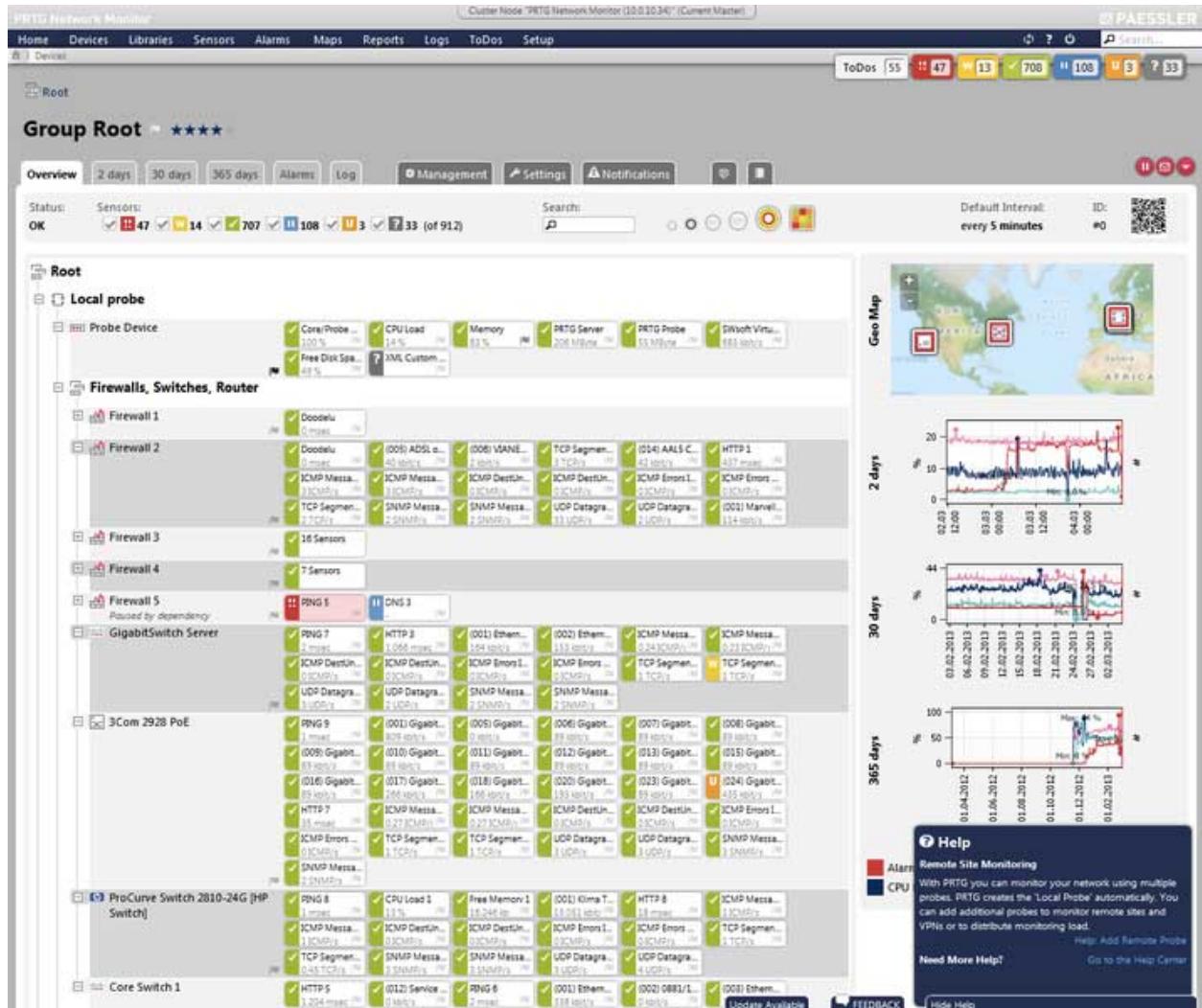


Easy to spot traffic jams.



# Make Visual Searches Easier, cont'd

Easy to spot  
network problems.



Designing with the Mind in Mind, 2<sup>nd</sup> ed.  
by Jeff Johnson  
Morgan Kaufmann, 2014



Computer Science Dept.  
Spring 2015: March 5

CS 235: User Interface Design  
© R. Mak

# Affordance

---

- The **property** of affordance for a control.
  - The design of the control should suggest (afford) how it should be operated by a user.
  - Donald Norman: Affordance is “strong clues to the operation of things. When affordances are taken advantage of, the user knows what to do just by looking: no picture, label, or instruction is required.”
- Perceived affordance: What action a user thinks can be done with a control.
  - The control **affords** the action.

# Feedback

---

- Feedback is information sent back to the user that a control has accomplished its action.
- It should be obvious to the user of a control that the control's action has taken place.

# Standard and Nonstandard Affordances

---

- Standard affordances
  - Raised-edge button: affords pressing it
  - Slider: affords dragging it
  - Text field: affords typing
  - Blue underlined text: affords clicking
- A nonstandard buttons and other control can present an unusual user interface.
  - But is it well designed?
  - Good affordance and feedback?

# Affordances Example: Apple GarageBand

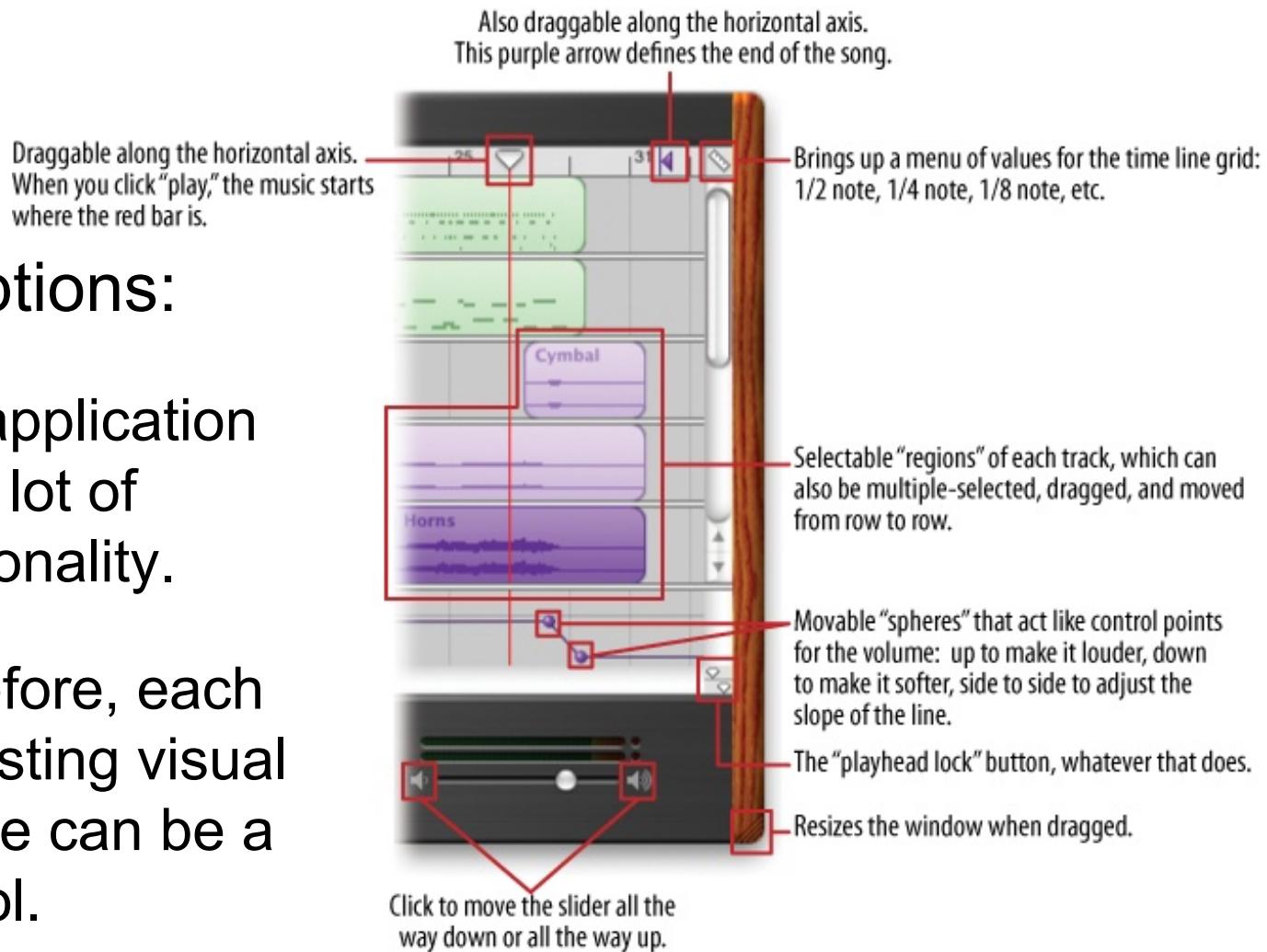


What controls are there between the red vertical line and the right edge of the tool?

# Affordances Example, cont'd

## □ Assumptions:

- This application has a lot of functionality.
- Therefore, each interesting visual feature can be a control.



Lecture #14

# CS 235: User Interface Design

## March 10 Class Meeting

---

Department of Computer Science  
San Jose State University

Spring 2015  
Instructor: Ron Mak

[www.cs.sjsu.edu/~mak](http://www.cs.sjsu.edu/~mak)



# Schedule for Prototype Presentations

---

- Thursday, March 12
  - Innovative Designers
  - Team Four
  - Thundercats
  
- Tuesday, March 17
  - Unknown
  - UXability
  - X Factor

# Prototype Presentation Reports

---

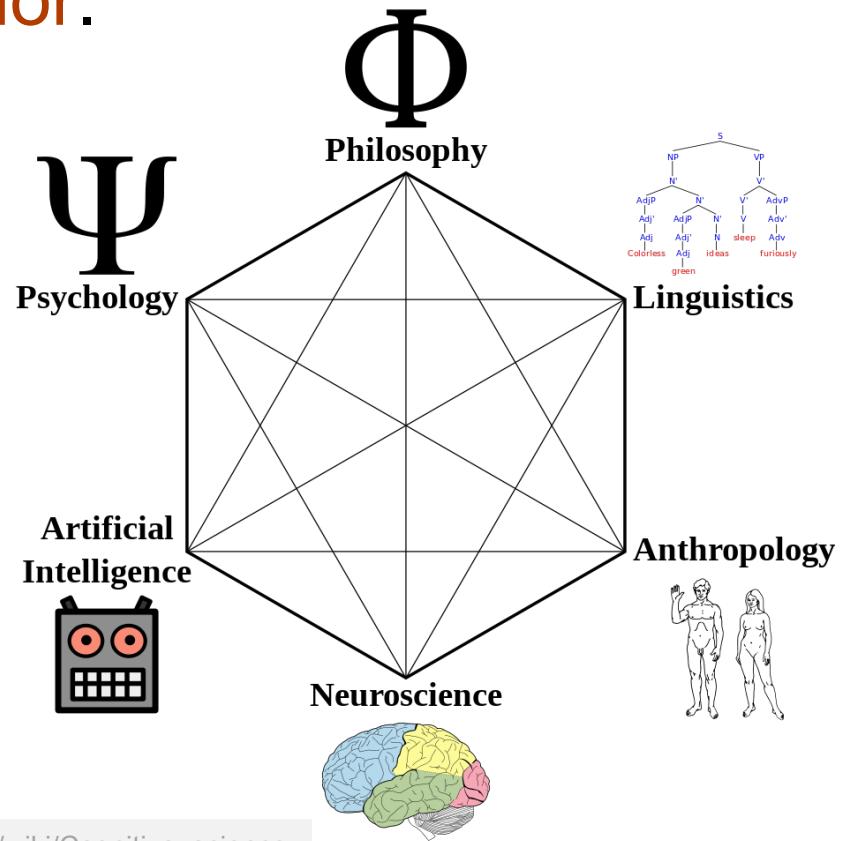
- A short report:
  - What is your application.
  - What UI design patterns did you use and why.
    - Describe up to 5 patterns.
  - Screen shots that illustrate your design patterns.
- Your PowerPoint slides, if any.
- Your prototype code, if runnable by anyone.
  - Include installation and how to run instructions.
- Due Friday, April 3 (after Spring Break)

# Cognitive Science

- The interdisciplinary study of the **mind**, **intelligence**, and behavior.

- How does the mind process information?

- perception
- language
- memory
- reasoning
- emotion



[http://en.wikipedia.org/wiki/Cognitive\\_science](http://en.wikipedia.org/wiki/Cognitive_science)

# Hand-Eye Coordination

---

- Pointing at objects on a display and moving pointers along constrained paths follow consistent, quantitative laws.



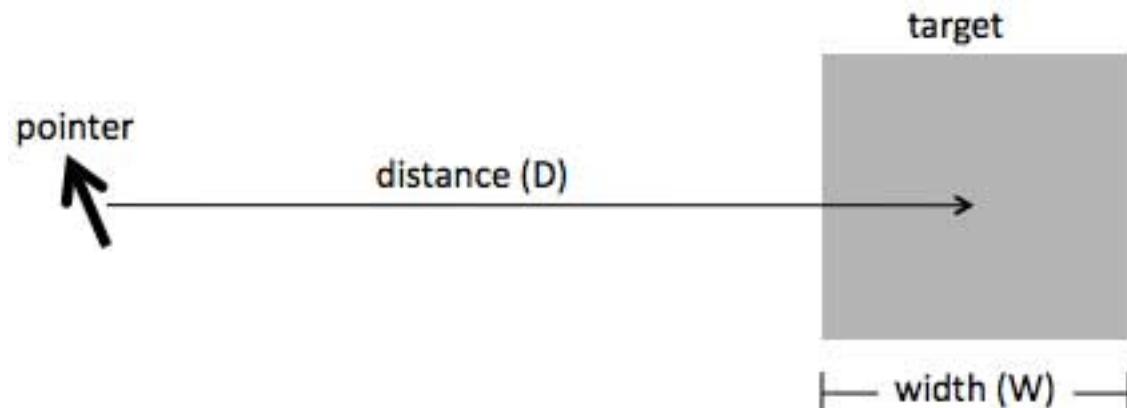
# Fitts's Law for Pointing at Displayed Targets

$$T = a + b \log_2 \left( 1 + \frac{D}{W} \right)$$

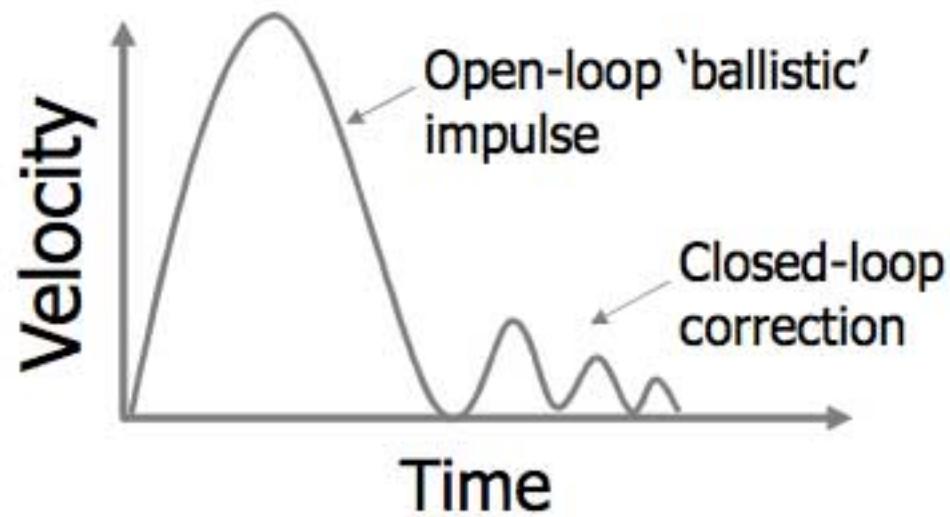
## □ Where:

- $T$  = time to move to the target
- $D$  = distance to the target
- $W$  = width of the target
- $a$  = a measure of the ease of starting and stopping the movement
- $b$  = a measure of the average difficulty of moving the hand and pointing the device

# Fitts's Law, *cont'd*



- Velocity over time as the pointer moves to the target:



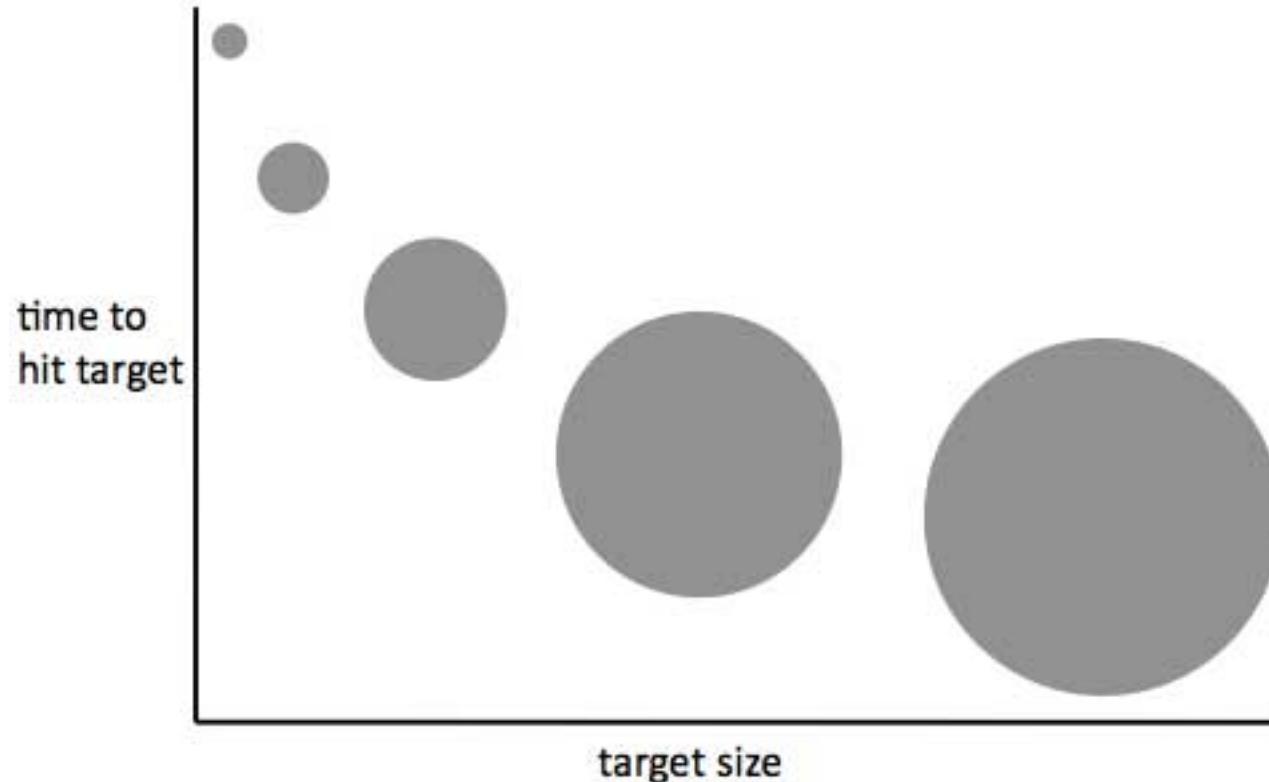
# Fitts's Law, *cont'd*

---

- Users hit on-screen targets faster the **closer and larger** the targets are.
- The more the distance decreases or the target size increases, the less the decrease in pointing time.
  - Beyond a certain size, making the target even larger adds little benefit.
  - Below a certain distance, making the target closer doesn't help much.

# Fitts's Law, *cont'd*

---



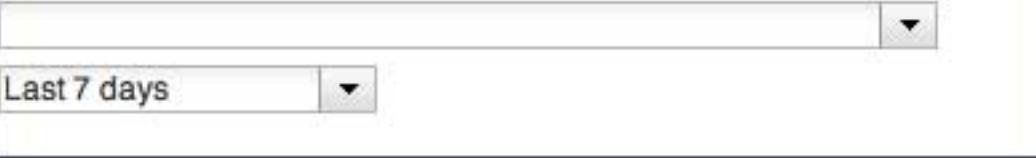
# Targets Too Small

Recent Banking Activity

list graph

Please select an account and click on list or graph to view recent activity

Last 7 days

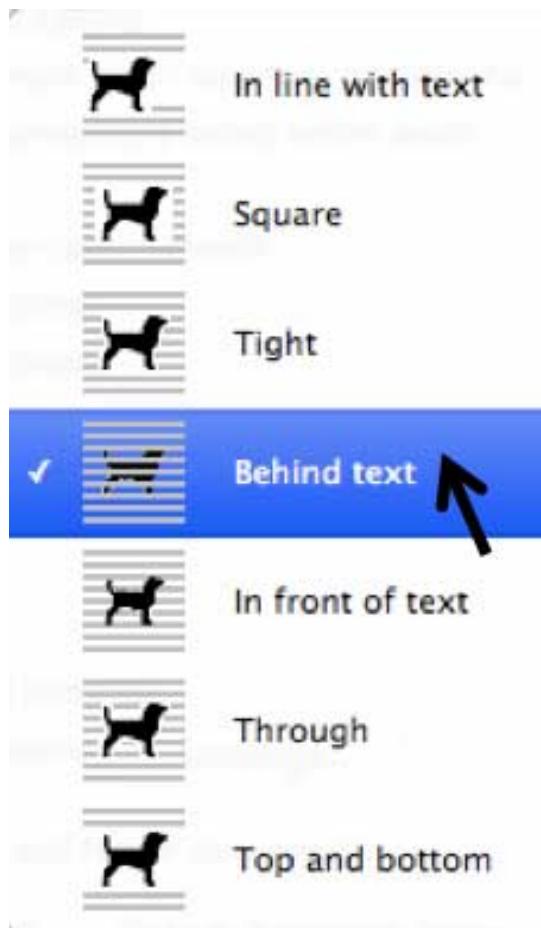


Designing with the Mind in Mind, 2<sup>nd</sup> ed.  
by Jeff Johnson  
Morgan Kaufmann, 2014

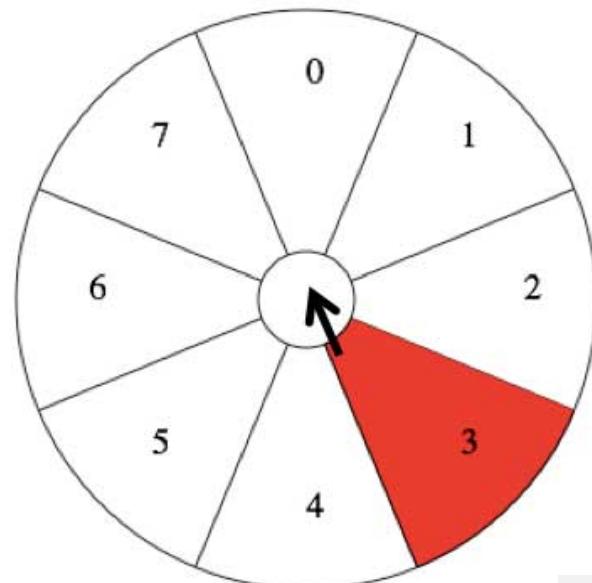


The website features a large banner at the top with the conference logo, "AGING IN AMERICA 2011 ASA CONFERENCE April 26-30 · San Francisco", and a scenic view of the San Francisco skyline at sunset with the Golden Gate Bridge. Below the banner, the text "ANNUAL CONFERENCE OF THE AMERICAN SOCIETY ON AGING" is displayed. On the left side, there's a sidebar with "Aging in America Sponsors" featuring the AARP logo, and buttons for "Home", "General Information", and "Register Now!". The main content area has a large blue callout box with the text "april 26-30 is just around the corner... ARE YOU GETTING EXCITED?" and four categories: "GET READY!", "ASA AWARDS", "CONTINUING EDUCATION", and "WHO'S EXHIBITING?". At the bottom of this box, there's a link to a web seminar recording with a 10% discount offer. To the right of the main content, there's a "Quick Links" sidebar with links to "Schedule at a Glance", "General Sessions", "National Forums", "Affiliated Organization Programs", "Constituent Group Programs", "Critical Issues in Aging", "Networking Events", and "Site Visits".

# Larger Target Sizes



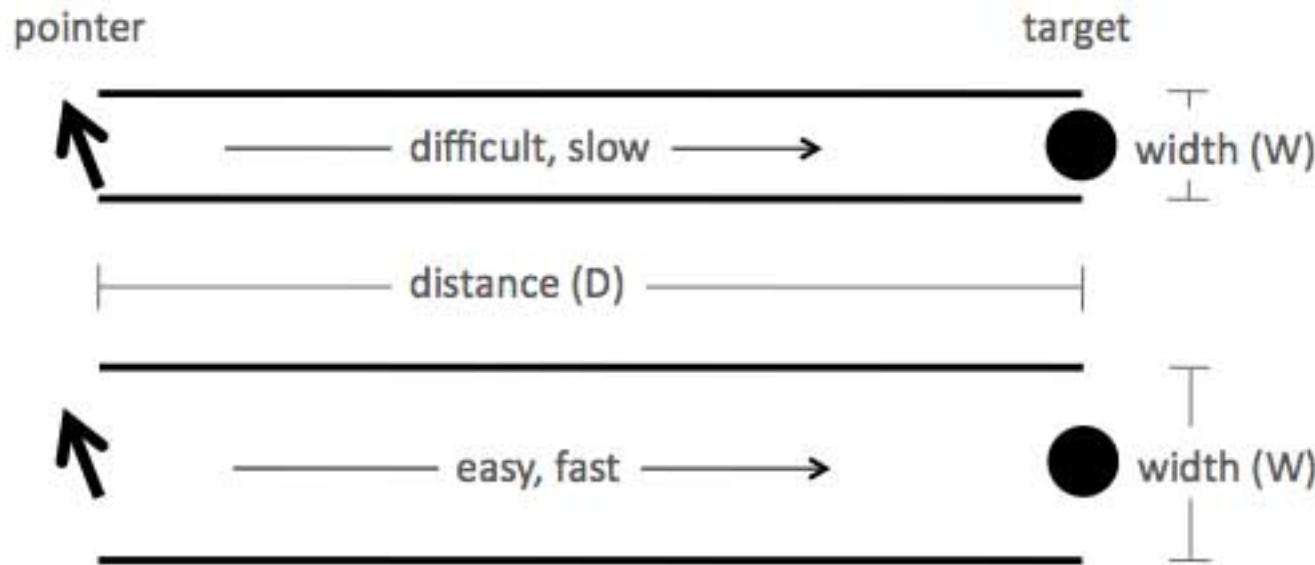
The screenshot shows the ElderLawAnswers website. At the top right are links for Contact Us, About Us, For Attorneys, a search bar, and a 'go' button. Below the header are logos for WSJ, Forbes, and CBS MarketWatch. The main navigation menu includes Learn More, Ask a Question, Tools & Calculators (which is highlighted with a yellow dotted border), Find an Attorney, About Us, and For Attorneys. A large banner features a photo of an elderly couple and the text 'Find Local Elder Law Assistance'. To the right are sections for 'Browse Local Elder Law Attorneys Listings' (with a photo of G. Archer Bakerink) and a 'Search for Elder Law Attorneys' form.



CS 235: User Interface Design  
© R. Mak

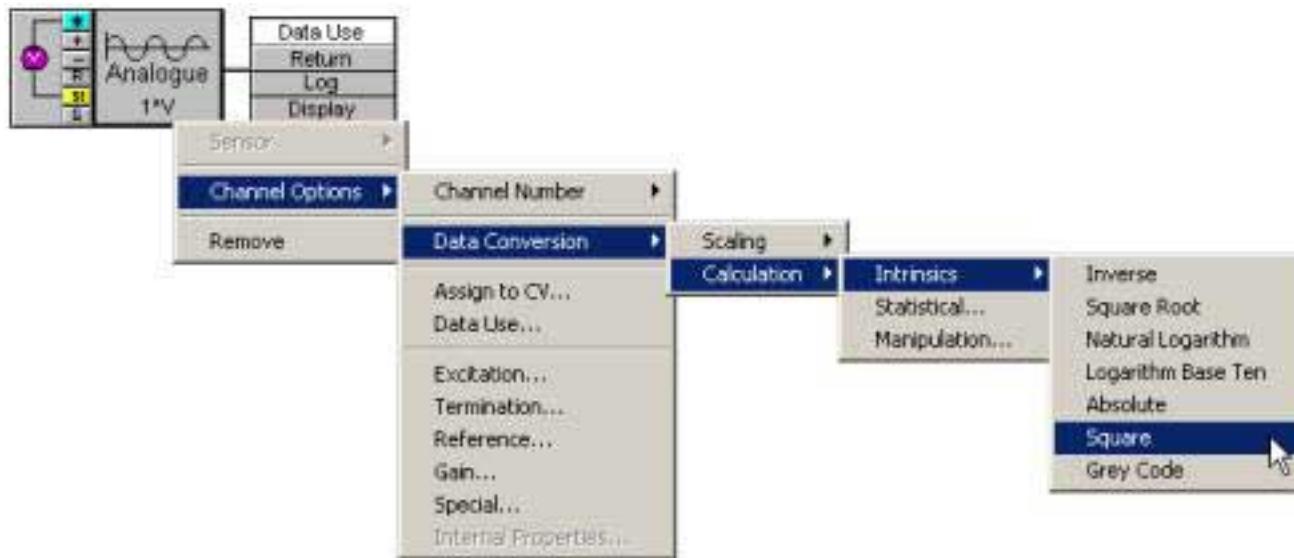
# Steering Law

- If you must keep a pointer within a certain **confined path** while moving it to a target, then the wider the path, the faster you can move the pointer to the target.



# Steering Law, *cont'd*

- Difficult: Multi-level pull-right menus:



# Steering Law, *cont'd*

## Original

Africa & Middle East	Asia	Australia & South
Africa	Botswana	
Middle East	Egypt	
	Ghana	
	Kenya	
	Libya	
	Madagascar	
	Malawi	
	Mali	
	Morocco	
	Namibia	
	Seychelles	
	South Africa	
	Tanzania	
	Tunisia	
	Uganda	
	Zambia	
	Zimbabwe	

## Improved

Africa & Middle East	Asia	Australia & South
Africa	Botswana	
Middle East	Egypt	
	Ghana	
	Kenya	
	Libya	
	Madagascar	
	Malawi	
	Mali	
	Morocco	
	Namibia	
	Seychelles	

# How Your Brain Learns

---

- **Brain plasticity:** Your brain learns a new perception or behavior by rewiring itself.
  - Neurons that previously fired independently connect into a network and fire in concert.
  - Neurons that were formerly in one network can join another network.
-

# How Your Brain Learns, *cont'd*

---

- You perform the first few times of an activity in a highly controlled and conscious manner.
- The neural network forms after you've practiced. Then the activity becomes automatic.
- We learn faster when:
  - Practice is frequent, regular, and precise.
  - Operation is task focused, simple, and consistent.
  - Vocabulary is task focused, familiar, and consistent.
  - Risk is low.

# Frequency of Practice

---

- Consider **frequency of practice** when designing the UI for an interactive system.
- Example: Bank ATM
  - Used by a person every few days or weeks.
  - Present a short list of goals (withdraw, deposit, etc.)
  - Guide the user to achieve a goal.
- Example: Email reader
  - Used frequently by a person.
  - Help and tips can be shown on demand only.

# Regularity of Practice

---

- Habits form faster if practiced regularly, such as daily.
- Skipping a practice session now and then isn't detrimental.
- Skipping a lot of practice significantly slows habit formation.
- Therefore, design your application's UI to encourage people to **use it regularly**.

# Precision of Practice

---

- Precise practice trains neuron networks to fire in concert with less “noise”.
- The more **carefully and precisely** a person practices an activity, the more systematic and predictable the activation of the neural network.
- Design the UI of an application to
  - Help people become precise, such as by providing guides and grids.
  - Encourage people to use it purposefully and carefully by providing a clear conceptual model.

# Task Focused, Simple, and Consistent

---

- “**Gulf of execution**”: The gap between what an application user wants and the application’s operations.
- Design applications that match what users want to do.
  - Perform a task analysis.
  - Design a task-focused conceptual model consisting mainly of object-action pairs.
  - Design a UI based strictly on the task analysis and the conceptual model.

# Task Analysis

---

- What are the users' goals for this application?
- What tasks should the application support?
- Which tasks are common or important?
- What are the steps of each task?
- What are the results and outputs of each task?
- How are the results and outputs used?
- Who does which task?
- What terminologies do they use?
- What problems do users have performing tasks?

# Conceptual Model

---

- Users build a model in their minds of how to use an application and how the application works.
- A **conceptual model** of an application is the one that designers want users to understand.
- Try to match as closely as possible the models of the users and the designers.
- Focus the conceptual model on tasks.

# Conceptual Model, *cont'd*

---

- Make the model as simple as possible with fewer concepts for users to learn.
- Example of excess complexity:  
Separate concepts that are too similar.
  - **Membership:** Your company paid for a customer support service.
  - **Subscription:** Your company subscribed to a customer support newsletter.
  - **Access:** Which areas of the customer support website users in your company can access.
  - **Entitlements:** Services provided at each level.

# Consistency

---

- Users learn faster if an application is **consistent and predictable**.
- Unpredictability forces a user to constantly be learning something new.
- UI designer's goal: A conceptual model that is task focused, simple, and consistent.
  - Conceptual consistency
  - Keystroke consistency

# Task Focused Vocabulary

- Example: Website for investment transactions that allows users to save templates locally or to a database on a remote server.

Bad: Not task focused	Good: Task focused
local	private
database	shared, public

- Example: An insurance company for military veterans asks for a database instead of for a country.

**i**casualties.org  
IRAQ COALITION CASUALTY COUNTS

Please Select Database:

- Iraq
- Afghanistan



# Avoid Geek Speak

<http://www.bu.edu>

**abort**

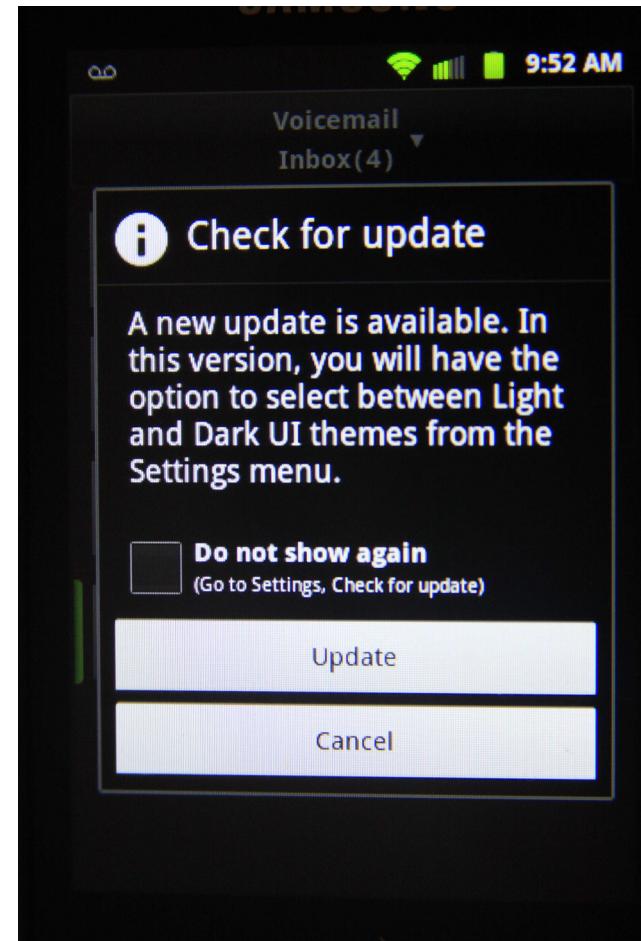
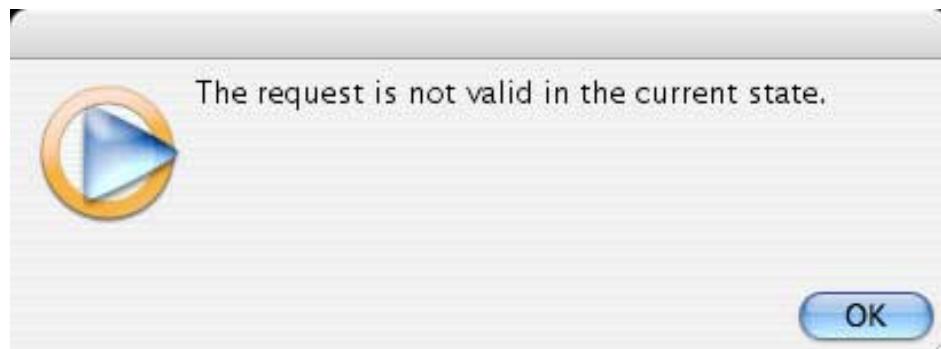
ROM

1000Base-T  
firewire

ADSL modem

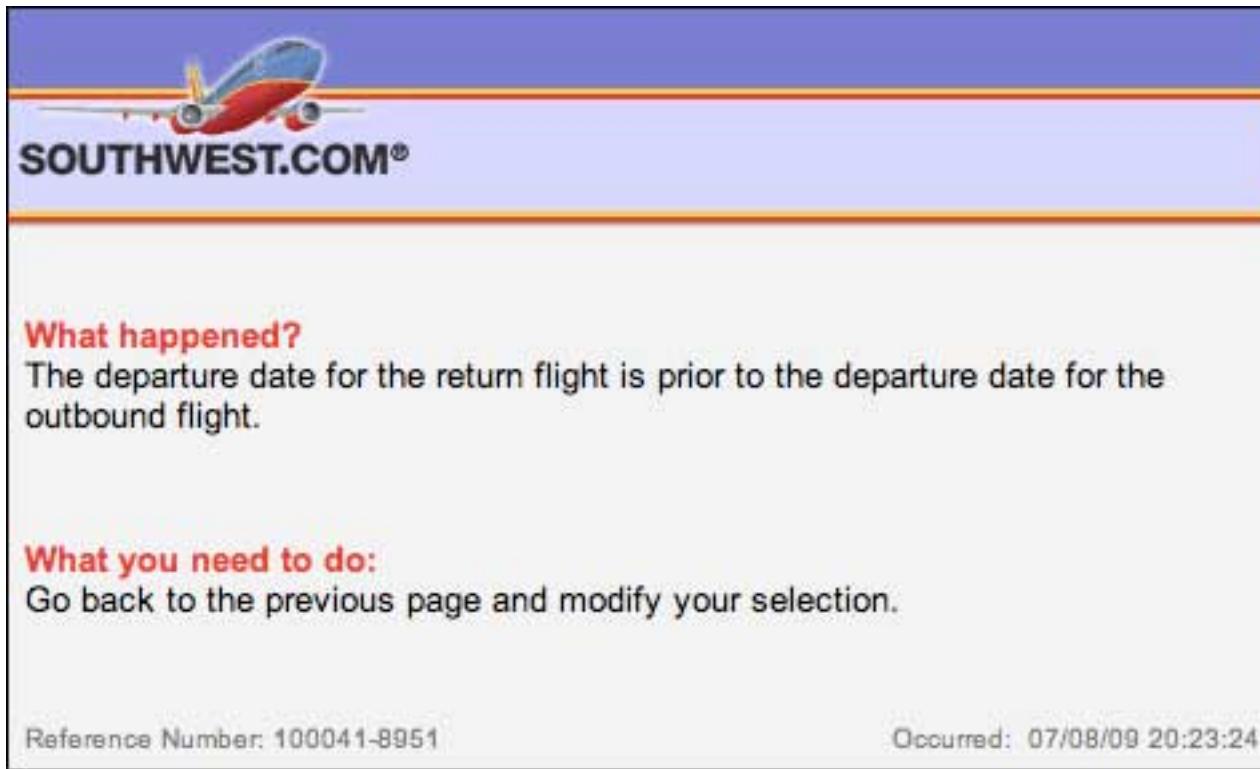
John23764@AOL.com

index.php?display=EIRMapNation&tab=ALL



# Avoid Geek Speak, *cont'd*

- Example: Error messages that are task focused and clear, which fosters learning.



The screenshot shows a Southwest.com error message. At the top is the Southwest logo and the word "SOUTHWEST.COM®". Below this is a large empty input field. Underneath the field, the error message is displayed in two parts: "What happened?" and "What you need to do:". The "What happened?" section states: "The departure date for the return flight is prior to the departure date for the outbound flight." The "What you need to do:" section states: "Go back to the previous page and modify your selection." At the bottom of the message, there are two small pieces of text: "Reference Number: 100041-8951" and "Occurred: 07/08/09 20:23:24".

**What happened?**  
The departure date for the return flight is prior to the departure date for the outbound flight.

**What you need to do:**  
Go back to the previous page and modify your selection.

Reference Number: 100041-8951      Occurred: 07/08/09 20:23:24

# Use Consistent Terminology

- Terminology is consistent when each concept has **one and only one name**.
  - Same name, same thing;  
different name, different thing.

- Example of inconsistent terminology:

**What are the differences between Windows-based and UNIX-based platforms?**  
For detailed information on choosing between these two operating systems, please visit our Windows or UNIX page.

For a quick look at which features and programs each platform supports, please consult the chart below:

Feature	Standard	ASP
Microsoft FrontPage Extensions	YES	YES
RealVideo and RealAudio	YES	NO
ASP (Active Server Pages)	NO	YES
ADO (ActiveX Data Objects)	NO	YES
ODBC Data Sources	NO	YES
Windows Media Server	NO	YES
PHP	YES	NO
MySQL Databases	YES	NO
Free ready-to run scripts, such as a hit counter, forum, e-mail form, blog, and guestbook	YES	NO

Designing with the Mind in Mind, 2<sup>nd</sup> ed.  
by Jeff Johnson  
Morgan Kaufmann, 2014



Computer Science Dept.  
Spring 2015: March 10

# Lower the Risk of Using the Application

---

- Users will explore more and learn more if the application has **less risk**.
- Users may be afraid to use unfamiliar features.
- Don't make it easy for users to make mistakes.
  - Prevent user errors where possible.
  - Make errors easy to detect.
  - Tell users clearly what they did wrong.
  - Allow users to undo wrong actions easily.

# The Perception of Time

---

- Perceived responsiveness of an application is more important than effectiveness.
- An application can have poor responsiveness even if it is fast.
  - How?

# Responsive Application

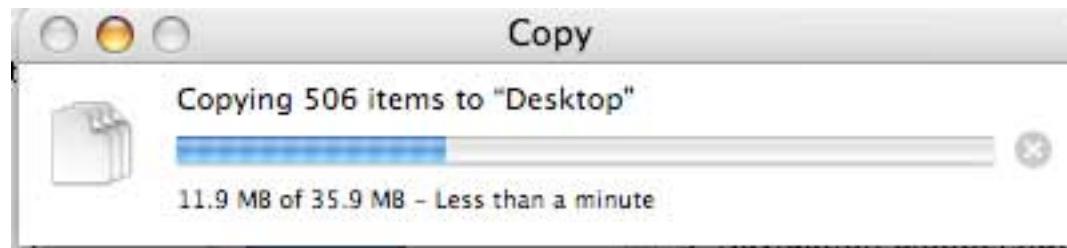
---

- Keep up with the user.
- Don't make the user wait unexpectedly.
- Keep the user informed about the application's status even if it cannot fulfill the user's request immediately.
- Let the user know immediately that input was received.
- Indicate how long an operation will take.

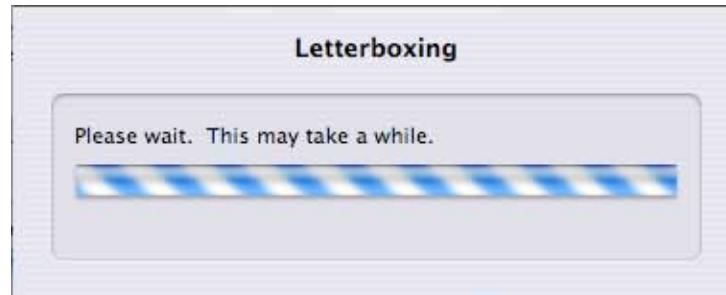
# Responsive Application, *cont'd*

- Allow the user to do other things while waiting.
- Manage queued events intelligently.
- Perform low-priority tasks in the background.
- Anticipate a user's most common requests.

- Good:



- Poor:



# Busy Indicators

---

- Display **busy indicators** for any operation that blocks further user actions even if the function executes quickly.
  - Example: hourglass cursor

# Progress Indicators

---

- Display for any operation that takes longer than a few seconds.
- Show work remaining, not just work completed.
- Show total progress, not just the current step.
- Show percentage of operation completed.
  - Start at 1%, not 0%.
- Don't display 100% for more than 1-2 seconds.
- Show smooth progress, not erratic bursts.
- Use human scale precision: “about 4 minutes”, not “240 seconds”

# Delays

---

- Delays **between** unit tasks of a large operation are less annoying than delays **within** unit tasks.
- A user keeps the ultimate goal of an operation in working memory.
  - Naturally relaxes between unit tasks.
- Unexpected delays during unit tasks can cause a user to lose track what is happening.

# Task Closure

---

- **Task closure:** The difference between the impact of delays during and between unit tasks.
- **High level of closure**
  - Less sensitive to response time delays.
  - Example: Saving a file
- **Low level of closure**
  - Most sensitive to response time delays.
  - Example: Typing a character and seeing it echoed on the screen.

# Display Important Information First

---

- Example: Open a document
  - Display the first page immediately rather than waiting for the entire document to load.
- Example: Search results
  - Display found items immediately while continuing to search for more.
- Example: Display a large downloaded image
  - Good: Display a complete low-resolution version first and then progressively improve the resolution.
  - Bad: Progressively display the full-resolution image.

# Display Important Information First, *cont'd*

---

Good



Bad



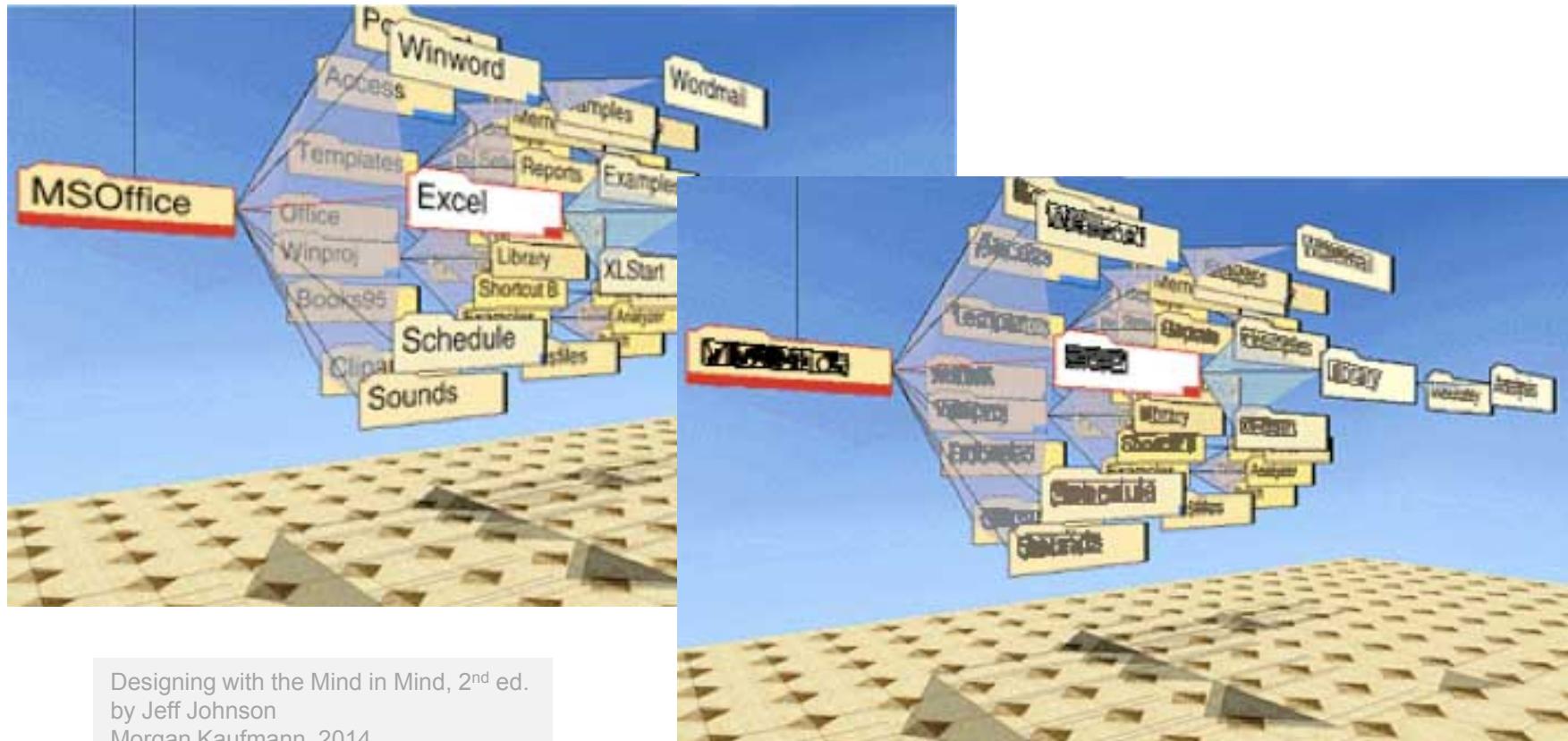
# Animated Tasks

---

- A user action may require rapid successive adjustments with hand-eye coordination until a goal is achieved.
- If heavy computation is involved such as while moving a large image, feedback may lag user actions.
- **Fake the feedback** by moving only a rubber-band outline of the image.

# Animated Tasks, cont'd

- Example: A rotating “cone tree” renders labels as blobs.



# Work Ahead

---

- Stay ahead of the user whenever possible.
- An application can use low-load periods to pre-compute responses to high-probability requests.
- Example: Search for a word in a document
  - While displaying a find, search for the next occurrence of the word.
- Example: Display a document
  - Render the next page while displaying the current page.

# Prioritize User Actions

---

- Process user actions by priority, not by the order they were received.
- An interactive application should reorder tasks in its queue.
  - Simply executing tasks in the order they were received can waste resources and create extra work.
  - Reordering tasks can make work more efficient.
- Example: Web browser
  - If the user clicks the “back” or “stop” button, immediately abort loading the current page.

# Timely Websites

---

- Minimize the number and size of images.
- Provide quick-to-display thumbnails or overviews of large amounts of content.
  - Allow the user to drill down for more detail.
- Use Cascading Style Sheets (CSS).
- Use built-in browser dialog boxes.
- Use browser-side scripting and applets.