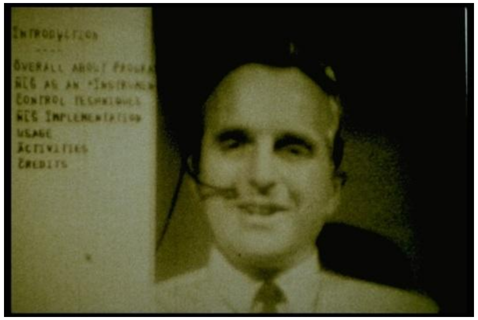
CS160: User Interface Design, Prototyping and Evaluation

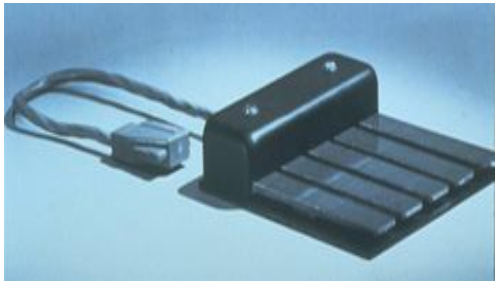
Prof: John Canny
GSIs: Pablo Paredes
Anuj Tewari

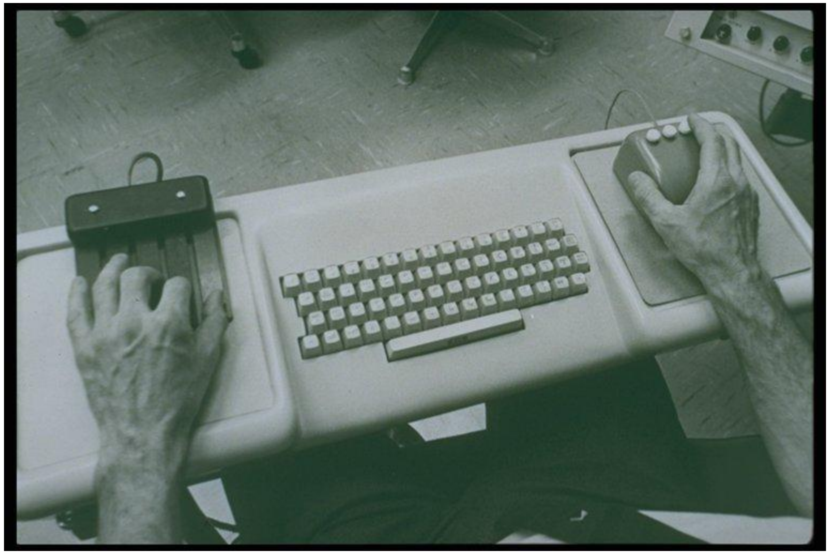
Where we were (1968)

Engelbart demo, NLS (oNLine System)

- Video screen, chording keyboard, mouse, videoconferencing, hyperlinking, word processing, email
- User studies

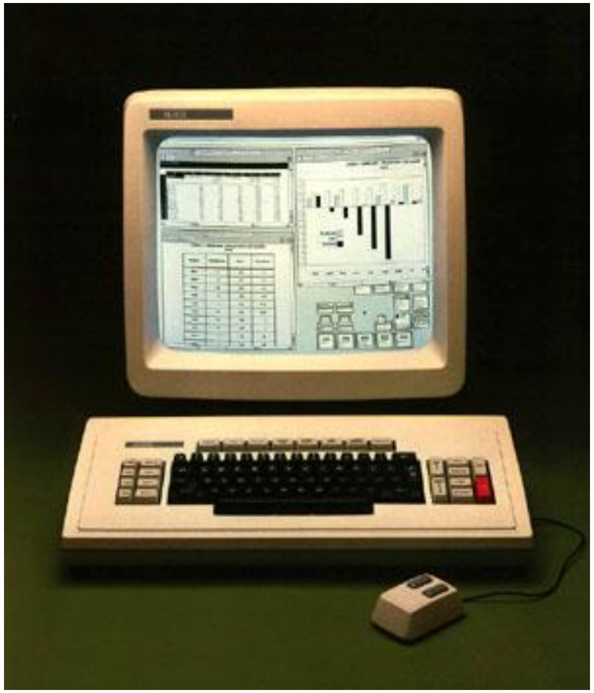


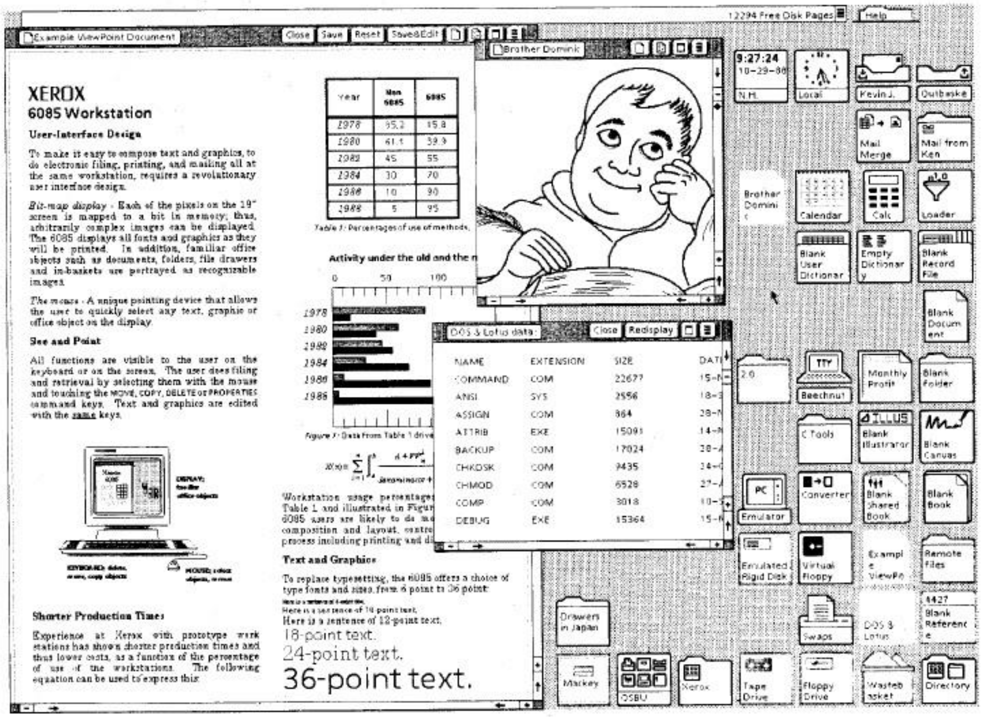




Video: 10:54 – 17:00

Where we were (1981)





XEROX
6085 Workstation

User-Interface Design

To make it easy to compose text and graphics, to do electronic filing, printing, and mailing all at the same workstation, requires a revolutionary user interface design.

Bit-map display - Each of the pixels on the 19" screen is mapped to a bit in memory; thus, arbitrarily complex images can be displayed. The 6085 displays all fonts and graphics as they will be printed. In addition, familiar office objects such as documents, folders, file drawers and in-baskets are portrayed as recognizable images.

The mouse - A unique pointing device that allows the user to quickly select any text, graphic or office object on the display.

See and Point

All functions are visible to the user on the keyboard or on the screen. The user does filing and retrieval by selecting them with the mouse and touching the MOVE, COPY, DELETE or PROPERTIES command keys. Text and graphics are edited with the same keys.

Shorter Production Times

Experience at Xerox with prototype work stations has shown shorter production times and thus lower costs, as a function of the percentage of use of the workstations. The following equation can be used to express this:

Year	Non 6085	6085
1978	95.2	15.8
1980	61.1	39.9
1982	45	55
1984	30	70
1986	10	90
1988	5	95

Table 1: Percentages of use of methods.

Figure 1: Data from Table 1 drive

$$X(x) = \sum_{i=1}^{n} \int_{a}^{b} \frac{d + PP_{n}^{j}}{denominator +}$$

Workstation usage percentage
Table 1 and illustrated in Figur
6085 users are likely to do m
composition and layout, centr
process including printing and di

Text and Graphics

To replace typesetting, the 6085 offers a choice of type fonts and sizes, from 6 point to 36 point:

[illegible]
Here is a sentence of 10-point text,
Here is a sentence of 12-point text.
18-point text.
24-point text.
36-point text.

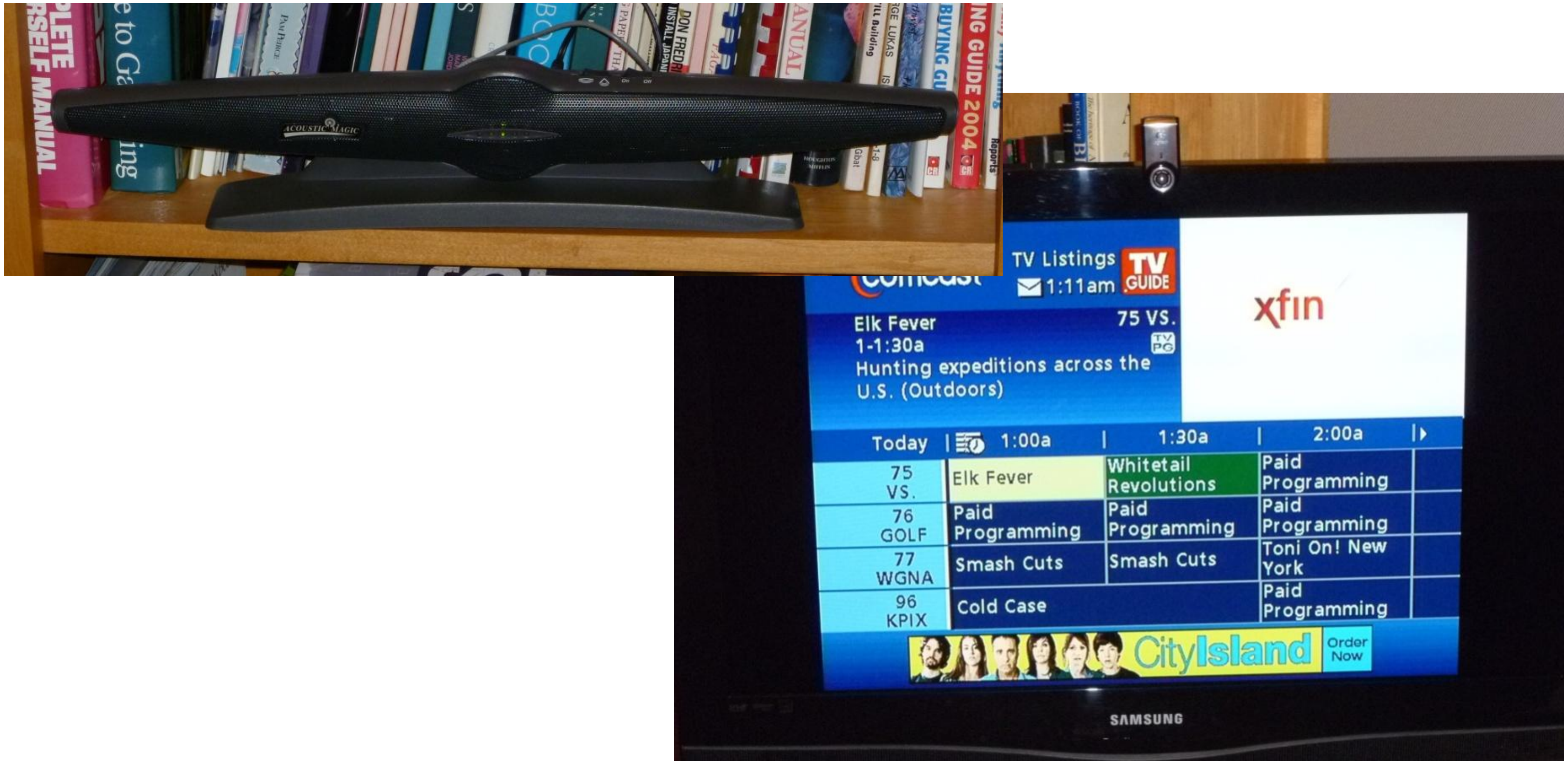
NAME	EXTENSION	SIZE	DATI
COMMAND	COM	22677	15-N
ANSI	SYS	2556	18-3
ASSIGN	COM	864	28-N
ATTRIB	EXE	15091	14-N
BACKUP	COM	17024	28-A
CHKDSK	COM	9435	24-C
CHMOD	COM	6528	27-A
COMP	COM	3018	10-5
DEBUG	EXE	15364	15-N

Xerox Star: Bitmapped display, windows, icons, menus, pointer, desktop, direct manipulation, WYSIWYG …

Video: 1:11 – 8:20

Where we are now

Where we are now



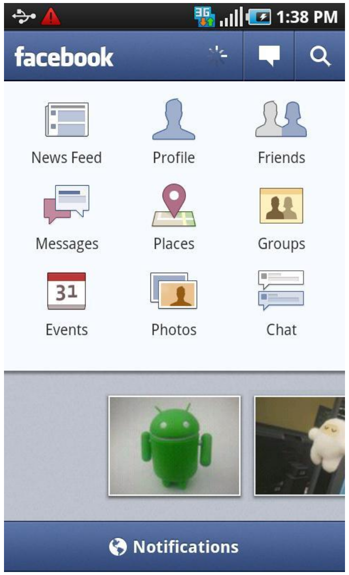
Home and media control through voice, video

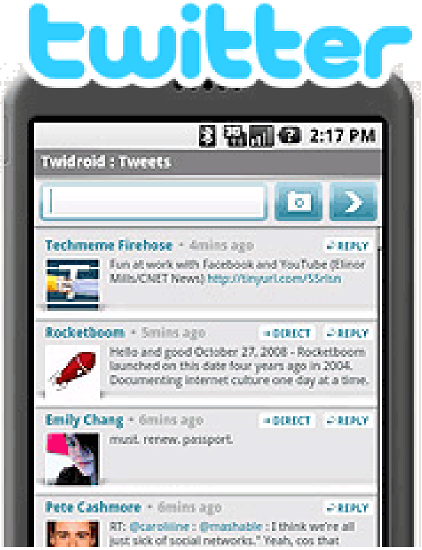
Whole-body interaction

Microsoft Kinect

Whole-body interaction via 3D computer vision for gaming.

On Mobile Devices





Techmeme Firehose • 4mins ago ↩REPLY

Fun at work with Facebook and YouTube (Elinor Mills/CNET News) http://tinyurl.com/55rlxn

Rocketboom • 5mins ago ⇒DIRECT ↩REPLY

Hello and good October 27, 2008 - Rocketboom launched on this date four years ago in 2004. Documenting internet culture one day at a time.

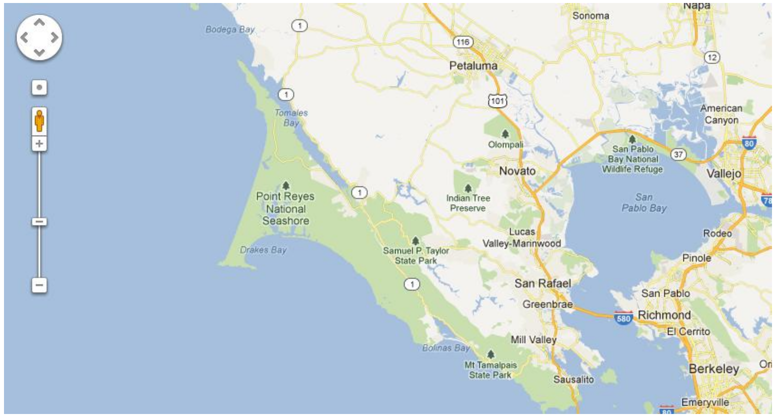
Emily Chang • 6mins ago ⇒DIRECT ↩REPLY

must. renew. passport.

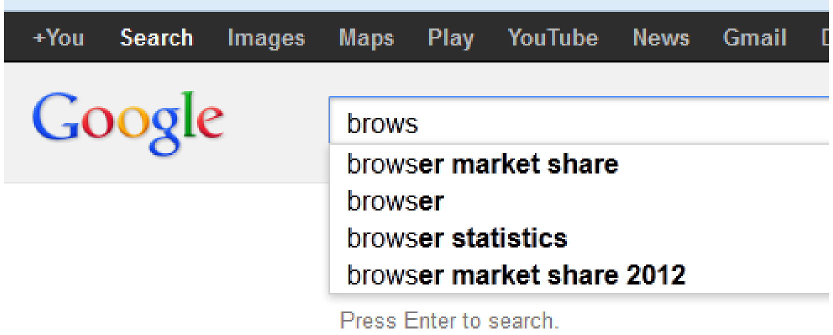
Pete Cashmore • 6mins ago ↩REPLY

RT: @caroliine : @mashable : I think we're all just sick of social networks." Yeah, cos that

And on Web Browsers



Google Maps



brows

browser market share

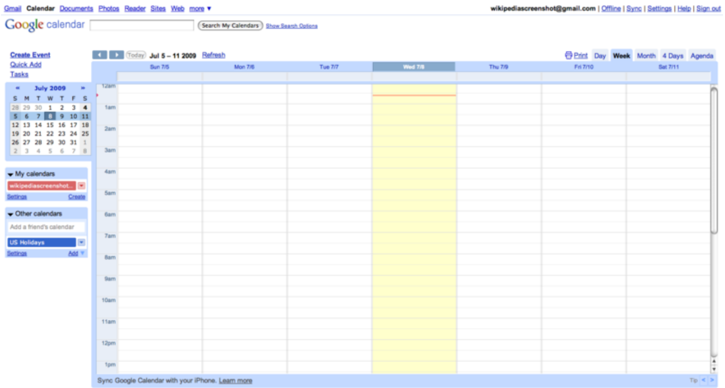
browser

browser statistics

browser market share 2012

Press Enter to search.

Google Search Bar



Google Calendar

Angry Birds for Chrome

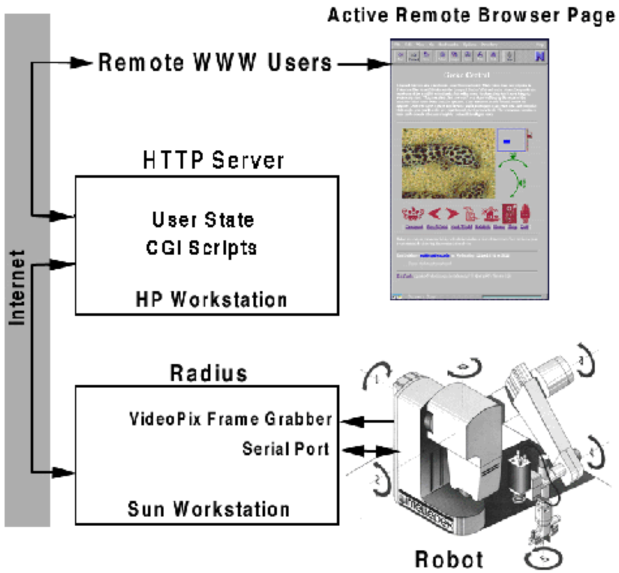
Theme for this semester: Web Applications

Interfaces (and applications) which the user interacts with via a web browser.

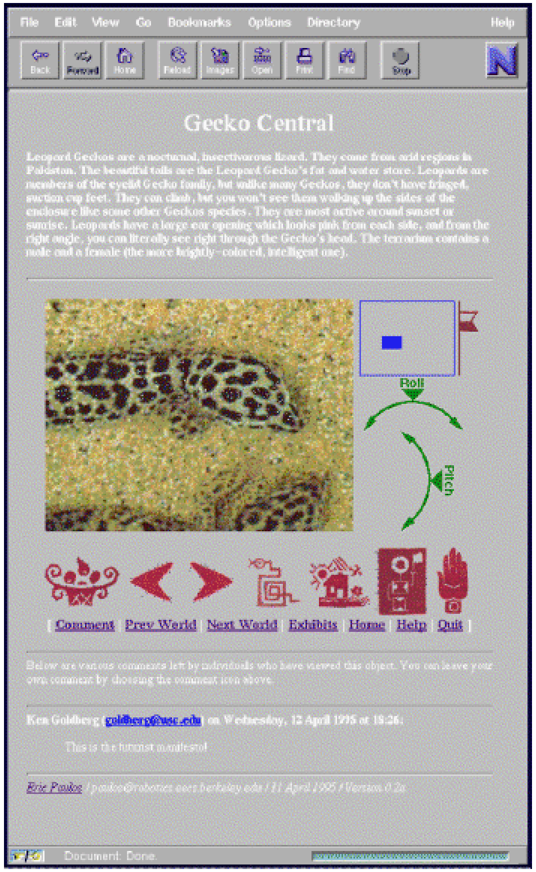
- Better interactivity, streamlined workflow, without the need to install a client program.
- The front-end of many cloud services.
- Highly portable – the experience is similar (or different!) across devices.

Brief History

- 1995: Netscape introduces client-side scripting using Javascript.
- “Hotjava” browser followed but is soon replaced by Java applet support in other browsers.



Telerobotic camera control using client-side scripting



Gecko Central

Leopard Geckos are a nocturnal, insectivorous lizard. They come from arid regions in Pakistan. The beautiful tails are the Leopard Gecko's fat and water store. Leopards are members of the eyelid Gecko family, but unlike many Geckos, they don't have fringed, suction cup feet. They can climb, but you won't see them walking up the sides of the enclosure like some other Geckos species. They are most active around sunset or sunrise. Leopards have a large ear opening which looks pink from each side, and from the right angle, you can literally see right through the Gecko's head. The terrarium contains a male and a female (the more brightly-colored, intelligent one).

[Comment | Prev World | Next World | Exhibits | Home | Help | Quit]

Below are various comments left by individuals who have viewed this object. You can leave your own comment by choosing the comment icon above.

Ken Goldberg (goldberg@usc.edu) on Wednesday, 12 April 1995 at 18:26:

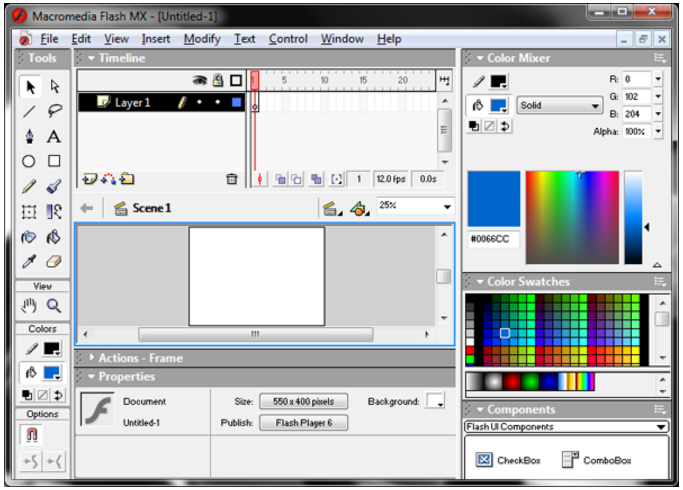
This is the futurist manifesto!

Eric Paulos / paulos@robotics.eecs.berkeley.edu / 11 April 1995 / Version 0.2a

Document: Done.

Brief History

- 1996: Macromedia Introduces Flash (later bought by Adobe)
- Much richer experiences, animation, sound, games.
- The web as an art form.



30 DAY TRIAL

macromedia® FLASH™

Version 2.0
©1997 Macromedia, Inc. All rights reserved. Macromedia, the Macromedia logo, and Flash are registered trademarks of Macromedia, Inc.
www.macromedia.com phone: 415 252 2000 fax: 415 626 0554

macromedia®

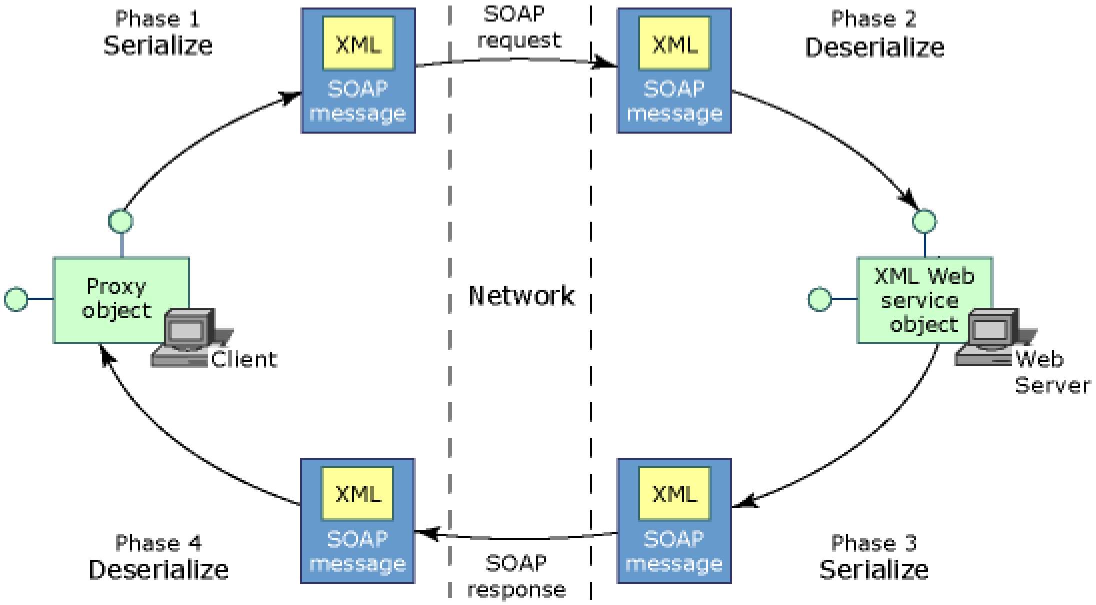
The trial version expires in 30 days.

One web crawl (2011) found Flash files on 30-40% of all websites

Web Interfaces History

1998: XML becomes the defacto standard for web app. communication.

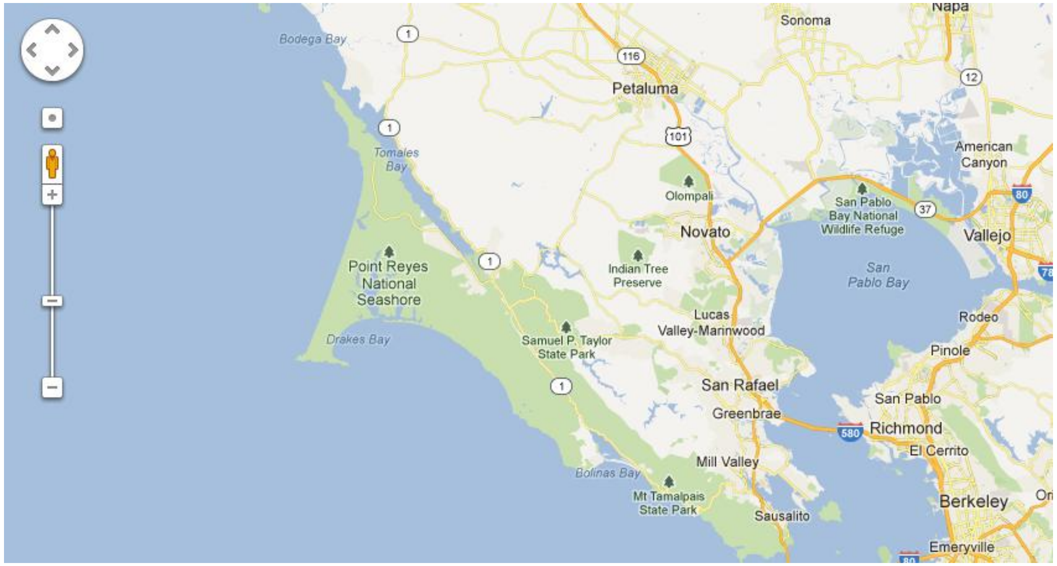
SOAP messages, DOM, improve and simply app. design.



Web Interfaces History

2005: The term “Ajax” is coined by Jesse James Garrett to describe the suite of technologies in use for web interfaces.

Google Maps is created (acquisition from Where2 Tech.)



Web Interfaces History

2001-????: SVG (Scalable Vector Graphics) standard – 2014?

1995-2000: PHP evolves to public release

1996: CSS style sheet standard published

1997-2000: HTML 4 Standard

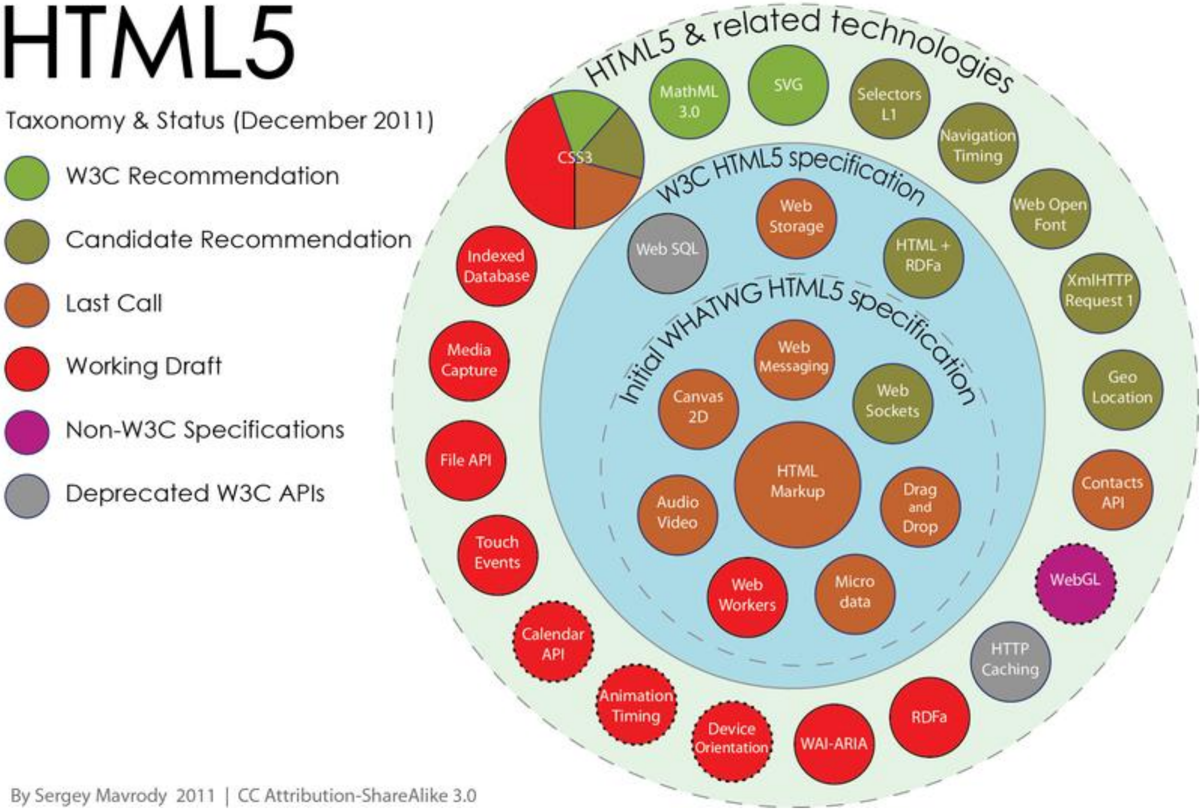
1999-2006: XHTML standard

2004-????: HTML 5 standard

2002: Microsoft ASP.NET released

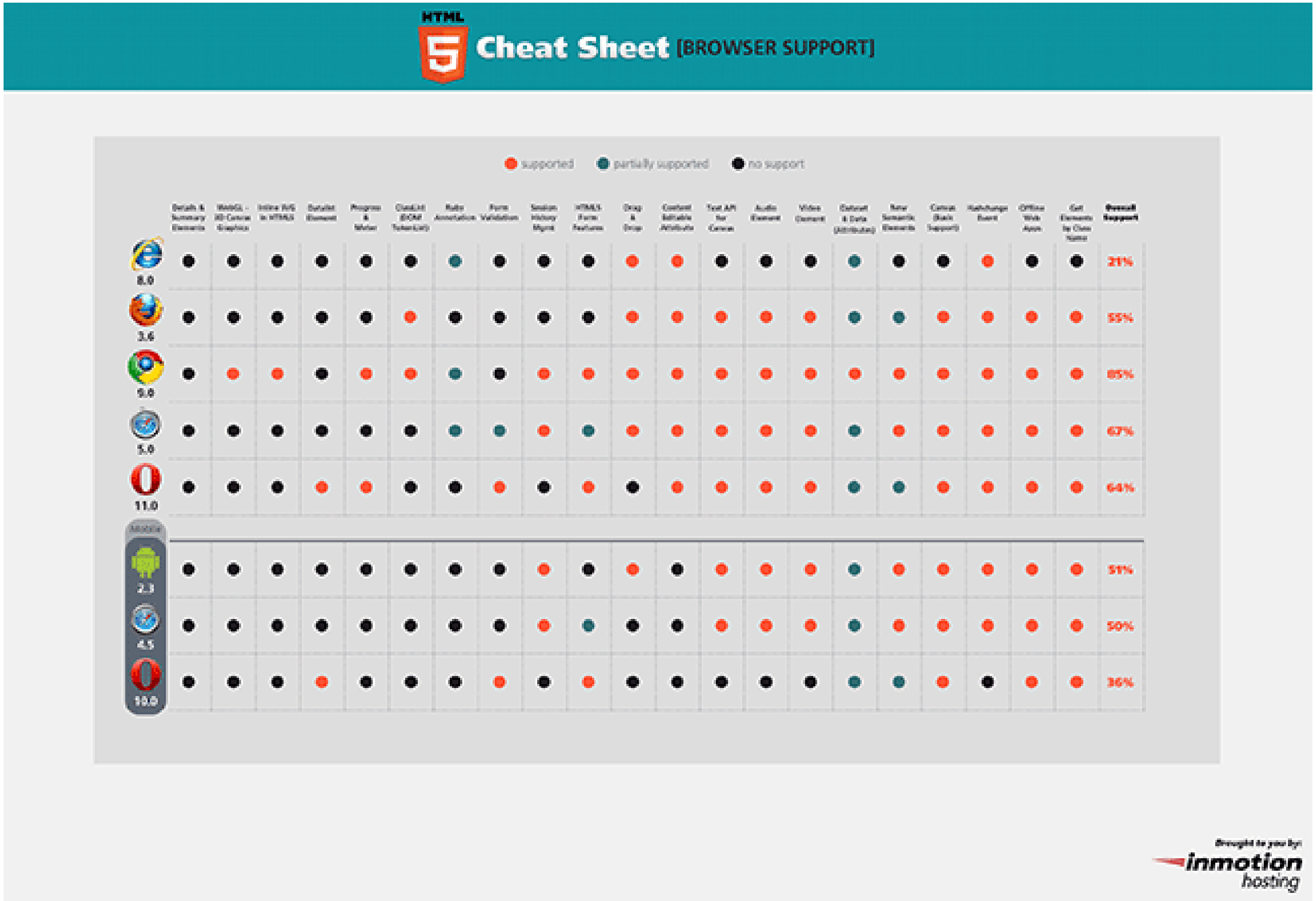
2004: Ruby on Rails released

Web Interfaces Challenges



By Sergey Mavrody 2011 | CC Attribution-ShareAlike 3.0

Web Interfaces Challenges



HTML5 Cheat Sheet [BROWSER SUPPORT]

🔴 supported   ◐ partially supported   ● no support

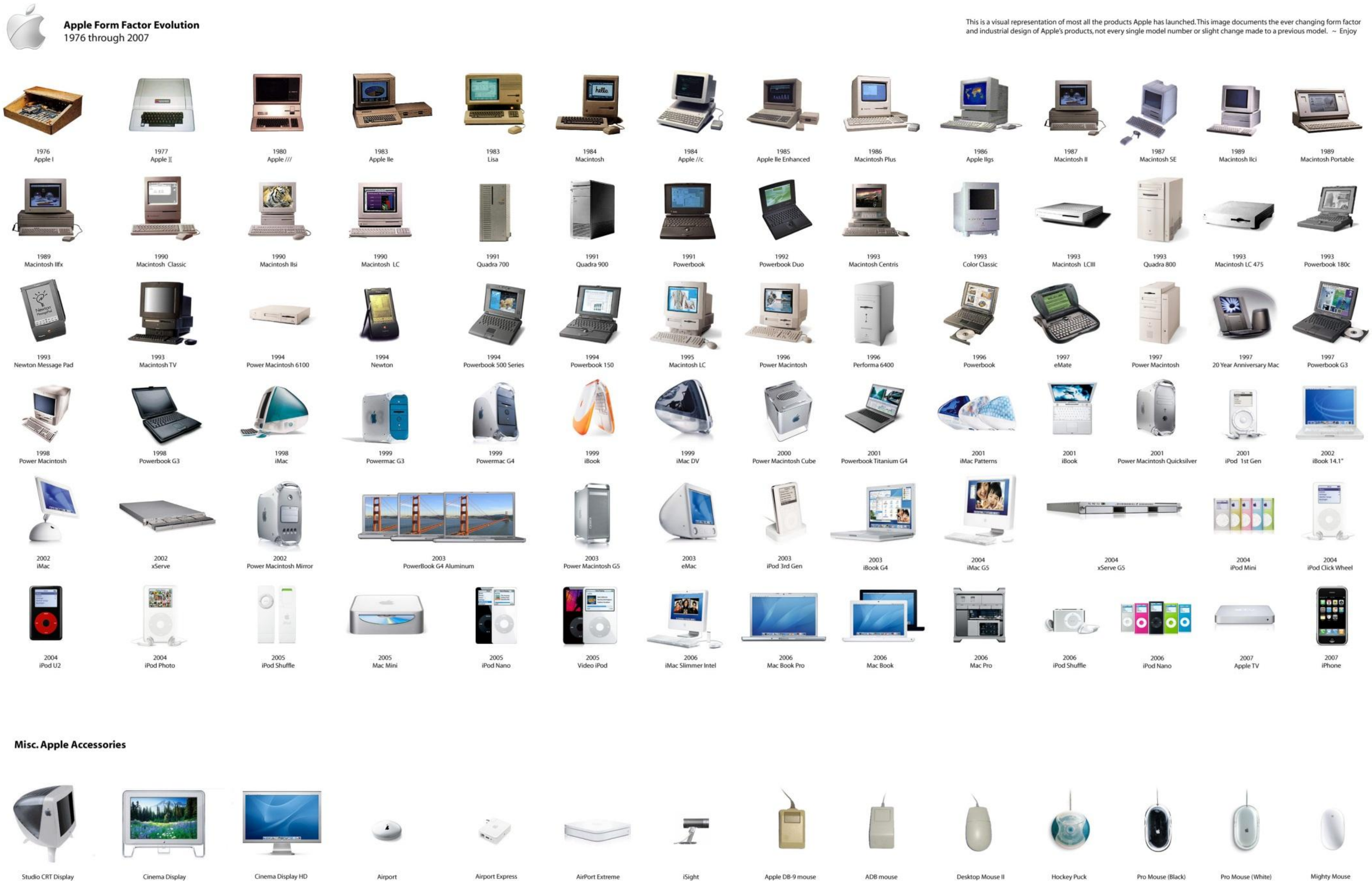
	Details & Summary Elements	WebGL - 3D Canvas Graphics	Inline SVG in HTML5	Datalist Element	Progress & Meter	ClassList (DOM TokenList)	Ruby Annotation	Form Validation	Session History Mgmt	HTML5 Form Features	Drag & Drop	Content Editable Attribute	Text API for Canvas	Audio Element	Video Element	Dataset & Data (Attributes)	New Semantic Elements	Canvas (Basic Support)	Hashchange Event	Offline Web Apps	Get Elements by Class Name	Overall Support
8.0	●	●	●	●	●	●	◐	●	●	●	🔴	🔴	●	●	●	◐	●	●	🔴	●	●	21%
3.6	●	●	●	●	●	🔴	●	●	●	●	🔴	🔴	🔴	🔴	🔴	◐	◐	🔴	🔴	🔴	🔴	55%
9.0	●	🔴	🔴	●	🔴	🔴	◐	●	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	85%
5.0	●	●	●	●	●	●	◐	◐	🔴	◐	🔴	🔴	🔴	🔴	🔴	◐	🔴	🔴	🔴	🔴	🔴	67%
11.0	●	●	●	🔴	🔴	●	●	🔴	●	🔴	●	🔴	🔴	🔴	🔴	◐	◐	🔴	🔴	🔴	🔴	64%
Mobile																						
2.3	●	●	●	●	●	●	●	●	🔴	●	🔴	●	🔴	🔴	🔴	◐	🔴	🔴	🔴	🔴	🔴	51%
4.5	●	●	●	●	●	●	●	●	🔴	◐	●	●	🔴	🔴	🔴	◐	🔴	🔴	🔴	🔴	🔴	50%
10.0	●	●	●	🔴	●	●	●	🔴	●	🔴	●	●	●	●	●	◐	◐	🔴	●	🔴	🔴	36%

Brought to you by: inmotion hosting

User Interface Design

We're studying the science of UI design, and a little of the art.

The Impact of Design



Apple Form Factor Evolution
1976 through 2007

This is a visual representation of most all the products Apple has launched. This image documents the ever changing form factor and industrial design of Apple's products, not every single model number or slight change made to a previous model. ~ Enjoy

This course

Is about the science of **Usability**.

It's also about the process of **user-centered design**.

This semester the focus is **Web Interfaces.**

The goal is not to build a working system, but an "interactive prototype."

Emphasis is on rapid prototyping and user testing to avoid obvious and not-so-obvious mistakes.

Platform: Web Interfaces

We wont constrain your project beyond the theme, but there will be programming assignments with specific tools.

These will likely include

- AJAX
- XML/SOAP/DOM
- Scripting languages
- Ruby on Rails
- Adobe/Apache Flex
- Scala/LIFT/Comet

Instructor: John Canny

Professor in EECS

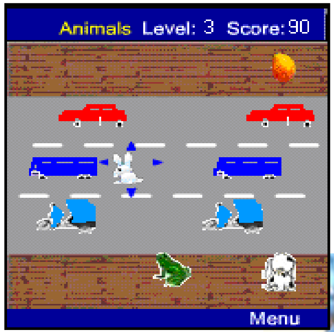
Joined Berkeley in 1987

Work in HCI, Education, Health, Data Mining

Language learning games

Persuasive technologies

Mobile applications



GSI

Name: Pablo Paredes

Areas: Behavior Change Technology

- Focus on inner circle and long term engagement

Engaging CBT (Cognitive Therapy)

Machinima, Narrative and Gaming

Tangible devices for emotion

- Happy Cube
- Tangible devices for calming technologies

Advisor: John Canny

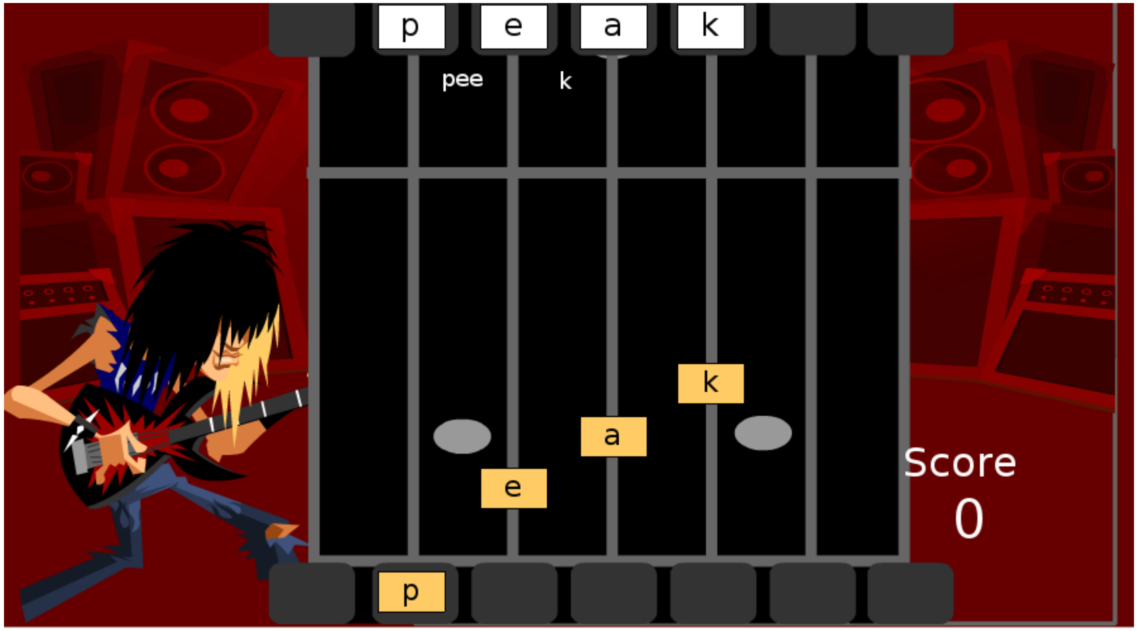
GSI

Name: Anuj Tewari

Areas of interest: Educational technology, Speech recognition, Game design

Research topic: Speech and Pronunciation Improvement via Games, for Hispanic Children

Advisor: John Canny



Topics

- Course Mechanics
- Course Overview
- Project Description

Course Mechanics

TAs, Office Hours, Sections

Teaching Assistants

- Pablo Paredes, Anuj Tewari: EECS grad students

Office Hours

- John Canny: MW 2:30-3:30, in 637 Soda Hall
- Also by appointment

Sections

- Friday 10-11AM, 11-12N, 405 Soda Hall (this week)
- Friday 1-2PM, 320 Soda
- Will cover new material. You should attend!

Reaching Us

Email: cs160@imail.eecs.berkeley.edu

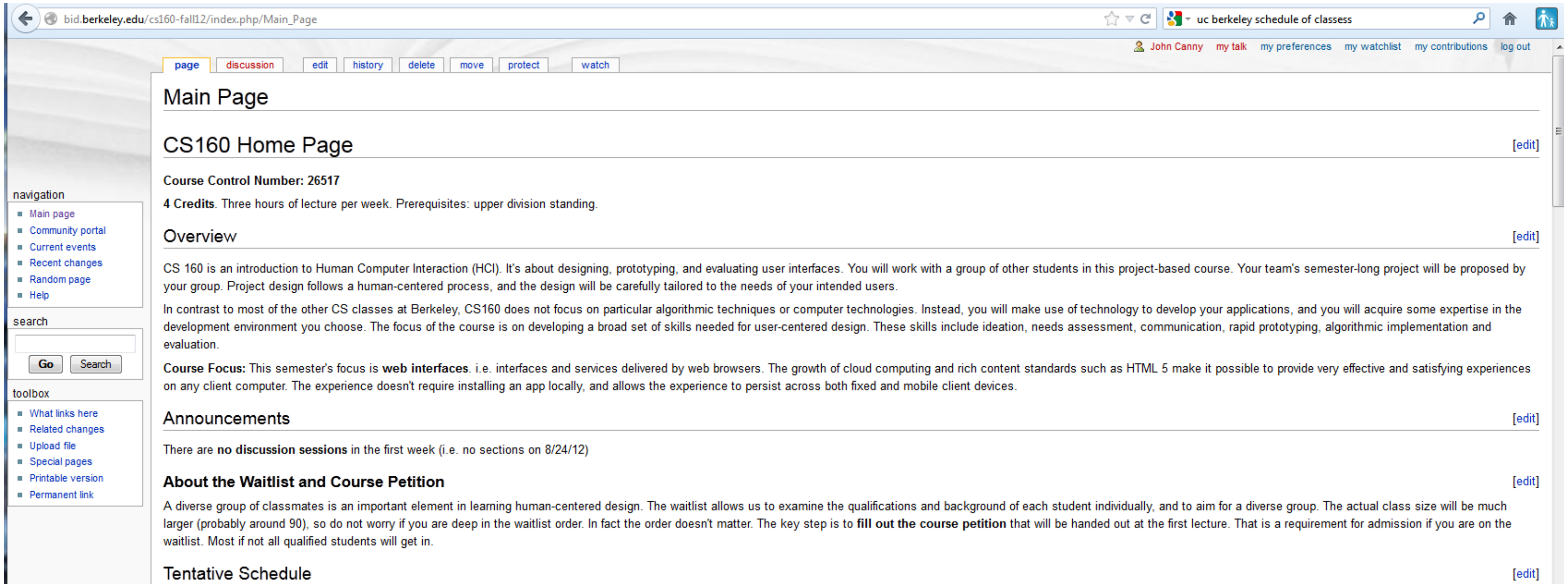
- Mail sent here will get the fastest response
- Please avoid mailing us directly

Admission: Course Petition

- Due by this Weds, bring to next class
- All **waitlisted** students should submit
- Information **will determine admission**

We're looking for diversity – i.e. qualified non-majors as well as majors.

Class Wiki



Main Page

CS160 Home Page [edit]

Course Control Number: 26517

4 Credits. Three hours of lecture per week. Prerequisites: upper division standing.

Overview [edit]

CS 160 is an introduction to Human Computer Interaction (HCI). It's about designing, prototyping, and evaluating user interfaces. You will work with a group of other students in this project-based course. Your team's semester-long project will be proposed by your group. Project design follows a human-centered process, and the design will be carefully tailored to the needs of your intended users.

In contrast to most of the other CS classes at Berkeley, CS160 does not focus on particular algorithmic techniques or computer technologies. Instead, you will make use of technology to develop your applications, and you will acquire some expertise in the development environment you choose. The focus of the course is on developing a broad set of skills needed for user-centered design. These skills include ideation, needs assessment, communication, rapid prototyping, algorithmic implementation and evaluation.

Course Focus: This semester's focus is **web interfaces**. i.e. interfaces and services delivered by web browsers. The growth of cloud computing and rich content standards such as HTML 5 make it possible to provide very effective and satisfying experiences on any client computer. The experience doesn't require installing an app locally, and allows the experience to persist across both fixed and mobile client devices.

Announcements [edit]

There are **no discussion sessions** in the first week (i.e. no sections on 8/24/12)

About the Waitlist and Course Petition [edit]

A diverse group of classmates is an important element in learning human-centered design. The waitlist allows us to examine the qualifications and background of each student individually, and to aim for a diverse group. The actual class size will be much larger (probably around 90), so do not worry if you are deep in the waitlist order. In fact the order doesn't matter. The key step is to **fill out the course petition** that will be handed out at the first lecture. That is a requirement for admission if you are on the waitlist. Most if not all qualified students will get in.

Tentative Schedule [edit]

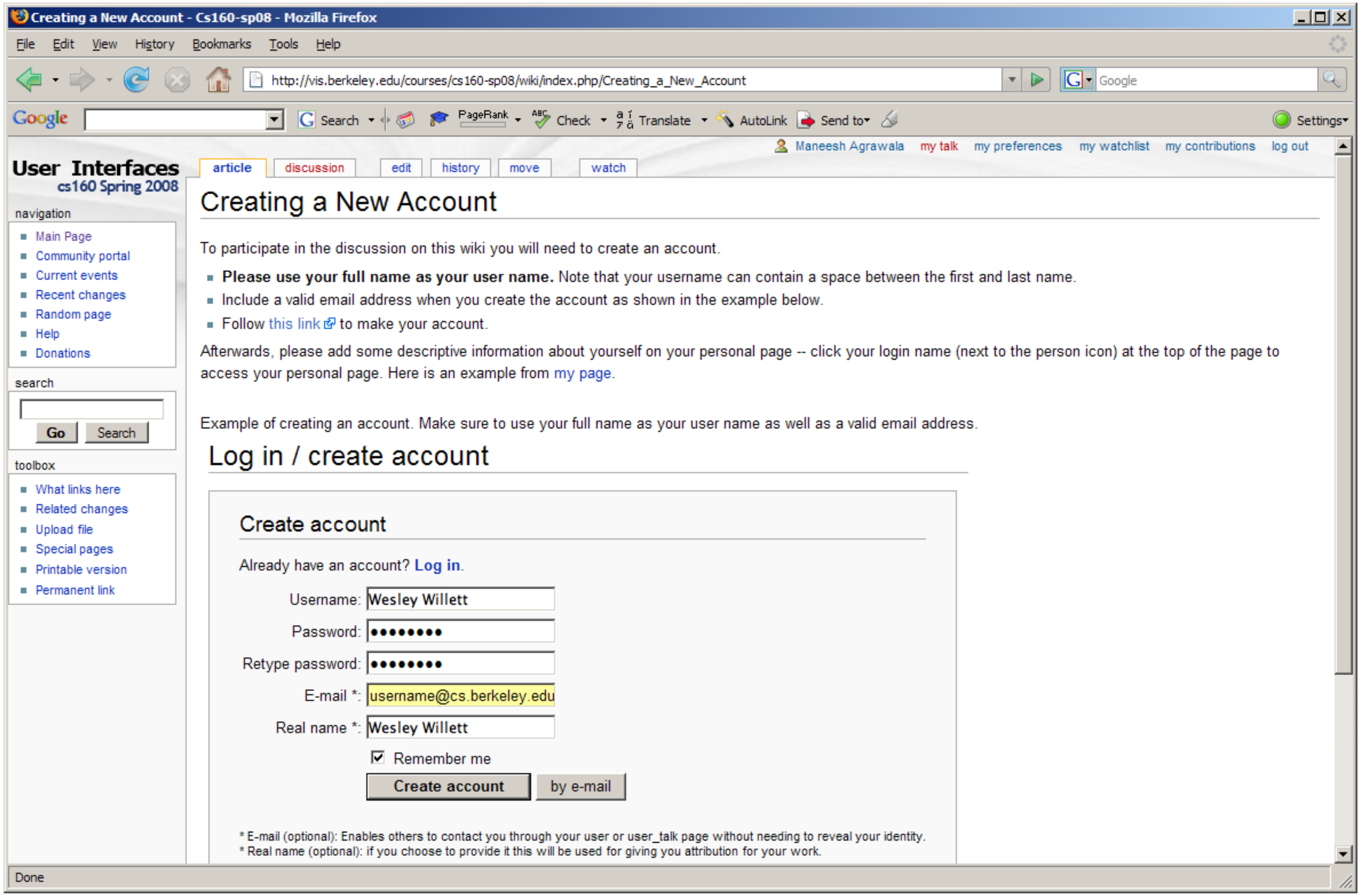
http://bid.berkeley.edu/cs160-fall12/

Link from John Canny's page

Link for CS160 from EECS instructional page

Create Wiki Account

Your 1st assignment (due by this Weds)


Creating a New Account - Cs160-sp08 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://vis.berkeley.edu/courses/cs160-sp08/wiki/index.php/Creating_a_New_Account
Google
Search PageRank Check Translate AutoLink Send to Settings
Maneesh Agrawala my talk my preferences my watchlist my contributions log out
User Interfaces
cs160 Spring 2008
article discussion edit history move watch
navigation
Main Page
Community portal
Current events
Recent changes
Random page
Help
Donations
search
Go Search
toolbox
What links here
Related changes
Upload file
Special pages
Printable version
Permanent link
Done

Creating a New Account

To participate in the discussion on this wiki you will need to create an account.

- **Please use your full name as your user name.** Note that your username can contain a space between the first and last name.
- Include a valid email address when you create the account as shown in the example below.
- Follow this link to make your account.

Afterwards, please add some descriptive information about yourself on your personal page -- click your login name (next to the person icon) at the top of the page to access your personal page. Here is an example from my page.

Example of creating an account. Make sure to use your full name as your user name as well as a valid email address.

Log in / create account

Create account

Already have an account? **Log in**.

Username:	Wesley Willett
Password:	••••••••
Retype password:	••••••••
E-mail *:	username@cs.berkeley.edu
Real name *:	Wesley Willett

☑ Remember me

Create account | by e-mail

* E-mail (optional): Enables others to contact you through your user or user_talk page without needing to reveal your identity.
* Real name (optional): if you choose to provide it this will be used for giving you attribution for your work.

Readings

Readings are very important to the class

- Make sure you do the reading *before class*
- Midterm/Final will include things only in readings

Most readings will be posted on wiki

- Require username/password:

Online reading discussions (ongoing assignment)

- Must post *one substantial comment* per lecture
- We will **not** accept late comment
- Will be the major factor in you class participation grade

Content

This is a design-oriented course which focuses on several skills:

- User-centered design, needs analysis etc.
- Problem formulation
- Brainstorming
- Rapid prototyping and iterative design
- Usability evaluation
- Interactive system design and programming

Grading

Class participation (5%)
Individual (non-programming) assignments (15%)
Pair Programming Assignments (15%)
Group project (50%)
Midterm (20%)

Score distribution is high with small variance. So every point counts! Make sure you turn in all the reading comments.

Policies

Late Assignments

- Most assignments will be due before class on the due date
- Group assignments will not be accepted late
- Individual assignments lose 20% per day

Cheating (official)

- Will get you an **F** in the course
- More than once can get you dismissed from Cal

Assessment

Goal of cs160 is to teach you to *design* and *evaluate* interfaces

- There is often **more than one good design**
- But, there are also **lots and lots of poor designs**
- Be critical of your own work (point out pros and cons)
- As in many design disciplines, grading will be qualitative

Specific assessment guidelines will be given in each assignment

Good **communication** expected in oral & written presentations

Groups **self-assess** participation

- Should monitor it throughout the project
- Meet with us as soon as problems emerge

Course Overview

Human-Computer Interaction (HCI)

Human

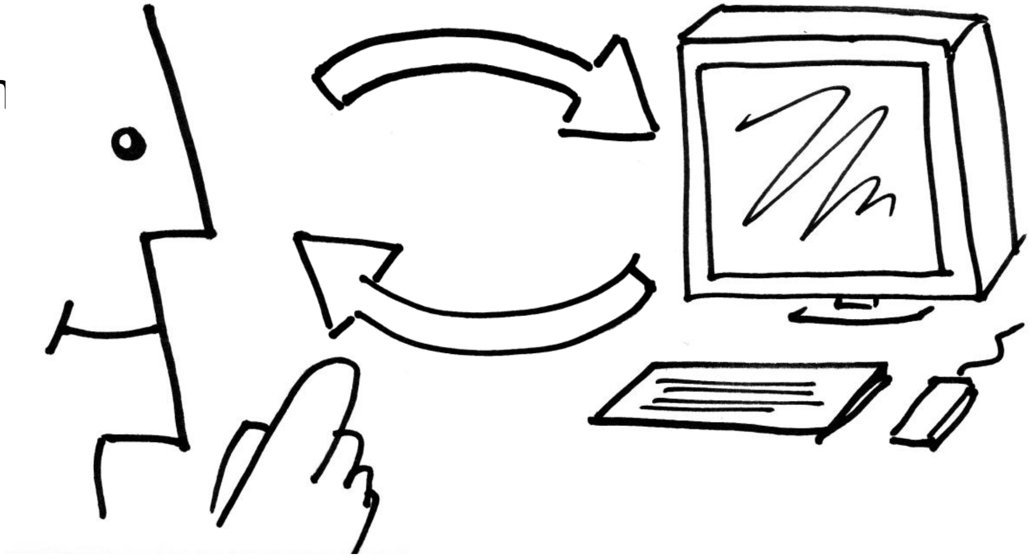
- User of program
- Others (friends, collaborators, coworkers)

Computer

- Machine program runs on
- Often split: clients & servers

Interaction

- User tells the computer what they want
- Computer communicates results



User Interfaces (UIs)

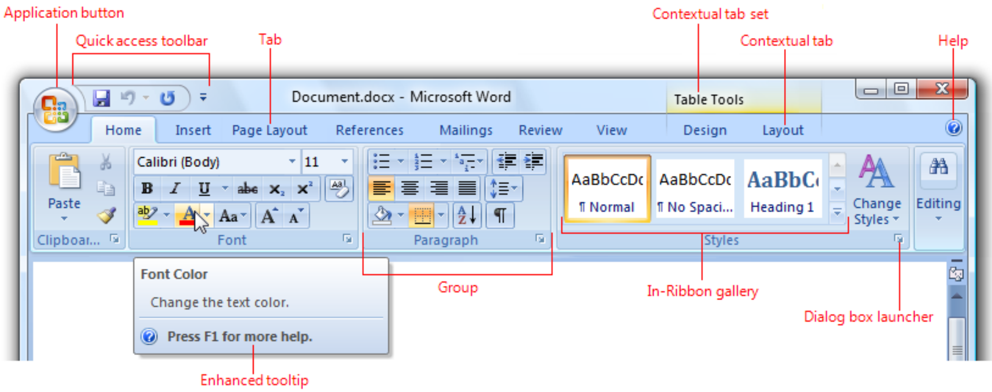
Part of application that allows

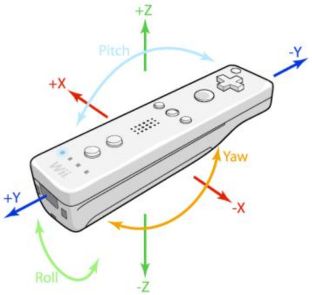
- People to interact with computer
- Computer to communicate results

Can include hardware design

- Buttons, sliders, other sensors

HCI =
design, prototyping, implementation & evaluation of UIs





http://www.reactable.com

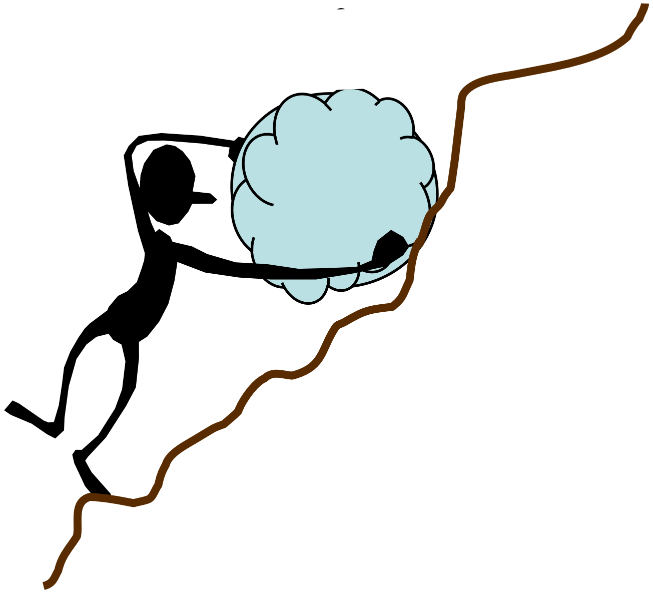
Why Study User Interfaces?

Major part of work for most commercial programs
- Approximately 50%

You will work on software for a market
- Intended for people other than yourself

Bad user interfaces cost
- Money (5%↑ satisfaction → up to 85%↑ profits)
- Lives

User interfaces hard to get right
- People are unpredictable



Life-Threatening Errors

- 1995 American Airlines jet crashed into canyon wall, killing all aboard
  - On approach to **Rozo** airport in Colombia
  - Pilot skipped some of the approach procedures
  - Pilot typed in "**R**" and system completed full name of airport to **Romeo**
  - Guidance system executed turn at low altitude to head for Romeo airport
  - 9 seconds later plane struck canyon wall

- Is the pilot to blame?

http://en.wikipedia.org/wiki/American_Airlines_Flight_965

What is Usability?

Intuitive

- The design should seem natural

Ease of learning

- Faster the second time and so on...

Productivity

- Perform tasks quickly and efficiently

Minimal error rates

- If they occur, good feedback so user can recover

High user satisfaction

- Confident of success

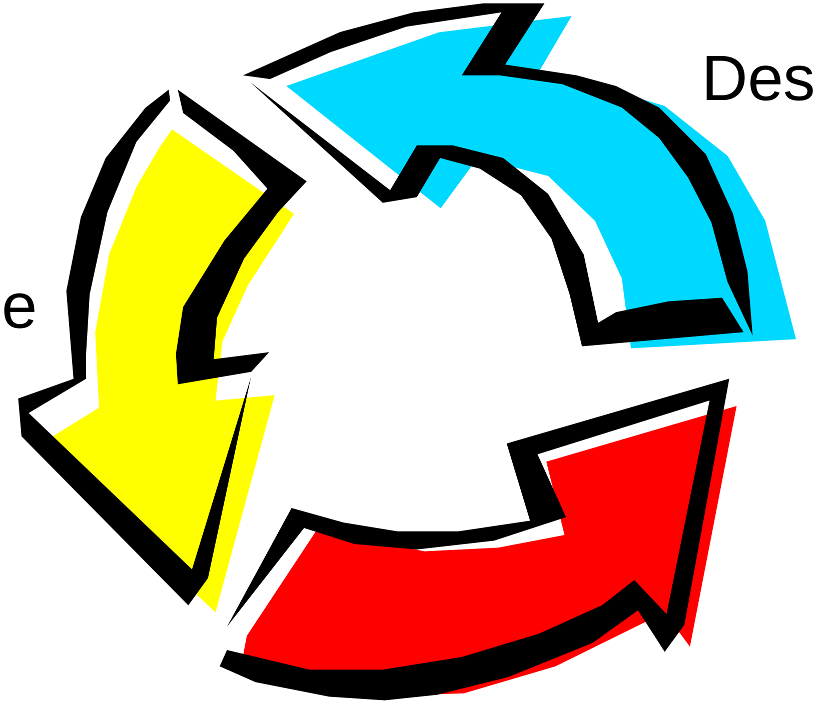
Who Builds Interfaces?

Ideally a team of specialists

- graphic designers
- interaction / interface designers
- technical writers
- marketers
- test engineers
- software engineers
- customers

Some engineers become very good at user-centered design, but its not for all engineers.

Interface Design Cycle



Design

Prototype

Evaluate

Building Successful Interfaces

1. Task analysis & contextual inquiry
2. Rapid prototyping
3. Evaluation
4. Iteration: Back to 1

Task Analysis & Contextual Inquiry

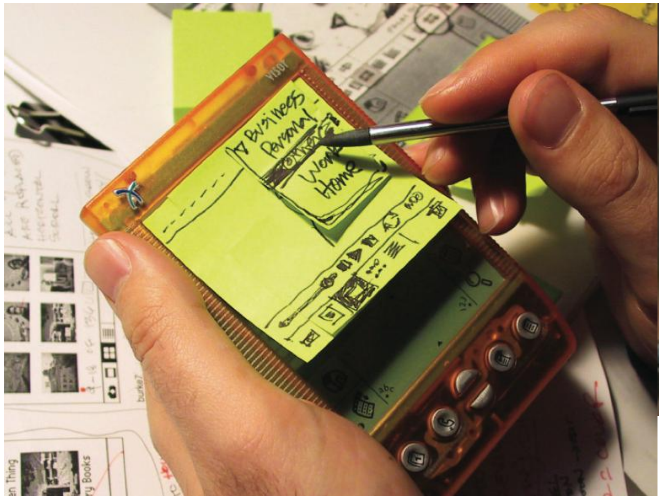
- Observe existing practices
- Create scenarios of actual use
- Create models to gain insight into work processes

CS247, Stanford, 2006

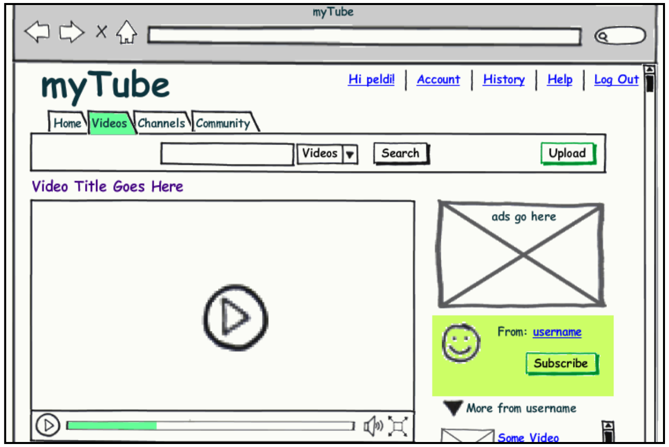
http://www-personal.umich.edu/~chrisli/m2.html

Rapid Prototyping

- Build a mock-up of design (or more!)
- Low fidelity techniques
  - Paper sketches
  - Cut, copy, paste
  - Video segments
- Interactive prototyping tools
  - HTML, Flash, Javascript, Visual Basic, C#, etc.
- UI builders
  - Interface Builder, Visual Studio, NetBeans



Moggridge, Designing Interactions, p.704



http://www.balsamiq.com/products/mockups/examples#wiki

Evaluation

Evaluate analytically (no users)

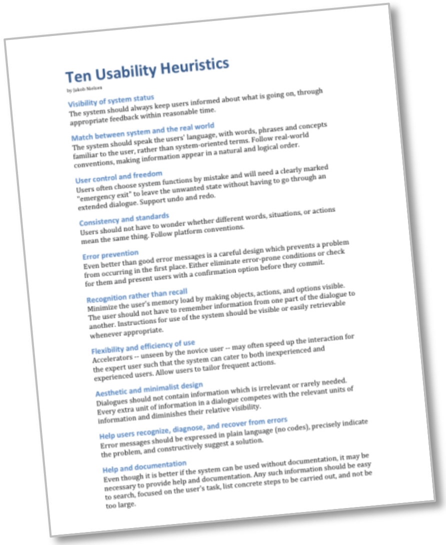
Test with real target users

Low-cost techniques

- expert evaluation
- walkthroughs

Higher cost

- Controlled usability study



Ten Usability Heuristics

by Jakob Nielsen

Visibility of system status
The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

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

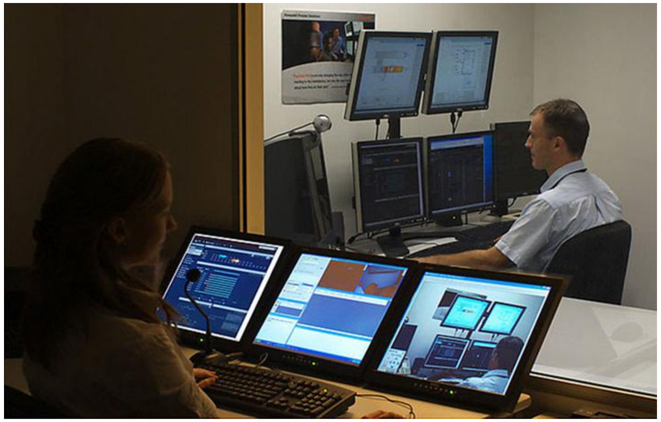
Recognition rather than recall
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use
Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design
Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors
Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.



http://www.laurasmith.info/UsabilityTest.jpg

Building Successful Interfaces

- Task analysis & contextual inquiry
- Rapid prototyping
- Evaluation
- Iteration

Evaluation brings **real users** into the design loop.

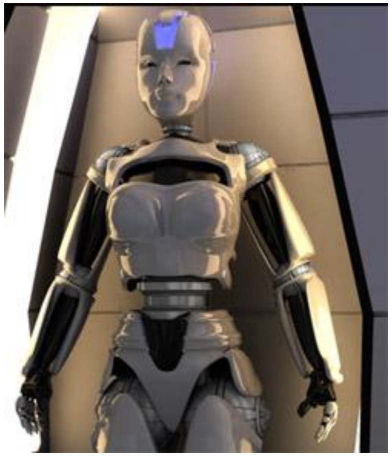
Design stays **user-centered** throughout the process.

Why not simulate the user?

People have certainly tried. Its useful in certain special cases, e.g. pointing and typing evaluation.

For most applications, people are far too complex to simulate. Behavior depends on just about every external factor.

Users are their own best simulation.
In fact they are ground truth…



Goals of the Course

Learn to design, prototype, evaluate interfaces

- Discover needs and preferences of real customers
- Cognitive/perceptual constraints that effect design
- Building and rapidly evolving interactive systems
- Techniques to test and evaluate a product
- How to work together on a team project
- Communicate your results effectively

Goals of the Course

Learn to design, prototype, evaluate interfaces

- Discover needs and preferences of real customers
- Cognitive/perceptual constraints that effect design
- Building and rapidly evolving interactive systems
- Techniques to test and evaluate a product
- How to work together on a team project
- Communicate your results effectively

These non-technical skills become more important the further you go in your career.

Pair Programming Assignments

Pair Programming Teams

Pair programming is an agile Software Engineering technique where two people work together on the same program.

Goal is to add "another set of eyes"

Spot bugs early

Support peer learning

In this course it allows us to include compulsory programming assignments without assuming advanced programming skills

You will need to find a programming partner by next Monday. Email us if you need help.

Project Description

Teams

Each of you will individually propose an application idea

- We follow a studio model to help you think outside the box
- Learning and working with others is central to the course

Groups

- 4 or 5 students to a team
- Work with students with different skills/interests

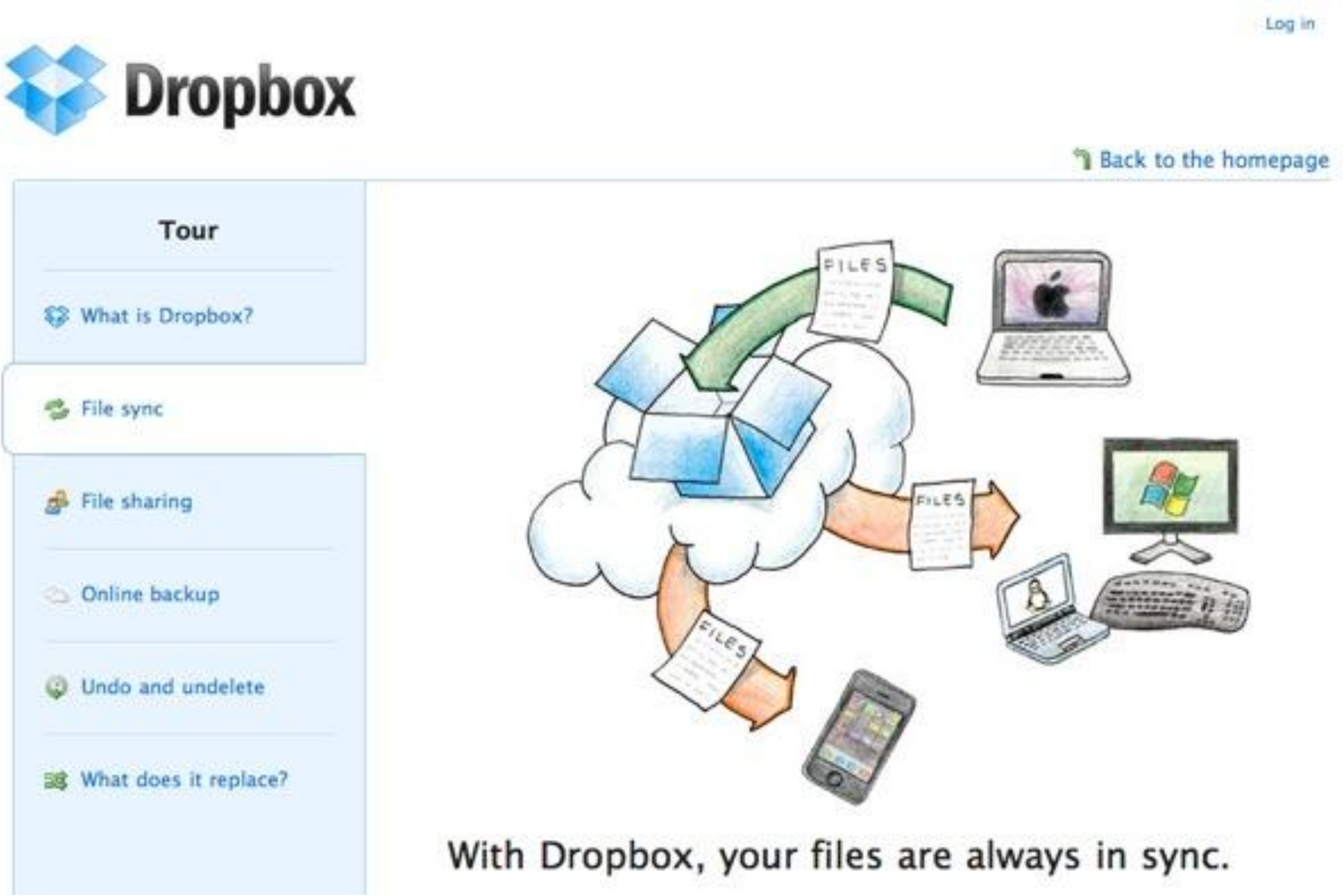
Cumulative

- Apply several HCI methods to a single interface

Theme: Web Interfaces

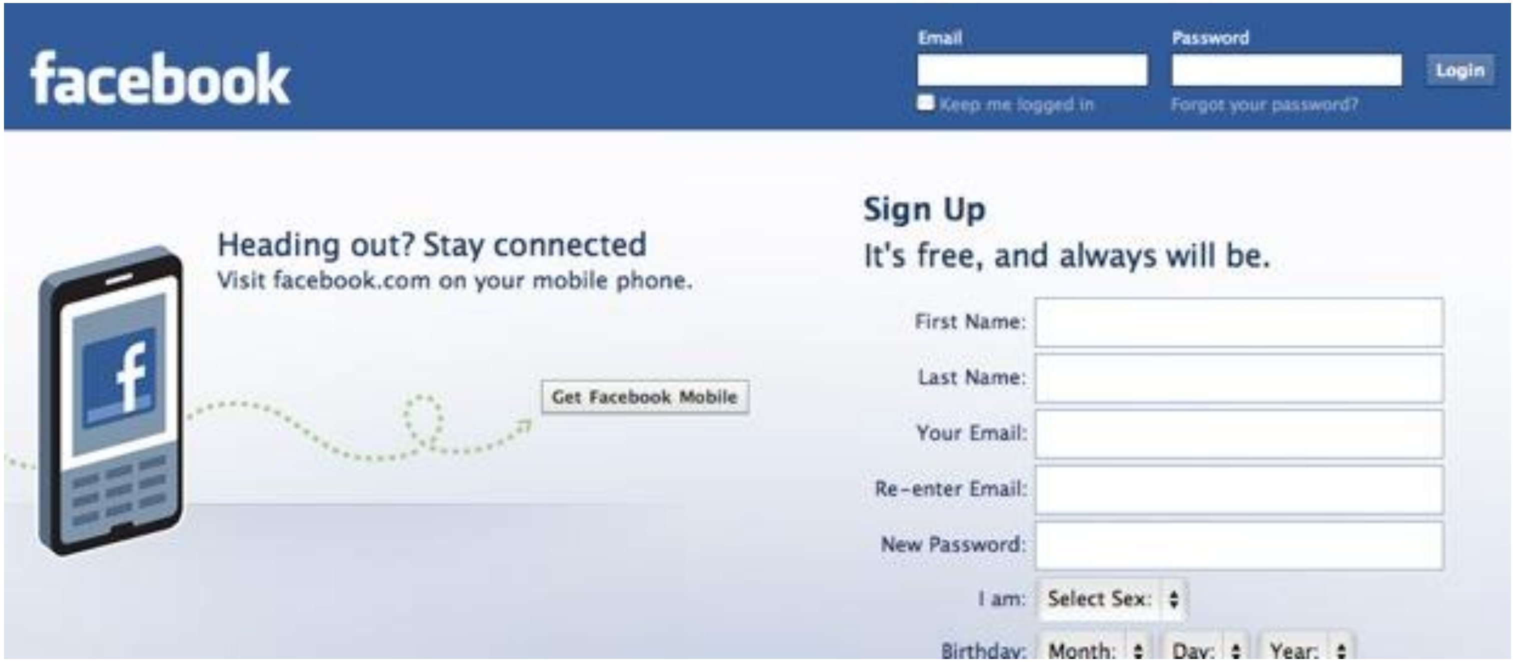
- Plenty of design freedom
- Be careful not to over-design
  - Simplicity is the best design principle
  - “one-size-fits-all” is rarely true across devices
- You’re designing an interactive ***prototype*** – you don’t need to implement much (or any) back-end functionality.

Inspiration: Dropbox



With Dropbox, your files are always in sync.

Inspiration: Social Media



Heading out? Stay connected

Visit facebook.com on your mobile phone.

Sign Up

It's free, and always will be.

Inspiration: Input

Google Voice Search - http://www.youtube.com/watch?v=y3z7Tw1K17A

Inspiration: Voice

- IBM Websphere voice
- IBM “Watson” voice service

Next Time

The Design Cycle and Brainstorming

- The Task-Centered Design Process. *Task-Centered User Interface Design.* Chap 1. Lewis & Rieman
- The Perfect Brainstorm. *The Art of Innovation.* Kelley

  Will need username/password for this one