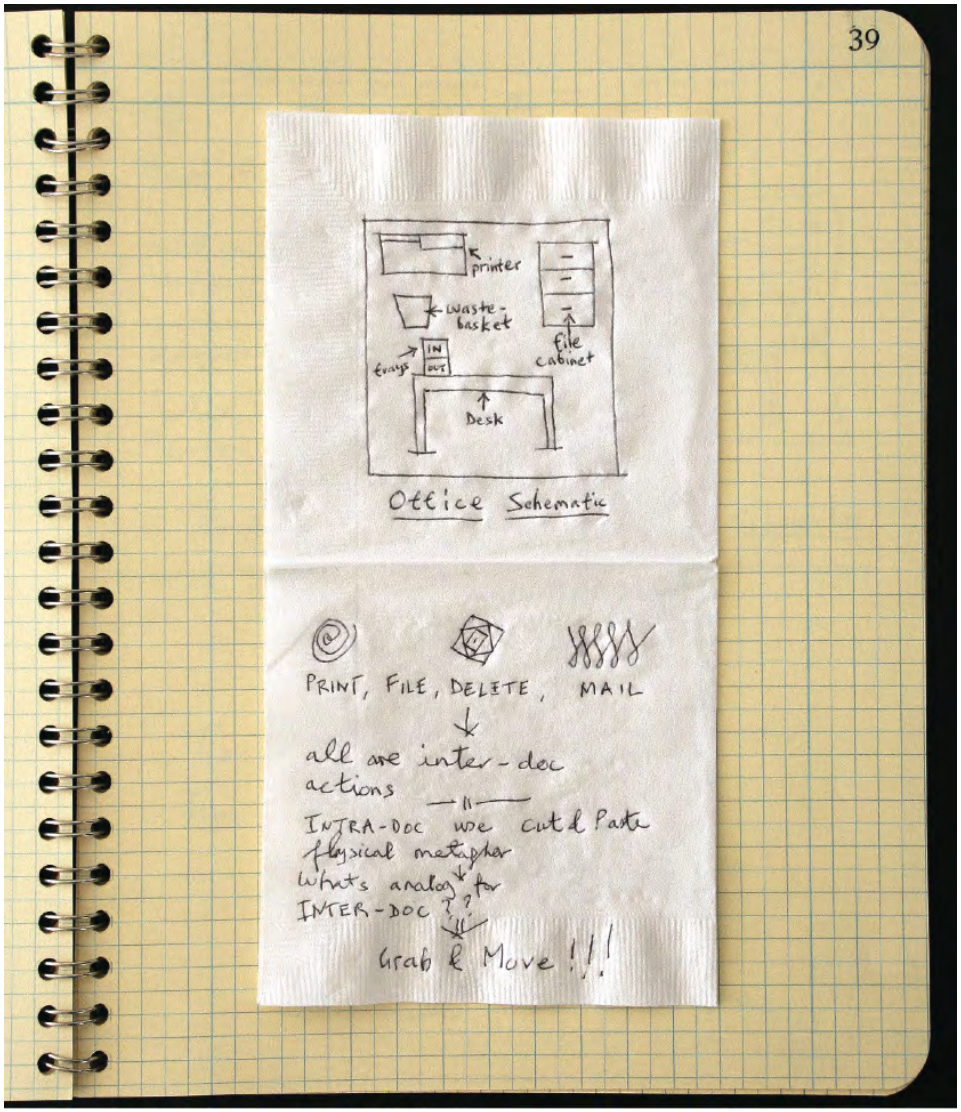
CS160: User Interface Design

Maneesh Agrawala & Björn Hartmann, *Spring 2010*

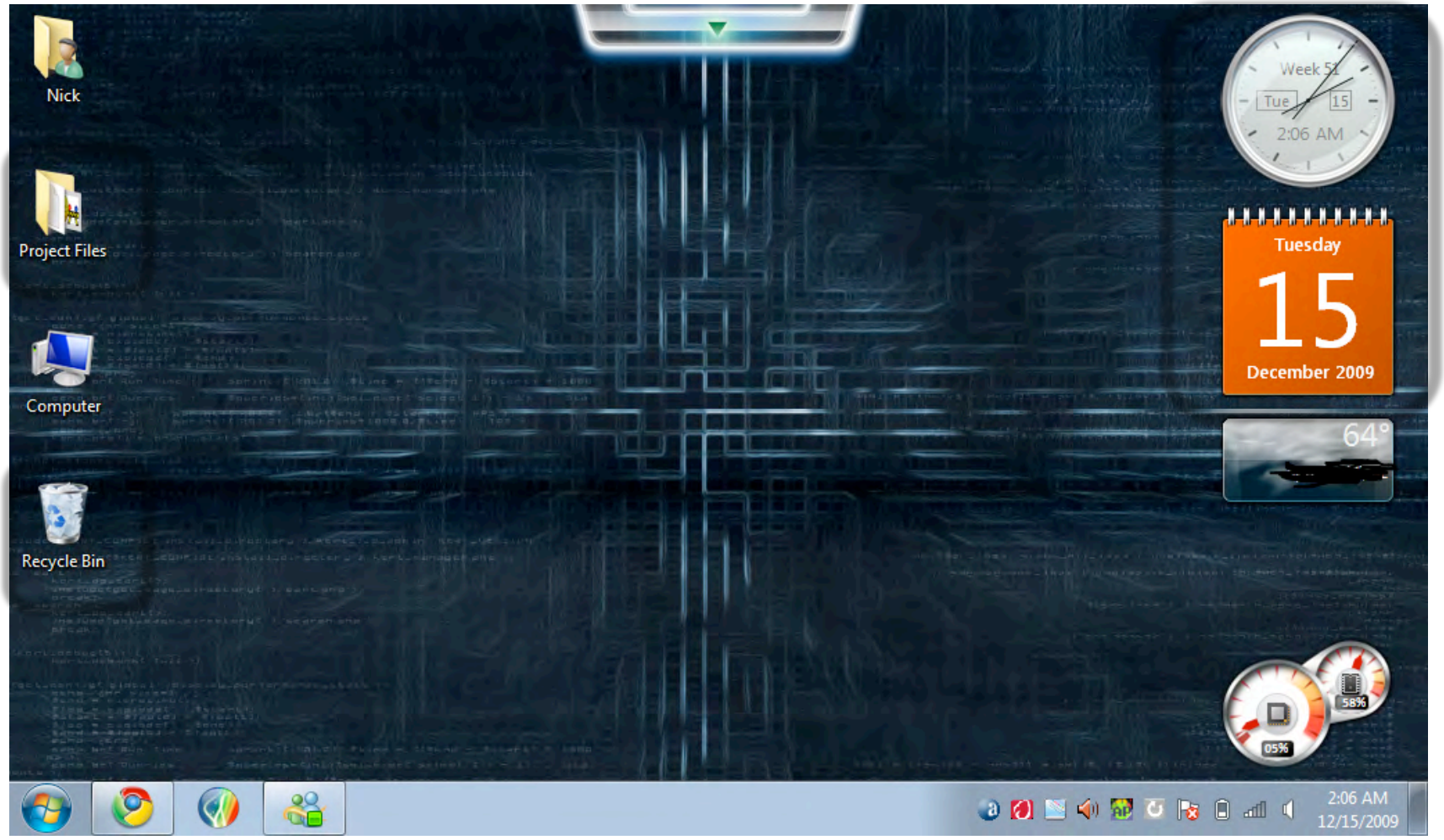
Berkeley
UNIVERSITY OF CALIFORNIA

http://stackoverflow.com/questions/238177

The Desktop Metaphor…



Tim Mott, mid-1970s, from Moggridge, Designing Interactions, p. 52

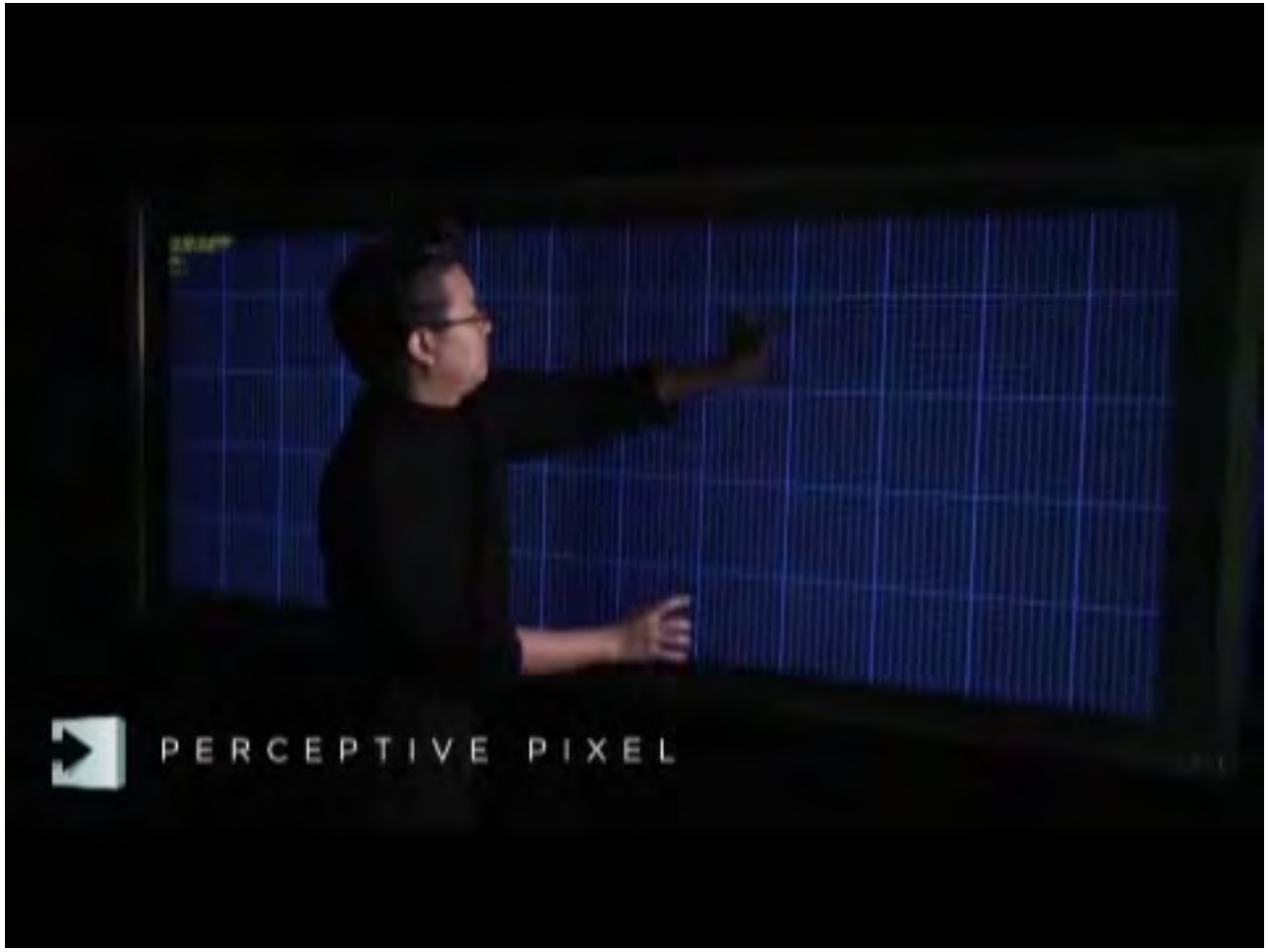


Windows 7 - http://i47.tinypic.com/2zp1kzt.jpg

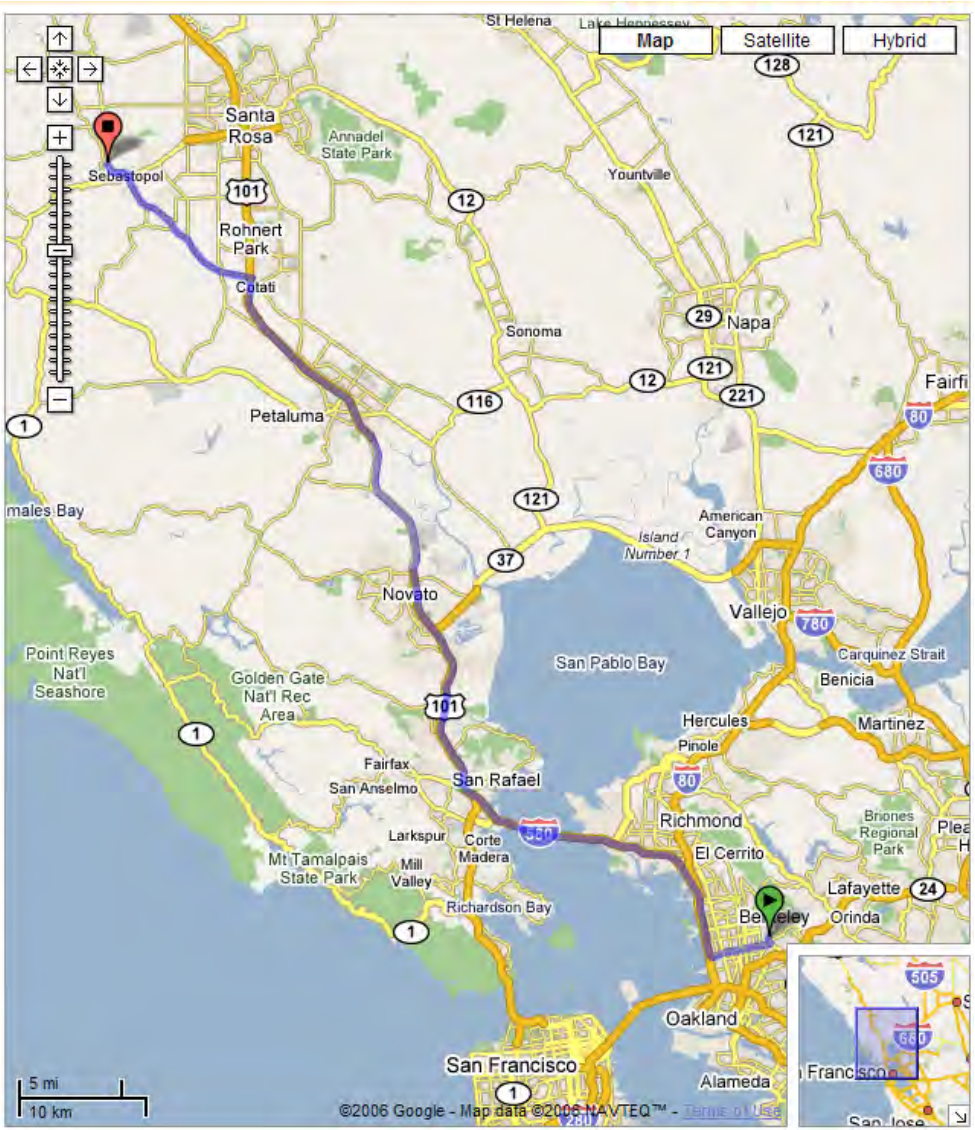
Is this a good idea? When?

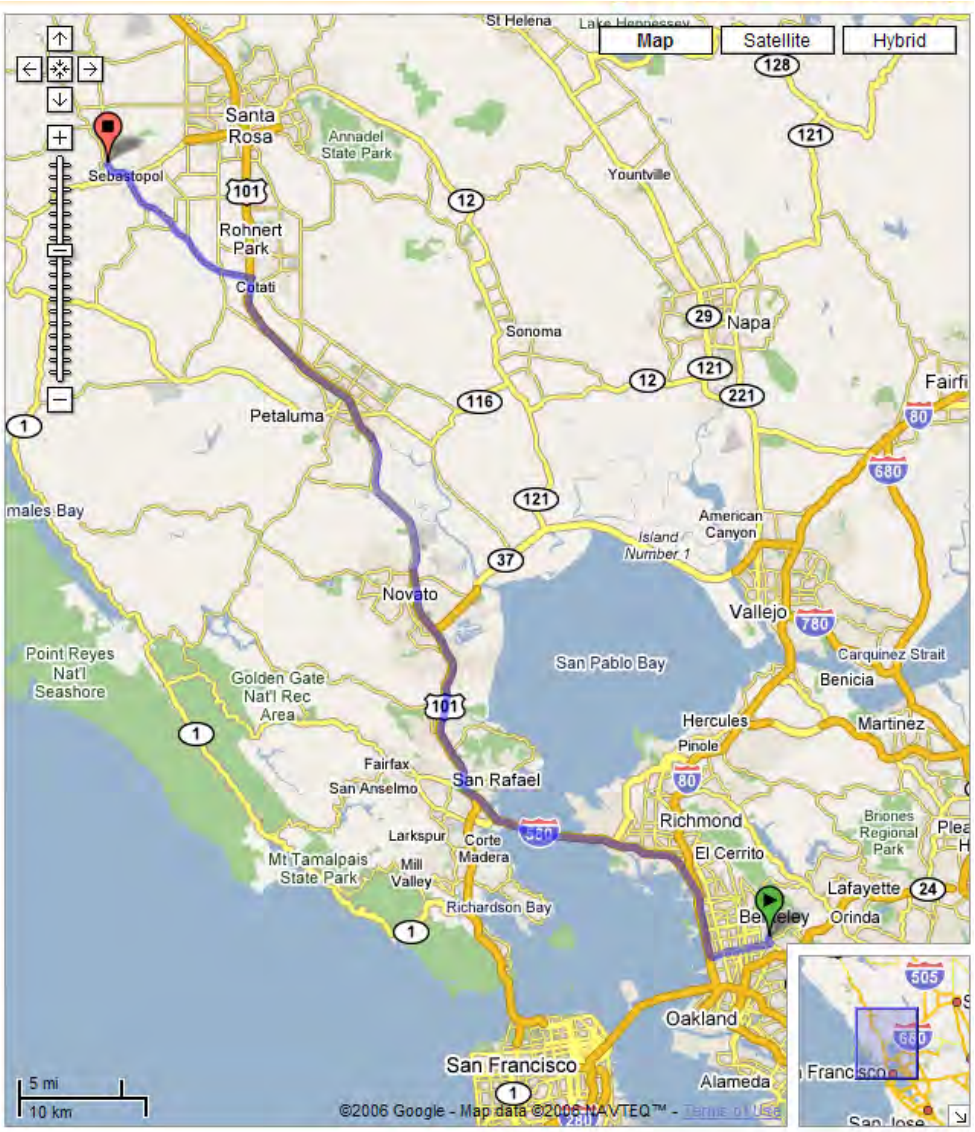
http://www.bumptop.com

How about this? (Implement as Homework!)

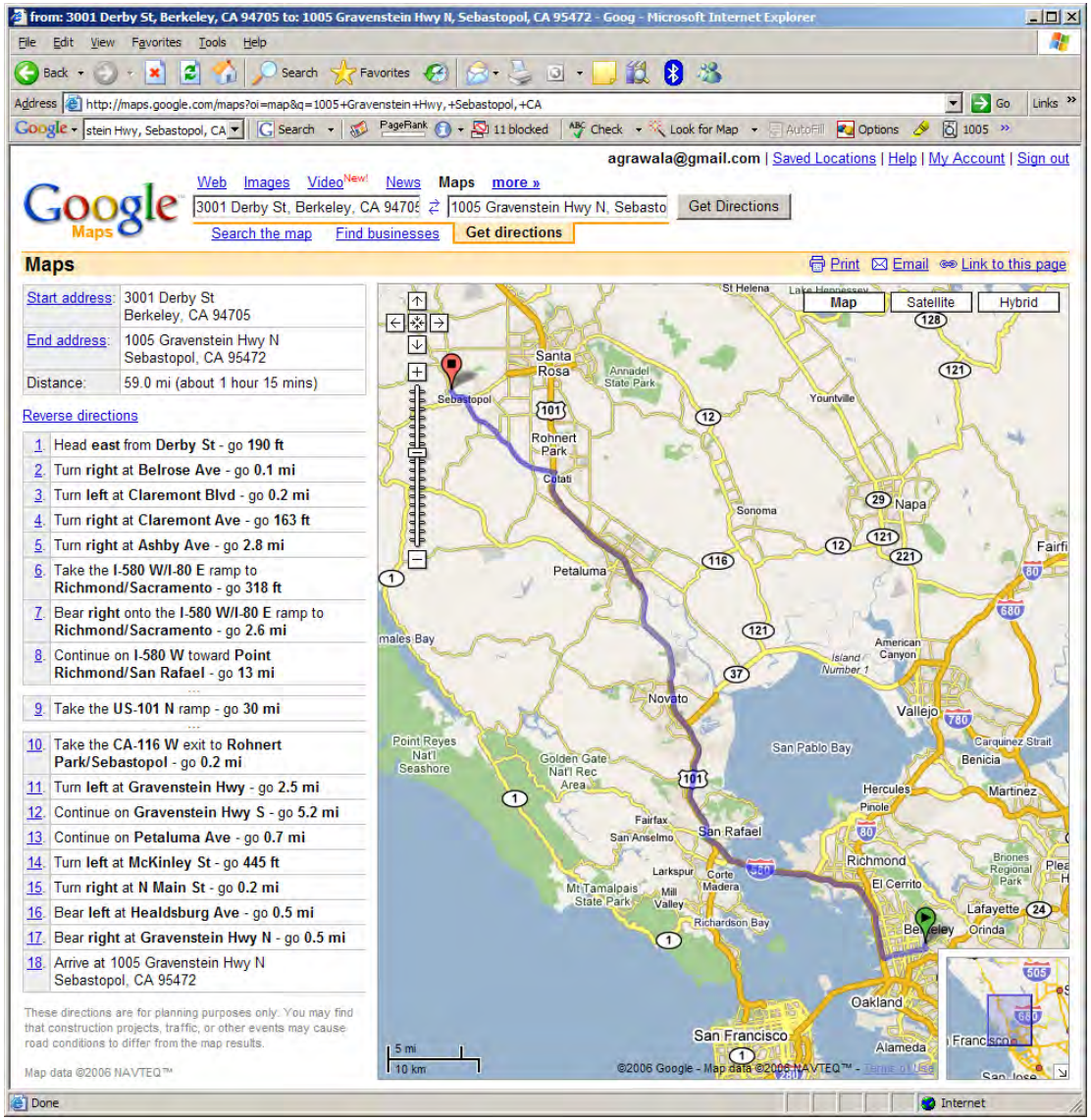


Jeff Han, Perceptive Pixel


Map
Satellite
Hybrid
Santa Rosa
Sebastopol
Annadel State Park
Rohnert Park
Cotati
Petaluma
Sonoma
Yountville
Napa
Fairfi
American Canyon
Island Number 1
Vallejo
Novato
Point Reyes Nat'l Seashore
Golden Gate Nat'l Rec Area
San Pablo Bay
Carquinez Strait
Benicia
Hercules
Martinez
Pinole
Fairfax
San Anselmo
San Rafael
Richmond
El Cerrito
Briones Regional Park
Larkspur
Corte Madera
Mt Tamalpais State Park
Mill Valley
Richardson Bay
Lafayette
Orinda
Oakland
San Francisco
Alameda
5 mi
10 km
©2006 Google - Map data ©2006 NAVTEQ™ -



Where do I need to turn?



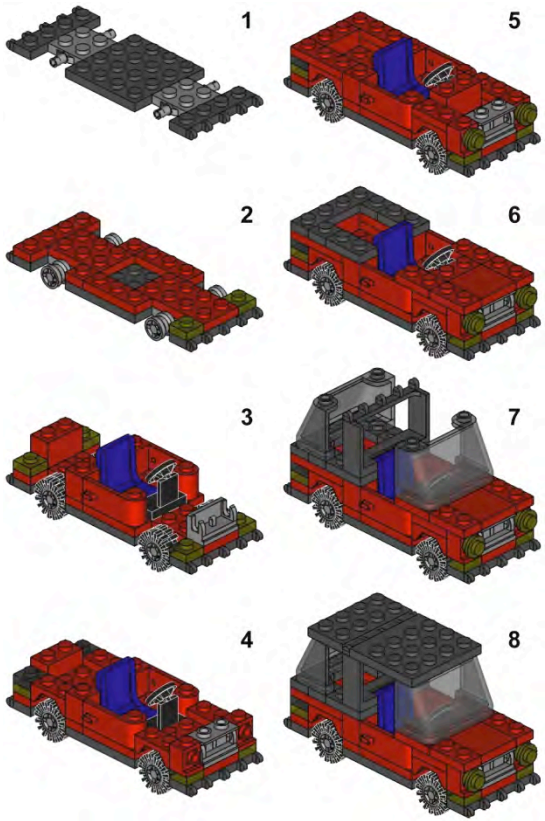
Where do I need to turn?

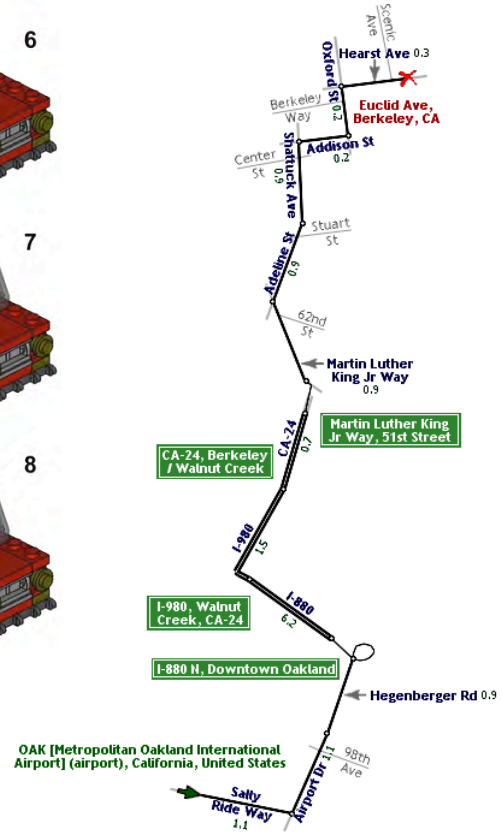
Instructor: Maneesh Agrawala

Associate Professor in EECS, joined Berkeley in 01/2006

Work in Graphics, HCI, and Visualization

- Visual Interface Design
- Perception & Cognition of Displays



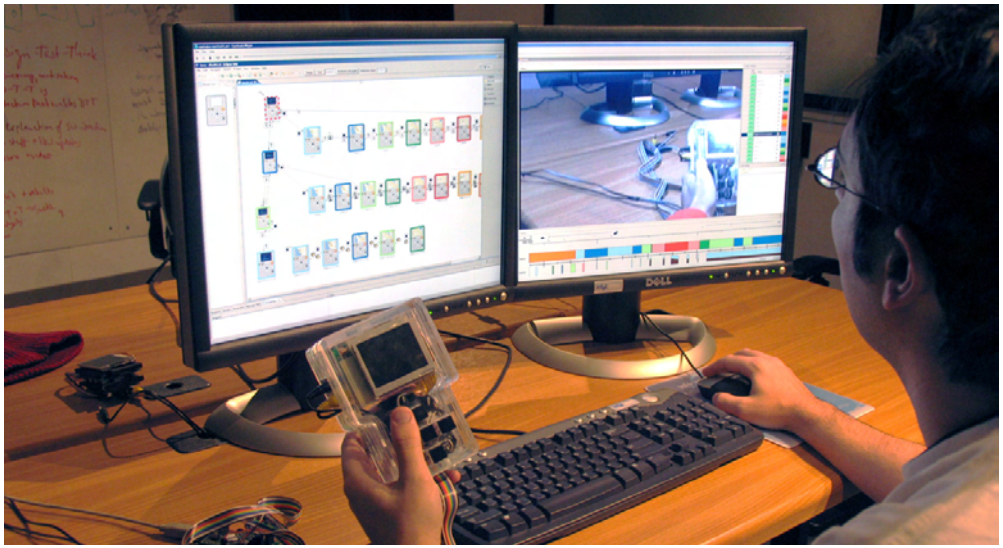


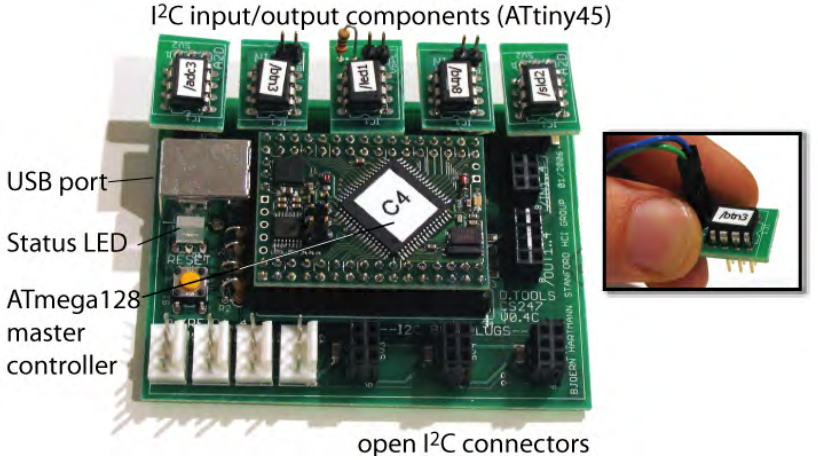
Instructor: Björn Hartmann

Assistant Professor in EECS, joined Berkeley in 01/2010

Research in HCI & Ubiquitous Computing

- User Interface Design Tools
- End-user Programming
- Ubiquitous Computing





GSI: Kenrick Kin

kenrick(at)cs.berkeley.edu

523 Soda Hall

4th year PhD student

Working on: Multitouch interfaces

Awesome

GSIs: Anuj Tewari

Topics for Today

1. Course Overview
2. Project Description
3. Course Mechanics

Course Overview

HCI, UI, Usability, Iterative Design

This Course

Is about reliably building very good interactive systems.

This semester we focus on **mobile applications**.

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

We place emphasis on **fieldwork**, **rapid prototyping** and **user testing** to find the right design and avoid obvious and not-so-obvious mistakes.

Human-Computer Interaction (HCI)

Human

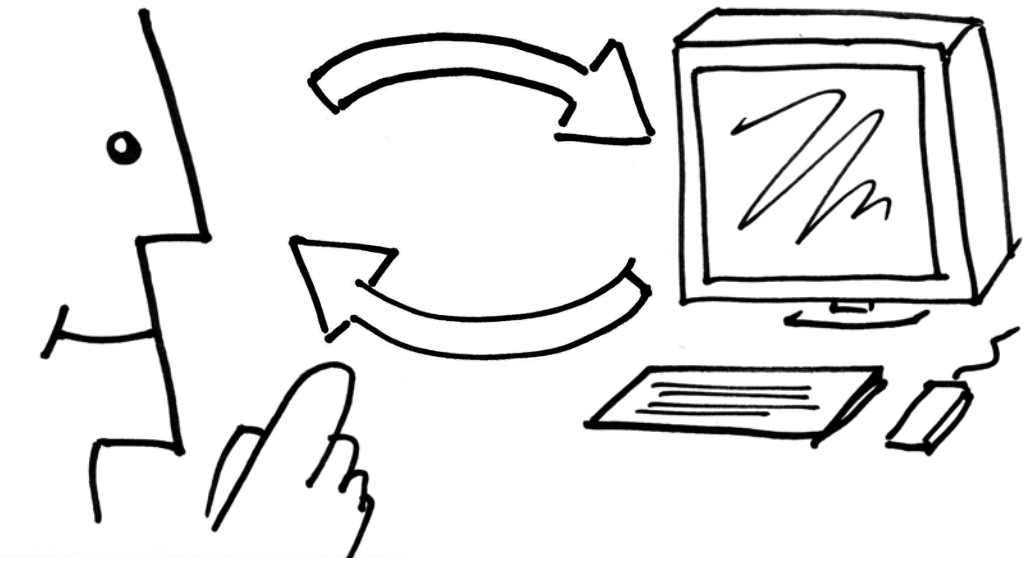
- End-user of program
- Others (friends, collaborators, coworkers)

Computer

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

Interaction

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



User Interfaces (UIs)

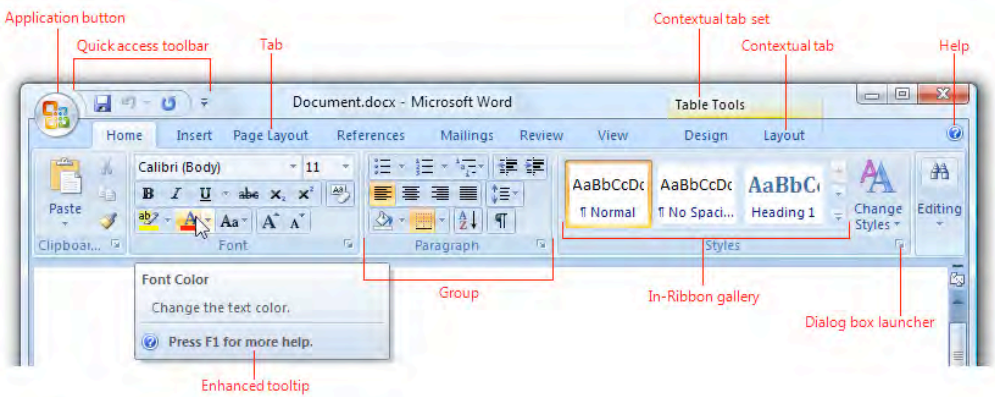
Part of application that allows

- 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?

"The results show that in today's applications, an average of ***48% of the code*** *is devoted to the user interface portion.*

The average time spent on the user interface portion is 45% during the design phase, 50% during the implementation phase, and 37% during the maintenance phase."

– Myers & Rosson, CHI'92

Why Study User Interfaces?

Major part of work for "real" programs (approx 50%)

You will work on "real" software

Intended for people other than yourself

Bad user interfaces cost

Money, Lives, Votes, …

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?

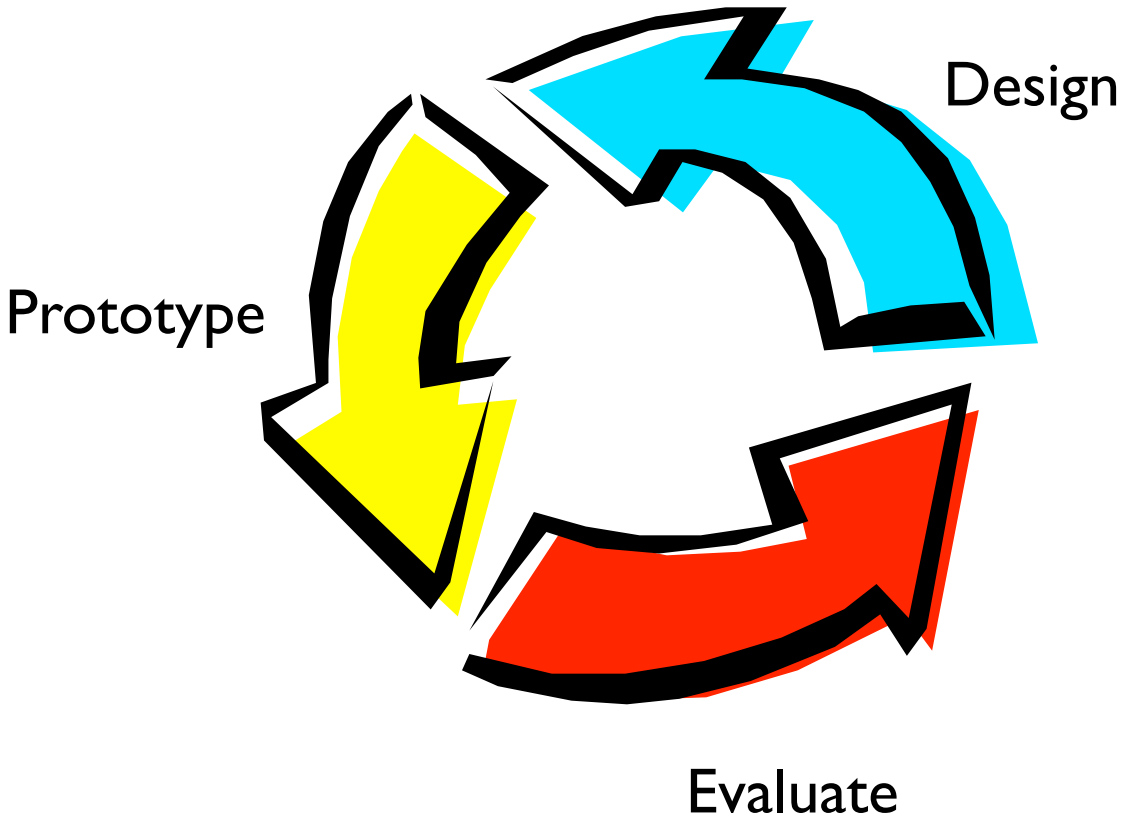
http://en.wikipedia.org/wiki/American_Airlines_Flight_965

Who Builds Interfaces?

Ideally a team of specialists

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

Interface Design Cycle



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

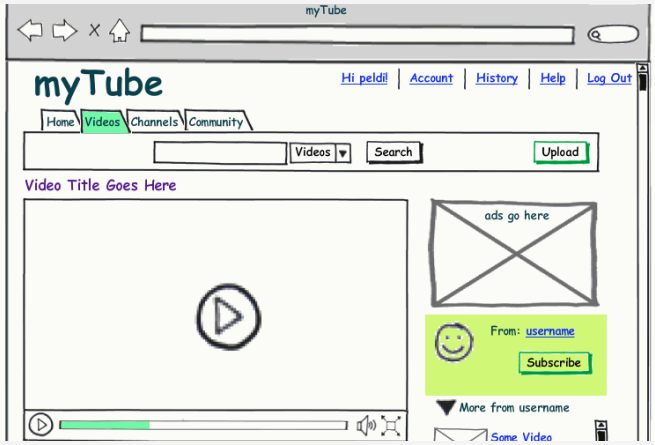
Interactive prototyping tools

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

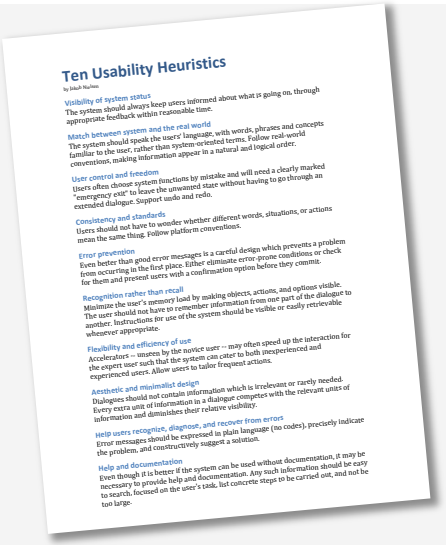
Low-cost techniques

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.

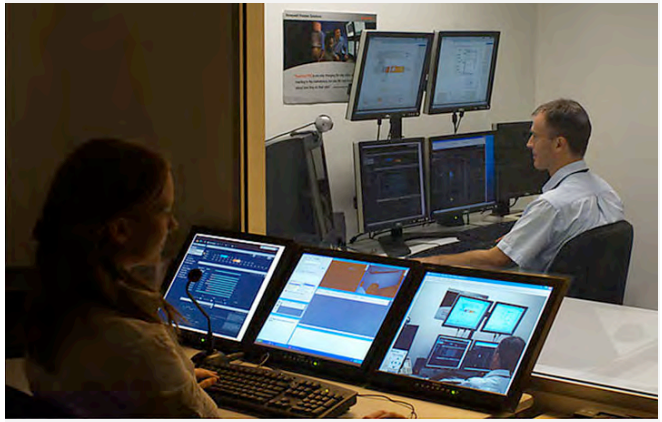
Recognition rather than recall
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable wherever appropriate.

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

Goals of the Course

Learn to design, prototype, evaluate interfaces

- Discover tasks of prospective users
- Cognitive/perceptual constraints that effect design
- Techniques for evaluating an interface design
- Importance of iterative design for usability
- Technology used to prototype & implement UI code
- How to work together on a team project
- Communicate your results to a group

Many of these will be key aspects of your future jobs

CS160 and the CS Curriculum

Most courses for learning algorithms and technology

Compilers, operating systems, databases, etc.

CS160 concerned with
design, implementation & evaluation

We assume you are comfortable programming

Technology as a tool to evaluate via prototyping

Class Project Overview

Mobile Applications, Developed in Teams

Theme: Mobile Applications

Mobile applications are different:

- Different tasks (local search, not word processing)
- I/O constraints (slow text entry, few pixels)
- Input opportunities: Sensing (orientation, acceleration, location, camera)
- Internet connectivity

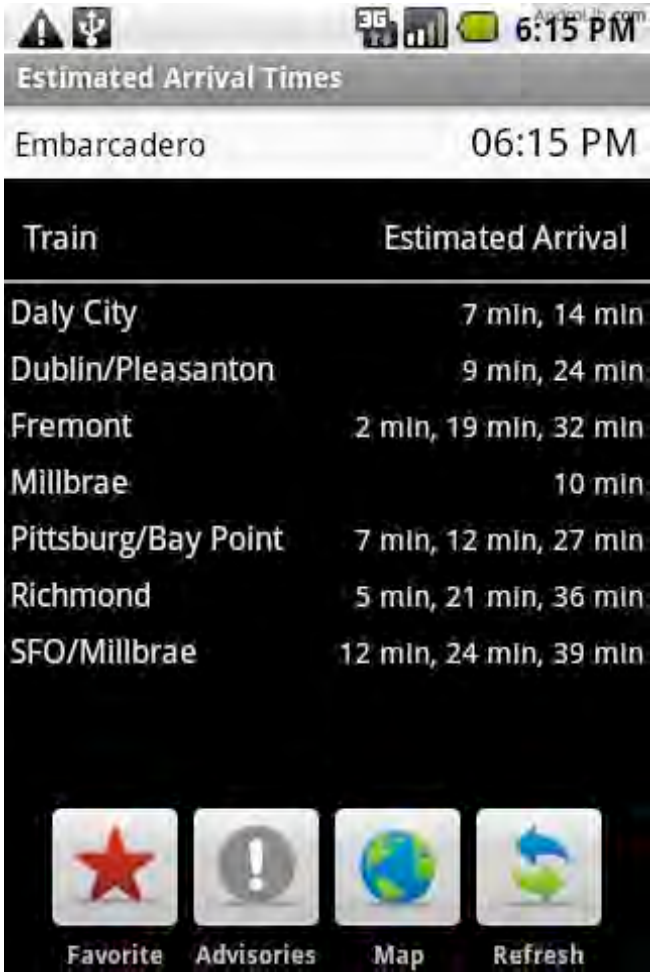
Course Platform

Apple iPhone / iPod Touch

- We have loaner devices (1 iPod touch per team), or use your own.
- Orchard Mac Lab has development environment installed.
- Coding assignments can be completed in simulator.
- Development path:

  Objective C – 4 assignments to get you up to speed

Inspiration: Design for a Particular User



Bart Rider Android App

iBird

Inspiration: Location-based Apps

RedFin iPhone App

Navigon Mobile Navigator

RunKeeper

Inspiration: Input

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

Inspiration: Input

Bump Technologies - http://www.youtube.com/watch?v=kCJ5dyNDfkE

Inspiration: Device-As-Instrument

Smule.com - http://www.youtube.com/watch?v=RhCJq7EAJJA

Inspiration: Device Ecologies

Project Constraints

1. Must be uniquely useful for mobile/handheld devices

   No mobile versions of desktop applications

2. Must have local target users (you'll talk to them!) but must not exclusively target college students

   No alarm clocks, dining hall apps, homework reminders, etc.

Teams

Each of you will individually propose a project idea

- Fixing something you don't like or a new idea
- Novelty and creativity will be considered

Groups will form in week 2

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

Cumulative

- Apply several HCI methods to a single interface

Course Mechanics

Office Hours & Sections, Course Wiki, Assignments

Office Hours, Sections

Office Hours

Maneesh: Tuesday 11-noon in 635 Soda Hall (+by appt)

Björn: Wednesday 1-2pm in 629 Soda Hall (+by appt)

Anuj: TBD

Kenrick: TBD

Sections

Tuesday 2-3pm, 405 Soda,

Wednesday 12-1pm, 310 Soda

You must attend to get full class participation credit

No section this week

Reaching Us

Email:
cs160@imail.eecs.berkeley.edu

Mail sent here will get the fastest response

Please avoid mailing us directly

Class Wiki - http://bit.ly/cs160-sp10

Bjoern Hartmann my talk my preferences my watchlist my contributions log out

User Interfaces
CS160 Spring 2010

page | discussion | edit | history | move | watch

navigation
- Main Page
- Community portal
- Current events
- Recent changes
- Random page
- Help

search
Go Search

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link

Main Page

CS 160 is an introduction to Human Computer Interaction (HCI). You will learn to prototype, evaluate, and design a user interface. You will be expected to work within a group of four or five students in this project-based course. The project topic will be proposed by your group, and your implementation will be tailored to your users' needs based on interviews with them.

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.

Readings

Readings are very important to the class

Make sure you do the reading before class.

Midterm will include topics only covered in readings

Most readings will be posted on wiki

Some require username/password: **cs160/cs160Readings**

Online reading discussions (**ongoing assignment**)

You must post one substantial comment per lecture, **before** class.

We will not accept late comments.

Comments are the major factor in your class participation grade.

Assignments

Four individual programming assignments during first half of semester. Goals:

- Make sure you have the skills to implement your group project
- Individual performance metric

Group project assignments throughout semester

Assessment

The goal of CS160 is to teach you to design and evaluate interfaces.

Specific assessment guidelines will be given in each assignment.

Good communication expected in oral & written presentations.

Groups self-assess participation.

Grading

1. Class & Section participation (10%)
2. Individual Programming Assignments (20%)
3. Project Assignments (50%)
4. Midterm (20%)

Policies

Late Assignments

- Most assignments will be due before class on the due date
- Group assignments will not be accepted late
- Individual assignments lose 33% per day (weekends count)

Cheating (official)

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

Assignments

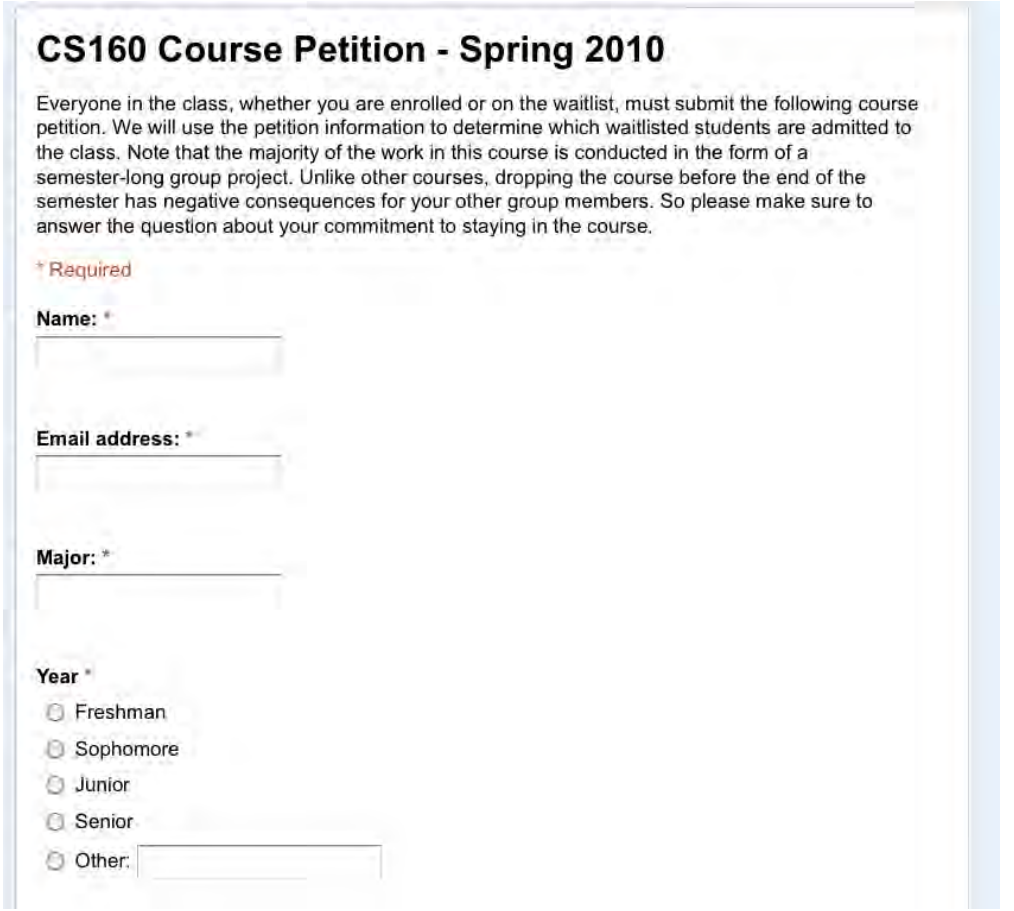
Assignment 1: Course Petition

Due **Friday, Jan 22**, 5pts

Both **enrolled and waitlisted** students have to submit

Information **will determine admission**

http://bit.ly/cs160-sp10-petition



CS160 Course Petition - Spring 2010

Everyone in the class, whether you are enrolled or on the waitlist, must submit the following course petition. We will use the petition information to determine which waitlisted students are admitted to the class. Note that the majority of the work in this course is conducted in the form of a semester-long group project. Unlike other courses, dropping the course before the end of the semester has negative consequences for your other group members. So please make sure to answer the question about your commitment to staying in the course.

* Required

Name: *

Email address: *

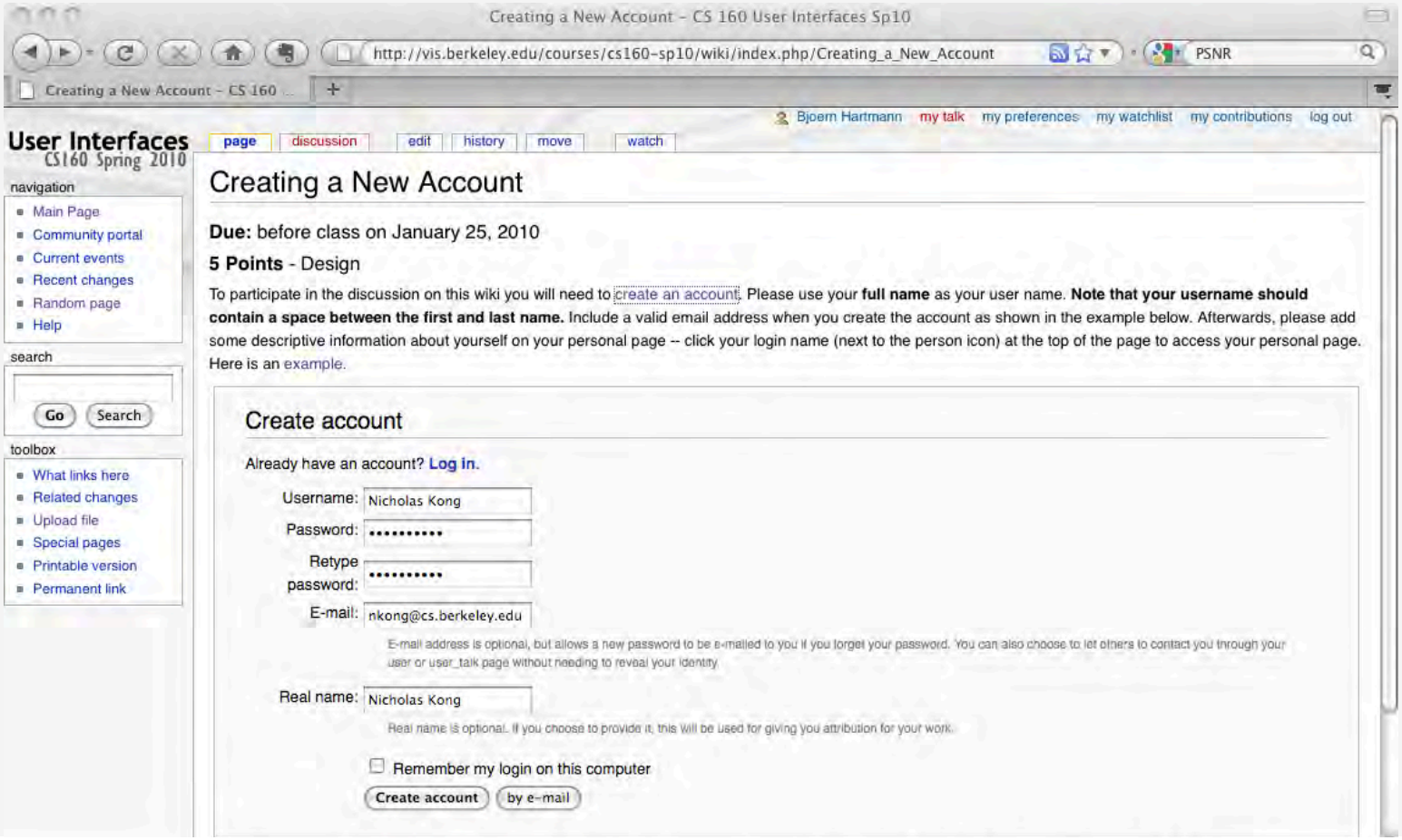
Major: *

Year *

- Freshman
- Sophomore
- Junior
- Senior
- Other:

Assignment 2: Create Wiki Account

Due Friday, Jan 22, 5pts Use Your **Full Name**



Assignment 3: Individual Project Idea

Due before class **Wednesday**, Jan 27; 5pts

Start gathering ideas now! Project should be:

Exciting to you!

Creative!

Consider the needs of a well-defined target user group

Include sketches to visualize your ideas

Grading details on the web (20 points total)

Description must be posted to wiki before class on 01/27

Assignment 4: Hello, World!

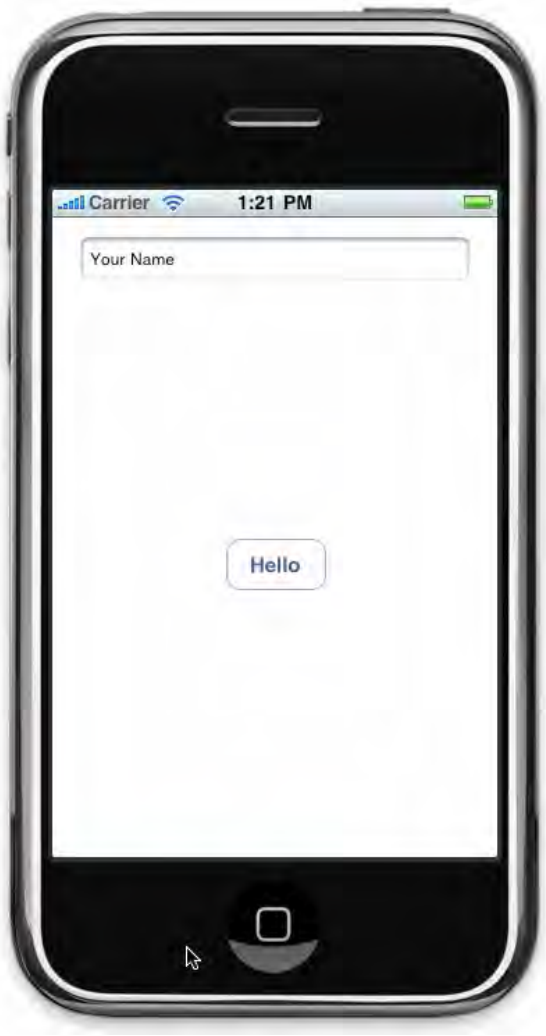
Due before class

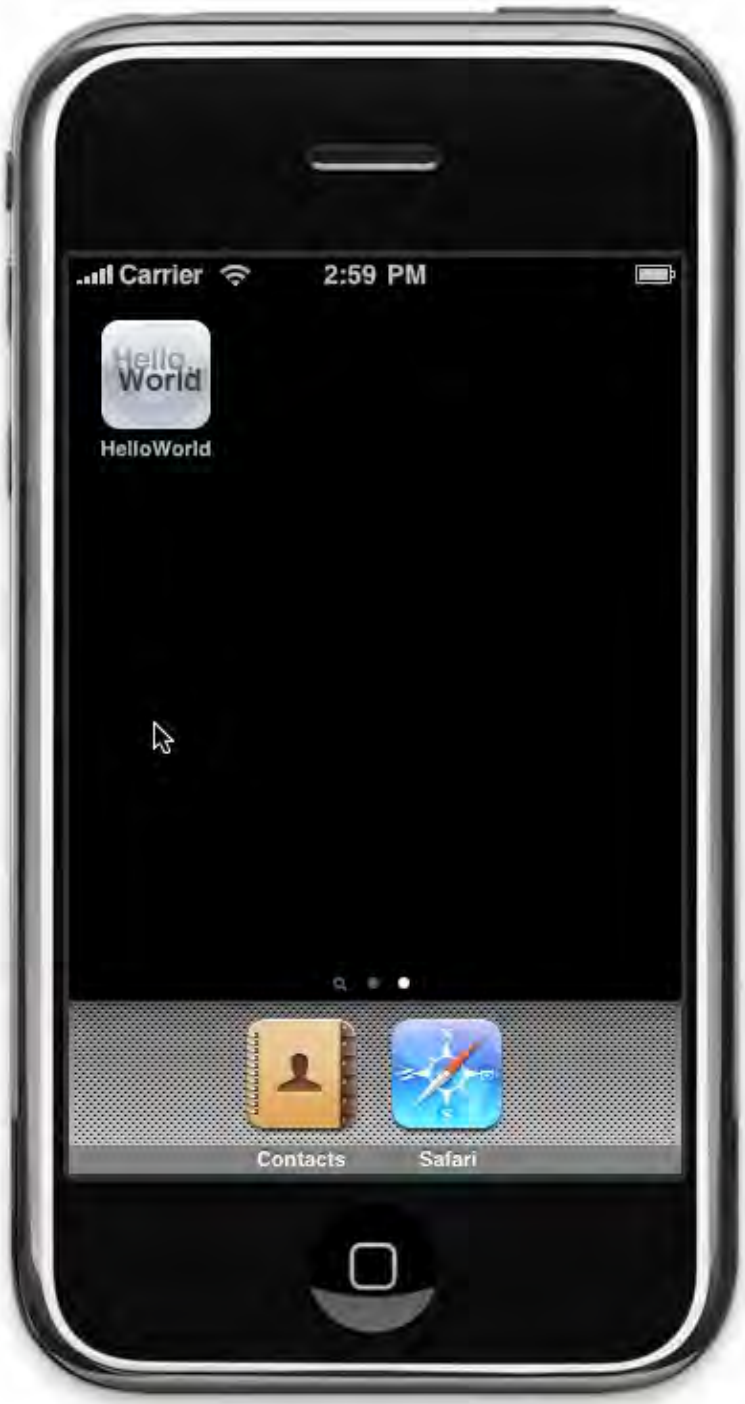
Monday, Feb 1; 5pts

Instructions on wiki. Summary:

Set up XCode development environment and follow Apple tutorial to create a simple Hello World app in the iPhone simulator.

Submit your binary and source to us.




Carrier
2:59 PM
Hello World
HelloWorld
Contacts
Safari

Reading Assignment

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
(cs160/cs160Readings)

CS160: User Interface Design

Maneesh Agrawala & Björn Hartmann, *Spring 2010*

Berkeley
UNIVERSITY OF CALIFORNIA