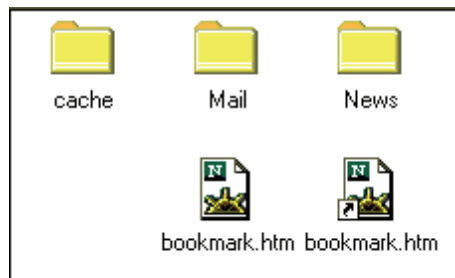
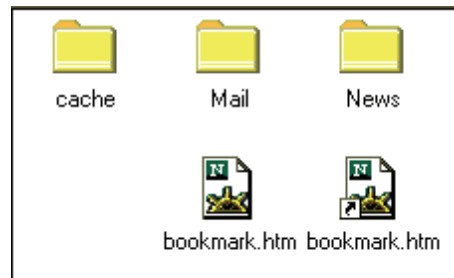


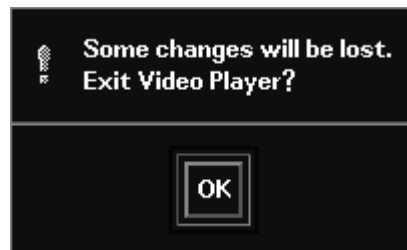
# CSI 60: User Interfaces

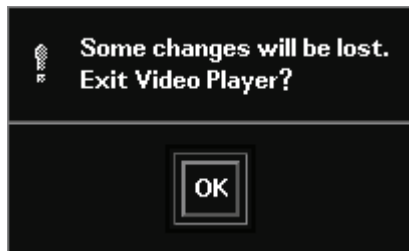
Maneesh Agrawala



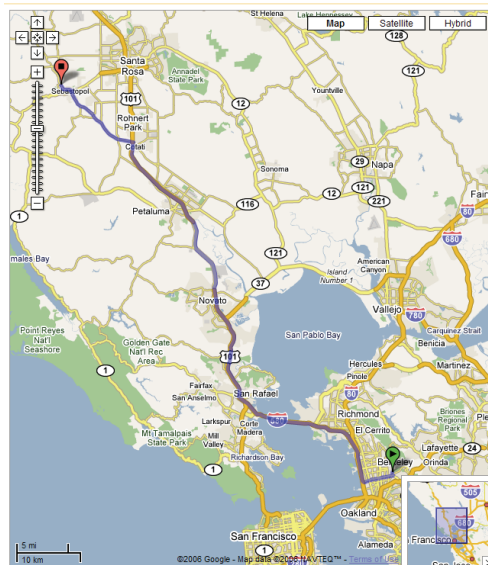


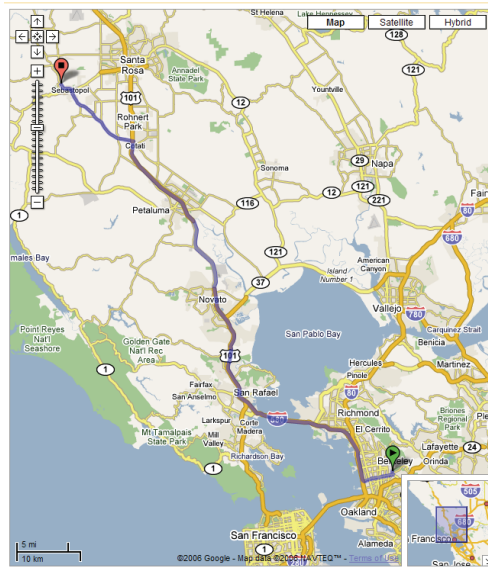
Hard to tell the difference between the icons & names





How do you cancel?





Where do I need to turn?

From 3001 Derby St, Berkeley, CA 94705 to 1005 Gravenstein Hwy N, Sebastopol, CA 95472 - Google Maps

Address: <http://maps.google.com/maps?hl=en&map=1005+Gravenstein+Hwy+N,+Sebastopol,+CA>

Distance: 59.0 mi (about 1 hour 15 mins)

Reverse directions

- 1 Head east from Derby St - go 190 ft
- 2 Turn right at DeRose Ave - go 0.1 mi
- 3 Turn left at Claremont Blvd - go 0.2 mi
- 4 Turn right at Claremont Ave - go 163 ft
- 5 Turn right at Ashby Ave - go 2.8 mi
- 6 Take the I-580 W/I-80 E ramp to Richmond/Sacramento - go 318 ft
- 7 Bear right onto the I-580 W/I-80 E ramp to Richmond/Sacramento - go 2.6 mi
- 8 Continue on I-580 W toward Point Richmond/San Rafael - go 13 mi
- 9 Take the US-101 N ramp - go 30 mi
- 10 Take the CA-116 W exit to Robert Park/Sebastopol - go 0.2 mi
- 11 Turn left at Gravenstein Hwy - go 2.5 mi
- 12 Continue on Gravenstein Hwy S - go 5.2 mi
- 13 Continue on Petaluma Ave - go 0.7 mi
- 14 Turn left at McKinley St - go 445 ft
- 15 Turn right at N Main St - go 0.2 mi
- 16 Bear left at Healdsburg Ave - go 0.5 mi
- 17 Bear right at Gravenstein Hwy N - go 0.5 mi
- 18 Arrive at 1005 Gravenstein Hwy N Sebastopol, CA 95472

These directions are for planning purposes only. You may find that construction projects, traffic, or other events may cause road conditions to differ from the map results.

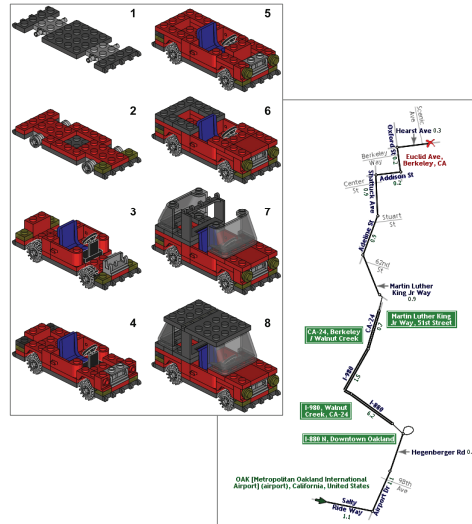
Map data ©2006 NAVTEQ™

Where do I need to turn?

# Instructor: Maneesh Agrawala

**Asst. Professor in EECS**  
Joined Berkeley Jan. 2006

**Work in HCI, Graphics, Vis.**  
Visual interface design  
Perception/cognition of displays



## Topics

- Course Overview
- Project Description
- Course Mechanics

# Course Overview

## Human-Computer Interaction (HCI)

### Human

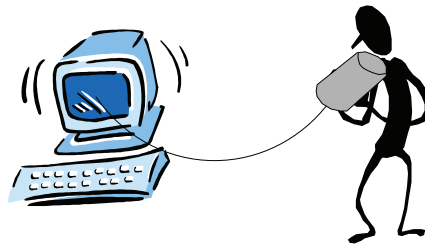
- End-user of program
- Others in the organization

### Computer

- Machine the program runs on
- Often split between 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



User vs. Customer vs. Client

- **User:** term only used by 2 industries
- **Customer:** person who will use the product you build
- **Client:** person/company who is paying you to build it

HCI = design, prototyping, evaluation, & implementation of UIs

# Why Study User Interfaces?

Major part of work for “real” programs

- Approximately 50%

You will work on “real” software

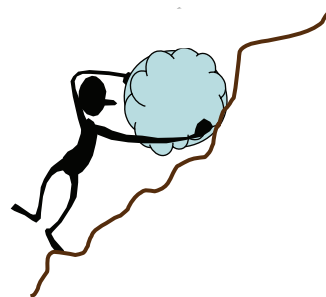
- Intended for people other than yourself

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 Am. 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

[http://en.wikipedia.org/wiki/American\\_Airlines\\_Flight\\_965](http://en.wikipedia.org/wiki/American_Airlines_Flight_965)

## Life-Threatening Errors

1995 Am. 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](http://en.wikipedia.org/wiki/American_Airlines_Flight_965)



## **What is Usability?**

### Ease of learning

- Faster the second time and so on...

### Recall

- Remember how from one session to the next

### Productivity

- Perform tasks quickly and efficiently

### Minimal error rates

- If they occur, good feedback so user can recover

### High user satisfaction

- Confident of success

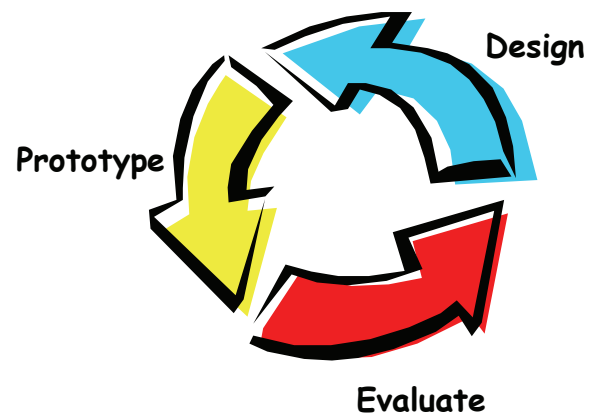
## **Who Builds Interfaces?**

### 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

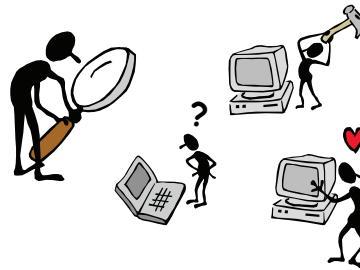


## Building Successful Interfaces

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

# Task Analysis & Contextual Inquiry

- Observe existing work practices
- Create scenarios of actual use
- Try-out new ideas before building software



# Rapid Prototyping

Build a mock-up of design

Low fidelity techniques

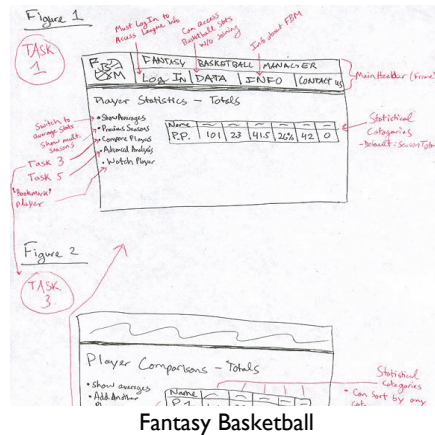
- Paper sketches
- Cut, copy, paste
- Video segments

Interactive prototyping tools

- HTML, Visual Basic, HyperCard, Director, etc.

UI builders

- Fusion, NeXT, Visual Cafe



# Evaluation

Test with real customers (participants)

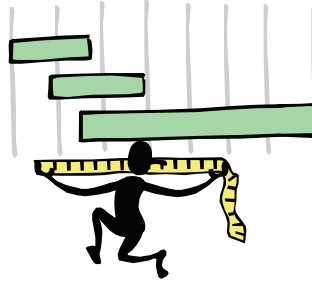
Build models

Low-cost techniques

- expert evaluation
- walkthroughs

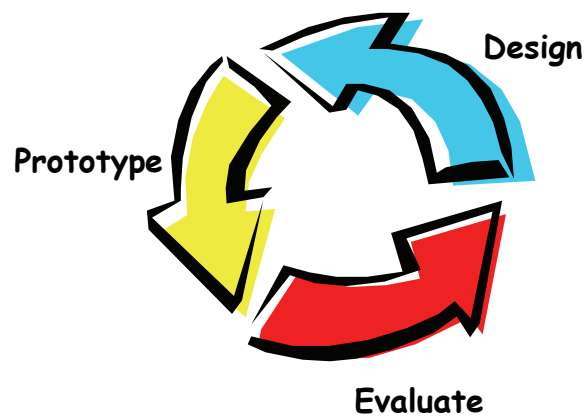
Higher cost

- Controlled usability study



# Iteration

At every stage!



## 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

## CSI60 and the CS Curriculum

Most courses for learning algorithms and technology

- Compilers, operating systems, databases, etc.

CSI60 concerned with **design**, *implementation* & **evaluation**

- Assume you are comfortable programming
- Technology as a tool to evaluate via prototyping
- Skills will become very important upon graduation
  - Complex systems, large teams
  - Don't look for large immediate impact in other CS courses

# Project Description

## Teams

Each of you will individually propose an interface idea

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

### Groups

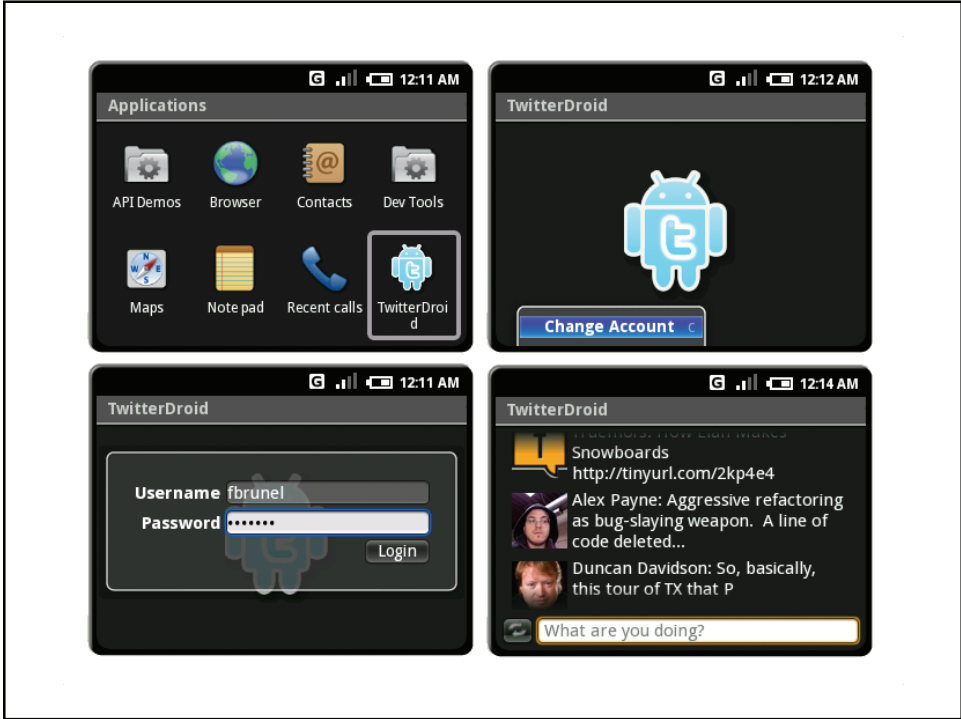
- 3 or 4 students to a team
- Work with students with different skills/interests

### Cumulative

- Apply several HCI methods to a single interface

# Theme: Mobile Applications







## **Why are mobile apps interesting?**

## **Google's Android**

Video: <http://code.google.com/android/>

What applications do they show?

## **Project: Mobile Applications**

Projects must use Android

SW emulator available now

May have some HW later

We will assume you are comfortable coding in JAVA (61B)

We will cover some aspects of Android APIs in sections

We will expect you to learn aspects of the APIs on your own

## **Course Mechanics**

## **TAs, Office Hours, Sections**

### **Teaching Assistants**

- Wesley Willett: EECS grad student
- Seth Horrigan: EECS grad student

### **Office Hours**

- Maneesh: TBD in 635 Soda Hall
- Wes: F 10-11am, Soda 511
- Seth: M 9-10am, Soda 551
- Also by appointment

### **Sections**

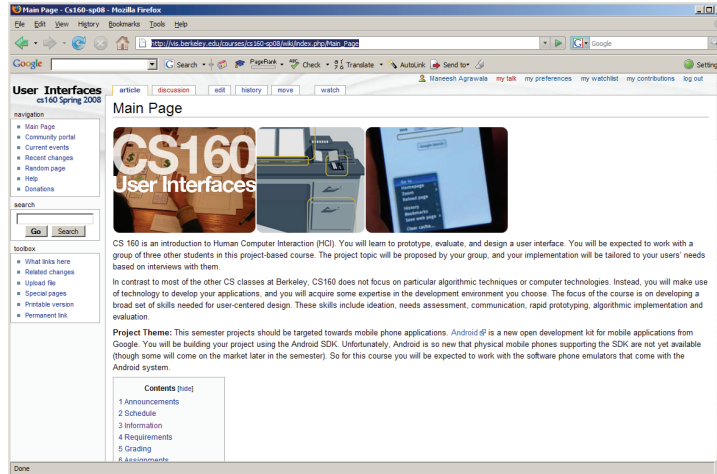
- T 1-2pm 320 Soda, T 2-3pm 310 Soda
- Will cover new material. You should attend!
- No section this week

## **Reaching Us**

**Email:** [cs160@imail.eecs.berkeley.edu](mailto:cs160@imail.eecs.berkeley.edu)

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

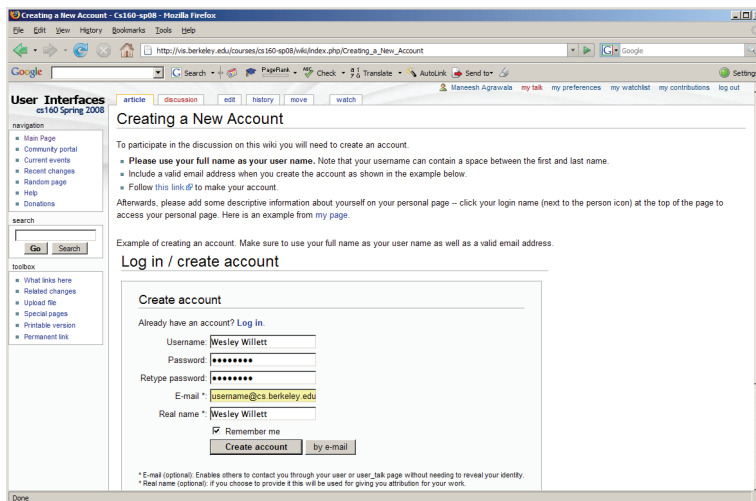
# Class Wiki



[http://vis.berkeley.edu/courses/cs160-sp08/wiki/index.php/Main Page](http://vis.berkeley.edu/courses/cs160-sp08/wiki/index.php/Main_Page)

# Create Wiki Account

Your 1<sup>st</sup> assignment (due before class Th Jan 24)



# Course Petition

Your 2<sup>nd</sup> assignment (due before class Th Jan 24)

Petition for Admission to CS160

---

Name:

Email:

Major:

Year: (Freshman, Sophomore, Junior, Senior)

GPA:

Are you committed to remaining in the course through the semester and collaborating with teammates on a group project?:

Reasons for taking the course:

What skills you would bring to team projects:

Relevant experience (employment or undergraduate research):

**Email:** cs160@imail.eecs.berkeley.edu

Both **enrolled and waitlisted** students should send us petition

Information **will determine admission** for **waitlisted** students

# Readings

Readings are very important to the class

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

Most readings will be posted on wiki

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

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

## Grading

- Class participation (20%)**
- Individual assignments (20%)**
- Group project (40%)**
- Midterm (20%)**

## Schedule of Assignments

- Create Wiki Account (individual) (1/24)
- Course Petition (individual) (1/24)
- Individual Project Proposal (individual) (1/31)
- Group Brainstorm (2/7)
- Contextual Inquiry and Task Analysis(2/19)
- Hello-World Application (individual) (2/26)
- Low-Fidelity Prototype (3/11)
- Interactive Prototype (4/1)
- Team Assessment (4/8)
- Pilot Usability Study (individual) (4/15)
- Final Presentation and Report (4/29)
- Final Team Assessment (5/8)

**Some weeks lighter than others (plan accordingly)**  
**Most assignments turned in through the wiki**  
**Most assignments oriented around project**

# 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  
<http://www-inst.eecs.berkeley.edu/~cs160>

# 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

# Next Time

## The Design Cycle and Brainstorming

- [The Task-Centered Design Process](#). *Task-Centered User Interface Design*. Chap I. Lewis & Rieman
- [The Perfect Brainstorm](#). *The Art of Innovation*. Kelley  
Will need username/password for this one