You must sign in to be considered for a spot in the class.

If 306 is full, go to our overflow room: 310 Soda.

If 310 Soda is full watch live video online: http://bit.ly/cs160-sp14-lecture01
Generated random password is: 'oOmunHz&wCql#FL#|tiTh#GQ:sc/mI:'
Make sure you write this down because it will be needed for future upgrades.

http://stackoverflow.com/questions/238177
splat:~$ cd development/
splat:~/development$ ls
apps personal proposals teaching webpage
jobtalk2 phd recs testCode
machineSetup presentations reviews tmp
splat:~/development$ cd teaching/cs160-sp12/ppt/
splat:~/development/teaching/cs160-sp12/ppt$ ls
01-intro 04-brainstorm
02-design-cycle-brainstorming 05-task-analysis-ci
03-sketching
splat:~/development/teaching/cs160-sp12/ppt$
The Desktop Metaphor...
Is this a good idea? When?

http://www.bumptop.com
How about this?
Where do I need to turn?
Where do I need to turn?
Instructor: Maneesh Agrawala

Professor in EECS, joined Berkeley in 01/2006

Work in HCI, Graphics and Visualization
  • Visual Interface Design
  • Perception & Cognition of Displays
Instructor: Björn Hartmann

Assistant Professor in EECS, joined Berkeley in 01/2010

Teaching:
• CS160
• CS260B: HCI Research
• CS294-84: Interactive Device Design

Research:
• User Interface Design Tools
• Personal Digital Fabrication
• Crowdsourcing
tap to connect
(U)GSIs and Readers:

Brittany Cheng
Steve Rubin
Eric Xiao
Elizabeth Lin
Topics for Today

1. Enrollment
2. Course Overview
3. Project Description
4. Course Mechanics
5. Assignments
Enrollment
Design- and project-centric courses don’t scale well. We have 250 interested students; at most 100 seats. There is no way to avoid disappointing many of you.

We’ll attempt to make decisions quickly – by next Monday’s class. These will be final.
You must sign in today and submit the online petition by Friday, 11:59pm. If you don’t we’ll drop you.
You **must** complete two short assignments (a reading response and a design exercise) by Friday, 1/24, 11:59pm. If you don’t, we’ll drop you.
CS160 has a Heavy Workload

Workload is heavier than for courses of comparable credit
If this is not the class for you, please **drop immediately**, to give others a fair chance to get in.
No room? Can’t see the screen?

All relevant information is on the class wiki. We are recording this lecture – you can watch it live online or archived on the wiki by tomorrow.

Class Wiki:
http://hci.berkeley.edu/cs160

Video for this lecture:
Course Overview

HCI, UI, Usability, Iterative Design
This Course

Is about reliably building very good interactive systems.

This semester we focus on mobile applications that use sound input as a key feature.

The goal is to build a working 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 & 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
Pilots in Crash Were Confused About Control Systems, Experts Say

http://www.nytimes.com/2013/12/12/us/asiana-airlines-crash-san-francisco-airport.html?_r=0

Officials with the Federal Aviation Administration and Boeing at a hearing Wednesday about the fatal crash of an Asiana jet in July.
Interface Design Cycle

Design → Prototype → Evaluate → Design
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

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

CS247, Stanford, 2006
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

http://www.balsamiq.com/products/mockups/examples#wiki
http://www.nngroup.com/reports/prototyping/video_stills.html
Moggridge, Designing Interactions, p.704
Evaluation

Evaluate analytically (no users)

Test with real target users

Low-cost techniques
  expert evaluation
  walkthroughs

Higher cost
  Controlled usability study

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 affect 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, prototyping & evaluation
We assume you are comfortable programming.
Technology as a tool to evaluate design ideas via prototyping.
Class Project Overview

Mobile Apps with Sound, Developed in Teams
Mobile applications are different:

• Different tasks (local search, not word processing)
• Different context (portability)
• I/O constraints (slow text entry, small ocular angle)
• Input opportunities: Sensing (sound, multi-touch, IMU sensing, location, camera)
Project Constraints

Must make use of sound input in a non-trivial way
   When is sound-based interaction appropriate?
   Who are the target users? (Be specific.)
   What are the target tasks? (Be specific.)

Target user group cannot be overly broad
   College students is too broad a group

Must solve a real-world problem
   Not simply a game, especially if similar game already exists
Inspiration: Device-As-Instrument

Smule.com - http://www.youtube.com/watch?v=RhCjq7EAJJA
Voice Recognition

Google Voice Search - http://www.youtube.com/watch?v=y3z7Tw1Kl7A
Previous Projects

**Grocery Guardian:**

Checks whether foods are safe for you and your friends with food allergies.

Detects product ingredients with a barcode scanner.
Previous CS 160 Projects

**NetSnow:**
Social network on the slopes.
Previous CS160 Projects

**Yo DJ!**

Crowdsourced party playlists with iTunes integration
Get a recent Android device if you can
(Nexus 7: $200 = little more than the cost of textbook)

We have a few loaners to share, but having your own
will make things much easier for you
All individual assignments can be accomplished in emulator

Development path
Java + Android SDK – 3 hws to get you up to speed
Use Eclipse with Android extensions
(works on Win, Mac, Linux)

2nd Floor Soda Hall Linux machines have dev tools installed
Most students prefer to use their own laptop
Teams

Instructors will form groups in week 2
• 4-5 students to a team
• You’ll work with students with different skills/interests

Team Project is Cumulative
• Apply several HCI methods to a single interface
Course Mechanics

Office Hours & Sections, Course Wiki, Assignments
Prerequisites

You must be comfortable with programming.

Individual programming assignments require you to write code in Java with the Android SDK.

You must be able to attend one of the sections.

You must commit to working with your assigned team on your group project.
Office Hours

Maneesh: 535 Soda Hall, T 4-5pm and by appointment
Bjoern: 533 Soda Hall, M 4:30-5:30pm and by appointment
Brittany: 651 Soda Hall Alcove, Th 2:30-3:30pm
Steve: 510 Soda Hall (Visual Computing Lab), W 1-2pm
Eric: 651 Soda Hall Alcove, M 11am-12pm
Sections

Thursday 11-noon, noon-1pm, 4-5pm, 5-6pm 405 Soda

**Section starts this week (tomorrow)**

1\textsuperscript{st} half of the semester: Lecture material + some Android

2\textsuperscript{nd} half of the semester: Design critiques

Attend any section until we’ve sorted out enrollment.

Section leaders will NOT discuss how to get admitted to CS160.
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 (only) 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.

Project Theme: This semester, projects will focus on mobile applications that make use of sound input as a key feature of the application. Mobile applications present unique opportunities (e.g., sensors, microphone, camera) and challenges (e.g., text input). Your team will be developing applications using the Google Android SDK. You can use your own personal Android device for development. We encourage teams to purchase a recent Android device, e.g., the Nexus 7 tablet ($230) if none of your team members have a suitable device yet. We will also have a small number of Android phones and tablet devices available to borrow for the semester (one per team); these are older models and limited in number.

Announcements [edit]

- First Day of Class: Wednesday, January 22, 2014.
  You must at sign in during this class if you are on the wait list to be considered for admission.

Schedule [edit]

Week 1 [edit]

W Jan 22: Introduction [ Slides | Live Lecture Video (later also archived on YouTube) ]

    Overflow: After 306 Soda fills up, please go to our overflow room, 310 Soda. We will post a video of the first lecture online as
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/cs160Reading

Online reading responses (ongoing assignment)
You must post one substantial answer or comment per lecture, before class. We will not accept late responses. Responses are the major factor in your class participation grade.

Response for next Monday is due this Friday and will be part of our admissions decisions.
Reaching Us

Questions about course material, assignments:
http://piazza.com/berkeley/spring2014/cs160

Private questions:
If other students will benefit from an answer, ask publicly on Piazza. If it’s personal, use Piazza’s private messaging feature. You can also post anonymously.

Do not email us directly.
Assignments

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

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

Two design assignments

• Practice design and evaluation
• Also an individual performance metric

Group project assignments throughout semester
Groups self-assess participation
(you evaluate your team mates and vice versa).
Grading

1. Participation
   (Reading responses, class, Piazza - 10%)
2. Individual Programming Assignments (25%)
3. Midterm (25%)
4. Project Assignments (40%)
Policies

Late Assignments
• Most assignments will be due before class on the due date
• Individual assignments lose 33% per day (weekends count)
• Group assignments will **not** be accepted late
• Reading responses will **not** be accepted late.
• All assignments due this Friday will **not** be accepted late.

Cheating (official)
• Will get you an F in the course
• More than once can get you dismissed from Cal
Assignments
Assignments are on the Schedule

Week 1

W Jan 22: Introduction [ Slides | Live Lecture Video (later also archived on YouTube) ]

Overflow: After 306 Soda fills up, please go to our overflow room, 310 Soda. We will pos
Assignment due by 11:59pm on Friday Jan 24: Course Petition
Assignment due by 11:59pm on Friday Jan 24: Response to Monday's Reading
Assignment due by 11:59pm on Friday Jan 24: Individual Design Exercise, Part 1
Assignment due by 11:59pm on Friday Jan 31: Individual Design Exercise, Part 2
Assignment due by 11:59pm on Friday Jan 31: Individual Programming Assignment 1

Th Jan 23: Section 1

F Jan 24: No class, but assignments due by 11:59pm

Due: Course Petition
Due: Reading Response to Monday's Reading
Due: Individual Design Exercise, Part 1
Assignment 1: Course Petition

Due **FRIDAY**, 11:59pm

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

Information **will determine admission**


(Link is also on wiki)

---

**CS160 Course Petition, Spring 2014**

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. Enrolled students who do not submit a petition will be dropped.

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

First Name: *

Last Name: *

Berkeley email address: *

Student ID: *

Major: *

(If Computer Science, please let us know whether EECS or L&S CS)
Chapter 1: The Task-Centered Design Process

1.1 Figure Out Who's Going to Use the System to Do What

The industry terminology for this step is “task and user analysis.” The need for the task analysis should be obvious: if you build an otherwise great system that doesn’t do what’s needed, it will probably be a failure. But beyond simply “doing what’s needed,” a successful system has to merge smoothly into the user’s existing work and work. It should request information in the order that the user is likely to receive it; it should make it easy to correct data that’s often entered incorrectly; its hardware should fit in the space that users have available and look like it belongs there. These and a multitude of other interface considerations are often lost in traditional requirements analysis, but they can be uncovered when the designer takes time to look into the details of tasks that users actually perform.

Understanding of the users themselves is equally important. An awareness of the users’ background knowledge will help the designer answer questions such as what names to use for menu items, what to include in training packages and help files, and even what features the system should provide. A system designed for Macintosh users, for example, should provide the generic Mac features that the users have come to expect. This might mean including a feature like cut and paste even though cut and paste plays no important part in the system’s main functionality. Less quantifiable differences in users, such as their confidence, their interest in learning new systems, or their commitment to the design’s success, can affect decisions such as how much feedback to provide or when to use keyboard commands instead of on-screen menus.
Assignment: Reading Response

Due Friday, 11:59pm.
Answer concrete questions about the reading.
Will be graded.

CS160 Reading Response for 01/24/14

* Required

Your Name*

This is a required question

Your Email *

Please address the following questions in your response:

How does the Lewis & Riemann design cycle compare to the three stage (design, prototype, evaluate) cycle described in lecture?

This article was written at a time when most software was sold "shrink-wrapped" in stores. Does their argument about the design process hold equally for today, when most mobile applications are downloaded through app stores? What changes in the design process for these apps?

Where do you disagree with Lewis & Riemann? Why?

Your Response

Submit
Assignment: Individual Design Exercise

Part 1 Due this Friday by 11:59pm
Part 2 Due next Friday by 11:59pm
Design Brief: Redesign interaction “on the go”

Submission must be posted to wiki.
For this week: observe and interview at least 2 non-college students; document their difficulty in using mobile devices on the go. Upload two photos of you and them.

A mini design cycle: Observe, ideate, prototype, get feedback
Get started right away!
Assignment: Hello, Android!

Due next Friday, Jan 31, 11:59pm;

Instructions on wiki. Summary:
Set up Eclipse development environment for Android and create a simple cryptocurrency converter.
Post description, screenshot and screencast video on the wiki;
Use “submit” command line tool to submit source code.
If this is not the class for you, please drop immediately, to give others a fair chance to get in.