



# The Beauty and Joy of Computing

## Lecture #1 Welcome; Abstraction

UC Berkeley EECS  
Lecturer  
Gerald Friedland

**BJC: YOU'LL LOVE IT!**  
~~BJC: YOU'LL LOVE IT!~~  
Watch the student  
testimonials about the  
course, what it means to  
them, and how it has  
changed their lives.  
Inspiring!



[inst.eecs.berkeley.edu/~cs10/](http://inst.eecs.berkeley.edu/~cs10/)



# BJC in one slide

## ▪ Big Ideas of Programming

- 
- 
- 
- 
- 
- 
- *Programming Paradigms*
- *Concurrency*
- *Distributed Computing*

## ▪ Beauty and Joy

- "CS Unplugged" activities
- All lab work in pairs
- Two 3-week projects in pairs
  - Of their own choice!! (data + prog)
- One **writeup**
  - Of students' own choice!!

## ▪ Big Ideas of Computing

- HowStuffWorks
  - 3D Graphics + Video Games
  - Internet
- Research Summaries
  - AI
  - HCI
- The Power of Data (big, small, etc)
- Apps that Changed the World
- Social Implications of Computing
- Saving the World with Computing
- Cloud Computing
- Limits of Computing
- Future of Computing





# Format & Textbooks

- Format (7 hrs/wk \* 14 wks)

Mon	Tue	Wed	Thu	Fri
Lecture	Lab	Lecture	Lab	Discussion
	Lab		Lab	

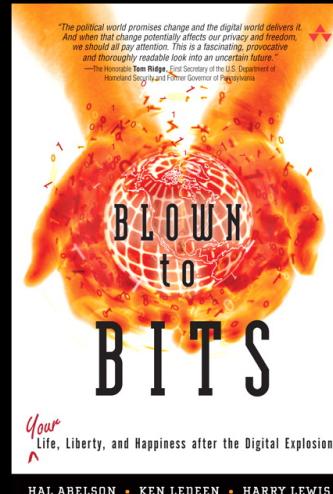
- Selected Reading

- Taken from great book ("Blown to Bits" by Abelson, Ledeen & Lewis) + articles + videos
- Current events EVERY LECTURE (e.g., IBM's Watson vs Jeopardy)

- All resources FREE

- Even clickers!

- Pair Programming!



HAL ABELSON • KEN LEDEEN • HARRY LEWIS

**contributed articles**

**Data generated as a side effect of game play also solves computational problems and trains AI algorithms.**

BY LUIS VON ANN AND LAURA DABISH

## Designing Games With A Purpose

MANY TASKS ARE trivial for humans but continue to challenge computers, such as playing checkers or solving jigsaw puzzles. Traditional computational approaches to solving such problems focus on improving artificial-intelligence algorithms. Here, we advocate a different approach: using games to solve problems and train the brainpower through computer games. Toward this goal, we present general design principles for the development and analysis of games that we call "games with a purpose," or GWAPs, in which people, as a side effect of playing, perform tasks computers are trained to perform.

The Entertainment Software Association ([www.theesa.com/facts/games\\_data.php](http://www.theesa.com/facts/games_data.php)) has reported that more than 200 million Americans are gamers today, playing on computers and video games in the U.S. Indeed, by age 21, the average American has spent more than 10,000 hours playing video games—equivalent to five years of working a full-time job at human speed.

58 COMMUNICATIONS BY ACM | AUGUST 2011 | VOL. 51 | NO. 1

**IS ABSTRACTION THE KEY TO COMPUTING?**

Why is it that some software engineers and computer scientists are able to produce clean, elegant designs and programs, while others cannot? Is it possible to improve their skills through education and training? Critical to these questions is the notion of abstraction.

By KELF KRAMER

For over 30 years, I have been involved in teaching and research in computer science and software engineering. My primary experience has been in distributed systems, distributed algorithms, concurrency, and software design. All these courses require that students are able to perform problem solving, conceptualization, modeling, and abstract expression. In the last few years, my students are clearly able to handle complexity and to produce elegant models and designs. The same students are also able to cope with the complexities of distributed algorithms, the applicability of various modeling notations, and other subtle issues.

When Maurice V. Wainman, Editor-in-Chief of Communications, invited us to submit an article, he recalled how he first learned about Scratch. "A colleague of mine, [UC Berkeley's] Jeff Kramer, had just moved to Princeton University and was interested in programming, and the only thing that appealed to her was Scratch."

Thus began our collaboration, in which we set out to develop Scratch six years ago. We wanted to develop an approach to programming that would appeal to people of all ages, backgrounds, and interests, to program them to create their own programs, simulations, and visualizations, and share their creations with one another. Since the public launch in May 2007, the Scratch community has grown to include more than 10 million users and a vibrant online community, with people sharing

59 COMMUNICATIONS BY ACM | AUGUST 2011 | VOL. 51 | NO. 1

**contributed articles**

**"Digital fluency" should mean designing, creating, and sharing, not just browsing, chatting, and interacting.**

BY MITCHELL BERNSTEIN, JOHN MALONEY, ANDREW MENSENSTEIN, NATALIE RUSK, EVELYN EASTMAN, KAREN SHERMAN, JEFF KRAMER, JEFFREY BRAY, JAY SILVER, BRIAN SILVERMAN, AND YASMIN KAFAI

## Scratch: Programming for All

designing and sharing one another's projects. Scratch has been called "the Scratch of the century" because the Scratchers from around the world continue to push the boundaries of what can be done with mouse, keyboard, and touch screen. The site's collection of projects is widely diverse, ranging from simple games to complex simulations, from the humanities, science, mathematics, and technology to art, music, dance, comedy, and interactive social media.

The new addition to the site is the ScratchJr app, designed for children aged 5 to 7, though a similar group of adults can benefit from it as well. ScratchJr is a simplified version of Scratch that allows younger children to learn the basics of programming, and the only thing that appealed to her was Scratch."

Thus began our collaboration, in which we set out to develop Scratch six years ago. We wanted to develop an approach to programming that would appeal to people of all ages, backgrounds, and interests, to program them to create their own programs, simulations, and visualizations, and share their creations with one another. Since the public launch in May 2007, the Scratch community has grown to include more than 10 million users and a vibrant online community, with people sharing

60 COMMUNICATIONS BY ACM | AUGUST 2011 | VOL. 51 | NO. 1





# Week at a glance

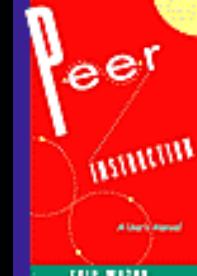
	Mon 9/8	Tue 9/9	Wed 9/10	Thu 9/11	Fri 9/12
9am	Lab 11 TA: Steven 200 SD	Lab 5 TA: Jaclyn 200 SD	Lab 10 TA: Andy 200 SD	Lab 5 TA: Jaclyn 200 SD	Lab 10 TA: Andy 200 SD
10am	Gerald's OH 329 Soda	Janna's OH 411 Soda	Victoria's OH 411 Soda		Disc. 1 TA: Sumner 6 Evans
11am	Lab 12 TA: Joseph 200 SD	Steven's OH Soda-Alcove-651 Cap:10	Lab 6 TA: Max 200 SD	Rachel's OH 611 Soda	Disc. 2 TA: Rachel 3109 Etcheverry
12pm		Sumer's OH Soda-Alcove-751 Cap:10		Joseph's OH 651 Soda	Lab 1 TA: Sumer 200 SD
1pm		Lab 7 TA: Jeff 200 SD	Lab 2 TA: Rachel 200 SD	Lab 7 TA: Jeff 200 SD	Disc. 3 TA: Janna 3105 Etcheverry
2pm					Disc. 4 TA: Victoria 3105 Etcheverry
3pm	Lecture 2050 VLSB	Lab 8 TA: Adam 200 SD	Lecture 2050 VLSB	Lab 8 TA: Adam 200 SD	Disc. 5 TA: Jaclyn 3111 Etcheverry
4pm	Lab 3 TA: Janna 200 SD		Lab 3 TA: Janna 200 SD		Disc. 11 TA: Steven 320 Soda
5pm		Lab 9 TA: Arany 200 SD		Lab 9 TA: Arany 200 SD	Disc. 6 TA: Max 310 Soda
6pm	Lab 4 TA: Victoria 200 SD		Lab 4 TA: Victoria 200 SD		Disc. 7 TA: Jeff 310 Soda
7pm					Arany's OH 411 Soda
					Disc. 8 TA: Adam 310 Soda
					Lab 12 TA: Joseph 200 SD
					Adam's OH 611 Soda
					Andy's OH 411 Soda
					Disc. 9 TA: Arany 310 Soda
					Disc. 10 TA: Andy 310 Soda





# Peer Instruction

- Increase real-time learning in lecture, test understanding of concepts vs. details
- As complete a “segment” ask multiple choice question
  - 
  - 
  - consensus. Teach others!
  - 2 minute discussion of answers, questions, clarifications



Friedland





# Piazza for {ask,answer}ing questions

PIAZZA CS 10 Questions Statistics 35

Search or ask a question... Add Question/Note

Dan Garcia Piazza Help

Popular tags: #instructor-question #admin #logistics #welcome

QUESTION FEED FILTERS

▼ This week

When are TA / professor office hours? Sun  
When can I meet up with a GSI or professor to get help with the course material? #admin  
#instructor-question #admin

▼ Last week

So, I'm here... now how exactly does Pia Mon  
(No question details) sr  
#logistics #welcome

question. 3 Views, 1 Follows Actions

**When are TA / professor office hours?**  
When can I meet up with a GSI or professor to get help with the course material? #admin  
Last updated by Luke Segars 2 days ago

Good Question!

Instructors' response.

We haven't established our office hours yet, but we'll make that information available as soon as possible. Check back here for an update by the second week of classes.  
Last updated by Luke Segars 2 days ago

Actions Good Answer! Ask a Followup »

Start off a Students' Response

followup discussions.

Still Confused? Ask New Followup

AVERAGE RESPONSE TIME SPECIAL MENTIONS USERS ONLINE THIS WEEK

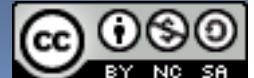
N/A Luke Segars answered When are TA / ... in 1.1 hr. 2 days ago 3 Online Now: 1

About Piazza Privacy Policy Copyright Policy Terms of Use Report a Bug!  
Copyright © 2011 Piazza Inc. All rights reserved.



UC Berkeley "The Beauty and Joy of Computing" : Welcome, Abstraction (6)

Friedland





# Pro-student Grading Policies

- EPA
  - 
  - Effort
    - every single lab, hw, reading Piazza pages
  - Participation
    - E.g., Raising hand in lec or discussion, asking questions on Piazza
  - Altruism
    - E.g., helping other students in lab, answering questions on Piazza
- You have 3 “Slip Days”
  - You use them to extend due date, 1 slip day for 1 day extension
  - You can use them one at a time or all at once or in any combination
  - They follow you around when you pair up (you are counted individually)
    - E.g., A has 2, B has 0. Project is late by 1 day. A uses 1, B is 1 day late
  - Late is 1/3 off/day





# What you will be able to do!

- [https://www.youtube.com/watch?  
v=L8QblaWD6o8](https://www.youtube.com/watch?v=L8QblaWD6o8)
- [https://www.youtube.com/watch?  
v=XqeQSSv8G4](https://www.youtube.com/watch?v=XqeQSSv8G4)
- [https://www.youtube.com/watch?  
v=\\_yAzgt4AGbY&feature=youtu.be  
&hd=1](https://www.youtube.com/watch?v=_yAzgt4AGbY&feature=youtu.be&hd=1)
- [https://www.youtube.com/watch?  
v=-CNTN92ptlo](https://www.youtube.com/watch?v=-CNTN92ptlo)



# Abstraction

- **Detail removal**

- “The act or process of leaving out of consideration one or more properties of a complex object so as to attend to others.”

- **Generalization**

- “The process of formulating general concepts by abstracting common properties of instances”

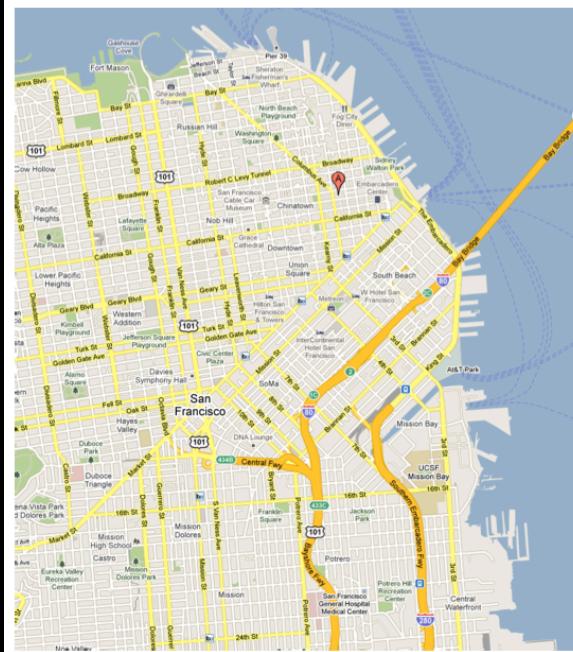


Henri Matisse "Naked Blue IV"

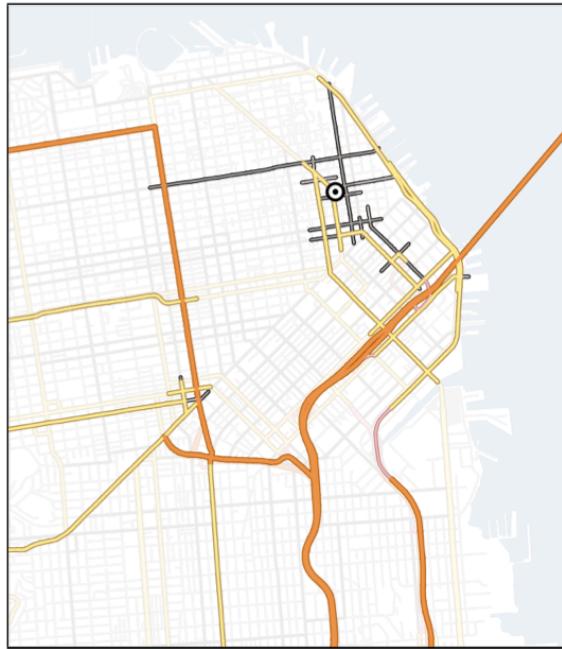




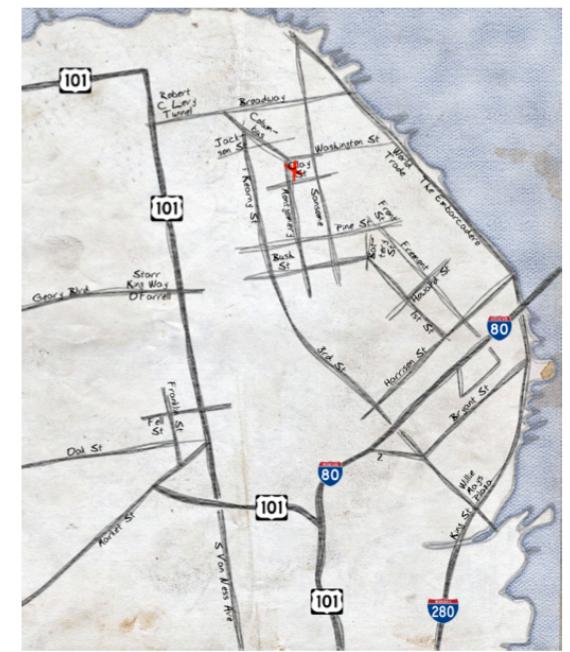
# Detail Removal



General Purpose Online Map



Selected Roads



Our Result

## Automatic Generation of Detail Maps

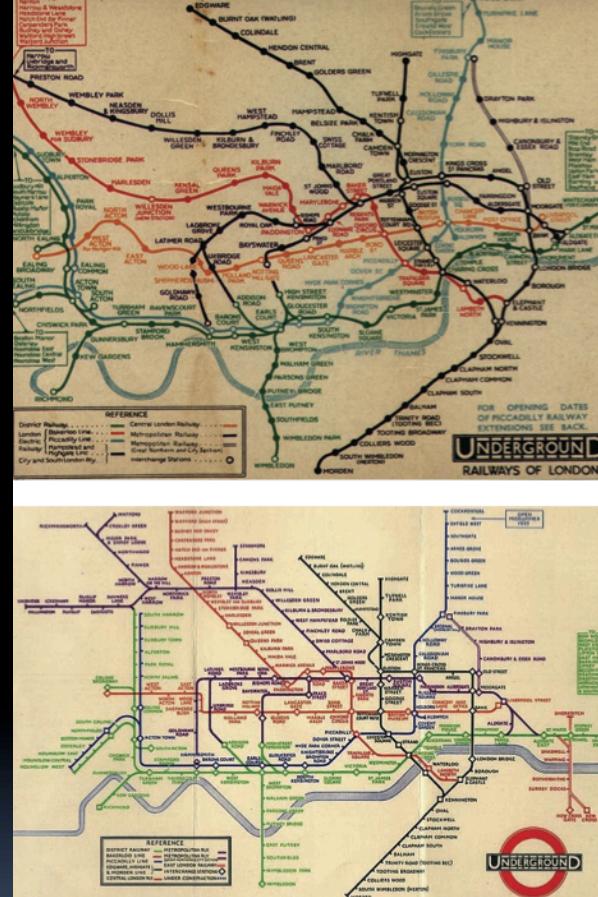
Maneesh Agrawala (UCB EECS), among others





# Detail Removal (in BJC)

- You'll want to write a project to **simulate a real-world situation**, or play a game, or ...
- Abstraction is the idea that you **focus on the essence**, the cleanest way to map the messy real world to one you can build
- Experts are often brought in to know what to remove and what to keep!



The London Underground 1928 Map & the 1933 map by Harry Beck.



# Generalization Example

- You have a farm with many animal kinds.
- Different food for each
- You have directions that say
  - To feed dog, put dog food in dog dish
  - To feed chicken, put chicken food in chicken dish
  - To feed rabbit, put rabbit food in rabbit dish
  - Etc...
- How could you do better?
  - To feed <animal>, put <animal> food in <animal> dish



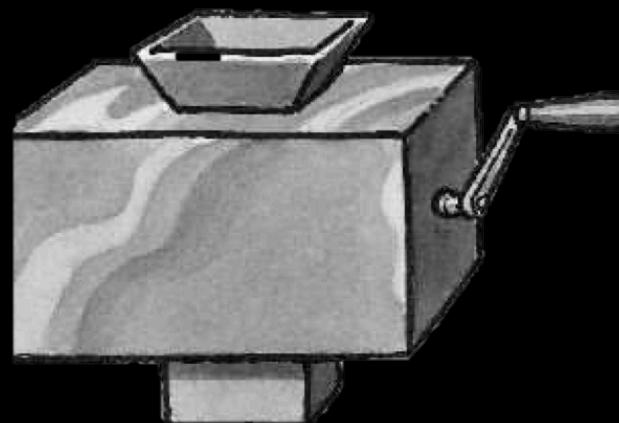
Friedland



# Generalization (in BJC)

- You are going to learn to write functions, like in math class:

$$y = \sin(x)$$



"Function machine" from *Simply Scheme* (Harvey)

- You should think about what inputs make sense to use so you don't have to duplicate code





# The Power of Abstraction, everywhere!

- Examples:

- Functions (e.g.,  $\sin x$ )
- Hiring contractors
- Application Programming Interfaces (APIs)
- Technology (e.g., cars)

- Amazing things are built when these layer

- And the abstraction layers are getting deeper by the day!

*We only need to worry about the interface, or specification, or contract  
NOT how (or by whom) it's built*

## Above the abstraction line

### Abstraction Barrier (Interface)

(the interface, or specification, or contract)

## Below the abstraction line

*This is where / how / when / by whom it is actually built, which is done according to the interface, specification, or contract.*





# Summary

- Abstraction is one of the big ideas of computing and computational thinking
- Think about driving. How many of you know how a car works? How many can drive a car?  
Abstraction!



Someone who drove in 1930 could still drive a car today because they've kept the same Abstraction!

*(right pedal faster, left pedal slow)*

