



Jon McKinsey



Michael Ball



The Beauty and Joy of Computing



Lecture #1 Welcome; Abstraction

BJC: YOU'LL LOVE IT! HARD WORK, BUT FUN!

Watch the student
testimonials about the
course, what it means to
them, and how it has
changed their lives.
It's quite Inspiring!





Computing in the News: BJC4NYC

- We're working with the NYC school district to bring BJC to 100 NYC teachers by 2020!
 - We were just mentioned by White House!



bjc.link/bjcwwhitehouse

UC Berkeley "The Beauty and Joy of Computing" : Welcome, Abstraction (2)



Ball / McKinsey

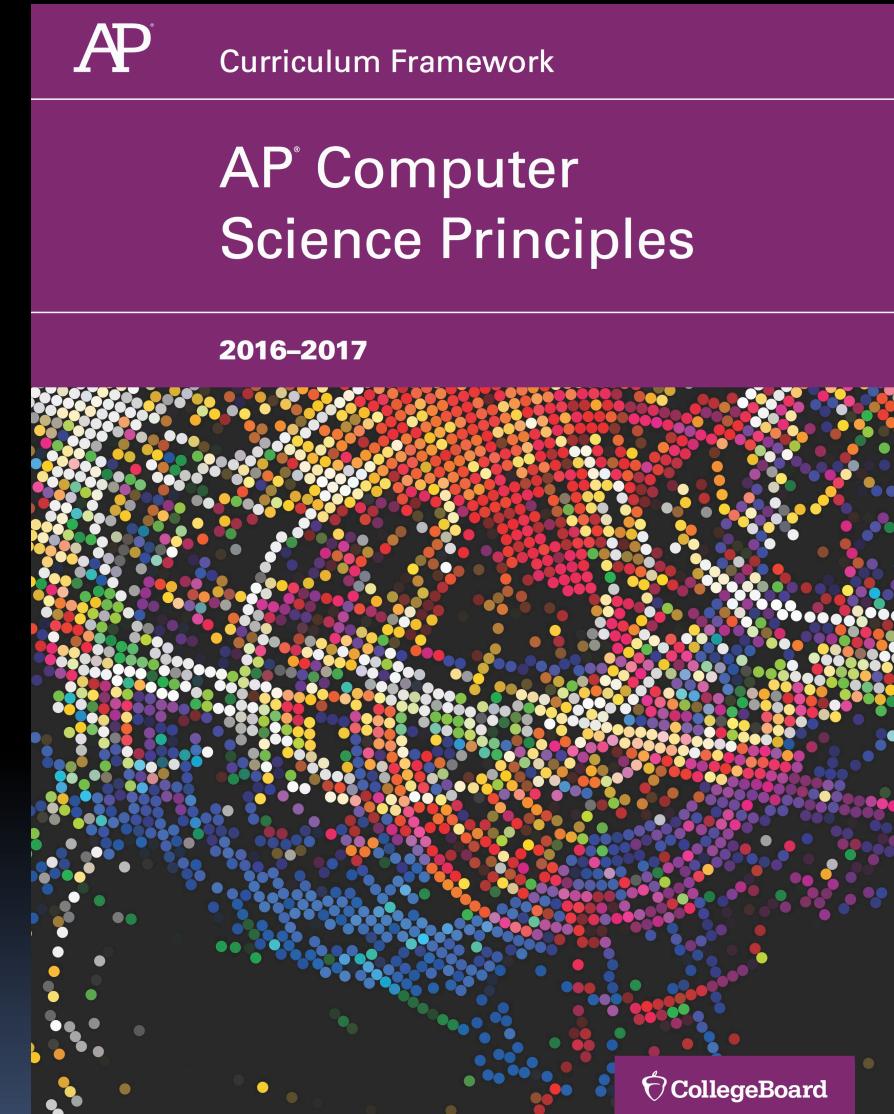


(AP) Computer Science Principles



7 Big Ideas

- Creativity
- Abstraction
- Data and Information
- Algorithms
- Programming
- The Internet
- Global Impact



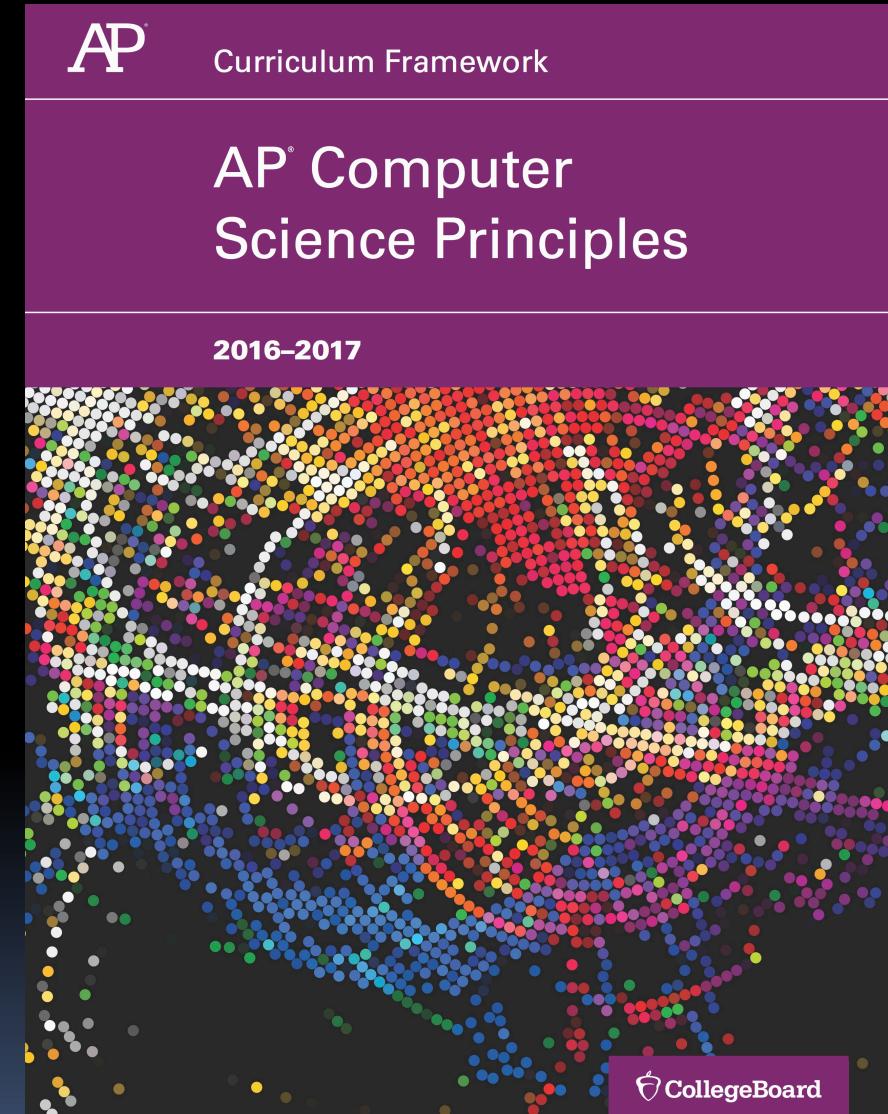


(AP) Computer Science Principles



6 Computational Thinking Practices

- **Connecting** computing
- **Creating** computing artifacts
- **Abstracting**
- **Analyzing** problems & artifacts
- **Communicating**
- **Collaborating**





Beauty and Joy of Computing in one slide

- **Big Ideas of Programming**
 - Abstraction
 - Algorithms (2)
 - Recursion (2)
 - Functions-as-data, I (2)
 - Programming Paradigms
 - Cloud Computing
- **Beauty and Joy**
 - “CS Unplugged” activities
 - If f2f, all lab work in pairs
 - “Create” performance task
 - Of your own choice!!
 - “Explore” performance task
 - Of your own choice!!
- **Big Ideas of Computing**
 - Daily “computing in the news”
 - How the Internet works
 - Research Summaries
 - AI
 - HCI
 - The Power of Data (big, small)
 - Social Implications of Computing
 - Saving the World w/Computing
 - Cloud Computing
 - Limits of Computing
 - Future of Computing
 - Robots...





Incredibly easy-to-learn coding in Snap!

The screenshot shows the 'Snap!' programming environment. On the left, the script palette lists various blocks: Motion, Control, Looks, Sound, Operators, Pen, and Variables. The main stage area contains a script for a 'Mandelbrot' sprite:

```
set pen size to 1
go to x: 0 y: 0
pen down
repeat (4)
  move (100) steps
  turn (90) degrees
end
pen up
set [iterations] to [Mandelbrot Iterations v]
set pen color to [iteration]
set pen shade to [if < (iterations) > 100 then (100) else (iteration)]
Draw newfilled Square Center
```

The 'Mandelbrot Iterations' block is expanded to show its internal logic:

```
warp
script variables [x] [y] [iteration] [max iteration]
set [x] to [0]
set [y] to [0]
set [iteration] to [0]
set [max iteration] to [100]
repeat until [not (< [x * x + y * y] < 4) and < (iteration) < (max iteration)]
  set [xtemp] to [x * x - y * y]
  set [y] to [2 * x * y + y0]
  set [x] to [xtemp]
  change [iteration] by [1]
end
report [iteration]
```

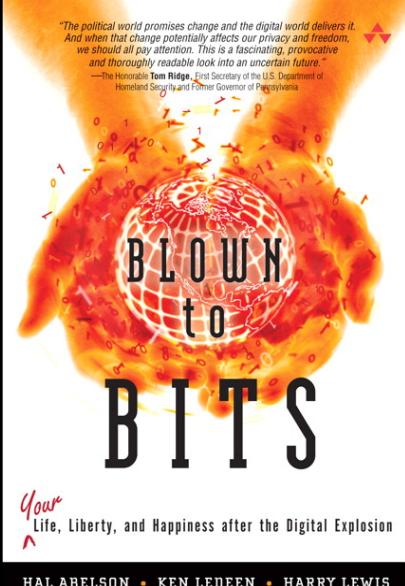
The stage on the right displays the resulting fractal image of the Mandelbrot set, rendered in purple and black.





Format & Textbooks

- **Format (14 hrs/wk * 7 wks)**
 - See the following slides
- **Selected Reading**
 - Taken from 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!**



contributed articles

[Data generated as a side effect of game play also solves computational problems and trains AI algorithms.](#)

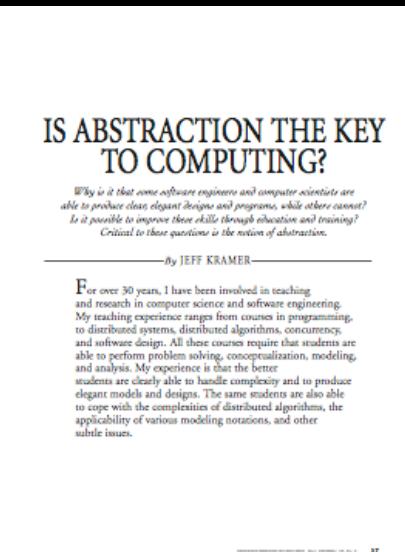
BY LUIS VON AHN AND LAURA DABISH

Designing Games With A Purpose

MANY TASKS ARE trivial for humans but continue to challenge even the most sophisticated computer programs. Traditional computational approaches to solving such problems focus on improving algorithmic algorithms. Here we advocate a different approach: the constructive channelling of human brainpower through computer games. Toward this goal, we present general design principles for the development and evolution of a class of games we call "games with a purpose," or GWPs, in which people, as a side effect of playing, perform tasks computers are unable to perform.

The Entertainment Software Association (www.theesa.com/facts/gamer_data.php) has reported that the average American spends more than 10 hours a day playing computer and video games in the U.S. Indeed, by age 21, the average American has spent more than 10,000 hours playing such games!—equivalent to five years of working a full-time job 40 hours per week.

58 COMMUNICATIONS OF THE ACM AUGUST 2010 VOL. 51 | NO. 8





Weekly Schedule (on cs10.org)

Jun 21 — 27, 2015

today < >

	Mon 6/22	Tue 6/23	Wed 6/24	Thu 6/25	Fri 6/26
9am	Lab 111 200 Sutardja Dai – Arany	Lab 111 200 Sutardja Dai – Arany	Lab 111 200 Sutardja Dai – Arany	Lab 112 200 Sutardja Dai – Erik	
10am					
11am		Discussion 112 3105 Etcheverry		Discussion 112 3105 Etcheverry	
12pm	Arany's OH Soda-Alcove-611 Cap:15		Arany's OH Soda-Alcove-611 Cap:15		
1pm	Lab 112 200 Sutardja Dai – Erik		Lab 112 200 Sutardja Dai – Erik	Erik's OH Soda-Alcove-651 Cap:10	
2pm		Erik's OH Soda-Alcove-651 Cap:10			
3pm		Discussion 111 320 Soda		Discussion 111 320 Soda	
4pm	Lecture 534 Davis	Lecture 534 Davis	Lecture 534 Davis	Lecture 534 Davis	
5pm	Jon's OH		Jon's OH		
6pm					
7pm					





Let's check enrollments...

- We have NEVER turned anyone away, and we don't intend to turn anyone away this semester!

Course: COMPUTER SCIENCE 10 P 001 LEC ([course website](#))
Course Title: The Beauty and Joy of Computing ([catalog description](#))
Location: MTuWTh 4-5P, 534 DAVIS
Instructor: MCKINSEY, J C
Status/Last Changed:
Course Control Number: 28605 [View Books](#)
Units/Credit: 4
Session Dates: 06/22-08/14/15
Summer Fees: UC Undergraduate \$1,624.00, UC Graduate \$2,040.00, Visiting \$1,840.00
Note: Also: BALL, M
Enrollment on 06/21/15: Limit:66 Enrolled:66 Waitlist:1 Avail Seats:0
[Click here for current enrollment information and course restrictions](#)



Peer Instruction for active learning!

- Increase real-time learning in lecture, test understanding of concepts vs. details
- As complete a “segment” ask multiple choice question
 - 1-2 minutes to decide yourself
 - 2 minutes in pairs/triples to reach consensus. Teach others!
 - 2 minute discussion of answers, questions, clarifications





Piazza: Q&A, Help, Announcements

The screenshot shows a web browser window for piazza.com/class/iag86w469hthu?cid=6. The title bar indicates the page is Su15 Calendar Plan - Google Sheets. The main content area shows a note titled "Welcome to CS10! :)" with 44 views. The note contains text about the welcome message, schedule, class materials, textbook availability, iClickers usage, and check requirements. On the left, there's a sidebar with pinned posts, a search bar, and a list of recent posts categorized by date (TODAY, LAST WEEK, WEEK 5/31 - 6/6).

Welcome to CS10! :)

Hi everyone!

Welcome to CS10 Summer 2015! If this is your first semester, then, welcome to Berkeley as well! We hope you've all had a great summer and we're excited to start class soon. We just wanted to update everyone on the schedule for the first couple weeks.

Our class website is available here: <http://cs10.org/>. We will have an updated semester schedule with exam dates and homework assignments soon!

Schedule:

- There will be lecture, lab, and discussion sections held on Monday, June 22.
- Lecture is held in 534 Davis Hall.
- Lab is in 200 Sutardja Dai Hall -- it is located to the right of the Qualcomm CyberCafe.
- Discussion is held in 3105 Etcheverry if you're in Section 112 or 320 Soda if you're in Section 111.

---Class Materials---

Textbook:
The textbook we use for CS10 is available for free. You're more than welcome to buy / rent a hard copy, but this isn't a necessity. Each chapter is linked from the class webpage.

iClickers:
CS10 uses iClickers to provide feedback during lecture and discussion. Answering questions will be worth a small amount of "sprinkle points" (extra credit) throughout the semester.

We will lend you an iClicker for the semester to use with CS10. To get an iClicker for the semester, please bring a check to your lab section for \$40. We will NOT cash any checks for iClickers if you return them back to us at the end of the semester. (Think of this as a form of insurance to guard against lost or broken iClickers.) Please only use personal checks. Please do NOT give us cashier's checks or money orders, cash, or any other collateral as we cannot accept these. Once you receive an iClicker, you will be able to register it for use in CS10 or another class. All iClickers must be returned at the end of the semester.

On the check you must:

- Clearly write YOUR name as it appears on your ID card. (The check does not need to be your own.)
- Clearly write YOUR student ID number on the check.
- Make the check payable to "UC Regents"





Pro-student Grading Policies

- **EPA**

- Rewards good behavior
- Effort
 - E.g., Office hours, doing 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



Ball / McKinsey

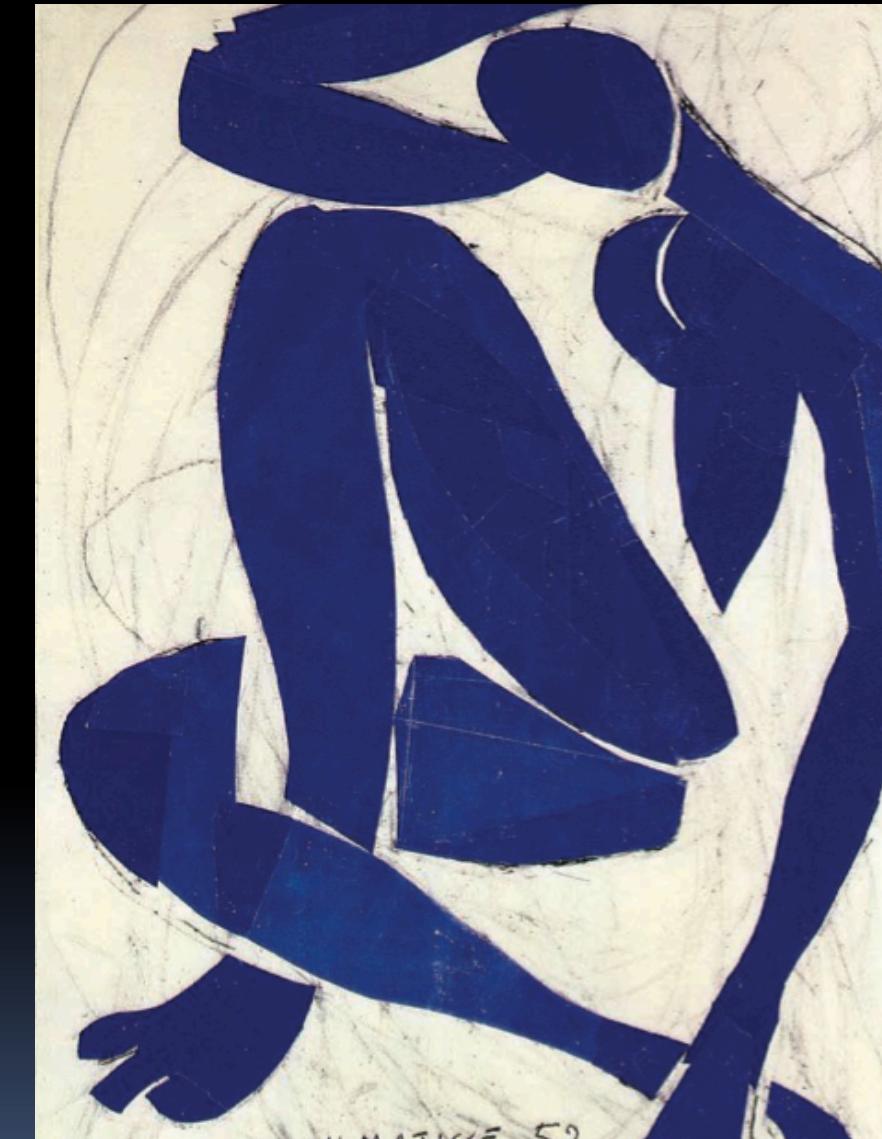


Abstraction & Detail Removal



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”

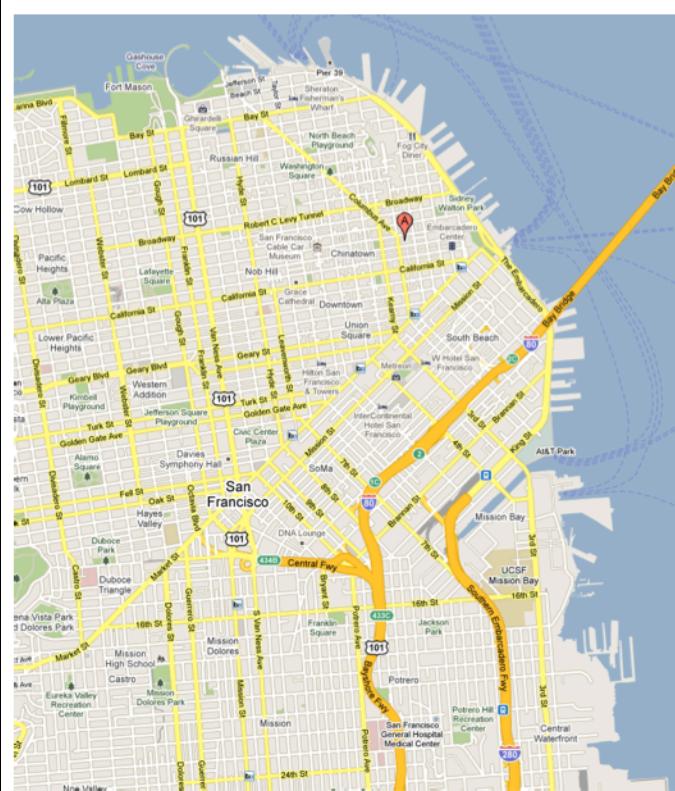


Naked Blue IV (Henri Matisse, 1952)

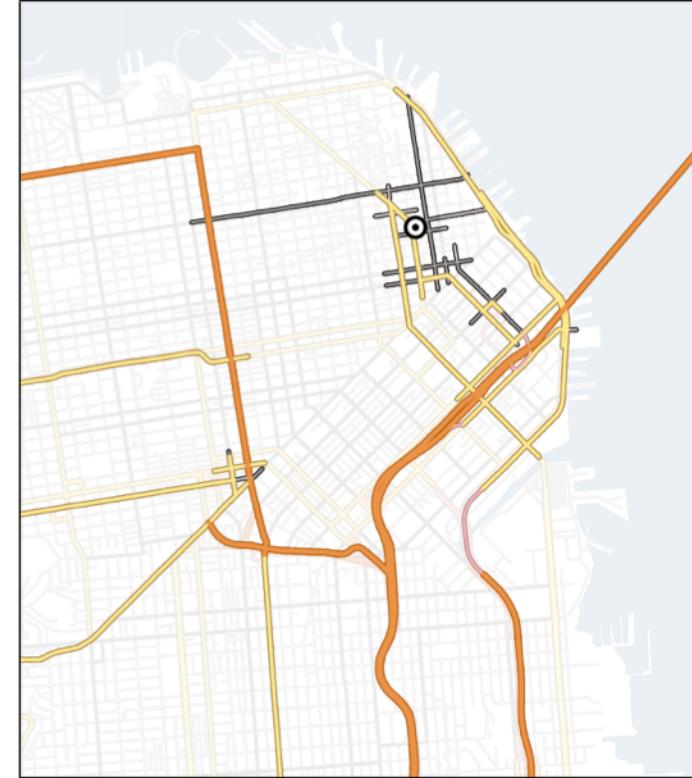




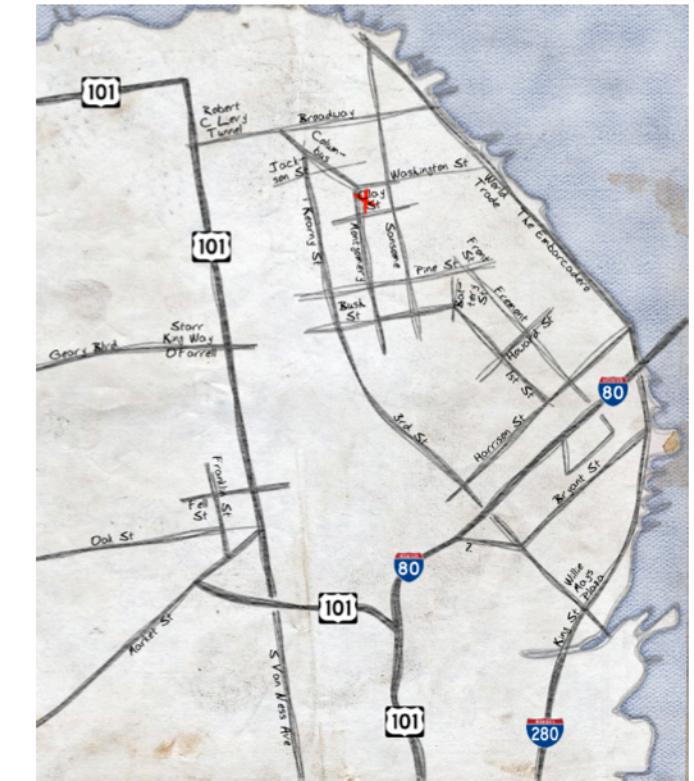
Detail Removal



General Purpose Online Map



Selected Roads



Our Result

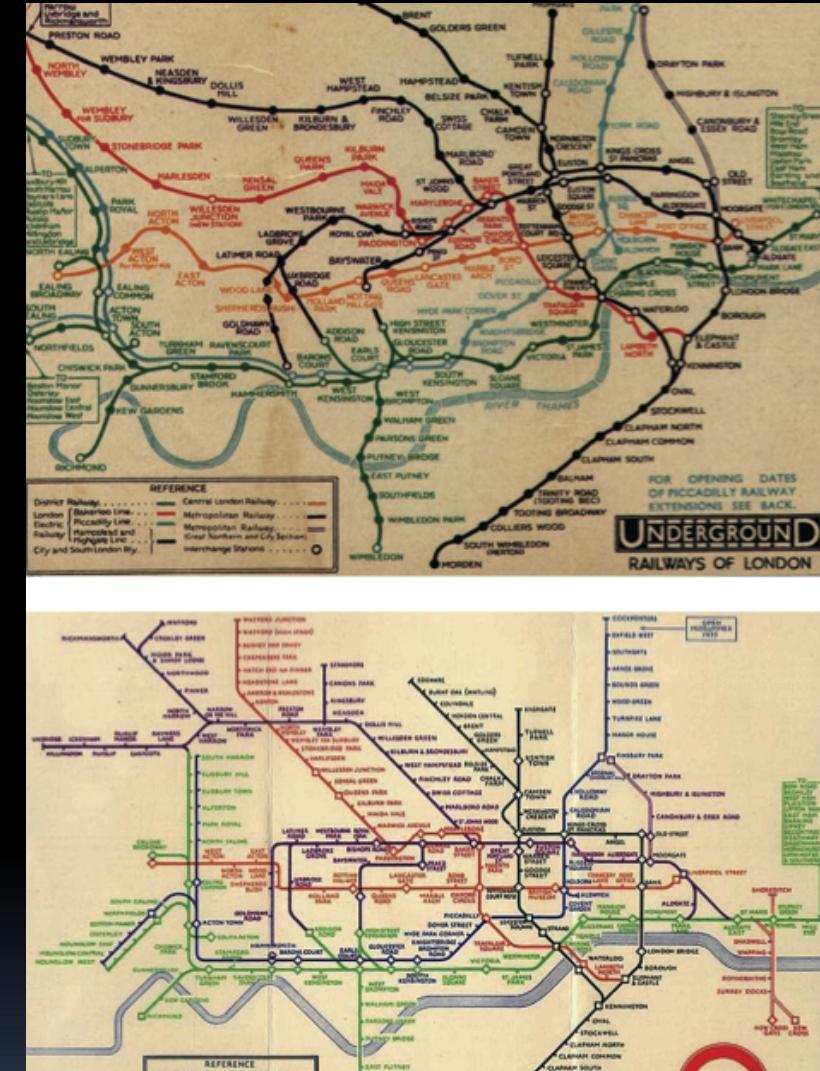
Automatic Generation of Detail Maps
(Maneesh Agrawala @ UC Berkeley, among others)





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 often brought in to know what to remove and what to keep!



The London Underground 1928 Map & Harry Beck 1933 map.





Generalization



Generalization Example

- You have a farm with many different animals.
- Different food for each
- You have directions:
 - 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

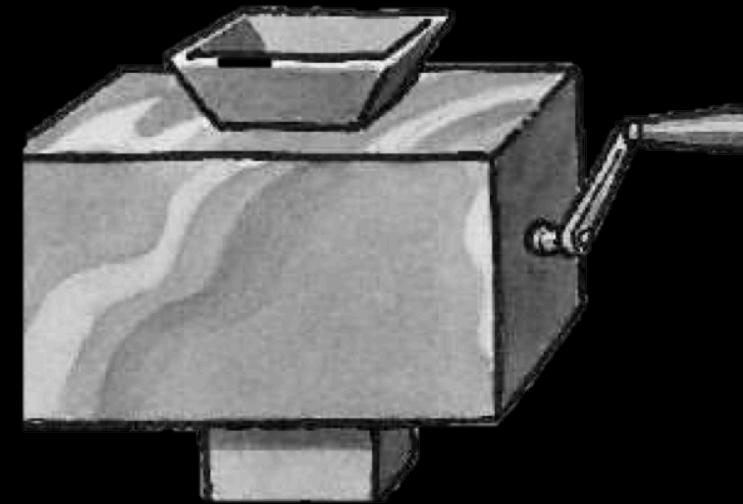


Generalization (in BJC) ... foreshadowing

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

$$y = \sin(x)$$

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



Function machine
(*Simply Scheme*, Harvey)





Abstraction Interfaces & Summary



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!
- It's how mankind has engineered some of the greatest structures and managed the complexity
- Two definitions
 - Detail Removal
 - Generalization



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

(right pedal faster, left pedal slow)

