Contexts of Computing Education: Graphics, Robots, Data, Information

Deepak Kumar
Department of Computer Science
Bryn Mawr College
Bryn Mawr, PA
Collaborations

Bryn Mawr College, MIT, Princeton, Howard, Purdue, Illinois, Texas A&M, UC-Berkeley, Stanford, UC-San Diego, Hawai‘i

Bryn Mawr College, Georgia Institute of Technology, Microsoft Research

Bryn Mawr College, Southern Methodist University, Sidwell Friends School, Arlington Independent School District

Thanks!
Enrollments in Computer Science
Enrollments in Computer Science
Enrollments in Computer Science

Entering College Students: Interest in CS

Year

Percent

2013: Year of Code.org

“In fifteen years we’ll be teaching programming just like reading and writing . . . and wondering why we didn’t do it sooner.”
— Mark Zuckerberg

10,000,000 have learned an HOUR of CODE
Anybody can learn.

Try an Hour of Code for Computer Science Education Week December 9–15. Anybody can learn!
http://code.org

Start
2014: Hour of Code
Enrollments in Computer Science

[Graph showing the percentage of entering college students interested in Computer Science from 1975 to 2010. The graph indicates a peak in interest around 1985, followed by a decline and a resurgence towards 2010.]
Gender Inequity

While, 57% of bachelor’s degrees are earned by women, just 12% of computer science degrees are awarded to women.

Source: Code.org
Key Issues

Diversity and broadening participation in computing curricula.
Much time and energy has been devoted to the design of introductory courses that serve as entry points into the programs.
Language Wars
Language Wars

C
Visual Basic
C++
Python
C#
Java
PHP
Perl
Objective C
JavaScript
Haskell
Ruby
Lisp
Scala
SML
OCAML
Prolog
Erlang
Ada
Pascal
FORTRAN
Cobol
Modula-2
PL/1
PL/1
ALGOL
Personal Robots

Mission
Explore making CS education more fun and effective through the context of a *personal robot*
Introductory computing courses serve as a gateway into the CS curriculum.

Should provide interesting and diverse range of examples and exercises.

Alignment of course content to student interests to increase engagement can have a positive impact on students choosing to enter computing as a major in college.

Most tasks should be attainable and provide a basis for supportive and positive feedback to students.
Personal Robots

CS ≠ Programming

Computing: *social activity*

Computing: *medium for creativity*

Performances vs. Competitions
Personal Robots

pic = takePicture()
show(pic)  # top picture

for pixel in getPixels(pic):
    r, g, b = getRGB(pixel)
    if r > 250 and b < 100 and g > 130:
        setColor(pixel, white)
    else:
        setColor(pixel, black)

show(pic)  # bottom picture
Personal Robots
Personal Robots

- Learned CS concepts through robots
- Robots made learning experience more hands-on, tangible, and exciting
- Most frustrating parts were dealing with robot hardware inconsistencies
- Viewed CS as a type of logic and problem solving; requiring patience & thought
- Discovered that CS and robots are applicable to the real world
More Dichotomies
Dichotomies
Mission
Explore making CS education more fun and effective through the context of *creative computing*
Processing is a programming language based on Java.
Since 2001, processing has promoted software literacy within the visual arts and visual literacy within technology.
Graphics

Processing:
A software sketchbook to teach computer programming fundamentals within a visual context.
```java
int x, y, w, h;

void setup()
{
  size(500, 500);
  smooth();
  noLoop();
  background(155);
  x = width/2;
  y = height/2;
  w = 40;
  h = 40;
}

void draw()
{
  noStroke();
  fill(0, 255, 0);
  translate(width/2, height/2);
  clover(8, 6, w, h);
  rotate(PI/2);
  clover(8, 6, w, h);
  rotate(PI/2);
  clover(8, 6, w, h);
  rotate(PI/2);
  clover(8, 6, w, h);
  // draw
}

void clover(int x, int y, int u, int h) {
  ...
Processing

Processing has evolved into a development tool for professionals artists.
There are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.
Ousterhout’s Dichotomy

Scripting Languages

Systems Programming Languages
Ousterhout’s Dichotomy

Scripting Languages
- glue languages
- emphasize reuse
- typeless
- Interpreted
- Slower
- Higher level

Systems Programming Languages
- Data Structures & Algorithms
- Strongly-typed
- Compiled
- Equiv. to Assembly Lang.
- “High-level”
- Standardized
Ousterhout’s Dichotomy

Scripting Languages

Systems Programming Languages
Ousterhout’s Dichotomy

Scripting Languages

Understand

Appreciate

Use

Computing

Systems Programming Languages
Ousterhout's Dichotomy

Scripting Languages
- Appreciate
- Understand
- Use

Computing

Systems Programming Languages
- Implement
- Design
- Algorithms
- Data Structures
Ousterhout’s Dichotomy

Scripting Languages
- Appreciate
- Understand
- Use

Programming Languages
- Systems
- Design
- Implement
- Data Structures
- Algorithms
- Computing

Computing
<Coding is Fun>
The Conceptual Legacy of CS1

Abstraction

Control Structures

Problem Solving

Program Design

The Durable Bones
By placing early learning of computing in more engaging and creative contexts, it’s possible to make the discipline and its practice more appealing and intellectually fascinating.
APIs are influencing a transformation in computing education.
Contexts

APIs are influencing a transformation in computing education.

APIs enable access to different contexts in computing.
Contexts

APIs are influencing a transformation in computing education.

APIs enable access to different contexts in computing.

Success relies on the ability to offer effective APIs.
Students should find our courses in resonance with the goals and aspirations that brought them to us in the first place.

Delete all indulgences to “create obstacles” to weed out students.

Context is important!

Be prepared to handle the capacity.

Do not succumb to the latest greatest technology fads.
Make sure it is inviting to all students.

Learn computing in engaging contexts

Design of Computing Curricula...

Flexibility in designing a major in computing (Threads at GeorgiaTech, Streams, ...)

Do not obsess over speed and efficiency in early curricula.
Computer Science: Curriculum (Re-Focus)
Elective Courses

- Artificial Intelligence
- Computational Linguistics
- Robotics
- Computer Graphics
- Computational Geometry
- Theory of Computation
- Compiler Construction
- Operating Systems
- Advanced Discrete Mathematics
- Geographical Information Systems
- Computational Biology
- Genomics
- Science of Information
- Emergent Computation
- Data Science
- Network Analysis
- Trees of Knowledge
- Mobile Dev. For Social Change
- Computing for Sustainability


data science
Soihub.org

Center for Science of Information

An NSF Science & Technology Center

Bryn Mawr College, MIT, Princeton, Howard, Purdue, Illinois, Texas A&M, UC-Berkeley, Stanford, UC-San Diego, Hawai‘i
Science of Information
Bringing Many Disciplines Together

Bryn Mawr College, MIT, Princeton, Howard, Purdue, Illinois, Texas A&M, UC-Berkeley, Stanford, UC-San Diego, Hawai‘i
Deliberate Practice

Program

Make it fun

Read Programs

10,000 Hours!

Work in Teams

Learn half dozen programming languages!

Encourage Playfulness

Practice Makes Perfect
I learned it last night! Everything is so simple!
Hello world is just print "Hello, world!"

I dunno...
Dynamic typing?
Whitespace?
Come join us! Programming is fun again!
It's a whole new world up here!
But how are you flying?

I just typed import antigravity
That's it?
... I also sampled everything in the medicine cabinet for comparison.
But I think this is the python.
Thank you!