The Case for Computing: Programming Across the Curriculum

Gary S. Stager, Ph.D.
gary@stager.org
cmkfutures.com

@garystager
Twitter

Resources

inventtolearn.com/stlinatl

cmkfutures.com

@garystager
Twitter

I seek to democratize computer programming

Reasons to Learn to Program

- Make things
  - Make things work
- Express yourself
- Develop habits of mind
- Solve problems
- Concretize abstractions
- Contextualize mathematics
- Mirrors the writing process and various design cycles
- You can do it by yourself or with others
- "Hard fun"
- Jobs / careers

Most “technology” in schools compares badly to clay or paint

Programming does not

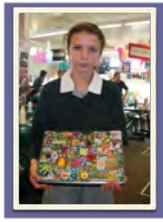
Middle School Me

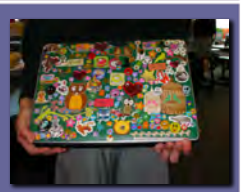
Me and Mr. Jones

& Mr. Gates, Mr. Paul Allen, and Mr. Steve Wozniak…

Equity

Personal Computing (1990)

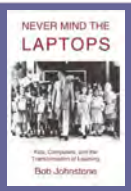




Early 1990s

Early 1990s

Never Mind the Laptops



http://stager.org/laptops

Lessons from Summer Camp

and the early days of 1:1 computing

- Boys and girls enjoy computer programming and will choose to engage in it under the right non-coercive conditions, especially when computers are abundant and personal
- Computer programming is intellectually and creatively rewarding
- When you don't know what's impossible, anything is possible
- Programming is a vehicle for matching a child's potential capacity for intensity
- Debugging is an essential skill
- Programming may be social and collaborative, but fundamentally it is a conversation between you and the computer. It concretizes abstractions.
- The computer is material with which you can make things, ideas, and control the world
- Every kid and their teachers could learn to program across disciplines
- Quality work takes time
- Programming justifies having computers in schools

Under what conditions could recreational computing return?

I thought that maybe the maker movement was the key to this renaissance…

We can make things with atoms, but we left the bits behind again

"There is no expedient to which man will not resort to evade the real labor of thinking."

— Sir Joshua Reynolds
(18th Century)

"We have spent an enormous amount of time and energy justifying not teaching kids to program"

Sound familiar?

The phrase, "technology and education" usually means inventing new gadgets to teach the same old stuff in a thinly disguised version of the same old way. Moreover, if the gadgets are computers, the same old teaching becomes incredibly more expensive and biased towards its dumbest parts, namely the kind of rote learning in which measurable results can be obtained by treating the children like pigeons in a skinner box.

"Teaching children Thinking"

Seymour Papert, 1971

The literal problem with CS4All

We need to be able to hold two thoughts simultaneously…

- Every kid needs to program across the curriculum
  - Bonus - computation is always involved
- Some children will choose computer science as their project and need formal course offerings available to them

If our goals are as modest as increasing achievement in the existing math curriculum, we would teach every child to program computers.

Absolute Value & Angle posters

Computational thinking without computers is just math

There is no computer science without computers & computing

"The activities introduce students to Computational Thinking through concepts such as binary numbers, algorithms and data compression, separated from the distractions and technical details of having to use computers. Importantly, no programming is required to engage with these ideas!" ("CS Unplugged," 2016)

Four Ways "CS Unplugged" Gets Learning Wrong

"The activities introduce students to Computational Thinking through concepts such as binary numbers, algorithms and data compression, separated from the distractions and technical details of having to use computers. Importantly, no programming is required to engage with these ideas!" ("CS Unplugged," 2016)

1. The fallacy that you cannot enjoy dessert without eating your vegetables first.
2. The debatable idea that binary numbers, algorithms, or data compression are appropriate or important curricular topics for young people, especially if there will no actual coding involved.
3. There is an implicit suggestion that these topics are easier to learn without programming or are even relevant in the absence of coding.
4. When "distractions and technical details" are used as an excuse for coding without computers, the writer presents an incorrect notion of cognitive development. Introducing these topics without the context of computing makes them more abstract, not less.

Gary S. Stager, Ph.D. constructingmodernknowledge.com

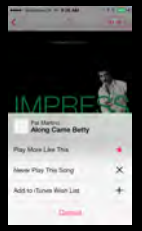
Computational Thinking

For when modeling is possible, but the programming is too difficult

Stephen Wolfram

Howard Gardner

Computational Thinking



“We do a little Scratch”

Fluency is the Goal

4 Forms of Programming

1. Make things (a game, simulation, thing, program)
2. Contextualizing mathematics
3. Instrumental purposes
4. Physical computing (programming the world)

Two Types of Languages

- Programming languages
- Programming languages for learning

Vocational education vs. The liberal arts

Teacher confusion
False complexity

DEBUGGING

From Mindstorms…

"Many children are held back in their learning because they have a model of learning in which you have either 'got it' or 'got it wrong.' But when you program a computer you almost never get it right the first time. Learning to be a master programmer is learning to become highly skilled at isolating and correcting bugs … The question to ask about the program is not whether it is right or wrong, but if it is fixable. If this way of looking at intellectual products were generalized to how the larger culture thinks about knowledge and its acquisition we might all be less intimidated by our fears of 'being wrong.'"

Papert, Seymour. (1980). Mindstorms: Children, Computers, And Powerful Ideas NY: Basic Books.

Modern knowledge construction is inseparable from computing

The standards are at best premature

Premature Irrational Assessment Disorder

Slow Down!

What's your hurry?

Fluency is the Goal

Less is more!

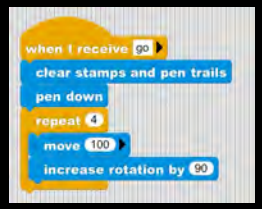
What is the smallest seed I can plant?

Generative vs. iterative processes

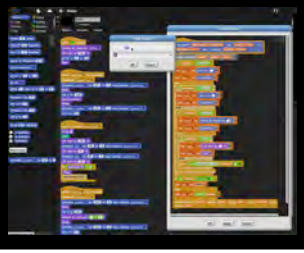
Elegance emerges from experience

“Block” Programming

Any Questions?



Any Questions?

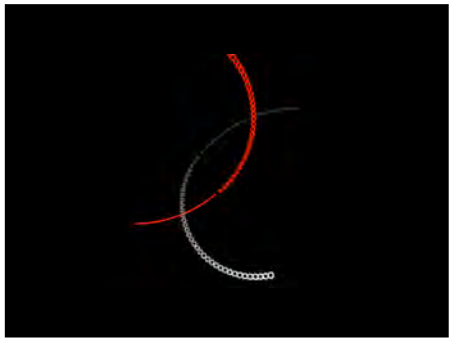


Research

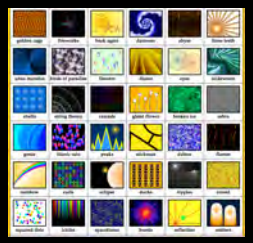
Building on these successes, national curricular efforts in the United States are starting to incorporate block-based programming into instructional materials alongside, or in place of, conventional text-based programming. To understand if this decision is helping learners from historically underrepresented populations succeed in computing classes, this paper presents an analysis of over 5,000 students answering questions presented in both block-based and text-based modalities. A comparative analysis shows that while all students perform better when questions are presented in the block-based form, female students and students from historically underrepresented minorities saw the largest improvements. This finding suggests the choice of representation can positively affect groups historically marginalized in computing.

“Blocks or Text? How Programming Language Modality Makes a Difference in Assessing Underrepresented Populations” by David Weintrop, Heather Killen, and Baker Franke (2018)

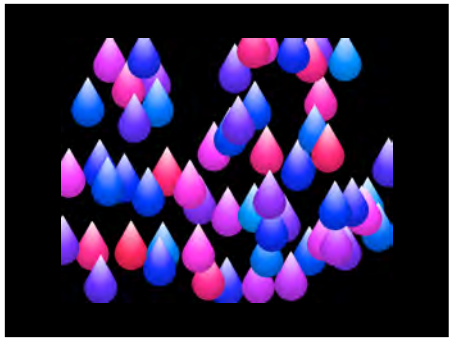
Programming is a way of seeing

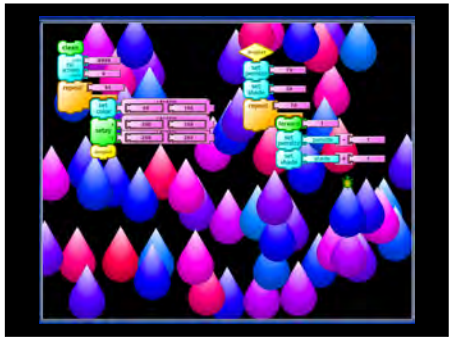


Turtle Art



turtleart.org





You can find inspiration everywhere

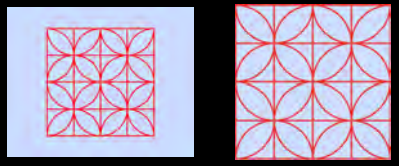
Islamic Tiling

Art, Math, History, Language Arts, Computer Science?

Digital to Analog Mathematics to Art

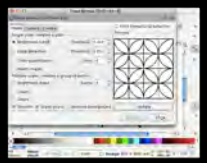
Programming in Turtle Art

Digital to Analog Mathematics to Art



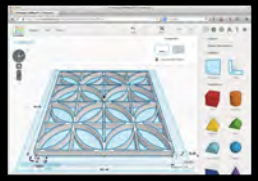
Crop Turtle Art image in Preview

Digital to Analog Mathematics to Art

Convert PNG to SVG files

Digital to Analog Mathematics to Art



Extrude in Tinkercad

Digital to Analog Mathematics to Art


3D print stamps

Digital to Analog Mathematics to Art

Digital to Analog Mathematics to Art

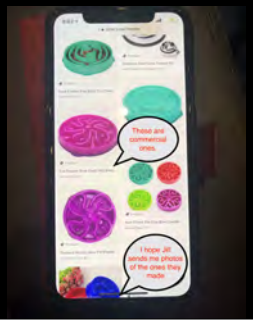
Digital to Analog Mathematics to Art

Digital to Analog Mathematics to Art

Hand paint

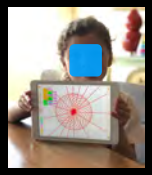
Digital to Analog Mathematics to Art

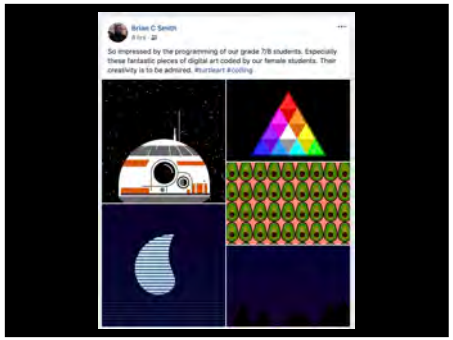
Digital to Analog Mathematics to Art

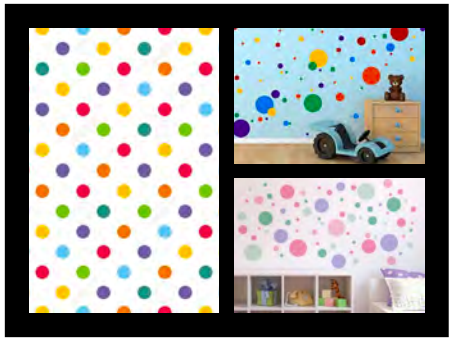


Slow-feed Dog Bowls

We are working on our dog bowl designs again! My 11 and 8 year olds had the best conversation. My 8 year old could not figure out the radius and my 11 year old drew it out for her and explained arc. I listened from the other room with sheer joy. I don't post pictures of my kids on twitter but thought I'd share this with you both. I'm also helping my local Mathnasium plan their summer camps. You both inspired me to change the camp sequence. We added a week that focuses on Turtle Art, Tinkercad, and I convinced them to buy a 3-D printer. The owner of Mathnasium just texted me to let me know her tutors stayed an extra hour to play with Turtle Art. I will have them share their designs with you this summer. Thank you!





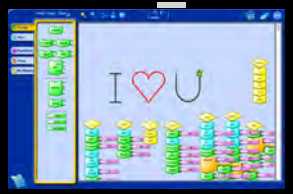


"To the extent that Logo "failed," it failed because the teachers never understood the mathematics the children were doing."
(Alan Kay)

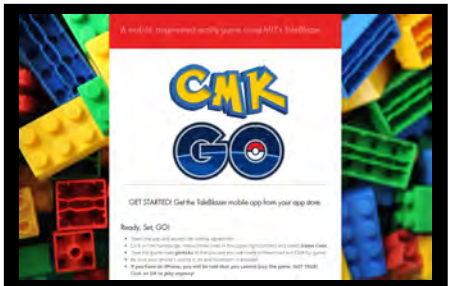
Love is a better master than

Nothing beautiful is forced

Love is a better master than



Nothing beautiful is forced



constructingmodernknowledge.com

Programming the World

In our image of a school computation laboratory, an important role is played by numerous "controller ports" which allow any student to plug any device into the computer… The laboratory will have a supply of motors, solenoids, relays, sense devices of various kids, etc. Using them, the students will be able to invent and build an endless variety of cybernetic systems.

(Papert & Solomon, *Twenty Things to Do with a Computer.* 1971)

Beauty and Joy of Computing

http://bjc.berkeley.edu

bjc

YouTube "Introduction to Wolfram Language"

inventtolearn.com/wolfram

Future Ready?

The Disappointing State of Computers in Education

"Why then should computers in schools be confined to computing the sums of the first twenty-odd numbers and similar so-called problem solving uses? Why not use them to produce some action? There is no better reason than the intellectual timidity of the computers-in-education community, which seems remarkably reluctant to use computers for any purpose that fails to look like something that has been taught in schools for the past centuries. "

- Papert & Solomon, 1971

Back to School?

- The near future is uncertain
- Resources are scarce
- The best makerspace is between your ears! Programming is the perfect COVID-19 makerspace.
- Kids have nothing but time
- Computer programming is the answer!

Resources related to this presentation may be found at

inventtolearn.com/STLinATL

Gary S. Stager, Ph.D. – cmkfutures.com/gary – gary@stager.org – @garystager