The *Communications* Web site, <http://cacm.acm.org>, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

Follow us on Twitter at <http://twitter.com/blogCACM>

DOI:10.1145/3374764

<http://cacm.acm.org/blogs/blog-cacm>

Sizing the U.S. Student Cohort for Computer Science

Mark Guzdial considers indicators that U.S. K-12 students will study computer science.

Mark Guzdial
The Size of Computing Education Today, By The Numbers

<http://bit.ly/2sxWlBq>

November 3, 2019

Alan Kay, Cathie Norris, Elliot Soloway, and I had an article in the September 2019 issue of *Communications* called “Computational Thinking Should Just Be Good Thinking” (access the article at <http://bit.ly/2P7RYEV>). Our argument is that “computational thinking” is already here — students use computing every day, and that computing is undoubtedly influencing their thinking. What we really care about is effective, critical, “expanded” thinking, where computing helps us think. To do that, we need better computing.

Ken Kahn engaged with our article in the comments section (thank you, Ken!), and he made a provocative comment:

There have been many successful attempts to add program-

ming to games: Rocky’s Boots (1982), Robot Odyssey (1984), RoboSport (1991), Minecraft (multiple extensions), and probably many more. But these efforts excite a small fraction relative to those who are excited about using general-purpose programming systems such as Logo, Scratch, Squeak, ToonTalk, or Snap! for their own projects.

That got me wondering: Is general-purpose programming more popular than specialized programming? That’s a hard question to answer, but there are other related questions that are answerable. What fraction of people use Scratch and other general-purpose programming systems, versus something like Minecraft? What fraction of students learn any kind of programming at all? How many students learn general-purpose programming today, compared to using other computing environments or learning other science, technology, engineering and mathematics (STEM) subjects?

Here is what I found. I decided to be U.S.-centric, so that I could compare to known overall populations.

Overall Populations

Number of people in the U.S. (World Population Review, <http://bit.ly/2Pg79Mu>): 329 million.

Number of school children in the U.S. (National Center for Educational Statistics, <http://bit.ly/2OKFhAK>): 56.6 million

► About 50.8 million in public schools, about 5.8 million in private schools. Only about 1.5 million children are homeschooled.

► 35.5 million are in pre-K to grade 8, 15.3 million in high school.

Learners of Computing vs. Users of Computing Tools

Number of Scratch users in U.S. (Scratch community statistics at a glance, <http://bit.ly/37YWwCl>): 19.2 million.

We don’t know ages (and we know that we can’t trust the age data in Scratch; see “Youth Computational Participa-

tion in the Wild: Understanding Experience and Equity in Participating and Programming in the Online Scratch Community,” <http://bit.ly/2ONsq0E>). Roughly one-third of all U.S. schoolchildren have tried Scratch. While Logo, Snap!, Squeak, and other tools are popular, they are nowhere near as popular as Scratch. Scratch is likely our best possible scenario.

Number of Code.org students in U.S. (2018, Computing Education Research Blog, <http://bit.ly/2Djiz5W>): 1.2 million.

Number of Minecraft users (from “‘Minecraft’ has been quietly dominating for over 10 years, and now has 112 million players every month,” *Business Insider*, September 14, 2019, <http://bit.ly/2OISLNv>): 112 million.

Obviously, that’s more than the U.S., and not just school age, since that’s even school children or a third of everyone in the U.S. I was unable to find a number just in the U.S.

I found this estimate for the number of children playing Minecraft (in “Minecraft teaches kids about tech, but there’s a gender imbalance at play,” *The Conversation*, January 16, 2018, <http://bit.ly/2OISLNv>):

The results show that 53% of children aged 6 to 8, and 68% of children aged 9 to 12, are actively playing Minecraft. More than half of those play more than once per week.

Number of Fortnite users (from Fall Skirmish Fortnite Series, Fall Skirmish Details, September 20, 2018): 78.3 million.

Again, more than in the U.S., and not limited to school children.

The best estimate I could find is that 61% of U.S. teens play Fortnite (from “Exclusive: ‘Fortnite’ survey shows kids are playing in class. So what can parents do?” *USA Today*, December 6, 2018, <http://bit.ly/363N02V>), so perhaps 9.3 million.

Bottom line: There is no computing education tool that comes anywhere close to the number of children who play Minecraft or Fortnite.

Learners of Computing vs. Other Learners of STEM

One way of getting a sense of access to computing vs. other subjects in the U.S. is to look at access to Advanced Placement, which is a program for accessing advanced subjects (for college credit or placement) in high school. It is generally considered elitist: not every student gets access to it (though there is some argument that it’s no longer elitist.) But if we’re going to compare between subjects, it’s elitist to all subjects equally.

Number of students who took AP Calculus AB (2018, <http://bit.ly/2sHB1cU>): 308,537

Number of students who took AP Calculus BC (2018, <http://bit.ly/2sHB1cU>): 139,376
Total: 347,913

Compared to the 15.3-million high schools students in the U.S., this is about 2%. It’s an upper bound on the number of students that one might get in an AP STEM class.

Number of students who took AP Physics 1 (2018, <http://bit.ly/2LhaXMg>): 170,653

Number of students who took AP Physics 2 (2018, <http://bit.ly/2LgcvpC>): 24,985

Total: 195,638

Number of students who took AP Computer Science A (2018, <http://bit.ly/360EMbS>): 65,133

Number of students who took AP Computer Science Principles (2018, <http://bit.ly/2ONsq0E>): 72,187

So, combined, AP exams in computer science have about 137,320 exam takers, which is about 70% of AP Physics. That’s about 40% of AP Calculus, or about 40% of 2% of all U.S. high school students.

The Bottom Line: Computing education is not even close to the mainstream of STEM.

Conclusion: We should try *everything*: In general, we have not yet created popular computing education. We reach very few students.

To respond to Ken, we only have reached a small fraction of students with *all* our tools combined. We should be programming extensions to games *and* general-purpose programming languages *and* everything else we can think of that might help kids to gain access to computing education.

There’s so little reaching students now that it doesn’t make sense to build only on what’s been most successful in the past. All past successes may be local maxima. We may have to try something radically different to reach many students.

Comments

Interesting numbers. It would be great to get a worldwide census. I think the numbers are better in the U.K. for 5-to-16-year-olds, and not as good for 16 to-18-year-olds. I’ve heard about countries with much better numbers (Costa Rica, Lithuania, ??).

Twelve years ago, I wrote “Should Logo Keep Going Forward 1?” (<http://bit.ly/2Lz2gx1>) for EuroLogo 2007. I tried to argue against incremental progress; instead, we should move towards “something radically different” as you suggest. I came up with a list of 12 ideas that looking at it now should be a much longer list. One thing I would add today is artificial intelligence playing multiple roles including being a guide/coach and providing new kinds of computational building blocks for students to use in their projects.

One topic that Seymour Papert brought up when I spoke to him about this just before his accident is, whose aesthetics should count: mathematicians or engineers? A mathematically aesthetic design would be based upon a very small set of very expressive building blocks. Nearly everything we build for children has been engineered to have a large number of nice features, instead of a beautiful kernel.

Ken Kahn

November 4, 2019

Mark Guzdial is a professor in the Computer Science & Engineering Division of the University of Michigan, USA.

© 2020 ACM 0001-0782/20/2 \$15.00