SCRATCH THAT

Mitchel Resnick '78 thinks coding is child's play

By David Walter '11

When is a toy more than a toy? When it's 2017, and you've got half an hour on your lunch break to run to the store and buy a birthday present for your niece Emma ... and Emma, meanwhile, has only 10 short years to ready her application to Princeton ... and Emma's parents have hinted that she can use all the help she can get. (Her best friend at Sunday school already speaks three languages, while Emma occasionally sings along to the Spanish parts of "Despacito.") It's not enough, in other words, that the toys you buy avoid polluting Emma's imagination with lazy stereotypes and corporate branding. You're looking for a toy that makes your niece smarter, kinder, and — ideally — proficient in Mandarin. A toy that gives her that ineffable EDGE ... but in a fun way! You're looking, you think — because it's lunchtime, and you're hungry — for the BEETS of children's toys: a toy that's sweet like candy, nourishing like a vegetable, and largely guilt-free, give or take a few clothing stains.

Mitchel Resnick '78 can help. Resnick is MIT's Lego Papert Professor of Learning Research — yes, Lego as in the toy blocks — and directs that university's Lifelong Kindergarten research group, which, according to its website, seeks to evoke "the spirit of the blocks and finger paint of kindergarten." Resnick has spent the last three decades working with Lego to come up with toys that support creative learning, most notably helping to develop the company's Mindstorms range of robotic building blocks. Ten years ago, he and his team invented a computer coding platform called Scratch, which has transcended mere toydom and has become an entirely new language for thinking, playing, and creating for millions of kids worldwide. That you probably haven't heard of Scratch (or Resnick) is in a way a testament to Scratch's (and Resnick's) humble, resolutely kid-centric ethos. Scratch has never advertised itself commercially, relying instead on word-of-mouth from parents, teachers, and above all, enthusiastic youngsters. Resnick is like the kindly uncle — in temperament, more Mister Rogers than Willy Wonka — working in the background so kids can have fun.

Last year, nearly 200 million people used Scratch — a simplified, visually based coding language — to create their own video games, serialized television shows, and dancing cat cartoons. About 20 million were active participants in Scratch's online community of makers. Forty-five percent of the users are girls. For kids, the appeal of Scratch is simple: It lets them invent their own stories, games, and animations and share them with an audience of their peers. But Scratch is also a bona fide programming language — one that's as theoretically consistent with the principles of computer science as heavy-hitters like Java or C++. It's fun, it's free (you can try it out at scratch.mit.edu), and it's globally minded: Its user interface has been translated into 50 languages. As of August 2017, Scratch was ranked as the 19th most popular coding language in the world — and the only one with a core user base of 8- to 16-year-old kids.

Ten years after its inception, Scratch accounts for 25 percent of MIT's web traffic and employs a small army of moderators, engineers, and researchers. And that's just the beginning: The upcoming release of Scratch 3.0, combined with a push by Scratch's philanthropic arm to promote the toy in the developing world, could send those numbers into the stratosphere.

But more on all that in a minute! First, let's get back to your toy-selection dilemma.

From his primary-colored, toy-strewn lab at MIT — picture pipe cleaners and Play-Doh, not protein sequencers or particle colliders — Resnick offers some guidelines for toy purchasing:

The first question adults should ask, he says, is: "What kind of thinking does a toy promote?" The hierarchy of thinking you might picture — mindless fun on the bottom of the pyramid, "critical thinking" at the top — doesn't jibe with current research on the science of play. Free play that engages the imagination is often more valuable than "educational" games that walk kids through a set of pre-planned puzzles.

Illustration: courtesy Mitchel Resnick '78; photo: Brian Smith

Scratch inventor Mitchel Resnick '78 with the program's mascot in his lab, part of MIT's Media Lab.

Resnick compares many "educational" toys to the outdated learning methods he was exposed to back in the 1970s. In Resnick's grade-school days, for instance, writing lessons focused heavily on grammar — diagramming parts of speech, fixing ungrammatical sentences, and so on, to the exclusion of much else. "That type of analytical thinking is valuable to learn, but if that's all we teach kids, there's a limit to how they can express themselves with language," he says. "If all that kids look at is what words they spelled wrong and how they got their grammar wrong, they're not going to be excited about writing."

To Resnick's mind, toys and games that focus on problem-solving to the exclusion of creative thinking commit the same sin. "There's nothing wrong with systematic thinking, but we shouldn't stop there," he says. Instead, he tells me, adults should "look for more activities that allow kids to exert more control over the process, that allow them to create and share their own ideas with others, as opposed to just solving someone else's problem." For Resnick, the ideal scenario is one that places analytical thinking in the service of creative design. Picture a kid who in the process of building a Lego castle can draw on skills like geometry and multiplication.

The second question adults should ask when buying educational toys is: What kind of learner is my child? Again, the categories you might expect — visual learner, kinetic learner, etc. — aren't the ones that Resnick favors. Instead, he uses terms like tinkerers, planners, patterners, and dramatists. The latter two terms were coined by education researchers Dennie Wolf and Howard Gardner in the 1970s; Resnick features the categories in *Lifelong Kindergarten,* his recently released book on creative play.

"Patterners," Resnick writes, "are fascinated by structures and patterns, and they typically enjoy playing with blocks and puzzles. Dramatists are more interested in stories and social interaction, and they often play with dolls and stuffed animals."

A single set of toys can be designed to serve both groups, Resnick says, but play scenarios should be framed differently depending on the intended audience.

In his book, Resnick recalls a robotics workshop he ran a few years back in which children were divided into two groups: one that happened to include mostly patterners; the other, mostly dramatists. The goal of the workshop was to build an amusement-park ride out of building blocks, motorized parts, and robotics software.

The group made up of patterners immediately began working on a merry-go-round: "They carefully drew up plans, then used Lego bricks, beams, and gears to build the structure and mechanisms. After they finished building the merry-go-round, they wrote a computer program to make it spin around, then added a touch sensor to control it. ... The whole project, from initial idea to final implementation, took just a couple of hours."

Spending hours in front of a computer is fine, Resnick says, as long as the child is an active builder of what's happening on screen, rather than a passive consumer.

The dramatists, meanwhile, decided to build a Ferris wheel. Resnick recalls that "after working for 30 minutes on the basic structure for the Ferris wheel, they put it aside and started building a refreshment stand next to the Ferris wheel. At first I was concerned. Part of the purpose of the activity was for students to learn about gearing mechanisms and computer programming. After finishing the refreshment stand, the students built a wall around the entire amusement park. Then, they created a parking lot, and added lots of miniature Lego people walking into the park. They developed an elaborate story about several families coming from different parts of the city to spend a day at the amusement park. Only then, after the whole amusement-park scene was complete, did the students go back and finish building and programming their Ferris wheel. To them, building the Ferris wheel wasn't interesting until they had imagined a story around it." It took the dramatists several hours longer to finish the exercise, but the result was no less technically impressive.

To Resnick, the lessons of workshops like these were profound — and troubling. "What if the amusement-park workshop had ended after an hour?" he wondered. It would have seemed that the patterners had an aptitude for robotics and programming, while the dramatists were doomed to lag behind. In reality, though, both groups of kids were capable of creating a sophisticated design.

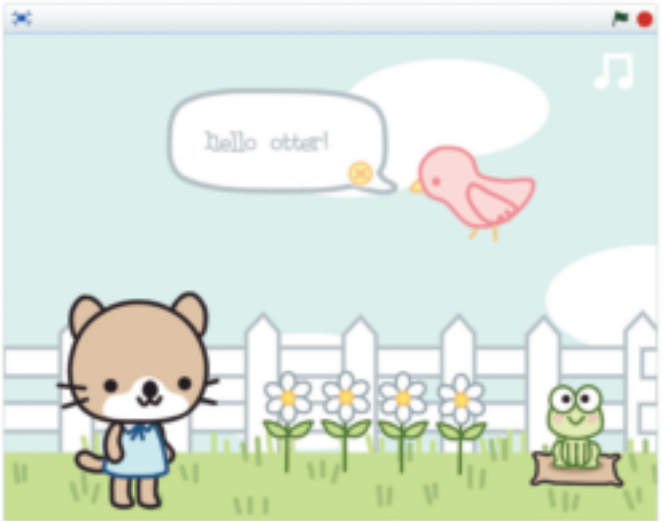
These divisions in play style have implications beyond the toy chest. "Math and science courses, from elementary school through college, have traditionally been designed in ways that favor patterners over dramatists," Resnick writes. "That's a big reason why many kids get turned off by math and science."

Resnick developed Scratch to provide children with an alternate path to creative learning. For a computer scientist — and MIT professor — Resnick is surprisingly platform-agnostic: He'd be equally happy if kids learned to create by writing, painting, or building in the real world. That said, he does not fear the perils of too much "screen time." Spending hours in front of a computer is fine, he says, as long as the child is an active builder of what's happening on screen, rather than a passive consumer. Do we worry about "page time" if a kid spends hours after school writing a story? he asks. No — we're just happy that the child is flexing those creative muscles.

Computer coding happens to be the medium Resnick chose for Scratch, but not because it's "better" than more traditional creative pursuits. For one thing, it's easier — and cheaper — to distribute a program like Scratch across the internet than it would be to, say, send painting supplies to tens of millions of kids across the world. (And unlike paint, Scratch enables kids to create dynamic, interactive projects.) What's more, Scratch makes it easier for users to rapidly make and undo mistakes in pursuit of perfection — the sort of "tinkering" mindset Resnick believes is key to design-based learning.

Ultimately, Resnick's goal with Scratch isn't to create a new generation of super-coders. Instead, Scratch is geared simply toward getting kids excited about creative self-expression. "So much of the education system is top-down. We need to give kids greater leeway to find their own path."

Scratch offers its coders near-total freedom to be as serious or silly as they want to be.



Courtesy Mitchel Resnick '78

Screenshots from Scratch games, from top: "Lemonade Time" by ipzy; an animated holiday card called "Holiday Card Eek!"; a collaborative animation called "I'd like to teach the world to code."

Resnick's own path through grade school and adolescence was more or less a conventional one. He grew up in the Philadelphia suburbs, the middle child of three in a family where "being able to succeed and achieve was seen as important." He always took school seriously. Maybe too seriously, sometimes: Resnick's teachers would observe that he was "too tense" when working on school projects. "I was someone who was following the rules and wanted to succeed within the system. I was good at the things schools value: certain types of intelligence, certain types of achievement," he recalls. At Princeton, he majored in physics and wrote for *The Daily Princetonian*.

Along the way, however, Resnick realized that the rules he'd played by didn't work for every (or even most) children. So he decided to work to change those rules. Resnick, who has no children of his own, had always been drawn to helping young people: from tutoring classmates in high school, to coaching a youth basketball team in Princeton, to working as a counselor at a summer camp. After a postgrad stint as a technology reporter at *BusinessWeek,* he began to chase a career at the crossroads of education and computer science, earning a master's degree and then a Ph.D. in the latter discipline at MIT.

At MIT, Resnick found a mentor in Seymour Papert, a legendary professor who in his earlier days had done pioneering work in artificial intelligence. In his later years, Papert pioneered a "constructionist" theory of education that emphasized hands-on, experiential learning rather than rote drills and memorization. Today, Scratch represents a living lab for Papert's hands-on ideals.

"It's both a positive space but also an open space," Resnick explains of his creation. Most online communities for children limit participants to a narrow set of responses and actions. Scratch offers its coders near-total freedom to be as serious or silly as they want to be. Kids have used Scratch to code Trump-themed "Build that wall!" video games, songs espousing LGBT pride, and animated simulations of violent flatulence. Paid adult moderators filter out anything that's directly insulting to groups or individuals; in general, though, the Scratch community is kept in check by its population of eager underage coders — most of whom take the time to comment on, collaborate with, and "remix" the projects of their fellow Scratchers. There's just not much incentive on Scratch for trollery: Coding projects can take hours to make using the language's step-by-step command blocks (MOVE 10 STEPS FORWARD; WAIT 3 SECONDS, PLAY "ATOMIC FART" SOUND). Why would anyone spend hours building a nasty insult that will only be deleted post-haste?

The result of this kid-first ethos is a rarity on the internet: a genuinely popular, yet non-hateful, social network. One Scratch staffer refers to the site as "a magical unicorn place."

Equally heartening is the Scratch team's focus on kids who aren't typically drawn to activities like coding — whether because of lack of exposure, lack of confidence, or a dramatist-type play style. "We've tried to put special focus on kids who haven't had educational opportunities," Resnick says. "We're not just picking off the ones that are already succeeding." While some patterner-type players might be drawn to Scratch solely to learn how to code, many new users are hooked by the chance to explore their existing passions. To this end, Scratch

Scratch backer David Siegel '83

greets novices with a number of tutorials and sub-communities geared toward subjects that have nothing (explicitly) to do with computer tech. Within a few hours on the site, for instance, they can learn how to make a basic hip-hop dance, design and sketch a prom dress, or create a "pong-style" sports-video game.

Scratch 3.0, which Resnick and his MIT team are prepping for release in 2018, should extend the platform's reach even further. The release will be the first designed for use on smartphones and tablets, and will include new ways to link Scratch coding to the real world with robotics and sensors.

Supporting this effort is the Scratch Foundation, which was started by Resnick and David Siegel '83 to spread the gospel of coding for all. The two men first met as graduate students in the computer science department at MIT. Resnick continued on to a career in academia, while Siegel founded Two Sigma Investments. Their paths crossed again earlier this decade after Siegel's son got hooked on Scratch. "My son was able to write cool little programs," Siegel recalls, "but what I most picked up on was that my son was starting to look at the world differently. He was starting to think about the world more as a series of processes, like algorithms working together." The foundation's support has allowed Scratch to remain free and not-for-profit and to employ the team of moderators and developers that keep the community vibrant at a time of rapid growth.

Beyond pouring millions into the Scratch Foundation, Siegel is eager to speak out on the need for the Scratch program. It's not just about teaching children to code, he says, or even about helping them to land better jobs in an increasingly digital economy. What's at stake is more fundamental, he says: determining how our youngest generation engages — or doesn't — with the tech-saturated world around it. *(I'll quote Siegel at length here because when he explained it all to me, I felt an exquisite shiver of doom that I'd like to pass on to you, my reader.)*

"Whether or not you are going to be a software engineer some day, for sure every person on this planet is going to be dependent on computers," Siegel begins. "The big problem is, most people in the world have no idea what's going on inside of a computer. They don't know what software is. They look at their phone and they see an incredible black box. When you go to your computer and you're typing into Microsoft Word, you have no idea what's going on. When you go to a web browser and you search for something, you have no idea what's going on behind the scenes." *(I nod.)*

"So is that OK?" he asks.

(I shake my head no, then look down at my phone with a newfound shame.)

"Well, I think that's not OK, and I'll tell you why.

"I think it's not OK because it destroys curiosity. If you have no clue how the most important things you're interacting with every day work, you begin to tune out from the world. Because it's sort of like this world is too complicated for me to understand. It's too complicated, it's magic, I'm just basically going to ignore it and be oblivious to what is happening around me. I think [coding illiteracy] causes people in general to tune out and to become somewhat despondent. Because if you're living in a world where you don't know how it works, then that's not a very good feeling, is it? That will cause you to feel like you don't even have any opportunity."

Another big problem, Siegel says, is that many parents and teachers also don't know how technology works. If your kid asks why the sky is blue, Siegel offers, you'd tell him. If he asks how airplanes work, you'd tell him. But if he asks how software works ... chances are you wouldn't know what to say. *(My shame intensifies.)* That's where Siegel sees the Scratch Foundation coming in: "It's critical that we rise to the challenge of demystifying computer technology. Whether or not you're going to be a programmer, you need this form of education. And Scratch is providing this foundation to tens of millions of kids around the world every day."

This is a bold, world-sweeping mission for a toy with a cartoon cat as its mascot. But it's a vision that Mitchel Resnick, for all his professorial modesty, shares with his partner. At the end of last year's Scratch conference at MIT, Resnick led the crowd through a round of Scratch's official theme song. Written by Resnick, and recorded with the help of Scratchers worldwide, it's set to the tune of Coke's utopian '70s jingle "I'd Like to Teach the World to Sing (In Perfect Harmony)." It's a song, it's a manifesto, it's a prayer. (When is a toy more than just a toy? When it's 2017, and we all need deliverance from the techno-illiteracy that enshrouds us.) And so the people sing:

I'd like to teach the world to code
And think creatively
Make art and stories come alive
While learning joyfully

I'd like to teach the world to code
It's everybody's right
It helps you think and share your thoughts
Like learning how to write

It's time to code
SCRATCH! ON!
It's time to code
SCRATCH! ON!
Design, debug, remix with friends
The learning never ends ...

David Walter '11 is a freelance journalist in New York.

Courtesy Scratch Foundation