Learning Computing through Creating and Connecting

Karen Brennan, ***Harvard University***

Despite increased cultural awareness of computing, programming remains inaccessible to many young people. In response to this challenge, MIT Media Lab researchers are developing Scratch, which combines an authoring environment for interactive media projects with an online platform for sharing those projects.

Computer scientists are often stereotyped as highly focused loners, trained primarily in algorithmic theory and fluent in the esoteric languages of computer programming. Yet, I imagine that even in college, while spending considerable time by themselves in computer labs, they share many learning experiences with others: hacking with peers, getting advice from instructors and teaching assistants, and drawing upon the expertise of colleagues near and far.

And what of typical computer scientists' experience after schooling? They never stop learning—the field of computing is far broader and deeper than their formal education could ever have prepared them for, and it is constantly changing and expanding. By necessity, computer scientists continue to be deeply engaged in the learning process, both through creating—writing code, designing programs, and building systems—and through *connecting*—whether with a coworker in the office next door or with a developer on the other side of the world.

Given computing's importance in nearly every aspect of life, it seems self-evident that computing should be included in the learning experiences of all young people. By this, I do not mean to suggest that everyone should be prepared to be a programmer or software developer; rather, young students should develop an appreciation for those fundamental computing practices that will serve them regardless of their ultimate career or life goals.

Here, I describe an approach to computing education that emphasizes creating and connecting through the Scratch programming language. First, I introduce Scratch and describe how it has evolved since its launch in May 2007. Next, I describe how educators can make the Scratch experience accessible to more young people and, in the process, help broaden participation in computing. Finally, I describe the ScratchEd professional development model, which supports teachers working with Scratch to create an engaging learning environment.

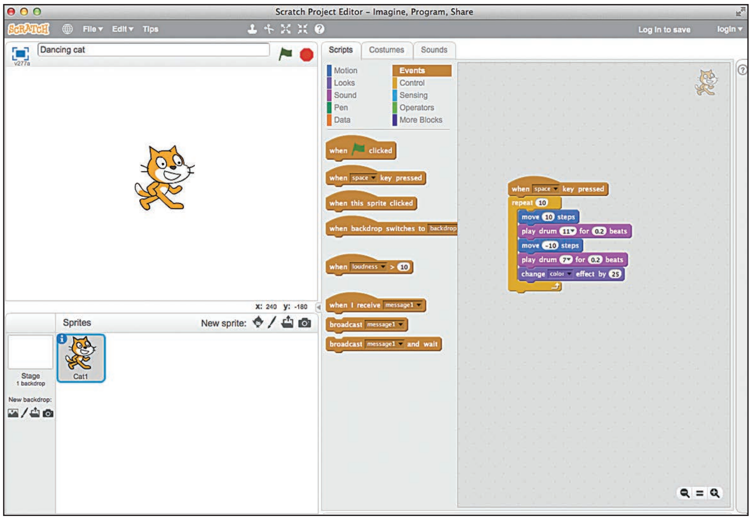
SCRATCH

Despite increased cultural awareness of computing—through events like Maker Faires (http://makerfaire.com) to movies like *The Social Network* to initiatives like Codecademy's Code Year (www.codecademy.com/tracks/code-year)—programming remains inaccessible to many

Published by the IEEE Computer Society

young people. This is due primarily to a lack of appropriate-age-level development tools as well as knowledgeable support from others. Programming languages have historically been challenging for young people, involving specialized syntax that is unforgiving of even the smallest error, such as a missing semicolon.[1] And although some can turn to family members and friends with computing experience to help them get started, finding communities of young people who share an interest in and passion for programming can be difficult.

In response to these needs, the MIT Media Lab's Lifelong Kindergarten research group is developing Scratch (http://scratch.mit.edu). Combining an authoring environment for interactive media projects with an online platform for sharing those projects, Scratch provides a rich context for informal and formal computing education.



```
when space ▾ key pressed
repeat 10
  move 10 steps
  play drum 11▾ for 0.2 beats
  move -10 steps
  play drum 7▾ for 0.2 beats
  change color ▾ effect by 25
```

Figure 1. **Sample Scratch project. The programmer "snaps together" various Events, Control, Motion, Looks, and Sound blocks much like physical Lego blocks. Pressing the space bar causes a cat to dance back and forth while changing color.**

Scratch origins

One of Scratch's significant predecessors and sources of intellectual inspiration is Logo (http://el.media.mit.edu/logo-foundation/logo/programming.html), a graphic-oriented Lisp-based programming language developed in the late 1960s by Seymour Papert and his colleagues at MIT in cooperation with the technology firm Bolt, Beranek, and Newman (now BBN Technologies). Papert's vision for Logo still resonates today:[2]

> In most contemporary educational situations where children come into contact with computers the computer is used to put children through their paces, to provide exercises of an appropriate level of difficulty, to provide feedback, and to dispense information. The computer programming the child. In the LOGO environment the relationship is reversed: The child, even at preschool ages, is in control: The child programs the computer.

With roots in the Intel Computer Clubhouse Network of after-school technology programs, Scratch builds on Logo as well as on other languages focused on making programming accessible to young people, including Design By Numbers (http://dbn.media.mit.edu), Logo Blocks (http://llk.media.mit.edu/projects.php?id=141), Etoys (www.squeakland.org), and Alice (www.alice.org/index.php).

How Scratch works

Scratch is a blocks-based programming language that enables young people to create interactive media, such as games, stories, and simulations. Its visual-oriented approach differentiates it from most programming languages, such as C and Java and even Logo, which typically require the programmer to type text-based instructions and grapple with any resulting syntax errors. Using Scratch, young designers "snap together" programming blocks to create artifacts in the digital world, just as one might snap together Lego bricks to create artifacts in the physical world.

Scratch offers more than 100 programming blocks, grouped into 10 different categories: Motion, Looks, Sound, Pen, Data, Events, Control, Sensing, Operators, and More Blocks. In a Scratch project, blocks are used to manipulate the attributes of objects, called *sprites*. For example, blocks in the Motion category can be used to modify a sprite's movement and position, and blocks in the Looks category can be used to modify a sprite's visual appearance.

Scratch projects are created by adding sprites and then programming their behaviors by snapping together blocks from these different categories. In the sample project shown in Figure 1, various Events, Control, Motion, Looks, and Sound blocks have been snapped together so that when the space bar is pressed, a cat—the default sprite—repeatedly dances back and forth to a drum beat, while simultaneously changing color.

Using this basic mechanism of adding sprites and then connecting blocks to specify the sprites' behavior, young people can develop a wide range of personalized projects. Figure 2 illustrates some of the possibilities: an interactive story featuring a series of knock-knock jokes between a monkey and a lion; an interactive art gallery featuring explorations into computer-based art; a Sims-inspired simulator that lets users design their own office; and a side-scrolling maze game that involves navigating a small green square past a series of hazards.

In addition to providing an authoring environment for computational media, Scratch includes a setting

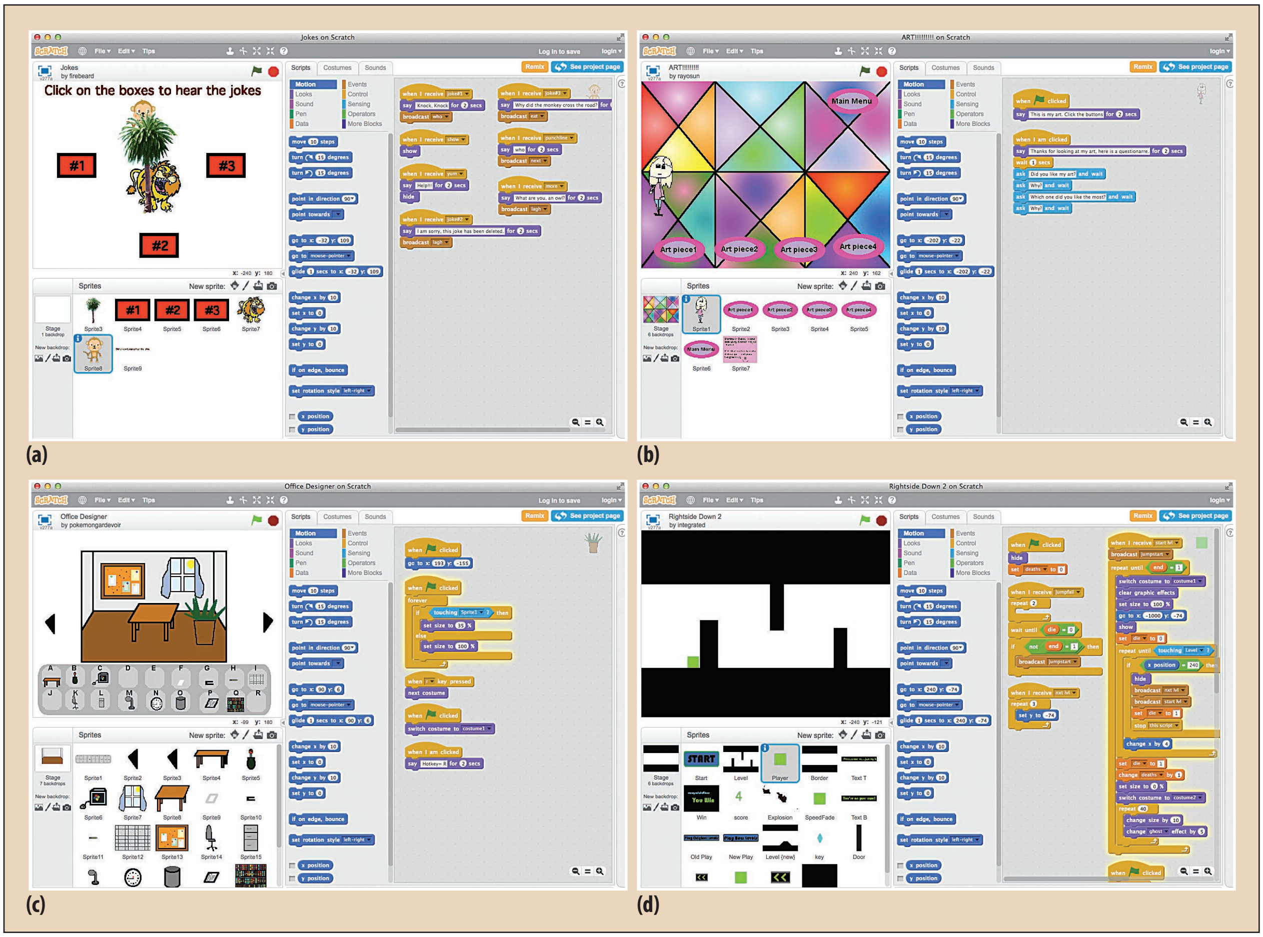


Figure 2. Four Scratch projects that explore a range of genres: (a) a story, (b) an art gallery, (c) a simulator, and (d) a game.

to share individual projects with others. Inspired by Papert's educational philosophy derived from Brazilian samba schools—in which learning is motivated by one's passions and occurs through exchanges between novices and experts—and by early experiments in computational creation communities,[3] Lifelong Kindergarten created an online network for Scratch.

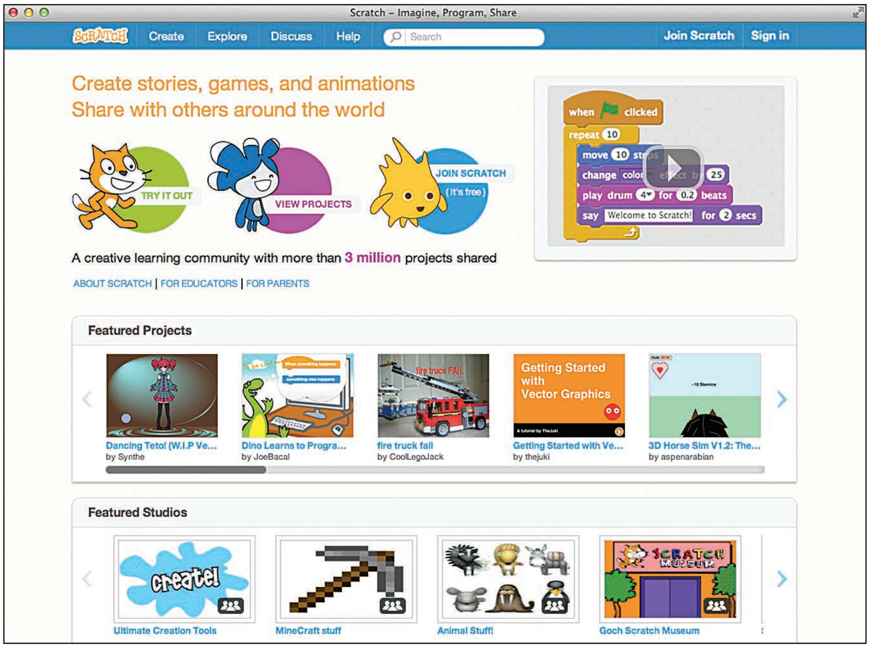
As Figure 3 shows, the Scratch online community provides a venue for members to connect with and support one another. It has grown considerably since the Scratch site was launched, with hundreds of thousands of members sharing Scratch projects.[4] Members—most of whom are between the ages of 8 and 16—post more than 2,000 new Scratch projects each day, and more than 3 million projects have been uploaded over the site's first six years.

The online community site includes numerous social networking components. Each member has a profile page displaying his or her projects and other dimensions of participation, such as friends and studios. Members can comment on others' work, express appreciation for projects through the Love It link, and add projects to studios. In addition to interacting with and providing feedback to others, members can examine a project's source code to study how it was created, looking at how the sprites and blocks have been connected.

Scratch users are also strongly encouraged to borrow from others' ideas, with appropriate credit. The most recent release—Scratch 2.0, available as of May 2013—integrates a project editor into the website that eliminates the barrier that once separated development activities in the authoring environment from social activities in the online community.

Scratch user experiences

My observations of and conversations with young Scratch users indicate that creating interactive media projects and connecting with others are both central to the learning experience. As Clark, a 12-year-old user, commented, "It's a really great program that allows you to create your own games, animations, and … your own media. So it's definitely a way you can be yourself, make whatever you want, and then, of course, there's an online community in which you can meet new people, be inspired to create your own stuff, and just experiment, and have fun."



Create stories, games, and animations
Share with others around the world

A creative learning community with more than 3 million projects shared

Figure 3. Scratch online community homepage.

Scratch users highly value the freedom they have to explore different creative methods—even though such freedom is often accompanied by many challenges. The rewards—and accompanying difficulties—of programming were recurrent themes in our conversations. Many talked about the great satisfaction and pride they felt struggling with and ultimately completing a Scratch project. Lindsay, for example, described learning "how difficult programming could be, but how great the results are once you finally finish it." Likewise, Jan felt that "if there's no challenge, it's not fun to create a project," and Allison described her Scratch experience as "an engaging puzzle," going on to say that programming is "a lot of fun. It's challenging. It makes you think."

Equally important to the Scratch users I interviewed is connecting with others.[5-8] In home settings, parents and other older relatives often provided support in breaking down a problem and thinking through its challenges logically, if not specific expertise in using Scratch. Siblings and friends also helped generate ideas for projects and occasionally provided technical assistance as well. However, many home users relied on the Scratch online community both as an audience and as support for their projects. In the words of 16-year-old Jan,

> I don't know how it feels to be a new Scratcher now, but I know that I got very constructive responses on my projects, a lot of help from other Scratchers—and this help has been very important for me. I think if that wasn't there, I might have not, you know, continued in Scratch and be where I am now. So I think it was really important for me and therefore I think it's important that we support this kind of giving and receiving help in Scratch.

BROADENING THE SCRATCH EXPERIENCE

Many of the young people I interacted with in the formative days of the Scratch community came from environments that encouraged and supported the creative exploration of technology. When asked how they had found Scratch, users would tell me, "My mom (or dad or sister or another relative) showed me," and on further inquiry I'd find that these family members more often than not were computer scientists, programmers, or engineers—thus providing crucial preparatory privilege.[9] So I began to wonder: in making computing available to as wide a range of young learners as possible, what was the best way to move beyond these early, privileged adopters and thereby broaden the Scratch experience?

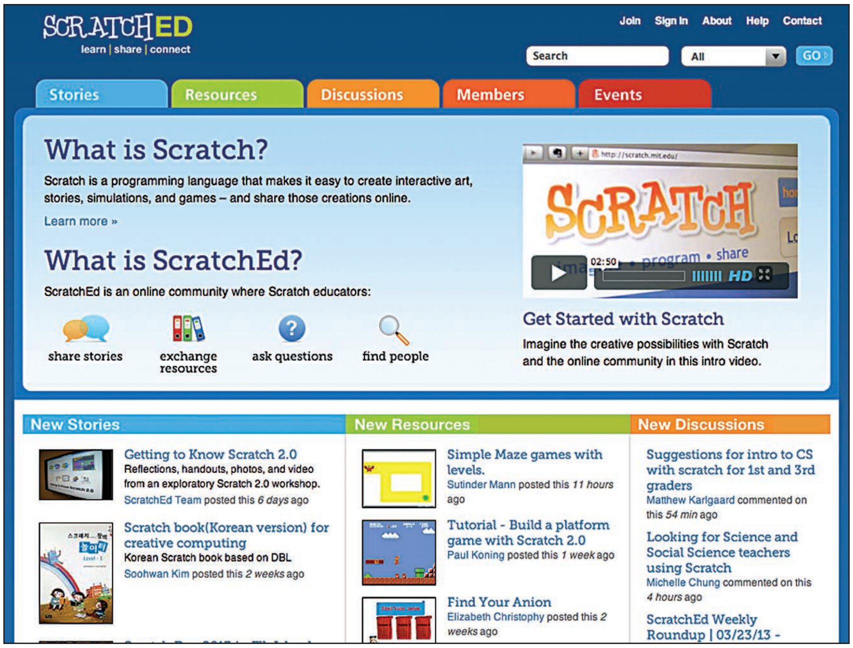
Scratch in the K-12 classroom

One approach, which I have been studying since 2007, is to include Scratch in the K-12 curriculum. Teachers play a critical role in Scratch's classroom adoption, and my work has focused on understanding how they work with Scratch across grade levels and curricular areas, and on developing materials to support teachers in these efforts. I have been particularly interested in how Scratch aligns—or is at odds—with teaching approaches that emphasize learning through creating and connecting. While some early classroom adopters did find ways to relate the Scratch experience to creating and connecting, I discovered that many others too often relied on strategies that subtly, or even overtly, undermined these goals. These less-than-successful applications of Scratch in terms of learning through creating and connecting could generally be traced to various pressures, including teachers' self-perceptions, school expectations, and student reticence.[6]

Challenges for K-12 teachers

Two challenges recurred in my interviews with K-12 teachers:

- finding a balance between providing support to students and allowing freedom for their creative processes, and



What is Scratch?

Scratch is a programming language that makes it easy to create interactive art, stories, simulations, and games – and share those creations online.

Learn more »

What is ScratchEd?

ScratchEd is an online community where Scratch educators:

Get Started with Scratch

Imagine the creative possibilities with Scratch and the online community in this intro video.

Getting to Know Scratch 2.0
Reflections, handouts, photos, and video from an exploratory Scratch 2.0 workshop.
ScratchEd Team posted this *6 days* ago

Scratch book(Korean version) for creative computing
Korean Scratch book based on DBL
Soohwan Kim posted this *2 weeks* ago

Simple Maze games with levels.
Sutinder Mann posted this *11 hours* ago

Tutorial - Build a platform game with Scratch 2.0
Paul Koning posted this *1 week* ago

Find Your Anion
Elizabeth Christophy posted this *2 weeks* ago

Suggestions for intro to CS with scratch for 1st and 3rd graders
Matthew Karlgaard commented on this *54 min* ago

Looking for Science and Social Science teachers using Scratch
Michelle Chung commented on this *4 hours* ago

ScratchEd Weekly Roundup | 03/23/13 -

Figure 4. ScratchEd online community homepage.

- promoting social learning within a school culture where assessment and evaluation are conducted primarily on a per-student basis.

Given time constraints and the large number of students in typical classrooms, teachers often regarded starting with a blank Scratch project as too risky. Accordingly, many provided structure to help students get started. In one such approach, teachers used published books about Scratch that provide step-by-step instructions for making successful projects, hoping this could garner expertise while simultaneously appealing to student interests. Unfortunately, and perhaps unsurprisingly, this knowledge building came at the expense of creative freedom—particularly when student experiences were entirely limited to following prescribed steps—and often undermined students' sense of creative self-direction, effectively stifling their long-term interest and motivation.

In addition, some of the teachers I interviewed were intrigued by the Scratch online community and saw its potential value, imagining ways their students could productively interact with other learners through collaboration, communicate with an audience, and critique others' projects. However, many still expressed anxiety about students having such unlimited access to the work of others, equating the Scratch remix culture with a lack of originality—or worse, plagiarism. In these instances, teachers often discouraged or attempted to prevent students from consulting other projects and people on the website, thereby denying them important feedback.

Meeting teachers' needs: ScratchEd

Teachers need support to negotiate these challenges successfully and implement new strategies. Although the Scratch online community has a large and active membership, it was designed for people who want to create and share projects. Educators, by contrast, are primarily concerned with helping others create projects. This disparity came into sharp focus for me when, shortly after joining Lifelong Kindergarten in 2007, I began receiving numerous emails from teachers requesting curricular resources or offering to contribute resources they had created, asking questions or suggesting possible answers if others had questions. Teachers were obviously looking for ways to connect with peers working to implement Scratch in their classrooms.

Based on such inquiries and motivated by the "community of practice" model—which gives teachers, as learners, access to peers with common goals and concerns[10]—I developed the ScratchEd site (http://scratch-ed.org), as Figure 4 shows. Teachers interested in or already working with Scratch can use the site to find other educators with whom they can share stories, exchange resources, and ask (or answer) questions. Since the public debut of ScratchEd in August 2009, more than 8,600 educators from around the world have joined the community, contributing some 215 stories, 580 resources, and 3,700 discussion posts. Over the past year alone, the site has received an average of 92,000 page views from 18,000 unique visitors each month.

From the start, the ScratchEd online community has provided broad support for creating and connecting with Scratch in the classroom—but more was needed, particularly for educators unfamiliar with computer science fundamentals and especially with programming practices.

With National Science Foundation funding, my colleagues and I have expanded the ScratchEd professional development model to include face-to-face and online gatherings where teachers can develop fuller competency with Scratch through hands-on experiences and synchronous discussions, including quarterly introductory workshops for teachers new to Scratch and monthly networking opportunities for experienced users.[11]

We have also developed new resources for teachers introducing Scratch to students and conducting workshops for colleagues—resources that, like much of the support material developed for Scratch, emphasize creating and connecting. A curriculum guide released in 2011[12] was within a year downloaded more than 40,000 times by ScratchEd community members and translated into many languages, including Portuguese, Spanish, Turkish, Korean, and Traditional Chinese. Teachers receive announcements about new resources not only on the ScratchEd site but also through other channels they rely on such as email, Twitter, and Facebook.

STRATEGIES FOR CREATING AND CONNECTING WITH SCRATCH

As education researchers Marlene Scardamalia and Carl Bereiter observed, technologies "cannot compel the adoption of a certain approach to teaching."[13] This means that educators must reflect on the type of learning culture in which they will use Scratch. How might students be provided opportunities for creating and connecting? What cultural transformations might need to take place, and how can educators initiate and sustain those transformations?

To support such reflection and exploration, the ScratchEd professional development model provides opportunities for teachers to create and connect with Scratch as well as to document those experiences for themselves and others. Based on analysis of data from classroom observations, personal interviews, and content posted on ScratchEd, I have identified five strategies for successfully implementing Scratch across age ranges and curricular areas.[6]

Try it yourself

Many teachers have remarked that cultural change in the classroom often starts with exploring the possibilities of Scratch based on publicly available resources, such as materials posted on ScratchEd, and on testimonials from other educators participating in face-to-face or online development forums. But however useful these resources may be, teachers also emphasize the importance of taking time to create Scratch projects on their own—not to attain mastery, but to develop basic familiarity with the interface and to foster individual confidence.

Teachers trying Scratch themselves also come to recognize the experiences and challenges students face. One teacher I interviewed talked specifically about how working on a Scratch project gave him new insight into his students' creative processes. He noted that he first spent considerable time thinking about what he wanted to do and then even more time thinking about how he would go about it. His personal experience made him more sensitive in responding to students' quiet moments "waiting for inspiration" and overcoming creative challenges.

Follow learner interests

Teachers often describe the importance of supporting students' interests and passions. For example, Larissa, a K-12 teacher, spoke extensively about allowing for such space, describing one student project in particular. Larissa had asked each of the students in her technology class to select a topic, research it using online and library resources, and develop a Scratch project that would help people interacting with the project learn more about that topic. One student, Maria, chose to focus on crocodiles and came to Larissa saying, "I want to know the sound the crocodile makes." Before responding positively to Maria's request ("I don't know what the sound of a crocodile is. Let's find out."), Larissa experienced a moment of hesitation, balancing Maria's priorities with her own as the teacher. Ultimately, as she explained, Larissa opted to invest the time necessary to follow Maria's interests, even though doing so was not necessarily the most expedient choice.

Educators must reflect on the type of learning culture in which they will use Scratch. How might students be provided opportunities for creating and connecting?

Teachers have offered various strategies for developing activities that appeal to students' personal interests while connecting them to larger sociocultural contexts. One high school computer science teacher, for example, described tapping in to his students' passion for videogames as a means of motivating them to learn about programming.

Be a guide

How can students learn to solve problems on their own if the teacher is always ready to step in as the "expert"? Teachers I have interviewed about using Scratch in the classroom often describe their role, rather, as a guide, mentor, supporter, or facilitator, helping students pursue goals through metacognitive means: asking questions, providing useful resources, breaking down problems into smaller components, and suggesting ways to reframe problems.

Teachers also talk about serving as emotional guides, helping students negotiate an expanding landscape of personal autonomy, experimentation, and creativity. They describe the importance of encouraging students to be fearless in setting goals and to persevere in the face of myriad difficulties. According to Taylor, an elementary school teacher,

> It starts on the first day of school—getting kids to appreciate that they're going to make mistakes and that I'm going to be asking them to do stuff that is hard. And they don't, at first, just because they want to succeed—and part of it is the age, too. Well, even adults don't like to fail, or make mistakes. But it's

> important, I feel, that when you do run into difficulties that it's not time to give up or cry. It's time to think about the strategies that you have to solve your problem, or to look for help. No reason to break down or give up—you keep at it.

Feel OK with not knowing

The cognitive and emotional uncertainty students encounter in creative challenges with no preset solutions applies equally to teachers. Good teachers come to realize that they cannot know the answer to every question, and learn to feel comfortable with this realization. Those I've talked to have achieved confidence using Scratch in the classroom by developing a general set of strategies for problem solving and a mind-set that recognizes the importance of being flexible in one's thinking and open to what the learning experience can lead to.

"Being OK with not knowing" creates opportunities to disrupt the teacher's role as the person to come to for answers. Teachers have described to me the sense of power, surprise, and even delight they experience working together with a student to untangle a difficult problem. For many teachers, this feeling is even greater when they are learning *from* as well as *with* students—a relationship that enhances the educational experience of both.

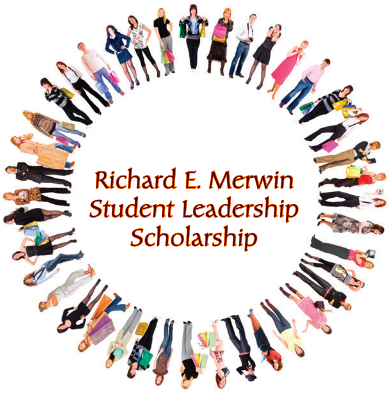
Create opportunities to share

Teachers working with Scratch in the classroom consistently describe the importance of student sharing—from code to ideas to feedback—at different stages of a project's development, with different groups of people, over different periods of time. One teacher I interviewed talked about the importance of students presenting their final work—participating in a "gallery walk" with classmates, interacting with one another's projects, and asking each other questions, whether as creator or user. Another teacher noted that students working on interactive narratives in conjunction with a partner thought differently about how users might interact with their narratives. A third teacher successfully used demoing to support peer learning, starting each class by having students share something they had figured out in the previous session.

Whatever form it takes, teachers value peer sharing as a powerful element of the Scratch learning process, critical to helping students overcome their creative challenges. As Sun, a teacher at both the elementary and college level, summarized it,

> The students wanted to learn more. They used to work one-by-one, but then they would encounter a big mountain or big barrier. That's why I encourage them to make a community—without community, it's not possible. The tool is important, but making community is more important for learning.

All young people should have opportunities for computing education, opportunities that position them not merely as consumers of computational artifacts but as explorers and designers of the worlds that computing offers. The goal of these learning experiences is to cultivate in students an understanding of and appreciation for the concepts and technologies that shape the world around them; for some, these lessons might even serve as a point of entry to a computing career.



K-12 teachers have a central role to play in these activities and must be supported in thinking about new approaches to computing education—approaches that encourage creating and connecting and that foster the capacities for computational thinking necessary for full participation in 21st-century life. Such support at the K-12 level will help broaden participation in computing, reaching those without access to computers and technical expertise at home and enabling a wider range of young people to imagine themselves as computational creators. C

Acknowledgments

This material is based on work supported by the National Science Foundation under grant no. 1019396. Any opinions, findings, and conclusions or recommendations expressed are those of the author and do not necessarily reflect the views of the NSF.

References

1. C. Kelleher and R. Pausch, "Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers," *ACM Computing Surveys*, June 2005, pp. 83-137.
2. S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, 1980.
3. A. Bruckman, "MOOSE Crossing: Construction, Community, and Learning in a Networked Virtual World for Kids," doctoral dissertation, MIT School of Architecture+Planning, 1997.
4. M. Resnick et al., "Scratch: Programming for All," *Comm. ACM*, Nov. 2009, pp. 60-67.
5. K. Brennan and M. Resnick, "Imagining, Creating, Playing, Sharing, Reflecting: How Online Community Supports Young People as Designers of Interactive Media," *Emerging Technologies for the Classroom: A Learning Sciences Perspective*, N. Lavigne and C. Mouza, eds., Springer, 2012, pp. 253-268.
6. K. Brennan, "Best of Both Worlds: Issues of Structure and Agency in Computational Creation, In and Out of School," doctoral dissertation, MIT School of Architecture+Planning, 2012.
7. K. Brennan et al., "More than Code: The Significance of Social Interactions in Young People's Development as Interactive Media Creators," *Proc. World Conf. Educational Multimedia, Hypermedia & Telecommunications* (EdMedia 11), AACE, 2011, pp. 2147-2156.
8. K. Brennan, A. Monroy-Hernández, and M. Resnick, "Making Projects, Making Friends: Online Community as Catalyst for Interactive Media Creation," *New Directions for Youth Development*, vol. 2010, no. 128, 2010, pp. 75-83.
9. J. Margolis et al., *Stuck in the Shallow End: Education, Race, and Computing*, MIT Press, 2008.
10. S.A. Barab, M. Barnett, and K. Squire, "Developing an Empirical Account of a Community of Practice: Characterizing the Essential Tensions," *J. Learning Sciences*, vol. 11, no. 4, 2002, pp. 489-542.
11. K. Brennan, "ScratchEd: Developing Support for Educators as Designers," *Designing with Teachers: Participatory Approaches to Professional Development in Education*, E. Reilly and I. Literat, eds., USC Annenberg Innovation Lab, 2012, pp. 67-77; http://dmlcentral.net/sites/dmlcentral/files/resource_files/pdworkinggroup-v6-reduced.pdf.
12. K. Brennan, *Scratch Curriculum Guide Draft*, 3 Nov. 2011; http://scratched.media.mit.edu/resources/scratch-curriculum-guide-draft.
13. M. Scardamalia and C. Bereiter, "Higher Levels of Agency for Children in Knowledge Building: A Challenge for the Design of New Knowledge Media," *J. Learning Sciences*, vol. 1, no. 1, 1991, pp. 37-68.

***Karen Brennan** is an assistant professor in the Graduate School of Education at Harvard University. Her research interests include studying and designing for creativity in the digital age, learning in communities, and broadening participation in computing. Brennan received a PhD in media arts and sciences from MIT. Contact her at karen_brennan@gse.harvard.edu.*