

Chapter 17

Imagining, Creating, Playing, Sharing, Reflecting: How Online Community Supports Young People as Designers of Interactive Media

Karen Brennan and Mitchel Resnick

We recently hosted a workshop for a group of 20 teenage girls to introduce them to work that is done at the Media Lab and, more specifically, in our research group, Lifelong Kindergarten. At the beginning of the visit, we asked the girls how they were currently using computers. Almost all of them had used computers to connect with other people—getting and giving personal updates through a social networking site, like Facebook, or chatting with friends and family through an instant messaging service, like Skype. They had also used computers to connect with content—watching videos on YouTube, listening to music on Grooveshark, reading articles on Wikipedia, or playing games on Miniclip. However, other than using office productivity software to write papers or create presentations, the girls were not actively engaged in creating their own media let alone interactive media, such as games.

The girls' answers were not particularly surprising—most young people do not have opportunities to engage in the design or creation of interactive media. We shared that one of the goals of our research group is to enable a wide variety of people to engage in technology design activities. Whether it is making your own robot or making your own software, we think people have powerful learning experiences when they are able to connect their personal interests with the design of artifacts. We added that we develop tools that make those design experiences available to new audiences.

In the workshop, we gave the girls a hands-on introduction to one of the tools that our research group has been developing, called Scratch. Scratch (<http://scratch.mit.edu>) is a programming environment that makes it easy to create interactive media such as games, stories, and simulations. Unlike text-based programming languages (e.g., Java or C++), with which you need to type out programming instructions,

K. Brennan (✉) • M. Resnick
Media Lab, Massachusetts Institute of Technology,
Cambridge, MA, USA
e-mail: kbrennan@media.mit.edu; mres@media.mit.edu

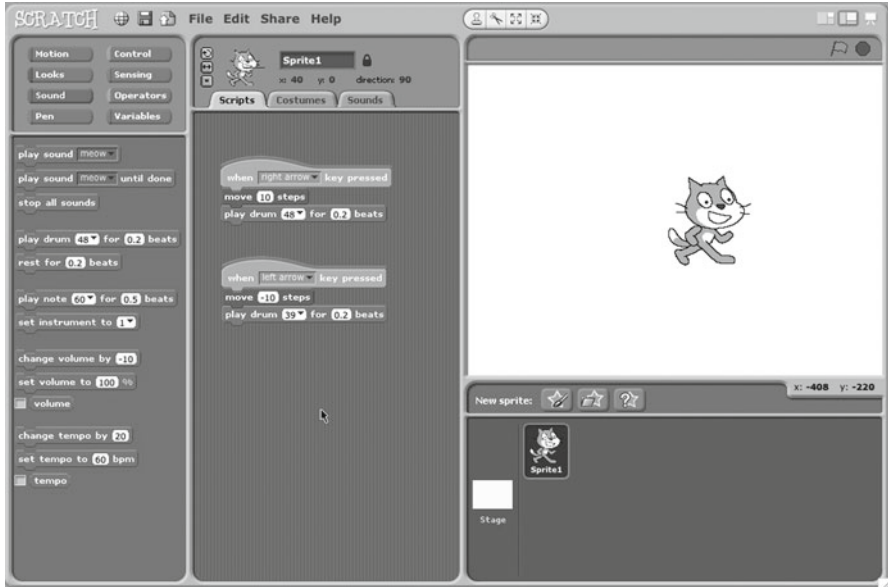


Fig. 17.1 Using the Scratch environment to program a cat sprite

Scratch uses a graphical, blocks-based language for programming instructions. Just as you can use LEGO bricks to build up a more complicated structure in the physical world, you can use Scratch programming blocks to build up a more complicated structure in the digital world—in this case, using programming blocks to control the behavior of media elements and objects, called *sprites*, within a Scratch project.

There are more than 100 programming blocks sorted into eight different categories: *motion* (blocks to control the position and direction of a sprite), *looks* (blocks to change the visual appearance of a sprite), *sound* (blocks to play audio clips and musical notes), *pen* (blocks to programmatically draw), *control* (blocks to make decisions or modify the flow of the program), *sensing* (blocks to get information about the state of sprites in a project), *operators* (blocks to perform arithmetic, logic, and string operations), and *variables* (blocks to store data). Blocks from all different categories can be snapped together to program different behaviors.

For example, the *when right arrow key pressed* block (*control* category), the *move 10 steps* block (*motion* category), and the *play drum* block (*sound* category) can be snapped together in a stack, which can be used to control the actions of a sprite (which, by default, is a cat). In this program, whenever the right arrow key on the computer keyboard is pressed, the sprite is moved 10 units to the right, and a tambourine noise is played. Another stack of blocks can be added to move the cat 10 units to the left and then play a handclap noise whenever the *left* arrow key is pressed (Fig. 17.1).

What specific media elements are being programmed (e.g., the cat and drum sounds in the program described above) are as important as *how* they are being

programmed, and Scratch was designed to make it as easy as possible for creators to personalize their Scratch projects. Although the cat is the default sprite, it is easy to remove, edit, or add different sprites. Scratch comes with a large media library of sprites, backgrounds, and sounds. Creators can use Scratch's built-in paint editor to create their own visual elements. They can also import audio/visual elements into their Scratch projects by using external tools (e.g., Photoshop or GarageBand) to create elements or by using a web browser to find elements online (e.g., photographs on Flickr).

From this simple process of snapping blocks together and customizing media elements, we have seen a wide variety of projects created. Young people have been using Scratch to create interactive stories and animations based on their favorite pop culture icons or imagined characters, simulations based on math and science concepts, and games that are recreations of classics (like Pac-Man and Super Mario Brothers) or inventions of their own. There is no *one* way that Scratch is being used and we have been continually surprised by how young people have stretched what we thought was possible to create with Scratch.

Background

Constructionism and Software Design

Scratch follows in the constructionist tradition—an approach to learning that emphasizes the importance of constructing, building, making, and designing as ways of knowing; “that knowledge is not simply transmitted from teacher to student, but actively constructed by the mind of the learner. Children don’t get ideas; they make ideas” (Kafai & Resnick, 1996, p. 1). Constructionism is grounded in the belief that the most effective learning experiences grow out of the active construction of all types of things, including the construction of computer programs. The Logo programming environment (developed by Seymour Papert and a team of researchers at MIT in the 1960s) was a major part of the constructionist tradition and has been a significant influence in Scratch’s development. Logo researchers studied how software design was a meaningful context for young people’s learning, particularly the ways in which the creation of computer games supported young people in developing design thinking and understanding mathematical concepts, such as fractions (Harel & Papert, 1990; Kafai, 1995).

More recent research has also supported both playing with and developing software as meaningful contexts for learning. Ito (2009) described the opportunities of children’s software for learning as three genres or cultural moments of software: *academic* software, *entertainment* software, and *construction* software. Unlike the academic and entertainment offerings, which organize learning around extrinsic rewards or amusement, Ito posited that the construction genre, which makes central the agency of young people as designers of their software experiences, offers the

greatest potential for learning and participation. Salen (2007), whose work has focused on the development of games, described the broad set of capacities that are required for game design—including critical thinking, complex problem solving, and persuasive expression—and the relevance of these capacities beyond a games context, forming the basis of a modern literacy that should be developed by all young people.

The design of software offers young people opportunities to engage in authentic challenges. Generalizing beyond software design, project- and problem-based approaches to learning recognize that the design of solutions to authentic problems contributes to deep and meaningful learning, going beyond the acquisition of superficial facts (Barron et al., 1998; Kolodner et al., 2003; Krajcik & Blumenfeld, 2006). Despite differences in *what* is being designed, all of the design-oriented approaches have a shared belief about the nature of knowledge—“knowledge as constructed by human inquiry rather than knowledge as ‘just there’” (Perkins, 1986, p. 19).

Iterative Design Process

This shared belief leads to a consideration of the *process* of design, which can be framed as an iterative approach that involves design cycles of *imagining*, *creating*, *playing*, *sharing*, and *reflecting* (Resnick, 2007). The *imagining* stage involves defining a problem space or imagining possibilities for an experience. A young person asks: What might I want to design? Why might I want to design it? The *creating* stage involves assembling the creative tools and starting to put the design together. A young person asks: What do I need to create my design? What are the pieces that make up my design? The *playing* stage involves testing out the artifact that is being created. A young person asks: Does my creation work? How is my creation aligned with what I imagined? The *sharing* stage involves presenting the designed artifact to others. A young person asks: Who can serve as an audience for my creation? What comments and feedback might I receive from others? The *reflecting* stage involves stepping back from the active design process to think critically about one’s progress. A young person asks: What have I figured out with my design? What remains to be understood and developed? These questions lead to new approaches and further iterations of the design cycle. Although described neatly here, the design process is often quite messy, with these stages sometimes happening concurrently, in a different sequence, or with uneven emphasis.

We illustrate this iterative design process with an example. Alex, a 9-year-old, was constantly sharing with his mother the ways in which he could *imagine* modifying and improving the games he enjoyed playing online. His mother introduced him to Scratch and he was excited about the possibility of making his own games. After tinkering with the basics of Scratch for a while, he started to *create* an elaborate maze game. Each level of the game involved navigating a protagonist through a complex maze structure with rewards to collect and punishments to avoid.

He particularly enjoyed recording his voice and programming the game to congratulate the player whenever a maze level was completed. He continually *played* during his development process—writing a bit of the program, testing it out, writing a bit more, having new inspiration, getting stuck, experimenting—alternating between testing and creating. After a few weeks, he felt that his game was ready to *share* with others. He invited his parents to the computer in the family room and had them try out his game. Both of his parents were suitably impressed by his creation, but his mother suggested that Alex could add instructions at the beginning of the game. Alex *reflected* on her suggestion. It made sense to him because, as a player, he had always read game instructions, but he was not sure how the instructions should be presented. After his parents left, Alex sat down with some paper and a pencil and sketched out what he *imagined* for the next set of refinements to his project.

Exemplars

In the example provided above, Alex worked primarily on his own. However, we know that learning and creativity are enhanced through interaction with others since they are social processes (Csikszentmihalyi, 1997; Sawyer, 2006a). Theories about communities of practice and situated learning give us ways of thinking about how community settings can support a designer's learning by providing the learner access to other designers and designed artifacts (Brown, Collins, & Duguid, 1989; Lave & Wenger, 1991; Rogoff, 1994). Based on these theories and inspired by Papert's (1980) model of the samba school, our research group created an accompanying website for Scratch, *the Scratch online community*, where people of all ages come together to share their design work and support each other's learning.

The Scratch online community, launched in May 2007, has become very active, with more than a million registered members sharing, discussing, and remixing one another's Scratch projects (Resnick et al., 2009). Each day, members (mostly ages 8–16) upload more than 2,500 new Scratch projects to the website—on average, two new projects every minute—with more than 2.7 million projects available. The collection of projects uploaded is incredibly diverse and includes interactive newsletters, science simulations, virtual tours, animated dance contests, interactive tutorials, and many others, all programmed with the Scratch environment and its graphical programming blocks (Fig. 17.2).

In addition to enabling people to upload their projects, the site was designed with features typical of community-based content-creation sites, such as Flickr and YouTube. Members can leave comments on projects, annotate projects with tags, indicate admiration of projects by clicking the *Love It* link, and bookmark projects in a list of favorites. Members can also download each other's projects to learn how they were made and then build on each other's work by remixing projects. Members can mark other members as friends, create galleries or collections of projects with others, and participate in discussion forums. Each member has a profile page that

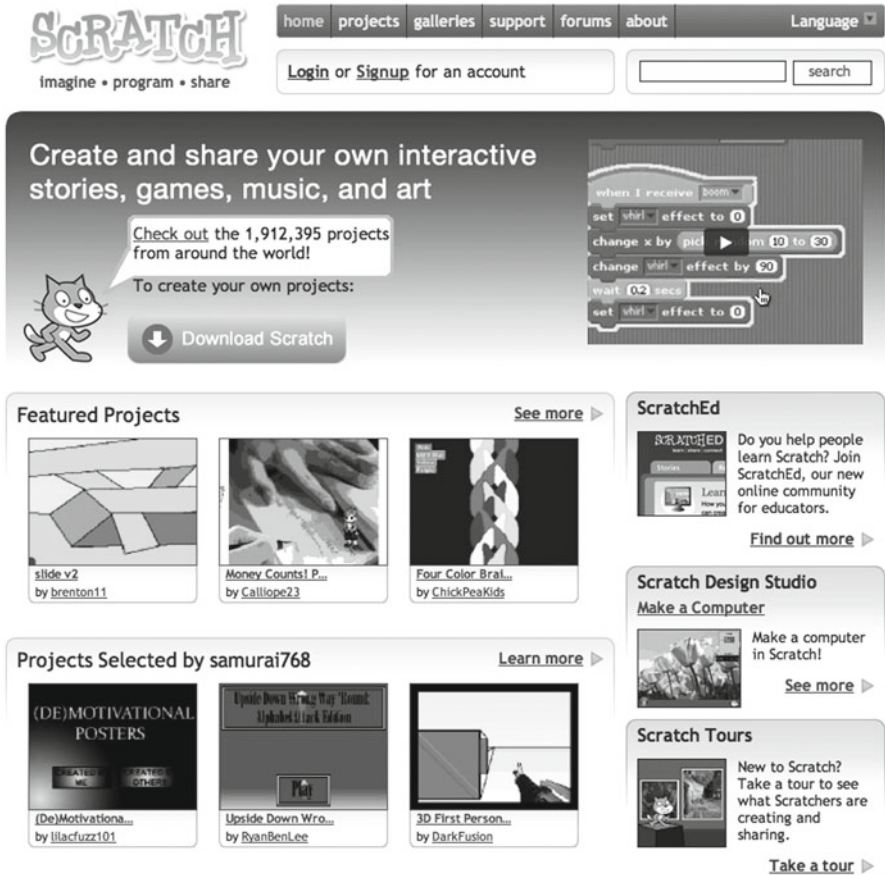


Fig. 17.2 The Scratch online community where young people share their interactive media creations

displays their alias and country, as well as his/her contributions and interactions, such as lists of projects, favorites, friends, and galleries.

Recent research has described the ways in which the social nature of young people's online participation serves as essential motivation and support for developing fluency of participation (Buckingham & Willett, 2006; Ito et al., 2009; Jenkins, Purushotma, Weigel, Clinton, & Robison, 2006). Whether hanging out with friends, playing games, or remixing media, having access to others makes for better participation, as young people are able to support each other in understanding practices and norms. Bruckman's (1998, 2006) work described the cognitive, social, and psychological benefits that an online community provided for individual learners

in constructionist activities. From technical support to emotional support, having access to others bolstered individuals' capacities for creative work.

We have seen that the Scratch online community supports young people's development as designers of interactive media. Having access to others supports *all* aspects of the iterative design process (*imagining, creating, playing, sharing, and reflecting*), not just the *sharing* stage of design. In the subsequent sections, we will share case studies from the Scratch online community to illustrate the ways in which having access to the community has supported young people's processes of *imagining, creating, playing, sharing, and reflecting*. These case studies are based on several years of Scratch online community observational field notes, as well as interviews with young Scratchers.

Exemplars of Imagining

For people who are new to a design tool like Scratch, the *imagining* stage is not just about defining a problem to solve or dreaming up an experience; it is about getting a sense of what might be possible to create with the tool. To help frame the possibilities, the Scratch application comes with a sample projects library. The online community significantly extends this library, with several million projects available online to serve as inspiration for people in the initial stages of a design.

Ten Levels

Courtney, 11, was introduced to Scratch by a friend from school. She was not sure what she might want to create, so she explored the Scratch online community to see what types of things other kids had been creating. She saw lots of different projects that she thought were interesting, but she found one that was particularly inspiring. The project was a game—a series of 100 mazes that increased in difficulty after each level. She thought that it was a great concept and wanted to make her own version of the game, but decided that she would start with fewer levels, perhaps 10 instead of 100. She gathered some paper and a pencil and started to sketch ideas for the mazes in her game. She imagined challenging obstacles to avoid, from spikes to lasers to lava pits, and tricky puzzles to solve. Courtney showed her sketches to her parents and her brother to get feedback for her maze levels, and looked for other maze projects on the website to get ideas.

The online community is not just a repository for projects that inspire the imagination. Rather, it is a location for people to explore shared interests in topics and to create interactive media together. These creative passions serve as another form of inspiration.

Mathematicians

Rebecca is 16 years old and loves mathematics. Rebecca found Scratch after her father suggested that, given Rebecca's interest in mathematical proofs, she might find the logical structures of programming appealing. When she first visited the Scratch website, she started by looking at the projects highlighted on the front page. She enjoyed looking at the list of most-recent projects, but was quickly overwhelmed by the number of projects that she found. Then she discovered the lists of most-viewed and most-loved projects. There were many games and animations, but Rebecca was not finding projects that she thought were personally interesting. She used the Scratch search engine to look for "math" projects and found hundreds of relevant projects in the search results. After interacting with a few dozen projects, which covered a wide range of mathematical concepts, she started to notice that particular member names were coming up again and again as the creators of and commenters on these math-focused projects. She had found a sub-community of mathematicians within the larger Scratch online community. Inspired by this group of people who share her passion for math, Rebecca created numerous projects about the different math concepts that she was learning in school and shared her projects with the online community. Rebecca thinks that the act of creating projects helps her to better understand the concepts that she is learning in school, and she hopes that her love of math will inspire others.

Exemplars of Creating

The large library of Scratch projects available online is meant to be not only a source of inspiration, but a source of building materials to help with *creating* Scratch projects. Not sure how to keep score in a game or how to make two sprites interact with each other? Find a project that does what you are hoping to achieve and examine its Scratch blocks. Every project on the site can be downloaded and its code studied as a way of learning particular techniques. New projects can be created by building up existing projects, becoming *remixes*. Remixed projects—created by young people finding projects, downloading them, changing them, and sharing them on the site—now constitute more than 15 % of all projects on the Scratch website.

Sidescroller Madness

Sean, 16 years old, loves playing video games, particularly sidescroller games. He tried to teach himself programming, but found that it was too complicated to make games on his own. After reading a news article about the launch of the Scratch online community, Sean was hopeful that Scratch might be a better tool for game design. He downloaded Scratch and looked at the sample projects. The sample games were

simpler than he had hoped, and there were no examples of the sidescroller games that he was trying to create. He turned to the online community, which at the time had only a few hundred projects, and still was unable to find an example of a sidescroller project. He decided to experiment on his own and discovered that it was easier to create a sidescroller game with Scratch than with other programming languages. When he created a basic game and posted it to the site, other community members responded enthusiastically to the emergence of this genre. Sean continued to make games, each one extending and refining his sidescroller techniques. Remembering his own initial excitement about creating a sidescroller game with Scratch, he decided to make a tutorial project for others. The project, explicitly intended for others to download and remix as the basis of their own sidescroller games, explained the mechanisms of a sidescroller game, step by step.

Studying the code of downloaded projects and developing an understanding of how projects work are powerful opportunities for learning. But whether someone has been using Scratch for 3 days or for 3 years, there will always be challenges that are just beyond understanding, even with access to others' programming blocks. Fortunately, each project on the website includes a link to the person who created the project, and the creators are often available for support and guidance. Sean, for example, made himself available as a consultant to others who needed support beyond his tutorial project. Community members have taken this peer support further, recognizing that when members work together as a *team*, ambitious projects can be created through their collaborative efforts and that the Scratch online community can be used to find others with similar design interests and goals.

Adventure in the Spooky Mansion

Sarah, a 13-year-old, and her 10-year-old brother love Halloween. Months before October 31st, they started planning their costumes and their route to visit neighborhood homes for treats. They both like creating Scratch projects and decided to create a spooky project to celebrate the day. Sarah loves the programming part and her brother loves to draw, but they wanted some help with both and with thinking of a concept for the project. They posted an announcement about their plan on the Scratch forums and invited others to participate in the creation of a project. Another Scratcher suggested creating an interactive project that would have the player navigating a spooky old mansion. Sarah and her brother loved the idea and the three of them started working on the plot of the story. They created an initial draft of the story and posted a link to the project in the forum thread. Other Scratchers were excited about the project and volunteered to help out—some were interested in working on the plot, others the programming, others the art. People working on the project downloaded the latest version, worked on it for a bit, and reposted it to the site, iteratively building up the project. On the day before Halloween, the group of contributors (which at its peak involved more than 20 community members) announced a final version. Community members gave the creators ample positive feedback on their project—a project that would have been challenging for any one of them to create on their own.

Exemplars of Playing

No design works as expected at first, and *playing* is important throughout the design process. Testing and experimenting with one's creations helps a designer understand what is (or is not) working, from trying out an individual block to experimenting with a stack of connected blocks to playing with a well-developed iteration of the project.

Works in Progress

Roan is 11 years old. He was introduced to Scratch at a lunch-hour school club and found that he loved using Scratch to create elaborate animations. But he never had enough time to perfect his creations during club time or at home, so he continued testing and developing his work across settings. He would start a project at the club and then upload it to the online community. Later, he would download it at home, assess what was not working yet, continue to work on it, and then upload it again. A single project sometimes resulted in dozens of uploaded iterations of his work. He knew that other people liked to keep their work secret until a final version was perfected, but he did not mind having his works in progress available to others. Although he initially adopted a post-early-and-often policy as a way of continuing his creative work between school and home, Roan found that he liked using it as a visual reminder of the decisions he made during his development process.

The participation of online community members provides new ways of thinking about the iterative development that emerges from playing with a project. Sometimes individual projects catch the attention of other Scratchers. Instead of one person taking responsibility for testing and refining a particular project, testing becomes an activity that spreads across Scratchers and new perspectives are incorporated in further iterations.

Tetris

Tetris is one of those classic computer games that everyone seems to know. So it was exciting (if somewhat unsurprising) to see a Scratch-based Tetris creation appear in the early days of the Scratch online community. The first version was a simple, elegant implementation of the game. Use the space bar to rotate and the left and right arrows to move the falling black blocks. Get a point for every full line of blocks that is created. Numerous people played the game and made suggestions for how it could be expanded. What if instructions were added to the project for people who do not already know how to play Tetris? What if the blocks were different

colors instead of all black? What if you could get a hint about which blocks would be appearing soon? What if you had a score and a count of how many lines were cleared? Over a period of several months, a few hundred implementations of Scratch Tetris appeared on the site, each one the result of a Scratcher having tested and played a previous iteration of the game.

Exemplars of Sharing

In some ways, the *sharing* aspect of the design process is the one to most obviously benefit from the online community. There is a continual sense of activity and audience in the community with more than a million registered members, roughly 300,000 of whom have shared projects on the site. Although sharing creative work with family and friends is a valuable experience, there is a different sort of excitement about connecting with and receiving feedback from people out in the world. In interviews, Scratchers frequently describe the motivation and satisfaction that an appreciative audience offers. However, in addition to more comments from more people, a larger audience can lead to different types of project development.

Response Tester

James is a 10-year-old boy who had been learning about response times in science class—i.e., how quickly a person can respond to an external stimulus and factors that can alter a person's response time. James was curious and wanted to experiment with response times himself. He had seen his older sister use Scratch to create interactive projects that she shared on the Scratch website and he decided to talk with her about his idea for a project that could be used as a response tester experiment. She helped him design a project that measured how quickly the person interacting with the project responded to changes in the project. At the end, the project reported the person's average response time and asked a few demographic questions (age, sex, number of hours of sleep per night). James posted the project to the website and hundreds of Scratchers tried it and shared their response times and demographic answers in comments below the project. He collected the data from the website, and with help from his mother analyzed the results. James wrote a report about the response tester project and shared it with his class at the annual science fair.

Individual projects can attract attention, but there are some Scratchers who have been able to achieve significant cultural resonance with the community by developing a series of popular projects. This situation can result in community-wide visibility for creators, leading to a large fan base for their work and to new forms of creation and participation.

Guest Spot

Caitlin is a 13-year-old girl. She is a devoted fan of anime (Japanese animation) and spends much of her free time sketching her own anime-style drawings. She recently started using her computer as a way of creating sketches. Her best friend learned about Scratch in an introductory computing course and suggested that Caitlin could use Scratch to create animations just like the anime that they both love so much. Caitlin started posting episodes in a story series, which gained a large following in the Scratch online community. Her projects regularly appeared on the front page of the site based on the number of community views, comments, and love-its that they received. Other Scratchers became invested in Caitlin's work, asking when the next episode would be released on the site, making requests for plot and character development, and creating fan projects as tribute to the characters. Caitlin appreciated her growing group of admirers and tried to think of ways to include them in her project development process, while still being able to maintain her vision for the series. She decided to have a "guest spot" in one of the episodes, and invited community members to submit entries for a new character who would appear in that episode.

Exemplars of Reflecting

Stepping back and *reflecting* on one's activities as a creator of interactive media is as important to the process as the other stages of design, and it is in large part what propels us to deeper understanding and learning (Sawyer, 2006b). The community artifacts that surround designers can support reflective activities, as the objects we create can be the objects that help us think about the meaning of our participation.

Scratch Stats

Fitch, a 10-year-old boy, who was relatively new to the Scratch online community, wanted to understand why some people are more popular or receive more attention than others on the Scratch website. On a visit to the website, Fitch found a page that contained visualizations of individual Scratchers' participation. He looked at his own visualization and discovered that the number of comments received was extremely low. For comparison, he decided to look at the visualization of Angela, who Fitch knew had been a Scratcher for several years and had received many more comments. Fitch saw that Angela's number of received comments had gone up and down over time, but what surprised him was that the graph of received comments was the same shape as (but three times larger than) the graph of given comments. Upon further reflection, Fitch realized that the differences in these visualizations were not just coincidence, and he shared his insights with other Scratchers in the Scratch online forums: to get comments, you need to give comments.

Having access to artifacts, such as visualizations and portfolios of projects, can effectively support learners' reflections. However, having access to other people can provide even more specialized scaffolding for learners' reflective practices. For example, others can ask questions about what creators of Scratch projects are (or should be) noticing about their own development as designers.

Puzzle Par

Tom is 13 years old and, for as long as he can remember, has enjoyed puzzles that explore patterns and combinations. One of his favorite games is Swap It, a logic puzzle where the player swaps adjacent colored tiles until the final colored tile pattern is achieved. He decided to create his own version of Swap It and share it with others in the online community. After Tom posted his project, Eric (a more advanced Scratcher) tried out Tom's project and left a congratulatory comment for Tom about his fun project, although Eric mentioned that the project was "pretty easy." Tom was very happy to get feedback on his game, and it helped him think about what it was like for someone else to experience playing his game. Tom was not sure how he might make the game harder, so he thanked Eric for the critique and asked for suggestions: "What do you think I could change to add a bit more of a challenge?" Eric responded with several detailed suggestions for extending the challenge of the game play, including adding the notion of par for each level, the minimum number of swaps needed to solve the level. Tom was very appreciative of the suggestions and thanked Eric again for his help, indicating that he would keep working on his project and add the par feature in the next version.

Next Steps

These case studies from the first 3 years of the Scratch online community illustrate some of the ways in which an online community supports young people's development of interactive media across the design process. They also provoke questions about the implications for other learning environments. In all of the design process stages, there is interplay between community *artifacts* and community *members*. For *imagining*, both people and projects serve as sources of inspiration, highlighting what might be possible to design and ways of being a designer. Future research might explore, *how imagination is ignited (or limited) by examples that we make available to young people*. For *creating*, the online community offers a library of projects to learn from and remix. There are also people who can serve as guides and collaborators, enabling a Scratcher to be involved in the design of artifacts that they would not have been able to develop on their own. Future research for this design process might explore, *how we can rethink what it means to create, moving away from individual-centric and instruction-centric approaches to learning*. For *playing*, the online community enables multiple Scratchers to be involved in testing and

experimenting with iterations of a project or for an individual Scratcher to engage in multiple iterations across contexts. In this area, future research might investigate, *how we can increasingly focus on the processes of design and learning, rather than the products*. For *sharing*, having such a large and appreciative audience for their projects (and sometimes for themselves as designers) is highly motivating to many Scratchers, even if a community the size of Scratch seems unimaginably large to most community members. For the process of sharing, we might ask, *how we can find ways of connecting young people to authentic and peer audiences*. Finally, for *reflecting*, community documentation supports critical self-examination, which is further supported by the active conversations that take place among Scratchers. In this area, future research might examine *how we can create opportunities for young people to do and to think about the doing*.

With Scratch, hundreds of thousands of young people are creating their own interactive media and participating as designers. Moreover, the online community supports them as they participate in project design. However, while many young people thrive in the self-directed environment of the online community, others find the space difficult to navigate. To facilitate participation, we have developed other forms of scaffolding and support, including tutorials, curated collections of projects to highlight specific computational concepts and practices, and design challenges and activities to encourage new computational explorations.

Although these resources have contributed to supporting young people as designers, work remains in making design experiences available to broader audiences of young people. Many of the early adopters of Scratch have been young people from homes with technology advocates: parents who are computer programmers, siblings who enjoy tinkering with programming tools, and aunts or uncles who are engineers. Regardless of our efforts with Scratch, these young people are certain to have many opportunities for positive technology experiences.

Given that the ability to understand and negotiate technological artifacts is becoming increasingly important in the lives of young people, the ability to design technology is not a luxury that should be reserved for a select few who have access to support at home. As a society, we need *all* young people to be able to solve open-ended problems and to be self-regulating, passionate learners—the very qualities that young people develop while engaging with Scratch and iterative design processes, qualities that we hope young people will develop in school settings.

We see schools as a critical venue for broadening participation in design activities, reaching young people who might not have this support at home, and giving young people additional opportunities to engage in the iterative design processes necessary to fully participate in society. To this end, we have been working with teachers to support their understandings of Scratch, iterative design processes, and ways of including design in teaching practices across age ranges, from elementary to college, and across the curriculum, from art to science to languages to social studies.

Our approach to supporting teachers mirrors our support of young Scratchers. Just as we see young people as designers, we see teachers as designers—not of interactive media, primarily, but of learning environments. As designers, teachers

can similarly benefit from an online community in which their processes of designing—the *imagining, creating, playing, sharing, and reflecting* of learning environments—are enhanced through interactions with artifacts and others. To this end, we developed a companion community to the main Scratch online community called ScratchEd (<http://scratched.media.mit.edu>) where Scratch educators can read and share stories, exchange and provide feedback on resources, ask and answer questions, and find each other based on geography or interests. Launched in summer of 2009, the Scratch educator community has grown to more than 4,600 educators in its first 2 years, and we have already seen benefits to teachers' design processes that parallel the benefits we have documented in young people's design processes.

Design experiences are not predetermined. The path that a designer will follow is uncertain and can lead to unexpected challenges. Whether a young person designing his/her own interactive media with Scratch or an educator designing learning environments, designers of all ages and backgrounds can find support for their learning experiences in contexts where they have access to others. An online community affords opportunities for designers to interact with new artifacts and new people, which provide support across the design cycle. Imagining, creating, playing, sharing, and reflecting are all enhanced through interactions with the community.

References

- Barron, B. J. S., Schwartz, D. L., Vye, N. J., Moore, A., Petrosino, A., Zech, L., et al. (1998). Doing with understanding: Lessons from research on problem- and project-based learning. *The Journal of the Learning Sciences*, 7, 271–310. doi:10.1080/10508406.1998.9672056.
- Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, 18(1), 32–42. doi:10.3102/0013189X018001032.
- Bruckman, A. (1998). Community support for constructionist learning. *Computer Supported Cooperative Work*, 7, 47–86. doi:10.1023/A:1008684120893.
- Bruckman, A. (2006). Learning in online communities. In K. Sawyer (Ed.), *Cambridge handbook of the learning sciences* (pp. 461–472). New York, NY: Cambridge University Press.
- Buckingham, D., & Willett, R. (Eds.). (2006). *Digital generations: Children, young people, and new media*. London, UK: Routledge.
- Csikszentmihalyi, M. (1997). *Creativity: Flow and the psychology of discovery and invention*. New York, NY: Harper Perennial.
- Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments*, 1, 1–32. doi:10.1080/1049482900010102.
- Ito, M. (2009). *Engineering play: A cultural history of children's software*. Cambridge, MA: MIT Press.
- Ito, M., Baumer, S., Bittanti, M., Boyd, D., Cody, R., Herr-Stephenson, B., et al. (2009). *Hanging out, messing around, geeking out: Living and learning with new media*. Cambridge, MA: MIT Press.
- Jenkins, H., Purushotma, R., Weigel, M., Clinton, K., & Robison, A. (2006). *Confronting the challenges of participatory culture: Media education for the 21st century*. Cambridge, MA: MIT Press.
- Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning*. Hillsdale, NJ: Erlbaum.

- Kafai, Y. B., & Resnick, M. (Eds.). (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Hillsdale, NJ: Erlbaum.
- Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., et al. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by Design™ into practice. *The Journal of the Learning Sciences*, 12, 495–548. doi:[10.1207/S15327809JLS1204_2](https://doi.org/10.1207/S15327809JLS1204_2).
- Krajcik, J. S., & Blumenfeld, P. C. (2006). Project-based learning. In K. Sawyer (Ed.), *Cambridge handbook of the learning sciences* (pp. 317–334). New York, NY: Cambridge University Press.
- Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. New York, NY: Cambridge University Press.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.
- Perkins, D. N. (1986). *Knowledge as design*. Hillsdale, NJ: Erlbaum.
- Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In *Proceedings of the 6th creativity & cognition conference* (pp. 1–6). New York, NY: Association for Computing Machinery. doi: [10.1145/1254960.1254961](https://doi.org/10.1145/1254960.1254961)
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67. doi:[10.1145/1592761.1592779](https://doi.org/10.1145/1592761.1592779).
- Rogoff, B. (1994). Developing understanding of the idea of communities of learners. *Mind Culture and Activity*, 1, 209–229. doi:[10.1080/10749039409524673](https://doi.org/10.1080/10749039409524673).
- Salen, K. (2007). *The ecology of games: Connecting youth, games, and learning*. Cambridge, MA: MIT Press.
- Sawyer, K. (2006a). *Explaining creativity: The science of human innovation*. New York, NY: Oxford University Press.
- Sawyer, K. (2006b). The new science of learning. In K. Sawyer (Ed.), *Cambridge handbook of the learning sciences* (pp. 1–16). New York, NY: Cambridge University Press.