brock

international prize in education

2015 Brock International
Prize in Education Nominee

Mitchel Resnick

Nominated by Tina S. Ornduff

September 1, 2014

Dear Brock Prize Jurors,

I am privileged to nominate Dr. Mitchel Resnick for this year's Brock International Prize in Education. I believe strongly that he is an exceptional Brock Laureate candidate and that the following summary of his outstanding accomplishments and contributions in education support his nomination.

As the LEGO Papert Professor of Learning Research and head of the Lifelong Kindergarten group at the MIT Media Lab, Mitchel Resnick leads this group as they explores how new technologies can engage students in creative learning experiences. The group is called Lifelong Kindergarten because they are inspired by the way children learn in kindergarten. In the classic kindergarten, children are constantly designing and creating things in collaboration with one another. Dr. Resnick is passionate about finding ways to continue and extend that kindergarten approach and experience to learners of all ages.

Dr. Resnick's research group also developed the "programmable brick" technology that inspired the LEGO Mindstorms robotics kit. He co-founded the Computer Clubhouse project, a worldwide network of after-school centers where youth from low-income communities learn to express themselves creatively with new technologies. Dr. Resnick's group developed Scratch, an online community where children program and share interactive stories, games, and animations.

Because of Dr. Resnick's unique and pioneering work in Computer Science education, students regardless of age, background, or interest have had the opportunity to try computer programming or coding. He believes that the ability to write code allows people to *write* new types of things – interactive stories, games, animations, and simulations. He also sees much deeper and broader reasons for learning to code. In the process of learning to code, people learn many other things. They are not just learning to code, they are coding to learn. In addition to learning mathematical and computational ideas, they are also learning strategies for solving problems, designing projects, and communicating ideas. Dr. Resnick believes that these skills are useful not just for computer scientists but for everyone.

Dr. Resnick earned a BA in physics at Princeton University (1978), and MS and PhD degrees in computer science at MIT (1988, 1992). He worked as a science-technology journalist from 1978 to 1983, and he has consulted throughout the world on creative uses of computers in education. He is author of *Turtles, Termites, and Traffic Jams* (1994), co-editor of *Constructionism in Practice* (1996), and co-author of *Adventures in Modeling* (2001). In 2011, Dr. Resnick was awarded the McGraw Prize in Education and was listed by Fast Company as one of the 100 Most Creative People in Business.

I strongly recommend Dr. Mitchel Resnick for the Brock Prize in International Education. His work is innovative, impactful, and inspiring. It is exactly what education needs if education is to keep up with a world that is changing more rapidly than ever before. His vision for using technology in education has helped students become creators, problem solvers, and critical thinkers. Isn't that what education is meant to do?

Sincerely,

Tina S. Ornduff

curriculum vitae

Mitchel Resnick
MIT Media Lab (E14-445A)
75 Amherst Street
Cambridge, MA 02142
617-253-9783
mres at media.mit.edu

Education

Massachusetts Institute of Technology
Masters (1988) and PhD (1992) in Computer Science
Princeton University
B.A. in Physics, 1978 (Phi Beta Kappa, magna cum laude)

Work Experience

MIT Media Laboratory
Assistant Professor, 1992-1996
Associate Professor (without tenure), 1996-2000
Associate Professor (with tenure), 2000-2006
Full Professor, 2006-present
Academic Head, Media Arts & Sciences Program, 2005-present
Director, Lifelong Kindergarten research group, 1992-present

Playful Invention Company
Co-founder, Chairman, 2003-2010
Boston Museum of Science
Board of Trustees, 1999-2008
Board of Overseers, 2008-present

Business Week Magazine
Science-technology correspondent, 1978-1983

Honors and Awards

- McGraw Prize in Education, 2011
- World Technology Award (category: education), 2011
- Kids@Play Award: Digital Pioneer for Kids, 2010
- Eliot Pearson Award for Excellence in Children's Media, 2008
- LEGO Papert Professor for Learning Research, 1998-present
- Fukutake Career Development Chair, 1995-1998
- National Science Foundation Young Investigator Award, 1993-1998
- Vannevar Bush Fellowship for Science Journalism, 1983-84

Projects

Scratch: Democratizing Digital Expression
With Scratch software, kids can create their own interactive stories, games, and animations - and share their creations online. Kids learn to think creatively, reason systematically, and work collaboratively, while learning important computational ideas. See Dr. Resnick's TED talk about learning to code (and coding to learn) with Scratch.

Programmable Bricks: Learning through Designing
With Programmable Bricks, children can build and program their own robots, kinetic sculptures, and other interactive inventions - and learn science and engineering concepts in the process. Programmable Bricks served as inspiration for the LEGO MindStorms, PicoCricket, and LEGO WeDo robotics kits.

Computer Clubhouse: Broadening Participation
Dr. Resnick co-founded the Computer Clubhouse project, an international network of 100 after-school centers where youth from low-income communities learn to express themselves creatively with new technologies.

Publications

Resnick, M. (2013). Learn to Code, Code to Learn. EdSurge, May 2013 [Spanish version].

Resnick, M. (2013). Lifelong Kindergarten. *Cultures of Creativity*. LEGO Foundation.
Resnick, M., & Rosenbaum, E. (2013). Designing for Tinkerability. In Honey, M., & Kanter, D. (eds.), *Design, Make, Play: Growing the Next Generation of STEM Innovators*, pp. 163-181. Routledge.

Flannery, L., Kazakoff, E., Bonta, P., Silverman, B., Bers, M., & Resnick, M. (2013). Designing ScratchJr: Support for Early Childhood Learning Through Computer Programming. Proceedings of the 2013 Interaction Design and Children (IDC) Conference. New York.

Resnick, M. (2012). Mother's Day, Warrior Cats, and Digital Fluency: Stories from the Scratch Online Community. Proceedings of the Constructionism 2012 conference. Athens, Greece.

Resnick, M. (2012). Reviving Papert's Dream. *Educational Technology*, vol. 52, no. 4, pp. 42-46.

Brennan, K., & Resnick, M. (2012). New Frameworks for Studying and Assessing the Development of Computational Thinking. Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.

Resnick, M. (2012). Still a Badge Skeptic. Digital Media and Learning blog.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education (TOCE)*, vol. 10, no. 4 (November 2010).

Brennan, K., Resnick, M., & Monroy-Hernandez, A. (2010). Making projects, making friends: Online community as catalyst for interactive media creation. *New Directions for Youth Development*, 2010(128), 75-83.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, vol. 52, no. 11, pp. 60-67 (Nov. 2009). [PDF][HTML]

Resnick, M. (2009). Kindergarten is the Model for Lifelong Learning. *Edutopia*, June 2009.

Rusk, N., Resnick, M., & Cooke, S. (2009). Origins and Guiding Principles of the Computer Clubhouse. In Kafai, Y., Peppler, K., and Chapman, R. (eds.), *The Computer Clubhouse: Constructionism and Creativity in Youth Communities*. Teachers College Press.

Resnick, M., Flanagan, M., Kelleher, C., MacLaurin, M., Ohshima, Y., Perlin, K., & Torres, R. (2009). Growing Up Programming: Democratizing the Creation of Dynamic Interactive Media.*Proceedings of the CHI (Computer-Human Interaction) '09 conference*. Boston.

Resnick, M. (2008). Falling in Love with Seymour's Ideas. American Educational Research Associaion (AERA) annual conference, New York.

Monroy-Hernandez, A., & Resnick, M. (2008). Empowering Kids to Create and Share Programmable Media. *Interactions*, vol. 15, no. 2, pp. 50-53.

Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. (2008). New Pathways into Robotics: Strategies for Broadening Participation. *Journal of Science Education and Technology*, vol. 17, no. 1, pp. 59-69.

Maloney, J., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by Choice: Urban Youth Learning Programming with Scratch. SIGCSE conference, Portland, March 2008.

Resnick, M. (2007). Sowing the Seeds for a More Creative Society. *Learning and Leading with Technology*, December 2007 [Spanish version].

Resnick, M. (2007). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. ACM Creativity & Cognition conference, Washington DC, June 2007.

Resnick, M. (2007). Learning from Scratch, Microsoft Faculty Connection, June 2007.

Resnick, M. (2006). Computer as Paintbrush: Technology, Play, and the Creative Society. In

Singer, D., Golikoff, R., and Hirsh-Pasek, K. (eds.), *Play = Learning: How play motivates and enhances children's cognitive and social-emotional growth*. Oxford University Press.

Resnick, M., and Silverman, B. (2005). Some Reflections on Designing Construction Kits for Kids.*Proceedings of Interaction Design and Children conference*, Boulder, CO.
Resnick, M. (2004). Computers and Mud. *Digital Dialogues: Technology and the Hand*. Haystack Monograph Series.

Resnick, M. (2004). Edutainment? No Thanks. I Prefer Playful Learning. *Associazione Civita Report on Edutainment*. (Also appeared on Parents' Choice website.)

Resnick, M. (2003). Thinking Like a Tree (and Other Forms of Ecological Thinking). *International Journal of Computers for Mathematical Learning*, vol. 8, no. 1, pp. 43-62. [Also available in Russian.]

Zuckerman, O., and Resnick, M. (2003). System Blocks: A Physical Interface for System Dynamics Learning. *International System Dynamics Conference*, New York.

Zuckerman, O., and Resnick, M. (2003). System Blocks: A Physical Interface for System Dynamics Simulation. *Proc. of CHI (Computer-Human Interaction)'03 Conference*, pp. 810-811.

Resnick, M. (2003). Playful Learning and Creative Societies. *Education Update*, vol. VIII, no. 6, February 2003.

Resnick, M. (2002). Rethinking Learning in the Digital Age. In *The Global Information Technology Report: Readiness for the Networked World*, edited by G. Kirkman. Oxford University Press.

Resnick, M. (2001). Closing the Fluency Gap. *Communications of the ACM*, vol. 44, no. 3 (March 2001). [Also available in Spanish.]

Resnick, M., et al. (2000). The PIE Network: Promoting Science Inquiry and Engineering through Playful Invention and Exploration with New Digital Technologies. Proposal to the National Science Foundation (project funded 2001-2004).

Resnick, M., Berg, R., and Eisenberg, M. (2000). Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Investigation. *Journal of the Learning Sciences*, vol. 9, no. 1, pp. 7-30.

Resnick, M. (2000). It's Not Just Information. *IBM Systems Journal*, vol. 39, nos. 3 & 4.
Shaffer, D.W., and Resnick, M. (1999). "Thick" Authenticity: New Media and Authentic Learning.*Journal of Interactive Learning Research*, vol. 10, no. 2, pp. 195-215.

Soloway, E., Grant, W., Tinker, R., Roschelle, J., Mills, M., Resnick, M., Berg, R., and Eisenberg, M. (1999). Science in the Palms of Their Hands. *Communications of the ACM*, vol. 42, no. 8, pp. 21-26. August 1999.

Wilensky, U., and Resnick, M. (1999). Thinking in Levels: A Dynamic Systems Approach to Making Sense of the World. *Journal of Science Education and Technology*, vol. 8, no. 1, pp. 3-19.

Resnick, M. (1999). Unblocking the Traffic Jams in Corporate Thinking. Focus E-zine, Cap Gemini Ernst & Young.

Resnick, M. (1998). Technologies for Lifelong Kindergarten. *Educational Technology Research and Development*, vol. 46, no. 4.

Resnick, M., Rusk, N., and Cooke, S. (1998). The Computer Clubhouse: Technological Fluency in the Inner City. In Schon, D., Sanyal, B., and Mitchell, W. (eds.), *High Technology and Low-Income Communities*, pp. 266-286. Cambridge: MIT Press.

Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., and Silverman, B. (1998).Digital Manipulatives. *Proceedings of the CHI '98 conference*, Los Angeles, April 1998.

Borovoy, R., Martin, F., Vemuri, S., Resnick, M., Silverman, B., and Hancock, C. (1998). Meme Tags and Community Mirrors: Moving from Conferences to Collaboration. Computer Supported Collaborative Work conference, Seattle, pp. 159-168.

Negroponte, N., Resnick, M., and Cassell, J. (1997). Creating a Learning Revolution.

Resnick, M., and Wilensky, U. (1997). Diving into Complexity: Developing Probabilistic Decentralized Thinking through Role-Playing Activities. *Journal of the Learning Sciences*, vol. 7, no. 2, pp. 153-172.

Resnick, M., Bruckman, A., and Martin, F. (1996). Pianos Not Stereos: Creating Computational Construction Kits. *Interactions*, vol. 3, no. 6 (September/October 1996).

Resnick, M., Berg, R., Eisenberg, M., Turkle, S., and Martin, F. (1996). Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Instruments. Proposal to the National Science Foundation (project funded 1997-1999).

Resnick, M., Martin, F., Sargent, R., and Silverman, B. (1996). Programmable Bricks: Toys to Think With. *IBM Systems Journal*, vol. 35, no. 3-4, pp. 443-452.

Resnick, M., and Rusk, N. (1996). The Computer Clubhouse: Preparing for Life in a Digital World.*IBM Systems Journal*, vol. 35, no. 3-4, pp. 431-440.

Borovoy, R., McDonald, M., Martin, F., and Resnick, M. (1996). Things that blink: Computationally augmented name tags. *IBM Systems Journal*, vol. 35, no. 3-4, pp. 488-495.

Resnick, M. (1996). Distributed Constructionism. *Proceedings of the International Conference of the Learning Sciences*, Northwestern University.

Resnick, M. (1996). Beyond the Centralized Mindset. *Journal of the Learning Sciences*, vol. 5, no. 1, pp. 1-22.

Resnick, M., and Silverman, B. (1996). Exploring Emergence. An "active essay" on the Web.

Resnick, M. (1995). New Paradigms for Computing, New Paradigms for Thinking. *Computers and Exploratory Learning*, A. diSessa, C. Hoyles, & R. Noss (eds.), pp. 31-43. Berlin: Springer-Verlag.

Bruckman, A., and Resnick, M. (1995). The MediaMOO Project: Constructionism and Professional Community. *Convergence*, vol. 1, no. 1, pp. 94-109.

Resnick, M. (1994). Learning About Life. *Artificial Life Journal*, vol. 1, no. 1-2, pp. 229-241.

Resnick, M. (1994). Changing the Centralized Mind. *Technology Review*, pp. 32-40 (July 1994).

Resnick, M. (1993). Behavior Construction Kits [pdf]. *Communications of the ACM*, vol. 36, no. 7, pp. 64-71 (July 1993).

Resnick, M. (1991). MultiLogo: A Study of Children and Concurrent Programming. *Interactive Learning Environments*, vol. 1, no. 3, pp. 153-170.

Resnick, M. (1991). Xylophones, Hamsters, and Fireworks: The Role of Diversity in Constructionist Activities. In *Constructionism*, edited by I. Harel & S. Papert. Ablex Publishing.

Hogg, D., Martin, F., and Resnick, M. (1991). Braitenberg Creatures. Epistemology and Learning Memo 13, MIT Media Lab.

Resnick, M., and Ocko, S. (1991). LEGO/Logo: Learning Through and About Design. In*Constructionism*, edited by I. Harel & S. Papert. Ablex Publishing.

Eisenberg, M., Resnick, M., and Turback, F. (1987). Understanding Procedures as Objects. *Empirical Studies of Programmers*, G. Olson, S. Sheppard, & E. Soloway (eds.), pp. 14-32, Ablex Publishing.

Books

Resnick, M. (1994). *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*. Cambridge, MA: MIT Press.

Kafai, Y., and Resnick, M., eds. (1996). *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*. Mahwah, NJ: Lawrence Erlbaum.

Colella, V., Klopfer, E., and Resnick, M. (2001). *Adventures in Modeling: Exploring Complex, Dynamic Systems with StarLogo*. New York: Teachers College Press.

Proposals

Coding for All: Interest-Driven Trajectories to Computational Fluency. Proposal to the National Science Foundation, 2013 (modified version funded 2014-2016).

ScratchJr: Computer programming in early childhood education as a pathway to academic readiness and success. Proposal to the National Science Foundation, 2011 (project funded 2011-2014).

Preparing the Next Generation of Computational Thinkers: Transforming Learning and Education Through Cooperation in Decentralized Networks. Proposal to the National Science Foundation, 2010 (project funded 2010-2014).

ScratchEd: Working with teachers to develop design-based approaches to the cultivation of computational thinking. Proposal to the National Science Foundation, 2010 (project funded 2010-2013).

Scratch 2.0: Cultivating Creativity and Collaboration in the Cloud. Proposal to the National Science Foundation, 2010 (project funded 2010-2013).
Making Programming Universally Accessible and Useful. Proposal to Google, 2009 (project funded 2009-2010).

Rethinking Robotics: Engaging Girls in Creative Engineering. Proposal to the National Science Foundation, 2005 (not funded).

A Networked, Media-Rich Programming Environment to Enhance Technological Fluency at After-School Centers in Economically-Disadvantaged Communities. Proposal to the National Science Foundation, 2003 (project funded 2003-2007).

The PIE Network: Promoting Science Inquiry and Engineering through Playful Invention and Exploration with New Digital Technologies. Proposal to the National Science Foundation, 2000 (project funded 2001-2005).

Learning with Digital Manipulatives: New Frameworks to Help Elementary-School Students Explore "Advanced" Mathematical and Scientific Concepts. Proposal to the National Science Foundation, 2000 (not funded).

Rethinking Learning in the Digital Age

Mitchel Resnick

The Media Laboratory
Massachusetts Institute of Technology

First, the good news: in the years ahead, the declining cost of computation will make digital technologies accessible to nearly everyone in all parts of the world, from inner-city neighborhoods in the United States to rural villages in developing nations. These new technologies have the potential to fundamentally transform how and what people learn throughout their lives. Just as advances in biotechnologies made possible the "green revolution" in agriculture, new digital technologies make possible a "learning revolution" in education.

Now, the bad news: while new digital technologies make a learning revolution possible, they certainly do not guarantee it. Early results are not encouraging. In most places where new technologies are being used in education today, the technologies are used simply to reinforce outmoded approaches to learning. Even as scientific and technological advances are transforming agriculture, medicine, and industry, ideas about and approaches to teaching and learning remain largely unchanged.

To take full advantage of new technologies, we need to fundamentally rethink our approaches to learning and education—and our ideas of how new technologies can support them.

Beyond Information

When people think about education and learning, they often think about information. They ask questions like: What information is most important for people to know? What are the best ways to transmit that information from one person (a teacher) to another (a learner)? What are the best ways to represent and display information so that it is both understandable and learnable?

It's not surprising that people see a natural connection between computers and education. Computers enable people to transmit, access, represent, and manipulate information in many new ways. Because education is associated with information and computers are associated with information, the two seem to make a perfect marriage.

This focus on information, however, is limiting and distorting, both for the field of education and for computers. If we want to take full advantage of new computational technologies, and if we want to help people become better thinkers and learners, we need to move beyond these information-centric views of computing and learning.

Over the past fifty years, psychologists and educational researchers, building on the pioneering work of Jean Piaget, have come to understand that learning is not a simple matter of information transmission. Teachers cannot simply pour information into the heads of learners; rather, learning is an active process in which people construct new understandings of the world around them through active exploration, experimentation, discussion, and reflection. In short: people don't *get* ideas; they *make* them.

As for computers, they are more than simply information machines, despite the common use of the phrase "information technology" or "IT." Of course, computers are wonderful for transmitting and accessing information, but they are, more broadly, a new medium through which people can create and express. If we use computers simply to deliver information to students, we are missing the revolutionary potential of the new technology for transforming learning and education.

Consider the following three things: computers, television, finger paint. Which of the three doesn't belong? For most people, the answer seems obvious: "finger paint" doesn't fit. After all, computers and televisions were both invented in the twentieth century, both involve electronic technology, and both can deliver information to large numbers of people. None of that is true for finger paint.

But until we start to think of computers more like finger paint and less like television, computers will not live up to their full potential. Like finger paint (and unlike television), computers can be used for designing and creating things. In addition to accessing Web pages, people can create their own Web pages. In addition to downloading MP3 music files, people can compose their own music. In addition to playing SimCity, people can create their own simulated worlds.

It is design activities such as these that offer the greatest new learning opportunities with computers. Research has shown that many of our best learning experiences come when we are engaged in designing and creating things, especially things that are meaningful either to us or others around us (e.g., Papert 1993). When children create pictures with finger paint, for example, they learn how colors mix together. When they build houses and castles with building blocks, they learn about structures and stability. When they make bracelets with colored beads, they learn about symmetries and patterns.

Like finger paint, blocks, and beads, computers can also be used as a "material" for making things—and not just by children, but by everyone. Indeed, the computer is the most extraordinary construction material ever invented, enabling people to create anything from music videos to scientific simulations to robotic creatures. Computers can be seen as a universal construction material, greatly expanding what people can create and what they can learn in the process (Resnick 1998).

Digital Fluency

Unfortunately, most people don't use computers that way today. When people are introduced to computers today, they are typically taught how to look up information on the Web, how to use a word processor, how to send e-mail. But they don't become *fluent* with the technology.

What does it mean to be digitally fluent? Consider the analogy with learning a foreign language. If someone learned a few phrases so that they could read menus in restaurants and ask for directions on the street, would you consider them fluent in the language? Certainly not. That type of phrase-book knowledge is equivalent to the way most people use computers today. Is such knowledge useful? Yes. But it is not fluency.

To be truly fluent in a foreign language, you must be able to articulate a complex idea or tell an engaging story; in other words, you must be able to "make things" with language. Analogously, being digitally fluent involves not only knowing how to use technological tools, but also knowing how to construct things of significance with those tools (Papert and Resnick 1995).

Fluency with language not only has great utilitarian value in everyday life but also has a catalytic effect on learning. When you learn to read and write, you are in a better position to learn many other things. So, too, with digital fluency. In the years ahead, digital fluency will become a prerequisite for obtaining jobs, participating meaningfully in society, and learning throughout a lifetime.

Today, discussions about the "digital divide" typically focus on differences in access to computers. That will change. As the costs of computing decline, people everywhere will gain better access to digital technologies. But there is a real risk that only a small handful will be able to use the technologies fluently. In short, the "access gap" will shrink, but a serious "fluency gap" could remain.

Computer Clubhouses

To provide more young people with the opportunity to become digitally fluent, the Massachusetts Institute of Technology (MIT) Media Lab and the Boston Museum of Science have established a network of learning centers in economically disadvantaged communities. At these centers, called Computer Clubhouses, young people become designers and creators with new digital technologies. Clubhouse members use leading-edge software to create their own artwork, animations, simulations, multimedia presentations, musical compositions, websites, and robotic constructions (Resnick et al. 1998).

The first Computer Clubhouse opened in 1993 in Boston, serving youth between the ages of ten and eighteen. Based on the success of the initial Clubhouse, a dozen more communities opened Computer Clubhouses over the next six years. Then, in 2000, Intel announced that it would provide support to open an additional hundred Computer Clubhouses around the world over the ensuing five years. There are now Clubhouses in India, Ireland, Israel, Colombia, Germany, the Philippines, and the United States, with new Clubhouses planned for 2002 in China, Costa Rica, Mexico, South Africa, and Taiwan.

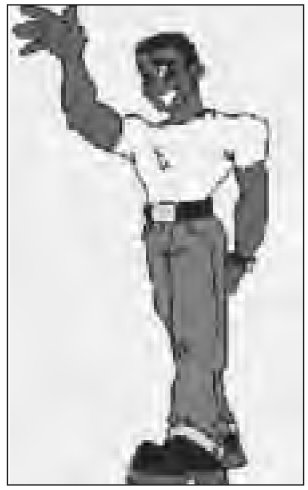
Computer Clubhouses are very different from most telecenters and community technology centers, which typically fall in one of two categories. Some technology centers merely provide access. People can do whatever they want: play games, surf the Web, use online chat rooms. Other centers offer structured courses teaching basic computer skills (such as keyboarding) and basic applications (such as word processing and spreadsheets).

Computer Clubhouses offer a third path, with different goals and a different approach. The aim is not simply to teach basic skills, but to help young people learn to express themselves and gain confidence in themselves as learners. If they are interested in video games, they don't come to the Clubhouse to play games; they come to create their own games. They don't download videos from the Web; they create their own videos. In the process, youth learn the heuristics of being a good designer: how to conceptualize a project, how to make use of the materials available, how to persist and find alternatives when things go wrong, how to collaborate with others, and how to view a project through the eyes of others. In short, they learn how to manage a complex project from start to finish.

The Computer Clubhouse approach strikes a balance between structure and freedom in the learning process. As Clubhouse youth work on projects based on their own interests, they receive a great deal of support from other members of the Clubhouse community (e.g., staff members, volunteer mentors, and other Clubhouse youth). There is a large collection of sample projects on the walls, shelves, and hard drives of the Clubhouses; these provide Clubhouse youth with a sense of the possible, and multiple entry points through which they can start. The goal is to provide enough freedom to enable Clubhouse youth to follow their fantasies, but also enough support to help them turn those fantasies into realities.

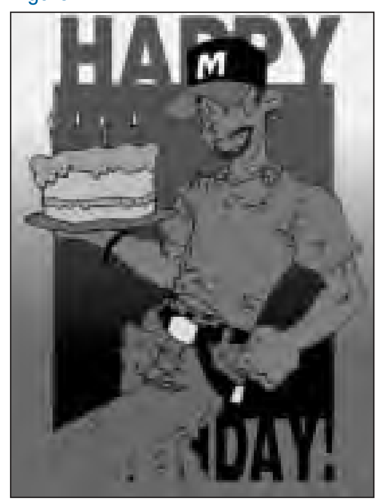
There is no doubt that the lives of many Computer Clubhouse members have been transformed by their time at the Clubhouses. Consider Mike Lee, who spent time at the original Computer Clubhouse in Boston. Mike first came to the Clubhouse after he had dropped out of high school. His true passion was drawing. He filled up notebook after notebook with sketches of cartoon characters. At the Clubhouse, Mike Lee developed a new method for his artwork. First, he would draw black-and-white sketches by hand. Then, he would scan the sketches into the computer and use the computer to color them in. His work often involved comic-book images of himself and his friends (Figure 1).

Figure 1



Over time, Mike learned to use more advanced computer techniques in his artwork (Figure 2). Everyone in the Clubhouse was impressed with Mike's creations, and other youth began to come to him for advice. Some members explicitly mimicked Mike's artistic style. Before long, a collection of "Mike Lee style" artwork filled the bulletin boards of the Clubhouse (Figure 3). "It's kind of flattering," says Mike.

Figure 2



For the first time in Mike's life, other people were looking up to him. He began to feel a new sense of responsibility. He decided to stop using guns in his artwork, feeling that it was a bad influence on the younger Clubhouse members. "My own personal artwork is more hard core, about street violence. I had a close friend who was shot and died," Mike explains. "But I don't want to bring that here. I have an extra responsibility. Kids don't understand about guns; they think it's cool. They see a fight, it's natural they want to go see it. They don't understand. They're just kids."

Figure 3

Mike Lee began working with others at the Clubhouse on collaborative projects. Together, they created an online art gallery. Once a week, they met with a local artist who agreed to be a mentor for the project. After a year, their online art show was accepted as an exhibition at Siggraph, the world's premiere computer-graphics conference.

Figure 4

As Mike worked with others at the Clubhouse, he began to experiment with new artistic techniques. He added more computer effects, and he began working on digital collages combining photographs and graphics, while still maintaining his distinctive style (Figure 4). Over time, Mike explored how he might use his artwork as a form of social commentary and political expression (Figure 5).

As he worked at the Clubhouse, Mike Lee clearly learned a lot about computers and about graphic design. But he also began to develop his own ideas about teaching and learning. "At the Clubhouse, I was free to do what I wanted, learn what I wanted," says Mike. "Whatever I did was just for me. If I had taken computer courses [in school], there would have been all those assignments. Here I could be totally creative." Mike remembers—and appreciates—how the staff members treated him when he first started at the Clubhouse. They asked him to design the sign for the entrance to the Clubhouse, and looked to him as a resource. They never thought of him as a "high-school dropout" but as an artist.

Mike's artwork still has the same distinctive style, but he has become more fluent in expressing himself in computer-based media. Describing his current work, Mike talks about "dither nightmares" and "anti-aliasing problems"—ideas that would have been alien to him a few years ago. He says his artwork is "ten times better than last year."

Rethinking Technologies

In addition to rethinking our approaches to learning and education, we also need to rethink the technologies that we provide to young people.

Most of today's computers were designed primarily for use by adults in the workplace. We need to develop a new generation of computer technologies *worthy* of the next generation of children. It's not enough just to make computers faster; we need to develop new types of computers. Today's youth are ready and eager to do more with computers. We need to provide the hardware and software that will enable them to do so.

These new technologies might look very different from traditional computers. For example, my research group has developed a family of "programmable bricks": tiny computers embedded inside children's building blocks (Martin et al. 2000; Resnick et al. 1996). With these bricks, children can build computational power directly into their physical-world constructions, using the

Figure 5

programmable bricks to control motors, receive information from sensors, and even communicate with one another. The LEGO Company now sells a commercial version of these programmable bricks, under the name LEGO MindStorms.

Children have used our programmable bricks to build a variety of creative constructions, including an odometer for rollerblades (using a magnetic sensor to count wheel rotations); a diary-security system (using a touch sensor to detect if anyone tried to open the diary); and an automated hamster cage (using a light sensor to monitor the hamster's movements).

One 11-year-old girl, named Jenny, was very interested in birds, and she decided to use programmable bricks to build a new type of bird feeder. She started by making a wooden lever that served as a perch for the birds. When a bird landed, it would trigger a touch sensor, sending a signal to a programmable brick, which turned on a LEGO mechanism, which pushed down the shutter of a camera, taking a picture of the bird.

The design-oriented nature of the project was clearly very important for Jenny. As she described it: "The fun part is knowing that *you* made it; *my* machine can take pictures of birds." At the same time, the project served as a rich context for engaging in scientific inquiry and learning science-related concepts. Jenny developed a deeper understanding of some concepts (such as mechanical advantage) that she had previously studied in school but had never really appreciated. She also began to work with some engineering concepts (related to feedback and control) that are traditionally taught only at the university level (Resnick et al. 2000).

Programmable bricks provided Jenny with "design leverage," enabling her to create things that would have been difficult for her to create in the past. At the same time, the bricks provided Jenny with "conceptual leverage," enabling her to learn concepts that would have been difficult for her to learn in the past.

Reforming Educational Reform

Increasingly, nations are recognizing that improving education is the best way to increase wealth, enhance health, and maintain peace. But there is little consensus on how to achieve an educated population, or even on what it means to have an educated population. Can progress towards an educated population be measured by counting the number of people in school? By the number of years they spend in school? By assessing their grades on standardized tests?

Every country in the world, it seems, has a plan for educational reform. But, in most cases, reform initiatives are superficial and incremental, and do not get at the heart of the problem. These initiatives often introduce new forms of testing and assessment, but leave in place (or make only small incremental changes to) existing curricula and existing teaching strategies. We need to reform educational reform.

Rethink how people learn. We need to fundamentally reorganize school classrooms. Instead of a centralized-control model (with a teacher delivering information to a roomful of students), we should take a more entrepreneurial approach to learning. Students can become more active and independent learners, with the teacher serving as consultant, not chief executive. Instead of dividing up the curriculum into separate disciplines (math, science, social studies, language), we should focus on themes and projects that cut across the disciplines, taking advantage of the rich connections among different domains of knowledge. Instead of dividing students according to age, we should encourage students of all ages to work together on projects, enabling them to learn from one another (and to learn by teaching one another). Instead of dividing the school day into hour-long slices, we should let students work on projects for extended periods of time, enabling them to follow through more deeply and meaningfully on the ideas that arise in the course of their work.

Rethink what people learn. Much of what children learn in schools today was designed for the era of paper-and-pencil. We need to update curricula for the digital age. One reason is obvious: Schools must prepare students with the new skills and ideas that are needed for living and working in a digital society. There is a second, subtler reason: new technologies are changing not only what students *should* learn, but also what they *can* learn. There are many ideas and topics that have always been important but were left out of traditional school curricula because they were too difficult to teach and learn with only paper, pencil, books, and blackboard. Some of these ideas are now accessible through creative use of new digital technologies. For example, children can now use computer simulations to explore the workings of systems in the world (everything from ecosystems to economic systems to immune systems) in ways that were previously not possible. Some ideas that were previously introduced only at the university level can and should be learned much earlier. Finally, and perhaps most importantly, we need to transform curricula so that they focus less on "things to know" and more on "strategies for learning the things you don't know." As new technologies continue to quicken the pace of change in all parts of our lives, learning to become a better learner is far more important than learning to multiply fractions or memorizing the capitals of the world.

Rethink where and when people learn. Most education-reform initiatives appear to assume that learning takes place only between the ages of 6 and 18, between 8:00 A.M. and 3:00 P.M.—that is, when children are in schools. But schools are just part of a broader learning ecosystem. In the digital age, learning can and must become a daylong and lifelong experience. National education initiatives should aim to improve learning opportunities not only in schools, but also in homes, community centers, museums, and workplaces. In Denmark, for example, the Ministry of Education joined with the Ministry of Business and Industry to create Learning Lab Denmark, a new research lab that studies learning in all settings, in all stages of life. In the years ahead, the Internet will open up new learning opportunities, enabling new types of "knowledge-building communities" in which children (and adults) around the globe collaborate on projects and learn from one another.

Towards the Creative Society

In the 1980s, there was much talk about the transition from the "Industrial Society" to the "Information Society." No longer would natural resources and manufacturing be the driving forces in our economies and societies. Information was the new king.

In the 1990s, people began to talk about the "Knowledge Society." They began to realize that information itself would not bring about important change. Rather, the key was how people transformed information into knowledge and managed that knowledge.

The shift in focus from "information" to "knowledge" is an improvement. But I prefer a different conception: the "Creative Society." As I see it, success in the future will be based not on how much we know, but on our ability to think and act creatively.

The proliferation of digital technologies has accentuated the need for creative thinking in all aspects of our lives, and has also provided tools that can help us improve and reinvent ourselves. Throughout the world, computing and communications technologies are sparking a new entrepreneurial spirit, the creation of innovative products and services, and increased productivity. The importance of a well-educated, creative citizenry is greater than ever before.

Children should play a central role in this transition to the Creative Society. Childhood is one of the most creative periods of our lives. We must make sure that children's creativity is nourished and developed, and we must help children learn how to extend and refine their creative abilities, so that the creativity of childhood persists and grows throughout life.

To achieve these goals will require new approaches to education and learning, and new types of technologies to support those new approaches. The ultimate goal is a society of creative individuals who are constantly inventing new possibilities for themselves and their communities.

Acknowledgements

Many people have contributed to the projects and ideas described in this paper. I am particularly grateful to Robbie Berg, Rick Borovoy, Gail Breslow, Stina Cooke, John Galinato, Fred Martin, Bakhtiar Mikhak, Seymour Papert, Natalie Rusk, Brian Silverman, and Claudia Urrea. The projects described in this paper have received generous support from many sponsors of the MIT Media Lab, most notably the LEGO Company, Intel Corporation, and the National Science Foundation. The ideas about the Creative Society were developed through my participation in the Next Generation Forum.

References

Martin, F., B. Mikhak, and B. Silverman. "MetaCricket: A Designer's Kit for Making Computational Devices." *IBM Systems Journal* 39, 3 & 4 (2000): 795–815.

Papert, S. *The Children's Machine: Rethinking School in the Age of the Computer*. New York: Basic Books, 1993.

Papert, S., and M. Resnick. "Technological Fluency and the Representation of Knowledge." Proposal to the National Science Foundation. MIT Media Laboratory (1995).

Resnick, M., F. Martin, R. Sargent, and B. Silverman. "Programmable Bricks: Toys to Think With." *IBM Systems Journal* 35 (1996):443–452.

Resnick, M. "Technologies for Lifelong Kindergarten." *Educational Technology Research and Development* 46, 4 (1998).

Resnick, M., N. Rusk, and S. Cooke. "The Computer Clubhouse: Technological Fluency in the Inner City." In *High Technology and Low-Income Communities,* edited by D. Schon, B. Sanyal, and W. Mitchell. Cambridge: MIT Press, 1998: 266–286.

Resnick, M., R. Berg, and M. Eisenberg. "Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Investigation." *Journal of the Learning Sciences* 9 (2000): 7–30.

Published in:
Singer, D., Golikoff, R., and Hirsh-Pasek, K. (eds.), Play = Learning: How play motivates and enhances children's cognitive and social-emotional growth*. Oxford University Press. 2006.*

Computer as Paintbrush: Technology, Play, and the Creative Society

Mitchel Resnick
MIT Media Laboratory

Introduction

Let's start with a familiar children's game: Which of these things is not like the other? Which of these things just doesn't belong?

Television. Computer. Paintbrush.

For many people, the answer seems obvious: the paintbrush doesn't belong. After all, the television and the computer were both invented in the 20^{th} century, both involve electronic technology, and both can deliver large amounts of information to large numbers of people. None of that is true for the paintbrush.

But, in my view, computers will not live up to their potential until we start to think of them less like televisions and more like paintbrushes. That is, we need to start seeing computers not simply as information machines, but also as a new medium for creative design and expression.

In recent years, a growing number of educators and psychologists have expressed concern that computers are stifling children's learning and creativity, engaging children in mindless interaction and passive consumption (Cordes and Miller, 2000; Oppenheimer, 2003). They have a point: today, many computers *are* used in that way. But that needn't be the case. This paper presents an alternate vision of how children might use computers, in which children use computers more like paintbrushes and less like televisions, opening new opportunities for children to playfully explore, experiment, design, and invent. My goal in this paper is not to provide conclusive evidence but rather, through illustrative examples, to provoke a rethinking of the roles that computers can play in children's lives.

An Example: Alexandra's Marble Machine

To provide a clearer sense of how computers can serve as paintbrushes, this section tells the story of Alexandra, an 11-year-old girl who used a tiny computer called a Cricket as a new medium for expression, experimentation, and exploration.

Alexandra wasn't very excited about school, but she loved coming to the Computer Clubhouse in her neighborhood in Boston. Alexandra's local Clubhouse was part of a worldwide network of after-school centers established to help young people (ages 10-18) from low-income communities learn to express themselves creatively with new technologies (Resnick, Rusk, & Cooke, 1998). At Computer Clubhouses, young people become actively engaged in designing with new technologies, creating their own graphic animations, musical compositions, and robotic constructions. Alexandra became particularly excited when two volunteer mentors (from a local university) organized a Clubhouse workshop for building "marble machines" – whimsical contraptions in which marbles careen down a series of ramps and raceways, bouncing off bells and bumpers.

The mentors, Karen Wilkinson and Mike Petrich, brought a variety of craft materials to the Clubhouse: pegboard, wooden slats, bells, string, marbles. They also brought a collection of tiny computers called Crickets, small enough to fit inside a child's hand (Resnick et al., 1996; Martin, Mikhak, & Silverman, 2000; Resnick, Berg, & Eisenberg, 2000). Crickets can be programmed to control motors and lights, receive information from sensors, and communicate with one another via infrared light. Children can use Crickets to make their constructions come alive – for example, making a motor turn on whenever a touch sensor is pressed, or whenever a shadow is cast over a light sensor.

Alexandra became interested in the marble-machine project right away. She cut wooden slats to serve as ramps, and inserted the ramps into a pegboard. She began playfully rolling marbles from one ramp to another, trying to create interesting patterns of motion, without the marbles dropping off. As the marbles dropped from one ramp to another, Alexandra giggled with delight.

Next, Alexandra created a Cricket-controlled conveyor belt with a small basket on top. Her plan: the marble should roll down a ramp into the basket, ride along the conveyor belt inside the basket, then drop onto the next ramp when the basket tipped over at the end of the conveyor belt. How would the conveyor belt know when to start moving? Alexandra programmed the conveyor-belt Cricket to listen for a signal from another Cricket higher up on the pegboard, alerting it that the marble was on its way. The conveyor-belt Cricket waited two seconds, to make sure the marble had arrived safely in the basket, before starting to move the conveyor belt and basket.

Alexandra worked on her project for several weeks, experimenting with many different configurations of the ramps, and adjusting the timing of the conveyor belt. She playfully tried out new features – for example, putting bells on the ramps, so that the marbles would make jingling sounds as they rolled past.

Alexandra decided to enter her marble machine into her school's science fair. But when she talked to her classroom teacher about it, the teacher said that the marble machine was not acceptable as a science-fair project. The teacher explained that a science-fair project must use the "scientific method": the student must start with a hypothesis, then gather data in an effort to prove or disprove the hypothesis. The marble machine, said the teacher, didn't follow this approach.

Alexandra was determined to enter her marble machine in the science fair. With support from mentors at the Clubhouse, she put together a sequence of photographs showing different phases of the marble-machine construction. Even though Alexandra never wrote a hypothesis for her project, her teacher ultimately relented and allowed her to enter the marble machine in the school science fair. Much to Alexandra's delight, she was awarded one of the top two prizes for the entire school.

What did Alexandra learn through her marble-machine project? A great deal. Although Alexandra's teacher was concerned that the project did not use the scientific method, the project is, in fact, a wonderful example of the scientific method. True, Alexandra did not start with a single overarching hypothesis. But as she playfully experimented with her marble machine, Alexandra was continually coming up with new design ideas, testing them out, and iterating based on the results. Each of these design ideas can be viewed as a "mini-hypothesis" for which Alexandra gathered data. Over the course of her project, she investigated literally dozens of these mini-hypotheses. While positioning the ramps, for example, Alexandra tested different angles to try to find the maximum range for the marble. Alexandra also experimented to find the right timing for the conveyor belt. She modified the conveyor-belt program so that the basket would make one complete revolution, returning to its original location, properly positioned for the next marble.

Through her playful experiments, Alexandra not only improved the workings of her marble machine but also developed a better understanding and appreciation of the process of scientific investigation. In the spirit of John Dewey's "theory of inquiry" (1910), Alexandra began to develop a scientific frame of mind through her playful yet systematic efforts to solve practical problems that arose in her marble-machine project.

Edutainment versus Playful Learning

The story of Alexandra's marble machine highlights how new technologies can support playful learning – and how playful activities can help children understand and make full use of new technologies. Of course, the idea of mixing play, technology, and learning is hardly new. In establishing the first kindergarten in 1837, Friedrich Froebel used the technology of his time to develop a set of toys (which became known as "Froebel's gifts") with the explicit goal of helping young children learn important concepts such as number, size, shape, and color (Brosterman, 1997). Other educators, such as Maria Montessori (1912), have built on Froebel's ideas, creating a wide range of manipulative materials that engage children in learning through playful explorations.

More recently, there has been a surge of computer-based products that claim to integrate play and learning, under the banner of "edutainment." But these edutainment products often miss the spirit of playful learning. Often, the creators of edutainment products view education as a bitter medicine that needs the sugarcoating of entertainment to become palatable. They provide entertainment as a reward if you are willing to suffer through a little education. Or they boast that you will have so much fun using their products that you won't even realize that you are learning – as if learning were the most unpleasant experience in the world.

Part of the problem is with word *edutainment* itself. When people think about *education* and *entertainment*, they tend to think of them as services that someone else provides for you. Studios, directors, and actors provide you with entertainment; schools and teachers provide you with education. New edutainment companies try to provide you with both. In all of these cases, you are viewed as a passive recipient. But that's not the way most learning happens. In fact, you are likely to learn the most, and enjoy the most, if you are engaged as an active participant, not a passive recipient (e.g., Bruner, 1963).

The terms *play* and *learning* (things that you do) offer a different perspective from *entertainment* and *education* (things that others provide for you). Thus the phrase *playful learning*, as opposed to *edutainment*, conveys a stronger sense of active participation. It might seem like a small change, but the words we use can make a big difference in how we think and what we do.

Alexandra's playful explorations with her marble machine were not a sugarcoating for science experiments; rather, play and learning were fully integrated in her project. Alexandra experimented with ramp angles and conveyor-belt timing not to get a reward or a grade, but as an integral part of her play experience. In other words, Alexandra was driven by "intrinsic motivation," not external rewards. That distinction is critical. Research has found that "self-motivation, rather than external motivation, is at the heart of creativity, responsibility, healthy behavior, and lasting change" (Deci, 1995). Indeed, in our studies, we have found many examples of youth who had short attention spans in traditional school classrooms but displayed great concentration when engaged in projects that interested them.

Alexandra's project was far from easy: she worked very hard on it, and parts of the project were very difficult for her. But the challenge of the project was one of the attractions. Too often, designers and educators try to make things "easy" for learners, thinking that people are attracted to things that are easy to do. But that is not the case. Mihaly Csikszentmihályi (1991) has found that people become most deeply engaged in activities that are challenging, but not overwhelming. Similarly, Seymour Papert has found that learners become deeply engaged by "hard fun" – in other words, learners don't mind activities that are hard as long as the activities connect deeply with their interests and passions (Papert, 1993).

Learning through Designing

Unfortunately, projects like Alexandra's marble machine are the exception, not the rule, in children's use of new technologies. Children have many opportunities to *interact* with new technologies – in the form of video games, electronic storybooks, and "intelligent" stuffed animals. But rarely do children have the opportunity to *create* with new technologies, as Alexandra did with the Crickets in her marble machine.

Research has shown that many of children's best learning experiences come when they are engaged not simply in interacting with materials but in designing, creating, and inventing with them (Papert, 1980; Resnick, 2002). In the process of designing and creating – making sculptures out of clay or towers with wooden blocks – children try out their ideas. If their creations don't turn out as they expected or hoped, they can revise their ideas and create something new. It's an iterative cycle: new ideas, new creations, new ideas, new creations.

This design cycle can be seen as a type of play: children play out their ideas with each new creation. In design activities, as in play, children test the boundaries, experiment with ideas, explore what's possible. As children design and create, they also learn new concepts. When they create pictures with a paintbrush, for example, they learn how colors mix together. When they build houses and castles with wooden blocks, they learn about structures and stability. When they make bracelets with colored beads, they learn about symmetries and patterns.

In my research group at the MIT Media Lab, our goal is to develop new technologies that follow in the tradition of paintbrushes, wooden blocks, and colored beads, expanding the range of what children can create, design, and learn. Our Programmable Brick technology, for example, is a natural extension of the LEGO brick. The original LEGO brick, developed in the 1950s, enabled children to build structures like houses and castles. In the 1970s, the LEGO Company expanded its construction kits to include gears, pulleys, and other mechanical parts, enabling children to build their own mechanisms. Programmable Bricks, which we developed in the 1990s in collaboration with the LEGO Company, represent a third generation. With these new bricks, children can program their LEGO creations to move, sense, interact, and communicate. Now, children can build not only structures and mechanisms but also behaviors.

Programmable Bricks are commercially available as part of a robotics kit called LEGO Mindstorms. Over the past decade, there have been hundreds of different robotic toys on the market, but Mindstorms is fundamentally different. With most robotic toys, children simply interact with a pre-built robot. With Mindstorms, children create their own robots: they use gears, axles, pulleys, and cams to build the mechanisms, connect motors to drive the motion, attach sensors to detect conditions in the world (temperature, light levels, etc.), and write computer programs to guide the robot's behavior (turning motors on and off based on inputs from the sensors).

By creating their own robots, children gain a deeper understanding of the ideas underlying the workings of robots. In one fifth-grade class, for example, students used a Programmable Brick to create a LEGO dinosaur that was attracted to flashes of light, like one of the dinosaurs in Jurassic Park. To make the dinosaur move toward the light, the students needed to understand basic ideas about feedback and control. They wrote a program that compared readings from the dinosaur's two light-sensor "eyes." If the dinosaur drifted too far to the left (i.e., more light in the right eye), the program made it veer back to the right; if the dinosaur went too far right (more light in the left eye), the program corrected it toward the left. This classic feedback strategy is typically not taught until university-level courses. But with the right tools, fifth graders were able to explore these ideas (Resnick, Bruckman, & Martin, 1996).

Crickets and Crafts

In her marble machine, Alexandra used a new version of Programmable Brick called the Cricket. While the Programmable Bricks in LEGO Mindstorms were designed primarily for controlling robots, the Crickets are designed for more artistic and expressive projects. The Crickets can control not only motors but also multi-colored lights and music-synthesis devices, so children can use Crickets to build their own musical instruments and light sculptures. The Crickets are also much smaller than previous Programmable Bricks, so they are well-suited for projects that need to be small and mobile, such as electronic jewelry.

The Cricket was designed to feel more like a craft material than an information-processing machine, in hopes that children would see the Cricket as just another object in their bin of construction parts – and use the Cricket just as playfully and creatively as they use traditional craft materials. One indicator of success: when Alexandra described the parts of her marble machine, she listed Crickets right along with all of the other materials: ". . . slopes, stoppers, Crickets, LEGOs, . . ."

To explore the possibilities of integrating Cricket technology with traditional craft activities, my research group co-organized a hands-on workshop (called Digital Dialogues) with Haystack Mountain School of Crafts, an internationally renowned craft center in Maine (Willow, 2004). At the workshop, artists worked alongside technologists and engineers, sharing ideas, techniques, and materials. Sally McCorkle, a sculptor from Penn State University, used a Cricket, a small fan, and a distance sensor to create an interactive sculpture that blew gold dust in interesting patterns whenever anyone approached. Artist Therese Zemlin created a series of handmade paper lanterns with small lights inside, and programmed the lights to change color and intensity based on the movements of the people around the lanterns. Three Media Lab researchers collaborated with blacksmith Tom Joyce to create a vessel that could "talk for itself," telling the story of its own making. When you reached into the vessel, sensors activated videos showing how the metal had been forged and riveted.

We have found that activities integrating computation and craft provide a good context for learning math, science, and engineering ideas – especially for young people who are alienated by traditional approaches to math and science education, which often emphasize abstract concepts and formal systems rather than hands-on design and experimentation. Although screen-based computer applications offer many advantages, Michael and Ann Eisenberg (2000) argued that "something is lost, too, in this move away from the physical – something pleasurable, sensually and intellectually, about the behavior of stuff." Computational crafts, they argue, combine the best of the physical and computational worlds:

> It's a natural desire to employ all one's senses and cognitive powers in the course of a single project. We do not feel that a love of crafts is incompatible with technophilia, nor that an enjoyment of computer applications must detract from time spent in crafting. The world is not, or should not be at any rate, a battleground between the real and the virtual. It is instead a marvelous continuum, a source of wonders that blend and knead together

> the natural and artificial, the traditional and novel, the scientifically objective and the personally expressive, the tangible and the abstract. We anticipate a future in which ever more astonishing things will present themselves to our minds, and ever more astonishing ideas to our hands.

Supporting Playful Learning (and Learningful Play)

Regardless of how innovative or evocative they are, new technologies can not, on their own, ensure playful-learning experiences. Technologies can always be used in multiple ways – including many ways not intended or desired by their designers. LEGO Mindstorms, for example, was designed as a "robotics invention system," to encourage people to develop their own robotic inventions. And, certainly, many children (and adults too) have used Mindstorms in creative and inventive ways. But there are also many classrooms where the teacher assigns students to build a particular robot according to pre-designed plans, then grades the students on the performance of their robots.

Our ultimate goal is not creative technologies, but rather technologies that foster creative thinking and creative expression. This section discusses several strategies that we have developed over the years to try to maximize the chances that children will use our technologies in creative, playful, and "learningful" ways.

Making It Personal

We have found that children become most engaged with new technologies, and learn the most in playing with these technologies, when they work on projects growing out of their own personal interests. When children care deeply about the projects they are working on, they are not only more motivated but they also develop deeper understandings and richer connections to knowledge.

Consider the case of Jenny, an 11-year-old girl. Jenny loved watching birds, so when she was introduced to the Cricket, she decided to use it to build a new type of bird feeder. Jenny already had a bird feeder in her backyard, but there was a problem: often, the birds would come while Jenny was away at school, so she didn't get to see the birds. With the Cricket, Jenny figured she could build a new bird feeder that would collect data about the birds that landed on it.

Jenny started by making a wooden lever that served as a perch for the birds. The long end of the lever was next to a container with food for the birds. At the other end of the lever, Jenny attached a simple homemade touch sensor consisting of two paper clips. Jenny's idea: When a bird landed near the food, it would push down one end of the lever, causing the two paper clips at the other end to move slightly apart. Jenny connected the paper clips to one of the sensor ports on a Cricket, so that the Cricket could detect whether the paper clips were in contact with one another.

But what should the bird feeder do when a bird landed on it? At a minimum, Jenny wanted to keep track of the number of birds. She also thought about weighing the birds.

But she decided it would be most interesting to take photographs of the birds. She began exploring ways of connecting a camera to her bird feeder, built a motorized LEGO mechanism that moved a small rod up and down, and mounted the mechanism so that the rod was directly above the shutter button of the camera. Finally, Jenny plugged the mechanism into her Cricket and wrote a program for the Cricket. The program waited until the paper clips were no longer touching one another (indicating that a bird had arrived), and then turned on the motorized LEGO mechanism, which moved the rod up and down, depressing the shutter button of the camera. At the end of the day, the camera would have taken pictures of all of the birds that had visited the bird feeder.

Jenny worked on the project for several hours a week over the course of three months. By the end, the sensor and mechanism were working perfectly. But when she placed the bird feeder outside her window at home, she got photographs of squirrels (and of her younger sister), not of birds.

Jenny never succeeded in her original plan to monitor what types of birds would be attracted to what types of bird food. But the activity of building the bird feeder provided a rich collection of learning experiences. While building the lever for the bird feeder, Jenny needed to experiment with different lever designs to achieve the necessary mechanical advantage for triggering the paper-clip touch sensor. Jenny also systematically experimented with the placement of her camera, testing it at different distances from the bird perch in an effort to optimize the focus of the photographs. Thus, the bird feeder activity provided Jenny with an opportunity to make use of scientific concepts in a meaningful and motivating context.

The fact that Jenny built the bird feeder herself put Jenny in closer contact with the technology – and with the scientific concepts related to the technology. Crickets provided Jenny with "design leverage," enabling her to create things that would have been difficult for her to create in the past. At the same time, the bricks provided Jenny with "conceptual leverage," enabling her to learn concepts that would have been difficult for her to learn in the past.

Consider Jenny's touch sensor. In general, touch sensors are based on a very simple concept: they measure whether a circuit is open or closed. People interact with touch sensors (in the form of buttons) all of the time. But because most touch sensors appear in the world as "black boxes" (with their internal working hidden from view), most people don't understand (or even think about) how they work. In Jenny's touch sensor, created from two simple paper clips, the completing-the-circuit concept is exposed. Similarly, Jenny's LEGO mechanism for pushing the shutter of the camera helped demystify the control process of the bird feeder; sending an infrared signal from the Cricket to trigger the camera might have been simpler in some ways, but also less illuminating.

Of course, not everything in Jenny's bird feeder is transparent. The Cricket itself can be seen as a black box. Jenny certainly did not understand the inner workings of the Cricket electronics. On the other hand, Jenny was able to directly control the rules underlying the functioning of her bird feeder. Through the course of her project, she continually modified the computer program on the Cricket, to extend the functionality of the bird feeder. After finishing the first version of the bird feeder, Jenny recognized a problem: If a bird were to hop up and down on the perch, the bird feeder would take multiple photographs of the bird. Jenny added a `wait` statement to her program, so that the program would pause for a while after taking a photograph, to avoid the "double-bouncing" problem.

This ability to modify and extend her project led Jenny to develop a deep sense of personal involvement and ownership. She compared her bird-feeder project with other science-related projects that she had worked on in school. "This was probably more interesting cause it was like you were doing a test for something more complicated than just what happens if you add this liquid to this powder," she explained. "It was more like how many birds did you get with the machine *you* made with this complex thing you had to program and stuff" [emphasis hers]. Jenny cared about her bird feeder (and the photographs that it took) in large part because she had designed and built it. The "fun part" of the project, she explained, "is knowing that you made it; *my* machine can take pictures of birds" [emphasis hers].

Many Paths, Many Styles

While developing an early version of the Programmable Brick technology, we tested some prototypes with a fourth-grade class in Boston. We asked the students what types of projects they wanted to work on, and they decided to create an amusement park, with different groups of students working on different rides for the park.

One group of three students worked on a merry-go-round. They carefully drew up plans, then built the structure and mechanisms according to their plans. After they finished building, they wrote a computer program to control the merry-go-round with a touch sensor. Whenever anyone touched the sensor, the merry-go-round would spin for a fixed amount of time. Within a couple hours, their merry-go-round was working.

Another group, also with three students, decided to build a Ferris wheel. But after working half an hour on the basic structure for the Ferris wheel, they put it aside and started building a refreshment stand next to the Ferris wheel. This decision could be viewed as a positive example of students following their interests. But there was a problem: By focusing on the refreshment stand, which did not have any motors or sensors or programming, the students were missing out on some of the important ideas underlying the activity. The students continued to work on structures (as opposed to mechanisms or programming) for several hours. After finishing the refreshment stand, the group built a wall around the amusement park. Then, they created a parking lot, and added lots of little LEGO people walking into the park.

Finally, after the whole amusement-park scene was complete, the students went back and finished building and programming their Ferris wheel. For this group, building the Ferris wheel wasn't interesting until they had developed an entire story and context around it. In the end, their Ferris wheel worked just as well as the first group's merry-go-round. And, like the first group, they learned important lessons about mechanical advantage as they built the gearing system for the Ferris wheel, and they developed their ability to think systematically as they wrote the programs to control the Ferris wheel. But the two groups travelled down very different paths to get to the same result.

These two groups represent two very different styles of playing, designing, and thinking. Turkle and Papert (1992) have described these styles as "hard" (the first group) and "soft" (the second). The hard and soft approaches, they explain, "are each characterized by a cluster of attributes. Some involve organization of work (the hards prefer abstract thinking and systematic planning; the softs prefer a negotiational approach and concrete forms of reasoning); other attributes concern the kind of relationship that the subject forms with computational objects. Hard mastery is characterized by a distanced stance, soft mastery by a closeness to objects."

In many math and science classrooms, the hard approach is privileged, viewed as superior to the soft approach. Turkle and Papert argue for an "epistemological pluralism" that recognizes the soft approach as different, not inferior. My research group has taken a similar stance in the design of new technologies and activities, putting a high priority on supporting learners of all different styles and approaches. We pay special attention to make sure that our technologies and activities are accessible and appealing to the softs; because math and science activities have traditionally been biased in favor of the hards, we want to work affirmatively to close the gap.

Using the Familiar in Unfamiliar Ways

Over the past five years, my research group has collaborated with a group of museums on an initiative called the Playful Invention and Exploration (PIE) Network. The museums have used Crickets to develop a new generation of hands-on activities that combine art, science, and engineering. By taking a playful approach to invention, and integrating engineering with artistic expression, the PIE museums have engaged a broad and diverse population of people in scientific inquiry and invention (Resnick et al., 2000).

Some of the most popular and successful activities at the PIE museums have been based on the use of familiar objects in unfamiliar ways. At the MIT Museum, for example, Stephanie Hunt and Michael Smith-Welch created workshops in which children turned food into musical instruments. At the core of the activity was a simple Cricket program that measured the electrical resistance of an object and played a musical note based on the resistance. The higher the resistance, the higher the note. Children could put different food items on a plate (with electrical connections to the Cricket), and hear the resistance. A marshmallow (high resistance) would play a high-pitched note, while a pickle (low resistance) would play a low-pitched note. Children could play songs by quickly replacing one piece of food with another.

In one workshop, a 9-year-old named Jonah took several pieces of cantaloupe and lined them up in a row. He attached one wire on the left end of the cantaloupe row, and moved a second wire gradually down the row. The musical notes got higher and higher as he moved down the row. The reason: with more cantaloupe pieces between the two wires, there was more resistance, hence higher notes. And thus the melon xylophone was born. Jonah found a xylophone mallet and connected a wire to it. Then, he could tap the cantaloupe pieces with the mallet to play different melodies, just as on a xylophone. As he worked on this playful project, Jonah learned about the workings of electrical circuits, the nature of electrical resistance and conductivity, and the electrical properties of everyday objects.

Inspired by the food-based musical instruments, another 9-year-old named George came up with an idea for a new type of robot. He attached two wires inside the "mouth" of his robot. When the robot bumped into a piece of food, the two wires formed a circuit with the food and measured its resistance. George programmed the robot so that it could tell one type of food from another, based on differences in resistance. George recorded sound clips for the robot to play when it encountered different food. When the robot bumped into a lemon, it would say: "Yuck, a lemon." When it bumped into a pickle, it would say "Yum, a pickle."

As they ran the musical-food workshops, Stephanie and Michael continued with their own food experiments. They discovered that the resistance of a hot dog changes as you bend it, so a hot dog could be used as a "bend sensor." The more you bend a hot dog, the higher the resistance. They experimented with green beans and string cheese too. "We never had a enough bend sensors," said Stephanie. "It was great to discover that we could make our own."

The musical-food activities led children (and the workshop organizers) to start to think about food in new ways. Typically, people think of food in terms of its color or texture or taste. Through Cricket music activities, children began to realize that food has other properties – in particular, electrical resistance. And resistance became not just an abstract concept learned in science class but a useful tool for creative expression.

Other PIE workshops used other familiar materials: Q-tips, pipe-cleaners, blocks of ice. As they played with familiar materials, children seemed more comfortable experimenting and exploring. At the same time, they were more intrigued when unexpected things happened. If you're playing with unfamiliar or complex materials and something unexpected happens, you're not so surprised. But if you're playing with something simple and familiar (like a hot dog or piece of cantaloupe) and something surprising happens, then you want to find out more. "The familiar doing the unfamiliar stops you in your tracks," said one PIE workshop leader. "It jars you to want to know more."

The Creative Society

In the 1980s, there was much talk about the transition from the Industrial Society to the Information Society (e.g., Beniger, 1986; Salvaggio, 1989). No longer would natural resources and manufacturing be the driving forces in our economies and societies. Information was the new king.

In the 1990s, people began to talk about the Knowledge Society (e.g., Drucker, 1994). They began to realize that information itself would not bring about important change. Rather, the key was how people transformed information into knowledge, and how they managed and shared that knowledge.

But, as I see it, knowledge alone is not enough. Success in the future – for individuals, for communities, for companies, for nations as a whole – will be based not on what we know or how much we know, but on our ability to think and act creatively. In the 21st century, we are moving toward the Creative Society.

The proliferation of new technologies is quickening the pace of change, accentuating the need for creative thinking in all aspects of our lives. At the same time, some new technologies can foster and support the development of creative thinking. We have seen, for example, how Cricket-based activities at the PIE museums can help children develop as creative thinkers.

In some ways, children can serve as models for the Creative Society. Childhood is one of the most creative periods of our lives. We must make sure that children's creativity is nurtured and developed, providing children with opportunities to exercise, refine, and extend their creative abilities. That will require new approaches to education and learning – and new types of technologies to support those new approaches. The ultimate goal is a society of creative individuals who are constantly inventing new possibilities for themselves and their communities.

A New Alliance

In March 2001, I had one of the most frustrating meetings of my life. Three leaders of the Alliance for Childhood came to visit me at the MIT Media Lab. The previous September, the group had published a report called *Fool's Gold: A Critical Look at Computers in Childhood* (Cordes and Miller, 2000). In reading the report, I found myself agreeing with the authors on many issues. The report emphasized the importance of nurturing children's creative abilities, arguing that "creativity and imagination are prerequisites for innovative thinking, which will never be obsolete in the workplace." I certainly agreed. And the report expressed concern that many new technologies restricted rather than encouraged creative thinking: "A heavy diet of ready-made computer images and programmed toys appear to stunt imaginative thinking." Again, I agreed: Most computer-based products for children are like televisions not paintbrushes, delivering pre-programmed content rather than fostering exploration and expression.

I was pleased that the leaders of the Alliance had asked to visit the Media Lab. I looked forward to showing them some of the projects that children had created with our Cricket technology. I felt that our Cricket research was grounded in the same core values expressed in their report. I wanted to show them that some technologies, rather than stunting imaginative thinking, could actually foster and support the development of creative thinking and creative expression.

But the meeting didn't go according to my expectations. After I showed the visitors Jenny's bird feeder, and told them the story of how Jenny had built and programmed it, one of the visitors turned to me and said: "Don't you think it's a problem to take children away from creative play experiences?" I couldn't believe it. I had just described what I considered to be an extraordinarily playful and creative project, but the visitor from the Alliance didn't see it that way. She saw a project using advanced technology, and immediately assumed that the child could not possibly have been doing anything creative.

The interaction made me aware of how polarized our discussions about children and technology have become. There is no doubt, as the *Fool's Gold* report persuasively argues, that the promoters of new technologies make excessive claims and promises, assuming that all technologies must be worthwhile technologies. But it is equally true that the critics of new technologies are too quick to lump all technologies together and dismiss them collectively.

Although I work at one of the world's leading centers of technological innovation, I often find myself sympathizing more with the techno-critics than with the techno-enthusiasts. I resonated with the *Fool's Gold* report when it asserted (p. 68): "Knowledgeable, caring teachers – not machines – are best able to mediate between young children and the world." I, too, am deeply skeptical about "intelligent tutoring systems" that try to put a computer in the place of a teacher. But in the very next sentence, the *Fool's Gold* report argues: "Low-tech tools like crayons, watercolors, and paper nourish children's inner capacities and encourage the child to freely move in, directly relate to, and understand the real world." Why restrict it to "low-tech" tools? Does the ability to "nourish children's inner capacities" really depend on the level of technology? A century ago, crayons were considered advanced technology. Did that make them less able to nourish children's inner capacities?

We need to move away from generalizations about all computers or all technologies, and consider instead the specifics of each technology and the context of its use. Some technologies, in some contexts, foster creative thinking and creative expression; other technologies, in other contexts, restrict it. Rather than focusing on the division between techno-critics and techno-enthusiasts, we need to focus on the difference between activities that foster creative thinking and creative expression (whether they use high-tech, low-tech, or no-tech) and those that don't.

New alliances are needed. At the Playing for Keeps conference in October 2004, I had the good fortune to meet again with Joan Almon, coordinator and president of the board of U.S. Alliance for Childhood. It was the first time Joan and I had met since the meeting at MIT in 2001. I told Joan how frustrated I had been by the earlier meeting – frustrated not because we disagreed (I disagree with many people) but because we allowed our disagreements to overwhelm and obscure what I thought were deep commonalities. We talked for several hours, and we did, indeed, find many shared values, beliefs, and goals. A few months later, Joan came to MIT and spent two days with my research group. We still have our differences, and I'm sure we always will. But those of us who believe in paintbrushes over televisions need to stick together.

Acknowledgements

Robbie Berg, Keith Braafladt, Mike Eisenberg, Stephanie Hunt, Fred Martin, Bakhtiar Mikhak, Steve Ocko, Seymour Papert, Mike Petrich, Margaret Pezalla-Granlund, Natalie Rusk, Brian Silverman, Michael Smith-Welch, Karen Wilkinson, and Diane Willow have made important contributions to the technologies and activities described in this paper. I want to give special thanks to all of the children who have participated in activities with our Crickets and other technologies. (Pseudonyms are used for all children mentioned in this paper.)

I am grateful to members of the LEGO Learning Institute, the Playful Learning Panel, and the PIE Network for many stimulating discussions about the nature and value of playful learning. This research has been supported by generous grants from the LEGO Company, the Intel Foundation, the National Science Foundation (grants CDA-9616444, ESI-0087813, and ITR-0325828), the MIT Media Laboratory's Things That Think and Digital Life consortia, and the Center for Bits and Atoms (NSF CCR-0122419). Portions of this paper previously appeared in Resnick, Berg, & Eisenberg (2000).

References

Beniger, (1986). *The Control Revolution: Technological and Economic Origins of the Information Society*. Cambridge, MA: Harvard University Press.

Brosterman, N. (1997). *Inventing Kindergarten*. New York: Harry N. Abrams Inc.

Bruner, J. (1963). *The Process of Education*. Cambridge, MA: Harvard University Press.

Cordes, C., & Miller, E. (2000). *Fool's Gold: A Critical Look at Computers in Childhood*. College Park, MD: Alliance for Childhood.

Csikszentmihalyi, M. (1991). *Flow: The Psychology of Optimal Experience*. New York: Harper-Collins Publishing.

Deci, E. (1995). *Why We Do What We Do: Understanding Self-Motivation*. New York: Putnam Publishing.

Dewey, J. (1910). *How We Think*. Boston: Heath.

Drucker, P. (1994). *Knowledge Work and Knowledge Society*. Edwin L. Godkin Lecture, John F. Kennedy School of Government, Harvard University, Cambridge, MA.

Eisenberg, M., & Eisenberg, A. (2000). The Developing Scientist As Craftsperson. In N. Roberts, W. Feurzeig, & B. Hunter (Eds.), *Computer Modeling and Simulation in Pre-College Science Education* (pp. 259-281). New York: Springer-Verlag.

Eisenberg, M. (2003). Mindstuff: Educational Technology Beyond the Computer. *Convergence*, *9* (2), 29-53.

Martin, F., Mikhak, B., & Silverman, B. (2000). MetaCricket: A designer's kit for making computational devices. *IBM Systems Journal, 39* (3-4), 795-815.

Montessori, M. (1912). *The Montessori Method*. New York: Frederick Stokes Co.

Oppenheimer, T. (2003). *The Flickering Mind: Saving Education from the False Promise of Technology*. New York: Random House.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.

Papert, S. (1993). *The Children's Machine: Rethinking School in the Age of the Computer*. New York: Basic Books.

Resnick, M., Martin, F., Sargent, R., & Silverman, B. (1996). Programmable Bricks: Toys to Think With. *IBM Systems Journal*, *35* (3-4), 443-452.

Resnick, M., Bruckman, A., & Martin, F. (1996). Pianos Not Stereos: Creating Computational Construction Kits. *Interactions*, *3* (6), 41-50.

Resnick, M., Rusk, N., & Cooke, S. (1998). The Computer Clubhouse: Technological Fluency in the Inner City. In D. Schon, B. Sanyal, & W. Mitchell (Eds.), *High Technology and Low-Income Communities* (pp. 266-286). Cambridge, MA: MIT Press.

Resnick, M., Berg, R., & Eisenberg, M. (2000). Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Investigation. *Journal of the Learning Sciences*, *9* (1), 7-30.

Resnick, M., Mikhak, M., Petrich, M., Rusk, N., Wilkinson, K., & Willow, W. (2000). *The PIE Network: Promoting Science Inquiry and Engineering through Playful Invention and Exploration with New Digital Technologies*. Proposal to the U.S. National Science Foundation (project funded 2001-2004). MIT Media Laboratory, Cambridge, MA.

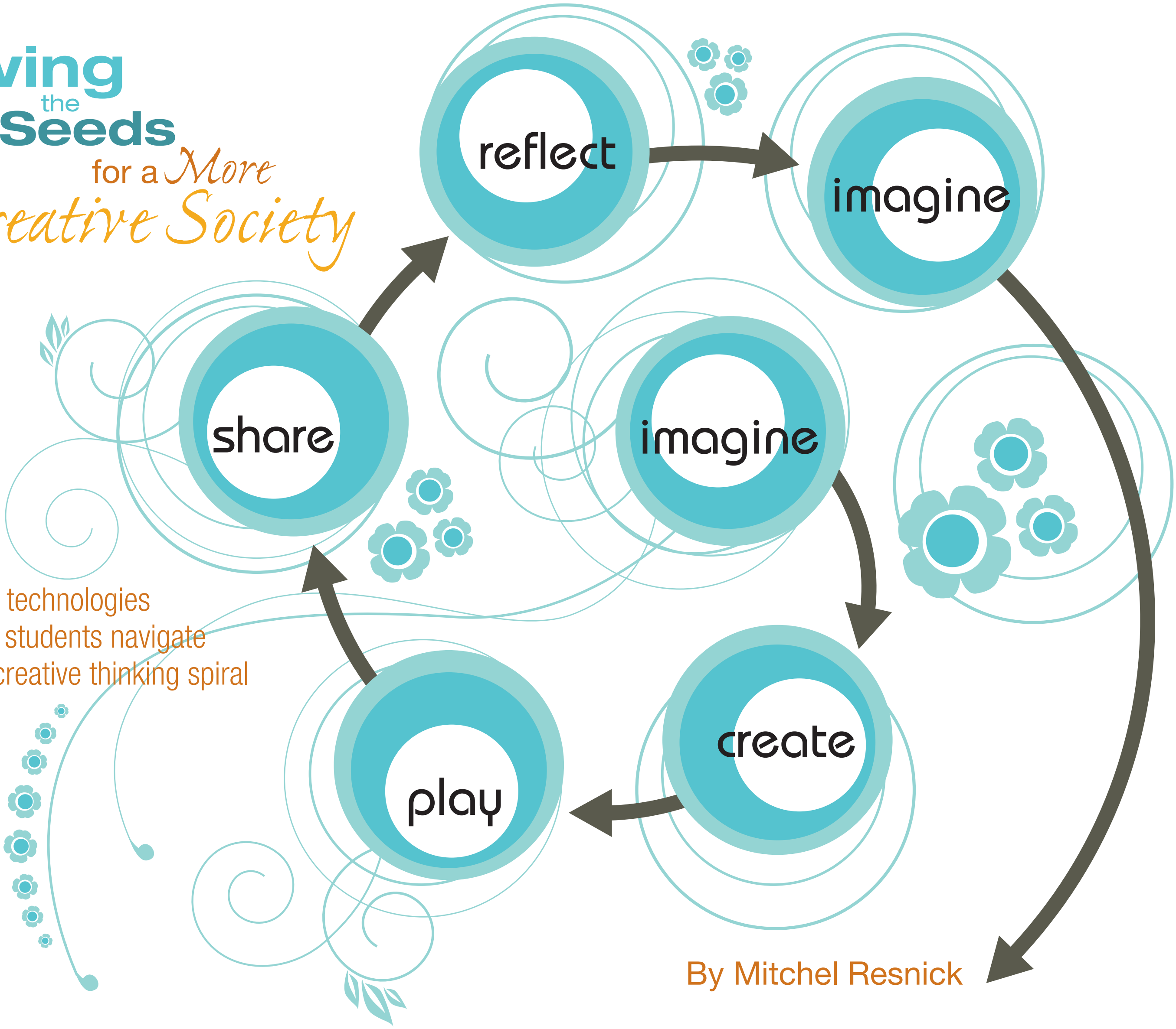
Resnick, M. (2002). Rethinking Learning in the Digital Age. In G. Kirkman (Ed.), *The Global Information Technology Report: Readiness for the Networked World* (pp. 32-37). Oxford: Oxford University Press.

Salvaggio, J. (1989). *The Information Society: Economic, Social, and Structural Issues*. Mahwah, NJ: Lawrence Erlbaum.

Turkle, S., & Papert, S. (1992). Epistemological Pluralism and the Revaluation of the Concrete. *Journal of Mathematical Behavior*, *11* (1), 3-33.

Willow, D. (Ed.). (2004). *Digital Dialogues: Technology and the Hand*. Haystack Monograph Series, Haystack Mountain School of Crafts, Deer Isle, ME.

Sowing the Seeds for a More Creative Society



New technologies help students navigate the creative thinking spiral

By Mitchel Resnick

In the 1980s, there was much talk about the transition from the Industrial Society to the Information Society. Then in the 1990s people began to talk about the Knowledge Society, noting that information is useful only when it is transformed into knowledge.

But as I see it, knowledge alone is not enough. In today's rapidly changing world, people must continually come up with creative solutions to unexpected problems. Success is based not only on what you know or how much you know, but on your ability to think and act creatively. In short, we are now living in the Creative Society.

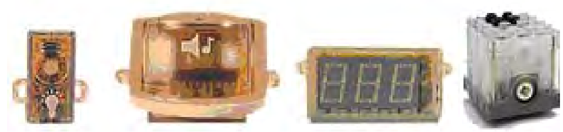
Unfortunately, few of today's classrooms focus on helping students develop as creative thinkers. Even students who perform well in school are often unprepared for the challenges that they encounter after graduation, in their work lives as well as their personal lives. Many students learn to solve specific types of problems, but they are unable to adapt and improvise in response to the unexpected situations that inevitably arise in today's fast-changing world.

New technologies play a dual role in the Creative Society. On one hand, the proliferation of new technologies is quickening the pace of change, accentuating the need for creative thinking in all aspects of people's lives. On the other hand, new technologies have the potential, if properly designed and used, to help people develop as creative thinkers, so that they are better prepared for life in the Creative Society.

In this article, I discuss two technologies developed by my research group at the MIT Media Lab with the explicit goal of helping people develop as creative thinkers. The two technologies, called Crickets and Scratch, are designed to support what I call the "creative thinking spiral." In this process, people *imagine* what they want to do, *create* a project based on their ideas, *play* with their creations, *share* their ideas and creations with others, and *reflect* on their experiences—all of which leads them to *imagine* new ideas and new projects. As students go through this process, over and over, they learn to develop their own ideas, try them out, test the boundaries, experiment with alternatives, get input from others, and generate new ideas based on their experiences.

Cricket

Cricket Sensors:
Light sensor, touch sensor, sound sensor, resistance sensor.



Cricket Outputs:
Multi-colored light, sound box, numerical display, motor.

Crickets

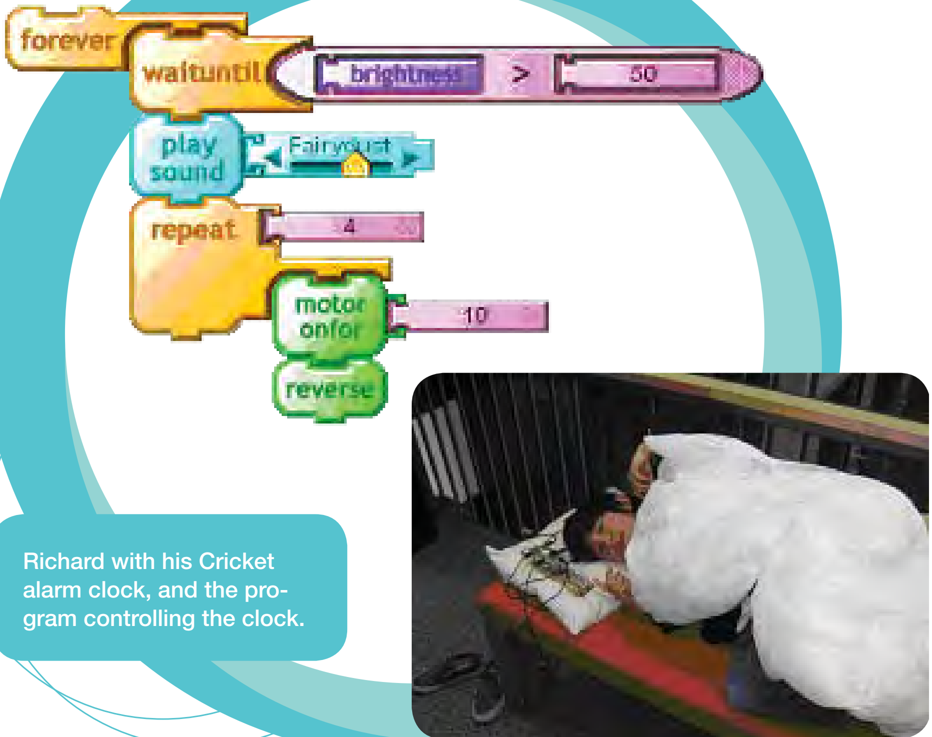
Today's world is full of objects that sense and respond: doors that open automatically when you walk toward them, outdoor lights that turn on automatically when the sun goes down, stuffed animals that talk to you when you squeeze them. Children interact with these objects all of the time, but most have no idea how they work. And if children want to create their own interactive objects, most have no idea how to do it.

The Cricket is designed to change that. Children can connect lights, motors, and sensors to a Cricket, then program their creations to spin, light up, and play music. Children can use Crickets to create all types of interactive inventions: musical sculptures, interactive jewelry, dancing creatures. In the process, children learn important science and engineering concepts, and they develop a better understanding of the interactive objects in the world around them.

At a week-long workshop in Iceland, for example, Richard, an 11-year-old boy, decided to use a Cricket to create an automatic alarm clock to wake him in the morning. He connected a light sensor, a motor, and a sound box to a Cricket, and he attached a feather to the motor. Then Richard programmed the Cricket so that it would play a melody and gently twirl the feather against his face when the light sensor detected the sun shining through his bedroom window in the morning. Richard experimented with his new alarm clock, and it seemed to work well. But a friend pointed out a problem. Because Iceland is located so far to the north, sunrise occurs at very different times over the course of the year, so the alarm clock wouldn't be very reliable. Richard thought about this problem, and when he created a poster about his project for the public exhibition at the end of the workshop, he included a warning at the bottom: "For Export Only."

As Richard worked on his alarm clock project, he actively engaged in all parts of the creative thinking spiral: he came up with an idea, created a prototype, experimented with it, shared his ideas with others, and revised his plans based on the feedback. By the end, Richard was full of ideas on how to improve his alarm clock—and he had refined his skills as a creative thinker.

In many ways, Crickets are similar to the Lego Mindstorms robot construction kits now used by millions of students around the world. But there are also important differences. While Mindstorms kits are designed especially for making robots, Crickets are designed especially for making artistic creations with colored lights, sound, music, and motion. Crickets are now sold commercially as part of a kit, called the PicoCricket Kit, that includes not only Lego bricks and electronic parts, but also arts-and-craft materials such as pom-poms, pipe cleaners, and googly eyes. By providing a broader range of materials and supporting activities involving light and sound (in addition to motion), we hope to encourage a broader range of projects—and spark the imaginations of a broader range of children. We are especially interested in broadening participation among girls. Even with strong efforts to increase female participation, only 30% of the participants in Lego Mindstorms robotics competitions are girls. In Cricket



Richard with his Cricket alarm clock, and the program controlling the clock.

activities at museums and after-school centers, participation has been much more balanced among boys and girls.

Crickets have become especially popular in Hong Kong, where government and industry leaders are concerned about the outward migration of manufacturing jobs to other parts of China, and thus feel an urgent need to develop a more creative workforce. Cricket workshops in Hong Kong provide a glimpse into an alternative educational approach, where creative thinking is a top priority.

At one Hong Kong workshop, an 11-year-old girl named Julia was inspired by a pair of shoes that she had seen that contained embedded lights that flashed as the shoes moved. But Julia wasn't interested in buying shoes with pre-programmed lighting patterns; she wanted to create her own patterns. So she connected a Cricket and a series of lights to her boots, then installed a sensor near the bottom of the boot, where it could detect the up-and-down motion of her foot. She programmed the Cricket to change the colors of the lights, based on how fast she was walking.

At the same workshop, an entrepreneurial 12-year-old named Anthony came up with a business idea: a wearable jukebox. He cut a coin slot in the top of a cardboard box, then installed sensors on the underside of the slot to measure the size of the coin inserted. He then programmed the Cricket to play different songs based on what coin the customer put into the box.

For Julia and Anthony, the Cricket provided a way to create and personalize their own interactive inventions. As Julia explained, "With Crickets, you don't have to use what someone else made. You can make it yourself."

Scratch

Just as Crickets give students the power to create and control things in the physical world, Scratch gives them the power to create and control things in the online world.

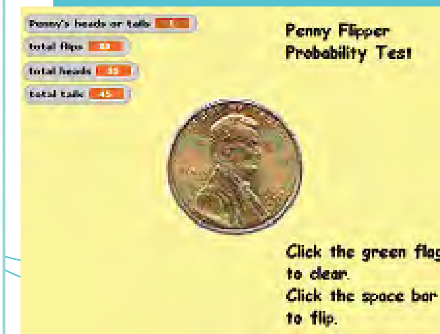
For many students, the Web is primarily a place for browsing, clicking, and chatting. With Scratch, students shift from media consumers to media producers, creating their own interactive stories, games, and animations—then sharing their creations on the Web.

In classrooms, students have begun to use Scratch to create reports and presentations—replacing traditional PowerPoint presentations with content that is far more dynamic and interactive. At the Expo Elementary School in St. Paul, Minnesota, one student created a book report on Ben Franklin, including an interactive game inspired by Franklin's experiments with lightning. Another student created an animated documentary on the dangers of mercury in their school building. At another school, students created a penny-flipping simulation,

Workshop participants in Hong Kong engage their creative thinking to develop Cricket projects.

The Scratch Web site is a YouTube-like environment for sharing and exploring student creations.

Scratch projects.



then ran experiments to test theories in probability and statistics.

"There is a buzz in the room when the kids get going on Scratch projects," says Karen Randall, a teacher at the Expo Elementary School. "Students set design goals for their projects and problem-solve to fix program bugs. They collaborate, cooperate, co-teach. They appreciate the power that Scratch gives them to create their own versions of games and animations."

Students program their Scratch creations by snapping together graphical blocks, without any of the obscure punctuation and syntax of traditional programming languages. In this way, Scratch makes programming accessible to a much broader audience—at a younger age—than ever before.

In the process of programming their Scratch creations, students learn important mathematical concepts in a meaningful and motivating context. While visiting an after-school center, I met a student who was creating an interactive game in Scratch. He didn't know how to keep score in the game, and asked me for help. I showed him how to create a variable in Scratch, and he immediately saw how he could use a variable for keeping score. He jumped up and shook my hand, saying "Thank you, thank you, thank you." I wondered how many eighth grade algebra teachers get thanked by their students for teaching them about variables?

Students can share their Scratch projects on the Scratch Web site (http://scratch.mit.edu), just as they share videos on YouTube. After the site was publicly launched in May 2007, more than 20,000 projects were uploaded to the site in the first three months. Students can browse the site for inspiration and ideas, and if they see a project that they like, they can download it, modify it, and then share the revised version with the rest of the community. The Web site has become a bustling online community. Members are constantly asking questions, giving advice, and modifying one another's projects. More than 15 percent of the projects on the site are extensions of previous work.

Collaboration on the Scratch Web site comes in many different forms. A 15-year-old girl from the UK with the screen name BeeBop created a project full of animated sprites, and encouraged others to use them in their projects. Another 10-year-old girl, using the name MusicalMoon, liked BeeBop's animations and asked if she'd be willing to create "a mountain background from a bird's-eye view" for use in one of her projects. MusicalMoon then asked BeeBop if she wanted to join Mesh Inc., a "miniature company" that MusicalMoon had created to produce "top quality games" in Scratch. A few days later, a 14-year-old boy from New Jersey who went by the moniker Hobbit discovered the Mesh Inc. gallery and offered his services: "I'm a fairly good programmer, and I could help with de-bugging and stuff." Later, an 11-year-old boy from Ireland calling himself Marty was added to the Mesh staff because of his expertise in "scrolling backgrounds."

The Scratch Web site is part of a broader trend toward a more participatory Web, in which people not only point and click but also create and share. Many Web sites enable students to share text, graphics, photos, and videos. Scratch goes a step further, providing the tools for students to create and share *interactive* content, and thus become full participants in the online world.

Learning in the Creative Society

Today's students are growing up in a world that is very different from the world of their parents and grandparents. To succeed in today's Creative Society, students must learn to think creatively, plan systematically, analyze critically, work collaboratively, communicate clearly, design iteratively, and learn continuously. Unfortunately, most uses of technologies in schools today do not support these 21st-century learning skills. In many cases, new technologies are simply reinforcing old ways of teaching and learning.

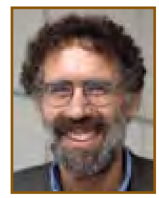
Crickets and Scratch are part of a new generation of technologies designed to help prepare students for the Creative Society. But they are just the beginning. We need to continually rethink our approaches to education and rethink our uses of educational technologies. Just as students need to engage in the creative thinking spiral to prepare for the Creative Society, educators and designers must do the same. We must imagine and create new educational strategies and technologies, share them with one another, and iteratively refine and extend them.

Acknowledgements

Many people have contributed to the development of Crickets and Scratch—and to the educational ideas underlying them. The core design team for the latest version of Crickets included Brian Silverman, Paula Bonta, Natalie Rusk, Robbie Berg, and me. The core design team for the Scratch software and Web site included John Maloney, Natalie Rusk, Andres Monroy-Hernandez, Evelyn Eastmond, Tammy Stern, Amon Millner, Jay Silver, Eric Rosenbaum, Han Xu, Brian Silverman, and me at the MIT Media Lab, in collaboration with Yasmin Kafai's research team at UCLA. Many others have contributed to field-testing and development of activities. Our work on Crickets and Scratch was deeply influenced and inspired by other educational technologies, most notably Logo, Mindstorms, and Squeak Etoys. We greatly appreciate financial support from the Lego Company, Intel Foundation, the National Science Foundation (grants CDA-9616444, ESI-0087813, and ITR-0325828), and MIT Media Lab consortia. All children's names in this article are pseudonyms.

Resources

Cricket: http://www.picocricket.com
Lifelong Kindergarten: http://llk.media.mit.edu
Scratch: http://scratch.mit.edu.



Mitchel Resnick is a professor of learning research and director of the Lifelong Kindergarten research group at the MIT Media Lab. Resnick earned a BS in physics from Princeton and an MS and PhD in computer science from MIT. He is the author of the book Turtles, Termites, and Traffic Jams.

CHAPTER 1

Origins and Guiding Principles of the Computer Clubhouse

Natalie Rusk, Mitchel Resnick, and Stina Cooke

Technology has changed a great deal in the 15 years since we started the first Computer Clubhouse. At that time, no one was carrying around cell phones. Most people had never heard of the Internet. The most popular Web sites today—such as Google, Yahoo, and YouTube—did not yet exist. Although technologies have changed radically, the motivations and needs that led to starting the Computer Clubhouse program have remained the same and continue to drive the program today. So we find it useful to reflect back on the ideas and issues that sparked us to start the first Clubhouse. In this chapter we tell the story of the origins of the first Computer Clubhouse and then discuss the four core principles that have guided the development of the Clubhouse program since its beginning in 1993.

HOW THE COMPUTER CLUBHOUSE STARTED

The first Computer Clubhouse was created in response to a group of children sneaking into a museum. During school vacation week in December 1989, the Computer Museum in downtown Boston offered a robotic workshop for families, using LEGO-Logo robotics materials borrowed from the MIT Media Lab. Anyone could drop in to participate. On the second day, a group of four children showed up, speaking to each other in a combination of English and Spanish. One of the boys in the group, about age 11, picked up a small gray LEGO motor. He was shown how to plug it into a power source to turn it on. The motor began to spin. He called out excitedly for his companions to come see. “Míra, míra! Look at this!” The children started to build a car out of LEGO materials and began to program a computer to control the movements of their car. The children came back to the museum day after day, eager to learn more. After playing with the car for a while, they built and programmed a crane to lift the car. At the end of the week, the robotics workshop was over, and the LEGO-Logo robotics materials were returned to MIT.

The next week, the museum was very quiet. At 3:00 in the afternoon, the doors to the museum’s large elevator opened. Inside were the boy and his friends. They asked, “LEGO-Logo?” We explained that we no longer had the materials available. They wandered around the museum trying out the exhibits. However, museum exhibits are typically designed for short-term interaction and do not offer opportunities for open-ended design. The children looked disappointed.

A couple weeks later, a museum administrator sent an e-mail message to the staff, warning them to be on the lookout for a group of kids sneaking into the museum, and to alert security if the children were seen. It turned out that these were the same children who had enthusiastically participated in the weeklong robotics workshop. Now, because they were hanging around the museum, they were beginning to get into trouble with security.

We asked around to see if there were local after-school centers where these children could participate, but there were none in the downtown area. We also investigated what technology-based learning programs were available for youth in the greater Boston area. We found community technology centers that offered children opportunities to play educational games or to take classes on basic computer skills, but no programs that provided opportunities for youth to develop their own creative projects.

The children sneaking into the museum wanted something different. They were eager to try out new technologies. Here was a group of children who wanted to keep coming back to the museum to work on projects that we knew were educationally valuable (Resnick, 2006). They were reaching out, but there was nowhere for them to go.

THE CREATION OF THE COMPUTER CLUBHOUSE MODEL

So we began to explore the possibility of creating a new type of learning center that would address the needs and interests of these and other young people in the area. Our goal was to create a learning space where youth could have not just to the latest computer technology, but also access to people who could inspire and support them as they developed creative projects based on their interests. As we developed our plans, we drew on the latest ideas from educational researchers and practitioners, and on our own experiences working in experimental educational projects. We brought together advisors from university research groups and community youth programs. We also met with local youth and put together a youth advisory board.

Out of these discussions emerged the ideas and plans for the first Computer Clubhouse. Early on, we identified four Guiding Principles for the Computer Clubhouse (Resnick & Rusk, 1996a). We applied these principles to set up the first Computer Clubhouse at the Computer Museum. But the principles have continued to play an important role as the Clubhouse Network expanded to more than 100 sites over the past 15 years.

Principle 1: Support Learning Through Design Experiences

What was the secret to the success of the LEGO-Logo workshop that sparked the idea for the first Computer Clubhouse? A key factor, in our minds, was the way that participants were actively engaged in designing, creating, and inventing things. Too many educational initiatives try to transmit or deliver information to learners. The Computer Clubhouse is based on a different model of learning and education, where the focus is on *construction* rather than *instruction.*

Indeed, the Clubhouse learning approach draws on an educational philosophy known as *constructionism*, developed by MIT Professor Seymour Papert (1993a). Constructionism is based on two types of construction. First, it asserts that learning is an active process, in which people actively construct knowledge from their experiences in the world. People don't *get* ideas; they *make* them. This aspect of construction comes from the *constructivist* theory of knowledge development by Jean Piaget. To Piaget's concept, Papert added another type of construction, arguing that people construct new knowledge with particular effectiveness when they are engaged in constructing personally meaningful products. Learners might be building a sculpture, writing a poem, composing a song, or programming a computer animation, but what's important is that learners are actively engaged in creating something that is meaningful to themselves or to others around them.

These ideas are at the core of the Clubhouse learning approach. At Clubhouses, young people don't simply interact with technologies, they design and create with technologies. Rather than just watching animations and videos on the Web, Clubhouse members create their own animations and videos. Rather than playing computer games, Clubhouse members create their own computer games (see also Chapter 3).

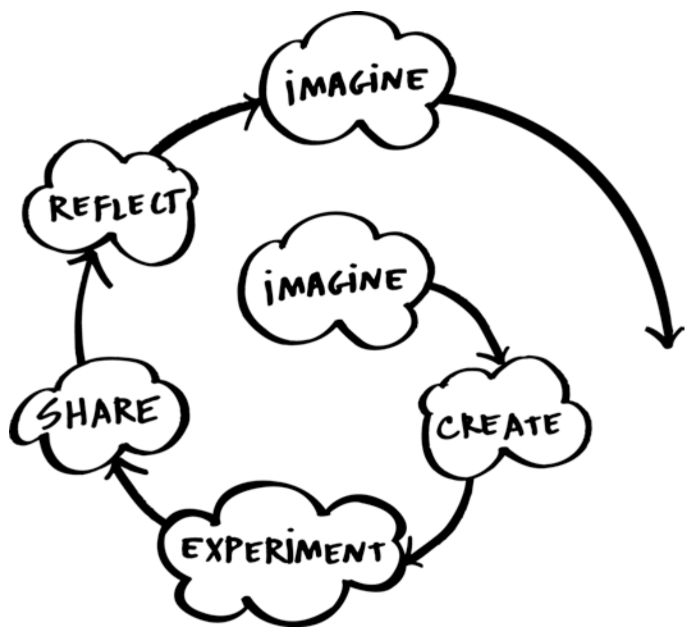
Activities at Clubhouses vary widely, from constructing robotic inventions to orchestrating virtual dancers to writing lyrics to a song. But these varied activities are all based on a common framework: engaging youth in learning through design. To support these activities, Clubhouses provide a variety of design tools, including tools for digital music recording and editing; Web publishing; computer programming and animation;

image and video editing; designing and rendering three-dimensional models, and creating and controlling robotic machines. Clubhouse members often transition quickly from entry-level software to professional-level tools. As Clubhouse members work with these tools, they build toward greater confidence and technical fluency. For example, a young person may start by creating images with a simple paint program like KidPix, then shift to Photoshop to explore more advanced image manipulation and visual effects, then learn to use Scratch or Flash to animate their creations.

At Clubhouses young people not only learn how to use these tools, they learn how to express themselves through these tools. They learn not only the technical details, but also the heuristics of being a good designer: how to conceptualize a project, how to make use of the materials available, how to persist and find alternatives when things go wrong, and how to view a project through the eyes of others. In short, they learn how to manage a complex project from start to finish.

As Clubhouse members work on design projects, they move through what we call the *creative design spiral* (see Figure 1.1). In this process, they imagine what they want to do, create a project based on their ideas, experiment with alternatives, share their ideas and creations with others, and reflect on their experiences—all of which leads them to imagine new ideas and new projects. As youth go through this process, over and over, they learn to develop their own ideas, try them out, test the boundaries, solve problems, get input from others, and generate new ideas based on their experiences.

Figure 1.1: Creative design spiral



Young people often begin with a relatively simple design project, such as taking photos of themselves and placing them into a scene. This initial type of project engages them in the creative design spiral over an afternoon or two. For example, they might start by imagining what kind of scene they want to create, then take a photo of themselves, edit it into a background (such as a sporting event or favorite place), experiment with visual effects, print and show it to others, and discuss ideas for further projects. After some reflection, they might decide to add more characters to the scene and continue with the next iteration of the spiral.

As young people become more fluent with various tools and aspects of the design process, they often develop bigger plans requiring longer time scales, such as making a stop-motion animation, a sophisticated 3-D model, or a collection of songs for a music album. These projects often become complex and involve more people working together as a team.

Principle 2: Help Members Build on Their Own Interests

In schools of education, the focus is usually on methods of teaching, not motivations for learning. Many courses for educators emphasize how and what to teach, but seldom examine why students might want to learn. When the issue of motivation is addressed, the emphasis is often on extrinsic motivators and incentives, such as grades and prizes based on performance. Why? Many people assume that learning is inherently boring. To motivate students to learn, some educators assume that they need to offer rewards, or turn the subject matter into a competitive game, with prizes for those with the best scores.

If you look outside of school, however, you can find many examples of people learning—in fact, learning exceptionally well—without explicit rewards. Youth who seem to have short attention spans in school often display great concentration on projects that they are truly interested in. They might spend hours learning to play the guitar or perform tricks on a skateboard. Indeed, many of the most successful designers, scientists, and other professionals trace their involvement and success in their fields back to a childhood interest. Clearly, youth interests are a great untapped resource.

When youth care about what they are working on, the dynamic of teaching changes. Rather than being "pushed" to learn, youth work on their own and seek out ideas and advice. Not only are youth more motivated, but they also develop deeper understandings and richer connections to knowledge.

At first, some youth interests might seem to be trivial or shallow, but youth can build up large networks of knowledge related to their interests. Pursuing any topic in depth can lead to connections to other subjects and disciplines. The educational challenge is to find ways to help youth make those connections and develop them more fully. For example, an interest in riding a bicycle can lead to investigations of gearing, the physics of balancing, the evolution of vehicles over time, or the environmental effects of different transportation modes.

Clubhouses are designed to support youth in developing their interests. While youth from high-income households generally have many opportunities to build on their interests (for example, music lessons and specialty camps), the youth who typically come to Computer Clubhouses have had few such opportunities. Many have not had the resources and support to identify and explore potential interest areas, let alone to build on them.

Clubhouse participants are encouraged to make their own choices. Just coming to a Clubhouse involves a choice: All of the youth at Clubhouses have chosen to be there, and they can come and go as they please. Once inside a Clubhouse, participants continually confront choices on what to do, how to do it, and whom to work with. Clubhouse staff and mentors help these youth gain experience with self-directed learning, helping them recognize, trust, develop, and deepen their own interests and talents.

Helping youth develop their interests is not just a matter of letting them do what they want. Young people must be given the freedom to follow their fantasies, but they also need the support to make those fantasies come to life. On the walls, shelves, and hard drives of Clubhouses, there are large collections of sample projects, designed to provide participants with a sense of the possible and with multiple entry points for getting started. In one corner of each Clubhouse is a library of books, magazines, and manuals filled with more project ideas (and a sofa to make reading more comfortable). Many youth begin by mimicking a sample project, then work on variations on the theme, and soon develop their own personal path, stemming from their personal interests.

This approach works only if the environment supports a great diversity of possible projects and paths. Young people have a wide variety of different interests, so Clubhouses need to provide a wide variety of different activities to match those interests. The computer plays a key role here. The computer is a type of "universal machine," supporting design projects in many different domains: music, art, science, and mathematics. At any given time, a pair of youth might be using a computer to create a graphic animation, while at the next computer another participant might be using a similar computer to program a robotic construction.

Clubhouse projects often require expertise in a variety of different domains. For example, creating a music video involves recording in the music studio, shooting and editing video, designing an album cover for the CD, and creating a Web site for the group. Such projects allow Clubhouse members with different interests to work together and learn new skills from one another.

Sometimes people misinterpret this guiding principle. When they hear that Clubhouses encourage youth to build on their own interests, they assume that adults need to get out of the way, and let Clubhouse members do everything themselves. For example, we once heard someone propose to lead a workshop for Clubhouse members, helping them learn to create animated comic books. Another person initially dismissed the idea, explaining: "We don't do workshops at the Clubhouse. We let Clubhouse members follow their own interests." But that's not what is intended by this guiding principle. It's important for young people to have choice in what to explore, but they often need a great deal of support in identifying and pursuing their interests. We would advise against a Clubhouse organizing a mandatory workshop where all Clubhouse members were required to learn about animated comic books. But as long as members have the choice of whether or not to participate, we think it's a great idea to offer workshops for Clubhouse members. Such workshops can help Clubhouse members discover what areas that they are (or are not) interested in and help them learn new skills that will be useful in pursuing their interests.

Principle 3: Cultivate an Emergent Community of Learners

A typical computer lab for 30 children is set up with 30 computers on tables in straight rows facing the front of the room. This setup is designed for children to face the teacher at the front of the room and to work alone. In contrast, we designed the Clubhouse space with an explicit goal of encouraging and supporting collaboration.

In a typical Computer Clubhouse, each table with a computer has two or three chairs to facilitate youth working together. The tables are placed in small clusters around the edges of room, leaving more space for circulating around the room. The chairs in Clubhouses all have rolling wheels, allowing members to interact with others more easily by rolling over to see what is on another computer. In the middle of each Clubhouse is a large green table without any computers on it. This table acts as a type of village common, where people come together to share ideas and to work on plans, drawings, crafts, and building projects—or simply to have a snack and catch up.

The Clubhouse space is designed to have the feel of a creative design studio, a combination of an art studio, music studio, video studio, and robotics lab. Some of the design choices might seem unimportant (or even extravagant), but we have found that the design of the space deeply influences the attitudes and activities of the participants. As soon as youth walk into the Clubhouse, the setup of the space suggests possibilities. They can see tools and examples to spark their interest and imagination. At one new Computer Clubhouse, the director remarked with surprise that the behavior of the young people changed dramatically for the better when track lighting was installed. And many Clubhouse staff members have noted that the rolling chairs, though sometimes a distraction, make it much easier and more likely for Clubhouse members to share and collaborate with one another.

At Clubhouses, projects are not fixed entities; they grow and evolve over time. Similarly, no one is assigned to work on any particular team, but rather, communities emerge over time. Design teams form informally, coalescing around common interests. Communities are dynamic and flexible, evolving to meet the needs of the project and the interests of the participants (Resnick, 1996).

To support these evolving collaborations, Computer Clubhouses recruit a culturally diverse team of adult mentors—professionals and college students in art, music, science, and technology. Mentors act as coaches, catalysts, and consultants, bringing new project ideas to their Clubhouses. Most mentors volunteer their time (see also Chapter 8). On a typical day there are two or three mentors at a Clubhouse. For example, engineers might be working on robotics projects with Clubhouse participants, artists on graphics and animation projects, programmers on interactive games. For youth who have never interacted with an adult involved in academic or professional careers, this opportunity is pivotal to envisioning themselves following similar career paths.

In this way Clubhouses provide more than just access to technology. Youth in low-income neighborhoods need access not only to new technologies but also to people who know how to use technology in interesting and creative ways. Clubhouses take advantage of an untapped local resource, providing a new way for people in the community to share their skills with local youth.

By involving mentors, Clubhouses provide inner-city youth with a rare opportunity to see adults working on projects. Mentors do not simply provide support or help; many work on their own projects and encourage Clubhouse youth to join in. John Holt (1977) argued that children learn best from adults who are working on things that they themselves care about: "I'm not going to take up painting in the hope that, seeing me, children will get interested in painting. Let people who *already* like to paint, paint where children can see them" (p. 5).

At Clubhouses youth also get a chance to see adults learning. In today's rapidly changing society, perhaps the most important skill of all is the ability to learn new things. It might seem obvious that youth, in order to become good learners, should observe adults learning. But that is rarely the case in schools. Teachers often avoid situations where students will see them learning; they don't want students to see their lack of knowledge. At Clubhouses, youth get to see adults in the act of learning. For some Clubhouse participants it is quite a shock. Several of them were startled one day when a Clubhouse staff member, after debugging a tricky programming problem, exclaimed, "I just learned something!"

For example, two graduate students from a local university decided to start a new robotics project at one of the Boston-area Clubhouses. For several days, they worked on their own; none of the youth seemed particularly interested. But as the project began to take shape, a few youth took notice. One decided to build a new structure to fit on top of the robot, another saw the project as an opportunity to learn about programming. After a month, there was a small team of people working on several robots. Some youth were integrally involved, working on the project every day. Others chipped in from time to time, moving in and out of the project team. The process allowed different youth to contribute to different degrees, at different times—a process that some researchers call *legitimate peripheral participation* (Lave & Wenger, 1991). As youth become more fluent with the technologies at Clubhouses, they too start to act as mentors. Over time, youth begin to take on more mentoring roles, helping introduce newcomers to the equipment, projects, and ideas of the Clubhouse.

Principle 4: Create an Environment of Respect and Trust

When visitors walk into a Clubhouse, they are often impressed by the artistic creations and the technical abilities of Clubhouse participants. But just as often, they are struck by the way Clubhouse youth interact with one another. The Clubhouse approach puts a high priority on developing a culture of respect and trust. These values not only make the Clubhouse an inviting place to spend time, but they are essential for enabling Clubhouse youth to try out new ideas, take risks, follow their interests, and develop fluency with new technologies. Indeed, none of the other guiding principles can be put into practice without an environment of respect and trust.

There are many dimensions to "respect" at Clubhouses: respect for people, respect for ideas, respect for tools and equipment. Mentors and staff set the tone by treating Clubhouse youth with respect. Right from the start, participants are given access to expensive equipment and encouraged to develop their own ideas. "You mean I can use this?" is a common question for youth to ask when they first visit a Clubhouse and find out about the resources and options available to them.

Even with all these options, youth won't take advantage of the opportunities unless they feel "safe" to try out new ideas. In many settings, youth are reluctant to do so, for fear of being judged or even ridiculed. At Clubhouses, the goal is to make participants feel safe to experiment and explore. No one should get criticized for mistakes or "silly" ideas.

Youth are given the time they need to play out their ideas; it is understood that ideas (and people) need time to develop. One new Clubhouse participant spent weeks manipulating a few images over and over. But then, like a toddler who is late in learning to talk but then starts speaking in full sentences, she started using these images to create spectacular graphic animations.

Clubhouse youth are given lots of freedom and choice. One participant explained why he liked the Clubhouse more than school: "There's no one breathing down your neck here." But with this freedom come high standards and high expectations. Clubhouse staff and mentors do not simply dole out praise to improve the self-esteem of the youth. They treat youth more like colleagues, giving them genuine feedback, and pushing them to consider new possibilities. They are always asking: "What could you do next? What other ideas do you have?" Many Clubhouse youth are learning not only new computer skills but new styles of interaction. Clubhouse youth are treated with respect and trust—and they are expected to treat others the same way.

THE EVOLUTION OF THE GUIDING PRINCIPLES

Over the past 15 years these four Guiding Principles have continued to provide a framework of shared values for the expanding network of Computer Clubhouses. But the principles are not static. As new Clubhouses have opened around the world, the Guiding Principles have evolved to fit changing contexts.

When we first talked about "emergent community," for example, we were thinking about the community of staff, mentors, and members within an individual Computer Clubhouse. As time went on, the idea of "community" evolved. Clubhouses began reaching beyond their walls to develop collaborations with their local communities. And as more and more Clubhouses opened, they began to focus on another type of community: the extended community of Clubhouses around the world. Just as new ideas emerge through interactions among members, mentors, and staff within each individual Clubhouse, new ideas also emerge through interactions among the worldwide network of Clubhouses.

The Clubhouse Guiding Principles need not be limited to Clubhouses themselves. In recent years, a growing number of schools and community organizations have expressed interest in the Clubhouse learning approach. One aspect that has received attention is the role of mentors collaborating with youth on creative projects, which differs from the typical one-on-one tutoring in many after-school programs. Hirsch and Wong (2005), in the *Handbook of Youth Mentoring*, describe the Computer Clubhouse approach as a promising direction for mentoring in after-school centers.

A key challenge for the years ahead is to provide support and connections among educators and program staff interested in applying the Clubhouse Guiding Principles in their local settings. With increased access not just to creative applications of technology but also to a dynamic and supportive learning community, more young people around the world will have opportunities to develop as capable, confident, and creative thinkers.

NOTE

Portions of this article previously appeared in Resnick, Rusk, & Cooke, 1999; Resnick & Rusk, 1996a; and Resnick & Rusk, 1996b.

THE COMPUTER CLUBHOUSE

Constructionism and Creativity in Youth Communities

Edited by

Yasmin B. Kafai
Kylie A. Peppler
Robbin N. Chapman

with Forewords by
Barton Hirsch
Rosaline Hudnell

TEACHERS COLLEGE PRESS
Teachers College, Columbia University
New York and London

DOI:10.1145/1592761.1592779

"Digital fluency" should mean designing, creating, and remixing, not just browsing, chatting, and interacting.

BY MITCHEL RESNICK, JOHN MALONEY, ANDRÉS MONROY-HERNÁNDEZ, NATALIE RUSK, EVELYN EASTMOND, KAREN BRENNAN, AMON MILLNER, ERIC ROSENBAUM, JAY SILVER, BRIAN SILVERMAN, AND YASMIN KAFAI

Scratch: Programming for All

WHEN MOSHE Y. VARDI, Editor-in-Chief of *Communications*, invited us to submit an article, he recalled how he first learned about Scratch: "A colleague of mine (CS faculty)," he said, "told me how she tried to get her 10-year-old daughter interested in programming, and the only thing that appealed to her was Scratch."

That's what we were hoping for when we set out to develop Scratch six years ago. We wanted to develop an approach to programming that would appeal to people who hadn't previously imagined themselves as programmers. We wanted to make it easy for everyone, of all ages, backgrounds, and interests, to program their own interactive stories, games, animations, and simulations, and share their creations with one another.

Since the public launch in May 2007, the Scratch Web site (http://scratch.mit.edu) has become a vibrant online community, with people sharing, discussing, and remixing one another's projects. Scratch has been called "the YouTube of interactive media." Each day, Scratchers from around the world upload more than 1,500 new projects to the site, with source code freely available for sharing and remixing. The site's collection of projects is wildly diverse, including video games, interactive newsletters, science simulations, virtual tours, birthday cards, animated dance contests, and interactive tutorials, all programmed in Scratch.

The core audience on the site is between the ages of eight and 16 (peaking at 12), though a sizeable group of adults participates as well. As Scratchers program and share interactive projects, they learn important mathematical and computational concepts, as well as how to think creatively, reason systematically, and work collaboratively: all essential skills for the 21st century. Indeed, our primary goal is not to prepare people for careers as professional programmers but to nurture a new generation of creative, systematic thinkers comfortable using programming to express their ideas.

In this article, we discuss the design principles that guided our development of Scratch and our strategies for making programming accessible and engaging for everyone. But first, to give a sense of how Scratch is being used, we describe a series of projects developed by a 13-year-old girl with the Scratch screen name BalaBethany.

BalaBethany enjoys drawing anime characters. So when she started using Scratch, it was natural for her to program animated stories featuring these characters. She began sharing her projects on the Scratch Web site, and other members of the community responded positively, posting glowing comments under her projects (such as "Awesome!" and "OMG I LUV IT!!!!!!"), along with questions about how she achieved certain visual effects (such as "How do you make a sprite look see-through?"). Encouraged, BalaBethany then created and shared new Scratch projects on a regular basis, like episodes in a TV series.

ask What's your name? and wait
say join Hello answer for 2 secs
y position < -150
wait until touching > 80
switch to costume jump
say game over
NEO
Click To start
I hope to add some measure of grace to the world.
KICKS N FLIPS
Archery
glasses
CAN ANYONE TELL ME WHERE I AM!?!!!

Figure 1. Screenshots from BalaBethany's anime series, contest, and tutorial.

She periodically added new characters to her series and at one point asked why not involve the whole Scratch community in the process? She created and uploaded a new Scratch project that announced a "contest," asking other community members to design a sister for one of her characters (see Figure 1). The project listed a set of requirements for the new character, including "Must have red or blue hair, please choose" and "Has to have either cat or ram horns, or a combo of both."

The project received more than 100 comments. One was from a community member who wanted to enter the contest but said she didn't know how to draw anime characters. So BalaBethany produced another Scratch project, a step-by-step tutorial, demonstrating a 13-step process for drawing and coloring anime characters.

Over the course of a year, BalaBethany programmed and shared more than 200 Scratch projects, covering a range of project types (stories, contests, tutorials, and more). Her programming and artistic skills progressed, and her projects clearly resonated with the Scratch community, receiving more than 12,000 comments.

Figure 2. Sample Scratch scripts.

Why Programming?

It has become commonplace to refer to young people as "digital natives" due to their apparent fluency with digital technologies.[15] Indeed, many young people are very comfortable sending text messages, playing online games, and browsing the Web. But does that really make them fluent with new technologies? Though they interact with digital media all the time, few are able to create their own games, animations, or simulations. It's as if they can "read" but not "write."

As we see it, digital fluency requires not just the ability to chat, browse, and interact but also the ability to design, create, and invent with new media,[16] as BalaBethany did in her projects. To do so, you need to learn some type of programming. The ability to program provides important benefits. For example, it greatly expands the range of what you can create (and how you can express yourself) with the computer. It also expands the range of what you can learn. In particular, programming supports "computational thinking," helping you learn important problem-solving and design strategies (such as modularization and iterative design) that carry over to nonprogramming domains.[18] And since programming involves the creation of external representations of your problem-solving processes, programming provides you with opportunities to reflect on your own thinking, even to think about thinking itself.[2]

Previous Research

When personal computers were first introduced in the late 1970s and 1980s, there was initial enthusiasm for teaching all children how to program. Thousands of schools taught millions of students to write simple programs in Logo or Basic. Seymour Papert's 1980 book *Mindstorms*[13] presented Logo as a cornerstone for rethinking approaches to education and learning. Though some children and teachers were energized and transformed by these new possibilities, most schools soon shifted to other uses of computers. Since that time, computers have become pervasive in children's lives, but few learn to program. Today, most people view computer programming as a narrow, technical activity, appropriate for only a small segment of the population.

What happened to the initial enthusiasm for introducing programming to children? Why did Logo and other initiatives not live up to their early promise? There were several factors:

- Early programming languages were too difficult to use, and many children simply couldn't master the syntax of programming;
- Programming was often introduced with activities (such as generating lists of prime numbers and making simple line drawings) that were not connected to young people's interests or experiences; and
- Programming was often introduced in contexts where no one could provide guidance when things went wrong—or encourage deeper explorations when things went right.

Papert argued that programming languages should have a "low floor" (easy to get started) and a "high ceiling" (opportunities to create increasingly complex projects over time). In addition, languages need "wide walls" (supporting many different types of projects so people with many different interests and learning styles can all become engaged). Satisfying the triplet of low-floor/high-ceiling/wide-walls hasn't been easy.[3]

In recent years, new attempts have sought to introduce programming to children and teens.[7] Some use professional programming languages like Flash/ActionScript; others use new languages (such as Alice[7] and Squeak Etoys[5]) developed specifically for younger programmers. They have inspired and informed our work on Scratch. But we weren't fully satisfied with the existing options. In particular, we felt it was important to make the floor even lower and the walls even wider while still supporting development of computational thinking.

To achieve these goals, we established three core design principles for Scratch: Make it more tinkerable, more meaningful, and more social than other programming environments. In the following sections, we discuss how each of these principles guided our design of Scratch.

More Tinkerable

Our Lifelong Kindergarten research group at the MIT Media Lab (http://llk.media.mit.edu) has worked closely with the Lego Company (http://www.lego.com/) for many years, helping develop Lego Mindstorms and other robotics kits.[17] We have always been intrigued and inspired by the way children play and build with Lego bricks. Given a box full of them, they immediately start tinkering, snapping together a few bricks, and the emerging structure then gives them new ideas. As they play and build, plans and goals evolve organically, along with the structures and stories.

We wanted the process of programming in Scratch to have a similar feel. The Scratch grammar is based on a collection of graphical "programming blocks" children snap together to create programs (see Figure 2). As with Lego bricks, connectors on the blocks suggest how they should be put together. Children can start by simply tinkering with the bricks, snapping them together in different sequences and combinations to see what happens. There is none of the obscure syntax or punctuation of traditional programming languages. The floor is low and the experience playful.

Scratch blocks are shaped to fit together only in ways that make syntactic sense. Control structures (like `forever` and `repeat`) are C-shaped to suggest that blocks should be placed inside them. Blocks that output values are shaped according to the types of values they return: ovals for numbers and hexagons for Booleans. Conditional blocks (like `if` and `repeat-until`) have hexagon-shaped voids, indicating a Boolean is required.

The name "Scratch" itself highlights the idea of tinkering, as it comes from the scratching technique used by hip-hop disc jockeys, who tinker with music by spinning vinyl records back and forth with their hands, mixing music clips together in creative ways. In Scratch programming, the activity is similar, mixing graphics, animations, photos, music, and sound.

Scratch is designed to be highly interactive. Just click on a stack of blocks and it starts to execute its code immediately. You can even make changes to a stack as it is running, so it is easy to experiment with new ideas incrementally and iteratively. Want to create parallel threads? Simply create multiple stacks of blocks. Our goal is to make parallel execution as intuitive as sequential execution.

The scripting area in the Scratch interface is intended to be used like a physical desktop (see Figure 3). You can even leave extra blocks or stacks lying around in case you need them later. The implied message is that it's OK to be a little messy and experimental. Most programming languages (and computer science courses) privilege top-down planning over bottom-up tinkering. With Scratch, we want tinkerers to feel just as comfortable as planners.

The emphasis on iterative, incremental design is aligned with our own development style in creating Scratch. We selected Squeak as an implementation language since it is well-suited for

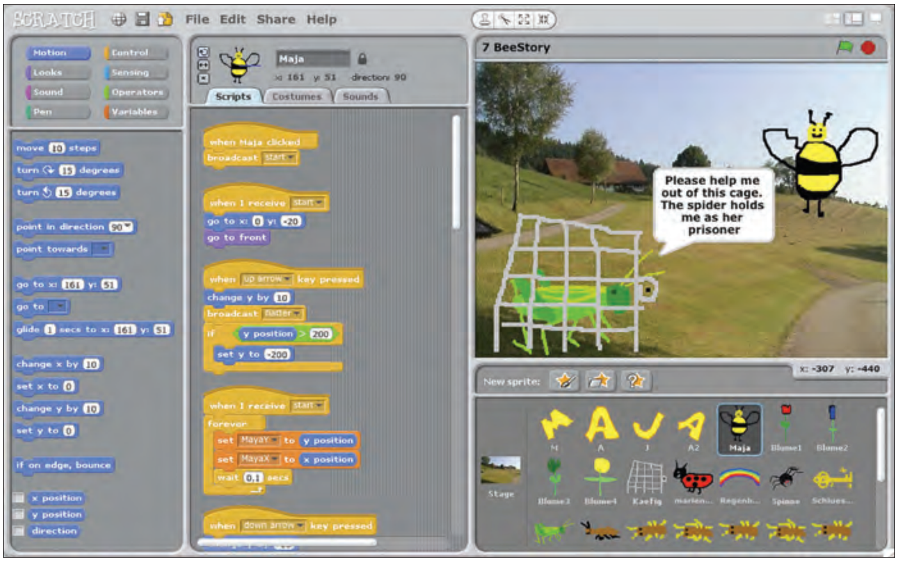


Figure 3. Scratch user interface.

Figure 4. Screenshots from sample Scratch projects.

rapid prototyping and iterative design. Before we launched Scratch in 2007, we continually field-tested prototypes in real-world settings, revising over and over based on feedback and suggestions from the field.[4]

More Meaningful

We know that people learn best, and enjoy most, when working on personally meaningful projects. So in developing Scratch, we put a high priority on two design criteria:

Diversity. Supporting many different types of projects (stories, games, animations, simulations), so people with widely varying interests are all able to work on projects they care about; and

Personalization. Making it easy for people to personalize their Scratch projects by importing photos and music clips, recording voices, and creating graphics.[14]

These priorities influenced many of our design decisions. For example, we decided to focus on 2D images, rather than 3D, since it is much easier for people to create, import, and personalize 2D artwork. While some people might see the 2D style of Scratch projects as somewhat outdated, Scratch projects collectively exhibit a visual diversity and personalization missing from 3D authoring environments.

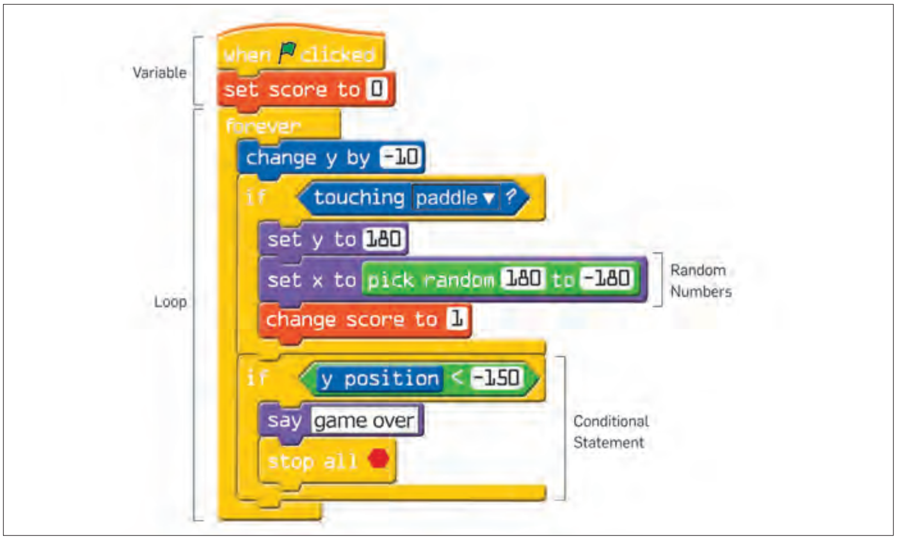


Figure 5. Sample Scratch script (from Pong-like paddle game) highlighting computational and mathematical concepts.

The value of personalization is captured nicely in this blog post from a computer scientist who introduced Scratch to his two children: "I have to admit that I initially didn't get why a kids' programming language should be so media-centric, but after seeing my kids interact with Scratch it became much more clear to me. One of the nicest things I saw with Scratch was that it personalized the development experience in new ways by making it easy for my kids to add personalized content and actively participate in the development process. Not only could they develop abstract programs to do mindless things with a cat or a box, etc… but they could add their own pictures and their own voices to the Scratch environment, which has given them hours of fun and driven them to learn."

We continue to be amazed by the diversity of projects that appear on the Scratch Web site. As expected, there are lots of games, ranging from painstakingly recreated versions of favorite video games (such as *Donkey Kong*) to totally original games. But there are many other genres, too (see Figure 4). Some Scratch projects document life experiences (such as a family vacation in Florida); others document imaginary wished-for experiences (such as a trip to meet other Scratchers). Some Scratch projects (such as birthday cards and messages of appreciation) are intended to cultivate relationships. Others are designed to raise awareness on social issues (such as global warming and animal abuse). During the 2008 U.S. presidential election, a flurry of projects featured Barack Obama and John McCain and later a series of projects promoted members of the Scratch online community for the not-quite-defined position of "President of Scratch."

Some Scratch projects grow out of school activities. For an Earth-science class, a 13-year-old boy from India created a project in which an animated character travels to the center of the Earth, with a voice-over describing the different layers along the way. As part of a social-studies class, a 14-year-old boy from New Jersey created a simulation of life on the island of Rapa Nui, designed to help others learn about the local culture and economy.

As Scratchers work on personally meaningful projects, we find they are ready and eager to learn important mathematical and computational concepts related to their projects (see Figure 5). Consider Raul, a 13-year-old boy who used Scratch to program an interactive game in his after-school center.[9] He created the graphics and basic actions for the game but didn't know how to keep score. So when a researcher on our team visited the center, Raul asked him for help. The researcher showed Raul how to create a variable in Scratch, and Raul immediately saw how he could use it for keeping score. He began playing with the blocks for incrementing variables, then reached out and shook the researcher's hand, saying "Thank you, thank you, thank you." The researcher wondered how many eighth-grade algebra teachers get thanked by their students for teaching them about variables?

Three core design principles for Scratch: Make it more tinkerable, more meaningful, and more social than other programming environments.

More Social

Development of the Scratch programming language is tightly coupled with development of the Scratch Web site.[12] For Scratch to succeed, the language needs to be linked to a community where people can support, collaborate, and critique one another and build on one another's work.[1]

The concept of sharing is built into the Scratch user interface, with a prominent "Share" menu and icon at the top of the screen. Click the Share icon and your project is uploaded to the Scratch Web site (see Figure 6) where it is displayed at the top of the page, along with the "Newest Projects." Once a project is on the Web site, anyone can run it in a browser (using a Java-based player), comment on it, vote for it (by clicking the "Love It?" button), or download it to view and revise the scripts. (All projects shared on the site are covered by Creative Commons license.)

In the 27 months following the Scratch launch, more than 500,000 projects were shared on the Scratch Web site. For many Scratchers, the opportunity to put their projects in front of a large audience—and receive feedback and advice from other Scratchers—is strong motivation. The large library of projects on the site also serves as inspiration. By exploring projects there, Scratchers get ideas for new projects and learn new programming techniques. Marvin Minsky once said that Logo had a great grammar but not much literature.[11] Whereas young writers are often inspired by reading great works of literature, there was no analogous library of great Logo projects to inspire young programmers. The Scratch Web site is the beginning of a "literature" for Scratch.

The site is also fertile ground for collaboration. Community members are constantly borrowing, adapting, and building on one another's ideas, images, and programs. Over 15% of the projects there are remixes of other projects on the site. For example, there are dozens of versions of the game Tetris, as Scratchers continue to add new features and try to improve gameplay. There are also dozens of dress-up-doll projects, petitions, and contests, all adapted from previous Scratch projects.

At first, some Scratchers were upset when their projects were remixed, complaining that others were "stealing" from them. That led to discussions on the Web site's forums about the value of sharing and the ideas behind open source communities. Our goal is to create a culture in which Scratchers feel proud, not upset, when their projects are adapted and remixed by others. We have continually added new features to the site to support and encourage this mind-set. Now, when someone remixes

The Scratch Web site has become a vibrant online community, with people sharing, discussing, and remixing one another's projects.

a project, the site automatically adds a link back to the original project, so the original author gets credit. Also, each project includes links to its "derivatives" (projects remixed from it), and the "Top Remixed" projects are featured prominently on the Scratch homepage.

Some projects focus on the site itself, providing reviews and analyses of other projects there. One early example was called SNN, for Scratch News Network, featuring the Scratch cat (the default character in Scratch) delivering news about the Scratch community, much like a CNN anchor. At first, we saw it as a "simulated newscast" but then realized it was a real newscast, providing news of interest to a real community—the Scratch online community. The SNN project inspired others, leading to a proliferation of online newsletters, magazines, and TV shows, all programmed in Scratch, reporting on the Scratch community.

Other Scratchers formed online "companies," working together to create projects that their individual members could not have produced on their own. One company got its start when a 15-year-old girl from England, with screen name BeeBop, created a project full of animated sprites and encouraged others to use them in their projects or place special requests for custom-made sprites. She was setting up a no-fee consulting business. A 10-year-old girl, also from England, with screen name MusicalMoon, liked BeeBop's animations and asked if she'd be willing to create a background for one of her projects. This collaboration gave rise to Mesh Inc., a self-proclaimed "miniature company" to produce "top quality games" in Scratch. A few days later, a 14-year-old boy from New Jersey, screen name Hobbit, discovered the Mesh Inc. gallery and offered his services, saying, "I'm a fairly good programmer, and I could help with debugging and stuff." Later, an 11-year-old boy from Ireland, with screen name Marty, was added to the Mesh Inc. staff due to his expertise in scrolling backgrounds.

Such collaborations open opportunities for many different types of learning. Here's how a 13-year-old girl from California, who started a Scratch company called Blue Elk Productions, described her experience:

"What is fun about Scratch and about organizing a company to write games together is that I've made a lot of friends and learned lots of new things. I've learned a lot about different kinds of programming by looking at other games with interesting effects, downloading them, and looking at and modifying the scripts and sprites. I really like programming! Also, when I started with Scratch I didn't think I was a very good artist. But since then, just by looking at other people's art projects, asking them questions, and practicing drawing using programs like Photoshop and the Scratch paint editor, I've gotten a lot better at art… Another thing I've learned while organizing Blue Elk is how to help keep a group of people motivated and working together… I like Scratch better than blogs or social networking sites like Facebook because we're creating interesting games and projects that are fun to play, watch, and download. I don't like to just talk to other people online, I like to talk about something creative and new."

To encourage international sharing and collaboration, we've placed a high priority on translating Scratch into multiple languages. We created an infrastructure that allows the Scratch programming blocks to be translated into any language with any character set. A global network of volunteers has provided translations for more than 40 languages. Children around the world now share Scratch projects with one another, each viewing the Scratch programming blocks in their own language.

Future Directions

A growing number of K–12 schools around the world, and even some universities (including Harvard and the University of California, Berkeley),[8] use Scratch as a first step into programming. A natural question is What comes next? In the Scratch discussion forums, there are ongoing debates about what programming language should be used after Scratch. We receive many requests to add more advanced features to Scratch (such as object inheritance and recursive list structures), hoping that Scratch itself could be the "next step."

We plan to keep our primary focus on lowering the floor and widening the walls, not raising the ceiling. For some Scratchers, especially those who want to pursue a career in programming or computer science, it is important to move on to other languages. But for many other Scratchers, who see programming as a medium for expression, not a path toward a career, Scratch is sufficient for their needs. With Scratch, they can continue to experiment with new forms of self-expression, producing a diverse range of projects while deepening their understanding of a core set of computational ideas. A little bit of programming goes a long way.

As we develop future versions, our goal is to make Scratch even more tinkerable, meaningful, and social. With our Scratch Sensor Board (http://info.scratch.mit.edu/Sensor_Boards), people can create Scratch projects that sense and react to events in the physical world. We are also developing a version of Scratch that runs on mobile devices and a Web-based version that enables people to access online data and program online activities.

Probably the biggest challenges for Scratch are not technological but cultural and educational.[10] Scratch has been a success among early adopters, but we need to provide better educational support for it to spread more broadly. We recently launched a new online community, called ScratchEd (http://scratched.media.mit.edu), where educators share their ideas, experiences, and lesson plans for Scratch. More broadly, there needs to be a shift in how people think about programming, and about computers in general. We need to expand the notion of "digital fluency" to include designing and creating, not just browsing and interacting. Only then will initiatives like Scratch have a chance to live up to their full potential.

Acknowledgments

Many people have contributed to the development of Scratch and even more to the ideas underlying Scratch. We'd like to thank friends and former members of the Lifelong Kindergarten group who have worked on Scratch, especially Tammy Stern, Dave Feinberg, Han Xu, Margarita Dekoli, Leo Burd, Oren Zuckerman, Nick Bushak, and Paula Bonta. We are grateful to Kylie Peppler, Grace Chui, and other members of Yasmin Kafai's research team, who conducted and participated in field studies in Scratch's early development. Scratch was deeply influenced and inspired by the work of Seymour Papert and Alan Kay. We appreciate financial support from the National Science Foundation (grant ITR-0325828), Microsoft, Intel Foundation, Nokia, and MIT Media Lab research consortia. The names of all children mentioned here are pseudonyms.

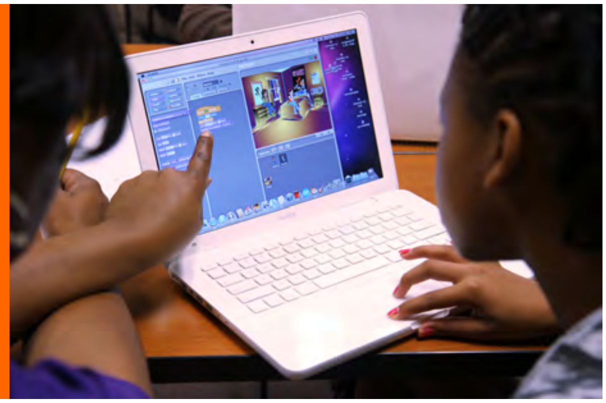
Figure 6. Scratch Web site.

References

1. Bransford, J., Brown, A., and Cocking, R. *How People Learn: Mind, Brain, Experience, and School.* National Academies Press, Washington, D.C., 2000.
2. diSessa, A. *Changing Minds: Computers, Learning, and Literacy.* MIT Press, Cambridge, MA, 2000.
3. Guzdial, M. Programming environments for novices. In *Computer Science Education Research*, S. Fincher and M. Petre, Eds. Taylor & Francis, Abingdon, U.K., 2004, 127–154.
4. Kafai, Y., Peppler, K., and Chiu, G. High-tech programmers in low-income communities: Seeding reform in a community technology center. In *Communities and Technologies*, C. Steinfield, B. Pentland, M. Ackerman, and N. Contractor, Eds. Springer, New York, 2007, 545–564.
5. Kay, A. Squeak etoys, children, and learning; http://www.squeakland.org/resources/articles.
6. Kelleher, C. and Pausch, R. Using storytelling to motivate programming. *Commun. ACM 50*, 7 (July 2007), 58–64.
7. Kelleher, C. and Pausch, R. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys 37*, 2 (June 2005), 83–137.
8. Malan, D. and Leitner, H. Scratch for budding computer scientists. *ACM SIGCSE Bulletin 39*, 1 (Mar. 2007), 223–227.
9. Maloney, J., Peppler, K., Kafai, Y., Resnick, M., and Rusk, N. Programming by choice: Urban youth learning programming with Scratch. *ACM SIGCSE Bulletin 40*, 1 (Mar. 2008), 367–371.
10. Margolis, J. *Stuck in the Shallow End: Education, Race, and Computing.* MIT Press, Cambridge, MA, 2008.
11. Minsky, M. *Introduction to LogoWorks. In LogoWorks: Challenging Programs in Logo*, C. Solomon, M. Minsky, and B. Harvey, Eds. McGraw-Hill, New York, 1986.
12. Monroy-Hernández, A. and Resnick, M. Empowering kids to create and share programmable media. *Interactions 15*, 2 (Mar.–Apr. 2008), 50–53.
13. Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books, New York, 1980.
14. Peppler, K. and Kafai, Y. From SuperGoo to Scratch: Exploring creative media production in informal learning. *Journal on Learning, Media, and Technology 32*, 7 (2007), 149–166.
15. Prensky, M. Digital natives, digital immigrants. *On the Horizon 9*, 5 (Oct. 2001), 1–6.
16. Resnick, M. Sowing the seeds for a more creative society. *Learning and Leading with Technology* (Dec. 2007), 18–22.
17. Resnick, M. Behavior construction kits. *Commun. ACM 36*, 7 (July 1993), 64–71.
18. Wing, J. Computational thinking. *Commun. ACM 49*, 3 (Mar. 2006), 33–35.

Mitchel Resnick, **John Maloney**, **Andrés Monroy-Hernández**, **Natalie Rusk**, **Evelyn Eastmond**, **Karen Brennan**, **Amon Millner**, **Eric Rosenbaum**, and **Jay Silver** are all researchers and members of the Scratch Team (http://scratch.mit.edu) at the Media Laboratory of the Massachusetts Institute of Technology, Cambridge, MA. **Brian Silverman** is president of the Playful Invention Company, Montreal, Quebec, Canada. **Yasmin Kafai** is a professor in the Graduate School of Education of the University of Pennsylvania, Philadelphia, PA.

Learn to Code
Code to Learn



Is it important for all children to learn how to write? Very few children grow up to be journalists, novelists, or professional writers. So why should everyone learn to write?

Of course, such questions seem silly. People use writing in all parts of their lives: to send birthday messages to friends, to jot down shopping lists, to record personal feelings in diaries. The act of writing also engages people in new ways of thinking. As people write, they learn to organize, refine, and reflect on their ideas. Clearly, there are powerful reasons for everyone to learn to write.

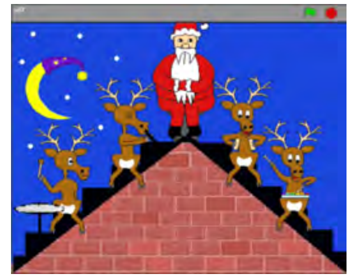
I see coding (computer programming) as an extension of writing. The ability to code allows you to "write" new types of things – interactive stories, games, animations, and simulations. And, as with traditional writing, there are powerful reasons for everyone to learn to code.

Recently, there has been a surge of interest in learning to code, focusing especially on career opportunities. It is easy to understand why: the number of jobs for programmers and computer scientists is growing rapidly, with demand far outpacing supply.

But I see much deeper and broader reasons for learning to code. In the process of learning to code, people learn many other things. They are not just learning to code, they are coding to learn. In addition to learning mathematical and computational ideas (such as variables and conditionals), they are also learning strategies for solving problems, designing projects, and communicating ideas. These skills useful not just for computer scientists but for everyone, regardless of age, interests, or occupation.

In May 2007, my research group at the MIT Media Lab launched the Scratch programming language and online community (http://scratch.mit.edu) in an effort to make coding accessible and appealing to everyone. Since then, young people (ages 8 and up) have shared more than 4.5 million projects on the Scratch website, with thousands of new projects added every day. Scratch is used in many contexts (homes, schools, libraries, community centers), at many age levels (from elementary school to college), and across many disciplines (math, computer science, language arts, social studies).

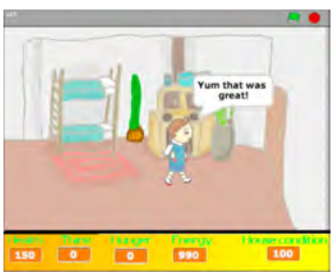
We’ve been amazed with the diversity and creativity of the projects. Take a look at the Scratch website and you’ll find animated stories, virtual tours, science simulations, public-service announcements, multimedia art projects, online newsletters, interactive tutorials, and much more.



As an example, let me describe some of the projects created by a young Scratcher who I’ll call BlueSaturn. When BlueSaturn started using Scratch at age 12, one of her first projects was a Christmas card with cartoon images of Santa and his reindeer. Each reindeer was holding a musical instrument and, when clicked, played a different part of the song “We wish you a merry Christmas.” BlueSaturn sent her friends a link to the project as holiday greeting.

As she worked on the Christmas card, BlueSaturn realized that what she enjoyed most was creating animated characters. So she developed a project that featured a series of different animated characters: dinosaurs, dragons, flying horses. In the Project Notes, she encouraged other members of the community to make use of her characters in their own projects – and she offered to make custom characters upon request. In effect, BlueSaturn was setting up a consulting service. We had never imagined that the Scratch website would be used this way.

One community member wanted a cheetah for his Scratch project, so BlueSaturn made an animated cheetah, based on a video that she saw on a National Geographic site. For another community member, BlueSaturn created a bird with flapping wings – and then she posted a step-by-step tutorial showing how she had created the animation.



BlueSaturn became well-known in the community, and she began to receive requests to join collaborative teams, or “collabs” as they are often known in the Scratch community. In one collab, BlueSaturn worked with four other young people from three different countries to produce an elaborate adventure game. BlueSaturn created animated characters while other members of the collab developed game scenarios, created music and sound effects, and drew backgrounds.

In the process of working on these projects, BlueSaturn certainly learned coding skills, but she also learned many other things. She learned how to divide complex problems into simpler parts, how to iteratively refine her designs, how to identify and fix bugs, how to share and collaborate with others, how to persevere in the face of challenges.

We find that active members of the Scratch community start to think of themselves differently. They begin to see themselves as creators and designers, as people who can make things and express themselves with digital media, not just browse, chat, and play games. While many people can *read* digital media, Scratchers can *write* digital media – and are thus prepared to become full participants in today’s digital society.

Mitchel Resnick
Professor of Learning Research
MIT Media Lab

Published in EdSurge
May 2013