Making Programming Universally Accessible and Useful

Mitchel Resnick
Media Laboratory | Massachusetts Institute of Technology
20 Ames Street | Room 020A | Cambridge, MA 02142-1308
mres@media.mit.edu | 617-253-9783 | http://www.media.mit.edu/~mres

Research Abstract and Goals

Scratch (http://scratch.mit.edu) has been called the "YouTube of interactive media." With its building-block programming approach and online community, Scratch makes it easy for everyone to program and share interactive games, stories, and animations – and, in the process, learn important computational ideas and problem-solving skills. But there is still work to be done: the Scratch programming application (running locally) and Scratch online community (on the web) live in different worlds, limiting possibilities for sharing and collaboration.

We propose to design a new version of Scratch – called Scratch 2.0 – that is more fully integrated into the web, making it possible for people to author Scratch programs directly on websites, program interactions between Scratch and other social media websites, access and output online data, and experiment with new forms of online collaboration. In short, our goal is to explore ways of migrating Scratch to the cloud, so that programming becomes more accessible and useful for everyone.

Keywords: Education Innovation, Human-Computer Interaction

Technical Description

Background. In 2003, our research group at the MIT Media Lab was awarded a four-year grant from the National Science Foundation to develop a new programming platform, called Scratch, that enables young people to create their own interactive stories, games, animations, and simulations – and share their creations with online. The Scratch website (http://scratch.mit.edu), launched in May 2007, has become a vibrant online community, with people sharing, discussing, and remixing one another's Scratch projects. Each day, members of the Scratch community (mostly ages 8 to 16) upload roughly 1500 new Scratch projects to the site – on average, a new project every minute. The collection of projects is incredibly diverse: interactive newsletters, science simulations, virtual tours, animated dance contests, visual tutorials, and many other themes, with code for each project available for everyone to view and remix.

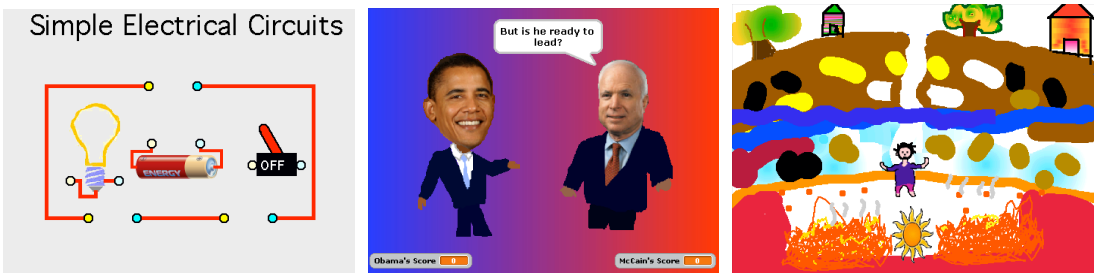

Figure 1: Screenshots from sample Scratch projects

The process of programming in Scratch feels somewhat like tinkering with LEGO bricks. The programming grammar is based on a collection of graphical "programming blocks" that Scratchers snap together to create programs. As with LEGO bricks, connectors on the blocks suggest how they should be put together. There is none of the obscure syntax of traditional programming languages. As young people program and share projects with Scratch, they begin to develop as *computational thinkers*: they learn important computational concepts, while also learning strategies for designing, problem solving, and collaborating. Scratch is increasingly used not only in K-12 education, but also in introductory computer-science courses at universities (including Harvard and Berkeley).

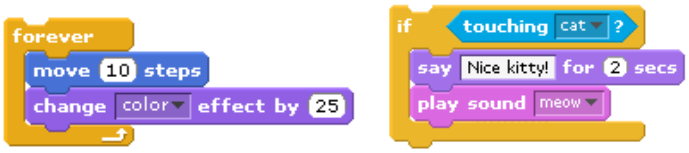

Figure 2: Sample Scratch scripts

Our research group has already collaborated with Google on several Scratch-related initiatives. In July 2009, we organized a three-day CS4HS workshop based on Scratch (http://cs4hs.media.mit.edu). A member of the Google India team recently asked us for permission to distribute Scratch as part of a student outreach effort in India. A former member of our group (Tammy Stern), who now works for Google in New York, has organized several Google-led Scratch workshops for K-12 students in New York. And Scratch served as an inspiration for Hal Abelson's App Inventor project at Google.

Research Plan. The current version of Scratch is two separate pieces of software: a stand-alone programming environment running on a local machine (implemented in Squeak, a version of Smalltalk), and a web-based online community where people share and discuss their Scratch projects (implemented in PHP/MySQL, with a Java-based applet serving as the player for Scratch projects). People create projects locally, then share on the web. The two worlds are connected only through upload and download.

We propose to design a new version of Scratch – called Scratch 2.0 – that is more fully integrated into the cloud. We will design Scratch 2.0 to make the process of sharing and collaborating on Scratch projects more seamless, and to support new categories of Scratch projects that leverage social media and resonate more strongly with the interests of today's youth. Over the next year, we propose to:

- *Design and implement a web-based version of the Scratch programming language.* There are several obvious candidate technologies (including JavaScript/HTML5). We will experiment with several different approaches to see which best fits our needs.
- *Design and implement a new back-end for the Scratch online community*. The Scratch website should be tightly integrated with the Scratch authoring environment – and also with other social media websites, to facilitate sharing of media and data across sites. Our current plan is to use an easily scalable service like the Google App Engine.
- *Explore integration of Scratch with App Inventor*. We have had initial discussions with Hal Abelson and Mark Friedman of Google's App Inventor team about ways to integrate our projects, to create a system that works across desktops and phones.

- *Explore new frameworks for sharing*. The current Scratch website is designed primarily to support sharing of Scratch projects. For Scratch 2.0, we want to explore sharing at many different levels of granularity: scripts, graphics, sounds, sprites.
- *Explore new forms of communication*. How should Scratch projects communicate with one another – and with other applications and social media websites? What if you could text a message or Twitter update to an online Scratch project? We will follow up Tammy Stern's NetScratch research, exploring the concept of "shariables" (variables shared between Scratch projects) and "web sensors" (enabling Scratch projects to access real-time data from the web).
- *Expand the family of Scratch applications*. Currently, there is a single Scratch application and a single online community. We will experiment with different variants of Scratch for different audiences (e.g., Scratch for design students or Scratch for kindergarten students). As part of this effort, we will develop protocols to enable other developers to integrate Scratch programming blocks into their own applications.

Expected Outcomes and Results

Over the course of the year, we will go through several iterations of prototyping and field testing. In the process we will develop:

- A better understanding of how to migrate end-user programming to the cloud
- A web-based version of Scratch that is (almost) ready for wider dissemination
- A new framework for sharing Scratch scripts, sprites, media, and projects online
- New strategies for communication with and between online Scratch projects
- Ideas on integrating Scratch and App Inventor, to run on multiple platforms
- Sample projects highlighting the benefits and new possibilities of web-based Scratch

Budget

We are requesting total funding of $147,392. The funding will be used to support two PhD students at the MIT Media Lab for one year.

RA stipend ($28,703/student)	$ 57,406
Laptop ($2,000/student)	$ 4,000
Travel to conference ($1,000/student)	$ 2,000
Travel to Google ($1,000/student)	$ 2,000
MIT Overhead	$ 44,476
Tuition ($18,755/student)	$ 37,510
Total	**$147,392**

(Note: Tuition for the RAs is budgeted at 50% of full tuition; MIT pays the other 50%.)

Google Contacts

Mark Friedman has agreed to serve as Google Technical Sponsor for this project. Several others have also agreed to serve as Google contacts: Hal Abelson (Visiting Faculty at Google), Leslie Yeh Johnson (who supported our CS4HS Scratch workshop), Tammy Stern (who worked with our MIT Scratch Team before taking a job at Google).