GIVE P’S A CHANCE: PROJECTS, PEERS, PASSION, PLAY

Mitchel Resnick

Abstract
To thrive in today’s rapidly-changing world, young people must learn to think and act creatively. This paper discusses how our Lifelong Kindergarten group at the MIT Media Lab designed the Scratch programming language and online community with the explicit goal of helping young people to develop as creative thinkers. The paper analyzes the design of Scratch through the lens of the Four P’s of Creative Learning: Projects, Peers, Passion, and Play.

Keywords Scratch, coding, programming, constructionism, creativity

1. Introduction

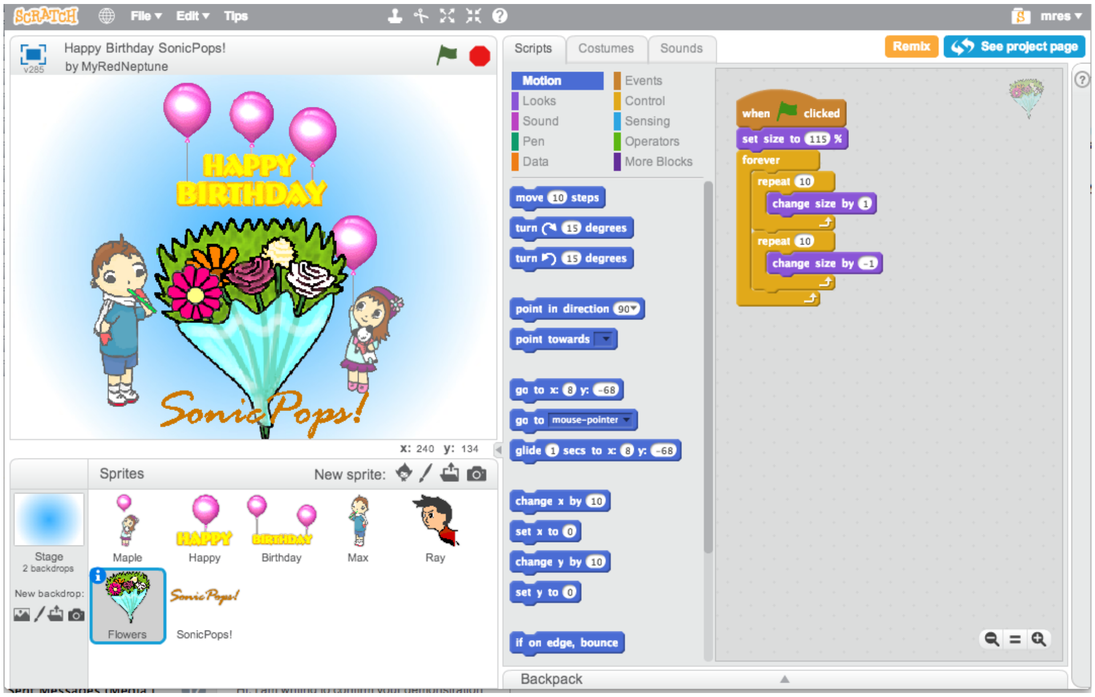
We live in a world that is changing more rapidly than ever before. Today’s children will face a continual stream of new issues and unexpected challenges in the future. Many things that they learn today will be obsolete tomorrow. To thrive, they must learn to design innovative solutions to the unexpected problems that will undoubtedly arise in their lives. Their success and satisfaction will be based on their ability to think and act creatively. Knowledge alone is not enough: they must learn how to use their knowledge creatively.

How can we help today’s young people grow up as creative thinkers? My Lifelong Kindergarten research group (http://llk.media.mit.edu) at the MIT Media Lab has been developing new technologies, activities, and strategies to engage young people in creative learning experiences, so that they can develop as creative thinkers. Our approach is based on four core elements, which we sometimes call the Four P’s of Creative Learning:

- *Projects*. People learn best when they are actively working on meaningful projects – generating new ideas, designing prototypes, refining iteratively.
- *Peers*. Learning flourishes as a social activity, with people sharing ideas, collaborating on projects, and building on one another’s work.
- *Passion*. When people work on projects they care about, they work longer and harder, persist in the face of challenges, and learn more in the process.
- *Play*. Learning involves playful experimentation – trying new things, tinkering with materials, testing boundaries, taking risks, iterating again and again.

These four P’s are strongly aligned with (and inspired by) the Constructionist approach to education, which emphasizes the value of learners playfully creating personally-meaningful projects in collaboration with peers [2] [5].

This paper describes how our Lifelong Kindergarten group has used the Four P's of Creative Learning as guiding principles in our design of the Scratch programming language and online community (http://scratch.mit.edu). Since the public launch of Scratch in 2007, millions of young people around the world have used Scratch to program their own interactive stories, games, and animations – and share their creations with one another online [7] [8]. Scratch has been deeply inspired by the Logo programming language [4], but goes beyond Logo by making programming *more tinkerable* (providing a graphical building-block approach to programming – see Fig. 1), *more meaningful* (supporting a diversity of project genres, so that people with widely varying interests can all work on projects they care about), and *more social* (enabling learners to share, remix, comment on, and build upon one another's projects).

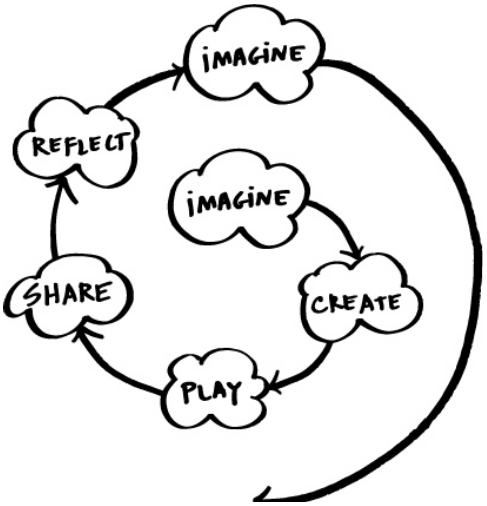


```
when ⚑ clicked
set size to 115 %
forever
    repeat 10
        change size by 1
    repeat 10
        change size by -1
```

Fig. 1: Scratch programming editor

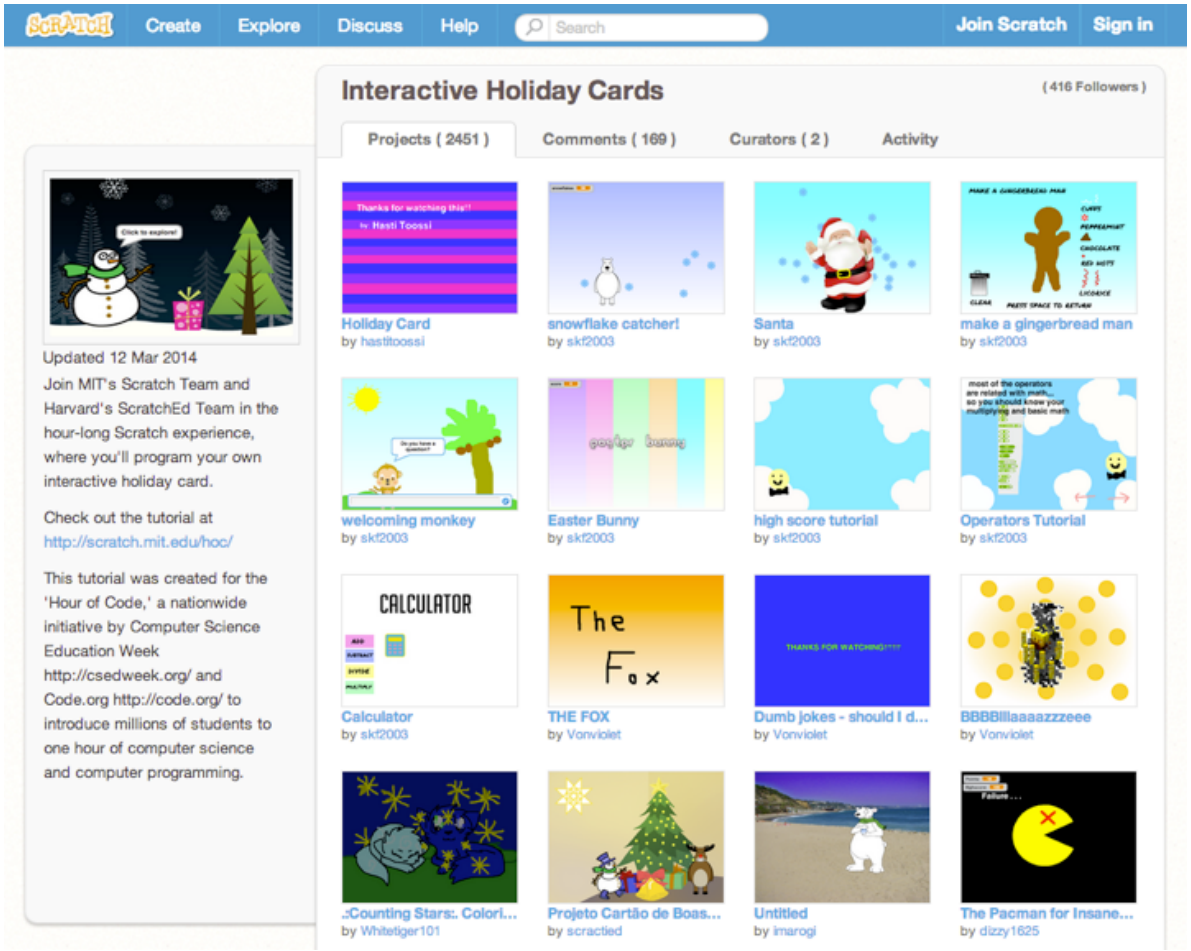
The following sections describe how the design of Scratch (and related activities) has been guided by the Four P's of Creative Learning: Projects, Peers, Passion, and Play.

2. Projects



Scratch was designed with projects in mind. Young people have created and shared more than 5 million projects in the Scratch online community, with roughly 10,000 new projects every day. As young people create Scratch projects, they engage in the "creative learning spiral" [6]: they *imagine* what they want to do, *create* a project based on their ideas, *play* with their creations, *share* their ideas and creations with others, *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. In the process, they develop as creative thinkers.

To us, it seems natural to introduce programming to young people in a project-oriented way, so that they learn to express themselves creatively as they learn to code. But many introductions to coding take a very different approach. Consider, for example, the introductory tutorials for the Hour of Code, a high-profile campaign to introduce young people to coding (http://csedweek.org). The Scratch tutorial for Hour of Code (created by Karen Brennan and colleagues) was explicitly project-oriented, helping young people create their own interactive holiday cards. Thousands of young people around the world programmed holiday cards and shared their creations on the Scratch website (Fig. 3).



Interactive Holiday Cards

(416 Followers)

Projects (2451) Comments (169) Curators (2) Activity

Updated 12 Mar 2014

Join MIT's Scratch Team and Harvard's ScratchEd Team in the hour-long Scratch experience, where you'll program your own interactive holiday card.

Check out the tutorial at http://scratch.mit.edu/hoc/

This tutorial was created for the 'Hour of Code,' a nationwide initiative by Computer Science Education Week http://csedweek.org/ and Code.org http://code.org/ to introduce millions of students to one hour of computer science and computer programming.

Holiday Card
by hastitoossi

snowflake catcher!
by skf2003

Santa
by skf2003

make a gingerbread man
by skf2003

welcoming monkey
by skf2003

Easter Bunny
by skf2003

high score tutorial
by skf2003

Operators Tutorial
by skf2003

Calculator
by skf2003

THE FOX
by Vonviolet

Dumb jokes - should I d...
by Vonviolet

BBBBlllaaaazzzzeee
by Vonviolet

.:Counting Stars:. Colori...
by Whitetiger101

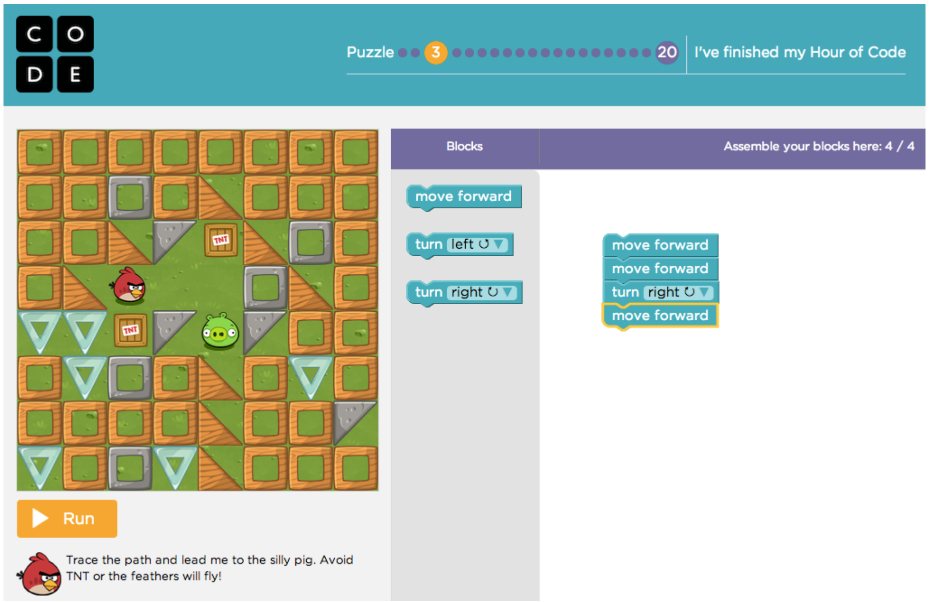
Projeto Cartão de Boas...
by scractied

Untitled
by imarogi

The Pacman for Insane...
by dizzy1625

Fig. 3: Scratch projects for Hour of Code

By contrast, most of the other Hour of Code tutorials focused on puzzles, not projects. These tutorials presented students with a series of logic puzzles in which they needed to program animated characters to move from one location to another (Fig. 4). When students successfully solved one puzzle, they could move on to the next. Students undoubtedly learned some useful computational concepts while working on these puzzles. But learning to code by solving logic puzzles is like learning to write by solving crossword puzzles. That's not the way to become truly fluent. Just as students developing fluency with language need opportunities to write stories (not just play word games), students developing fluency with coding need opportunities to create projects (not just solve puzzles).



Trace the path and lead me to the silly pig. Avoid TNT or the feathers will fly!

Fig. 4: Puzzle-based tutorial for Hour of Code

3. Peers

When we launched the Scratch programming language in 2007, we launched the Scratch online community at the same time. We believed, from the beginning, that interaction with peers should be a central element in the learning process.

The Scratch online community serves two roles. It provides an *audience*: when young people share their projects, they get feedback and suggestions from peers in the community. At the same time, the community provides *inspiration*: when young people try out projects made by their peers, they can get new ideas (and even borrow scripts and images) for their own projects [1].

We designed Scratch so that it would be easy for people to *remix* one another's projects [3]. All projects in the Scratch community are covered by a Creative Commons license, so that people are free to build on one another's work and ideas, as long as they give appropriate credit. More than a quarter of the projects on the Scratch website are remixes of other projects. The website provides visualizations to show how ideas spread among peers in the community (Fig. 5).

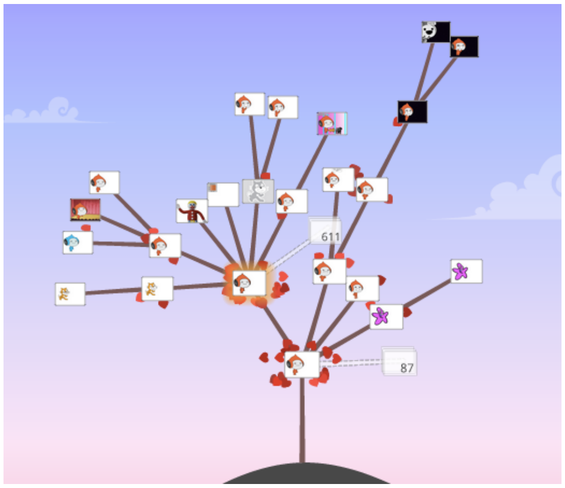
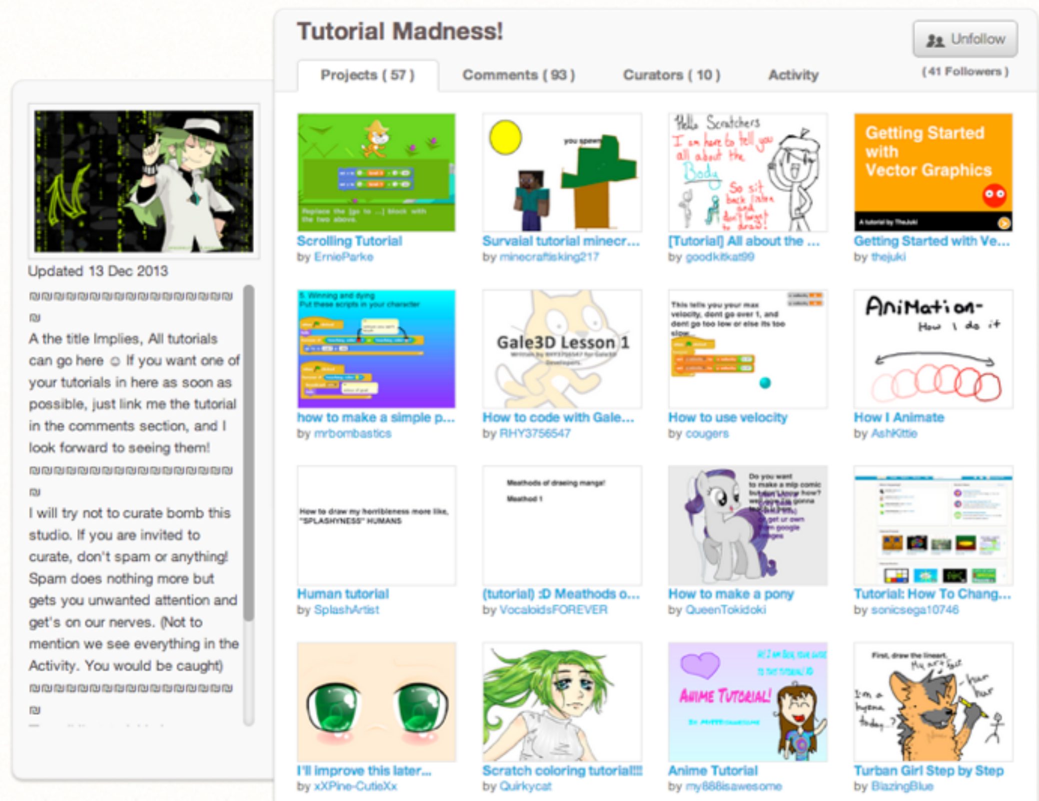


Fig. 5: Remix Tree

We have been surprised by the extent to which members of the Scratch community want to help one another. When we created Scratch, we expected that some teachers would create tutorials in Scratch, but we didn't expect young people to create tutorials. In fact, young people have created and shared thousands of tutorials, helping others learn how to code in Scratch, how to draw graphic characters in Scratch, and even how to make your projects popular on the Scratch website (Fig. 6). Young Scratchers have also created galleries where they can ask for (and offer) help with projects. It is this type of sharing and collaboration that makes Scratch a creative learning community.



Updated 13 Dec 2013

A the title Implies, All tutorials can go here ☺ If you want one of your tutorials in here as soon as possible, just link me the tutorial in the comments section, and I look forward to seeing them!

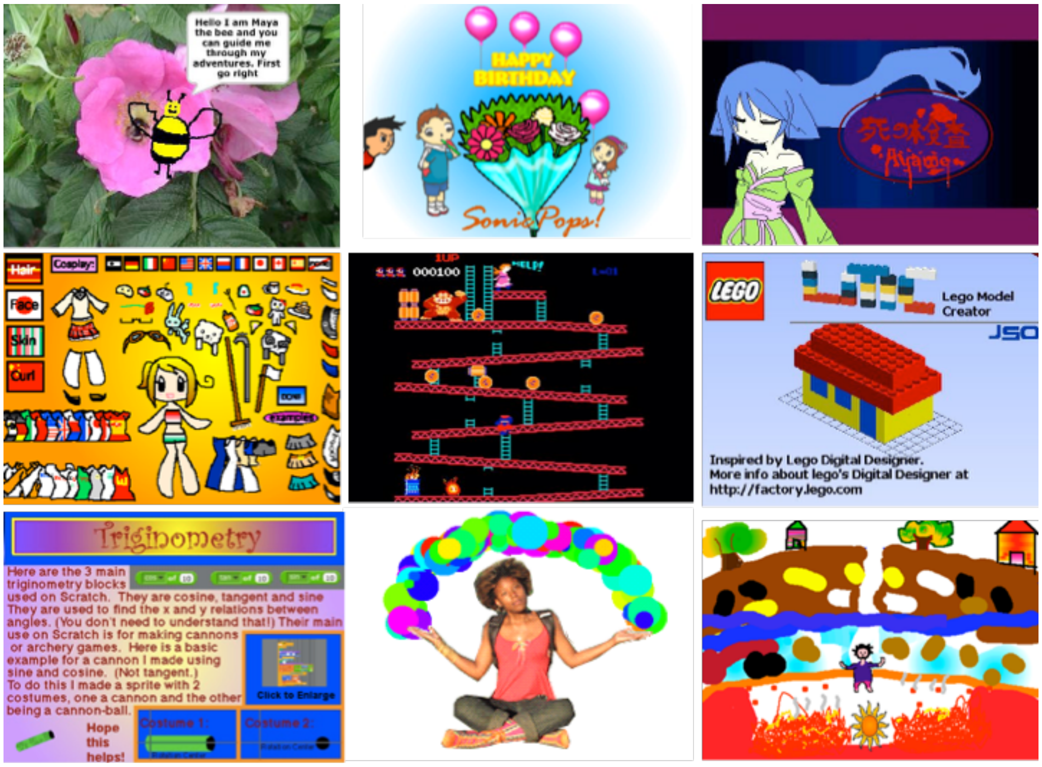
I will try not to curate bomb this studio. If you are invited to curate, don't spam or anything! Spam does nothing more but gets you unwanted attention and get's on our nerves. (Not to mention we see everything in the Activity. You would be caught)

Fig. 6: Tutorials by Scratchers for Scratchers

4. Passion

We're happy that young people have shared more than 5 million projects on the Scratch website. But we're even happier about the *diversity* of the projects on the website. Young people have shared not only stories and games, but also interactive birthday cards, anime comic strips, interactive newsletters, virtual tours, dance contests, public service announcements, and much more (Fig. 7).

Why is this diversity is important? It is an indication that young people are working on Scratch projects that they really care about. This diversity of projects is a reflection of the diversity of interests of young people. With Scratch, young people from many different backgrounds, with many different interests, can all work on projects that they care about. And when people work on projects that they care about, they tend to work harder and learn more.



Inspired by Lego Digital Designer. More info about lego's Digital Designer at http://factory.lego.com

Here are the 3 main triginometry blocks used on Scratch. They are cosine, tangent and sine They are used to find the x and y relations between angles. (You don't need to understand that!) Their main use on Scratch is for making cannons or archery games. Here is a basic example for a cannon I made using sine and cosine. (Not tangent.) To do this I made a sprite with 2 costumes, one a cannon and the other being a cannon-ball.

Fig. 7: Variety of Scratch projects

In designing Scratch, we emphasized the use of media (music, sounds, graphics, photos) since we know that many young people are passionate about media. Some parents and educators have been skeptical about the central role of media in Scratch projects, worrying that media will distract young people from the more "educational" aspects of Scratch. Here's a message that we received from a Scratch parent who was skeptical at first, but then gained an appreciation for the role of media in Scratch:

> *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 clearer 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.*

5. Play

When most people think about play, they think about fun and enjoyment. But when my research group thinks about play, we think about it somewhat differently. We think of play as an attitude and an approach for engaging with the world. We associate play with taking risks, trying new things, and testing boundaries. We see play as a process of tinkering, experimenting, and exploring. These aspects of play are central to the creative learning process.

In designing Scratch, we wanted to encourage young people to tinker, experiment, and play with their projects. We designed the Scratch programming blocks to be "tinkerable." You can easily snap the blocks together and take them apart, just like LEGO bricks. You can continually experiment and iterate your programming scripts.

Similarly, the Scratch website encourages young people to tinker and experiment with projects. You can drag scripts, images, and sounds from a project into your "backpack," then move them into a different project. You can also declare a project to be a "draft," to let others know that it is a work in progress.

For example, a Scratcher with the username EmeraldDragon started working on a game featuring a dragon that runs across the screen (Fig. 8a). In the notes, EmeraldDragon wrote: "This is all I have right now. I am working on being able to run back and forth without the rock disappearing. Any tips or help? This is just a stage in a long process." EmeraldDragon named the project *My Dragon Game (NOT finished)*.

A little while later, EmeraldDragon added a comment to the project: "I was just tinkering with the scripts in the game and i finally figured out how to make it so you can run back and forth! I'll fix up the game and put out the new and improved still not yet a game version."

Soon after, EmeraldDragon posted an improved version of the game, including a new process for keeping score (Fig. 8b). But EmeraldDragon wanted to emphasize that the project was still a work in progress. The title of the revised project: *My Dragon Game (Still NOT finished)*.

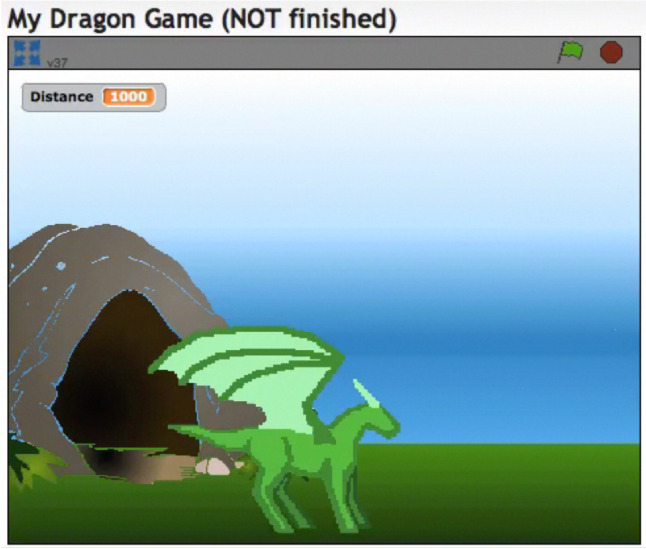
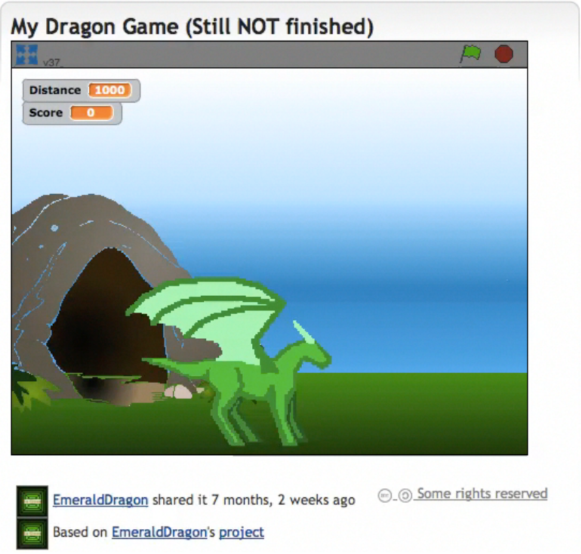


Fig. 8a



EmeraldDragon shared it 7 months, 2 weeks ago

Based on EmeraldDragon's project

Fig. 8b

6. All We are Saying is Give P's a Chance

In the past few years, there has been a surge of interest in making and coding. Maker Spaces and Coding Clubs are opening up everywhere. The enthusiasm surrounding the Maker Movement and the Coder Movement provides an opportunity for reinvigorating and revalidating the Constructionist tradition in education. But making and coding are not enough. To help young people prepare for a world that is changing more rapidly than ever before, we must embed making and coding in a creative-learning process characterized by Projects, Peers, Passion, and Play.

Acknowledgements

Many members of the Lifelong Kindergarten group at the MIT Media Lab contributed to the ideas and technologies described in this paper. In particular, the Four P's of Creative Learning framework was developed in close collaboration with Natalie Rusk and Philipp Schmidt, as part of our Learning Creative Learning online course (http://learn.media.mit.edu).

References

[1] K. Brennan, M. Resnick, and A. Monroy-Hernandez, "Making projects, making friends: Online community as catalyst for interactive media creation," in *New Directions for Youth Development*, vol. 128, 2010.

[2] Y. Kafai and M. Resnick, *Constructionism in practice: Designing, thinking, and learning in a digital world*. Lawrence Erlbaum, 1996.

[3] A. Monroy-Hernandez, *Designing for Remixing: Supporting an Online Community of Amateur Creators*. PhD dissertation, MIT Media Lab, 2012.

[4] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, 1980.

[5] S. Papert, "Situating Constructionism," in *Constructionism*, I. Harel and S. Papert, Eds. Ablex Publishing, 1991.

[6] M. Resnick, "All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten," in *ACM Creativity & Cognition conference*, 2007.

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

[8] M. Resnick, "Reviving Papert's Dream," in *Educational Technology*, vol. 52, no. 4, 2012, pp. 42-46.