New Pathways into Data Science

Extending the Scratch Programming Language to Enable Youth to Analyze and Visualize Their Own Learning

1. Introduction

Over the last few years, new methods of collecting, processing, and analyzing massive amounts of data from digital learning environments have led to new possibilities for understanding and evaluating trajectories of learners. These new data techniques and abilities have the potential to transform educational approaches and strategies for a variety of stakeholders, including teachers, administrators, policy-makers, researchers, and learners themselves (Romero & Ventura, 2007). Thanks to these new methods for capturing different aspects of learning trajectories, the focus for evaluating and understanding learning is shifting from the product of learning to the process of learning (e.g. Baker et al., 2010; Blikstein, 2011; Montalvo et al., 2010).

Although large data sets from digital learning environments are increasingly available, the investigation, exploration, and analysis of these data sets remains largely restricted to researchers with technical expertise in data analysis. There are some initiatives that allow learners to view data about their own learning trajectories through dashboards with graphs and statistics (Duval, 2011; Santos et al., 2012, 2013). But even in these cases, students are not provided with tools and techniques for analyzing and designing their own representations of data.

In this project, we propose a novel approach that enables young people to analyze and visualize data from their online learning activities and apply the results of these analyses in structuring their learning activities. In particular, we will provide young people with opportunities to explore and experiment with data about their participation in the Scratch online community (http://scratch.mit.edu), a large and rapidly-growing website where young people, ages 8 and up, learn to program their own interactive stories, games, and animations — and share their creations with one another. We will design and develop new programming tools so that youth in the Scratch community can easily access, analyze, and represent data about the ways they program, share, and discuss Scratch projects.

With access to data about their own participation in the community, plus aggregate data from the whole community, Scratch members will be able to explore, tinker, design, and program with data in a context that is meaningful and personally relevant and that they can use to shape and direct their learning. New programming tools will enable young people to create dynamic and interactive visualizations of their participation patterns in the community. Building on the social infrastructure of Scratch, young people will be able to collaborate on these data visualizations — working together to gain new insights into the activities of individuals, interactions among friends, and trends within the community.

Although our proposed activities focus on the Scratch online community, the goals of the project are much broader. The process of creating, sharing, and analyzing Scratch visualizations will provide young people not only with a deeper understanding of their learning and participation within the Scratch community, but with a meaningful and motivating introduction to the core ideas, techniques, and challenges of data science. We will study this new pathway into data science, examining how young people learn to analyze and make sense of data patterns, where they experience difficulties (and how they overcome them), and how they engage with important issues of data science (such as privacy and anonymity). Finally, we look at how learners in Scratch apply these results and we will look at how the use of learning data by learners shapes and changes the way that young people structure their activity in the site.

Our proposal responds to the sixth challenge in the program solicitation:

> *How can the information extracted from large datasets be represented and communicated to maximize its usefulness to both teachers and students in a real-time educational setting? What delivery mechanisms are most effective for specific learning environments?*

Our project aligns with this challenge in providing learners with access to data about their own learning. But we aim to go a step further. Rather than looking for "delivery mechanisms" for communicating data representations to students, our goal is to provide learners with tools and support so that they can create (and manipulate and analyze and share) their own representations of the data.

Although the proposed project will focus specifically on the Scratch online community, its results will have broad applicability to all online STEM learning environments. Just as our Scratch programming language and online community have revolutionized how young people learn to code and how they develop as computational thinkers, we expect that our proposed approach to learner-driven learning analytics will play a similar role in revolutionizing online STEM learning. By enabling young people to analyze their own learning data, this new approach will motivate and empower young people to develop greater agency in shaping and directing their STEM learning trajectories.

2. Background

2.1 Computational Thinking and Data Science

Over the past few years, there has been a growing recognition among educators, researchers, and policy-makers that ideas from the field of computer science can be useful in understanding and solving problems in a wide range of disciplines and contexts. Jeannette Wing (2006) coined the term *computational thinking* and explained that, "it represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" (p. 33). Although there is continuing debate about the scope and definition of computational thinking, there is widespread agreement that one central component of computational thinking is the ability to analyze and visualize data (Computer Science Teachers Association, 2011; National Research Council, 2011). In the report on an NSF-initiated workshop on computational thinking (NRC, 2011), experts identified three foci for pedagogical environments for cultivating computational thinking and all three involved active engagement with data science: (a) interacting with visualizations; (b) modeling and troubleshooting of data sets; and (c) searching for patterns in large data sets.

At the same time, there has been a growing interest in applying data science to understand and improve student learning. While much of this work on learning analytics has focused on opportunities for researchers, there is also recognition of the importance and potential of providing students with opportunities to engage in analysis of their own learning data. For example, the US Department of Education Educational Data Mining Brief (Bienkowski, Feng, & Means, 2012) put forth the following challenge:

> *As colleges and schools move toward the use of fine-grained data from learning systems and student data aggregated from multiple sources, they need to help students understand where the data come from, how the data are used by learning systems, and how they can use the data to inform their own choices and actions.*

In order to address this challenge, simply displaying data to students in the form of dashboards is not enough. Learning-science research has shown that students gain a deeper understanding of concepts and practices when they actively engage in investigations and project-based learning experiences (e.g., Donovan et al., 2000; Kolodner et al., 1998, 2003). Our proposed project takes a similar approach to the introduction of data science, providing young people with opportunities to engage with data-science ideas in the context of creating personally-meaningful projects.

2.2 Scratch and Scratch Research

With strong support from the National Science Foundation, the Scratch programming language has emerged as the most popular way for young people around the world to learn to code (Resnick et al., 2009). The National Science Foundation has supported the initial development of Scratch (ITR-0325828), the study of collaboration in the Scratch online community (OCI-1027848), the design of resources to support educators working with Scratch (DRL-1019396), and the development of a cloud-based version of Scratch (IIS-1002713).

With Scratch, young people (ages 8 and up) create computer programs by snapping together graphical programming "blocks" (Figure 1). Millions of young people have used Scratch to program their own interactive stories, games, animations, and simulations — and to share their creations with one another. The Scratch website is a vibrant online community, with young people constantly experimenting with, commenting on, and remixing one another's projects. In the process, members of the Scratch community develop as computational thinkers and creators: they learn core computational and mathematical concepts, while also learning important strategies for designing, problem solving, and collaborating.

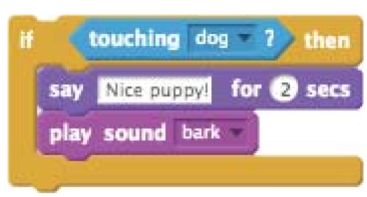

Figure 1: Sample script in Scratch programming language

Scratch is available free of charge and has been translated into more than 50 languages. It has been used extensively in schools as well as in informal settings (homes, libraries, museums, and community centers). More than 4 million projects have been shared on the Scratch website, with 6,000 new projects shared every day (roughly four new projects every minute). The Scratch online community contains more than 21 million comments and hundreds of millions of other recorded interactions (e.g., views, favorites, “loveits”), and it continues to grow rapidly (for the latest statistics, see http://scratch.mit.edu/statistics/).

The full dataset, which contains the source code and media files for every version of every project shared in Scratch, includes more than three terabytes of raw data. The Scratch team has developed a series of tools for analyzing data from the online community, making it possible to track a variety of individual and project-level outcomes in order to investigate the development of computational thinking (e.g., Hill et al., 2010; Monroy-Hernández et al., 2011; Monroy-Hernández, 2012; Hill & Monroy-Hernández, 2013a, 2013b). In addition to the analysis of online data, field studies have documented how Scratch can support the development of computational thinking (e.g., Ericson & McKlin, 2012; Kafai et al., 2011; Malan & Leitner, 2007; Maloney et al., 2008; Rizvi et al., 2012). For a more complete list of publications, see http://scratch.mit.edu/info/research/.

Researchers from a variety of fields have expressed interest in accessing the Scratch online dataset. The project team is currently working on the release of an academic dataset that includes public data from the first five years of the Scratch online community.

3. Project Goals

This project aims to:

- Create personally-meaningful pathways for young people to become engaged in data science;
- Enable young people to develop greater agency over their own learning trajectories;
- Advance our understanding of how young people can use data-science techniques to analyze and visualize data — and apply those results to shape and direct their own learning.

To accomplish these goals, we will:

- Develop a data-programming toolkit that will enable young people to access, manipulate, analyze, and visualize data from a large and personally-meaningful dataset (in particular, participation data from the Scratch online community);
- Study what young people create with new data-programming tools, how they analyze and visualize data patterns, and how their data explorations influence their learning trajectories.

4. Project Plan

This project will leverage Scratch’s enormous longitudinal dataset to give young people the ability to analyze and present their own data and to evaluate their ability to apply these analyses and representations to shape and affect their own learning processes.

The project consists of two interwoven strands. In the first strand, **Development of Data Programming Tools for Youth**, we will create a set of new Scratch programming “blocks” that will allow young people to integrate data from the Scratch online community (including their own participation data) into their Scratch projects and build their own interactive data visualizations.

In the second strand, **Empirical Analysis of Youth Use of Data Programming Tools**, we will investigate what types of projects young people create with the new data-programming blocks and how the new blocks support their data investigations and influence their learning experiences.

4.1 Development of Data Programming Tools for Youth

We will design and implement a data-programming toolkit using the Scratch programming environment as the platform, and the Scratch online community as the context and source of data. Youth are already using Scratch for a variety of data-centric projects, such as visualizations, surveys, and collaborative tools, especially since the introduction of support for persistent and shared "cloud data" in the programming language (Dasgupta, 2012). For example, youth have created surveys that tabulate the political preferences of Scratch community members, games that keep track of high scores, and collaborative stories with contributions from multiple community members. A survey project featuring characters from the Scratch media library, created by a youth in the online community is shown in Figure 2.

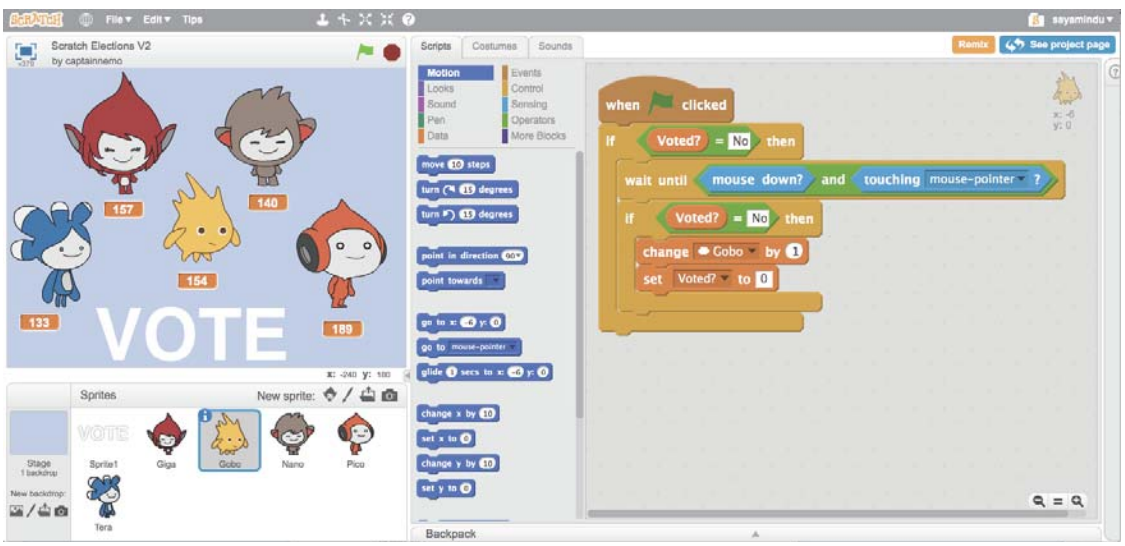

Figure 2: Example of a Scratch survey project created by a youth in the online community

In designing the new data-programming features, we will build upon our experience in designing previous versions of Scratch. Throughout the process, we will seek input and feedback from youth members of the Scratch community, through online forums and comments on projects. We will continue to iteratively refine the toolkit based on our observations of how community members use initial prototypes of the toolkit.

4.1.1 Extending Scratch

At the core of our development process is the design of a set of new programming blocks for accessing data from the Scratch online community. Each of these blocks will represent a query on the Scratch website database, such as "fetch the number of people who have remixed this project" or "fetch a list of all Scratch community members whom I follow." These blocks will enable young people in the Scratch community to build their own tools to analyze, visualize, and understand their own participation and learning in the context of the wider community.

Our goal is to create a system that is powerful and flexible (*high ceiling*) while also being understandable by young people without any traditional data-science skills or experience with statistics (*low floor*) and suitable for a wide variety of different types of projects (*wide walls*) (Resnick & Silverman, 2005).

The new data-programming blocks fall into several categories:

- Fetch **metadata about a Scratch user** (followers, projects, studios, bookmarks, etc.)
- Fetch **social metadata about a Scratch project** (number of views, number of love-its, number of comments, etc.)
- Fetch **code metadata about a Scratch project** (number of blocks used, number of sprites, number of blocks from a given category, whether the project uses a certain block or not, etc.)
- Fetch **aggregate community statistics** (total number of projects, total number of community members, total number of comments, etc.)

For example, Figure 3 shows several of the new data-programming blocks used in a program that generates a visualization of how often projects have been remixed. The size of each project thumbnail is based on the number of times the project has been remixed.

Figure 3: Example program using several of the proposed data programming blocks

In order to help youth learn how to use the new programming tools, we will develop support materials that build on the existing support infrastructure on the Scratch website. Specifically, we will develop: (a) **help screens** that introduce and provide example code for each of the new data programming blocks; **(b) idea cards** that illustrate data analysis and visualization strategies in Scratch (e.g., how to filter a list of projects to find those that contain musical blocks); (c) **starter projects** that illustrate the use of the data-programming blocks for analysis and visualization in clear ways which can be easily remixed and built upon by youth. The help screens will be available within the Scratch online project editor. The idea cards will be available in the Scratch online tips window as well as downloadable as PDFs from the Scratch support page. The starter projects will be highlighted on the Scratch starter projects page and featured on the Scratch home page. These support materials will be tested and refined as an integral part of the development and testing of the new data programming tools.

4.1.2 Sample Data-Programming Scenarios

In designing the new data-programming blocks and support materials, one of our guiding principles is to support many different types of projects, so that young people can use the blocks to create projects related to their own personal interests. As members of the Scratch community become familiar with the new data-programming blocks, we expect to see a wide variety of new Scratch projects, such as the ones described below.

> *Jose likes to keep track of the total number of projects on the Scratch website. When he notices that the number is about to reach 5 million, he decides to create a collage of images in honor of the milestone. He creates a Scratch program that fetches the thumbnail of all projects shared in the week leading up to the 5 millionth project, and generates a collage with 100 randomly-chosen thumbnails from the set.*
>
> *Emily loves Scratch projects that generate music. She logs into the Scratch website and wants to find the newest projects that generate music. To do this, she creates a Scratch program which iterates through the 100 most recent projects shared on the website, and sees if any of them use programming blocks for generating musical notes. She finds a couple of projects that match her criteria, and she adds them to the "Musical Projects" studio that she curates.*
>
> *Maria often gets new project ideas by checking what her friends are doing on the Scratch website. The homepage of the website automatically shows projects that her friends have created recently or "loved" recently. But Maria is also interested in what projects her friends have bookmarked recently (that is, projects they marked as "favorites"). She creates a Scratch program that fetches a list of her friends, and then iterates over the bookmarked projects for each friend, finally coming up with a set of bookmarked projects that is common to all her friends. She adds a title "Projects Recommended by My Friends," and features the project on her Scratch profile page.*
>
> *Arun recently got a physical activity tracker device as a holiday present, and he finds out that he can compete with his friends online through a dashboard using the device. He decides to create a similar competitive dashboard for Scratch. He creates a project that keeps track of the sprites, comments, and projects created by his friends, and does a weekly visualization showing which of his friends is leading.*
>
> *Marco creates a project with a "comment-powered" flower visualization: if the project gets a comment, the flower grows a little; on the other hand, if the project does not get comments for more than a week, the flower starts to wilt.*

We anticipate that young people will become engaged with the new data-programming features in a variety of different ways. Many will start by using data-programming visualizations created by others in the community. Some will "look inside" to see how the projects work, and then remix the project with their own customizations. Some young people will create timeline visualizations highlighting their own use of Scratch over time, others will focus on visualizations of their friends networks, while others will create projects highlighting trends in the overall community.

Whenever we have introduced new features to Scratch in the past (such as motion sensing using webcams and persistent data using cloud variables), members of the Scratch community have used the new features in ways that we never imagined. We expect the same thing to happen with the new data-programming features, as community members explore and exploit the new capabilities to create new types of projects.

4.2 Empirical Analysis of Youth Use of Data Programming Tools

In the second strand of the project, we will conduct an empirical analysis of youth programming with data in the Scratch community. This strand will document the ways that learners use data-programming blocks in Scratch projects, and how their analysis and visualization of online data impacts their learning and participation in the Scratch community. We will use a combination of qualitative and quantitative research techniques to address three main research questions:

1. How do young people use programming tools to engage in data analysis?
2. What kinds of visualizations do young people create to represent and understand their learning processes and trajectories?
3. How do young people apply these new analyses and representations to improve their learning processes and build skills associated with computational thinking?

To answer research questions (1) and (2), we will use data from the Scratch website to produce a detailed descriptive study of the way that young people use the new data-programming blocks. We will evaluate first *if*, and then *how*, young people take advantage of the new features, and we will characterize the nature of work within Scratch that takes advantage of and builds on these blocks. This study will use a mixed-method approach to describe the quantity and quality of Scratch projects that are created using the new data-programming blocks. We will compare Scratch projects using the blocks to Scratch projects without them — and, in particular, to projects without the blocks created by the same individuals.

We will describe the way that these blocks are used along a series of dimensions that take advantage of the rich longitudinal data recorded in the Scratch online community database. In particular, we will collect and present qualitative coding of projects and quantitative analysis of observational data that speaks to the following sets of questions:

- *Qualities of projects using data-programming blocks:* What new types of representation and data visualizations do Scratch community members create? What types of projects use the new blocks? How do these projects differ from other Scratch projects in complexity, size, media-richness, code-use, and other observable characteristics? Do the new blocks lead to new types or new genres of projects in the community?
- *Social reception and learning around data-programming blocks*: How are projects using the new blocks received by the Scratch community? Which projects are more popular, more viewed, and more highly rated? Which projects using the new blocks generate more comments and discussion, and what type of reactions do the projects elicit? Which projects using the new blocks inspire more remixes and derivatives?
- *Qualities of young people using data-programming blocks:* Which Scratch community members adopt the new blocks? How do patterns of adoption change over time? Do Scratch community members employ the blocks in different ways over time?

A second study will focus on investigating research question (3) and will consider how young people can apply their work with data-programming blocks to influence their learning trajectories. Specifically, we will investigate:

- *Behavior of young people who interact with projects using data-programming tools:* Do young people who view and interact with representations of community-participation data change their behavior as a result? Do they upload more or different projects? Do they shift their patterns of interaction in the Scratch community?
- *Behavior of young people who use data-programming tools:* Do community members who use the data-programming blocks produce more or different types of projects than those who do not? Do their projects differ in complexity, size, or variety over time from their earlier projects and from those who have not explored these blocks? Do they interact with more individuals, or in more ways, over time? Do they produce more highly rated, more popular, or more generative projects?
- *Learning trajectories from experience with data-programming tools*: As young people analyze data and create visualizations within Scratch, do they begin to explore new ways of programming and develop further as computational thinkers?

Like most previous empirical analyses of Scratch, these studies will describe patterns of activities and interactions within the community. That said, we will also take advantage of the fact that the data-programming blocks reflect a design *change* to the Scratch community and we will compare work in Scratch before and after the introduction of the feature. As a way of drawing causal inferences about the effects of introduction of data-programming tools, we will treat the introduction of the tools as a “natural experiment” in which individuals provided with access to tools can be imagined to have received a “treatment” and are compared to a “control” group of Scratch community members without access to the tools.

Randomized controlled field experiments in the context of social media are challenging because individuals who receive a feature can collaborate, communicate, and interact with those who do not, thus affecting one another’s behavior in the site. That said, we can employ a “regression discontinuity design” technique from econometrics to evaluate the causal effect of the introduction of a feature by comparing behavior immediately before and after its introduction while taking into account the underlying trends in variables of interest within the community (see Murnane and Willett, 2010 for a description of this technique in the context of learning and education research).

In a pilot study using Scratch data, PI Hill has used this method to evaluate the introduction of a new social feature introduced to the site designed to promote more positive attitudes towards collaboration among Scratch community members. He found evidence both of this new feature’s success in promoting collaborative work as well as of important changes in the qualities of the kinds of projects being shared in Scratch. Moreover, the technique made it possible to unpack the dynamics of these changes by estimating what portion of these changes were driven by new community members attracted to participate by the features and by changes in the behavior of existing community members.

5. Evaluation Plan

The evaluation process is an integral part of the Scratch iterative development approach, and it has been a key factor in making the online programming environment and community accessible and engaging for youth of diverse ages and backgrounds in both formal and informal settings (Maloney et al., 2010). We will extend this intensive evaluation process to ensure that the new data-programming tools, support materials, and research and dissemination activities meet the project goals. Given the focus of this project on data-programming tools, we are particularly targeting youth ages 13 to 16, who are among the most active participants in the Scratch online community and are often the first to request, experiment with, and adopt new features in Scratch.

The evaluation will consist of three phases: front-end, formative, and summative evaluation. The Scratch initiative has extensive connections with youth and educators in the online community as well in a variety of school and after-school settings, who will provide feedback and engage in testing of the new data-programming tools and support materials.

Front-end evaluation: The front-end evaluation process will have two parts: (a) assessing how youth in the target age group (13-16) are currently discussing data-related topics in the Scratch online community, and (b) gathering and analyzing feedback from youth and educators on the proposed programming tools. The front-end assessment of current use of data-related concepts will examine how youth are utilizing the current data types in Scratch (variables, cloud variables, and lists) and the types of statistics that youth are informally discussing in their current projects, profiles, and forum posts (e.g., number of remixes or “love-its”). Youth and educator feedback on the proposed blocks will be gathered by creating projects that illustrate proposed features and asking for comments, as well as by posting in online discussion forums on the Scratch website and the ScratchEd website for educators (http://scratch-ed.org). This process of youth and educator feedback helped inform the design and improvement of features added to the most recent version of Scratch, such as cloud variables and video-sensing programming blocks (launched in May 2013). The front-end evaluation will be compiled as an illustrated linked document that will inform the initial design of the data-programming tools and support materials.

Formative Evaluation: The development of the data-programming tools and support materials will involve an extensive and iterative process of formative evaluation—including pilot and beta testing. Because the Scratch programming environment is embedded in the Scratch website (as well as available as a downloadable auto-updated application), it can be continually revised and improved based on ongoing feedback and testing. (The current version of the Scratch programming editor is now on iteration version 379, with improvements and bug fixes deployed each week.)

The formative evaluation will involve a series of testing and deployment with youth both online and in face-to-face (school and after-school) settings. The first phase of formative evaluation will involve pilot testing of the initial designs with a subset of the online community by offering links to prototype versions in the Scratch online discussion forums (which are frequented by youth actively involved in the community). The prototype tools will also be piloted locally with youth in Computer Clubhouse learning centers and in other school and afterschool sites that are part of our network. The observations, feedback, and analysis of artifacts from the pilot testing will be reviewed and discussed in-depth by all members of the project team. This analysis will inform the design of the beta version of the data-programming tools and support materials. These beta versions will be available online and will be refined before officially launching the complete set of data-programming tools and support materials for broader dissemination by the end of the first year of the project. These tools and materials will then be integrated into the ongoing cycle of iterative improvements to the Scratch website. An emphasis in the formative evaluation process will be to ensure that the tools, examples, and materials provide access to data-science techniques and concepts in ways that are engaging and accessible for youth with diverse interests.

Summative Evaluation: The summative evaluation of the project will focus on the use of the data-programming tools and the related comments and discussions by youth in the Scratch online community. The evaluation will incorporate results from the empirical analysis of youth use of the new programming tools. We will examine who does (and does not) make use of the blocks, the types and range of projects created, which data-programming techniques and concepts are readily adopted (and which are more challenging or difficult to understand and use). We will also examine how the concepts, tools, activities, and materials are discussed by youth and educators in their project comments and forum posts.

At the end of the second year of the project, we release a summative report examining how well the overall project achieved its three main goals (listed in section 3, above). The summative evaluation will be based on the results of the empirical analyses, as well as other evidence of broader project impact. To understand the broader project impact, we will be examining the ways in which the activities and approaches from this project influence the broader discussion and initiatives focused on learning analytics, computational thinking, and STEM learning. We also will summarize lessons learned, describe key challenges encountered, and identify promising areas for further research.

6. Timeline

Year 1: Design and Deployment, June 2014 - May 2015
- Design and implement initial data programming tools, help screens, and example projects - Pilot test data programming tools with youth through Scratch online forums and local outreach sites - Revise and refine based on evaluation of initial use of tools by youth - Develop support materials for the refined set of programming tools - Promote and support use of new data programming tools on Scratch online community
Year 2: Learning Analysis and Dissemination, June 2015 - May 2016
- In-depth analysis of log files and projects using community data blocks - Follow-up interviews and surveys - Broader promotion of data programming tools and support materials - Presentations of findings at educational, learning, and computing conferences - Publication of findings in journals, websites, blogs, and social media - Completion and dissemination of summative evaluation report

7. Dissemination

Our research team has a strong track record for getting our ideas, activities, technologies, and strategies disseminated nationally and internationally. Millions of young people are currently using technologies based on our research (including Scratch and LEGO MindStorms robotics kits), in both formal and informal educational settings. Our team has also developed educational outreach programs to reach diverse, under-represented populations: the Computer Clubhouse network of after-school learning centers, founded by two members of our team, has more than 100 sites in 20 countries, reaching 20,000 young people in low-income communities (Rusk, Resnick, & Cooke, 2009).

We will make the new data programming tools available as an integrated part of our free Scratch authoring environment and online community. The Scratch website currently attracts more than 1.5 million unique visitors each month.

We will disseminate our research results through research papers, publications, and conference presentations. Members of our project team have a strong record of publishing in academic journals and broader-circulation publications to parents, educators, and the general public, and we regularly present at major research and educational conferences, including Digital Media and Learning, SIG-CSE, IDC, CSCL, AERA, ISTE, EdMedia, and many others. We will also highlight the data programming tools and research results at Scratch community events such as Scratch Day and the Scratch@MIT conference.

8. Expected Outcomes

8.1 Intellectual Merit

This project will advance the state of research on how to engage young people in the practice and understanding of data science. The project responds to the sixth challenge in the program solicitation: "*How can the information extracted from large datasets be represented and communicated to maximize its usefulness to both teachers and students in a real-time educational setting? What delivery mechanisms are most effective for specific learning environments?*" The project aligns with this challenge by providing young people with access to data about their own learning, but it goes a step further. Rather than looking for "delivery mechanisms" for communicating data representations to students, it provides young people with tools and support so that they can create (and manipulate and analyze and share) their own representations of the data. In this way, the project brings together two approaches from the learning sciences, integrating the quantitative, evidence-based approach of learning analytics with the constructionist tradition of young people learning through design experiences based on their own interests.

8.2 Broader Impacts

By building on top of the Scratch programming language, this project has immediate access to a large and rapidly-growing worldwide community with more than 2 million registered members and a dataset with more than 3 TB of raw data, including 4 million programming projects, 21 million comments, and detailed longitudinal data of how young people create and share their programming projects. By providing Scratch community members with programmable access to data about their own participation in the community, the project introduces data science in a context that is meaningful and personally relevant to, making it more likely for young people to make deep connections to the ideas and practices of data science. The project also enables young people to develop greater agency over their own learning trajectories, providing them with data that they can use to shape and direct their learning.

9. Project Team

Mitchel Resnick (PI), PhD, is Professor of Learning Research at the MIT Media Lab. His research focuses on the design and study of new technologies to engage children in creative learning experiences. His Lifelong Kindergarten research group developed technologies and ideas underlying the LEGO Mindstorms robotics kits and the Scratch programming language and online community. In this project, he will guide the overall direction of the project, including the design of the new data programming tools.

Benjamin Mako Hill (PI), PhD, is Acting Assistant Professor of Communication at the University of Washington. He is also a fellow at the Berkman Center for Internet and Society and an affiliate at the Institute for Quantitative Social Science — both at Harvard University. He has also been a leader, developer, and contributor to the free and open source software community for more than a decade as part of the Debian and Ubuntu projects. He is the author of several best-selling technical books, a member of the Free Software Foundation board of directors, and an advisor to the Wikimedia Foundation. Hill has a Masters degree from the MIT Media Lab and a PhD from MIT in an interdepartmental program between the Sloan School of Management and the Media Lab. He has a background in software development and data-driven statistical analyses of online communities. He has conducted a series of data-driven quantitative analyses of the Scratch online community (e.g., Hill and Monroy-Hernández 2013a, 2013b; Monroy-Hernández et al. 2011; Hill et al. 2010). In the proposed project, Hill will lead the empirical analysis of youth use of the data programming tools, in collaboration with the MIT project team.

Natalie Rusk (Co-PI), PhD, is Research Scientist at the MIT Media Lab, focusing on the research and development of technology-based programs that build on youth's interests. She is a lead designer of the Scratch programming language and leads development of educational materials for the Scratch project. She co-founded the Computer Clubhouse after-school program and led the development of the Learning Technologies Center at the Science Museum of Minnesota. She served as Network Director of the NSF-funded PIE Network, a collaboration with six museums to develop a new generation of hands-on science activities. She earned a PhD in Child Development at Tufts University and an EdM from Harvard Graduate School of Education, and completed post-doctoral mixed methods research on youth development in after-school programs at the University of Illinois at Champaign-Urbana. In the proposed project, Rusk will lead the evaluation of the project, ensuring that the design of the data programming tools and support materials address the needs and interests of a broad diversity of young people.

Sayamindu Dasgupta is a PhD student and Research Assistant at the MIT Media Lab. His research explores how children can program and interact with data, especially online-data using tools like Scratch. Dasgupta has been closely involved in the design and implementation of

Scratch 2.0, where he was in charge of designing and implementing the backend system of the online community website. As a part of his Masters thesis (Dasgupta, 2012), Dasgupta designed and implemented "Cloud data structures", a programming feature in Scratch 2.0 that enables the use of persistent and shared data in Scratch projects. Dasgupta has a Bachelor of Technology degree in Computer Science & Engineering from the West Bengal University of Technology, and a Master of Science in Media Arts and Sciences from MIT. In the proposed project, Dasgupta will lead the design and implementation of the data programming blocks in Scratch and will extend the existing Scratch data-logging infrastructure to collect information on block usage.

Additionally, one graduate student at University of Washington will contribute to the Empirical Analysis, with supervision from Dr. Hill.

10. Prior NSF Support

Resnick and Rusk at the MIT Media Lab have served as PIs (individually or collectively) on five NSF-funded projects to support development, dissemination, and study of Scratch:

Developing a Media-Rich Networked Programming Environment for Community Technology Centers in Economically Disadvantaged Communities (ITR-0325828: $1,999,435, 9/15/03 - 8/31/07)
Intellectual Merit: Advanced the understanding of the design of programming languages for youth
Broader Impacts: Created online community where youth share and collaborate on programming projects
Publications: Maloney et al. 2008; Monroy-Hernández & Resnick 2008; Resnick et al. 2009; Maloney et al. 2010

Scratch 2.0: Cultivating Creativity and Collaboration in the Cloud (IIS-1002713: $798,204, 7/01/10 - 6/30/13)
Intellectual Merit: Developed a new model for integrating a programming language into the cloud
Broader Impacts: Expanded opportunities for youth creativity and collaboration in programming activities
Publications: Dasgupta 2012; Monroy-Hernandez 2012; Resnick 2013

Preparing the Next Generation of Computational Thinkers: Transforming Learning and Education Through Cooperation in Decentralized Networks (OCI-1027848: $1,179,020, 10/01/10 - 9/30/14)
Intellectual Merit: Designed strategies for supporting cooperation in large-scale decentralized communities
Broader Impacts: Broadened participation in computing by using cooperation as a motivating context for learning
Publications: Resnick 2012; Roque 2012; Roque, Rusk, & Blanton 2013

ScratchEd: Working with Teachers to Develop Design-Based Approaches to the Cultivation of Computational Thinking (DRL-1019396: $2,158,587, 8/15/10 - 7/31/14)
Intellectual Merit: Developed strategies to support and assess design-based approaches to coding
Broader Impacts: Supported development of active community of educators helping students learn to code
Publications: Brennan et al. 2010; Brennan 2012; Brennan & Resnick 2012

ScratchJr: Computer Programming in Early Childhood Education as a Pathway to Academic Readiness and Success (DRL-1118682: $531,278, 8/01/11 - 7/31/14)
Intellectual Merit: Created new framework for educational technologies in early childhood
Broader Impacts: Developed programming language accessible to younger children, ages 5-7
Publications: Flannery et al. 2013

References Cited

Baker, R. S. J. d., D'Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, *68*(4), 223–241. doi:10.1016/j.ijhcs.2009.12.003

Bienkowski, M., Feng, M., & Means, B. (2012). *Enhancing teaching and learning through educational data mining and learning analytics: An issue brief*. Office of Educational Technology: US Department of Education.

Blikstein, P. (2011). Using learning analytics to assess students' behavior in open-ended programming tasks. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge* (pp. 110–116).

Brennan, K. (2013). *Best of both worlds□: Issues of structure and agency in computational creation, in and out of school* (Thesis). Massachusetts Institute of Technology. Retrieved from http://dspace.mit.edu/handle/1721.1/79157

Brennan, K., Monroy-Hernández, A., & Resnick, M. (2010). Making projects, making friends: Online community as catalyst for interactive media creation. *New Directions for Youth Development*, *2010*(128), 75–83. doi:10.1002/yd.377

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*.

Computer Science Teachers Association. "Computational Thinking: Teacher Resources," 2011. http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf

Dasgupta, S. (2012). *Learning with data□: A toolkit to democratize the computational exploration of data* (Thesis). Massachusetts Institute of Technology. Retrieved from http://dspace.mit.edu/handle/1721.1/78203

Donovan, M. S., Bransford, J. D., & Pellegrino, J. (2000). *How people learn: Brain, mind, experience, and school: Expanded edition*. Washington, DC: National Academy Press.

Duval, E. (2011). Attention Please!: Learning Analytics for Visualization and Recommendation. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge* (pp. 9–17). New York, NY, USA: ACM. doi:10.1145/2090116.2090118

Ericson, B., & McKlin, T. (2012). Effective and Sustainable Computing Summer Camps. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (pp. 289–294). New York, NY, USA: ACM. doi:10.1145/2157136.2157223

Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M. (2013). Designing ScratchJr: Support for Early Childhood Learning Through Computer Programming. In *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 1–10). New York, NY, USA: ACM. doi:10.1145/2485760.2485785

Guzdial, M. (2008). Education: Paving the Way for Computational Thinking. *Commun. ACM*, *51*(8), 25–27. doi:10.1145/1378704.1378713

Hill, B. M. (2013). *Causal effects of a reputation-based incentive in an peer production community*. Working Paper.

Hill, B. M., & Monroy-Hernández, A. (2013a). The cost of collaboration for code and art: Evidence from a remixing community. In *Proceedings of the 2013 conference on Computer supported cooperative work* (pp. 1035–1046). New York, NY, USA: ACM. doi:10.1145/2441776.2441893

Hill, B. M., & Monroy-Hernández, A. (2013b). The Remixing Dilemma The Trade-Off Between Generativity and Originality. *American Behavioral Scientist*, *57*(5), 643–663. doi:10.1177/0002764212469359

Hill, B. M., Monroy-Hernández, A., & Olson, K. (2010). Responses to remixing on a social media sharing website. In *Proceedings of the 4th AAAI Conference on Webslogs and Social Media* (pp. 74–81). Washington, D.C.: AAAI.

Kafai, Y., Roque, R., Fields, D., & Monroy-Hernández, A. (2011). Collaboration by choice: Youth online creative collabs in Scratch. *Proceedings of the International Conference on Computers in Education*.

Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., … Ryan, M. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting Learning by Design (tm) into practice. *The Journal of the Learning Sciences*, *12*(4), 495–547.

Kolodner, J. L., Crismond, D., Gray, J., Holbrook, J., & Puntambekar, S. (1998). Learning by design from theory to practice. In *Proceedings of the International Conference of the Learning Sciences* (pp. 16–22).

Malan, D. J., & Leitner, H. H. (2007). Scratch for Budding Computer Scientists. *SIGCSE Bull.*, *39*(1), 223–227. doi:10.1145/1227504.1227388

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by Choice: Urban Youth Learning Programming with Scratch. *SIGCSE Bull.*, *40*(1), 367–371. doi:10.1145/1352322.1352260

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *Trans. Comput. Educ.*, *10*(4), 16:1–16:15. doi:10.1145/1868358.1868363

Monroy-Hernández, A. (2012). *Designing for remixing : Supporting an Online community of amateur creators* (Thesis). Massachusetts Institute of Technology. Retrieved from http://dspace.mit.edu/handle/1721.1/78202

Monroy-Hernández, A., Hill, B. M., Gonzalez-Rivero, J., & boyd, danah. (2011). Computers can't give credit: How automatic attribution falls short in an online remixing community. In *Proceedings of the 2011 annual conference on Human factors in computing systems* (pp. 3421–3430). Vancouver, BC, Canada: ACM.

Monroy-Hernández, A., & Resnick, M. (2008). Empowering Kids to Create and Share Programmable Media. *interactions*, *15*(2), 50–53. doi:10.1145/1340961.1340974

Montalvo, O., de Baker, R. S. J., Michael, A., Pedro, S., Nakama, A., & Gobert, J. D. (2010). Identifying Students' Inquiry Planning Using Machine Learning. In *EDM* (pp. 141–150).

Murnane, R. J., & Willett, J. B. (2010). *Methods matter: Improving causal inference in educational and social science research*. Oxford University Press, USA.

National Research Council. (2011). *Report of a workshop on the pedagogical aspects of computational thinking*. Washington, DC: The National Academies Press. Retrieved from http://www.nap.edu/catalog.php?record_id=13170

Resnick, M. (2006). Computer as paint brush: Technology, play, and the creative society. *Play= Learning: How play motivates and enhances children's cognitive and social-emotional growth*, 192–206.

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

Resnick, M. (n.d.). Learn To Code, Code To Learn (EdSurge News). *EdSurge*. Retrieved December 1, 2013, from https://www.edsurge.com/n/2013-05-08-learn-to-code-code-to-learn

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. In *Proceedings of the 2005 conference on Interaction design and children* (pp. 117–122). New York, NY, USA: ACM. doi:10.1145/1109540.1109556

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., … Silver, J. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60. doi:10.1145/1592761.1592779

Rizvi, M., & Humphries, T. (2012). A Scratch-based CS0 course for at-risk computer science majors. In *2012 Frontiers in Education Conference Proceedings* (Vol. 0, pp. 1–5). Los Alamitos, CA, USA: IEEE Computer Society. doi:10.1109/FIE.2012.6462491

Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, *33*(1), 135–146.

Roque, R. V. (2012). *Making together□: Creative collaboration for everyone* (Thesis). Massachusetts Institute of Technology. Retrieved from http://dspace.mit.edu/handle/1721.1/77814

Rusk, N., Resnick, M., & Cooke, S. (2009). Origins and guiding principles of the computer clubhouse. *The computer clubhouse: Constructionism and creativity in youth communities*, 17–25.

Santos, J. L., Govaerts, S., Verbert, K., & Duval, E. (2012). Goal-oriented Visualizations of Activity Tracking: A Case Study with Engineering Students. In *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge* (pp. 143–152). New York, NY, USA: ACM. doi:10.1145/2330601.2330639

Santos, J. L., Verbert, K., Govaerts, S., & Duval, E. (2013). Addressing Learner Issues with StepUp!: An Evaluation. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge* (pp. 14–22). New York, NY, USA: ACM. doi:10.1145/2460296.2460301

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215