ScratchJr: Computer programming in early childhood education as a pathway to academic readiness and success

In response to challenge 5 that invites the development and testing of innovative technologies to transform STEM learning experiences (Full research and development projects) we propose to design, develop, implement, and evaluate a new version of the Scratch programming language, to be called ScratchJr, designed specifically for early childhood education (K-2). This is an educational segment where there are few powerful educational technologies that specifically take into consideration what is developmentally appropriate in terms of interface design as well as what is needed to improve educational outcomes. The PI, Marina Bers, with funding from NSF (DRL-0735657) has experience developing programming languages for young children, kindergarten curriculum to teach powerful ideas from computer science, and evaluating learning outcomes (Bers, in press; Bers & Horn, 2010). Results of this research show that, when given a developmentally appropriate interface and curriculum, young children are able to acquire programming skills and improve in other cognitive tasks such as sequencing (Kazakoff & Bers, 2011). Also with NSF funding, Bers has worked on a framework for designing curricular experiences and technologies to support children's positive development (Career IIS-0447166) (Bers, 2006). These guidelines will inform the design of ScratchJr.

Our proposed project will build on the PI's research on technologies in early childhood education, and on the work of the co-PI, Mitchel Resnick at the MIT Media Lab, developing Scratch,, a free graphical-programming language developed with funding from the National Science Foundation (NSF grant ITR-0325828). With Scratch, students can program their own interactive stories, games, animations, and simulations. By snapping together graphical programming blocks in Scratch, children can create a story with characters that dance, sing, and interact with one another, or they can create an interactive birthday card for a friend. In the process, children learn important literacy and mathematical concepts, and they develop valuable problem-solving skills (Resnick, 2007b). Since the launch of Scratch in 2007, more than 2 million people have downloaded the software. There is a vibrant online community (http://scratch.mit.edu), where students around the world share more than 1500 new Scratch projects every day.

The current version of Scratch is intended for students ages 8-16. For the proposed project, we will create a modified version of Scratch to meet the needs of early childhood education (K-2) and to target specific learning outcomes. These learning outcomes are grouped in three areas:

1. **Foundational knowledge structures**: ScratchJr will engage young children in building foundational knowledge structures that apply to most academic domains, and are particularly important for developing both early literacy and early mathematics. Most specifically, we will focus on ***sequencing, causality, classification, composition, symbols, patterns, estimation and prediction***.
2. **Discipline specific knowledge**: Within its open-ended project-based platform, ScratchJr will integrate curricular modules designed to target the knowledge and skills identified by national and state frameworks (National Council of Teachers of English, 1996; National Council of Teachers of Mathematics, 2000) as well as

by the National Association for the Education of Young Children (NAEYC), as learning outcomes in literacy and mathematics that are expected to be achieved by the end of second grade. Other modules will take advantage of ScratchJr's computational platform to introduce more advanced concepts such as variables and randomness in mathematics, and the emphasis on narrative structure in literacy.

3. **Problem-solving skills**: In the process of designing and programming interactive multimedia projects with ScratchJr, children engage in a set of problem-solving skills: ***1) identification of a goal; 2) formulation of a plan; 3) development of an initial attempt at meeting the goal; 4) testing, evaluating, and sharing outcomes; 5) debugging, editing, and revising the initial attempt based on feedback.*** These skills involve the development of capacities that apply to most academic domains and are facilitated by the computational nature and designed-based approach supported by ScratchJr. These study skills also correspond to executive functions, which are strong predictors of a child's future language skills and social-emotional behavior (Bierman et al., 2008; Friedman et al., 2006). Furthermore, the executive functions of planning, inhibition, and updating are found to be strong factors in the development of numeracy, counting and academic success (Blair, 2002; Kroesbergen et al., 2009).

Our proposed ScratchJr project will have three components: 1) a developmentally appropriate interface, with both touch screen and keyboard/mouse options; 2) an embedded library of curricular modules with math and literacy content to meet federal and state mandates in early childhood education; and 3) an on-line resource and community for early childhood educators and parents.

Project Rationale and Theoretical Background

Our proposal is based on the need for developing educational technologies specifically tailored for early childhood education that are rooted in developmentally appropriate practice (DAP). This is a perspective within early childhood education concerned with creating learning environments sensitive to children's social, emotional, physical, and cognitive development (Copple & Bredekamp, 2009).

In the early days of personal computing, there was lively debate over the developmental appropriateness of technology use in early elementary classrooms (Clements & Sarama, 2003). Today, however, the pressing question is no longer whether but how we should introduce computer technology in early elementary school (Clements & Sarama, 2002). "What software is being used? And how is it integrated with and supported by the broader classroom curriculum?" are the essential questions that we need to address. Given the increasing mandate to make early childhood programs more academically challenging, while honoring the importance of play in the developmental trajectory, technology can provide a playful bridge to integrate academic content with meaningful projects (Bers, 2008a).

Our proposal to develop ScratchJr is intended to promote learning outcomes in well-established academic domains in early childhood, such as literacy and mathematics, but also advances the notion that young children can benefit from learning how to program at an early age. Previous research has shown that children as young as four years old can understand the basic concepts of computer programming and can build and program simple robotics projects (Bers, 2008b; Bers, 2007, Cejka, Rogers, & Portsmore, 2006; Rogers & Portsmore, 2004). Furthermore, early studies with the text-based language Logo have shown that computer programming, when introduced in a structured way, can help young children with a variety of cognitive skills, including basic number sense, language skills, and visual memory (Clements, 1999).

A large body of interdisciplinary research on technology and early childhood education has been conducted in the past three decades. Efforts focused on the impact of technology on children's cognitive and academic development (Fletcher-Flinn and Gravatt, 1995; Clements, 2002) and socio-emotional development (Crook, 1998; Medvin, Reed, Behr, and Spargo, 2003; Muller and Perlmutter, 1985; Shade, Nida, Lipinski, and Watson, 1986; Wang & Ching, 2003).

Nonetheless, computer programming is difficult for novices of any age (Kelleher and Pausch, 2005). There are many conceptual hurdles faced by novice programmers such as fundamental misconceptions about the nature of computers (Norman, 1986; Ben-Ari, 1998; McKeithen, Reitman, Rueter, and Hirtle; 1981). In addition to the challenges faced by all novice programmers, we must also consider the developmental needs and capabilities of young children (Rader, Brand, and Lewis; 1997). One problem is that the syntax of text-based computer languages, such as Logo, can be unintuitive and frustrating for novice programmers. This is exacerbated for young children who are still learning how to read. Modern visual programming languages such as ROBOLAB allow children to program by dragging and connecting icons on the computer screen (Rogers and Portsmore, 2004). While this approach simplifies language syntax, the interface requires young children to use a mouse to navigate hierarchical menus, click on icons, and drag lines to small target areas on a computer screen. All of this requires fine motor skills that make it difficult for young children (Hourcade, Bederson, Druin, & Guimbretière, 2004). As a result, adults often sit with young children and give *click-by-click* instructions to make programming possible, which poses counterintuitive challenges for children's learning (Beals & Bers, 2006). Attempts have been made to create simpler versions of these languages, such as the PILOT level in ROBOLAB. However, the resulting interface often obscures some of the most important aspects of programming, such as the notion of creating a sequence of commands to form a program's flow-of-control.

Research conducted by Bers, the PI, shows that young children, when presented with the appropriate context and curriculum, can use both screen-based and tangible programming environments to control the behaviors of LEGO-based robots that they themselves constructed (Bers et al, 2002; Bers, 2007; Bers, 2008a; Bers & Horn, 2010 Horn, Bers, & Jacob, 2009). Building on this work, our proposed educational software, ScratchJr, integrates ideas of graphical computer programming, with an option of a touch-screen interface, in the design of a developmentally appropriate technology. We also extend the user experience to specifically address concepts and skills in our three outcome areas:

foundational knowledge structures, domain-specific knowledge in early literacy and early mathematics, and problem solving skills.
Our work is inspired by the tradition of constructionist programming environments for experimenting with disciplinary powerful ideas (Papert, 1980). The open-ended nature of these environments, combined with the inherent structure of computer programming, has been shown to effectively foster imagination and creativity (Resnick, 2007).

Project Hypotheses
ScratchJr will provide children with new learning opportunities through designing and programming personally meaningful projects and sharing them with a community of peers and adults who can provide feedback. Our hypotheses are:

1. In terms of **foundational knowledge structures**, ScratchJr will provide opportunities for children to make sequences, understand causality, classify letters or numbers as well as media types, understand and construct symbols and patterns, create compositions, predict outcomes and estimate results.

These foundational knowledge structures will be encountered every time a child programs the behaviors of story characters (called "sprites" in ScratchJr) to make a coherent multimedia story and to add interactivity to it. The development of these foundational knowledge structures will be supported and promoted by the nature of ScratchJr as a programming environment.

2. In terms of **discipline specific knowledge**, ScratchJr will provide specifically designed scaffolds, through curricular modules, for children to gain content knowledge in the areas of early literacy and early math, as mandated by state and federal frameworks.

Most specifically, the curricular modules in early literacy will address: Language development and vocabulary (word families, high frequency words, syntactic, end marks capitalization); Decoding (sound recognition, correspondence); Text understanding (connections, genre, etc.); and Composition (writing, conventions, research, media production). In early math, the curricular modules will address number sense and operations, patterns, relations and algebra, geometry, measurement, data analysis, statistics and probability. ScratchJr will be unique in integrating an open-ended project-based approach with curricular modules to meet state standards. The modules will be organized in libraries searchable through key early math and early literacy concepts. Through its virtual community component, ScratchJr will invite early childhood teachers and parents to expand the library by adding new modules.

3. In terms of **problem-solving skills**, students using ScratchJr will be involved in all aspects of developing skills that will help them accomplish a task. When students create their own projects on ScratchJr, they begin by identifying goals for their actions, formulating an action plan through the use of an interactive design journal that prompts them with questions about their ideas and the needed steps, developing an initial attempt at meeting their goals, testing and evaluating their

projects while sharing them with others, revising their ideas by assessing what went wrong and what they could do better, and formulating a second attempt.

This iterative process of problem solving shares many aspects with the software development and the engineering design process. However, the advantage of our approach is that it conceives these steps as capacities that can be built in a child. These can be applied to all domains of academic life. For example, the same iterative process can prove useful when writing a poem, solving a mathematical equation or developing a science experiment.

The Tool: ScratchJr

In the same spirit as its parent tool, ScratchJr is a graphical programming language that lets children control the actions and interactions among different media. However, ScratchJr will have two major extensions and modifications that make it a unique tool for early childhood education: a developmentally appropriate interface that includes touch screen capabilities in a hand held tablet, and a library of curricular modules to meet federally and state mandated curricular content knowledge.

Creating computer programs in the current version of Scratch is much easier than in traditional programming languages: children can simply snap together graphical blocks, much like LEGO bricks or puzzle pieces. But there are still significant challenges. Programming in Scratch still requires reading and writing, and the number and complexity of the programming blocks can be overwhelming for young children. Figure 1 shows a screenshot of the way in which current programs are created in Scratch. Figure 2 shows the Scratch user interface, which includes a large number of features and icons.

Figure 1: Programming in Scratch

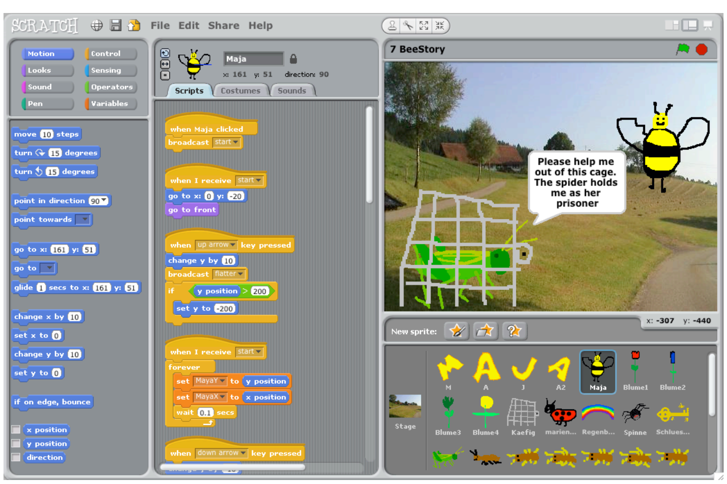


Figure 2: Scratch user interface

While respecting the main tenet of providing opportunities for children to learn by creating their own meaningful projects, ScratchJr will adapt and extend the current Scratch system in the following ways:

1. A developmentally appropriate interface that combines the use of icons and words with the option of using a touch-screen interface for handheld tablets (such as the iPad). The interface will increasingly introduce elements of literacy and numeracy.
2. Content domain modules that engage learners on projects aimed at promoting each of the three outcome areas: Foundational knowledge structures, Discipline specific knowledge, and Problem solving skills
3. A ScratchJr virtual community for adults. This community will include a section for teachers to share newly developed modules focused on specific themes to address the core curricular areas in early math and literacy. The online community will also include guidelines for parents to support their children's learning with ScratchJr, and ideas for home projects that reinforce the skills learned at school. Research shows that parents and extended families play a crucial role in their children's development and academic success (Senechal &LeFevre, 2002; Teale, 1984; Fan & Chen, 2001).

In the following paragraphs we expand on each of the three extensions for ScratchJr.

1. A developmentally appropriate interface

The developmentally appropriate interface we will design for ScratchJr will mix icons and words and will introduce elements of literacy. Below is an example of what programming in ScratchJr might look like. The first script tells a character to walk 10 steps, say "Hi", wait six seconds, then play a song. The second script tells the character to walk a little bit, then wait a little bit, then walk a little bit, then wait a little bit (repeated 5 times).

There are several aspects of Scratch that need to be rethought and revised in order to make it more appropriate for early childhood:

- Reduced set of features. The current version of Scratch has more than 100 programming blocks. ScratchJr will have a smaller collection of blocks.
- Higher-level programming blocks. Some of the programming blocks in ScratchJr will control aggregate movements, such as "dance" or "jump" (rather than simply "move"), making it easier for young children to create meaningful narratives with a small number of blocks.

- Mixture of words and icons. By using both icons and words in the menus and programming blocks, ScratchJr will serve as a stepping stone for early literacy.
- Developmentally-appropriate user interface. ScratchJr will use larger programming blocks and simpler means for selecting inputs to blocks, to fit with the motor skills of younger children. We will develop the interface taking into consideration the particular needs of touch-screen interfaces, so ScratchJr can be used on handheld tablets, which we expect will become more popular and less expensive in the future.
- Story starters. The current version of Scratch starts with a "blank screen". ScratchJr will provide "story starters" to help young students get started on their projects.

2. STEM content domain modules

Research has shown that mere exposure to computer programming in an unstructured way has little demonstrable effect on student learning. Therefore, an important aspect of our work is to develop curriculum that utilizes programming activities as a vehicle to introduce a series of powerful ideas from the early math and early literacy content domains in a structured, age-appropriate way. Next are examples of ScratchJr curricular modules to address learning standards mandated by the state of MA in the K-2 range in both literacy and mathematics.

<u>Literacy</u>

Standards on Reading and Literature

(PreK - K): Retell a main event from a story heard or read.

(Gr 1 - 2): Retell a story's beginning, middle, and end.

Standards on Composition

(PreK - K): Draw pictures and/or use letters or phonetically spelled words to tell a story.

(Gr 1 - 2): Write or dictate stories that have a beginning, middle, and end.

Connection to other Learning Standards

(PreK - 2): Identify different interpretations of plot, setting, and character in the same work by different illustrators.

ScratchJr Activity

As a response to stories that children have read or heard, students will use ScratchJr to create three panels representing beginning, middle, and end of the main idea of a story. ScratchJr will provide them with a space to create images and sounds representing important parts of the story. Then, they will elaborate on the first panel and code its animation to connect it to the next panel and so on. Children will be presented with the challenge of thinking about sequence of events and how the storyline gets from one point to another, as well as the sequence of actions to get elements of the animation from one stage to another. Children will play their animation to verify that their story is properly sequenced and complete. They may add audio narration and/or text to go along with specific panels.

Math

Learning Standards

(PreK - K): Use objects and drawings to model and solve addition and subtraction problems (to ten).

(Gr 1-2): Identify different patterns as parts of the hundred's chart.

Connection to other Learning Standards

(PreK - Gr 2): Patterns, Relations, and Algebra

Identify, reproduce, describe, extend, and create color, rhythmic, shape, number, and letter repeating patterns (PreK-K: with simple attributes, e.g., ABABAB....)

(PreK - K): Number Sense and Operations (Count by ones to at least 20, Identify positions of objects in sequences up to fifth); Patterns, Relations, and Algebra (Count by fives and tens at least up to 50); Data Analysis, Statistics, and Probability (Collect, sort, organize, and draw conclusions about data using concrete objects, pictures, numbers, and graphs).

(Gr 1-2): Number Sense and Operations (Estimate, calculate, and solve problems involving addition and subtraction of two-digit numbers); Patterns, Relations, and Algebra (Identify, reproduce, describe, extend, and create simple rhythmic, shape, size, number, color, and letter repeating patterns, Create and extend addition and subtraction number patterns, Skip count by twos, fives, and tens up to at least 50, starting on any number); Data Analysis, Statistics, and Probability (Organize, classify, represent, and interpret data using tallies, charts, tables, bar graphs, pictographs, and Venn diagrams; interpret the representations)

ScratchJr Activity:
The ScratchJr animation panel will be filled in with a 10-by-10 grid, numbered from 1 to 100. This is known as the "hundred's chart". Children will explore patterns in the number system, patterns of addition and subtraction, and patterns within the hundred's chart as they program an interactive character to move systematically through the chart and perform a dance when it encounters numbers based on a chosen characteristic (odd or even, less than 10, more than 40, etc.) For instance, children may program the interactive character to step through the hundred's chart one block or number at a time; or they may program the blocks to change colors according to a chosen criteria, such as having a 2 in the one's place. As another example, children could program the interactive character to move in a regular pattern, plus 10, minus 1, over and over, highlighting the block the character lands on. Thus, through programming the characters' behaviors with mathematical expressions, children will visualize the rich patterns that are inherent in the hundred's chart.

3. A ScratchJr virtual community space for adults

ScratchEd (http://scratched.media.mit.edu) was launched in July 2009 as a new online community where Scratch educators share stories, exchange resources, ask questions, and find people with similar interests. We will develop a special subsection of ScratchEd specifically to connect early childhood educators utilizing ScratchJr. They will be able to share notes and stories about both the *process* of using the technology in the classroom and the *products*, such as newly developed curricular modules or adapted versions of already existing ones. They will be able to give feedback to each other and rank projects according to curricular relevance in terms of national frameworks, originality, difficulty of implementation, etc. We will also include "family modules" of ScratchJr activities for families to do at home to supplement the work happening at school. Research shows that one of the most important predictors for academic success is parental involvement. Thus, we will develop resources specifically directed to parents of young children.

Research Design

In order to develop and pilot test ScratchJr, we will follow the *design experiments* methodological framework (Bell, 2004; Barab and Squire, 2004). Due to their interventionist method, design experiments are particularly helpful when exploring how a technological innovation affects student learning and educational practice (diSessa & Cobb, 2004). Following the design experiments methodology, we will engage in an iterative development, evaluation, and redevelopment process that will consist of four phases spanning three years. Studies involving students or teachers will strictly adhere to policies and procedures for human subject research and will protect their anonymity.

During **phase 1**, we will conduct studies in kindergarten, first and second grade. The children will use the current version of Scratch for a total of twenty hours and we will collect data regarding 1) difficulties encountered with the interface and conceptual programming difficulties, and 2) data regarding advances in what we have identified as our three major outcomes (foundational knowledge, discipline-specific knowledge, and problem solving skills). In parallel, we will conduct focus groups with 15 early childhood educators. Based on analysis of children and teacher's data, we will develop the first ScratchJr prototype by focusing on goal 1 (a developmentally appropriate interface) and goal 2 (content-domain module). We will present it to our advisory board and incorporate their feedback into our second phase.

During **phase 2**, we will conduct two studies with the K-2 groups using the first prototype of ScratchJr. The first study, done at the DevTech lab at Tufts University, will be aimed at debugging and improving the interface design. Based on feedback from this study, we will develop ScratchJr version 2. This second prototype will be used at Eliot Pearson Children's School (EPCS), a lab school associated with Tufts' Department of Child Development, to support existing teaching practice. Pilot evaluation will focus on: feasibility of incorporating ScratchJr into the classroom routines and effectiveness of the curricular modules in improving outcomes in our three core areas. Analysis of these data will inform the development of the third ScratchJr prototype. In parallel, our development team will start to implement goal 3 (a ScratchJr virtual community space for adults), as well as create new curricular modules in literacy and math. The first prototype of the virtual community will be tested with early childhood teachers and parents through focus groups in the DevTech lab at Tufts University.

During **phase 3**, the third ScratchJr prototype will be ready to use in the classroom. A pilot study will be conducted with K-2 students at EPCS to assess the impact of using ScratchJr for improving our three outcomes. During this study the use of the virtual community aspects will be incorporated for use by teachers and parents.

During **phase 4**, the final version of ScratchJr will be introduced to a larger community of early childhood educators through connections with early childhood settings that the PI already has in place. For example, we will invite K-2 classes at the JCDS school in Watertown, MA, and the Healey school in Somerville, MA, to work with us.

Research Sample

During Phase 1, we will conduct three pilot studies using the current Scratch software with a total of 40 children in K-2 at EPCS and focus groups with 15 early childhood educators. During Phase 2, we will conduct a study using ScratchJr version 1 in the DevTech lab with a total of 30 children in K-2. Later, we will conduct a study using ScratchJr version 2 at EPCS with a total of 40 children in K-2. We will run focus groups with 9 teachers from EPCS to evaluate the first implementation of the virtual community aspects of ScratchJr. During Phase 3, we will conduct a pilot study using ScratchJr version 3 at EPCS with a total of 40 children in K-2. We will also conduct a pilot study of the virtual community aspects with the teachers and parents of the 40 students. During Phase 4, we will invite other settings to download ScratchJr and work with it. We will follow all of them up for a period of six months, but will work closely with 10 teachers and their classrooms of approximately 20 students. Thus, in terms of total sample, during the three-year project we would have conducted a pilot evaluation of ScratchJr with approximately 350 students in K-2, 40 parents, and 58 early childhood educators.

Timeline

The following table summarizes each of the four phases, describing the tasks to be accomplished, the data collection methods, the measures and ways of analysis, and the criteria for assessment.

	Task Summary	Data Collection	Measures & Analysis	Criteria for Assessment
Phase 1 (mos. 1-12)	Focus groups and Scratch workshops yield data for the development of prototype of ScratchJr v1. Feedback from advisors. ScratchJr v1 tested with children at DevTech lab	1. Focus group with teachers 2. Computer logs 3. Classroom observation 4. Advisory group	1. Qualitative coding of transcripts. 2. Quantitative coding of computer logs	1. Developmental appropriatenss of prototype interface. 2. Content modules that integrate 3 outcome areas.
Phase 2 (mos. 12-18)	Develop ScratchJr v2, which is tested in a Eliot-Pearson classroom, yielding feedback for prototype 3. Develop virtual community aspects.	1. Classroom observations 2. Interviews with teachers & students 3. Computer log 4. Pre & post task assessments on three outcomes	1. Qualitative coding of transcripts 2. Event-history analysis of logs 3. ELSA & TEMA-3 pre- & post assessments	1. Feasibility in various classroom contexts. 2. Initial data on gains in content from pre to post 3. Robustness and fidelity to designs in prototype

Phase 3 (mos. 18-28)	Prepare ScratchJr v3 prototype and virtual community for pilot test. Results used to refine virtual community and develop ScratchJr v4, release candidate.	1. Classroom observations 2. Computer log 3. Virtual community log. 4. Feedback from teachers/parents	1. Content analysis of logs. 2. Aggregate questionnaire data.	1. Value found in use of virtual community 2. Data on gains in content (pre/post) 3.
Phase 4 (mos. 29-36)	10 educators selected for in-depth follow up. Continued technical support and virtual community moderation.	1. Questionnaire 2. Interviews 3. Classroom observations. 4. Use log from prototype. 5. Community log.	1. Qualitative coding of transcripts. 2. Aggregate questionnaire data. 3. Content analysis of logs.	1. Robustness of technology 2. Ease of use and adaptation. 3. Impact on the three outcomes 4. Sustainability and scalability of virtual community. 5. Feasibility of implementation in ECE settings

Assessment plan
The measures and assessments that will be used in this iterative pilot study are organized into three categories according to our three outcomes.

Outcome 1: Foundational knowledge structures
Assessment of foundational knowledge in the context of ScratchJr will draw on classic Piagetian tasks to tap into children's understanding of concrete operations such as sequence, causality, classification, composition, symbols, patterns, and prediction. Piagetian tasks are used as readiness assessment in early education, especially in relation to early mathematics and reasoning (Hiebert and Carpenter, 1992), and as indicators of a child's ability to perform and think about fundamental concepts that are needed for all academic subjects. These tasks will be adapted and integrated within the ScratchJr environment and will be presented as tutorials to teach children the different components of ScratchJr. They will be completed before children create their own projects and prior to the use of the content-domain curricular modules. Similar tasks will be given again at the end of the study and will be presented as challenges to solve within ScratchJr.

Outcome 2: Discipline specific knowledge
Disciplinary knowledge will be assessed in four manners during the experience at the EPCS lab school:

1) Teachers' assessments: Pre- and post assessment using a standard instrument before and after using ScratchJr. to evaluate general readiness skills, early literacy, and early mathematics (Brassard and Boehm, 2007)

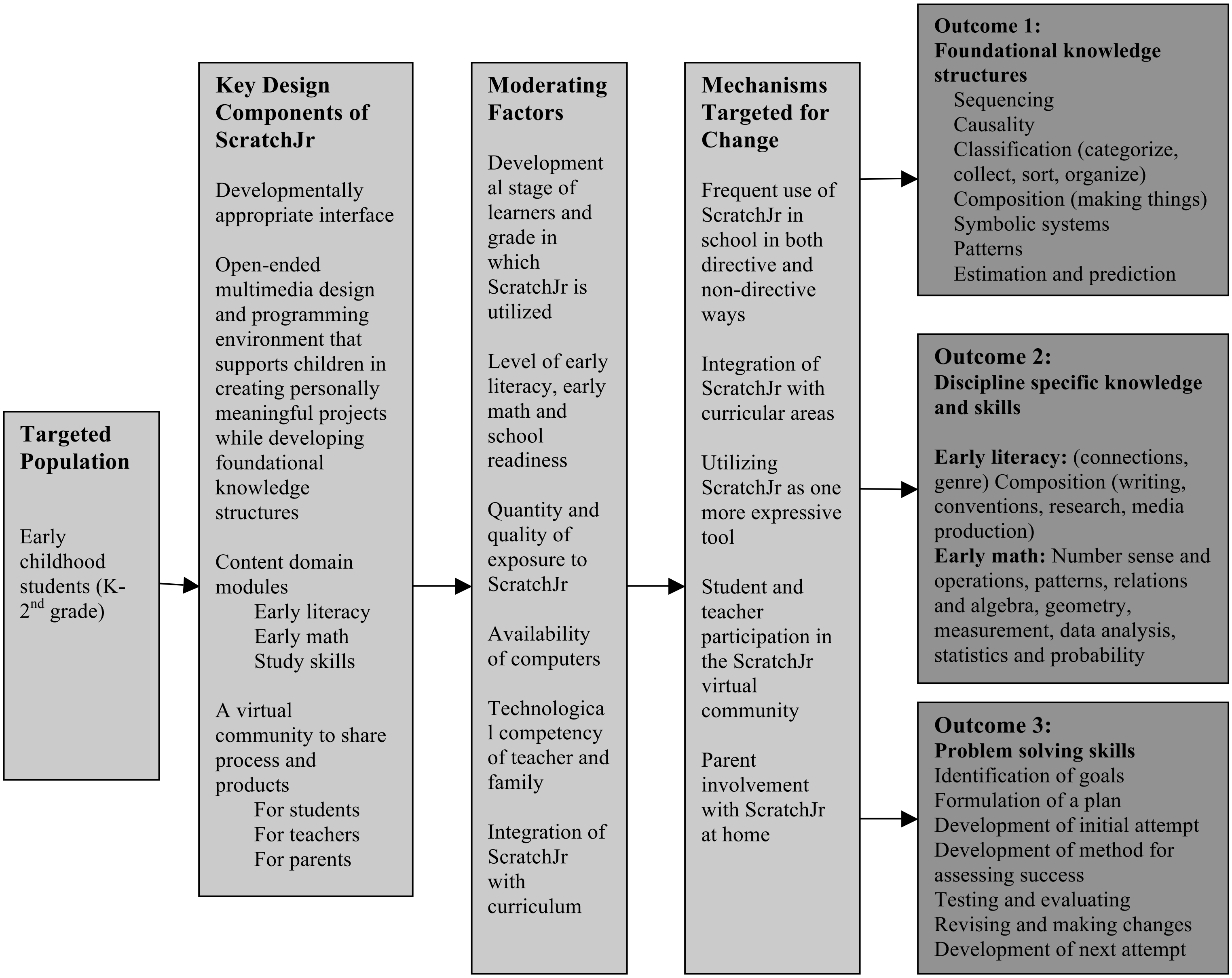
2) School portfolios: The school compiles portfolios of student work as evidence of children's learning. The projects created with ScratchJr will be integrated into this documentation and will be analyzed to see change over time.
3) Researcher's evaluations: Selected questions from *The Early Literacy Skills Assessment* (ELSA) from HighScope (DeBruin-Parecki, 2004), and the Test of Early Mathematics Ability-Third Edition (TEMA-3) by Herbert Ginsburg and Arthur Baroody (2003), will be used before and after the use of ScracthJr.
4) ScratchJr. computer logs will be analyzed to see instances of math and literacy skills.

Outcome 3: Problem solving skills

We will focus on five problem solving skills: 1) identification of a goal; 2) formulation of a plan; 3) development of an initial attempt at meeting the goal; 4) testing, evaluating and sharing outcomes; 5) debugging, editing, and revising the initial attempt based on feedback. Throughout the process of creating their own interactive projects, students will develop these skills by following the design process commonly used in software development and engineering. We will build into ScratchJr an interactive icon-based design journal that will remind students to engage in all five steps. Data of how often they use the journal and how often they revise their projects will be automatically collected by the logs. Longitudinal data will be analyzed to see change over time in student's design and implementation of projects. During Phase 3, questionnaires will be given to parents to assess parental participation as a mediating factor. Computer logs will be analyzed to detect projects created by families at home.

Theory of change

The following diagram shows our proposed theory of change. It is important to note that the success of the use of ScratchJr will depend not only on the developmentally appropriate interface design and the inclusion of modules to specifically target our three outcomes, but also on the way ScratchJr is integrated into the curriculum during school time. Mediating factors for successful integration will be: 1) balance between teacher-planned activities to meet mandated frameworks and activities that emerge from the children's interests, 2) educational settings that invite and promote inquiry, problem solving, discovery and application of key issues and concepts, 3) learning environments in which connections within and across disciplines are made through project-based learning.



External project evaluation

An external project evaluation will be conducted by Ponte & Chau, a consulting firm with experience evaluating educational programs for early childhood. The evaluation will be framed by Jacob's (1988) multi-tiered approach. Formative evaluation includes tiers 1-3 and summative includes tier 4-5. Tier 1 will focus on defining project goals and collecting data to establish a baseline. Tier 2 will focus on ensuring accountability toward research goals. The evaluation team will assess the reliability of any coding schema. Interrater reliability will be assessed against the evaluation team. Face validity of content modules will be assessed by a panel of teachers and random visits to classrooms will be made to assess fidelity to research procedures. Tier 3 will focus on reviewing progress and recommending adjustments. At each review, progress toward milestones will be assessed and recommendations will be made to improve fidelity of classroom implementation, school collaboration, and the content modules. Tier 4 evaluation will focus on determining the values of research outcomes in three ways: 1) outcomes will be mapped against the theory of change to monitor the validity of the model; 2) outcomes will be mapped against each prototype to assess the value of each modification; and 3) outcomes will be mapped against each hypothesis to assess progress and feasibility. Tier 5 will focus on attesting to the integrity of the final ScratchJr software, curriculum modules, and theory. The two teams will work to distill the theory of change based on outcomes and to ensure reliability and validity of any analysis. A final report of summative evaluation will be shared with the research team.

Dissemination

Our interdisciplinary work contributes to the fields of child development, early childhood education, human-computer interaction, and programming languages for children. Most specifically, the proposed project will make available the ScratchJr software for free download. The ScratchJr virtual community for adults will also be of free access and will provide resources for early childhood educators and parents. Besides delivering the ScratchJr product, we will present our findings at conferences aimed for practitioners and researchers that focus on early childhood education such as NAEYC , SRCD and AERA, and human computer interaction such as CHI and IDC. We will publish in journals read by both communities, such as the Journal of the Learning Sciences, Early Childhood Research Quarterly, Interactions, and Journal of Science Education and Technology.

Expertise

The PI, **Marina U. Bers**, PhD, is an associate professor in the Eliot Pearson Department of Child Development and an adjunct professor in Computer Science at Tufts University. She heads the DevTech research group. Dr. Bers brings to the project expertise in the design and evaluation of developmentally appropriate educational technologies and early childhood technology-rich curriculum, as well as experience working with early childhood pre-service and in-service teachers. The Co-PI **Mitchel Resnick**, PhD, is Professor of Learning Research at the MIT Media Lab, and develops new technologies to engage children in creative learning experiences. He and his Lifelong Kindergarten research group will develop the software infrastructure for ScratchJr, building on their long experience developing innovative educational technologies, such as Scratch, for children. **Debbie LeeKeenan,** MA**,** is the Director of the Eliot Pearson Children's School. She will serve as a liaison with teachers and parents and will advise on literacy in the curriculum. **Brian Silverman** and **Paula Bonta** will serve as technology consultants on the project, contributing to the design and development of the ScratchJr software. The **advisory board** is composed of (see letters of support): **James Elicker,** PhD, Associate Professor and Director of Early Childhood Programs in Child Development and Family Studies at Purdue University, will bring expertise in early childhood education; **Herbert P. Ginsburg,** PhD, Jacob H. Schiff Foundation Professor of Psychology & Education at Teachers College, Columbia University, will contribute on early mathematics. **Rebecca New,** Ed.D, professor at School of Education at the University of North Carolina at Chapel Hill, will bring her experience on early childhood curriculum. **Doug Clements**, PhD and **Julie Sarama**, PhD, professors of education at University at Buffalo, SUNY, will contribute their expertise on software and curriculum for early childhood.

Results from Prior and Current NSF Support
PI Bers serves as PI on an NSF-funded project titled "Tangible Programming in Early Childhood: Revisiting Developmental Assumptions through New Technologies" (NSF DRL-0735657).The team developed the CHERP hybrid programming system, a complementary kindergarten robotics-based curriculum, as well as a research protocol and experimental tasks to study children's learning. Evaluation has been conducted in several kindergarten classrooms and at the DevTech lab with over 150 kindergartners and their teachers. Bers and colleagues published numerous journal articles, book chapters, and conference papers based on this research (Bers, M (in press); Kazakoff & Bers (2011); Horn, M.S., Crouser, R.J., Bers, M.U. (in press); Horn, M.S., Solovey, E.T., Crouser, J.R., and Jacob, R.J.K. (2009);Bers, M.U. & Horn, M.S. (2010); Horn, M., Bers, M., & Jacob, R. (2009).

Bers also received an NSF Career Award (2004-2009) titled "Communities of learning and care: Multi-user environments that promote positive youth development"(NSF IIS-0447166) that allowed her to develop a theoretical framework for technology rich educational interventions in real-world settings to promote positive youth development. Relevant publications are Bers, M., Beals, L., Chau, C., Satoh, K., Blume, B., DeMaso, D., & Gonzalez-Heydrich, J. (2010); Bers, M & Chau, C. (2010); Beals, L. & Bers, M.U. (2009); Bers, M (2006) and Bers (in press). Bers has experience working on interdisciplinary teams as she has served as co-PI in the following NSF grants "CUSP: Computing Undergraduate Scholars Program" (With Diane Souvaine, Robert Jacob, Misha Kilmer, Caroline Kao, Soha Hassoun); "Telling the Story - Learning Math, Science and Engineering Through Animation (With Chris Rogers- Tufts-, A Finkelstein, M Klawe & Rusinkeiwicz –Princeton-) (IIS-0511979); Multi-threaded Instruction: Forming Multi-disciplinary Research Groups to Improve Undergraduate Education. (With Chris Rogers, Eric Wang, Stephen Morrison, Caroline Cao

Co-PI Resnick served as PI on a previous NSF-funded project, titled "Developing a Media-Rich Networked Programming Environment for Community Technology Centers in Economically Disadvantaged Communities" (ITR-0325828; 09/15/03-08/31/08), which supported the initial development of Scratch and the study of how Scratch can foster the development of technological fluency in youth ages 8 to 16. Resnick and colleagues published numerous journal articles, book chapters, and conference papers based on this research (e.g., Kafai, Peppler, & Chiu, 2007; Maloney et al., 2008; Monroy-Hernández & Resnick, 2008; Peppler & Kafai, 2007; Resnick et al., 2009). Resnick is currently PI on three new NSF-funded projects (all granted in 2010) that will extend the development and study of Scratch as a learning environment:

- Scratch 2.0: Cultivating Creativity and Collaboration in the Cloud (CreativeIT #IIS-1002713)
- Preparing the Next Generation of Computational Thinkers: Transforming Learning and Education Through Cooperation in Decentralized Networks (Cyber-Enabled Discovery and Innovation #OCI-1027848)
- ScratchEd: Working with Teachers to Develop Design-Based Approaches to the Cultivation of Computational Thinking (Discovery Research K-12 #DRL-1019396)