# ITR: A Networked, Media-Rich Programming Environment to Enhance Technological Fluency at After-School Centers in Economically-Disadvantaged Communities

Principal Investigators:
Mitchel Resnick, MIT Media Laboratory
Yasmin Kafai, UCLA
John Maeda, MIT Media Laboratory

Information Technology Research (ITR) Program Solicitation NSF-02-168

## 1. Introduction

The MIT Media Laboratory and UCLA propose to develop and study a new networked, media-rich programming environment, designed specifically to enhance the development of technological fluency at after-school centers in economically-disadvantaged communities.

This project will build on our research team's previous experience and success in two areas: the development of innovative programming environments for youth, and development of innovative informal-learning centers for inner-city youth. Our team's research on "programmable bricks" has been commercialized as LEGO MindStorms, used by millions of youth around the world. Members of our research team co-founded the Computer Clubhouse project, a network of after-school learning centers for youth from economically-disadvantaged communities. The Clubhouse network has expanded to 75 sites in 14 countries, with 20,000 youth members, and it received the prestigious Peter Drucker Award for Nonprofit Innovation.

In our proposed project, we bring together these two themes to develop a new programming environment (to be called Scratch) that is grounded in the practices and social dynamics of Computer Clubhouses. Just as LEGO MindStorms added programmability to an activity deeply rooted in youth culture (building with LEGO bricks), Scratch will add programmability to the media-rich and network-based activities that are most popular among Clubhouse youth. Taking advantage of the extraordinary processing power of current computers, Scratch will support new programming paradigms and activities that were previously infeasible, making it much better positioned to succeed than previous initiatives to introduce programming to youth.

We expect that the use of Scratch at Computer Clubhouses will serve as a model for other after-school centers in economically-disadvantaged communities, demonstrating how informal-learning settings can support the development of technological fluency, enabling young people to design and program projects that are meaningful to themselves and their communities.

## 2. Rationale

A flurry of recent policy reports (*Being Fluent with Information Technology* from National Research Council, 1999; *Standards for Technological Literacy* from International Technology Education Association, 2000; *Technically Speaking: Why All Americans Need to Know More*

*About Technology* from National Academy of Engineering, 2002) have drawn attention to a critical societal problem: Even as new technologies proliferate and play increasingly important roles in all aspects of society, most people are "poorly equipped to recognize, let alone ponder or address, the challenges technology poses or the problems it could solve" (NAE, 2002).

To address this problem, the reports call for new initiatives to help people become more fluent with technologies. The NRC report defines "fluency" with information technologies as "the ability to reformulate knowledge, to express oneself creatively and appropriately, and to produce and generate information (rather than simply to comprehend it)." Fluency, according to the report, "goes beyond traditional notions of computer literacy…[It] requires a deeper, more essential understanding and mastery of information technology for information processing, communication, and problem solving than does computer literacy as traditionally defined."

In the past, most initiatives to improve technological fluency have focused on school classrooms (often computer-science classes). But there is a growing recognition that after-school centers and other informal learning settings can play an important role. The NAE report writes: "The informal education system must become a major focus" for promoting fluency. The NSF Directorate for Education and Human Resources describes informal learning as "self-directed, voluntary, and motivated mainly by intrinsic interests, curiosity, exploration, and social interaction." Thus, informal-learning settings are well-positioned to leverage young people's passions for new technologies as a starting-point for developing technological fluency.

The focus on informal-learning settings is especially important in economically-disadvantaged communities, where schools typically have few technological resources and many young people are alienated from the formal education system. After-school centers can serve as an important "middle ground" between home and school, providing a comfortable, supportive, and safe space for youth to explore new ideas and develop new skills.

During the past decade, more than 2000 community technology centers (CTCs) opened in the United States, specifically to provide better access to technology in economically-disadvantaged communities. But most CTCs support only the most basic computer activities (such as word processing, email, and Web browsing), so participants do not gain the type of fluency described in the NRC report. Similarly, many after-school centers (which, unlike CTCs, focus exclusively on youth) have begun to introduce computers, but they too tend to offer only introductory computer activities, along with educational games.

A small subset of after-school centers and CTCs, such as those in the Computer Clubhouse network, explicitly focus on the development of technological fluency, moving beyond basic computer skills and helping youth learn to design, create, and invent with new technologies (Resnick, Rusk, & Cooke, 1998). Walk into any Computer Clubhouse and you are likely to see youth creating and manipulating graphics, animations, videos, and music (and often integrating multiple media). The professional image-processing tool Photoshop is particularly popular. Indeed, a "Photoshop culture" has emerged at many Clubhouses, with youth proudly displaying their Photoshop creations on bulletin boards (both physical and online), sharing Photoshop techniques and ideas with one another, and helping Clubhouse newcomers get started with the software.

But there is a further step on the path towards fluency that is rarely seen at after-school centers and CTCs. Even at those centers focusing on fluency, youth rarely become engaged in computer programming. There is no “programming culture” analogous to the “Photoshop culture.” That is unfortunate: according to the NRC report, skills associated with programming play a “central role” in fluency. The algorithmic thinking inherent in programming, writes the NRC, “is essential to comprehending how and why information technology systems work as they do.” In addition, the report argues that “the continual use of abstract thinking in programming can guide and discipline one’s approach to problems in a way that has value well beyond the information technology-programming setting. In essence, programming becomes a laboratory for discussing and developing valuable life skills, as well as one element of the foundation for learning about other subjects.” Many others (e.g., Papert, 1980; Kay, 1991; diSessa, 2000) have made similar arguments on the benefits of learning to program.

Many previous initiatives to introduce programming to youth have not lived up to their promise. Too often, computer programming has been introduced using programming languages that are difficult to use, with proposed activities that are not connected to young people’s interests, and in contexts where no one has enough experience or expertise to provide guidance. As a result, many people now view computer programming as a narrow, technical activity, appropriate only for a small segment of the population. But that need not be the case. The extraordinary increase in computational power over the past two decades makes possible a new generation of programming tools and activities that can help overcome the shortcomings of previous initiatives, and make computer programming more accessible to everyone.

## 3. Goals

- Transform the use of technology at after-school centers, moving beyond basic computer activities to enable young people to achieve deeper fluency with information technologies

- Broaden opportunities for youth from under-represented groups to become designers and inventors with new technologies

- Advance understanding of the effective and innovative design of new information and communications technologies for informal math and IT education

- Make research-based educational technologies, and the ideas underlying those technologies, accessible to larger and more diverse audiences

- Further collaboration of young people across geographic, cultural, and language barriers

## 4. Research Context: Computer Clubhouses

We choose community technology centers as a research context in an explicit effort to reach youth from economically-disadvantaged and culturally-diverse communities.

Computer Clubhouses will serve as the primary sites for testing our new technologies and ideas. The MIT Media Laboratory co-founded the first Computer Clubhouse in 1993, in collaboration with The Computer Museum (now part of the Boston Museum of Science). Since then, the Computer Clubhouse Network (with major financial support from Intel) has expanded to more than 75 sites, with more than 50 Clubhouses in the United States, and international Clubhouses in India, China, Taiwan, Philippines, Mexico, Costa Rica, Colombia, Brazil, Ireland, Netherlands, Germany, Israel, and South Africa.

Computer Clubhouses are intended to provide youth (ages 10-18) from underserved communities with opportunities to design, create, and invent with new technologies, in order to become more capable, creative, and confident learners. Young people become *members* of the Clubhouse – at no cost to them or their families. Youth at Computer Clubhouses work on design projects based on their own interests and the needs of their communities. Adult staff and volunteer mentors play a critically important role in providing technical, intellectual, and emotional support for the youth.

Computer Clubhouse members use leading-edge software to create artwork, animations, and musical compositions. A visual-design culture has developed and is thriving at Clubhouses across the Network, with young people creatively expressing themselves with professional graphic-design and image-processing software. Many Clubhouses have also developed a thriving music-production culture. But, to date, only a handful of Clubhouse youth have become deeply engaged in computer-programming activities. Our proposed project aims to change that, providing the technologies and support that are needed to develop a "programming culture" at Computer Clubhouses.

## 5. Technology Development: Scratch

The centerpiece of our technology-development effort is a new programming environment called Scratch. Our project team will build upon many years of experience developing innovative programming environments, including Design By Numbers (currently used to introduce visual-art students to programming at more than 25 higher-education institutions worldwide) and LogoBlocks (which served as the basis for the programming language used in LEGO MindStorms).

The design and development of Scratch will be guided, at every step in the process, by the needs and constraints of Computer Clubhouses. Existing programming tools were not designed with places like Clubhouses in mind. By creating a new programming environment explicitly for Clubhouses, we believe that we can provide a better foundation for the development of a "programming culture" at Clubhouses (and other informal learning settings).

**5.1 Design criteria**. From our years of experience in Clubhouses, we have found that new tools (and associated activities) are used successfully and productively at Clubhouses over extended periods of time only if:

- youth see the tool/activities as "cool," resonating with their interests and passions
- youth see the value and potential of the tool right away
- youth can create a first project with the tool quickly and easily

- youth can create "products" that they can show off to others (pride of authorship)
- the tool supports a wide range of different types of activities
- the tool/activities appeal to youth of different backgrounds and cultures
- the activities fit into the social dynamic of the Clubhouse
- youth can learn features of the tool gradually and incrementally
- youth can continue to use the tool in ever more complex ways over time

**5.2 Core features**. With these design criteria in mind, we plan to design Scratch with the following set of core features. These features are designed specifically to address the problems that derailed many earlier efforts to introduce programming to youth.

- ***Building-block programming***. Scratch programming will be based on a building-block metaphor, in which learners build procedures by snapping together graphical blocks much like LEGO bricks or pieces in a jigsaw puzzle. Different data types will be represented by blocks of different shapes, with pieces fitting together in only syntactically-correct ways. This approach eliminates the possibility of syntax errors (which have proven to be a major obstacle for learning text-based languages), allowing youth to focus on the problems they want to solve, not the mechanics of programming. Each object in Scratch will have a library of primitive building blocks (based on the class of the object). Learners will drag-and-drop blocks from the library to create "stacks" (procedures) that govern behaviors of the object. Multiprocessing will be smoothly integrated into Scratch: different stacks of blocks will automatically execute in parallel. To allow a smooth progression to more complex programs, Scratch will include an underlying text-based programming language.

- ***Programmable manipulation of rich media***. The most popular projects at Computer Clubhouses involve manipulation of images, video, and music, using programs such as PhotoShop, Premiere, and Acid Pro. By contrast, initial activities in traditional programming environments typically involve manipulation of numbers and simple graphics. By providing Clubhouse youth with programmable control over rich media, Scratch opens up new programming activities that resonate more strongly with youth interests. For example, Scratch will include image filters similar to the ones in PhotoShop, but will provide programmable control over these filters, so that youth could create videos in which the parameters of these filters vary over time – and, in the process, get a deeper understanding of the concept of filtering and mathematical functions.

- ***Deep shareability***. Work at Clubhouses has an important social component: youth are constantly looking at one another's projects, trading ideas, sharing techniques. To fit into this context, we will design the object architecture of Scratch to support what we call "deep shareability" – meaning that youth will be able to share objects at all levels (from procedure blocks to animated characters to full projects) and to exchange them between all types of devices (desktops, laptops, tablets, handhelds, mobile phones, embedded devices). For example, a Clubhouse member designing a video game in Scratch could import (via the Internet) a character developed by a Clubhouse member in another country, integrate the character into her own game, download the game to play on a handheld device, control a LEGO construction (via RF) as part of the game, and then trade the game (via IR) with another local Clubhouse member. Through these activities, we expect that an *ecosystem of*

*Scratch creations* will develop, with Clubhouse youth constantly trading and modifying one another's creations. In our proposed research, we will use commercial handheld devices for some activities, but we will also develop new custom handheld devices (such as handheld displays for low-resolution videos, based on Maeda's LittleVision project).

- ***Seamless integration with the physical world***. Building on our previous research on LEGO/Logo and programmable bricks (e.g., Resnick, 1994; Resnick, Berg, Eisenberg, 2000), we will design Scratch so that youth can program physical objects (such as motors, lights, MIDI synthesizers) in the same way they program virtual objects on the screen – and use input from physical sensors (distance sensors, motion detectors, sound sensors) to control the behaviors of both physical and virtual creations. For example, a Clubhouse member could connect an accelerometer (with RF communicator) to her arm and program an animated graphic to change its behavior based on how she moves her arm (and, in the process, gain new insights into the concepts of acceleration, sensing, and feedback). As part of our research, we will develop new types of sensors and input devices, drawing on advanced sensor research at the MIT Media Lab and ITR-funded Center for Bits and Atoms (and adapting those research prototypes to make them appropriate for use in after-school settings).

- ***Support for multiple languages***. The Computer Clubhouse is a global community, with sites in more than a dozen countries, with young people speaking many different languages. Even in a single Clubhouse in the US, it is not unusual to hear three or four different languages in the course of an afternoon. To support collaboration and sharing in this context, it is essential for Scratch to be a multi-language, multi-cultural environment. In developing EToys and LogoBlocks, we found that the building-block programming approach makes it easy to handle multiple languages and character sets. EToys allows the language to be changed dynamically, even while scripts are running. The ability to effortlessly switch between languages will allow Clubhouse youth and mentors to work and think in the language most comfortable to them, while also allowing kids to read and change Scratch programs written by Clubhouse youth in countries halfway around the globe.

**5.3 Implementation**. Scratch will be written in Squeak, an open-source implementation of the Smalltalk-80 language. Squeak is extremely portable, with existing implementations for desktop platforms (Windows, Macintosh, Linux/Unix, Acorn, BeOS), handhelds (Windows CE, Zaurus OS, Compaq "Itsy"), and game consoles (Sony Playstation). Squeak has even been ported to hardware without any underlying operating system at all, including an experimental Mitsubishi processor (M32R/D) and a StrongARM hardware developer board. This extreme portability will allow Scratch applications to be deployed on devices with a wide range of form factors, from desktops to pen-based tablet computers to handhelds. We plan to port a subset of the Scratch runtime system to Java J2ME to allow deployment of selected Scratch applications on Java-enabled cell phones. Our group's experience with embedded processors (as part of our LEGO programmable-brick research), along with the increasing computational power available in low-cost, low-power packages, will also allow us to deploy Scratch content on pocket-sized and wearable devices, or on low-cost toys such as the Nintendo GameBoy Advance.

Sharing and exchanging of Scratch projects and their components will be supported through a combination of standard web servers (with content viewed using a browser) and a custom

"Scratch Object Library" server which we will build. The latter will allow Scratch components at the personal, Clubhouse, and Clubhouse network levels to be explored and downloaded seamlessly without leaving the Scratch environment. We will also develop ways to exchange Scratch components among handheld and embedded processors using IR and RF technologies, extending our earlier work on the memetag and i-ball projects (Borovoy et al., 2001). We also plan to develop new sensor technologies, leveraging research elsewhere in the Media Lab – for example, Joe Paradiso's work on sensor-equipped shoes that communicate via RF.

Scratch source code will be made freely available via periodic code releases to allow collaborators to augment the core system with their own custom features and extensions.

### 5.4 Related Programming Environments

Scratch draws ideas and inspiration from a number of other programming environments designed for young people or novice programmers. The building-block approach draws on previous research on LogoBlocks (Begel, 1996) and Etoys (Steinmetz, 2001), which have proven to be very intuitive for beginning programmers. Its user interface and page-navigation system are inspired by Logo Microworlds. Like AgentSheets (Repenning & Ambach, 1996), Scratch will encourage sharing of projects and components on the web, and like Boxer (diSessa, 2000), it will make program elements (such as variables) into visible, manipulable objects on the screen. Alice2 (Pausch, 1995) also uses drag-and-drop program construction to make programming easier for novices, but its domain is exclusively 3D. Scratch complements commercial animation and video production tools such as Macromedia Flash and Adobe Premier, providing youth with greater programmable control, and a framework designed to support the learning of important computational ideas.

## 6. Sample Scenarios

We will study how and what Clubhouse youth learn as they use Scratch to design and program *digital arts* projects – that is, projects that integrate art and technology for new forms of personal expression. In our previous research, we have found that digital-arts projects can serve as a particularly effective and engaging introduction to the activity and ideas of programming, especially in informal learning settings. In this section, we present three short scenarios of how we expect youth will use Scratch at Computer Clubhouses.

***Programmable image processing***. In Clubhouses today, youth often use Photoshop filters (e.g., blur, distort, pixelate, sharpen) to manipulate and transform photographs and scanned images. Scratch will provide youth with much greater control over the image-filtering process, expanding the expressive possibilities. In particular, Scratch will enable youth to program how a filter should be applied over time (writing mathematical functions to control how the parameters of the filter should vary over time). A Clubhouse member might start by programming the brightness parameter to control how quickly an image fades to black. Then, she might create a program to control the RGB values in the image (or the HSV values) to create videos with color-to-monochrome effects, or strange hue-shifting effects. Or she might program a filter that controls alpha values to create blue-screening effects. Later, she might place a fish-eye lens at the edge of an image, and program the movement of the lens so that it gradually spirals in, at an everquickening pace. By putting all of these programs together, she could create an entire video of special effects (with effects that would be impossible to create with standard software packages).

***Sensor-controlled music***. Many current Clubhouse members enjoy the arcade game "Dance Dance Revolution" (DDR), in which players dance on a floor pad with embedded sensors, aiming to synchronize their movements with music and images on the screen. With Scratch, Clubhouse youth could create their own version of DDR. A Clubhouse member could download MIDI files of songs from the Web (or compose and mix new songs in the Clubhouse music studio), design a floor pad with four touch sensors, connect the sensors to the computer, then create programs that check how well the dance steps synchronize to the music. Once the floor pad is in place, it could be used in different types of projects. Another Clubhouse member might decide to use the floor pad to create music (rather than follow it). She could write programs that map sensor inputs to different music outputs (using sound clips and music samples from the Scratch library). Later, she might add other sensors, to create new types of musical instruments.

***Networked animations***. Making animated characters (with tools such as Flash) is an increasingly popular activity at Clubhouses. With Scratch, Clubhouse members will be able to trade their animations with one another – and then track where their animations go, and how they are modified by others. Youth can trade their animations with members of other Clubhouses (via the Internet), or download the animations to handheld devices and trade them locally (via IR communication). Youth can modify the animations that they receive (since all Scratch "program blocks" are accessible), and they can even program their animations to behave in different ways depending on age or gender or location of the person receiving the animation. The Scratch server will automatically keep track of all transactions, so youth can view tree-like graphs representing the spread of their animations, with indicators of how and where the animations have been modified.

## 7. Results from Prior NSF Research

The proposed collaborative research project builds on the results of several successful NSF-funded projects conducted by Mitchel Resnick (PI) and Yasmin Kafai (co-PI) over the past decade. Previous and current NSF funding has allowed our project team to develop a strong foundation in four prime areas of the proposed research: educational technology development, design of new programming environments, informal learning environments, and diversity.

The technology development for Scratch will build on results from Resnick's three-year project entitled *Beyond Black Boxes: Bringing Transparency and Aesthetics Back to Scientific Instruments* (NSF grant CDA-9616444). That project focused on the development of new technologies (including "programmable bricks" and new programming environments) to enable youth to build their own scientific instruments, enabling them to become engaged in scientific inquiry not only through observing and measuring but also through designing and building (Resnick, Berg, Eisenberg, 2000).

Resnick has also received NSF funding for research involving informal learning. He is currently Principal Investigator for "PIE Network: Promoting Science Inquiry and Engineering through Playful Invention and Exploration with New Digital Technologies" (ESI-0087813). He is working with a network of museums to develop a new generation of public programs integrating art and technology, in the same spirit as the digital-arts projects described in this proposal. Resnick was also Co-PI of another informal-learning project (ESI-9627672), focused on the development of a major museum exhibit (the Virtual Fishtank) to help the general public learn important ideas from the sciences of complexity.

Resnick is also co-PI for the Center for Bits and Atoms (CBA), a large-scale NSF/ITR center (CCR-0122419) that develops and explores new technologies at the interface of the digital and physical worlds. Resnick leads the educational outreach initiatives for the center. Our proposed project will make use of some of the advanced sensor technology developed at CBA.

While Resnick's projects have focused on the design of new educational technologies, especially for use in informal-learning settings, the co-PI Yasmin Kafai has investigated the growth of a programming culture in elementary schools. Funded by a NSF Early Career award on "Learning Science by Design: Creating Information-Rich Learning Environments for Young Software Designers" (REC 9632695), Kafai followed a class of elementary students over the course of four years as they programmed instructional science simulations. This study pioneered a successful peer apprenticeship model in which "oldtimer" fifth-grade students worked together in teams with newcomer fourth-grade students on the multiple aspects of their software designs (e.g., Kafai & Ching, 2001; Ching, Kafai & Marshall, 2000; Kafai, Ching, & Marshall, 1998; Kafai, 1998; Ching & Kafai, under review). This grant influenced several national public policy studies, including the National Research Council's "Being Fluent in Information Technology" (1999) and the American Association of University Women's "Tech-Savvy: Educating Girls in the Computer Age" (2000).

The issue of diversity in the information technology workforce will be addressed in Kafai's current grant "Bridging the Gap" (PGE 0220556) which will synthesize findings from over 400 studies sponsored in the last ten years by the National Science Foundation and the American Association of University Women (AAUW). The goal is to summarize research and practitioners' efforts to address gender equity and diversity issues in technology, mathematics, science and engineering. The final goal is a report of best practices that will be published in late 2003 by the AAUW.

## 8. Research Plan

**8.1 Research Questions**. Our research will investigate how Scratch can support the development of technological fluency in informal-learning settings. Our research will focus on the following three areas:

***Engagement.*** We will study how young people become engaged in programming activities and what keeps them interested in pursuing more complex projects.

- <u>Entry points</u>: How do young people become interested in using Scratch? What aspects of Scratch do they find most engaging?
- <u>Activities</u>: What types of projects do they choose to develop?
- <u>Persistence</u>: How does youth interest in Scratch evolve over time? What supports are needed to keep them involved in projects?

***Learning*****:** We will study what programming concepts and abilities youth learn as they work with Scratch, and what factors support and impede their learning.

- Concepts**:** What computational and related concepts do youth learn through use of Scratch?
- Capabilities: How does work with Scratch further young people's abilities to engage in sustained reasoning, debug problems as they arise, break complex problems into simpler parts, express their ideas in new media (developing ideas from initial conception to completed project)?
- Reflection: In what ways do young people talk about their programming projects, both online and in person?

***Community*****:** We will study how programming knowledge is built and shared within and across after-school centers.

- Sharing projects: What types of Scratch objects and artifacts do youth share with one another? What do they share with friends and family outside the center? How does the availability of handheld devices influence patterns of sharing?
- Sharing ideas**:** How does use of Scratch spread over time within and across sites? What types of programming techniques and strategies do youth share with one another?
- Support**:** What are the supports and barriers to the sharing of knowledge within and across sites? What role do staff and mentors play? How do youth support one another?

**8.2 Research Approaches**. To address these research questions, we will use several approaches for collecting and analyzing data. We have chosen methods that are most appropriate for informal learning settings, where participation is self-directed and voluntary (Falk, Brooks, & Amin, 2001).

***Participatory Design*****:** We will use a participatory design approach (Schuler & Mamoika, 1993) to inform the design of the Scratch programming environment. This process will guide the iterative development of interfaces, features, and activities that are accessible to youth and resonate with their interests.

- Cooperative Prototyping Sessions: We will engage in rapid prototyping of Scratch programming environment based on bi-weekly interactions with youth at each Boston and Los Angeles Clubhouse research site. Researchers and youth will cooperatively explore each prototype and discuss possible changes to help facilitate use of the tool (Druin, Bederson, Boltman, Miura, Knotts-Callahan, & Platt, 1998).
- Project Storyboarding: We will discuss project ideas with Clubhouse members using a low-tech prototyping approach (Scaiffe & Rogers, 1998). We will use our scenarios as conversation starters to explore possible digital-arts project activities.

***Case Studies*****:** We will use case studies, conducted in the ethnographic tradition, for two purposes: to study engagement and learning by individual Clubhouse youth, and to study community knowledge-building within and across Clubhouse sites. Our study of how individual youth develop fluency with Scratch will be based on a framework and interview protocol developed by the Center for Children and Technology (CCT, 2002). The Clubhouse-wide studies will construct portraits of the use of Scratch and related technologies over time, based on the work of Oakes & Margolis (2000). The portraits will allow us to compare and contrast the use of Scratch at different Clubhouse research sites.

- Baseline data. At the beginning of the study, we gather baseline data (through field observations and interviews) on youth interests, relationship and experience with technology, patterns of technology use, and level of technological fluency prior to introduction of Scratch.
- Field observations: Researchers from the project team will visit each Clubhouse research site in Boston and Los Angeles on a weekly basis, using observation forms to record what activities and tools are in use, types of Scratch activities, and differences in girls and boys' choice of activities.
- Participant observations will include interactions with members and report more closely on the kind of activities and on-going conversations at the clubhouse. We will pay particular attention to ways in which Clubhouse members deal with learning new Scratch features, debugging situations, and providing explanations and help to others. These observations will be documented in field notes.
- Participant interviews: We will talk with Clubhouse youth about their Scratch projects to better understand what aspects of Scratch activities they find most engaging, and to assess their evolving understanding of computational concepts and development of technological fluency, based on the CCT protocol. We will also interview Clubhouse coordinators to gain a fuller picture of youth engagement and learning, and how Scratch fits into other events and activities in the Clubhouse.

***Analysis of Network Activity and Digital Artifacts.*** We will study the artifacts created by Clubhouse youth, sharing of artifacts among youth, and online discussion about Scratch projects.

- Individual Development: We will study individual member's portfolios, from initial starter projects to more complex digital-arts projects, to analyze the evolution in the complexity and expressiveness of their projects, and the development of their programming and design skills.
- Collective Knowledge Building. We will study the evolving database of projects across the Clubhouse Network, to analyze how new techniques and project ideas spread within individual Clubhouses and throughout the extended Clubhouse community. We will also study how the introduction of new devices (including handhelds and mobile phones) influences the spread of projects and ideas.
- Evaluation by Peers: In addition to "expert" analysis by project researchers, the team will also seek young people's evaluation of programs created within the community. We will study which kind of digital-arts projects are held in high esteem within the Clubhouse community and ask members for their rationales.

### 8.3 Timeline

Initially, we will test our new technologies primarily at Clubhouses in low-income neighborhoods in Boston and Los Angeles, so that our research teams at MIT and UCLA can iteratively redesign our technologies and activities in collaboration with Clubhouse staff, mentors, and members. In Year 3, we will expand to Clubhouses throughout the network, and in Year 4 to other after-school centers, to examine if and how programming cultures form with less direct support from university researchers. We have selected sites with varied characteristics (age of Clubhouse, level of technology fluency, type of community organization) to gain an understanding of how Scratch might work in a variety of settings.

| Year One: Initial Scratch Development and Participatory Design Studies |
| --- |
| - Collect baseline data of activities at Clubhouse research sites<br>- Initial development of Scratch for desktops and laptops<br>- Development of hardware to connect sensors and output devices<br>- Participatory design studies at Clubhouse research sites (2 Boston, 1 L.A.)<br>- Iterative development of Scratch based on participatory design studies<br>- Initial testing of Scratch network and database infrastructure |
| **Year Two: Field Studies, Iterative Development, and Handheld Implementation** |
| - Field observations (weekly) at 4 Clubhouse sites (2 Boston, 2 L.A.)<br>- Participant observations and interviews (weekly) at 4 Clubhouse research sites<br>- Iterative refinement of Scratch based on observations and interviews<br>- Implementation of Scratch on handheld devices and tablets |
| **Year Three: Initial Dissemination and Digital Artifact Evaluation** |
| - Ongoing field observations at Clubhouse research sites<br>- Implementation of selected Scratch applications on cell phones<br>- Ongoing participant observations and interviews at Clubhouse research sites<br>- Translation of Scratch into multiple languages, with support for world character sets<br>- Initial workshops at Computer Clubhouse Network annual and regional meetings<br>- Analysis of online projects and discussions from additional Clubhouse sites |
| **Year Four: Final Research Studies and Broader Dissemination** |
| - Final field observations at Clubhouse research sites<br>- Make Scratch publicly available for download via the Web<br>- Workshops at national conferences for Boys & Girls Clubs, CTCNet, ASTC<br>- Workshops at Clubhouse Network regional meetings and international Teen Summit<br>- Analysis of online projects and discussions from additional sites<br>- Research publication and presentations at national conferences |

## 9. Dissemination and Broader Impact

***Dissemination of Technology***. Our project team has an exceptional track record for getting our research-based technologies out to the world in large numbers. The LEGO Mindstorms robotic construction kit, based on Media Lab research, has been used by millions of youth around the world. Design By Numbers is available free of charge on the Web and is currently in use at more than 25 higher education institutions worldwide as a core curriculum component in the digital arts. StarLogo is also available free of charge and is in use in both high-school and university courses. Similarly, we will make Scratch available free of charge on the Web, and will promote it to maximize public distribution.

***Dissemination of Results of the Research***. We plan to share and discuss our research findings with other researchers in the fields of educational technology, informal science and technology education, IT and computer science education, and human-computer interface research.

Members of the project team regularly make presentations at leading academic conferences and industry symposia, including CHI (Computer-Human Interaction), SIGCSE (Technical Symposium on Computer Science Education), OOPSLA (Object-Oriented Programming Languages, Systems, and Applications), CSCL (Computer-Supported Collaborative Learning), ASTC (Association of Science Technology-Centers), ICLS (International Conference of the Learning Sciences), and AERA (American Educational Research Association). Members of the project team have a strong record of publishing in leading academic journals, as well as policy and broader-circulation publications.

***Dissemination through Informal Education Networks***. We will broaden the impact of the project by offering workshops through after-school and other informal-education organizations and networks, focusing specifically on organizations serving economically-disadvantaged communities. These workshops will focus on modeling engaging and effective approaches to support the use of Scratch. In Years 3-4 of the project, we will offer hands-on workshops at the annual conference of the Computer Clubhouse Network, and we will also collaborate with Clubhouses to offer in-depth workshops directly for young people through regional and Teen Summit gatherings. In Year 4, we also plan to offer workshops at national conferences for CTCNet (a national network of more than 1,000 community technology centers), the Boys & Girls Clubs of America (which serves more than 3.3 million young people, primarily from disadvantaged circumstances), and the Association of Science-Technology Centers.

## 10. Key Personnel

The multidisciplinary team for this project includes leaders in the fields of educational-software design, digital media design, human-computer interaction, and informal learning.

**Mitchel Resnick**, director of the Lifelong Kindergarten group at the MIT Media Laboratory, specializes in the development and study of computational tools to help people (particularly children) learn new things in new ways. Resnick's research group led the development of the ideas and technologies underlying the LEGO MindStorms robotics construction kit (and the associated graphical-programming language). He is co-founder of the Computer Clubhouse network of after-school learning centers and the NSF-funded PIE museum network. He led the development StarLogo (the first massively-parallel programming language for nonexperts) and the NSF-funded Virtual Fishtank museum exhibit, both designed to help people learn about complex systems. Resnick was awarded an NSF Young Investigator Award in 1993.

**Yasmin Kafai**, associate professor at the UCLA Graduate School of Education & Information Studies, has led NSF-funded research projects that studied children as software designers of interactive games, simulations and archives for learning science and mathematics. She has been influential in several national policy efforts from briefing the National Research Council for "Being Fluent with Information Technology" (1999) to serving on the national commission of the American Association for University Women for "Tech-Savvy: Educating Girls in the Computer Age" (2000). Kafai received an Early Career Award from the National Science Foundation in 1996 and a fellowship from the National Academy of Education in 1997.

**John Maeda** is director of the Aesthetics & Computation research group at the MIT Media Laboratory. Maeda is the recipient of the highest design career honor in the United States, the 2001 National Design Award, and Japan's equivalent honor, the 2001 Mainichi Design Prize. Maeda's work has focused on ways to combine his MIT background in computer science with his later training in Japan in the arts. His award-winning Design By Numbers (DBN) project leads "mathematically challenged" visual-art students through a series of exercises as a means to teach the spirit of computer programming. Maeda's research group is currently developing a graphics programming system called "Proce55ing."

**John Maloney**, a visiting researcher at the MIT Media Lab, was an member of the original Squeak development team at Apple Computer and Disney Imagineering, and made major contributions to Squeak's virtual machine, user interface, and sound, music, and networking facilities. At Disney, John headed the software team for a prototype handheld device for Disney theme parks combining a digital camera, map, guidebook, location sensing, and games.

**Natalie Rusk** co-founded the Computer Clubhouse project and PIE Museum Network. She has guided the application of new technologies to support informal science learning at The Computer Museum, Science Museum of Minnesota, and the Exploratorium.

**Brian Silverman** has led the development of the leading commercial versions of the Logo programming language. As a consultant to MIT, he has guided the technology development of Programmable Bricks, StarLogo, and LogoBlocks.

Graduate students: **Yvonne de La Pena Ay** worked as a software engineer before joining the UCLA Graduate School of Education & Information Studies as a graduate student. **Leo Burd**, before coming to MIT as a graduate student, worked as a software engineer and helped organize a network of community technology centers in his native Brazil. **Nina Weber**, now a graduate student at UCLA, brings to the project her masters degree in design, technology and learning from Stanford University and an extensive career as a teacher working in bi-lingual inner-city classrooms.

## 11. Advisory Board

Our advisory board brings together a multidisciplinary and multicultural group of innovators, with expertise in the fields of computer science, informal learning, design, and community development. The board includes:

**Amy Bruckman**, Associate Professor in the College of Computing at Georgia Tech
**Stina Cooke**, Project Developer for the Intel Computer Clubhouse Network
**Mark Guzdial**, Associate Professor in the College of Computing at Georgia Tech
**Alan Kay**, Senior Fellow at HP Labs and creator of Smalltalk and Squeak
**Geetha Narayanan**, Director of Srishti School of Art, Design and Technology, Bangalore, India
**Elliot Soloway**, Professor, Dept. of Electrical Engineering & Computer Science, Univ. Michigan
**John Henry Thompson**, inventor of MacroMedia Director Lingo scripting language
**Shira Womack**, Strategic Initiatives and Pipeline Programs, Global Diversity, Intel Corp.