Hook-ups: How Youth Learn Through Creating Physical Computer Interfaces

by
Amon Daran Millner

Bachelor of Science in Computer Science
University of Southern California, 2001

Master of Science in Human-Computer Interactions
Georgia Institute of Technology, 2003

Submitted to the
Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements of the degree of
Master of Science
at the Massachusetts Institute of Technology

June 2005

Signature of Author ____________________________________
Program in Media Arts and Sciences
May 6, 2005

Certified by ____________________________________
Mitchel Resnick
LEGO Papert Associate Professor of Media Arts and Sciences
Program in Media Arts and Sciences, MIT

Accepted by ____________________________________
Andrew B. Lippman
Chairperson
Departmental Committee on Graduate Studies

Hook-ups: How Youth Learn Through Creating Physical Computer Interfaces

by
Amon Daran Millner

Bachelor of Science in Computer Science
University of Southern California, 2001

Master of Science in Human-Computer Interactions
Georgia Institute of Technology, 2003

Submitted to the
Program in Media Arts and Sciences,
School of Architecture and Planning, on May 6, 2005
in partial fulfillment of the requirements of the degree of
Master of Science
at the Massachusetts Institute of Technology

June 2005

ABSTRACT

The Hook-ups project introduces a new set of tools, materials, and activities intended to support children in creating physical computer input devices for computer programs they write. This project introduces a new approach to learning-through-design by providing opportunities for children to engage in both physical and computational design concurrently. This thesis describes the design of Hook-ups tools and materials, including the development of Scratch Patches - a new puzzle-piece-like set of technological building blocks used to build computer input devices. Also presented are classifications of the types of Hook-ups developed by youth, an analysis of what and how youth learned through Hook-ups design activities, and a roadmap for future work in the area of interaction design for children.

Thesis supervisor: Mitchel Resnick
Title: LEGO Papert Associate Professor of Media Arts and Sciences

Hook-ups: How Youth Learn Through Creating Physical Computer Interfaces

by
Amon Daran Millner

Advisor ____

Mitchel Resnick
LEGO Papert Associate Professor of Media Arts and Sciences
Program in Media Arts and Sciences, MIT

Reader ____

Neil Gershenfeld
Associate Professor of Media Arts and Sciences
Director, Center for Bits and Atoms, MIT

Reader ____

Mel King
Senior Lecturer Emeritus
Department of Urban Studies and Planning, MIT

Reader ____

Katherine Bielaczyc
Assistant Professor of Technology in Education
Graduate School of Education, Harvard University

ACKNOWLEDGMENTS

My academic advisor, my thesis readers, and I are credited on the title pages for making this thesis what it is, but I would be remiss not to acknowledge not only them, but a wider set of wonderful people who supported this work.

Mitchel Resnick, my academic advisor, does an amazing job of guiding his students. In him we find a friend, an inspiration, and a person who works tirelessly to help everyone around him reach their potential. Thanks M…

Neil Gershenfeld continues to set an example for how to blend multiple technologies to make new types of design possible. I appreciate how he passionately pursues his ambitious goals but always finds time to advise me on my technical developments. Thanks for showing me how to make (almost) anything.

Mel King embodies what a community leader and educator should be. He makes an excellent thesis reader as well. Thanks for being Mel (and for the food).

Katerine Bielaczyc understands many issues in technology in education around theory, design, and practice. I benefit from her growing interest in informal learning and value her opinion. Thanks for always being willing to work with me and bringing a smile to my face when I need one.

The members of the Media Lab's Lifelong Kindergarten group all deserve recognition, but explaining how each one of them helps make both my research and life rich would take another document the length of this thesis. I would specifically like to thank Leo Burd for being an extraordinary officemate, friend, and partner in community service. Working with the far-flung Scratch research team (from Boston-based members led by John Maloney and Margarita Dekoli to our Los Angeles partners led by Yasmin Kafai to our Canadian collaborator Brian Silverman) continues to be a pleasure.

I cherish the support I receive from the Computer Clubhouse community, from the founders to the members. Danielle Martin, Marlon Orozco, Edgar Gresores, and John Harrison are especially great Hook-ups helpers. I am also fortunate to receive support from a growing community of Fab Labbers. Hook-ups work would not be the same without Amy Sun, Sherry Lassiter, John DiFrancesco, and Raffi Krikorian.

People who have helped me make advances in sharing my work through writing about work I do in informal settings include: Amy Bruckman, Mary Helen Ramming, David Millen, Bob Irwin, and Pamela Siska.

It goes without saying that without my family and friends pushing me, the Hook-ups project would probably not exist. I know that my grandfather, who seeded my inventiveness, lives on in me. My mother shows me how to be strong in the toughest of circumstances. My father exemplifies someone who is passionate about educating others.

My amazing aunts and uncles invest love and time into my development as if I were their child. My cousins, nieces, and nephews continue to show me that the future is bright.

The Hook-ups research project was also made possible by financial support from:

- Scratch project (NSF EIA-0325828)
- Intel Corporation and Intel Foundation for the generous support of the Computer Clubhouse Project
- MIT Center for Bits and Atoms (NSF CCR-0122419)

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

In 2003, Eisenberg challenged designers to blend physical and virtual activities to improve the types of computer-enhanced educational experiences envisioned by Papert in the 1980's (Eisenberg, 2003; Papert, 1993). This thesis reflects the author's desire to take on this challenge and discover more engaging ways for youth to blend technology with design. The research presented in this document charts the exciting, yet relatively unexplored, area of how youth creating customized physical computer input devices permits new entry points to (and alternative pathways through) computer programming and physical design activities. I have developed the "Hook-ups" system: a set of tools, activities, and materials to support youth aged 10-18 in learning through constructing physical input devices that control interactive computer programs. Youth make Hook-ups to control games, animations, and other computer programs they create. In making Hook-ups, youth work with materials ranging from everyday objects to output from personal fabrication devices such as laser-cutters.

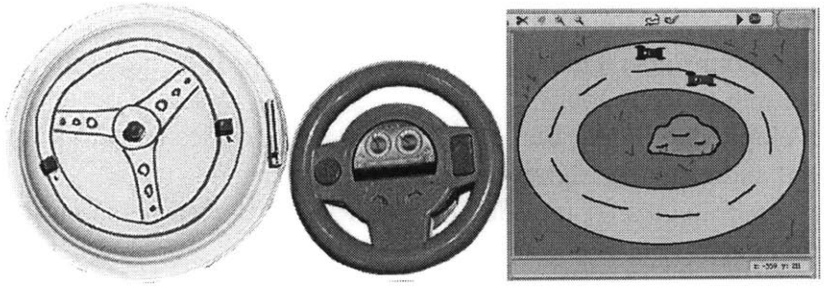


Figure 1: Two examples of Hook-ups for a computer driving program

For example, a Hook-up steering wheel can be constructed in numerous ways, depending on the materials a learner chooses to start with. In making a "basic" Hook-up, a learner could draw a steering wheel on an empty paper plate, tape four pushbuttons onto the drawing, connect those buttons to a computer through a serial cable, and write a computer program that moves an animated car around the screen in different ways for each button (Figure 1). Alternatively, the learner could make a "repurposed" Hook-up by taking apart an electronic toy steering wheel, using wire to connect its buttons to a computer (via a sensor interface board). In a third option, making a "fabrication" style Hook-up, a learner could use computer-aided-design (CAD) software to draw a steering wheel shape, use a laser-cutter to cut the shape from a piece of acrylic (plastic), place button cut-outs (from an adhesive copper sheet) on the steering wheel print-out, and add a tiny homemade circuit board to the back of the structure and connect it to the computer running the interactive car simulation.

Each Hook-up creation process includes integrating some combination of the following elements: sensors, physical materials to which sensors attach, sensor-querying computer interface boards, and sensor-input-ready programming environments.

There are three main ideas on which the Hook-ups project is based:

(1) Design is a good context for learning;
(2) Design activities should take learners' individual styles into account;
(3) Supporting design-based learning in informal-learning environments is challenging, yet possible.

This thesis presents the findings of an ongoing study I am conducting to examine how and what people learn by creating Hook-ups. The three main goals of the study are:

(1) To deepen our understanding of how the Hook-ups learning system can inspire creative output from youth;
(2) To develop technologies that enable a new range of design activities that combine physical and virtual design processes;
(3) To design strategies for introducing a new system to learners in informal-learning environments.

Preliminary findings suggest that the Hook-ups system provides a wide range of good design contexts. One of the most important outcomes was that youth learned core design principles through combining physical and virtual design to build input devices in informal settings. These learners primarily exercised two design principles:

- building complex objects from simple parts
- embracing the unexpected.

This document reports on insights I gained through reaching the study's goals. The chapters that follow ground Hook-ups work in the theories relating to learning-through-design, explain the design of the Hook-ups system, introduce new technology developed for the system, detail selected Hook-ups created by youth, describe the methods used to introduce youth to Hook-ups and observe their learning, provide analysis of how and what youth learned through Hook-ups activities, and suggest future directions for Hook-ups research.

2. BACKGROUND AND FOUNDATIONS

Much of this thesis responds to Eisenberg's challenge to blend physical and virtual design activities. This chapter presents the educational approach on which Hook-ups work is based: constructionism. Sections describe how Hook-ups work is situated in a trajectory of projects. The trajectory started on a project established before the term "constructionism" was coined. The trajectory thus far ends with current projects related to Hook-ups that are intended to shape the means of learning-by-constructing in the future.

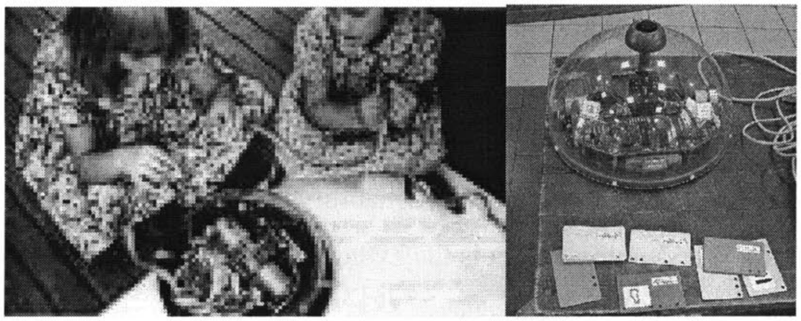


Figure 2: LOGO controlled Floor-Turtles

The first section visits the first physical object one could design with by programming it: the "Floor-Turtle" (Figure 2). The next chronicles how the LEGO/LOGO team combined computation and LEGO bricks to increase the range of physical and virtual design activities accessible to youth. The following section demonstrates how well-known craft activities such as paper origami can be combined with computer and personal fabrication technology to create homemade computational crafts. The next section further develops the idea of how personal fabrication tools will empower youth to invent technologies in their own neighborhoods through the Fab Lab project (Mikhak et al., 2002). The last section presents a design challenge a Fab Lab member wanted to overcome that helped bring about the development of certain technological tools for the Hook-ups project.

2.1 Constructionism and the Floor-Turtle

Papert's constructionist theory of learning suggests that young people learn best through the process of constructing artifacts (Papert, 1986). Long before coining the term constructionism, Papert realized that computers were powerful tools that could help children build and learn. Eisenberg's challenge was based on the ideas Papert put forth in the 1980's about using computers as enhanced educational tools. However, Hook-ups work expands upon one of Papert's ideas from the 1960's: to expand the range of learning activities for youth by enabling them to change the physical computational tools they use.

Papert's work in the 60's demonstrated that computers were powerful sandboxes for children with the help of a computationally enhanced robot turtle. This turtle engaged children in playful computer programming by enacting computer commands issued by the children. The Floor-Turtle helped Papert lay the foundation for youth learning through controlling computers to explore the physical world, but the learner could not modify the turtle itself. Papert lamented the inability to give children the power to change the physical form of technological tools they were learning with. He felt that this drawback limited children's inventiveness.

2.2 From Floor-Turtle to Programmable Bricks Via LEGO/LOGO

In the 1980's, a project called LEGO/LOGO, started by Mitchel Resnick and colleagues, fulfilled one of Papert's early wishes by integrating computation with LEGO brick building blocks (Resnick, 1991). At the core of the original LEGO/LOGO kit, an interface box linked (through a tethered wire) a desktop computer running the LOGO programming environment with sensors and motors attached to physical LEGO constructions. A family of construction kits and projects that combine physical and computational design has grown out of the LEGO/LOGO system.

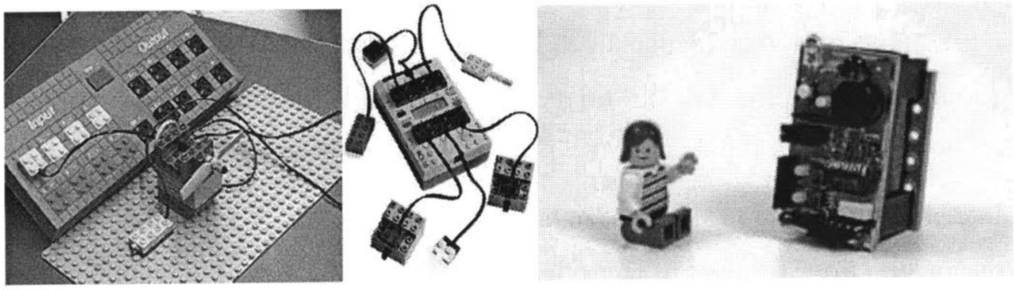


Figure 3: LEGO/LOGO, LEGO RCX brick, Cricket

One LEGO/LOGO descendant is the "programmable brick" – a device that stores its own programs without being tethered to a computer, thus creating opportunities to take computation into the physical world (Resnick, 1993; Resnick, Martin, Sargent, & Silverman, 1996). Many variations of the programmable bricks have emerged, including the RCX brick commercialized in LEGO's Mindstorms robotics kit and a series of tiny devices called Crickets (Mikhak, Berg, Martin, Resnick, & Silverman, 2000). These programmable bricks have been used in many different settings for different types of applications such as interactive art, robotics, and music.

Hook-ups tools, activities, and materials can be used in a variety of applications as well, but are particularly designed to support youth combining physical and computational design to create their own physical interfaces – an area which branches of LEGO/LOGO family tree have not been designed to support.

2.3 Computationally-enhanced Crafts

Similarly to how LEGO/LOGO used technology to enhance the already popular and educational activity of building with LEGO bricks, Mike Eisenberg and colleagues have developed a series of computationally-enhanced crafts in recent years that use computers to bring new life to craft activities children have enjoyed and learned from for many years (Figure 4). Craft technology activities have involved designing with computers to generate output that would then be used as craft material. For example, a powerful software tool for modeling origami shapes, Hypergami, can turn a printed piece of paper into a ready-to-build origami creation (Blauvelt, Wrensch, & Eisenberg, 1999). Craft technology work also leverages the abilities of personal fabrication devices (such as laser-cutters) to broaden the range of types of material with which computers can generate output (Blauvelt et al., 1999 ; Eisenberg et al., 2003).

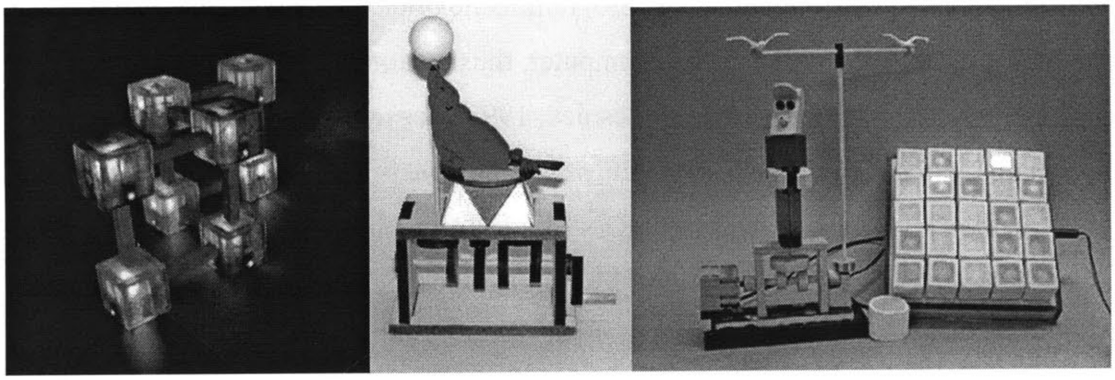


Figure 4: Computationally-enhanced crafts from Craft Technology project

As with part of the craft technology work, one thrust of the Hook-ups project looks to create new design activities by learning from past ones. Connecting output from traditional craft activities to personal computers is supported. Hook-ups work also supports connecting to computers materials generated by personal fabrication tools. The factor that sets Hook-ups work apart from existing computational crafts work is the act of re-connecting the generated craft to the computer to form an input device rather than a creative physical structure.

2.4 Increasing Access to Personal Fabrication Tools

Many learners with access to now-ubiquitous paper-printing technology can take part in design activities offered through projects like Hypergami. The same cannot be said for learners wanting to take part in the small but growing number of design activities that require personal fabrication tools. The MIT Center for Bits and Atoms' "Fab Lab" project intends to remedy this lack of access. Fab Labs, centers that make personal fabrication tools available to communities, are currently being deployed in several countries in preparation for what Gershenfeld calls the forthcoming age of digital fabrication (Gershenfeld, 2005). A Fab Lab's equipment (computer-controlled knives, desktop milling machines, and laser-cutters similar to the model depicted in Figure 5) approximates the vision of personal fabrication machines that will be as ubiquitous as laser printers in the future.

Figure 5: An Epilog laser-cutter

2.5 Personal Fabrication for Self Expression

The growing number of Fab Labs in underserved communities around the world is making it possible for youth to work with personal fabrication devices as they wish in informal settings. The still-developing learning model for some Fab Labs resembles parts of the learning model from a similar network of informal learning centers: Computer Clubhouses (Resnick & Rusk, 1996).

Part of the Clubhouse learning model entails giving young people the support they need to express themselves through technology (Resnick, Rusk, & Cooke, 1999). I have mentored at Computer Clubhouses since 2001. I have noticed that they can be great places for youth to gain access to computer technology and support from mentors. I endeavor to discover ways to better use this access and support to help youth find more avenues for self-expression through physical and virtual design. I began to mentor at a Boston area Fab Lab in the beginning of 2004. I quickly realized that, with engaging design activities, youth would be able to combine physical and virtual design in these locations in ways that were previously not accessible to youth. (The Boston Fab Lab and Computer Clubhouses mentioned in this thesis are referred to by pseudonyms, as are the names of the young Hook-ups builders.) My suspicion that Fab Labs would, like Clubhouses, serve as safe places where youth could follow their interests with the support from mentors was confirmed when a member shared with me a desire to express himself by customizing a gaming controller. This desire ultimately helped me improve the Hook-ups project.

Don's Desire to Customize His Input Device

One of the members of the Commu-Tech Fab Lab, Don, wanted to have the ability to temporarily change the physical configuration of a game controller he used so that he could be more efficient at certain tasks he encountered in his favorite game. At times, he held the controller in awkward positions to complete different tasks and was annoyed by this. He expressed interest in making a button on the front of his controller appear on both the front AND the back of his controller for different uses. Don and I were both aware that no vendor offered a controller that fit Don's needs. We also knew of no building blocks available to build customized controllers from parts. Don's unique desire helped me think about what custom-made input devices and their building blocks could look like and sparked one of the redesigns of the early Hook-ups systems.

3. HOOK-UPS SYSTEM DESIGN

This chapter describes the design of the Hook-ups learning system. A learning system is a combination of an educational philosophy, tools embodying the philosophy, and a setting in which the tools are used. An effective Hook-ups system requires an appropriate combination of tools, engaging activities, a community of learners, and learning spaces conducive to physical and virtual design.

This work required creating appropriate activities around Hook-ups tools. An iterative approach to the system's design allowed insights from using the system in the research sites to continually influence the development of parts of the system. Improvements to areas of the system included:

- redesigning existing Hook-ups tools and developing new ones
- adding/refining effective activities and introduction strategies
- making better connections to resources in the learning space
- adding complementary design tools and materials to the learning space

3.1 Hook-ups Tools and Materials

The constructionist learning philosophy influenced decisions during the design processes of the Hook-ups hardware tools, software tools, and construction materials. The design criteria of "low floors with high ceilings," familiarity, and expandability also guided their development.

Low floors with high ceilings

Hook-ups activities needed to have easy starting points for novices (low floors). Advanced Hook-ups builders needed increasingly challenging projects (high ceilings).

Familiarity

The ability to integrate familiar materials into Hook-ups projects represented a low floor for entry. When activities incorporate objects learners are already familiar with, more learners can become involved – especially when time is constrained (Kahn, 1999). For example, learners may have already played out plots for puppets they already own; turning that familiar puppet into a Hook-up and using it to control the flow of an animated story based on an existing plot could decrease the amount of time necessary to start the activity.

Expandability

With parts one finds, borrows, or makes, one can make simple mechanisms. These simple mechanisms can be combined to create increasingly complex objects. The increasing complexity of objects that can be made using simple parts can afford increasingly challenging projects with high ceilings for ambitious Hook-ups builders.

3.1.1 Hardware Tools

A simple interface board was used to connect the sensors on many Hook-ups to computers. Over the course of this project, new hardware tools were developed (that were influenced by the standard Hook-up sensor-querying computer interface board shown in Figure 6). These developments are chronicled in chapter 5.

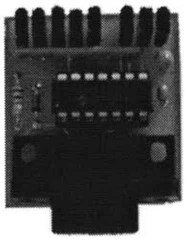


Figure 6: A standard Hook-up board

Hook-ups sensor boards could turn analog readings from up to eight (16 in some cases) sensors into signals computer programs could understand. If the board depicted in Figure 6 were connected to a computer's serial port, if a light sensor was plugged into sensor plug 1 (top left) and a pushbutton was plugged into sensor plug 2 (adjacent to plug 1), then the board would constantly send to the computer signals reflecting how much light the sensor detects and when the pushbutton is being pressed. Programs on the computer determine what to do with this information.

3.1.2 Software Tools

Most Hook-ups were programmed in the Scratch programming environment (Resnick, Kafai, & Maeda, 2003), the centerpiece of a technology development effort by the

Lifelong Kindergarten group at the time of this thesis. Scratch supports standard Hook-ups boards. Listed below are some features of Scratch that made programming (with sensor inputs) manageable for youth.

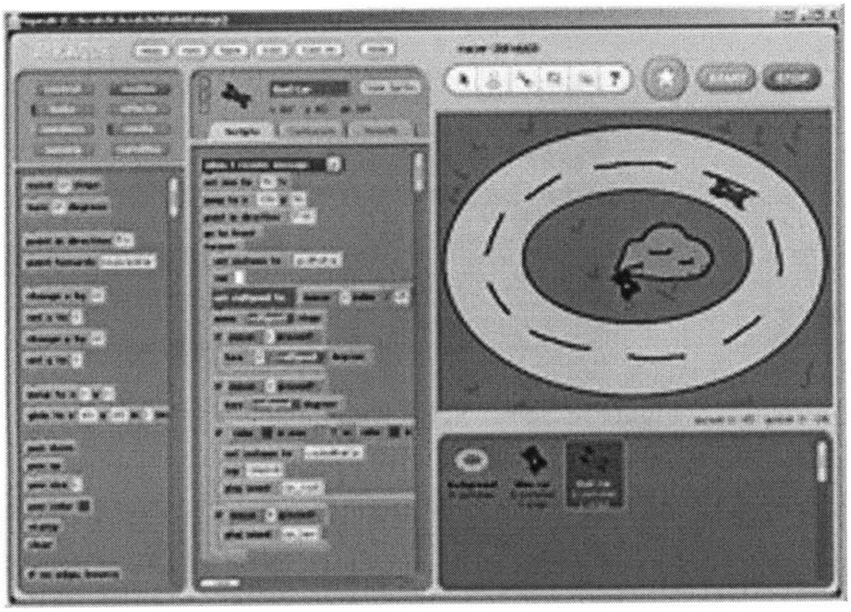

| command | scripting | stage (top) |
| blocks | area | Scratch object library (bot) |

Figure 7: The Scratch programming environment interface

Seamless integration with the physical world

Through Hook-up boards, youth can connect sensors to the programming environment in order to control the behaviors of their virtual creations.

Building-block programming

Scratch programming is 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. (See Figure 7 for an example of a script and Figure 8 for samples from the variety of current Scratch command blocks.) This approach virtually eliminated the possibility of learners encountering errors for mistyping commands (which have proven to be a major obstacle for learning text-based languages).

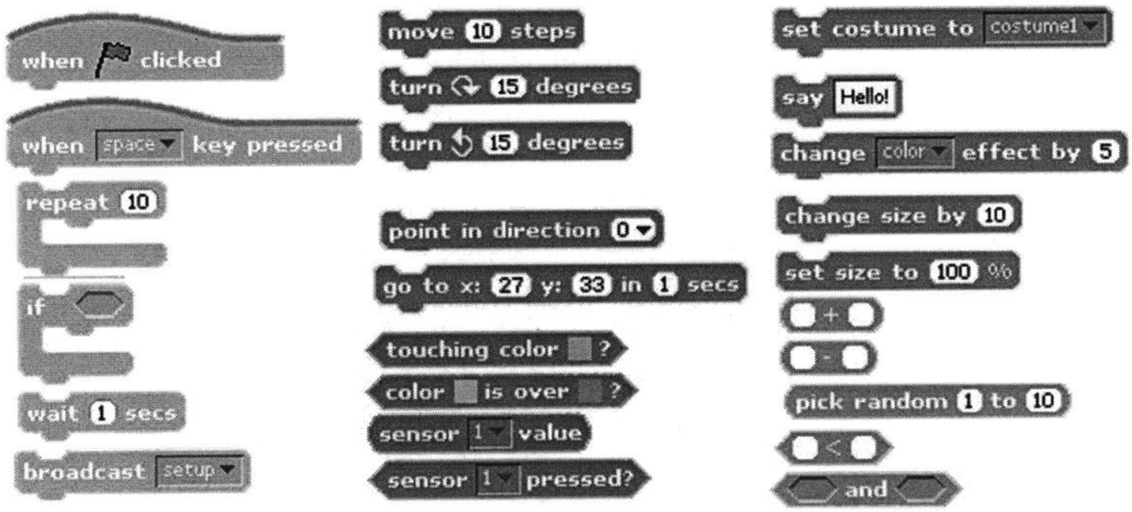


Figure 8: Select Scratch command blocks

Scratch provides command blocks that allow users to control objects on the screen. These control and sensing blocks can be mapped to keyboard keys, mouse movements, or external sensors.

Programmable manipulation of rich media

Scratch provides youth with programmable control over rich media, so input devices can be made to manipulate media such as music and images in new ways. For example, an input device with a temperature sensor attached could control when a programmable "color filter" is applied to change the color of an on-screen character's face. A user could hold the temperature sensor until the temperature changes and program the character's face to turn red as a result.

Scratch is implemented in Squeak (Steinmetz, 2001), an open-source implementation of the Smalltalk-80 language that is extremely portable and capable of running on various computing devices. Multiple development releases of Scratch were used during this research.

3.1.3 Construction Materials

Sensors were attached to a wide range of physical materials, from everyday objects to output from personal fabrication devices such as laser-cutters. Materials included paper cups, existing electronic devices, and stock materials such as acrylic used with personal fabrication tools. Sensors were fastened to these materials using glue, tape, press-fit edges, string, or whatever else the learners found useful (see Figure 9 for an example of how a sensor can be mounted to construction materials).

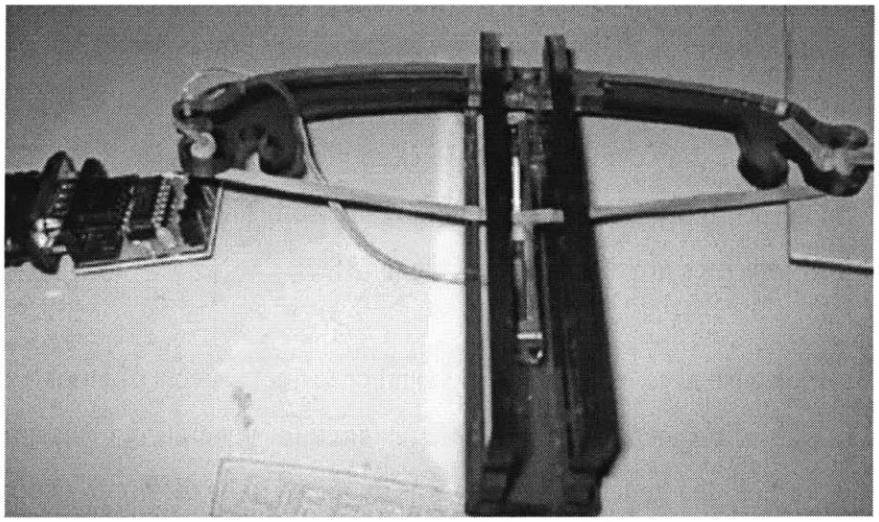


Figure 9: A press-fit acrylic crossbow housing a sliding potentiometer

3.2 Learners in Informal Settings

The target users for Hook-ups research were youth aged 10-18 in informal after-school learning environments. I worked with local Computer Clubhouses (Resnick & Rusk, 1996) and Fab Labs. Diverse sets of learners populated these after-school drop-in centers and had opportunities to use Hook-ups tools. Invariably, learners came to activities with unique experiences, diverse interests, and varying access to resources. This included both youth who approached design activities with excitement and curiosity and those who had previously been uninterested in (or frustrated by) such activities and had elected not to participate. Not everyone using the Hook-ups system was expected to like it or learn from it, let alone push the ceilings of the technologies. But in designing the system, I included opportunities for complex input devices and did my best to allow for unanticipated uses. In my conception, the youth who worked with Hook-ups should have been able to:

- see the technology and activities as relevant to their interests and passions
- see the value and potential of the tools right away
- create a first project with the tools quickly and easily
- recover from mistakes without too much hassle
- manage their physical and virtual creations over time
- find a low barrier for entry and a high ceiling for potential projects

3.3 Activities to Facilitate Building Hook-ups

Most Hook-ups activities were designed to introduce an area of programming or physical design to explore. In chapters 6 and 8, I discuss how I created activities such as introducing sample projects that would serve as launching points for design projects. In the event of a smooth introduction, the facilitator would tailor the rest of the time with the learner to realize ideas they came up with. Backup plans were necessary at times, in case someone lost interest in one activity, for whatever reason. These were called "re-start" activities (more activities are discussed in chapter 8).

Whether a learner wished to explore the physical or virtual aspects of Hook-ups first, showing examples of projects was a valuable resource for generating ideas amongst learners. The project idea generation phase was expected to be different, depending on the size and makeup of the group receiving the introduction. Later chapters illustrate several strategies used for introducing youth to Hook-ups.

3.4 Learning Spaces Conducive to Physical and Virtual Design

Activities that accompanied early versions of Hook-ups tools were designed to mesh well with the Computer Clubhouses and Fab Labs in which they were introduced (described in detail later in this chapter). The facilities and the people in these learning spaces were generally conducive to exploration, free of judgment, full of encouragement, and able to sustain mentor support. Although youth were able to re-arrange parts of the centers for their needs (such as putting on a play), the staff would usually determine the set-up of the learning spaces. Sometimes introducing physical or virtual aspects of Hook-ups activities required that minor changes to the space be made prior to starting the activities. Unpacking physical Hook-ups examples, bringing craft materials to a table, and setting up a laptop or a projector were necessary at some points and not at others – but obtaining permission to make changes was never an issue.

In order to work on physical projects like Hook-ups over time, it was imperative that safe places were available to house Hook-ups artifacts. Centralized storage for virtual information on a networked hard drive enabled learners to retrieve files from wherever they chose to work. The delicate process of sustaining children's interest in new types of activities ran more smoothly with such storage support in place.

I found that a laptop machine was conducive to Hook-ups work because it could be placed in areas where learners had room to design physical constructions. Many learning spaces (that do not stock laptops) designate craft areas and computer areas – but not areas where learners are expected to bridge the two activities. Hook-ups create the catalyst for placing computers on the physical craft table or making space on the computer tables for crafts. In a Clubhouse in Mexico, I witnessed a member building a model airplane from balsa wood and recycled electronics less than 10 feet away from a peer who was detailing the 3D plane he drew on the computer (Figure 10). Without support for hooking the two projects up, both learners missed an opportunity to program, bring each other's project to life, and learn more about design from another perspective.

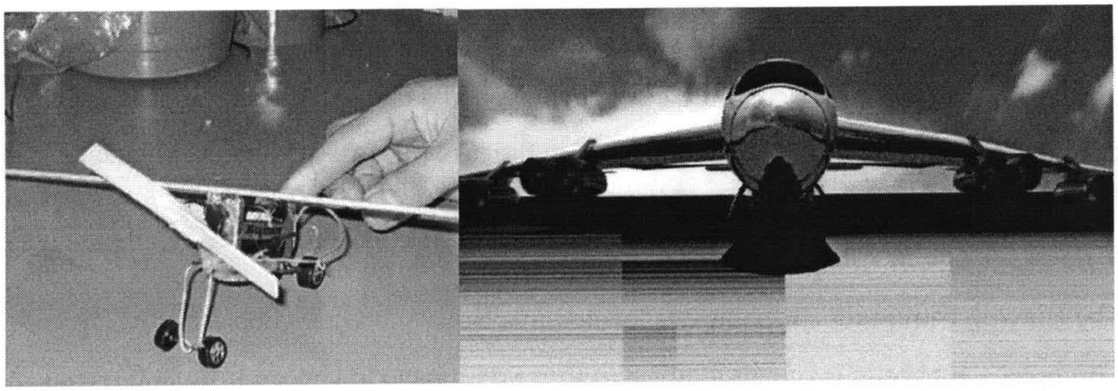


Figure 10: Clubhouse members' physical and virtual airplane projects

Mentor support

Learning centers like Computer Clubhouses run best when they receive support from mentors who understand the center's learning approach. Understanding the approach is one issue; the act of mentoring is another. I believe that providing guidelines for mentoring is worthwhile, but also feel that mentoring is essentially an art. In Hook-ups settings, mentors were both adults and youth. One of the many services rendered by mentors in Hook-ups work was to ensure that learners' ideas were taken seriously and, when necessary, "smoothed" (adapted) into projects learners could achieve with the given resources. What was considered a realistic project was up to the mentor (this issue is covered in chapter 8).

It is important for mentors to challenge youth. A young mind's potential is sometimes wasted when an adult does not expect that youth can handle challenging projects. Children are generally not opposed to hard activities, as long as they consider them fun as well. However, overly difficult challenges can have lasting effects on learners, such as low self-esteem in areas (or "I'm not good at that" mentalities).

Mentors who made challenges to learners according to what they thought they could handle, were informally assessing what Vygotsky calls the learners' Zones of Proximal Developments (ZPDs) (Wertsch & Stone, 1985). The ZPD is the difference between what someone can do by themselves versus what they can do with the help of someone or something else. Either the mentor would ensure that the learner could meet the challenge or the Hook-ups tools would have to help the learner develop as a designer. In an attempt to provide better tools for enabling youth to design and create their own input devices, I developed a set of building blocks to assist in the task (discussed in the next chapter).

3.4.1 The Clayton Computer Clubhouse Research Site

The Clayton Computer Clubhouse is located in a Boys and Girls Club facility. The Computer Clubhouse is one of many activities youth have to choose from at Clayton. During periods of the day, youth of certain age groups are allotted time slots for certain activities in Clayton. For example, at 3 p.m., the Computer Clubhouse might be open to 10- to 12-year olds while the gym is open for teens. Clayton's hours of operation were typically 2 p.m. to 8 p.m. Monday through Friday. As with most Computer Clubhouses, Clayton has:

- over a dozen windows-based personal computers featuring several professional design applications
- an area where craft materials can be found
- a music studio,
- a large table where the community congregates.

3.4.2 The Commu-Tech Center Research Site

The Commu-Tech Center is situated within a set of housing developments. Many of the Center's members live in one of the apartment buildings affiliated with the Center. Commu-Tech is open daily from roughly 10 a.m. to 9 p.m. Monday through Friday and from 10 a.m. to 6 p.m. on Saturday. The hours at Commu-Tech are very flexible because the Center is considered open whenever a staff member is present.

The Commu-Tech population is comprised of mixed-generation learners. People from the ages of 4 through "I don't count any more" visit to take formal and informal classes that range from Fuel Cells to Fabrication. Technology in the Center is also available for drop-in use when classes are not in session. The Fab Lab area features:

- five linux-based personal computers
- a laser-cutter (with cardboard and acrylic available as stock material)
- a vinyl-cutter (with adhesive-backed copper and plastic as stock material)
- a desktop milling machine
- workbench space with soldering irons and glue guns
- circuit-making electronic components.

3.4.3 Other Places Hook-ups Were Created

Hook-ups reported in this document were primarily created at Commu-Tech and Clayton; however, members of other Computer Clubhouses have also built sample projects. (Over the course of a year, I typically get to spend time at several Clubhouses, in Boston and elsewhere.) Some Hook-up work was also conducted during visits to the MIT Media Lab.

4. DEVELOPING SCRATCH PATCH BUILDING BLOCKS

In the initial stages of Hook-ups research, Hook-ups builders would connect their creations to computers through standard Hook-up boards (Figure 6). By using these boards with youth, I began to notice some of their drawbacks, such as:

- limited to eight sensor plugs per board
- sensor plugs were unlabeled
- sensor plugs were extremely close together
- limited to passive resistors
- boards lacked protective or decorative casing
- redesigns were time-consuming due to the PCB mail-ordering process

I attempted to work around some of the standard Hook-up board's drawbacks. For example, I sometimes populated the first two or three sensor plugs with wires that terminated with alligator clips that learners could easily clamp to sensors. Learners typically assumed (correctly) that these clips represented the first ports and thus knew the orientation of the ports without labels. The raw PCB board left little room to label the ports with numbers. In addition to making minor improvements to the standard Hook-ups board's sensor plugs, I ultimately created a new technology called Scratch Patches to expand the work that could be done with Hook-ups.

Scratch Patches are a set of puzzle-piece-like technological building blocks used to build computer input devices. They are an enabling technology for building complex input devices from simple components. Each Patch is a puzzle-shaped plastic structure with an embedded microcontroller. Patches connect together like traditional puzzles to form a sensor network that connects to a computer. The Scratch Patches send information from their sensors to programs running on the computer.

The following sections detail how Scratch Patches overcome several drawbacks of the standard Hook-up board. By doing so, they also made noteworthy contributions to personal fabrication research and fields related to learning and technology. Scratch Patches represented the first building block set created to help youth construct custom physical input devices. It also was a computational construction kit developed in a community center where youth could not only produce input devices with the kit, but also had the ability to produce the standard kit or a version they modified. Below is an example of how Scratch Patches could be used.

Sample Scenario

In making an interactive fishing application, for example, a learner could arrange and decorate Scratch Patches in configurations that resemble lakes (Figure 11). They could use everyday materials to make magnetic model fishing poles and decorate Patch surfaces. Dangling magnets over Patches that hold magnetic sensors could then trigger the Scratch Patches to send a message to a Scratch program running on the computer, causing a virtual fishing rod to catch a virtual fish in the Scratch program.

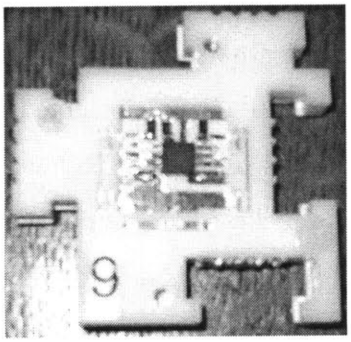


Figure 11a: A single Scratch Patch

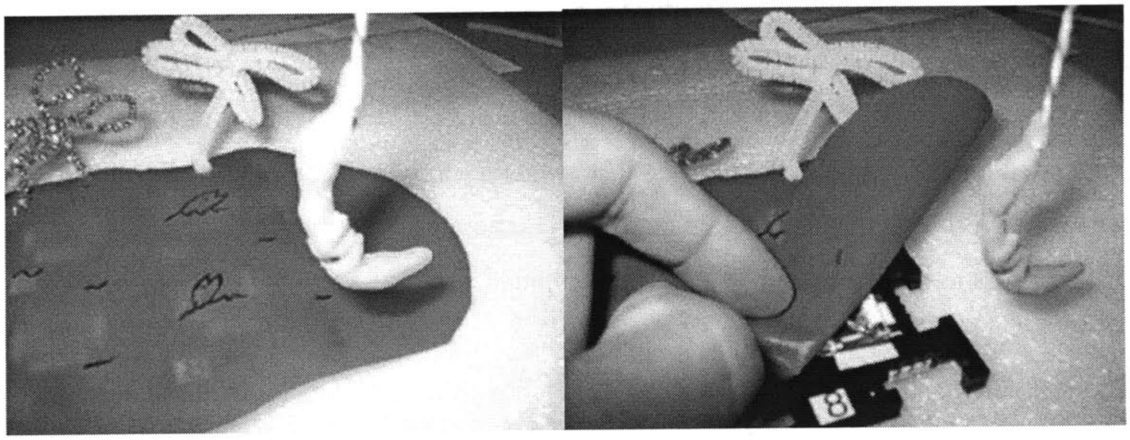


Figure 11b: A sample Scratch Patch interface – the Fake Fishing Hole

The rest of this chapter presents the system requirements, original conception, initial design, subsequent redesigns, the current design, and the next directions for Scratch Patches.

4.1 System Requirements

Developing a technological tool that fit into the greater goals of the Hook-ups research required that it was usable, reliable, and extensible.

Usable by youth

This interface was designed for youth who were familiar with how puzzle pieces connected. The Scratch Patches pieces leveraged a puzzle metaphor to afford familiarity and make building multi-sensor input devices intuitive.

Reliable and durable

An objective of the project was to make Patches durable enough to withstand some bending and dropping, while maintaining the ability to make good connections with its neighbors.

Extensible

Patches were meant to be integrated with sensors and traditional craft materials. Areas on the surface were designed to secure craft materials. At any time, youth could design their own layers to stack upon Scratch Patch bases or create completely new Patches, making the system highly extensible.

4.2 Scratch Patch Design Process

All the tools necessary for building Scratch Patches must be in place before beginning. One Scratch Patch required the following materials:

- solder
- ribbon cable – approximately 4 ft
- 1 female serial DB9 snap-on plug
- 1 male parallel DB25 snap-on plug
- 1 8-pin SOIC programming test clip
- 1 Atmel ATtiny15L microcontroller
- 10K resistors – approximately 3
- 1 sheet of adhesive-backed copper – approximately 8" by 8"
- 1 sheet of adhesive-backed epoxy film – approximately 8" by 8"
- 1 piece of acrylic – approximately 6" by 6"
- sensors – for example, a magnetic reed switch.

The steps I took to implement each Scratch Patches design follow. Details of the initial Scratch Patch design and significant iterations that led to the current design are then presented.

Design the Form Factor and Draw it

When thinking about what kinds of objects people are familiar with putting together, puzzles came to mind. The idea that one puzzle piece has a tab and another has a receptacle for tabs influenced how I drew the Scratch Patch form factor (Figure 12 below). This tab design naturally led to a card edge connection between pieces (discussed later).

Design Circuit(s) for the Form Factor

With the surface space I allocated in the form factor design, I had to design a circuit that afforded the required Scratch Patch functionality, yet fit in a constrained space. Because young people would be helping build these devices, the KISS principle ("keep it simple, stupid") applied. I desired few jumps and not many components for youth to solder; since some projects require many Patches, a smaller production time was desired. The surface mount electronics size of 1206 was selected because such components are small enough to create circuits on small amounts of surface area yet they are large enough to handle with adult fingers. The circuit, like the form factor, was designed in a CAD program. This approach was unconventional because most circuits are designed in dedicated circuit design software such as Eagle®.

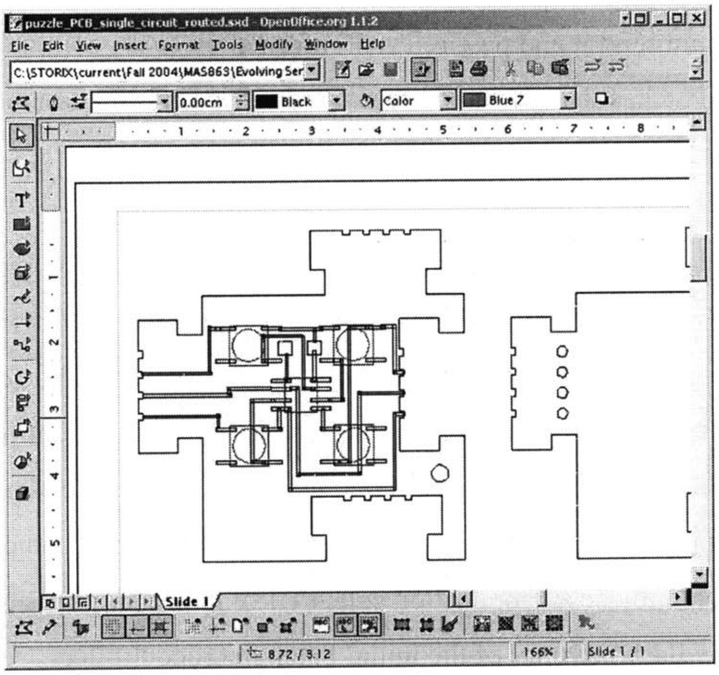

puzzle_PCB_single_circuit_routed.sxd - OpenOffice.org 1.1.2
File Edit View Insert Format Tools Modify Window Help
C:\STORIX\current\Fall 2004\MAS863\Evolving Ser
0.00cm
Black
Color
Blue 7
Slide 1
8.72 / 3.12
166%
Slide 1 / 1

Figure 12: Initial Scratch Patch circuit design in CAD program

Cut and Paste the Materials

The designs for the form factor were sent to a laser-cutter that cut Patches from acrylic sheets. The designs for the circuits were sent to a vinyl-cutter that carved circuit outlines into a sheet of adhesive-backed copper. The unwanted copper in the circuit was removed and the remaining circuit was transferred to an epoxy film that could withstand heat from soldering irons. The epoxy was cut out around the circuit and wrapped around the blank acrylic Patch shape. Electronic components were then soldered to the circuit traces.

Write the Program and Download It to the Patch Microcontroller

Using a standard text editor, I wrote sets of "rules" to tell each Patch's microcontroller how to function. The rules were written in a computer language (made specifically for the Atmel brand microcontrollers) called AVR Assembler. The rules written in computer code were saved as assembly files. The commands in the program file would tell the microcontroller how to manipulate voltage changes received from analog sensors and create digital signals to send to computers as bytes. These bytes were formatted for Scratch to understand them. Several variants of the core Patch program were necessary. Any Patch that communicated with the computer needed code to decode the information other Patches would send it and pass relevant information to the computer through a serial cable. The Patches that read and transmitted sensor information did not need code to talk to the computer.

A special cable was needed in order for computers to transfer assembly programs to the microcontroller. This cable was made by connecting one end of a 25-wire ribbon cable to a male parallel port plug (DB25). Five wires on the other end would be connected to an 8-pin SOIC chip test clip. By connecting this test clip to the microcontroller, I was able to read a special calibration code on the chip that was used to calibrate all chips to communicate at the same speed (9600 bits per second). With a method called "bit banging," I was able to use an application called UISP to put programs from my computer onto the flash memory of the initial Patch microcontrollers.

4.3 Initial Design

The initial Scratch Patches started as 2" by 2" squares on a computer screen (Figure 13). They would become puzzle-piece-shaped 1/8" thick pieces of plastic bearing Atmel microcontrollers that could connect to form computer input devices.

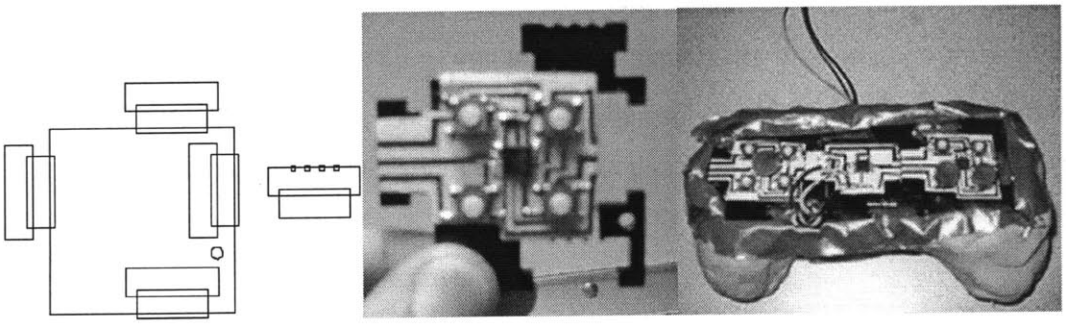


Figure 13: An Initial Scratch Patch early drawing, form factor, and sample application

The design process used to make the Scratch Patches used a combination of current fabrication tools, namely the laser-cutter and the vinyl-cutter, to make a physical form and a flexible circuit.

4.3.1 Initial Patch Physical Properties

Patches were connected to each other by sliding the tab from one Patch into the tab-insert on another. The Patches were designed in such a way that only one orientation could connect circuits (so that learners were less likely to short-circuit them while connecting them). A short circuit could result from one of the three contact leads touching the contact lead on another Patch that does not carry the same signal. Three copper trace lines that run through each Patch's microcontroller and fold over the Patch's edge (creating a card edge connection of sorts) are (from top to bottom) for power, data, and ground. Connecting mismatched signals could cause the input device to function incorrectly or to damage Patches. A considerable portion of the time spent designing the form factor was allotted to minimizing the chances of incorrect connections.

Slide-fit connections

Two Patches stay connected to each other without external adhesive because of the force that they put on each other. Placing tabs of two Patches side-by-side leaves wiggle room and does not press-fit together because, as the laser-cutter's beam goes over any path, the power settings determine the width of the line the laser disintegrates as it passes. Cutting Scratch Patches was done at low speeds (3) and high power (100) settings and thus decreased the width of the Patch by a few mil (1/1000in) on each side. This reduction in size causes plastic Patches without circuits on their surfaces to have wiggle room when they are connected tab-to-tab because there is no tension between the tabs. The few mils missing from the original width of the Patch are replaced and slightly exceeded when the circuit is bent around the edges of the plastic pieces. The circuit adds the width of the epoxy sheet to the width of the copper fixed to it. The combination of copper, epoxy, and plastic can compress to a certain extent. When a person slides copper-bearing tabs together, three things happen:

(1) materials on each tab compress and apply force to the other tab to keep the pieces connected securely;
(2) sliding one tab over the other cleans the copper surfaces of each to make sure electrical current can flow down a stream of many connections;
(3) the tabs connect three leads that send power, ground, and data lines through each Patch that is connected.

Enabling circuit expansion

Initial Patches were designed to expand the circuit of networked Patches by connecting to each other from left to right along one layer. This design decision was made due to initial difficulties encountered when attempting to route the circuit's three main data lines in four directions without adding extra components to solder. I needed to keep the processes required to build the Patches as simple as possible because I would need to build many to test if connecting multiple Patches was possible and reasonable. The initial communicator Patch design could connect to the computer through a serial cable port on its left side and to sensor or extender Patches on its right. The designs for initial sensor Patches varied, but they all had circuit connection points only on the left and right sides, and not on the top and bottom.

I created extender Patches to give builders options to expand the size and shape of their creations without using power-consuming sensor Patches where sensing was not desired. Initial extender Patches were built by simply wrapping a strip with three copper lines horizontally around the center of a Patch. Later, I realized that I could make creations with multiple rows of Patches possible. I realized that I could improve the extender Patch by folding the 3-line strip over itself in such a way as to create four connector tabs on a Patch. This folding was possible because no electronics were in the way and because the layer of epoxy below the copper prevents overlaying traces from connecting.

I found that the new extender Patches could expand circuits on different layers. Because the power, ground, and data lines were bent around the edge of each Patch, portions of those lines were exposed on the bottom of the Patch. Similar to how power, data, and ground passed through Patches at the tab connections, Patches with flat copper lines could be attached to the bottoms of Patches to extend the circuit underneath layers.

A good example of where circuit expanding would be useful is an interactive game of checkers. To make a checkerboard, the learner would need to arrange multiple rows of Patches – some with light sensors to determine if a physical checker piece was on them.

One could make rows of Patches possible by using extenders on the end of every row, or running them underneath the board.

The checkerboard could probably not be the size of an actual board because a circuit of networked Patches could extend only as far as the power (electrical current) could reach. Tiny losses of power occurred at each connection point and more significant amounts of power were consumed by each microcontroller in the circuit. The next subsection covers the electronic aspects of initial Patches in detail.

4.3.2 Initial Patch Electronic Properties

Each Patch (excluding extenders) contained a microcontroller-based circuit. Microcontrollers are chips that have built-in memory and interface circuits like analog-to-digital converters (ADCs). The microcontrollers used in Scratch Patches were Atmel ATtiny15Ls (Atmel) – shown in Figure 14.

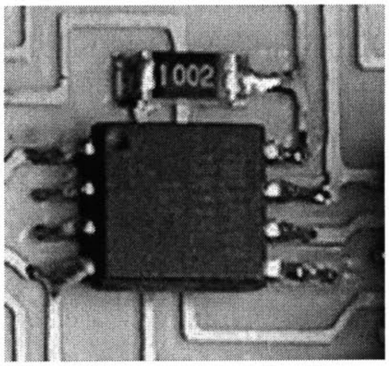


Figure 14: An ATtiny15L microcontroller

Microcontrollers needed power (current) to operate. A communicator Patch drew power through a cable connected to a computer's serial port (which provided +/- 12 Volts and ~ 10mA when a computer program opened the port for communication). A communicator Patch would convert the power into a 5V signal that could operate the microcontrollers of the communicator and the other Patches networked together. The initial communicator Patch relied upon the small amount of power drawn from the serial port to eliminate the need for bulky DC power adapters or battery packs (in early stages of development). The power drawn from a serial port was enough to power three Patches reliably.

Communication between Patches and the computer relied heavily upon timing. The network of Patches communicated using a software-implemented UART (universal asynchronous transmitter/receiver) protocol. A communication speed of 9600 bps was set. Each Patch connected to a communicator consumes some power to operate and leaves the rest on the shared power line. If a Patch did not draw ample power from the shared power line, it could not communicate at the speed the other Patches were and would report inaccurate sensor readings as a result.

Most testing of the code written to communicate sensor information to a network was performed using one communicator Patch and two sensor Patches. Chips on these Patches needed slightly different programs.

A communicator Patch needed a program that could read the shared data line and pass the information to a computer. A sensor Patch needed a program that could read the state of digital pins or ADC pins, append information to designate which number sensor the Patch represented in the network, and report that information to the communicator Patch when no other Patch was using the shared data line.

The first attempt at code to determine when the shared data line was available to transmit data over drew upon a technique used in Ethernet-based computer networks. I wrote several variants of random backoff algorithms that call for Patches to check the data line for communication for a set amount of time and wait for a specified time interval if other Patches were transmitting data before checking again.

4.4 Subsequent Redesigns

Several design iterations were made based on the performance of the initial system, feedback from peers, and limited exposure to youth at Commu-Tech.

Connecting Patches

Problem – sharing the data line: An initial ineffective routine for preventing data collision on the shared data line prompted me to consider and try out several design alternatives.

Colliding data created unreliable sensor information and would confuse learners – with two sensors attached, oftentimes bad data created by collisions gave the appearance that valid sensor data was being reported from more than two sensors.

Alternative 1 – "neighbor-to-neighbor": I could have replaced the shared data line with a neighbor-to-neighbor routing scheme. This alternative called for each Patch to know which one of its neighbors it should pass information to and listen to. This approach would hinder expandability because when later versions of the Patch were implemented with tab connections on all four sides, it would be difficult to determine which neighbors should communicate. At a time when I was using most of the 5 available input/output pins in the circuit design, I favored the shared data line because it required only one I/O pin (as opposed to neighbor-to-neighbor's 2-pin requirement).

Alternative 2 – "master polling": An alternative to the decentralized approach taken with the random backoff code would have been to deem one Patch "master." Being master gave the communicator Patch the ability to tell other Patches when to send information. The communicator Patch would use the shared data line to request a sensor reading from the Patch designated as #1, then #2, etc. After each request, the communicator would be in listen mode long enough to receive two bytes from the designated Patch (if it was attached). Each sensor Patch on the network would listen for its designated number every time the communicator polled the network and respond by sending its sensor data.

Alternative 3 – "time multiplexing": Similar to the second alternative, the third took a signaler/listener approach. The differentiating feature was that the communicator sent out a signal to which every Patch was programmed to respond, like the gun shot at the start of a race. The communicator would then listen for all sensors to report responses within a certain time interval before it sent another signal for updated information. Each Patch was programmed to represent a certain sensor in the network. For example, a sensor #4 Patch would detect the start signal on the shared data line, then wait the amount of time it would take for sensors #1-3 to transmit their information, before sending its own sensor readings. Then sensor #4 would wait for the next signal.

Considerations and current implementation: The time multiplexing approach is currently implemented. It required slightly less time than master polling and fewer lines of code.

Problem – insufficient power: In order to enable input devices with more than eight sensors, the inability for the serial port to power more than three Patches needed to be fixed.

Considerations and current implementation: I considered adding a port on the communicator Patch for a DC power adapter but elected not to as I found adapters cumbersome to carry around. I considered designing a Patch with a battery pack but did not always want to rely upon fresh batteries to work with Scratch Patches. I decided to draw power from another source on the computer: the USB port. Most USB ports could supply ~1A of power, which could reliably power more than eight Patches (more tests would be needed to determine if 16 current patches can run reliably from the available power).

Constructing Patches

Problem – keeping thin circuit traces down: Remaining affixed to the Patch surface was a problem for the tiny traces in the initial circuits.

Considerations and current implementation: The decision to initially mount sensors on Patches called for different circuit designs for (almost) each type of sensor supported (such as pushbutton switches and potentiometers). Routing around these components sometimes called for small traces. I began drawing larger traces in circuits. I also found that there were advantages to drawing the circuits in the same OpenOffice® Draw program I used to draw the Patch form factor (shown in Figure 12). I was able to modify the form factor and the electronics simultaneously and superimpose one on the other. I believed that making designs available in a program youth were already using would ultimately lower the barrier to youth making modifications to Scratch Patches.

The current batch of Patches has extremely large circuit traces for three reasons:

(1) Removing large sections of unnecessary copper between small traces (weeding) was daunting;
(2) Affixing the Patch circuit was easier when the circuit was shape-aligned with the edges of the Scratch Patch shape;
(3) More copper surface area had more adhesive to hold itself down with.

Problem - connector tabs shifting over time: The card edge connection points on Patch tabs occasionally shifted after multiple uses.

Considerations and current implementation: I made the data, power, and ground traces larger so that more surface area could better adhere to the epoxy layer. To mitigate the problem of the epoxy layer shifting, I extended the part of the circuit that folded around the horizontal tabs and back over the vertical tabs in such a way as to leave enough adhesive exposed to secure the last fold in place. Additionally, a protective layer of acrylic was added to each type of Scratch Patch (sensor, communicator, and extender) to clamp down elements of the surface that may have otherwise shifted or peeled. The second layer also:

- secured the circuit layer in place
- protected the electronic components
- made a socket in which sensors could be secured
- gave builders a surface to decorate
- made Patches more aesthetically pleasing
- afforded stacking of additional layers.

Sensor Representation in Scratch Software

The representation of sensor values in the Scratch environment initially did not make sense to learners. Early versions of Scratch reported sensor values as they were received from the sensor boards: in a range from 0 to 1024 (which are the values a 10-bit ADC reports). Later versions of Scratch normalized the sensor values to be reported as numbers from 0 to 100. Learners were more accustomed to such representations. For example, a learner may have explained to a peer that a reading of 80 from a light sensor could mean that an area in the room was 80% lit. The Scratch environment also had an on-screen sensor board that showed the values of all eight sensors it could read. The number of sensors in this on-screen sensor board was expanded from eight to sixteen.

4.5 Overview of Current Scratch Patches

Currently there are three types of Scratch Patches, which can combine to form highly customizable input devices with up to 16 sensors.

A communicator Patch (C-Patch) sends to Scratch programs real-time sensor information it receives from Patches attached to it. C-Patches connect to computer serial ports, power supplies, and other Patches.

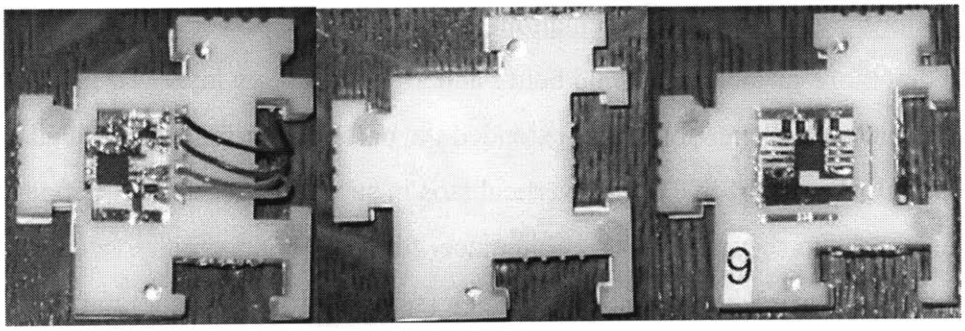


Figure 15: Current Scratch Patches: a communicator, an extender, and a sensor patch

A sensor Patch (S-Patch) can hold several types of sensors that measure one aspect of the physical world (such as light) and translate the measurements for Scratch. Each S-Patch can hold up to 2 sensors in three different ways: pressed into a "sensor-socket," soldered to copper contacts, or held in a slot for wires. Special sensor-inserts were created for the sensor socket.

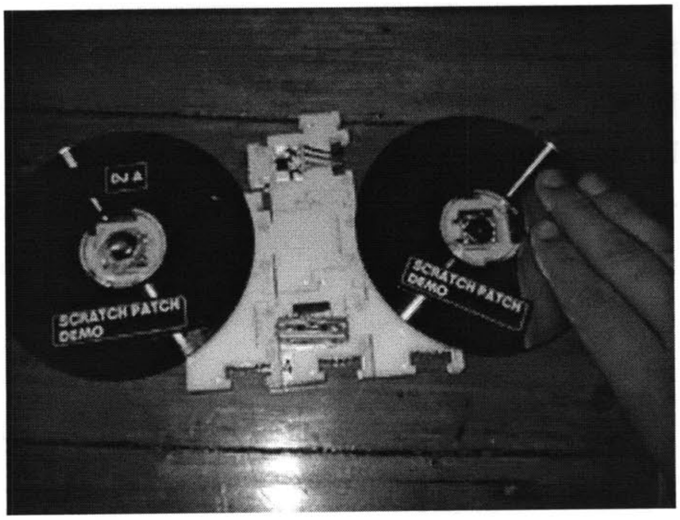


Figure 16: Sample Scratch Patch Two-turntable Hook-up

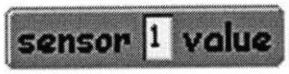
An extender Patch (E-Patch) relays power and data to other Patches. E-Patches change the size and shape of a Scratch Patch input device. To further change the look and feel of Scratch Patch input devices, each Patch surface has places where one can secure craft materials or Patch layers.

Current Scratch Patches (shown in Figures 15 and 16) reflect a series of redesigns intended to overcome the limitations of the standard Hook-up board. A description of the new features that were prompted by each limitation follows.

Increased the number of sensor plugs from eight to sixteen

The technical infrastructure of Scratch Patches allows for 16 sensors to be connected through a computer port at once. The limit of 16 is dictated by the current sensor board communication protocol set in the Scratch programming environment. The Scratch Patch architecture can currently handle more than 16 low-power sensors; further development of new sensor communication protocols would only be necessary if appealing Hook-ups activities required a large number of sensors.

Labeled each sensor plug

Scratch Patches are labeled 1 through 16. When using a sensor in Scratch programs, learners place the  command blocks into programs and designate the number inside the command block to be the one on the Scratch Patch.

Created space for easy sensor swapping

Sensor connectors developed for the current Scratch Patch design (presented below) overcome some of the limitations of standard Hook-up boards connectors. For example, the size of the Patches creates space between sensor sockets when multiple Patches are connected together. Connecting sensors using Scratch Patches can lessen the number of messy wires one needs to connect sensors to computers, compared to how many one would use with traditional PCB boards. Additionally, builders can use as few or as many Patches as they need.

Designed an infrastructure for future active sensors and actuation

The current Scratch Patch design is capable of powering active sensors. Enabling capacitive sensing is one of the features that future Hook-ups research is expected to explore.

Protected the circuitry and added space to decorate

In the Scratch Patch's current design, two layers of acrylic are used to encase the electronic components. The top layer's surface can be decorated on the area that is not used to hold sensors. Additional top layers can be added and decorated or customized to hold special sensors in place.

Personally fabricated the circuits

Using personal fabrication tools in the Fab Lab to prototype and redesign Scratch Patches enables quick testing of additional features.

4.6 Future Directions for Scratch Patches

The next directions for Scratch Patch development involve the implementation of a new (inter-Patch) communication protocol, exploration of actuators, and integration of ideas from youth.

Implement Internet 0 over DC power traces

By changing the inter-Patch communication protocol to that of Internet 0 (Gershenfeld, Krikorian, & Cohen, 2004), Patches can send native IP packets of information to each other without requiring a shared data line. Modulating the power line that runs through the network of Patches would transmit information. This change would improve the connections between Patches by only requiring two leads to make contact instead of three. By using IP packets, these new Patches could interface with the current Internet (perhaps through a wireless connection making new activities possible).

Add the Ability to Actuate

By adding actuation to Patches, features like force-feedback could be explored. (See chapter 8 for a case where modifying the force-feedback mechanisms in an input was an engaging activity.)

Integrate Ideas from Youth

Youth will be encouraged to explore extending Scratch Patches in any way they see fit. If youth wish to make room-sized interfaces, then perhaps larger versions of Scratch Patches would be appropriate for the task.

5. AN OVERVIEW OF HOOK-UPS YOUTH CREATED

Youth aged 10-17 built the Hook-ups characterized in the sections below. Their Hook-ups ranged from magnet-carrying mice to digital duck hunting devices. The first section describes each Hook-up. The second categorizes the Hook-ups according to properties they share and discusses how the design processes youth selected influenced the artifacts' characteristics.

5.1 Descriptions of Selected Hook-ups

This section organizes the descriptions of Hook-ups that youth built according to the materials used to create them. The descriptions in this section are listed for three types of Hook-ups: basic, repurposed, and fabricated Hook-ups.

Basic Hook-ups

Basic Hook-ups are typically everyday objects with sensors attached to them by tape or glue. Pre-made sensors (e.g., LEGO light sensors) are usually connected to user-friendly programming environments that communicate with pre-made computer interface boards. The idea to support "basic" Hook-ups is motivated by a lack of high-end technology access in underserved communities around the world. Tools, activities, and materials for "basics" are geared towards learners crafting housings for pre-built sensors from everyday materials.

Repurposed Hook-ups

Repurposed Hook-ups are electronic objects that were previously used to control something already (e.g., a remote control car controller). A Repurposed Hook-up results from learners discovering how to redirect existing controls of a product to a programming environment.

Fabricated Hook-ups

A fabricated Hook-up is an input device for which many of the parts were designed and built by the learner. The availability of personal fabrication tools such as laser cutters, vinyl cutters, and desktop mills are integral to their creation (Mikhak et al., 2002). The tools are expected to soon be as ubiquitous in learning settings as laser printers are currently (Gershenfeld, 2005). Fabrication tools, activities, and materials will give learners the ability to design objects such as sensors, circuits, and component casings using a variety of stock materials.

5.1.1 Descriptions of Youth-built Basic Hook-ups

CLOTHESPIN SLINGSHOT

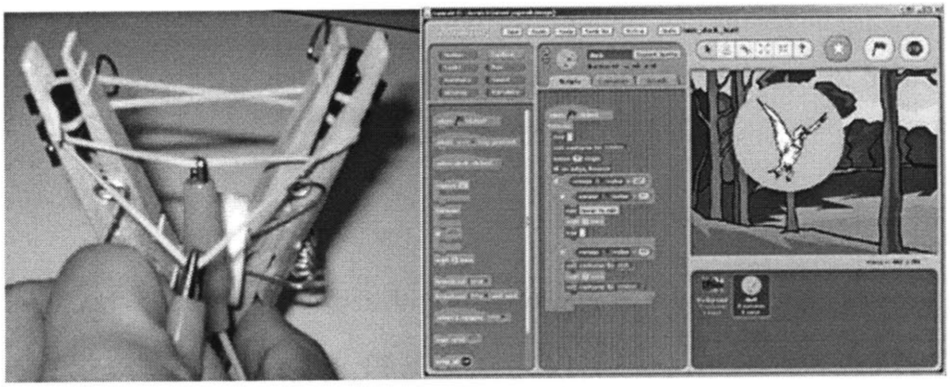


Figure 17: A Clothespin Slingshot and its duck hunting Scratch program

Desired interaction:
A thirteen-year-old (and a group of helpful peers) wanted to use a slingshot to feed or startle a duck flying around the screen.

How the Hook-up works:
Three reconfigured clothespins and a bottle cap form the base of a makeshift slingshot. A rubber band stretches across the top of the base. A pair of alligator clips is clamped to the front and rear side of the rubber band. When the rear side of the rubber band is pulled back and released, the exposed parts of the alligator clips contact each other and complete a circuit. A script on the computer recognizes when the circuit is complete and propels a virtual object across the screen towards the virtual duck.

DROPKICK DUCK

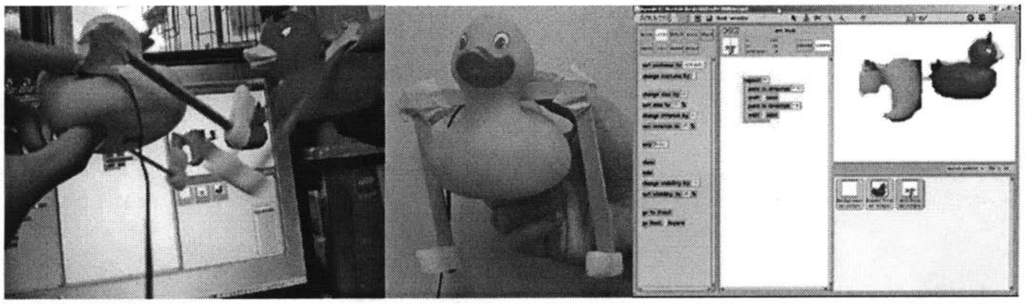


Figure 18: Dropkick Ducks controlling a Scratch animation

Desired interaction:
A thirteen-year-old wanted to make two real-world rubber ducks control rubber-duck-wrestling action on the screen.

How the Hook-up works:
A pair of duck legs are represented as a three Popsicle® stick structure (composed of two parallel sticks connected to each other by a stick taped across the top) that rests on the rubber ducks back. The lower sticks dangle over the sides and have Styrofoam® peanuts at the bottom serving as feet. An exposed wire wrapped around the middle of the top stick acts as one half of a switch. The other half is a wire connected to a piece of foil wrapped around the duck's tail. Tilting the stick structure towards the duck's tail swings the fake feet forward on the physical duck and causes an on-screen duck to swing its feet too. The swing action causes the wire on the stick structure to contact the foil on the ducks tail and close a switch. The computer program responds to the closing of a switch with a script that rotates the on-screen duck 90-degrees to make a kicking motion. If another on-screen duck is near the kicking one, it will appear as though it is getting kicked.

FAKE FISHING HOLE

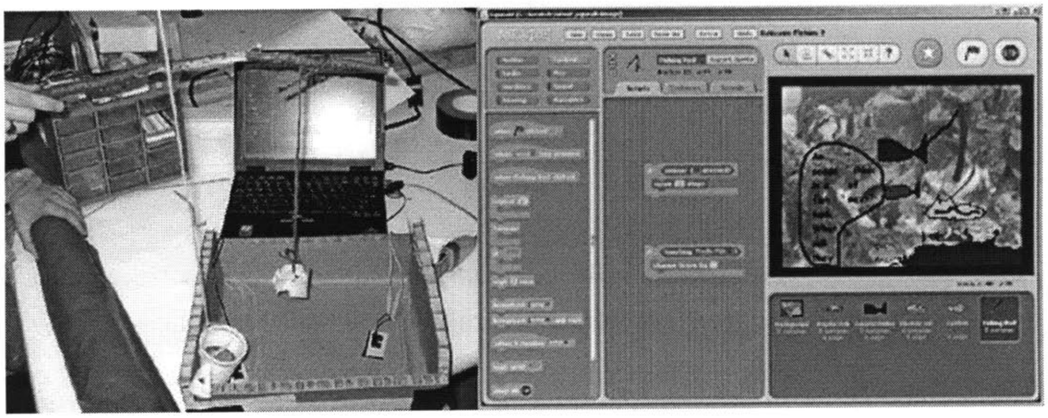


Figure 19: A Fake Fishing Hole (in progress) and its virtual fishing Scratch program

Desired interaction:
A thirteen- and eleven-year old wanted the ability to catch fish in a virtual fish tank with physical fishing poles.

How the Hook-up works:
Several magnetic sensors spread out across (or underneath) a surface constitute a Fake Fishing Hole. Placing a magnet near any of these sensors will complete its circuit. Fishing poles are anything that can dangle a magnet over the Fake Fishing Hole surface. Computer scripts wait for information from the magnetic sensors and cause virtual fish (that normally swim across the screen) to behave differently. The magnetic sensors of a Fake Fishing Hole can be moved around to represent fish in different parts of the hole. When sensors are underneath thin material like construction paper, the areas where magnets are can be decorated to look like fish or other sea life.

FAKE FISHING HOLE: POLES

Two small tree branches have string tied to their tips. Plastic cups are tied to the bottom of each piece of string. Underneath these cups, magnets are held in place with small lumps of clay. These poles catch fish by lowering the magnet-wielding cups over Fake Fishing Holes.

GRAVITY FIGHTING BALL BALANCER

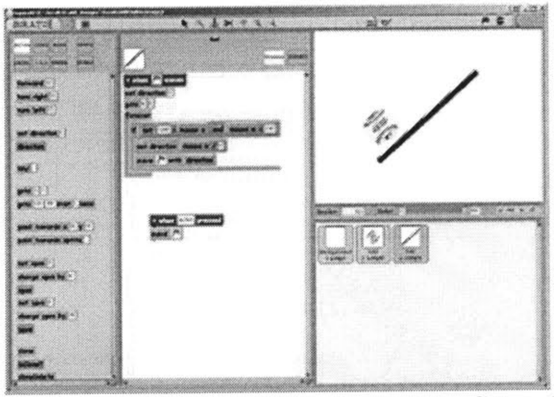


Figure 20: The Gravity Fighting Ball Balancer Scratch program

Desired interaction:
A fifteen-year-old wanted a physical device that could be manipulated to control the angle a virtual surface would tilt. A ball on the virtual surface would slide towards the edge of the surface faster as the surface angle increases.

How the Hook-up works*:
The physical component of the Gravity Fighting Ball Balancer was not finished by the time the builder completed a weeklong Scratch workshop. Had it been finished, it ould have worked by pushing an end of a ruler that was balancing on a plastic cup up or down. Had the young man finished this Hook-up, a sliding potentiometer would have changed according to the tilt and reported the value associated with the position of the slider in the potentiometer. This value would have caused a script on the computer to change the angle a surface was tilting on screen. This changing angle would cause the ball on the surface to fall faster or slower or come to a stop when balanced.

MAGNET-CARRYING MICE

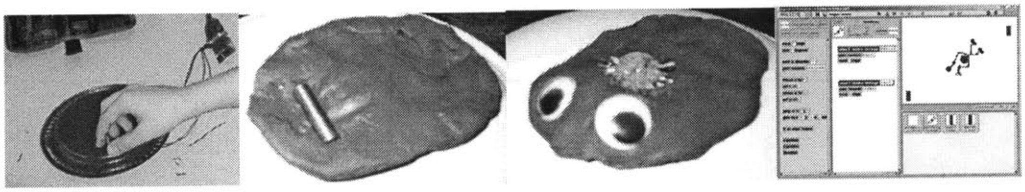


Figure 21: Magnet-carrying Mice's: sensor-plate, bottom, top, and Scratch program

Desired interaction:
A thirteen-year-old wanted to drag a virtual man across the screen using a magnet. The magnet would be like a mouse because it could move objects on the screen like computer mice can.

How the Hook-up works:
A magnet embedded in the belly of a clay mouse (decorated to look like a living mouse with eyes) is dragged over a plastic plate. This plate has four magnetic reed switches positioned underneath a in a manner that they're far enough apart to prevent two sensors from being tripped by one magnet. When a magnet causes a switch to close its circuit, a computer script makes an on-screen character move towards the corner of the screen that corresponds to the activated switch. Multiple mice can be introduced to attract the on-screen characters to multiple corners at once. Scratch Patches and magnetic-sensor-inserts can also be used.

SCISSOR SWITCH

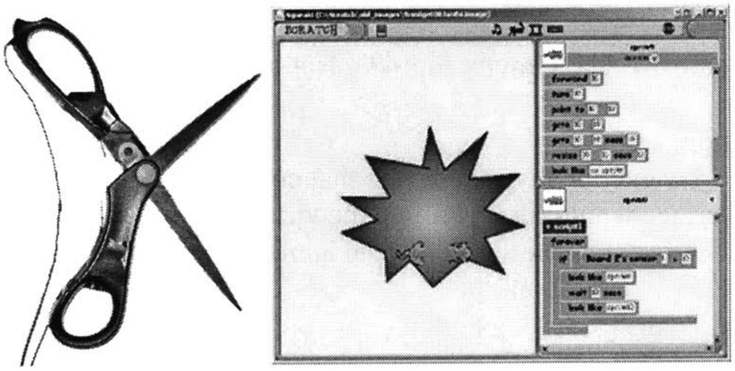


Figure 22: A Scissor Switch and an exploding cat Scratch program

Desired interaction:
A twelve-year-old wanted to invent a sneaky way of detonating on-screen explosions.

How the Hook-up works:
A general-purpose switch is made from two wires connected to a sensor board on one end and to copper pads near the base of scissor blades on the other. Closing the scissors makes the copper pads placed near the base of each blade come into contact with each other and complete a circuit. A computer program has a script that waits for this circuit to be completed. When the script receives notification of the sensor finding a completed circuit, it runs a script that makes a sprite appear to explode.

SPINDLE DRUM

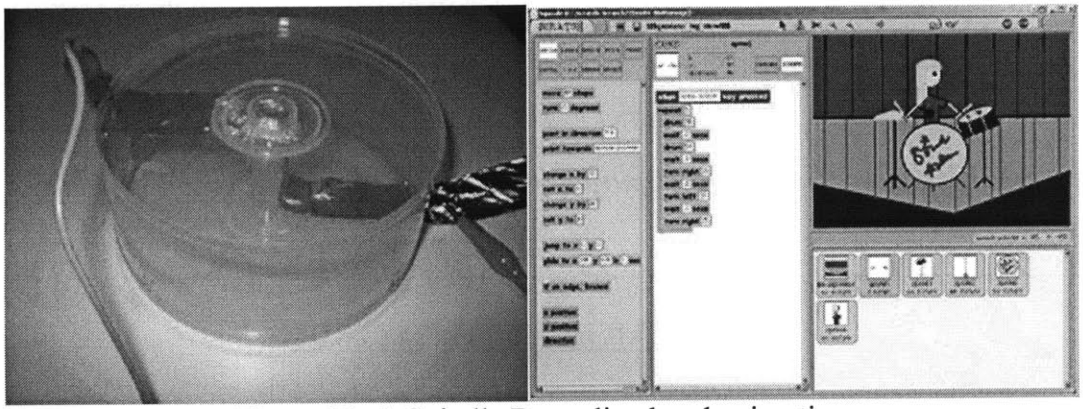


Figure 23: A Spindle Drum live band animation

Desired interaction:
A twelve-year-old wanted to enhance an interactive band project he wrote by letting a user hit a real drum to start animated drumming sequences.

How the Hook-up works:
The base and cover of a 50-CD spindle case are the housing for a simple switch that closes when a user beats on it (once). A wire runs up the pole in base and connects to a ball of aluminum foil taped to the top. Another wire and aluminum foil pad are on the inside of the CD spindle cover in a way that would causes the two foil contacts to meet when a user beats the drum. A computer script realizes when this switch is closed and causes an on-screen drummer to animate his arms and play a tune.

UFO PLATE

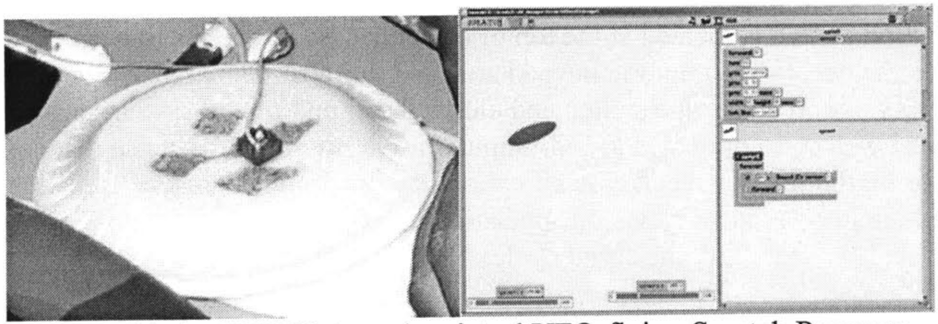


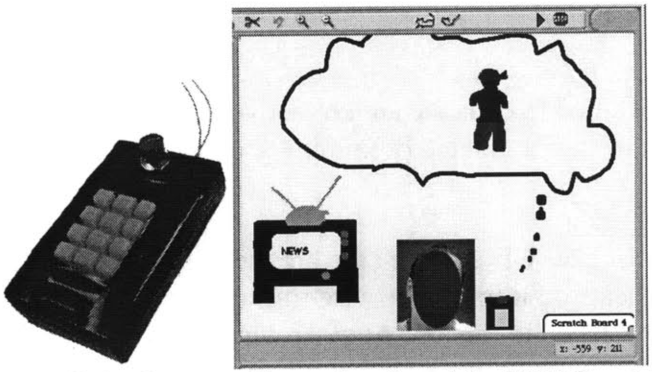
Figure 24: An UFO Plate and a virtual UFO-flying Scratch Program

Desired interaction:
A ten-year-old wanted to use a physical UFO model to make a virtual UFO race around the screen.

How the Hook-up works:
Two small Styrofoam plates stapled together form the body of a convex UFO with a pushbutton switch on its roof. The button connects to a computer program through a sensor board. Pushing this button start computer scripts that make an on-screen replica of the UFO Plate to travel along a path on the screen, fire an abduction ray or lasers, or change its lighting patterns.

5.1.2 Description of Youth-built Repurposed Hook-ups

CELLULAR REMOTE CONTROL



[redacted]

Figure 25: A Cellular Remote Control and a virtual TV Scratch program

Desired interaction:
A twelve-year-old wanted to make a remote control that had buttons to change channels on a virtual TV.

How the Hook-up works:
A pushbutton switch is located at the top of a plastic case that looks like a remote control (thanks to the decorative numeric keypad attached to it that belonged to a toy cell phone). Pushing this button closes the switch and closes the circuit on one of the sensor plugs on the sensor board it is connected to. A computer program running a script that constantly checks for button presses receives notification of a press and changes the channel being displayed on an on-screen television. Because there is only one channel-up button, when the last channel is reached, the script cycles back to the first channel.

DUCK HUNT ZAPPER

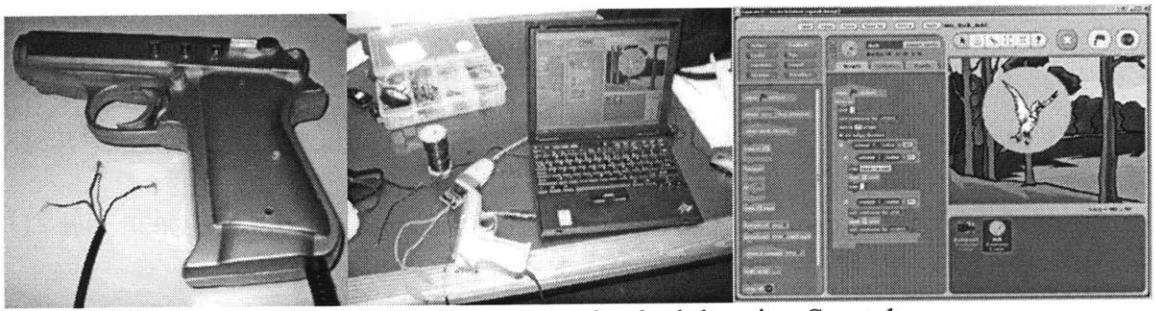


Figure 26: A Duck Hunt Zapper and a duck hunting Scratch program

Desired interaction:
A group of from thirteen- to sixteen-year-olds wanted to make their own version of the 1980's classic Duck Hunt Nintendo® game using a real Nintendo® light gun connected to a computer.

How the Hook-up works:
An aftermarket Nintendo® light gun has wires from its original plug re-routed to a computer through a sensor board. These wires correspond to a trigger button and a light sensor inside the toy gun's barrel. A program running on the computer constantly reads information from the gun's two sensors. When the trigger is pulled, the script responds by looking at the light sensor's latest reading to determine if the gun was pointed towards a bright light source or not. The computer program features a dark forest background image upon which a virtual duck flies. The duck, painted white with a bright yellow halo around it, is the brightest object on the screen. If the reading from the light sensor suggests that the toy gun is pointed at the bright light source, it assumes that the light source is the on-screen duck and shows a costume of the wounded duck.

CUSTOMIZED CONSOLE CONTROLLER

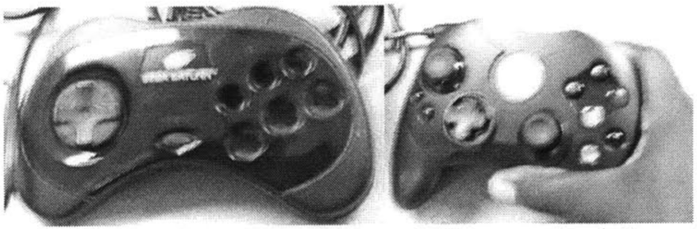


Figure 27: An old gamepad used for parts and a Customized Console Controller

Desired interaction:
A seventeen-year-old wanted to modify the buttons and directional pads of his current gaming system's controller to a set up that was familiar and comfortable to him.

How the Hook-up works:
A modified Xbox® S-controller has a custom-cut directional pad and features three buttons from a Sega Genesis® gamepad in the place of standard Xbox® buttons. This controller plays all Xbox® games.

*This is one of the Hook-ups that does not control Scratch programs.

PUPPETMASTER GLOVE

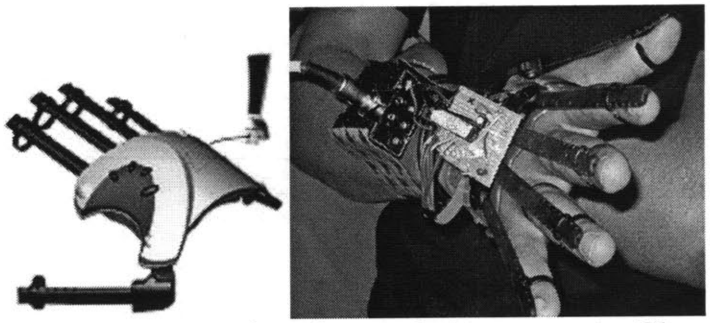


Figure 28: An unmodified P5 gaming glove and a Puppetmaster Glove made from it

Desired interaction:
Two adults (a peer and I) wanted to make an electronic glove that animated an on-screen character's mouth by open and closing a hand (per the request of a teenager)

How the Hook-up works:

A modified Essential® P5 gaming glove shell holds the custom LOGOchip-based circuit that sends different MIDI signals to a computer program for each finger a user bends. Computers running a program called Arkaos® VJ triggers Macromedia® Flash animations that are mapped to the MIDI key the glove sends.

*This Hook-up was designed mostly by adults. Some youth helped in the actually building of two of these gloves.
*This is one of the Hook-ups that does not control Scratch programs

5.1.3 Descriptions of Youth-built Fabricated Hook-ups

CROSSBOW

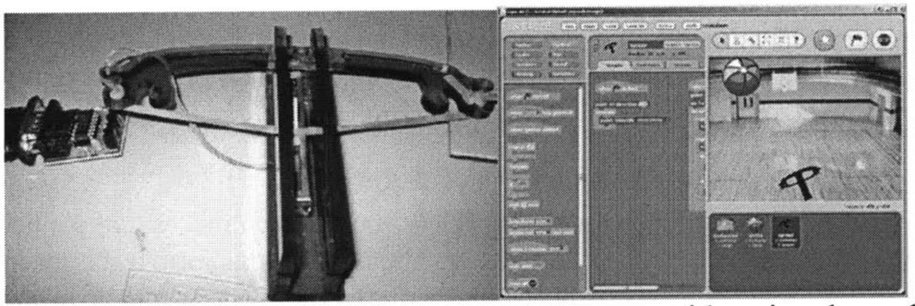


Figure 29: A crossbow Hook-up and a Scratch program with a virtual crossbow

Desired interaction:
An eleven-year-old started out wanting to make a slingshot but was drawn to an image of a crossbow and endeavored to make one instead.

How the Hook-up works:
Four laser-cut acrylic shapes form a press-fit crossbow structure. The base and side shapes hold a sliding potentiometer in place. An arched bow shape lays horizontally across the top of the structure. A rubber band is tied across the rear of an bow and attached to the slider on the potentiometer. Pulling the rubber band backwards moves the potentiometer backwards and releasing the rubber band springs the potentiometer forward. The slider's position in the potentiometer is constantly being reported to a computer program. A script determines when the sensor value received means that the rubber band is being stretched and causes an on-screen crossbow to appear stretched as well. When the program senses a released rubber band, it propels a virtual projectile across the screen at a moving target.

FIST EXTENDER

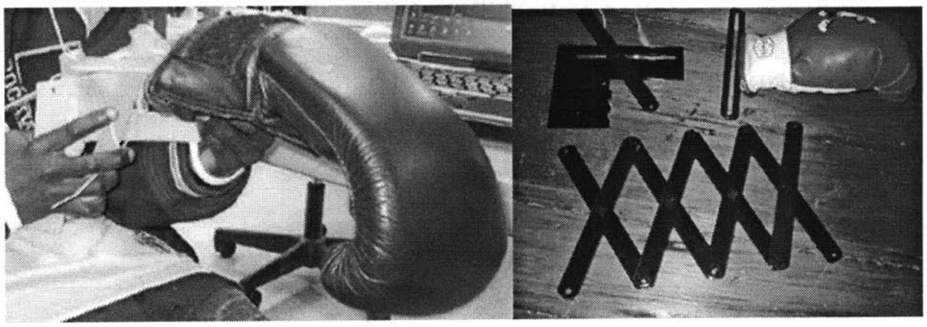


Figure 30: An early Fist Extender prototype and parts of a prototype redesign

Desired interaction:
An eleven-year-old wanted to make a way to extend one hand with the help of a scissor link extender mechanism. He would use the extension to (play) box in the real-world and extend a virtual boxer's reach in a boxing simulation.

How the Hook-up works:
A Fist Extender consists of a chain of Popsicle® stick shaped pieces that connect together through holes in their top, bottom, and center. The connections form a scissor link mechanism. This mechanism rests horizontally on top of a vertical handle. The handle secures one trigger lever that has one end between two links in the scissor chain. The lever is connected to a joint in the link chain in a way as to extend and retract the scissor link mechanism when it is pulled (the pull has to overcome the tension of a rubber band keeping the link chain compressed). A lightweight (6oz) boxing glove is attached to the end of the scissor link chain that extends and retracts. The boxing glove has a strong magnet near the knuckle area. When a player extends the glove near a magnetic sensor, a computer script makes a glove in a boxing simulation travel across the ring, farther than a boxer's natural arm would.

How an alternate implementation works:
If a pushbutton switch were near the end of the travel path of a Fist Extender's lever, a script would run if the lever makes contact with this switch.

How another alternate implementation works:
If a sliding potentiometer were connected to a Fist Extender's lever, the potentiometer would send different values to a computer program as the lever is pulled back and forth. A program would map the distance the slider moved to the distance of an on-screen extended punch.

How yet another alternate implementation works:

Two Scratch Patches with bendy sensors secured in their sensor-slots could serve as a way to feel a punch from the Fist Extender (or a regular fist). A punch could bend the sensors placed on the extreme ends of an arrangement of Scratch Patches (that are separated with extender Patches to match the shoulder-width of the player). These bends change the values the sensors report to a computer program. A change in the left sensor can start a left-handed punch script - the right can do the opposite. Additional punches, such as head-shots and body blows, can be sensed if a second row of Patches were to be added. If the extreme left and right Patches of the new row are stacked with additional SP layers as to hold bendy sensors higher than those in the front row, scripts for head punches could be mapped to them while body punch scripts are mapped to lower sensors.

HARD-TO-CATCH HEART

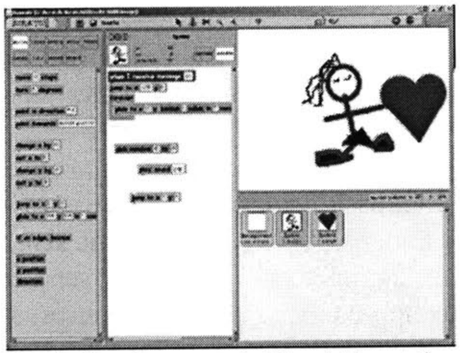
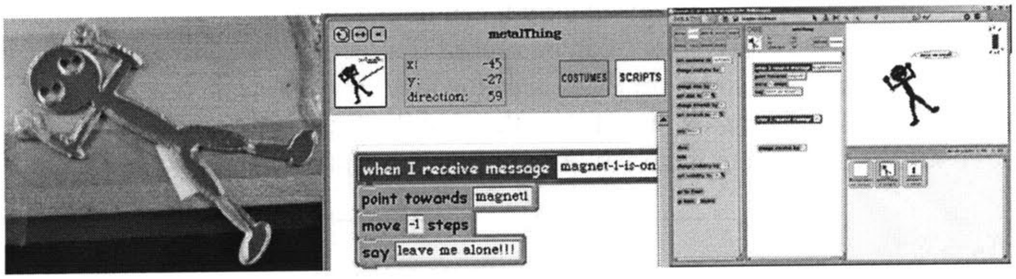


Figure 31: A Hard-to-catch Heart Scratch program

Desired interaction:
A thirteen-year-old wanted a physical object that could say, "I love you."

How the Hook-up works:
An animated heart chases a virtual girl around an ice-skating rink. The path that the heart slides along is influenced by the position of a sliding potentiometer with an acrylic hear affixed to it. When the two characters meet on the screen, a recording of the builder's voice says, "I love you!"

MAGNETIC STICKMAN PUPPET



```
when I receive message magnet-1-is-on
point towards magnet1
move -1 steps
say leave me alone!!!
```

Figure 32: A Magnetic Stickman Hook-up and its Scratch scripts and program

Desired interaction:
A thirteen-year-old wanted to make a virtual magnet attract and repel a virtual copy of his stick-figure puppet printout like a physical magnet does.

How the Hook-up works:
A laser-cut stickman figure that has magnets for hands is moved around a thin cardboard stage with magnetic sensors underneath. These magnetic sensors communicate to a computer program and invoke scripts to run when they send certain values. When the magnetic-stick-figure-puppet approaches a magnetic sensor, a computer script makes an on-screen representation of the stickman move away from an on-screen magnet while saying, "leave me along."

5.2 Comparing, Contrasting, and Categorizing Hook-ups: Conditions for Creativity

This section organizes the Hook-ups described in the previous section according to shared properties and discusses how the design processes youth selected influenced the artifacts' characteristics.

Figure 33 (below) shows that, given the same materials, builders created very different Hook-ups. For example, learners built basic Hook-ups based on items from scissors to rubber ducks. However, because all builders did not have access to all materials and tools, their choice of Hook-up was restricted. For example, everyday materials were plentiful in both research sites, whereas personal fabrication tools were only available at the Commu-Tech Center. Additionally, everyone was encouraged to bring electronic items from home, but only some builders brought them or had access to them. This availability, or lack thereof, influenced their designs. (I provided the P5 gaming glove that became the Puppetmaster Glove. I also sacrificed my aftermarket Nintendo® Zapper light gun for the Duck Hunt Zapper project.)

Made from everyday material	***Made from existing electronics***	***Made from fabricated material***
Dropkick Duck Fake Fishing Hole Gravity Fighting Ball-Balancer Magnet-carrying Mice Scissor Switch UFO Plate	Cellular Remote Control Customized Console-Controller Duck Hunt Zapper Puppetmaster Glove	Crossbow Fist Extender Hard-to-catch Heart Magnetic Stickman Puppet

Figure 33: Materials chosen by youth while creating a range of Hook-ups

Figure 34 focuses on the amount of autonomy exercised by youth. In some cases the help offered by adults was as minor as listening to young people's ideas or making sure that the learner used the construction tools in a safe manner. In other cases adults were barely involved or did much of the project.

Built mostly by a young person	***Built with minor adult help***	***Built with major adult help***
Dropkick Duck Fake Fishing Hole Magnet-carrying Mice Scissor Switch UFO Plate	Cellular Remote Control Crossbow Customized Console-Controller Duck Hunt Zapper Gravity Fighting Ball-Balancer Magnetic Stickman Puppet Spindle Drum	Fist Extender Hard-to-catch Heart Puppetmaster Glove

Figure 34: The amount of autonomy exercised by youth while designing Hook-ups

Figure 35 shows that the Hook-ups design processes were influenced by whether creators employed design processes in which they worked alone or teamed up with others. In the case of the Duck Hunt Zapper, the Hook-up was based on an interest shared by multiple learners who regularly attended Clayton. Most builders, however, chose to work alone. A factor that contributed to this pattern was that the drop-in nature of the informal learning setting made it difficult to gauge when team members would be available or interested in collaborating when they were. The projects created by more than one learner are discussed further in later chapters. The section on methods (in chapter 6) discusses how I typically worked with a small number of builders at once, enabling me to support learners in pursuing their own interests and ideas. The UFO Plate example, on the other hand, is a unique object that reflects the interest of an individual learner.

Created by one	***Created by many***
Cellular Remote Control	Duck Hunt Zapper
Crossbow	Fake Fishing Hole
Customized Console Controller	Fist Extender
Dropkick Duck	Puppetmaster Glove
Gravity Fighting Ball Balancer	
Hard-to-catch Heart	
Magnet-carrying Mice	
Magnetic Stickman Puppet	
Scissor Switch	
UFO Plate	

Figure 35: Hook-ups builders who worked alone vs. with partners

In Figure 36, the various Hook-ups created are categorized according to how many sessions builders needed to construct them. It appears that Hook-ups were as likely to be constructed over multiple sessions as they were to be completed in one. The Hook-ups that took one session all used only one type of sensor and were less complex than the multi-session projects. Some projects that spanned multiple sessions could have been completed in one, had there not been a queue for necessary support or tools such as the laser-cutter. The popularity of the laser-cutter is presented in an extended case study in a later chapter.

Constructed in one session	***Constructed over several sessions***
Cellular Remote Control	Crossbow
Magnet-carrying Mice	Customized Console Controller
Magnetic Stickman Puppet	Dropkick Duck
Scissor Switch	Duck Hunt Zapper
Spindle Drum	Fake Fishing Hole
UFO Plate	Fist Extender
	Gravity Fighting Ball Balancer
	Hard-to-catch Heart
	Puppetmaster Glove

Figure 36: The number of session Hook-ups builders needed to design Hook-ups

Figure 37 highlights how frequently Hook-ups design processes utilized one sensor or more. Although the designers incorporated multiple sensors into their designs almost as frequently as they used one, the learner did not necessarily deliberately decide upon the number of sensors used. Hook-ups that use one sensor such as the UFO Plate were good starter projects for learners. Two of the projects that used multiple sensors, the Customized Console Controller and the Duck Hunt Zapper, were repurposed electronics that came with multiple sensors. In addition, writing computer scripts for multiple sensors was moderately more difficult than doing so for one, so some learners scaled back their designs.

Used one sensor	***Used multiple sensors***
Crossbow	Customized Console Controller
Cellular Remote Control	Duck Hunt Zapper
Dropkick Duck	Fake Fishing Hole
Gravity Fighting Ball Balancer	Fist Extender
Hard-to-catch Heart	Magnet-carrying Mice
Scissor Switch	Magnetic Stickman Puppet
Spindle Drum	Puppetmaster Glove
UFO Plate	

Figure 37: The number of sensors utilized by the range of Hook-ups youth created

Figure 38 presents Hook-ups that were influenced by physical objects and/or computer programs. The lists suggest that Hook-ups designers slightly favored inspiration from physical objects over that from computer programs. A contributing factor for this slight preference is the ubiquity of physical objects in a learning setting. Physical objects are more visible to a community than local computer programs. Visible computer programs are limited to the number of screens in an environment. But computer programs are more portable to other learning locations than physical items and can therefore easily transmit ideas between geographically separated communities.

Influenced by physical object(s)	***Influenced by computer program(s)***
Cellular Remote Control	Fake Fishing Hole
Crossbow	Fist Extender
Customized Console Controller	Gravity Fighting Ball Balancer
Dropkick Duck	Magnet-carrying Mice
Duck Hunt Zapper	Magnetic Stickman Puppet
Fake Fishing Hole	Spindle Drum
Fist Extender	UFO Plate
Magnetic Stickman Puppet	
Scissor Switch	

Figure 38: The physical and virtual influences on Hook-ups youth created

Figure 39 lists the nature of the projects influenced by Hook-ups that were shown to learners as samples. When learners saw Hook-ups others had made, some were inspired to make their own input device to control a similar program and some were inspired to write new computer programs for the existing input device. The range of Hook-ups inspired more input devices than they did computer programs.

Has influenced multiple other input devices	***Has influenced multiple programs***
Crossbow	Dropkick Duck
Customized Console Controller	Duck Hunt Zapper
Dropkick Duck	Puppetmaster Glove
Duck Hunt Zapper	
Magnet-carrying Mice	
Puppetmaster Glove	

Figure 39: Sample Hook-ups that influenced later Hook-ups

Figure 40 presents two operating spaces for which Hook-ups were made: a hand, and a flat surface. Most of the Hook-ups listed are similar in size, yet some operate better in a hand than they do on a flat surface (and vice versa). More Hook-ups listed were made to fit in a hand than operate on a flat surface. Although some, such as the Fake Fishing Hole, could be expanded to larger sizes, doing so would not be practical for viewing the computer program (all of the Hook-ups were designed to work on a PC with a screen size of around 15").

Fits in a hand	***Need to operate on a flat surface***
Cellular Remote Control	Fake Fishing Hole
Crossbow	Fist Extender
Customized Console Controller	Gravity Fighting Ball Balancer
Dropkick Duck	Magnet-carrying Mice
Duck Hunt Zapper	Magnetic Stickman Puppet
Hard-to-catch Heart	Spindle Drum
Puppetmaster Glove	
Scissor Switch	
UFO Plate	

Figure 40: The operating space Hook-ups builders chose for their devices

Figure 41 shows the number of users Hook-ups builders designed for. Hook-ups builders elected to design most of their products to be used by one person. I suspect that this decision was influenced by the nature of the design environment, where learners were encouraged to design personally meaningful artifacts. Of course, it can be personally meaningful to design for others - in fact, designing for a set of users is a key design skill but the drop-in setting, noted above in connection with collaboration (see comment on Figure 35) probably was also an influence here.

Used by one person at a time	***Used by multiple people at once***
Cellular Remote Control	Fake Fishing Hole
Crossbow	Fist Extender
Customized Console Controller	Magnet-carrying Mice
Dropkick Duck	
Duck Hunt Zapper	
Gravity Fighting Ball Balancer	
Hard-to-catch Heart	
Magnetic Stickman Puppet	
Puppetmaster Glove	
Scissor Switch	
Spindle Drum	
UFO Plate	

Figure 41: The number of simultaneous users Hook-ups creators designed for

6. METHODS

This chapter is intended to give readers an understanding of how I interacted with people building Hook-ups. I interacted with them in a way that supported them in their design efforts and simultaneously gave me insights into how to redesign Hook-ups tools to better support their design experiences in informal settings. In doing so, I did my best not to compromise the relationship between the community and myself. Similarly, I made efforts not to negatively affect any relationships learners had with each other.

Because one of the ideals of scientific research is repeatability, I present issues that may come up for others when working with communities of learners in informal settings. I believe that similar explorations of learning in informal settings would be a great contribution to learning and technology fields; however, I acknowledge that the exact conditions in which this research was conducted cannot be reproduced.

Ethnographic-style research methods and a personal guiding principle influenced the methods I used for observing instances and patterns of learning while young people built Hook-ups. Ethnographic methods have roots in the field of anthropology. These methods are traditionally used to scientifically describe human cultures. A common way researchers attempt to understand a particular culture is by participating in it. These researchers become participant-observers, because they are simultaneously participating in a culture and making observations to advance the scientific fields to which they belong.

This chapter presents observations I made at the research sites as I played several roles as a community member and participant-observer. The sections introduce my involvement at the research sites as a mentor and how I interacted with the communities of learners. In chapter 8, I reflect upon the balancing act mentoring participant-observers perform when they play several roles in an informal learning setting.

6.1 My Involvement at the Research Sites

The first youth-made Hook-up was created in November 2003 in a Washington DC Computer Clubhouse. However, the majority of Hook-ups were created between April 2004 and April 2005 at after-school technology learning centers in the Boston area. See chapter 3 for detailed descriptions of the Clayton Computer Clubhouse and the Commu-Tech center.

I visited the Commu-Tech Center two days a week on average over the past twelve months (4/2004 – 4/2005) and visited Clayton once a week starting in September 2004. I played many roles with these visits. My involvement with Clayton began as a mentor who helped specifically with programming in Scratch. My initial involvement with Commu-Tech was as a learner of the personal fabrication tools, sharing what I learned with youth as I became more proficient. Over several months, I worked time for building with craft materials into my visits to Clayton and introduced computer programming into my repertoire at Commu-Tech. Since the start of 2005, I have spent days at each center focused on Hook-ups work, as well as on mentoring in my original role, and have also served as a general-purpose mentor at times. During each visit in 2005, however, I did have the ability to support Hook-ups activities if a learner were to ask.

6.2 How I Interacted with Youth

I interacted with youth at the research sites in several capacities. The subsections that follow describe, first, the settings in which I introduced Hook-ups activities, second, how I interacted with youth during the times I was working with them on Hook-ups projects, and third, what I thought was most noteworthy during my interactions.

6.2.1 Workshops and Drop-in Sessions

I primarily introduced learners to Hook-ups activities in open-activity drop-in sessions or through workshops.

Workshops

Over the past year, Clayton's Cyber Summit and the Computer Clubhouse's Teen Summit programs featured workshops for Scratch/Hook-ups. In addition, a summer-long workshop was held at Commu-Tech where some members elected to make Hook-ups projects using the personal fabrication devices. None of these workshops specifically focused on Hook-ups, but each presented building input devices for computers as an option for interested learners. Workshops consisted of a group of youth meeting together on a regularly scheduled basis with support from one or more facilitators.

Early sessions were usually used for setting the expectations of the participants. It was also important to set the expectations the participants would have of the technologies they would be using in the workshop. Letting learners know that technologies are still being developed can lessen the time it takes to recover from errors encountered during a workshop session. Early sessions are also useful for getting to know the participants so that facilitators can tailor demonstrations to their needs and interests. At least one session in each workshop included a brief demonstration of a sample Hook-ups project. In workshops, youth could choose to work alone, in pairs, or in groups. Sometimes workshops called for samples to be projected onto a wall so that large crowds could see what the facilitators were demonstrating. A series of workshops usually led to culminating events like presentations.

Open-Activity Drop-in Sessions

During visits to Clayton and Commu-Tech during open-activity hours, learners were not held to specific tasks. I usually presented to them Hook-ups projects that were in progress or completed, to spark their interest. Learners chose if they wanted to continue existing projects or start down their own paths - or could choose to ignore me. The drop-in Hook-ups work typically engaged one or two children at a time but was visible for onlookers. However, a large group of members (at the Bridgetown Clubhouse) usually met regularly during a scheduled time during drop-in hours. These meetings led up to the digital puppet show that featured the Puppetmaster glove.

6.2.2 Observing Interactions with Youth

The mind is a faulty recording device on its own (mine included). Other means of recording give researchers the ability to replay an event for accuracy and other cues they may have missed before. Augmenting memory with recording devices for audio and video is beneficial, but at times can be a distraction. I carry a still camera with me most of the time, but I generally do not carry video cameras. This is because I believe that what learners say while working vs. what they say for a camera afterwards may be different. Sometimes young people "freeze" on camera, but their artifacts don't. Current digital still cameras, like the one I use (Canon® Powershot S200), offer the ability to take small video clips. I usually bring it out after youth have produced an artifact. When capturing pictures of artifacts for youth who complete them, if a person is proud of their work and does not mind being recorded, I can store small video clips of him or her and the artifact on the camera.

When an environment proves conducive for dialogue amongst learners, researchers can get a sense for what learners find important or interesting. While learners explored the Hook-ups system, they often talked to each other. I noted when they spoke about design concepts or showed signs of understanding or struggling. I did not go out of my way to overhear every conversation.

Ethnographic-style research calls for the participant-observer to take detailed accounts of the practices (Emerson, Fretz, & Shaw, 1995); over time, a researcher can amass large amounts of written records and field notes. Taking frequent notes can be difficult during hands-on activities and give the appearance that the researcher is distracted. In my work, I found that digital technology helped make my note taking (and transcribing) less disruptive and arduous. While interacting with youth, when I saw a noteworthy event, I took notes on my Palm®-powered cell phone/PDA (Figure 42). I usually carried it with me and it never seemed to disturb the flow of Hook-ups work when I occasionally drew it from my pocket. (If I had access to a camera/PDA/phone, I could have carried less baggage to the research sites because each device comes with cables and chargers etc.)

Figure 42: Palm®-powered cell phone/PDA used for recording notes

The data I collected produced more information than I could have included in a thesis that is intended to be an overview of notable findings from the research. I drew on my field notes to give the reader a good picture of the factors that led to several youth-created Hook-ups. I found it valuable to have field notes and digital images relating to artifacts because the redundancy comes in handy when images taken are rushed and thus poor-quality (as are some presented in this thesis). Quality images sometimes reminded me to write something in my notes that I would have otherwise omitted.

Each researcher will read their notes subjectively and assess situations from his or her own perspective. When learners talk amongst themselves, researchers have a chance to hear learners' opinions in their own words. At times, I asked the learners if they were having trouble or understanding what they were doing so I could determine if I assessed their situation correctly. I also asked questions when learners were starting projects. I would give them opportunities to place themselves in the role of the designer for whatever project was being discussed. I asked questions like, "How would you begin building a way to control car animations on a computer screen?"

When learners in the setting were aware that they were helping out in Hook-ups research efforts, sometimes they offered suggestions for how to make the technology better. Including young people in the process of designing the technological tools they use is central to the concept of participatory design (Druin, 2002). In cases where technology is developed using the participatory design model, children are considered design partners and sometimes have the opportunity to see redesigns of a technology reflect some of their suggestions.

Data analysis included revisiting field notes, saved files, and acquired artifacts to classify and categorize the Hook-ups that youth built. I also composed several case studies to provide a bigger picture of the context in which some of the Hook-ups were created. Immediately following the case studies (chapter 7) are reflections and analyses (chapter 8) of the learning that took place in the study.

6.2.3 What I Deemed Noteworthy During Observations

In order to discuss how and what youth learned while building Hook-ups, I believed that it would be useful to observe:

- how learners developed their ideas
- the approaches learners adopted to build Hook-ups
- how they spoke about design in the Hook-ups context
- what facilitator strategies learners responded to well
- the range of artifacts created
- the design principles employed by learners
- the challenges encountered by learners.

Discussions among youth involving sharing design tips, taking credit for work, mentoring and complimenting each other, can be revealing. I looked for difficulties learners encountered. I viewed these difficulties not as failures on the part of the facilitator; rather, they were viewed as potential sparks for creative solutions and/or feedback for design revisions. I understand that what I present in my case studies and analysis does not begin to paint the picture of all of the learning that took place. However, the data I collected was sufficient to represent interesting learning experiences that provide fuel for future work.

7. CASE STUDIES

I present extended case studies of some youth who contributed to one or more of the Hook-ups listed in chapter 5. The first section represents a traditional case study by depicting a year of Scratch programming by a Clayton teen, Jack, whose first programming experience was through Hook-ups. The second is a less traditional case study that highlights a group of closely themed projects. The third case is a traditional case study of a set of siblings at Commu-Tech who took different routes to creating Hook-ups. The fourth is a nontraditional case that focuses on how a personal fabrication tool, the laser-cutter, can rapidly produce products that can quickly be converted to Hook-ups.

7.1 Jack's Range of Hook-ups Roles at Clayton

Since we first met in April 2004, Jack, an unassuming 13-year-old member of the Clayton Computer Clubhouse, has played many roles in the creation of several Hook-ups. Working with Hook-ups with Jack has often times paralleled a theatrical play, where I provide the stage to a cast of characters who come up with the script amongst themselves for better or worse. Jack has taken center stage at times and worked in the background at others. The following subsections show the range of roles Jack took in Hook-ups activities over the course of the last year.

Jack's stories bring to light some learning experiences he probably would not have had without Hook-ups. Jack's case study also illustrates some of the difficulties one can encounter while introducing youth to new activities like Hook-ups.

Role 1: lead hardware designer

I introduced Jack to Hook-ups while he was participating in a two-day workshop in which six 12-year-olds were challenged to collectively envision, create, and market their own mini-video-game using an early version of the Scratch programming environment. Each participant led an aspect of the production but Jack quickly lost interest in the task he volunteered for and showed little interest in programming initially. I then presented Jack with a box full of scissors, tape, wire, and old toys and asked if he was interested in creating a controller for the group's game – a game in which the main character interacted with a virtual TV. With a small amount of adult assistance, Jack created a one-button remote to control the TV channels. With a desire to add a script to the game's main program that responded to his home-made remote control, Jack became motivated to program. After a few tries, he understood the workings of a simple channel changing script example (which involved the use of "if-then" conditional programming) the workshop facilitators showed him. He listened to his peers and extended the example code by adding the TV channels they agreed upon. He even faced the challenge of programming the TV to loop back to channel 1 after reaching the highest channel but that required some additional guidance from the facilitators.

Role 2: Excited yet unsupported learner

While learning to program Scratch at the April 2004 Scratch workshop in which Jack built his first Hook-up, Jack explored the example Scratch projects we provided on the workshop computers. He was inspired by an example he encountered that was based on the classic arcade game Frogger. He expressed to everyone around him that he wanted to create his own version of Frogger and he could even envision a controller for it. With the workshop facilitators busy working with other members, he began to try to emulate the Frogger example himself by opening another version of Scratch and using the built-in paint editor to draw all of the components of the game. Although he showed great initiative by starting up a program on his own, he did not fully understand how each object in the game that moves needs to be drawn as its own object that can store unique scripts for its behaviors. Jack drew each object in the game in one big picture that all moved together – he did not have enough time to correct this problem in the workshop, but agreed to work on Frogger with me during regular Clubhouse hours later. (Jack and I worked on several Hook-ups projects later, Frogger was not one of them.)

Role 3: Guru today, gone tomorrow

In October 2004, a team of Scratch researchers began mentoring Scratch projects once a week at Clayton. The Clubhouse was open to seven through 12-year-olds with a few special exceptions. Jack, who turned 13 prior to the first fall visit, was invited and attended the second week (after missing the first due to homework). After brainstorming the first week, we let the participants use Scratch on their own during the second. Aware that there were a few participants who had seen Scratch before, I asked the group if someone would like to explain what Scratch is. Jack gave a pretty good description of what Scratch could do, mentioning "it lets you make objects do things," and he even referenced projects that he had seen as examples from the Scratch workshop (including the Hook-ups steering wheel driving game) he attended six months ago that April. His description conveyed one of the most important things he learned about Scratch (and other programming languages): that you program objects. He remembered not to draw multiple objects while editing the way one object appears. That was one of the last times that particular group got to learn from Jack's mistakes as he was unwilling or unable to come for the rest of the sessions as he frequently cited homework as the reason.

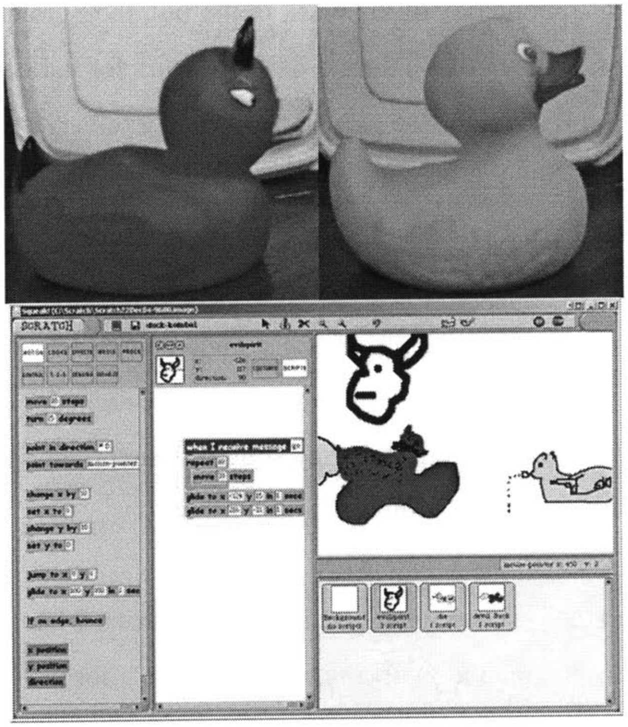
Role 4: Devious designer

After putting emphasis on one day of the week in the fall, the Scratch researchers split up and visited Clayton on different days during drop-in hours for various age groups starting in February 2005. On the days I chose to visit, teens had exclusive use of the Clubhouse but could also choose to be in other parts of the organization such as the gym, homework room, or recreation area. Jack would often be taking the afternoon time to relax before attending homework help sessions held at 6PM. Jack used this time for an assortment of activities including playing Gameboy ®, doing homework, and sometimes misbehaving. One afternoon I was visiting, the Clubhouse coordinator caught Jack roughhousing with another member, Tim, and told them that they should funnel their energy into making something with me instead of wrestling with each other.

I told the young men that the coordinator had good intentions when she brought them to me and urged them not to consider working with me to be a form of punishment. I told them to stay only if they really wanted to work on something that involved programming and hooking stuff up to the computer. They did not walk away. Tim then asked "what kind of stuff [can you hook up to a computer]?" I challenged them to find any object around them and we would see if we could hook it up to the computer somehow.

Jack and Tim rummaged around and eventually turned up a red rubber duck with devil horns and a yellow rubber duck (Figure 43). In order to generate ideas for what kinds of programs we should write for the two rubber ducks to do once they're connected to a computer, I asked Jack and Tim what their favorite things to do were. Tim elected wrestling; Jack favored playing Mortal Combat ®, one of the most violent video games to date.

Tim left the Clubhouse early but Jack remained to make some form of duck combat. Because the version of Scratch he was using had evolved since the last time Jack used it, he needed a little help getting the two images he drew to do anything. Jack wanted his simple game to involve the good (yellow) duck ridding society of the evil (red) duck. He made the game with an interesting ending – by killing the red duck, the good duck would become evil too.



```
when I receive message
repeat
  move steps
glide to x y in secs
glide to x y in secs
```

Figure 43: Two rubber ducks that influenced several Hook-ups and Scratch projects

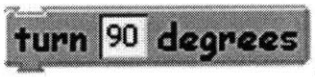
In order to get the effects he wanted, Jack had to learn how to control timing of instructions and how to write scripts that change an object's position in the coordinate system over time. He also had to define an event for when the animation would start – the part of the game that a Hook-up controller would ultimately induce.

Jack decided that the best way to kill an evil duck was with a gun. As he drew a costume for the good duck that depicted it wielding a gun with its wing, he asked if he could hook a gun up to this game like the gun he used to hunt ducks with in the classic Nintendo ® game Duck Hunt. I told him that if he had the old gun that came with Nintendo consoles, he should bring it in and we could work on our own duck hunt (the details of the Hook-up inspired by Duck Hunt are covered later)

7.2 Feet, Food, Firearms, and Flying Fowl

The rubber ducks that Jack and Tim pulled out of a box inspired all the projects described below.

Transforming Sitting Ducks Into Drop-kicking Ducks

In the weeks following Jack's initiation of the Duck Kombat game, Jack forgot to bring in his toy Nintendo gun controller and did not do anything with the Duck Kombat program he wrote in the meantime. Tim, however, initially wanted to make the ducks wrestle so he wasn't waiting for the Nintendo gun to arrive. Tim began his Scratch learning by importing profile photographs of the red and yellow ducks into two sprites. While learning how to make a duck move around the screen, Tim mapped a wrestling idea onto one of the command blocks,  turn 90 degrees. When he saw the duck appearing to lay flat in the air after he clicked on that block, Tim imagined that that duck was performing a drop-kick. He edited the sprite costume to have a leg in order to achieve the full drop-kick effect. Once the on-screen duck had legs and could drop kick, we began to think of ways to trigger the virtual drop-kick using the physical rubber duck.

We found an assortment of everyday items around Clayton to serve as craft materials for our Hook-up duck. We gathered Popsicle sticks, Styrofoam, and like materials. An onlooker suggested that we stick a Popsicle stick through the duck to make a leg that had lever action – you pull on the top part of the stick and the bottom part underneath the duck would make a kicking action. Neither Tim nor I wanted to impale the duck so we continued to brainstorm until Tim came up with a three-stick solution.

To give legs to the duck using Popsicle sticks while preserving the original toy, I challenged Tim to figure out a solution with three sticks (I had come up with a design for duck stick-legs by then but didn't want to ruin Tim's fun). Almost before he could get the third stick, Tim made a pair of "U" shaped legs by laying two sticks on a table parallel to each and taping them to each end of a third tick laying across their tops – then adding Styrofoam® peanuts to the bottoms of the parallel sticks to make feet. He draped them on the duck's back. Before the Clubhouse closed, we connected a wire lead to the middle of the duck's stick-legs and another to the ducks tail. When a person pulled the stick-legs backwards to touch the tail, this completed a switch and triggered a Scratch event such as an on-screen drop-kick.

Flinging Food Into Fowl Mouths… Virtually

Many Hook-ups can be inspired by one Scratch program. A creative person can turn a duck graphic flying around the screen into more than a duck hunting game. Chris, a 13-year-old member who had worked with Hook-ups before, was up to the challenge of thinking of something else to do with a flying duck program. We decided to build a slingshot because it can fling virtual rocks into bushes to flush ducks out of them or can be used to fling virtual worms to the ducks to give them energy. The catch was to build a slingshot Hook-up from the craft materials in Clayton. The context of building a slingshot from scratch was compelling enough for drifting members to get involved without knowing it was ultimately to be used with a computer program. Many members tweaked with slingshots designs and incorporated several craft materials. Ultimately, we made the Hook-ups slingshot sturdy enough to withstand tension from a rubber band by arranging disassembled clothespins and locking the sticks in place with the springs.

Fake Firearm Frenzy

In mid-March, Jack informed me that he finally confirmed his suspicion that his mother had thrown away his Nintendo gun. The next week I brought in a 3rd party Nintendo gun I owned to sacrifice to the game.

Before trying to recreate the whole Duck Hunt system, I needed to know how most video game firearms work so I went to HowStuffWorks (howstuffworks.com). I found an article about how light guns for video games work and left it up on a Clayton computer for reference.

Without the ability to open up the toy gun to inspect its inner workings as a group (due to the lack of a small screwdriver), I settled for experimenting with the gun's wire outputs. We tried hooking every possible combination of two color to the Hook-up sensor board to see if we could stumble upon the controls for the trigger.

The next week, equipped with a small screwdriver, I opened up the gun and placed it on the table and at the same time seemingly opened up Hook-ups activities to many curious onlookers. People that I had interacted with in Clayton for weeks but who had never taken an interest in what I was doing then sat down with me to chat.

Jack, who seemed preoccupied at the time, took notice of the group chatting with me. As the group began tinkering with the toy gun, Jack felt the need to bring his Duck Kombat game to the attention of the group's attention. He looked up from his Gameboy® long enough to utter, "You should see the other game I made!"

That was the extent of Jack's involvement with a game he helped get started. His peers, Steve and Dee contributed the most towards getting a functional duck hunt simulation running similar to the Nintendo version. Steve figured out how to make the data coming in from the gun's light sensor meaningful for the program. In doing so, he experimented with every source of light and light setting in the room – using a laptop LCD screen at one moment, a CRT at another, then trying both with the lights out etc. Dee added a lot to the game play by adding costumes to the duck to represent when it was hit or missed.

7.3 Sean and Don: Siblings Sharing the Commu-Tech Center

Shortly after the Commu-Tech Center added a Fab Lab facility, I met Don as part of the Teach-to-Learn/Learn-to-Teach (T2L-L2T) program. I was volunteering to help Don, and other teens from the community, learn how to use the various facilities in the Center for their own projects and later teach younger peers how to do the same. Don was familiar with most of the Center because he spent a lot of time there working on video projects, drawing, or attempting to make video games. I helped people in Don's program learn to use the Fab Lab to work on projects they found meaningful. (See chapter 2 for the project Don found meaningful.)

While working with Don, I met his younger brother Sean. Sean's initial role in the Fab Lab was that of the little brother: his visits were typically to tell Don a message from their parents. Don and Sean lived in Commu-Tech's neighborhood and I initially saw Sean more frequently outside of the Center (usually on his way to play basketball) than I did in the Fab Lab.

It seemed as though Sean regarded the Center as his older brother's space. Don began spending a lot of time at the Fab Lab because T2L-L2T members worked from 9 a.m. to 3 p.m. Sean rarely saw the Fab Lab used outside of the context of T2L-L2T and did not know that he could use the Fab Lab too. Before Sean discovered that there were times that Fab Lab volunteers would open the facility up for open activities, he did spend some time there. However, the time he spent was not taking advantage of the facility for himself; rather, he would occasionally be called upon to play a role some siblings deem fit for a little brother: that of a security guard. Sometimes Don would leave with his friends momentarily and would ask Sean to ensure that no one moved or stole the parts of his Custom Console Controller project in his brief absence. Don was aware that some people clean up around the lab and did not want smaller parts of his project discarded.

Sean began getting familiar with the Fab Lab environment itself. He began actively participating in the Fab Lab community (beyond his duties as a security guard) when he etched something he drew onto a piece of acrylic. After seeing that his brother's friends had written names that they call each other on Xboxes®, Sean approached me with a graffiti-like drawing and asked if he could etch it in acrylic.

From that point on, Sean found that he could do many things in the Fab Lab (and eventually got his little sister involved as well). He worked on one project, for example, that began as an extension of his body. The Fist Extender was a project that Sean found engaging because it related to a sport he enjoyed: boxing. Some of his brother's friends were recreational boxers. Sean was aware that he should not box with them because he knew that his shorter arms would not reach the friends before they could give him a sound pummeling.

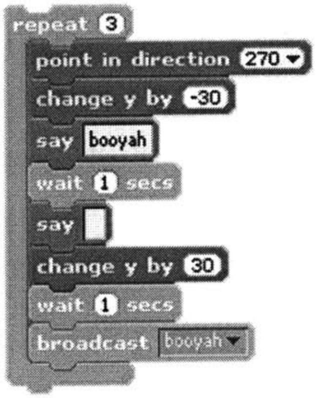
When Sean saw that some of his peers were improving upon a sample Scratch boxing game that I had shown them, he wanted to take part in the game. His stipulation was that he only wanted to be the main (controllable) character. He didn't want to do what his peers were doing: adding their faces to boxing opponents. We made a playable character in the boxing game that had a picture of the back Sean's head as a costume. Sean could then take on his big brother and others in the virtual ring.

Just as in the real world, the boxer in the game could only punch so far. When opponents are on the other side of the ring, the punches do not reach. It was easy to adjust the program to make punches reach across the ring, but then the game lost its boxing feel. Sean and I agreed to unlock the option for extended punches only if we were using a special controller.

The controller we came up with was modeled after a booby-trap featured in the 1980's film "Goonies." In this film, one of the weaker characters was armed with gadgets to escape trouble. One gadget was a boxing glove sprung from his belt. Sean and I could make a similar extending boxing glove by using the laser-cutter to make a chain of scissor-like connections (a scissor linkage structure is shown in Figure 50). The laser-cutter is precise and consistent when many copies of a certain shape are needed. We designed one link, an oval with holes for connecting pins near the top, middle, and bottom of it. We then designed a mechanism to extend and retract the links. Instead of creating our own makeshift boxing glove, we borrowed one from a friend of Sean's brother so we could test our creation.

This project is still evolving; Fist Extender's trigger structure needs to be strong enough to function mechanically with a boxing glove attached to the scissor linkages. We have been exploring ways to effectively sense when a person is punching with the extender. We attached a sliding potentiometer underneath the thin slot on the trigger-base. When the prototype (depicted in chapter 5) is assembled, the trigger is connected to the slider and the top-mounted scissor linkage. Moving the trigger back and forth extends and withdraws both the slider (which is connected to a sensor board) and the linkage (which is connected to a small boxing glove). Some of the options I would like to explore further involve arrangements of Scratch Patches. Sean learned a lot about design and leveraged his spatial learning abilities (to be discussed in chapter 8) in the process of making the Fist Extender.

Don did not think that the boxing game was fair, as it had only one controllable character. Even though the opponents in the game were heads with no arms, Don used Scratch to give them a way to fight back. He gave the opponents the ability to head-butt the armed boxer by writing the script shown in Figure 44 below.



```
repeat 3
    point in direction 270
    change y by -30
    say booyah
    wait 1 secs
    say
    change y by 30
    wait 1 secs
    broadcast booyah
```

Figure 44: Don's head-butt script

In the Custom Console Controller Hook-up Don made, he did not write his own programs to control it (because it worked with commercial Xbox® games). When Don wrote this script, he began to understand the basics of Scratch programming (and some advanced aspects, such as broadcasting). He spent much of his attention on programming the kinds of games he played on the Xbox®, but had never been able to program before. I never suggested that Scratch could make the kinds of games people play on high-end consoles such as the Xbox®, but Don was determined to find ways to incorporate characters from Xbox® games into Scratch projects.

Unlike Don, who spent a lot of his energy after the boxing game on programming, Sean typically approached projects from the physical side. The process Sean used to create a Hook-up with a laser-cutter is described below.

Sean created the Crossbow Hook-up after he laser-cut a simple slingshot from a piece of acrylic. While searching the Web for slingshot designs, Sean came across an image of a crossbow. He then created shapes in the OpenOffice® drawing program that would form a crossbow. The shapes could be cut out with the laser-cutter (Figure 45). One shape was a rectangle with slots into which tabs from other shapes were inserted. This shape was the base of the crossbow. Two shapes formed the sides of the crossbow's barrel. These sides had tabs on their bottoms that slid into slots in the base. They also had tabs on top to secure the arch of the crossbow that held a rubber band. The arch was a semi-circle with two slots removed where the upper tabs of the side fit. A sliding potentiometer rested on the base between the two sides. The arch's rubber band was tied to the slider's handle, so that when the rubber band was pulled back and released, the slider moved with it. Wires connected to the bottom of the potentiometer ran through the base to an interface board that sent changes in the slider's position to a computer program. The computer program displayed a version of the crossbow with two costumes: one with the rubber band relaxed, the other with the rubber band extended.

When a reading from the potentiometer was above a certain value, the on-screen crossbow would change its appearance to look stretched out (like the real-world crossbow) and fling a virtual projectile across the screen. The target on the screen could change costumes and its movement if Sean wanted to make the simulation easier or more difficult.

While building the crossbow Hook-up, Sean learned how to use simple parts to create complex objects in several contexts. He designed the crossbow itself to be made from several simple parts that press-fit together. While working in the CAD software to design the parts, he was forming elegant shapes by merging simple on-screen rectangles and ovals. The simple program to control the virtual program used simple Scratch command blocks to build a moderately complex interaction.

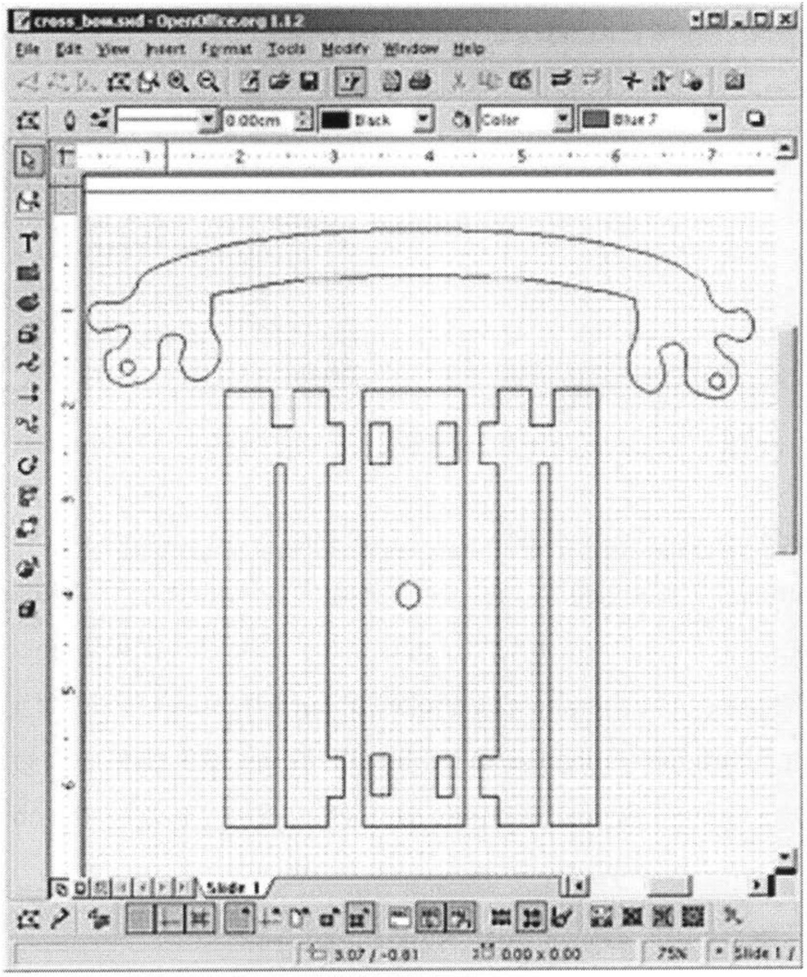

cross_bow.sxd - OpenOffice.org 1.1.2
File Edit View Insert Format Tools Modify Window Help
0.00cm
Black
Color
Blue 7
Slide 1
3.07 / -0.81
0.00 x 0.00
75%
Slide 1 /

Figure 45: The crossbow design

7.4 Laser-cut Personalized Plastic Puppets

The laser-cutter at Commu-Tech was the most popular machine to use. The following case demonstrates how laser-cutting activities could lead to the creation of Hook-ups.

In the process of making people-shaped refrigerator magnets (Figure 46), a group of three 10- to 13-year-old boys at the Commu-Tech Fab Lab were making up stories as each shape emerged from the laser-cutter. Each member of the group used a computer drawing program to make body shapes with holes just big enough for press-fitting tiny magnets (that usually looked like eyes). They would then convert the drawings into a tool path for the laser-cutter and choose a colored acrylic sheet to cut from. When each member of the group saw what a peer created, they immediately began making up stories about them, based on their properties.

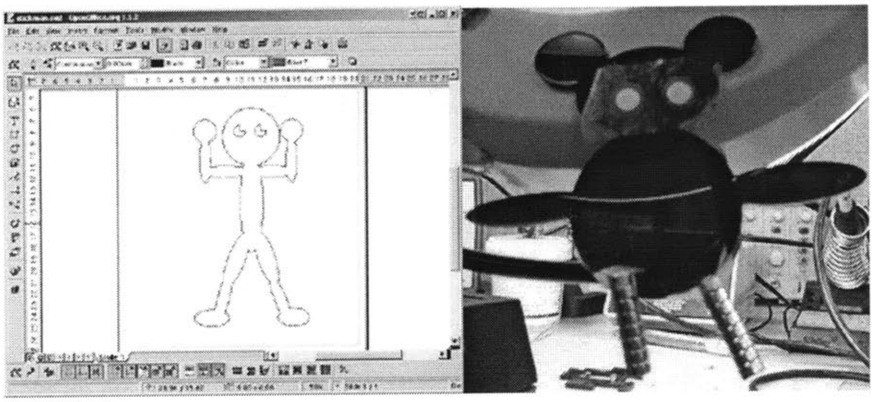


Figure 46: A refrigerator magnet design and broken example (repaired with magnets)

The group made up a story about how one figure they produced that had surface blemishes from being scorched by the laser became "Eczema Bear." Another figure was posed in a way that earned it the title "Breakdancing Bear" for how it looked when spinning on the table. The members also joked about a figure shaped like a man flexing but still unable to produce any muscles. Although the group was doing a great job coming up with stories about the figures, I introduced them to a way to have the objects tell their own stories.

By placing two magnetic sensors connected to a computer on a table, one member of the group was able to write a Scratch program that made a virtual representation of his flexing figure get fatter and skinnier as he walked towards one magnet or the other. To do this, the member learned how to import the file that he had drawn in order to cut the figure out of the plastic. He also had to discover which effects would make the figure skinnier and fatter. The script also called for the on-screen figure to say something as he grew and shrank. For each condition, the text above the figure's head changed accordingly. By moving the figure in his hand around the physical world like a puppet, the designer could enable the on-screen figure to tell a story.

8. ANALYSIS AND REFLECTIONS

In this chapter, I analyze and reflect upon how and what Hook-ups builders learned in the process of making input devices. This analysis is intended to give the reader an idea of the ways some individual learners conceived of and created their Hook-ups. The chapter also examines how learning outcomes were influenced by the informal nature of the learning setting. The sections of the chapter cover:

- how Hook-ups projects got started
- how learners approached projects with different learning styles
- how learners benefited from sample projects
- how learners collaborated
- how learners worked through projects on their own and in groups
- what Hook-ups builders learned
- what obstacles learners faced
- what I believe educators and designers can learn from these experiences.

8.1 How Learners' Individual Styles, Inspirations, and Ideas Influenced Designs

This section examines some of the styles individuals adopted, inspirations for projects, and ideas that prompted Hook-ups activities. The first subsection explores two approaches to design activities. The next examines how ideas come from people, media, and objects around a learner. The next subsection covers how Hook-ups experiences of others can influence learners.

8.1.1 Learning Styles of Bricoleurs and Planners

In principle, two learners could produce nearly identical Hook-ups, but arrive at the final result in dissimilar ways. Let's assume that one learner continually rearranged objects and construction material until an interesting input device emerged; the other made a step-by-step plan for how to make the same input device before even touching any construction material. Which style would be the right way? The answer is, both. There is no single "correct" way to build a Hook-up.

Once a learner has an idea that seems realizable with Hook-ups tools, the learner determines how to navigate to the finished project. Two methods to building Hook-ups described in chapter 5 are similar to approaches to design characterized by Papert and Turkle as those of bricoleurs and planners (Papert & Turkle, 1992). The terms refer to learners' preferred styles of approaching design tasks. The authors positioned the planner approach as the dominant approach taught in traditional computer programming courses. They presented a compelling case for formal educators to appreciate the bricoleur approach that favors thinking with rearrangeable objects over working with abstract formal rules. Educators in informal settings generally have more relaxed structures than formal programming classes and are usually as amenable to the bricoleur approach as to the planner approach.

Much of the work of designing the Hook-ups system was intended to make it adaptable to the learner's preferred approach to physical and virtual design. There were learners who preferred to take bricoleur and planner approaches to every aspect of their design projects (both physical and virtual) and there were individuals who used a bricoleur approach for one aspect of their project and a planner approach for others.

For example, a group of learners came up with a plan for the entire Duck Hunt Zapper project that included the following elements:

- determine when the gun's trigger was pulled
- determine if the gun was pointed at a duck on the screen when the trigger was pulled
- if true, make the duck look like it had been hit, and increase score
- if false, allow the duck to keep flying.

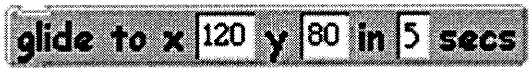
It was not difficult for the group of creators to gather around a common goal in this case. The goal was to rebuild behaviors they were familiar with from playing the original Nintendo® Duck Hunt game. They were not creating anything new, but programming a controller was new to most of them. This project's strong connection to the interests of the creators drove it to completion in two afternoon drop-in sessions.

In contrast, a member given a magnet and magnetic sensors took a bricoleur approach to a simple Hook-up. In creating the Magnet-carrying Mice Hook-up, Kris grabbed magnets and began playing with them before he did any planning. Getting a feel for how close the magnets were to each other before they began to attract and repel one another gave Kris an idea for how far apart the magnetic (reed switch) sensors had to be placed in order to prevent two sensors from being tripped by one magnet. He arranged four magnetic sensors so that a single magnet could not trip more than one simultaneously. He taped the arrangement underneath a plastic plate, plugged the sensor wires into a standard Hook-up board, and modified a sample program to control a character on the screen by moving a magnet over the hidden sensor in the plastic plate. He realized that what he had made worked similarly to a computer mouse and dashed across the room looking for craft material to decorate the magnet with, to give it a (computer) mouse look and feel. He molded a mouse case with clay he found, wedged the magnet into the bottom of the mold, and added eyes to the top to serve as fake mouse buttons and also decorative (biological) mouse eyes.

When Kris picked up the magnet, he had no idea that he would have a magnetic mouse an hour later, let alone a plan. He continued to tinker with magnets and sensors – getting a chuckle out of effects he could create when he used two magnets over his sensor plate (which resulted in the character feverishly attempting to go in two directions at once).

Opportunities for Mixing Styles and Approaches

Learners can prefer one style of working, yet still enjoy other styles and approach projects with a mixture of styles. Engaging in design activities that combine physical and virtual design components increases the opportunities a learner has to vary design styles. Hook-ups activities can cause bricoleurs to unexpectedly discover a part of a project that they get excited about and plan out thoroughly. Similarly, Hook-ups activities are good for prompting planners to adopt a bricoleur style in the course of working with two mediums and the area where they intersect (the sensor interface for physical and virtual mediums) because of likelihood that an unexpected event will warrant unplanned tinkering. For example, Beth originally planned for the Hard-to-catch Heart Hook-up to be a stand-alone heart-shaped device that could record and play back a message (without a computer). When she found out that recording and playing back audio clips in Scratch was easy for her to tinker with, she slowly integrated a computer into her plans for the previously all-physical project.

She created two virtual characters to carry out her original plan of saying "I love you" when one touched the other. She added unplanned features to the animated story as a result of tinkering. Experimenting with the manner in which the characters moved towards each other caused Beth to find that the  glide to x 120 y 80 in 5 secs command caused the characters to move across the screen as though they were ice-skating. She then imagined that the blank white background behind the characters was a skating rink. Because ice-skating is gliding on ice, she then thought of ways to control how the on-screen heart moved around the ice to chase the girl. She mapped a sliding potentiometer (with a physical heart-shaped piece of acrylic mounted on it) to the heart's movements.

A modified version of Beth's original plan became the basis for a Hook-up. The process of having a project develop from idea to Hook-up can go through many steps. When her plan to create an interaction that said, "I love you" was complete, she found inspiration by exploring the Scratch command blocks. She ultimately ended up with a Hook-up that had to do with ice-skating. Beth's project caused her to employ the styles of both the planner and the bricoleur at different stages in the design process. The next subsection delves into other factors that influenced Beth's project and discusses how ideas for projects can be sparked (by people, media, and physical objects).

8.1.2 Learners Finding Inspiration and Ideas

This subsection is intended to shed light onto how learners get ideas to start design projects in informal settings. Beth's Hard-to-catch Heart can illustrate how a multitude of factors can inspire projects to begin or to change directions later.

Some factors that played a part in the inception and trajectory of the Hard-to-catch Heart project were:

- the difficulty involved with the original idea
- me as a facilitator helping smooth the original idea into an attainable alternative
- an approaching Valentine's Day
- an upcoming ice-skating field trip
- a recent discussion of coordinates in a Math class
- influence of Scratch command blocks.

(Almost too) Big Project Ideas

Part of Hook-ups activities entailed learners coming up with ideas. At times, it was difficult to turn learners' ideas into feasible Hook-ups projects. For example, the Hard-to-catch Heart was a case where the initial goal was not feasible. This project was prompted by an idea that would have required the design of a complex circuit so that the designer could record and play back her voice on a physical object. The project became more feasible (for the amount of time the learner had access to the Fab Lab) when she modified her idea at my request. I suggested that she record and play back her voice using Scratch because it supported recording and playing back audio files and would not require creating a complex circuit. She was happy with the recording and the way she was able to make it play when two on-screen characters bumped into each other. Helping learners come up with realizable ideas is further discussed later in this chapter (see 8.5.2 Strategies of Facilitating Technological Design Activities in Informal Settings).

Part of my role as a mentor in the Clayton and Commu-Tech communities was to put forward stimulating ideas for possible projects, while trying to match projects to learners' existing sources of inspiration. Although this idea seemed like one that would be too big for some learners to swallow, the idea of controlling computer programs with homemade input devices served as a strong motivator to some all by itself. Many learners were initially unaware that they could program at all so programming anything beyond a keyboard and mouse to control a computer seemed absurd. Infusing the idea of homemade input devices into the communities expanded the range of possible design activities. Once learners realize that they are capable of completing Hook-ups style design projects they previously did not consider, their imaginations, people, media sources, physical objects, and example applications will influence the context for which they design.

Project Ideas from People

I was not the only person contributing ideas for design activities that called for Hook-ups. Other mentors also put forth ideas for projects based on their passions and interests. One mentor introduced the idea of creating a digital puppet show to a group of Clubhouse members. It did not take long for members to get excited about coming up with their own production. Many Computer Clubhouse members showed interest in taking part in the performance in different capacities. One member suggested that the person that would control a digital puppet on the screen should be able to make hand gestures (like s/he would when making a sock puppet talk) that make the on-screen puppet speak. He proposed an electronic glove to control the digital puppet's mouth. This idea led to emails a Media Lab colleague and I received requesting our help in the technology development of a digital Puppetmaster Glove.

Project Ideas from Media Sources

The many forms of media available to youth today also serve as the source of ideas. Media sources from television shows to video games influenced several of the Hook-ups projects described in the extended case studies:

- professional wrestling influenced the Dropkick Duck
- video games influenced the Duck Hunt Zapper
- TV programming influenced the Cellular Remote Control.

Costumes (images) available within the Scratch environment's media library have also influenced Hook-ups projects. For example, in the Scissor Switch project, the default on-screen programmable object (sprite) that is an orange cat (Scratch's mascot) was unable to avoid the on-screen explosion in that Hook-up.

Project Ideas from Physical Objects

Many computer programming environments such as Macromedia Flash® can give ambitious learners outlets for making creations that draw upon existing virtual media. But one of the distinguishing features of the Hook-ups system is that projects can integrate physical objects that inspired them:

- magnets inspired the Magnet-carrying Mice and Stickman Puppet Hook-ups
- rubber ducks inspired the Duck Hunt Zapper
- scissors inspired the Scissor Switch
- (specialty) video game controllers inspired the Duck Hunt Zapper and the Customized Console Controller Hook-ups.

8.1.3 Introducing Hook-ups Through Samples and Learners' Existing Knowledge

This subsection focuses on how learners with different backgrounds reacted to sample Hook-ups. A subsection relating to learners with little relevant design experience is presented first, followed by one that examines learners with Scratch experience. The last subsection covers learners with experience designing physical constructions.

In the early stages of the project, I created some sample Hook-ups that I thought would help generate ideas for other Hook-ups. Several other Media Lab colleagues contributed to the collection of samples. I initially used these samples with a small number of learners to better understand what aspects of Hook-ups learners would find interesting and/or need help with. Working with early users also generated Hook-ups that could serve as samples for other learners. I sought to have a collection of sample physical forms and computer scripts for Hook-ups that could appeal to a range of learners' interests. Learners were encouraged to borrow from sample projects: they could copy scripts from samples to get started or mimic parts of the physical model. For example, the first youth-built Hook-up created was inspired by a paper plate-based steering wheel. The learner used a pushbutton on a decorated plate in the UFO Plate Hook-up.

Learners' Reactions to Sample Projects and Starting From Scratch

I put a lot of work into helping newcomers see different aspects of the Hook-ups system and, when possible, built upon relevant design skills that learners brought to the table. Learners were introduced to Hook-ups projects in a variety of ways, including seeing samples, starting from scratch, building on a physical object, and hearing or seeing peers' Hook-ups work. The following paragraphs delve into learners' reactions to different forms of introductions.

Observing learners' reactions to starting from sample Hook-ups projects brought to light benefits and drawbacks of that approach. Because it was not possible to show all projects at once, I often chose the samples that I believed could be explained well to novices. These projects were also the ones that highlighted aspects of the system I thought would appeal to the community. In some cases, it seemed as though the sample was a project that showed a group of learners something that they did not know was possible. One example in which this occurred was the case with the Scratch boxing game associated with the Fist Extender Hook-up. Learners saw the sample boxing game (Figure 47) and controlled it with the keyboard; some became interested in other ways of controlling it.

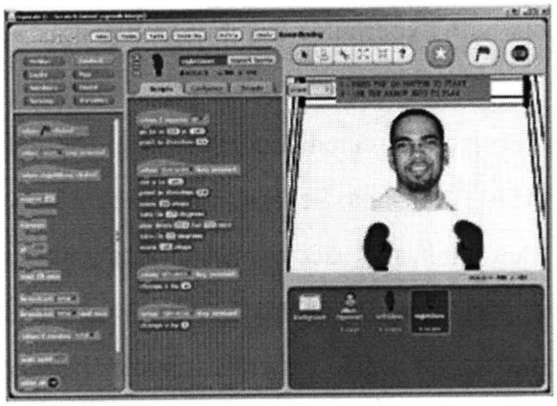


Figure 47: A sample Scratch boxing game

In other cases, the samples may not have resonated with some learners. My concern was that such samples could have given peripheral watchers an inaccurate impression of what kinds of Hook-ups could be made and influenced them never to approach me or other facilitators with their ideas. The samples presented were not intended to pigeonhole the system; rather, they were intended to serve as ideas, reference points for sample code, and embodiments of homemade physical input devices.

Leveraging Scratch Experience

There were times when I did not have to introduce learners to one aspect of Hook-ups because they were already familiar with the physical or virtual aspect. Learners who have written Scratch programs, or built custom objects before, can start Hook-ups activities by extending artifacts they have already created. If a learner has experience with Scratch, controlling characters in one of their projects with an input device s/he makes can be appealing. The extended case study of Jack demonstrated how creating a Hook-up could enhance a Scratch experience. For Jack, Hook-ups was offered as an alternative to a task he was not enjoying with Scratch alone. Later, after experiencing the combination of custom made input devices and Scratch, he began enjoying the combined result. Another example, two young men, Tom and Neal, were enjoying Scratch and, as a result of word-of-mouth, sought out Hook-ups activities to enhance their experience. These two gentlemen had learned Scratch from my former classmate. They heard of her experiences with Hook-ups and came to the Media Lab with her one day to have their own. They came armed with Scratch experience and a Hook-up idea and left with two fishing pole Hook-up add-ons. (Their building process is described later in this chapter.)

Leveraging Building Experience

Learners who have built physical objects before (using personal fabrication tools or construction kits like LEGO bricks) can start Hook-ups activities by using objects they have created to control computer programs. I was able to ensure that multiple Scratch programs were readily available for newcomers because they could be duplicated and transported easily. Changing the controls of an existing Scratch program to a control scheme that leverages previously built object by the learners can be appealing. For example, the Magnetic Stickman was originally a refrigerator magnet, but became a Hook-up when Earl (at Commu-Tech) saw me playing with the Scratch program that Kris (at Clayton) had made for his Magnet-carrying Mice.

As a result of doing Hook-ups work at Commu-Tech, I had access to a community of builders. One of the Fab Lab project's goals is to create communities of builders and inventors. At Commu-Tech, the learners responsible for early fabricated Hook-ups such as the Stickman puppets, Fist Extender, and Hard-to-catch Heart had already become familiar with the personal fabrication tools through other avenues. Introducing these learners to new design activities gave some of them more reasons to come to the learning Center, and more motivation increase both their fabrication and design skills.

8.2 How Hook-ups Builders Learned Individually and with Peers

This section draws attention to how social and group dynamics in an informal learning environment can affect the sharing of ideas and influence learners' styles and levels of participation. The first subsection introduces how learners participate on the periphery of Hook-ups activities. The second subsection shifts to how individuals engage in Hook-ups activities.

8.2.1 Participating in Hook-ups Activities From the Periphery

> Kris (talking to Amon): "You're going to hook that [slingshot] up to the computer?"
> Les: "He [Amon] hooks EVERYTHING up to the computer."

After reading these remarks, many would guess that Kris was passing by a table where Les was working on Hook-ups with me. In fact, at the time, Kris was the one working on a Hook-up and Les had never actually sat at a table with me to build any. Les had, however, been in the Clubhouse when Hook-ups activities were going on, and he had absorbed much of what the project was through observation. The example of Les reveals how Hook-ups activities can have an impact even on those not directly involved in a project. Les typically comes to Clayton to work on projects related to his expertise in 3D and audio. Les often makes comments about what he sees going on in the community; after he made the remark about hooking everything up to the computer, he helped Kris with the physical construction of the slingshot. He immediately went back to his 3D drawing program when the slingshot was sturdy. He did not take on adding the switch to the slingshot as Kris did once it was built. While he returned to only a peripheral connection to the Hook-up activity, it is both possible and likely he will become more directly engaged in the future.

Many people learn how to perform the major tasks involved in an activity by performing loosely related tasks on the periphery. Lave and Wenger (Lave & Wenger, 1991) suggested that legitimate peripheral participation (LPP) and forms of apprenticeship in social environments afford members of a community opportunities to learn about a practice both by observation and active participation. In this approach, apprentices often perform unrelated or trivial tasks while observing experts perform the major tasks of a practice within the community. Over time, the apprentices begin taking on more significant tasks on the way to becoming experts themselves.

In informal learning environments, the major tasks of a community can vary. There are often multiple "communities of practice" in each informal learning environment. Experts of one practice can be novices of another. They can gain expertise in other practices going on in their community by participating in the periphery. Hook-ups projects can build multiple kinds of expertise (particularly in the physical and virtual design domains). Learners can work towards building up expertise on their own, or through working with others in their community.

LPP is broader than a one-to-one apprentice/expert relationship; communities often include many people who can demonstrate expertise to newcomers. The inexperienced learners at Commu-Tech and Clayton were able to look to other learners who had different levels of understanding of design processes. When activities had physical world components, it was easier for the learners to become aware of the abilities and interests of others than was possible through catching glimpses of peers' all-virtual projects. This aspect of Hook-ups as a learning system is one of the ways this approach improves upon the traditional utilization of the personal computer for design projects in social settings.

Splitting Up Roles Among Hook-ups Builders

Learners collaborating on a Hook-ups project can work on one aspect of the Hook-up while participating on the periphery for the other aspect. For example, one of the learners involved in the digital puppet show was assisting with the storyline and character development on the software side of the project. When I would be working on the Puppetmaster Glove in the Clubhouse (as opposed to at MIT), sometimes one of the members working on the software side of the performance would lend me a hand by helping solder some of the electronics that needed to be re-connected to a glove. These learners were not ready to take on the full circuit design process, but they were willing to do some of the smaller tasks associated with electronics design such as soldering.

Roles Hook-ups Builders Play While Sharing One PC

The following passage illustrates an incident in which one Hook-up used simultaneously by two youth led to their collaborative authorship of a new computer program, even though only one could control the computer at once.

> Neal and Tom both were eager to use the stick-based fishing poles they made in the Fake Fishing Hole project they were working on. Magnets hung by strings on each of their poles, ready to be dangled over a pair of reed switches taped to the bottom surface of a cardboard box lake representation. They both stared at a Scratch screen and added sensor blocks to a program they had already written to simulate a lake with several fish swimming in it. Neal was in front of the mouse and keyboard adding an object to the program to represent his fishing pole. As Neal constructed a script to respond to changes in the values reported by the sensor plugged into the first sensor-plug of the Hook-up board, Tom was monitoring what Neal was doing by suggesting places where Neal may have placed the wrong block or typed in the wrong value. When Neal relinquished control of the computer to test the script he had pieced together by grabbing the fake lake and dangling his pole near a sensor, Tom was there to push the "start program" button for him and tell him how he thought Neal could fix the script if it did not work as expected. They continued this process until Tom and Neal both watched the on-screen fishing pole go deeper into the on-screen lake and get points for touching fish it encountered if the real world fishing pole magnet was near a sensor.

In the example above, Tom played the important role of debugger even though he was not controlling the actual programming himself. Neal was in the programmer role and recognized that he had an on-the-fly debugger in Tom and accordingly was thinking aloud as he programmed to ensure that Tom knew what he was trying to do and could correct him as necessary.

Assigning and Assuming Roles in Hook-ups Activities

Hook-ups activities were designed to grant individual learners opportunities to take on the role of the physical and of the virtual designer and also to foster collaboration. At times, the role switching and collaborating can be at odds. It is unclear if Tom and Neal planned to reverse roles if another opportunity to program a Hook-up presented itself, but Rogoff's research suggests they should. Rogoff asserts that people in communities of learners learn best through participation in multiple roles in that community (Rogoff, 1994). Because Hook-ups activities currently take place in after-school centers where there are rarely deadlines or graded assignments, youth can be encouraged to participate in multiple roles. One could argue that Tom and Neal's approach was more efficient than switching roles and therefore better. Rogoff suggests that this type of thinking may be a projection from the observers, rather than a true reflection of how youth learn best in communities.

8.2.2 Learning on an Individual Level

This subsection shifts the focus from examining how learners participate in learning communities to how they engage in Hook-ups activities on an individual level.

Adding Life to Crafted Goods… Digitally, At Least

Combining physical and virtual design can add much to traditional craft activities. Some learners added life to their UFOs and fishing tackle (below). Crafts have long been an important part of the educational landscape (Blauvelt et al., 1999). Craft materials can influence the virtual objects people create. Sometimes the attributes of a craft material remind people of similar physical objects. For example, a person working on a Fake Fishing Hole project held a tassel of string that looked like a Jellyfish in their hand and asked me if they could use a Jellyfish as bait in their program (I replied: "Sure, why not?"). Numerous other projects included an on-screen representation of some part of the Hook-up. Projects that contain virtual representations of some part of a real-world Hook-up include: Fake Fishing Hole, UFO Plate, Cellular Remote Control, Gravity Fighting Ball Balancer, Dropkick Duck, and the Fist Extender.

One learner had an idea for how to bring her UFO craft to life. She made a representation of a UFO using two Styrofoam plates stapled together and decorated with pens. She then added a pushbutton to the top side and connected it to the computer. I showed her how to program the button to give different features to the on-screen UFO; however, she sought more ways to bring the virtual extension of her UFO to life by adding lights to the UFO program with intentions to make multiple things respond to the device's one button.

Bringing an object to life typically starts with a model of that object. Many traditional craft materials can be used for making excellent real-world models. A set of computationally augmented physical design tools can also give learners access to model-making processes. With a model, one can experiment with how one believes something works in the world (if in fact the model is from this world, which was not the case in the UFO example). When learners' goals are not to create the fantastic (like underwater spaceships), making visible their beliefs about how real objects in our world work creates opportunities to confront misconceptions.

Although people consider themselves familiar with objects, we constantly go through life finding out new things that help us develop better understandings of how parts of the world around us work. Often in an attempt to understand how things operate, these mental models are often incomplete or flawed (Norman, 1983). Providing an experience in which people see what makes something work enables them to reconsider their mental models. According to the Piagetian constructivist theory, this reconsideration involves learners assimilating the information into knowledge they already possess or accommodating the new information by creating new mental models or modifying the existing ones that conflict with that new information (Piaget, 1972).

For example, a young person could wish to connect an airplane-like physical creation to a flying simulation they want to create. A learner who has never ridden in an airplane might program a virtual aircraft to take off vertically. If a peer notices this feature, they may start a conversation that corrects the learner's misconception.

Solving Problems and Managing Projects

Challenging one's misconceptions is valuable for learning – so is solving a problem. Hook-ups builders had to overcome difficulties and solve problems (as shown below). Because situations arise in almost every aspect of life where people need to solve problems, putting learners in situations where they can creatively invent solutions to problems provides valuable experience.

Sometimes learners face problems in a creative way if they are motivated to make progress on a personally meaningful project. For example, lacking an appropriate screwdriver for the task did not hinder progress on the Duck Hunt Zapper project. We simply cut the gun's cable and plugged the different wires that became exposed into Hook-up board sensor plugs until we noticed predictable value changes on the Scratch screen when we pulled the trigger.

Another example occurred during the first performance of the digital puppet show when the Puppetmaster Glove was not functioning as one puppetmaster expected. He approached the problem by ceasing to operate the glove the same way he had in the past; he began moving the glove's fingers slowly and holding them in a position that caused a change on the screen until he needed to trigger another event. This on-the-fly technique still preserved some of the show's story and even added some exciting unexpected effects. (They were especially exciting to the people who knew the timing of the scenes in the planned show. The audience did not know if the screen glitches were bugs or intended features because the puppetmaster kept his composure.)

Observation and experimentation often lead to learning opportunities. Staring at the inside of a console game controller, Don found an opportunity to enhance a feature he had always wanted to, but did not know how – the vibration system. He initially thought that it would be difficult to understand what caused vibration in his cutting-edge console controller. When he figured out that two motors with unbalanced weights were responsible for the vibrating he felt in games, he immediately saw the system as something he could modify. In order to achieve his goal of a super-vibrating controller (with more power than his peers' controllers), he began experimenting with the unbalanced motor weights. He simply cut pennies in half (so they matched the size and shape of the existing unbalanced weight) and glued them to the bottom of the weights hanging from the DC motors. After a few rounds of adding and subtracting weight from each side of the controller, he was very pleased with the results. The self-confidence and new knowledge that a young learner can acquire by the examination and reconsideration of long-held assumptions and pre-conceived notions can be empowering. The Hook-ups learning model provides such opportunities.

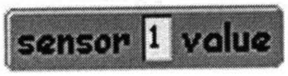
8.3 What Hook-ups Builders Learned

I have described in previous sections *how* learners were able to build a range of Hook-ups; this section focuses on *what* youth learned in the process of building Hook-ups. The first subsection focuses on sensing, programming and math. The second subsection covers design principles.

8.3.1 Programming and Sensing with a Touch of Math

Sensing

A unique contribution of the Hook-ups system is that it provides an engaging context for learning the art of sensing. The dominant field in which youth learn about sensing is in robotics, where the focus of sensing is on changing the behavior of a physical robot. In Hook-ups work focuses on controlling and manipulating virtual characters.

Steve became very engaged in the sensing aspects of the Duck Hunt Zapper project. Steve discovered that very subtle changes in the room's lighting conditions could shift the range of numbers associated with the  sensor 1 value command block he was using in his Scratch program. These constantly changing numbers corresponded to the amount of light a light sensor in the toy gun's barrel was reporting to Scratch. After doing a small amount of programming, Steve became aware that he could sense when a duck was hit more reliably if he could sense a wider range of values. Steve was trying to get a reading from the light sensor when he squeezed the toy gun's trigger sensor. He assigned a range of numbers < 70 to indicate that the gun's barrel was not in front of the most light-emitting source on the screen: the white duck. Steve then mapped sensor values > 70 to mean that the gun was picking up light from the bright duck image on the screen.

Steve spent much of the afternoon playing with numbers. He enjoyed exploring whether or not he could get a larger range of values if he changed the room's lighting settings. He did so by turning off the lights and opening windows. He did get slightly different sensor readings, but none that were very significant. He realized that there were numbers the sensor could not exceed (the sensor's sensitivity thresholds). Because Steve had not selected the sensor's sensitivity level, this was a property that was out of his control (or would be out of his control unless or until he bought another sensor or zapper).

If Steve were to build his own light gun in the future, he might select a light sensor that was sensitive to a broader spectrum of light. Selecting sensors is a critical part of achieving one's desired interaction style. Experimenting with sensors can answer the following questions a designer may ask:

- Does the range of values this sensor reliably produces vary enough to be useful for the script I have in mind?
- If the sensor is visible, does its size and mechanical function resemble what I want the interface to suggest?

The choice of which subset of sensors to initially make available to Hook-ups builders resulted from a combination of assessing the kinds of sensors the Hook-ups tools could support and determining which subset of simple sensors would enable the most types of Hook-ups. I included sensors that play roles in many people's lives, such as switches, because their familiarity helps me explain the concept of sensing and feedback in terms and scenarios learners already understand.

Steve spent a lot of time manipulating numbers in the Duck Hunt Zapper project because of the sensitivity of a sensor in the zapper. Had he built his own zapper with sensors I provided, he may not have had to use math to condition the sensor values as much. I understand that design choices affect what Hook-ups builders may learn through their explorations. I chose sensors that would allow for learners to understand the world around them.

When explaining to young people how sensors work, I often used terms the youth were familiar with (as opposed to techno-jargon) when it was possible. For example I used the term "volume dial" when referring to a rotating potentiometer. I also made sure to point out several places in the learning space that integrated sensor technology such as thermostats and alarm systems. For example, when I introduce learners to magnetic sensors, I discuss how they interact with these types of sensors every time they walk through the Center's front door. I further explain that an alarm system knows that the front door has a magnet on it that rests very close to a magnetic sensor on the door's frame when the door is closed. I then explain how the sensor changes states when an open door removes the magnet from the magnetic sensor's range. That discussion led to the question of how the alarm system was programmed to respond to the two different states. The system asks the sensor if a magnet is nearby. If a magnet is near: check again in 30 seconds. If no magnet is near: wait 30 seconds before sounding alarm. Introducing young people to their alarm system gave them an understanding of how magnetic sensors work in the context of a system that has two programmed states. Sean surprised me by speaking about how the motion sensor works for an alarm after I spent time talking about the magnetic sensor on the door. Speaking with Sean about an alarm checking the states of the sensors, it also launched us into discussion about how systems like alarms are programmed to check states.

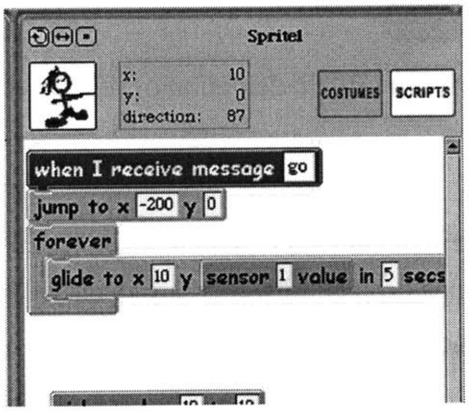
Programming

Hook-ups activities provide a good entry point for youth to explore and learn important programming concepts such as variables, conditionals, and communication between objects.

Sean was aware that sensors are usually integrated into systems that determine what to do with the sensor values. In order to make interesting interactions with sensors in Hook-ups projects, some aspects of programming proved useful. Sean learned that a simple computer script could be written to make an object on the screen transition from one state to another. To demonstrate to Sean a script with two conditions, we made our own version of an alarm: an alarm clock (of sorts) for his brother. We used a magnet and a magnetic sensor to cause a picture of his brother to appear awake when the magnet was near and asleep when it was not.

Just as Sean learned that a sensor can cause objects to vary their states and behaviors, other Hook-ups builders were learning how to use variables and conditional programming, which are useful constructs in computer programming. Physically manipulable sensors provide a good way to introduce learners to variables. The connection between physically changing a tangible sensor and a changing value in a computer program was within the grasp of novice programmers. Learning about variables through examples that only call for elements of computer programs to change the values of variables is less accessible to kinesthetic learners.

For example, the Hard-to-catch Heart mapped an on-screen character's movements to a variable. Beth changed the program to make her character move several ways, ranging from a smooth motion (similar to that of the sliding potentiometer she was using) to an unpredictable pattern (Figure 48 shows one Beth's experimental scripts). Her use of variables also called upon her knowledge of coordinate systems to make her characters perform the movements she desired.



x: 10
y: 0
direction: 87

```
when I receive message go
jump to x -200 y 0
forever
  glide to x 10 y sensor 1 value in 5 secs
```

Figure 48: Beth's early attempt at moving an on-screen character with a sensor value

Many builders interacted with on-screen objects through sensors. For some, this led to them learning how to program multiple on-screen objects to interact with one another. Kris programmed an on-screen magnet to influence another on-screen object in a manner similar to the way two physical magnets he was playing with influenced each other. He quickly picked up the concept of broadcasting messages from one Scratch object to all of the others. (An example of how a visual can help explain the concept of broadcasting is shown in Figure 49.)

I explained broadcasting to Kris in terms of his magnets (in an attempt to leverage physical materials to help explain the concept of broadcasting). He could imagine magnets broadcasting messages to other nearby magnets. One end of a magnet could broadcast a message, "Come to me," while the end of a nearby magnet is always listening for that message. When second magnet's end hears the message, it responds with an action, "move towards the broadcaster." A different magnet with one of its ends near the broadcasting magnet can hear that same message and do a completely opposite action, "run from the broadcaster." After this discussion, Kris made an on-screen magnet broadcast "magnet-on" and "magnet-off" messages. When a metal-carrying character on the screen heard the broadcast, it would move towards the broadcasting magnet.

Figure 49: An artist's depiction of object broadcasting

Spatial Learning

In programming with Scratch, learners program objects to explore in two dimensions. Physical aspects of Hook-ups project, however, are inherently three-dimensional. Some learners exercised their spatial abilities in the physical realm of Hook-ups projects.

For example, in the process of creating the Fist Extender project, Sean gained a realization that I did not expect: that the linkage should rest horizontally on top of the base (going into the Z-axis) instead of connecting to the surface of the base (along the Y-axis). He demonstrated spatial thinking skills at a point where we were not even working with the physical materials yet. In the process of drawing and cutting links on a piece of cardboard, Sean could envision turning the linkage mechanism we didn't have in front of us yet into the cardboard. He was thinking farther ahead than I was and showed me his idea by holding his hand like it was the scissor links and orienting his hand at a 90-degree angle to the cardboard and saying, "like this."

8.3.2 Design Principles

In the process of working on Hook-ups projects, youth develop a deeper understanding of particular design principles – that is, strategies and heuristics for guiding the process of design. Design principles are exercised and developed in every stage of Hook-ups projects from brainstorming to presenting a finished Hook-up to peers. Hook-ups are particularly well-suited for exercising two design principles:

(1) Build complex constructions from simple parts

(2) Use materials in unexpected ways.

Discovering the simple parts of complex constructions

In building Hook-ups, youth developed a better understanding of how complex constructions can be built from simple parts. Interfaces built with Scratch Patches inherently rely upon this skill. For example, the Fist Extender has many mechanical functions, but is primarily constructed with one simple part: a scissor link chain piece. Projects that could be broken into modules were excellent tools to introduce people to this design skill and the idea of modularity. (All Scratch Patch interfaces are modular.)

When learners began to understand that complex objects could be built from simple components, *deconstruction as a path to construction* became a valuable activity. Learners who looked inside objects got a sense of the kinds of components professional designers utilize. The learners gained new understandings about the world they live in by becoming familiar with components that were at the core of many objects around them. The notion that all systems are built from components at some level served the learners well in design activities.

As Don did with his Xbox® controller, Steve explored the unfamiliar side of an object he was familiar with and found a surprise. In the process of starting the Duck Hunt Zapper Hook-up, Steve unscrewed an existing video game gun and was shocked (as was I) when he realized that only a few electronics (just a photo transistor, a tiny PCB, and a switch) were necessary to make the gun work. It seemed like Steve had not imagined that most of the inside of this commercial electronic toy could be empty space.

In Jack's Cellular Remote Control, the first item he found when dissecting a toy cell phone was a piece of rubber with nine lumps instead of the nine individual key objects he expected. This discovery simplified the process of transferring the buttons to the surface of his remote control housing, but required him to rethink if/how he would be able to sense the press of a single button.

Earl was able to pinpoint where I "borrowed" a simple component from to use as a Hook-ups sensor. He saw a familiar dial and looked at me, asking "Isn't that the volume thingy?" He was correct; the volume thumbwheel typically found on personal audio players was a good size and feel for Scratch Patches and, as a bonus, afforded familiarity.

Using Materials in Unexpected Ways

By a construction material failing while Sean was piecing together parts of the scissor links to realize his new Fist Extender design, he discovered that connecting the link pieces with one another at different points created new types of toys. To make a scissor link chain, one long slender rounded rectangular piece is connected to its neighbors through pivot holes at its top edge, center, and bottom edge. When two of the cable ties that were connecting the link chain broke, and others were not in place yet, Sean noticed how the new mechanism fell to the floor and got an idea. He picked up the barely-connected pieces and began swinging them around like nunchucks (martial arts weapons shown in Figure 50). When Sean got too close to his friend while swinging the nunchucks, his peer picked up the other part of the former scissor-link chain (which was two links connected only by the middle pivot point) and spun one link around, creating a propeller that Sean protested would chop up the nunchucks if they got too close.

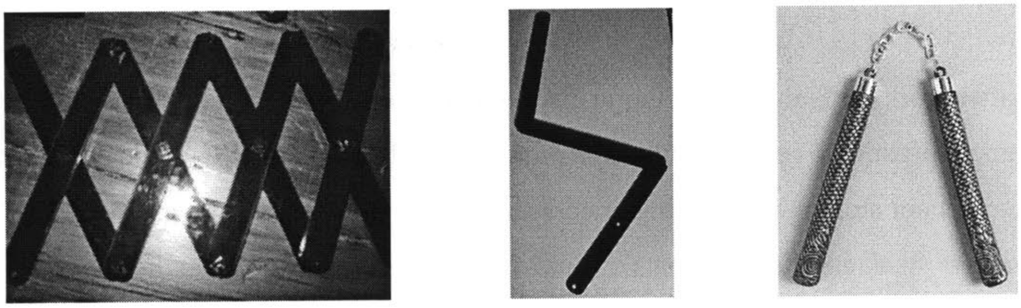


Figure 50: Scissor links, links connected like nunchucks, ninja nunchucks

Sean initially laser-cut links for the sole purpose of constructing scissor link chains; when the material holding one of his chains together broke, he realized that multiple uses he did not initially imagine were possible. At the moment of the break, Sean demonstrated what good designers do: use material in unexpected ways. Perhaps he will make a nunchucks-based Hook-up in the future?

In making fishing poles for the Fake Fishing Hole project, a learner rejected the prior art of moving a magnet over stationary sensors to "fish." Instead, he fished for stationary magnets with moving magnetic sensors. This idea came to his mind as one of his friends was looking for string to serve as the fishing line on a fake fishing pole. While holding the sensor and waiting for the friend to find string, he realized that he was holding an alternative to string in his hand: wire. The wire was already connected to the sensor. He found it easier to stick magnets on a surface and make the sensor go to them. Either way, the magnetic sensor switch gets closed and the computer scripts function as they are supposed to.

Instead of using scissors to cut, the Scissor Switch creator used them as a detonator for on-screen explosions by taping two wires to where the scissor handles meet. Everyone around this learner was using the scissors as a tool. No one expected the newcomer to transform the scissors into a design material and later into a detonator. This discovery caused other learners to think about what kinds of tools they took for granted that could be repurposed. The rest of the learners working on Hook-ups at the time the Scissor Switch was created tried to make detonators from forks and knives before the Clubhouse closed.

A learner working with Earl on fabricated puppets used magnets as building blocks to construct “prosthetic legs” for a broken puppet modeled after the Magnetic stickman. The learner did not stop after replacing once-acrylic legs with magnets, he gathered more magnets to stack another two rows of them to serve as crutches so that the figure could get used to its new legs.

8.4 Obstacles to Learning

This section brings to light obstacles I encountered while conducting this research. The first subsection relates to social barriers. The second subsection is about physical and logistical obstacles.

8.4.1 Social Obstacles

For activities designed to engage as many learners as possible in a mixed-age informal learning environment, it is important to understand that social hierarchies and interpersonal dynamics can compromise the potential for some to have beneficial learning experiences.

When physical materials are involved, sometimes age can play a restrictive role. For example, in the Custom Console Controller project, Don and his teenage peers assumed financial responsibility to replace parts if anything went wrong. One of the reasons that Sean, who is a pre-teen, was instructed to watch over the project instead of participate in it could have been that he had fewer economic resources and thus was less able to replace parts. Sean, and peers his age received little resistance from older peers when they worked on virtual objects because computer bits are free, for the most part. Young learners at Commu-Tech were generally allowed to turn their virtual creations into cardboard physical objects cut from the laser-cutter. They were constantly reminded to use the less abundant acrylic sheets sparingly.

In the Puppetmaster glove project, a number of factors contributed to young members participating more peripherally than their older peers. Older siblings in the Clubhouse sometimes felt as though they needed to protect the younger members from danger during the parts of the activity that required using a soldering iron to add electronic components to the glove. Additionally, the people who had the most say in how the glove operated were the two puppet masters who would be wearing them in a performance. Any member could have been a puppet master, but the role of being the lead performer in front of an unfamiliar audience did not attract the interest of any young members. They played roles such as carrying signs around the stage. In contrast, both young and old learners at the Commu-Tech center are trained on and use tools like soldering irons and computer-controlled drills because they are part of the everyday landscape.

Similar to the way social hierarchies can compromise the learning environment, social dynamics can be further complicated when family ties are also present in the groups that utilize community centers. In that case, the predefined roles of family interactions can be more difficult to deviate from than in the case of learners who are merely classmates.

8.4.2 Physical Obstacles

In addition to the social obstacles discussed in the previous subsecion, there are sometimes physical obstacles that compromise learning. Typical community centers do not have adequate space to provide lockers for each member the way schools can. Therefore, members sometimes come up with their own storage schemes. In the Customized Console Controller project, the construction material included controllers for cutting-edge gaming systems that other community members coveted. The people working on the project found a place to hide their parts when they were not in the Center in an unlabeled cardboard box. Their controllers lined the bottom of the box and undesirable objects like empty CD cases filled the top. Their hiding scheme was successful. Because of the informal setting, so many activities take place that people who were not around during any of the controller modifying sessions would be unlikely to seek out any hidden materials.

When people have the ability to reproduce physical objects, as is the case in Fab Labs, storage of physical objects is not so large an issue (although there are still many valuable physical objects that cannot be easily reproduced). A friend of Earl's was upset that one of his bear figures turned up missing. He was only upset for a few moments before he located the bear schematic on a computer and laser-cut another one. If the computer he used to create the schematic had been occupied, he could have downloaded his design file from an online repository to which learners save backups of their work.

Another problem with learners using computers to work on projects is that some people view *personal* computers as private places. If and when some learners choose to explore computer programming on personal computers in learning centers, their programming would be likely to go unnoticed. Peers sharing the learning space with these learners would be less likely to walk up to the programmers and ask them what they are doing with an on-screen program than they would be if the programmers were building physical objects. It is my impression that this inhibition makes the sharing of computer code rare in many Clubhouses because members are often unaware of their peers' programming efforts.

This lack of awareness could be the result of the Computer Clubhouse lacking a visible community of programmers. Even though some members could be separately engaging in programming efforts, factors other than the privacy of personal monitors can decrease the chances of communities forming around programming in informal settings. For example, searching through personal project folders on shared network drives in which peers store their programs is often taboo. These factors make it difficult for ideas embodied in computer programs to influence others unless learners happen to catch other in the act of programming.

In spite of the obstacles discussed, the Hook-ups learning model effectively engaged youth in a range of design activities. In the next section, I raise issues for educators and designers based on what I learned while interacting with these learners.

8.5 What the Learners' Experiences Tell Us As Educators and Designers

This section presents my reflections on the process of working with youth during this Hook-ups study. The first subsection describes the balancing act mentoring participant-researchers perform while working in informal-learning settings. The next subsection focuses on the tradeoffs of strategies for facilitating design activities in informal settings.

8.5.1 The Balancing Act of the Mentoring Participant-observer

I believe that mentoring researchers should wear many hats (play multiple roles) in a community.

A common means of gaining access to observing participants in an informal-learning setting is through mentoring. When a mentor is a participant-observer and intends to introduce the community to a technology being developed, the participant-observer boundaries may become unclear. There are many gray areas between introducing technologies to a community with the expectation that it will *help community members learn* versus having the expectation that the introduction will *help researchers learn how to help the community with their technology later*. The expectations of the participant-observer and the community's leader must be clear with each other about what each party expects.

Balancing Expectations

Many community leaders and members in informal learning settings do not want to feel like research subjects. It is important for researchers who work in community settings to be sensitive to this issue. Hook-ups research was conducted in several communities under several different understandings/expectations of the learners and group leaders. A participant can agree to help a community learn *WHILE* the research is being conducted or help the community learn solely *BY* including the members in the research. The former is the primary condition under which I conducted the Hook-ups research. Helping the community learn while conducting research often entails playing other roles as a mentor – sometimes at the expense of collecting data (specific to the research being conducted).

Some roles I normally play as a mentor include friend, part-time counselor, audience member, guide-on-the-side, workshop facilitator, and technical resource. I excluded the "harsh disciplinarian" role because playing that role can negatively affect relationships necessary for working with youth. There are ways for mentors to be non-harsh disciplinarians in that they set limits for what type of behavior they will tolerate before it negatively affects the activities. I also try to avoid giving children too much attention and help – a habit also known as hypermediating (Gutierez & Stone, 2002).

In contrast to providing so much help that it takes undue time away from their research, some researchers inadvertently fail to honor their commitment to help the community members learn in other avenues unrelated to a research agenda. This oversight is sometimes due to frustration on the part of the participant-observer. It is important for participant-observers to realize that making contributions to a small learning community is not a waste of time if the research results that were expected did not materialize.

Another potential problem arises when a researcher does not anticipate that serving multiple roles in a community can slow down the observation part of the research. Depending on the researcher's point of view, playing multiple roles can be a distraction from the intended research objectives or can enrich the findings. For researchers whose work is best informed by observing learners use a technology, playing multiple roles in a learning community may be considered a detriment. In contrast, those who are informed by the situations in which a use for a technology emerges can benefit greatly from observing while participating in the community when technology prototypes are not being used.

Conducting Hook-ups research while also playing several roles in the communities benefited my research by giving me a better understanding of the social dynamics of the settings. Playing several roles in the communities also satisfied the part of me that would have been in the centers mentoring anyway, had I not been researching the Hook-ups projects. That side of me is aware that a community of learners can benefit from my research, but also from sharing other life experiences as well.

A technology designer who plays multiple roles in a learning community has to manage his or her time effectively. Both the developing technology and gaining/maintaining social acceptance as a designer in an informal learning environment take time. This is one of the reasons I have chosen to make my system entirely from parts that are available in one of the research sites. At any given time, when I decide whether to focus on a system redesign or to help a learner with an unrelated task, I can switch back and forth without losing time for traveling. This way, I hope to have ample time to make an impact on the community with my presence as well as with technology I make increasingly capable of accommodating more of what learners wish to do.

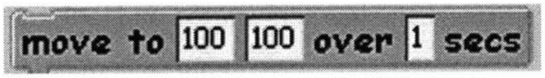
8.5.2 Strategies for Facilitating Technological Design Activities in Informal Settings

While participating in the Clayton and Commu-Tech communities (over the last 12 months) and simultaneously designing technology to support creative learning, I tried an assortment of introduction strategies, found inspiration for supporting certain interactions, and was confronted with tradeoffs.

A vital step to getting learners started on Hook-ups projects is to help them imagine the creations they would like to build but manage their expectations in the process. Examples of what can be done with a set of tools, like Papert's "Twenty things to do with a personal computer" article (Papert & Solomon, 1971), or sample projects, are useful for giving learners ideas for creatively using the tools they have available. The art of taking a grand idea that a child is in love with and making it a realizable project that is still interesting to them is an art that requires practice.

Once a learner is interested in making a realizable project with a set of technological tools that are in the prototype stages, it is imperative to have a backup plan, with a different technology, in case the prototype malfunctions. It is valuable for designers to use new prototypes with learners to expose what aspects worked well and which ones need work. But it is equally important for the learners to have the opportunity to go through their first design processes (while they are forming impressions of your system) with reliable tools that will help them progress towards their goal so that they can have something to show for their efforts. Much of the work presented in the previous sections was done with a standard Hook-ups board (instead of using Scratch Patches exclusively) to ensure that the activities would run smoothly and learners would get a chance to see their designs come to life.

The fact that Hook-ups projects can focus on physical aspects or virtual aspects at any given time provides an extra safety valve for technological malfunctions. In the event of technologies malfunction, learners can work on aspects of a Hook-ups project that do not require computers such as mechanical design.

The Scratch programming environment, like Scratch Patches, was evolving during this work. If a learner was having problems with a version of Scratch, an older version that had been tested more could usually be loaded. This caused some learners to become more accustomed to working with different versions of Scratch than others, but that did not amount to much of a problem. At times when added features of new Scratch releases would help a learner's project (for example, a command block's wording became easier to understand as was the case with the  move to 100 100 over 1 secs or when the interactions became more intuitive), it sometimes required small conversions to projects that were in-progress in order to make them compatible with the latest version of Scratch.

Putting the best possible tool in front of a learner for their design task was important. I based several design decisions about new features to incorporate into Hook-ups tools on what learners found useful while making Hook-ups. For example, after Scratch became more intuitive for learners to put costumes (images) over blank background canvases, I decided to do something similar with Scratch Patches. To make Scratch patches better suggest that they could be empty canvas awaiting decoration, I made all of the current Patches white, as to suggest blankness. They can be decorated like a canvas or dawn physical costumes. I added holes to Scratch Patch layers in which one can secure pipe cleaners and other decorative craft materials.

9. Towards Future Hook-ups Systems

I have developed the Hook-ups system to accept Eisenberg's challenge to blend physical and virtual activities to improve the types of computer-enhanced educational experiences envisioned by Papert. The system builds upon the early work of combining computation with physical objects like the LOGO Floor-Turtle and extends recent work with programmable bricks. This study of Hook-ups was the first (of many, hopefully) exploration of how and what young people learn while creating customized physical computer input devices.

The social conditions in which several Hook-ups were inspired and created have been described. By discussing case studies and revealing the trajectories of several Hook-ups, I have underscored the importance of having multiple paths for multiple kinds of learners.

Although this work involved the creation and enhancement of a computational construction kit and a programming environment, the goal was not to make revolutionary advances in the areas of physical design and computation, but rather to blend the two in a way that would help people design and learn in a new way in an unexplored domain. As stated at the outset, my goals were:

- to deepen our understanding of how the Hook-ups learning system can inspire creative output from youth
- to develop technologies that enable a new range of design activities that combine physical and virtual design processes
- to design strategies for introducing a new system to learners in informal-learning environments.

I did not intend for this work to end with succinct conclusions (although guidelines for educators and designers are presented throughout the thesis). I desired to justify future research in blending physical and virtual activities. The research represented in this thesis has supported my initial hypotheses that:

- design is a good context for learning
- design activities should take learners' individual styles into account
- supporting design-based learning in informal-learning environments is challenging yet possible.

These preliminary findings confirm that further research is in order. These findings suggest that the Hook-ups system provided a wide range of good design contexts. Through building Hook-ups in informal settings, youth learned core design principles, such as building complex objects from simple parts and embracing the unexpected. They were also able to express themselves in new ways.

Although the Hook-ups system gave youth with multiple learning styles a wide range of design activities, several obstacles to learning were still encountered and discussed. I included such discussion to serve as a testament to how many nuances and variables are realities of conducting research in actual informal-learning settings. The discussion also shows that there remains much room for improvement in the Hook-ups system. I intend to improve variants of this system as a part of my doctoral work.

To continue this work, I would need to focus on the following directions:

- continue to redesign and refine Hook-ups tools
- refine strategies for interacting with youth around Hook-ups
- help shape the learning model of future learning spaces, formal and informal alike.

I believe that pursuing these directions will help provide more contexts and means to take advantage of design-based learning centers. It is important for centers in underserved communities to provide options to youth who may have little access to other resources to keep them excited about learning. The establishment of such learning centers in underserved communities serves the additional purpose of creating islands of safety as well as scholarship in otherwise dangerous terrain.

The growing number of designers and educators interested in design and informal learning centers need to work as a team with the common goal of developing playfully inventive learners. Society will always need creative minds to be the next generation of scientists, inventors, and provocative artists. The right combination of capable tools, supportive people, and comfortable learning settings should serve to do more than just tell young people that they can be designers of the future: it should make them believe it and help others do the same.

9.1 Improving Hook-ups Tools

Future work on Scratch Patches will take into account the learners' experiences presented here and seek to better support the range of activities youth were attracted to. It will also incorporate advances in fabrication and communication technology that do not detract from the system's goals. I discussed the balancing act mentoring researchers must perform; advances in technology also play a part in that performance. I must continue to choose technologies that best support the learning goals of the system while making advancing technology a factor of the work, but not the project's dominating factor.

For the software side of the system, the Scratch research team will continue to refine the software until a stable release is ready for interested learners to download and explore without live support from the developers. Hook-ups will stay strongly connected to its roots in Scratch and also explore connections to other software, including 3D modeling environments. Scratch was created with young designers in mind: unfortunately, that was not a top design priority for much other design software youth would need in order to take advantage of forthcoming three-dimensional printing devices. As fabrication tools advance to make 3D output accessible to non-machinists, design environments that allow youth to make use of these devices for Hook-ups-like activities will be in order. This coming generation of technology will also need designers to connect it with craft applications.

It is to be expected that some young and imaginative minds will desire to do things that are presently outside of the capabilities of whatever system they use, Hook-ups included. As noted in the previous pages of this thesis, there will always be a need to help youth come up with realistic designs for the tools they have access to. At the same time, the tools should become more capable of doing what youth want them to. For youth who become interested in increasing the capabilities of the Hook-ups tools provided to them, I will work with them to help them take an active role in the design of the next set of tools. Youth should have access to the facilities to improve their tools in the future.

9.2 Improving Interactions with Learners

As the Hook-ups project matures, I hope to supplement the improved tools with improved means of documenting ideas and projects that show the range of what can be done. In addition to continuing to create and collect sample projects that can be used to inspire learners, I wish to document written ideas for sample projects and walk-through tutorials that can help support future Hook-ups builders.

As with Papert's list of 20 things to do with personal computers, proposed when they were in their infancy, I will make available a multitude of activities that Hook-ups tools could be used to achieve. Ideas for these documents will be gleaned from future projects created by youth.

As project tools, activities, and support materials improve, a natural question to ask is, "will the Hook-ups idea scale?" I hope to answer that question with future research. Due to the physical nature of Hook-ups work, scaling may be especially difficult throughout geographically separated communities. As it is difficult to characterize and describe 3D objects through 2D mediums like computer screens, sending project explanations over networks may not do the Hook-ups being sent justice. Additionally, learners in different parts of the city, country, or world may not have access to the same resources as other learners. This difference is why the Hook-ups process intends to develop learners who embrace and understand the materials around them. They will find what they need to design.

In Fab Labs, sending files containing schematics for building physical things makes sharing physical designs possible. In which case we would need to create better tools to support learners sharing designs across communities accessible. This may enable distant communities of inventors to be capable of supporting each other. When learning spaces can incorporate supporters from far-off and unexpected places, more opportunities for design may arise.

I also intend to refine the types of research methods I use to evaluate how technologies are introduced to informal settings. As the design phase of the Hook-ups project progresses and the communities of Hook-ups builders grow, rethinking the research techniques will be in order. As I mentioned, scaling needs to be considered, but continues to be an Achilles heel for designers and educators.

9.3 Preparing for Future Learning Spaces

The mainframe computers behind much of Papert's early research evolved into the personal computers that occupy today's learning spaces and can be powerful learning machines for the many children who have access to them. Similarly, I expect that evolved forms of the personal fabrication devices explored here will improve future learning environments. Future Computer Clubhouses may look somewhat like current Fab Labs. In these places, where personal fabrication devices are ubiquitous, perhaps youth will make their own novel physical input devices, computational construction kits, and even their own Floor-Turtles.

The existence of such centers is critical to the development of youth. Much of our education takes place outside of school settings. Outside of under-resourced schools, some underserved communities do not get much access to learning outside of these schools. It is my hope that future after-school learning centers will continue to grow in underserved communities to help mitigate a lack of resources problem. These learning centers should serve to keep the scientist alive in all young people. Technology and ideas should be made available so that they do not intimidate youth. Youth should be able to fearlessly embrace technologies to express themselves; they should be the ones designing the next generation of them. If they have early access to the tools, and people that believe in them, they may ultimately make this vision a reality.

REFERENCES

Atmel. Retrieved May, 2005, from the World Wide Web: http://www.atmel.com

Blauvelt, G., Wrensch, T., & Eisenberg, M. (1999). Integrating craft materials and computation *Proceedings of the 3rd conference on Creativity \& cognition* (pp. 50-56). Loughborough, United Kingdom ACM Press.

Druin, A. (2002). The Role of Children in the Design of New Technology. *Behaviour and Information Technology (BIT), 21*(1), 1-25.

Eisenberg, M. (2003). Mindstuff Educational Technology Beyond the Computer. *Convergence*(9:2), 29-53.

Eisenberg, M., Eisenberg, A., Hendrix, S., Blauvelt, G., Butter, D., Garcia, J., Lewis, R., & Nielsen, T. (2003). As we may print: new directions in output devices and computational crafts for children *Proceeding of the 2003 conference on Interaction design and children* (pp. 31-39). Preston, Engalnd ACM Press.

Emerson, R., Fretz, R., & Shaw, L. (1995). Fieldnotes in Ethnographic Research, *Writing Ethnographic* (pp. 1-16). Chicago, IL: University of Chicago Press.

Gershenfeld, N. (2005). *Fab: The Coming Revolution on Your Desktop - From Personal Computers to Personal Fabrication.* New York, NY: Basic Books.

Gershenfeld, N., Krikorian, R., & Cohen, D. (2004, October). The Internet of Things: The principles that gave rise to the Internet are now leading to a new kind of network of everyday devices, an "Internet-0". *Scientific American,* 76-80.

Gutierez, K., & Stone, L. (2002). Hypermediating literacy activity: How learning contexts get reorganized. In O. Saracho & B. Spodek (Eds.), *Contemporary Perspectives in Early Childhood Education* (pp. 25-51). Greenwich, CN: Information Age Publishing.

howstuffworks.com. Retrieved March, 2005, from the World Wide Web: http://www.howstuffworks.com

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitamite Peripheral Participation.* CAmbridge, UK: Cambridge University Press.

Mikhak, B., Berg, R., Martin, F., Resnick, M., & Silverman, B. (2000). To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines. Robots for Kids: Exploring New Technologies for Learning Experiences. In A. Druin (Ed.), *Robots for Kids: Exploring New Technologies for Learning Experiences.* San Francisco, CA: Morgan Kaufman / Academic Press.

Mikhak, B., Lyon, C., Gorton, T., Gershenfeld, N., McEnnis, C., & Taylor, J. (2002, December). *Fab Lab: An Alternate Model of ICT Development.* Paper presented at the Development By Design 2002, Bangalore, India.

Norman, D. A. (1983). Some Observations on Mental Models. In D. Gentner & A. Stevens (Eds.), *Mental Models.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Papert, S. (1986). *Constructionism: A New Opportunity for Elementary Science Education* (Proposal to The National Science Foundation). Cambridge, MA: MIT - The Media Laboratory.

Papert, S. (1993). *Mindstorms : Children, Computers, and Powerful Ideas* (2nd / ed.). New York: Basic Books.

Papert, S., & Solomon, C. (1971). *Twenty Things to do With a Computer* (3). Cambridge, MA: Massachusetts Institute of Technology: MIT Logo Laboratory.

Papert, S., & Turkle, S. (1992). Epistemological Pluralism and the Revaluation of the Concrete. *Journal of Mathematical Behavior, 11*(1), 3-33.

Piaget, J. (1972). *The Psychology of Intelligence.* Totowa, NJ: Littlefield, Adams & Co.

Resnick, M. (1991). Xylophones, Hamsters, and Fireworks: The Role of Diversity in Constructionist Activities. In I. Harel & S. Papert (Eds.), *Constructionism.*

Resnick, M. (1993). Behavior Construction Kits. *Communications of the ACM, 36*(7), 64-71.

Resnick, M., Kafai, Y. B., & Maeda, J. (2003). *(ITR proposal for) A Networked, Media-Rich Programming Environment to Enhance Technological Fluency at After-School Centers in Economically-Disadvantaged Communities* (proposal): National Science Foundation.

Resnick, M., Martin, F., Sargent, R., & Silverman, B. (1996). Programmable Bricks: Toys to Think With. *IBM Systems Journal, 35*(3-4), 443-452.

Resnick, M., & Rusk, N. (1996). The Computer Clubhouse: Preparing for Life in a Digital World. *IBM Systems Journal, 35*(3), 431-439.

Resnick, M., Rusk, N., & Cooke, S. (1999). The Computer Clubhouse: Technological Fluency in the Inner City. In D. A. Schoen & B. Sanyal & W. Mitchell (Eds.), *High Technology and Low-Income Communities: Prospects for the Positive Use of Advanced Information Technologies* (pp. 264-286). Cambridge, MA: MIT Press.

Rogoff, B. (1994). Developing Understanding of the Idea of Communities of Learners. *Mind, Culture, and Activity, 1*(4), 209-229.

Steinmetz, J. (2001). Computers and Squeak as Environments for Learning. In K. Rose & M. Guzdial (Eds.), *Squeak: Open Personal Computing and Multimedia.* New York, NY: Prentice Hall.

Wertsch, J. V., & Stone, C. A. (1985). The Concept of Internalization in Vygotsky's Account of the Genesis of Higher Mental Functions. In J. V. Wertsch (Ed.), *Culture, Communication and Cognition: Vygotskian Perspectives* (pp. 162-179). New York: Cambridge University Press.