5. Community Support for Construction

"Another thing about moose-crossing is that I feel as if I can really help someone. I like learning and doing stuff on my own, but the real reason I come to moose-crossing is that I feel needed, and wanted. While programing is a lot of fun, I don't think I'd do it, if there wasn't anyone who would appreciate it."
— Rachael (girl, age 13)

5.1 Uzi's Frustration

Uzi (boy, age 13) was excited about joining MOOSE Crossing, and his mother was delighted. She wrote, "He is 13 and is doing a lot of MUD'ing lately. I'd be thrilled to have that energy turned towards something more 'scientific.'" Approximately six percent of MOOSE Crossing members were already members of other MUDs before joining. Before he even sent in his permission forms, he was mailing me detailed questions from his mother's account:

```
Thank you for the help!  By the way, this is Uzi speaking.  How do I
program socials to do different tthings with different subjects?  For
example, I type safe jed".  I see " You drop a fifty ton safe on
jed", Jed sees "  AAAAAAH! there is a safe falling on you!",  and
everyone else in the room sees "SQUISH!".
```

He mailed in his permission forms, and first connected a few days later. No other kids were online at the time. A veteran MUDder, he knew immediately he wanted to describe himself and set his gender. Unfortunately, a couple days earlier I had changed the syntax of the "gender" command. (It was changed from "gender me is <gender>" to "gender me as <gender>" to avoid confusion with use of "is" as an infix operator.) I announced the change in the newspaper, but neglected to update the help message for "gender." Uzi tried several variations on the command to no avail. He finally pounded out on his keyboard "kill self." This, amusingly enough, printed out some help on how correctly to kill a task (running program). Uzi was used to MUDs where killing people and things was normal; on MOOSE Crossing the command translates to an obscure and rarely-used advanced technical command.

This was relatively early in the development of MOOSE Crossing, and the population was still small. Most kids knew to come by on Monday afternoons, when there were always lots of other kids around. On this Friday afternoon, only Case, an MIT undergraduate working on the development of the MacMOOSE client program, was logged on. Uzi couldn't figure out how to contact him. Uzi wandered around the virtual world a bit, and then gave up. Three days later he connected again at 9:30 PM. This time, no one at all was logged on—most MOOSE participants are younger than Uzi, and 9:30 PM is quite late. He wandered around the virtual world for forty minutes, and then gave up for good.

Uzi's experience was fraught with problems. His past MUDding experience worked against him—instead of consulting the list of basic MOOSE commands sent to him with his password, he kept trying commands from other MUDs that did not work. That gap was cultural as well as technical—he spent a significant chunk of his time repeatedly trying to kill things. Most kids intuitively understand from the code of conduct that killing is not allowed. Uzi was so accustomed to killing things on MUDs that he failed to make that inference. Most importantly, Uzi failed to meet anyone else online.

MUDding alone is rather like bowling alone (Putnam 1995)—it misses the point of the activity. Other children could have helped Uzi with his technical difficulties, and shown him around the virtual world. He did wander by a number of significant projects made by other children, but paid little attention to them. Those projects would have been much more interesting if proudly presented by their owners. His experience is in singular contrast to that of Storm, who was practically overwhelmed by Rachael's eagerness to help her. In this chapter, I detail ways in which community supports construction activities.

5.2 Pilot Study: Programming for Fun

In 1993-1994, I conducted an informal study of twelve adults who learned to program for the first time on MUDs, just for fun. Eight learned on MediaMOO. The twelve subjects programmed a variety of objects. Jim is a 43-year-old building contractor who lives in the Pacific Northwest of the United States. He built a "77 Jeep Cherokee." Susan is a thirty-eight year old professor of agriculture from the South. She built a simulation to help her students learn to perform certain laboratory procedures. Christopher is a 23-year-old graduate student in comparative literature in South Africa. He programmed a poetry generator. Neil is a 30-year-old graduate student in American cultural studies in the Midwest. He built a "RetroFuturist Aerodrome" complete with a biplane, a blimp, and a taxi stand for those who want to catch a ride.[1] I'll describe Jim's experiences in detail.

Jim has always lived in the Pacific Northwest of the United States. After he graduated from high school, he was drafted into the army. By a stroke of luck, he narrowly avoided being sent to Vietnam. After two years of military service, he attended two years of junior college on the GI Bill. He took what

[1]Parts of this section appeared in (Bruckman 1994).

he describes as "just the fun classes" and never got a degree. In need of work, he learned that a friend of a friend needed help building a house. After a number of years, he built up enough expertise to start his own building contracting business. Jim has little mathematical or scientific background, and had never programmed before he encountered MediaMOO:

> I work on a computer at home. I don't have it in an office, at school, or anything like that. We use a Macintosh. My wife was a graphics designer and she needed a computer to stay in her field. I'm a building contractor. She bought a Mac about four years ago and started doing design on the computer, and I started using it for my business, just playing with it, and really liked using it.

Asked why he decided to try programming, Jim describes a desire to contribute to the community:

> Since I started getting involved with this I became interested in programming because I wanted to be able to manipulate this world in a better way. And instead of just looking at things and creating things that already existed, I wanted to try and put my hand to actually creating something of my own.

Since the MOO language is object-oriented, it is easy to make useful objects simply by making something which inherits from an existing parent object. By "creating something that already existed," Jim means making an object which inherits its functionality from someone else's object. You can make a television set with one simple command, making something which has generic television set as its parent. Making something which inherits from an existing parent object is an easy first step towards programming. It is part of what encourages people that they can and would like to make something of their own.

The virtual world on MediaMOO is filled with interesting objects made by members. These serve as models for new users. Every object in the system functions as a possible model to learn from and be inspired by, just like every page on the World Wide Web is a potential model for learning HTML. It's significant that each model is closely associated with its author, whom you might happen to meet online. Examples to learn from are situated in a social context.

Jim also describes the MOO environment as an interactive story to which he wants to be able to contribute:

> It's like reading a story that I get to help write. And I wanted to get more input into what was happening, beyond just chatting with people

> and creating a box or whatever... I wanted to try and do something with it.

The community of learners present on MediaMOO was one of his key motivations for learning to program. He wanted to be able to contribute something to that community. Objects are conversation pieces in MUDs. When Neil flies by in his blimp, others stop to admire it and ask him how he created it. That becomes the basis for striking up a conversation, rather like dog owners in a park being able to strike up a conversation about dogs and then segue to other topics. Making an interesting object confers status within the community to the object's designer. Jim wanted to build something of his own rather than simply admiring everyone else's work. Analyses of individuals interacting with standalone computer systems such as those in Sherry Turkle's *The Second Self* often focus on the individual's quest for mastery (Turkle 1984). That quest takes on added dimensions when the activity is strongly situated in a supportive community context. Other residents of the virtual world form an appreciative audience for work. In addition to wanting the personal satisfaction of having mastered the technical environment, Jim also wanted to be able to use his new creation as a catalyst for social interaction and conveyer of added social status. On a more altruistic level, he said he wanted to contribute something to a society from which he has benefited. People's motivations for trying to program something in MUDs are primarily social.

The first step in learning to program is perhaps the hardest. The initial barrier is primarily emotional. Most adults who do not have formal technical education suffer from some degree of technophobia. On MediaMOO, a user named "cdr"[2] made a set of clear, simple tutorials. Seven of the eight MediaMOO programmers interviewed began by doing one of these tutorials. Susan comments that for her, the tutorials' primary function was to help her overcome that initial emotional barrier: "I did cdr's tutorials and I realized 'I can do this!' But then I had to step back and figure out what it was that I had done."

Cdr began by programming a "television set" object. You can turn on your television set and receive different "programs" on different channels. One of the first shows he added to the television schedule was called "Coding with cdr," an instructional video teaching you how to program a box of donuts that you can eat. The metaphor of an instructional video seems to put people at ease. The success of these tutorials also depends on cdr's friendly, warm personality. His manner puts people at ease. The tutorials are not detached technical instructions, but help from a caring, supportive person. Jim enjoyed doing the tutorials:

---

[2]The character "cdr" is Dr. Kenneth G. Schweller, Professor of Computer Science and Psychology at Buena Vista College in Storm Lake, Iowa.

> I'm totally program-language illiterate. Well, I found cdr's tapes. Cdr was able to lay out exactly what was going on and lay it out step by step—you watch his tape, and you do what he says. And it worked! I was able to start to understand what the basics of it was all about. I made his box of donuts, and then I made myself a car to drive around in. And I added some events so I personalized it somewhat. I really enjoyed that.

Jim was excited enough about his car to want to port it over to another MOO where he has an account, PostModern Culture MOO (PMCMOO). He discovered that one of the needed *parent objects* for the car, generic portable room, did not exist on PMCMOO. In solving this technical problem, he relied on the help of other members of the community:

> I got excited about programming just by making cdr's car and doing those things, to where I went back to the other MOO where I had an account and made myself a vehicle there from scratch. Cause they didn't have portable rooms there. I had help, of course. I got a couple people who knew more than me to give me a hand when I got stuck. Being in the MOO enables you to be... two people can work on the same project at once. I could work on my verbs, and when I had a problem I would call somebody over, and they could just list it out and they could say "gee, you need to do this" or "you need to do that." It was just like having someone sitting behind me at my terminal critiquing what I was doing.

Just as the community provided Jim's initial motivation for learning to program, it also provided support to help him through the process. In a MUD, you are almost never alone. If you have a technical problem, there are other people there to help you. Helping others is a central activity in MUD culture; it is a basis on which strong friendships are formed. People often speak warmly of those who helped them through technical difficulties.

Jim was inspired enough by his programming experiences to want to share them with teachers at his children's school:

> I plan on bringing high school kids online. I'll be working with a high school this fall. We're just getting the Internet connection. What I'm trying to do is get the teachers interested. I mean they're all interested in Internet because of all the research and all the facts that can be gleaned from there. But I also want to introduce them to the MOO just because it gets people excited about programming.

"This has inspired me," says Jim. He explains that he has just enrolled in a programming course at his local university's extension school.

For Jim, the online community provides:

- Role models,
- Situated, ubiquitous project models,
- Emotional support to overcome technophobia,
- Technical support, and
- An appreciative audience for his work

Each of these factors contributes to the individual's motivation to learn and ability to learn. In the rest of this chapter, I'll discuss each of these factors, and then conclude with a detailed analysis of an extended interaction among three children.

5.3 "Did You Really Make This?": The Importance of Role Models

Jim didn't immediately see himself as the sort of person who could learn to program a computer. He is, after all, "just a building contractor." Many assume that programming a computer is a difficult activity that should be undertaken only by the technically-educated elite; it's not the province of mere building contractors—or humanities majors, or women. Technology increasingly surrounds our everyday lives, but most people can't imagine themselves having meaningful control over it.

For girls and women, the problem is compounded: they may fear success with computers as much as they fear failure. Sherry Turkle writes that "the computer becomes a personal and cultural symbol of what a woman is not" (Turkle 1986). People define their sense of self in part through their activities. The cultural connotations of "computer programmer" do not conjure up Jane Fonda or even Jane Pauley. Contrast the connotations of programmer and figure skater. By becoming a successful figure skater, a woman comes closer to an American cultural ideal of womanhood. By becoming a successful computer programmer, a woman drifts farther from that ideal.

Positive role models can help to counter this problem, for men as well as women. Jim found himself surrounded by peers who could program. He decided to give it a try himself not only because he knew those friends could help him, but because he could imagine himself being like them.

The first time Miranda (girl, age 10) tried MOOSE Crossing she stopped and asked somewhat solemnly, "May I ask you something?" "Yes," I replied. She asked, "Did you really make this?" Her face glowed when I replied that I had. In the weeks to follow, her mother tried to take her to a concert at MIT. Miranda asked if I would be there. Her mother said she didn't think so. Miranda replied that then she didn't want to go! I had clearly become a role model for Miranda. Another eleven-year-old girl recently concluded a visit to the Media Lab with the exact same question. Both girls probably already knew or strongly suspected that I had designed and programmed MOOSE Crossing, but the explicit confirmation was satisfying. If you have any doubts about whether women can program computers, or whether it's a proper thing for a young woman to do, look around: there are women and girls all around the virtual world of MOOSE Crossing. The person who designed the system is a woman. These strong female role models encourage girls to become involved.

It's worth noting that MOOSE Crossing is a multi-age community. Role models can be of any age, but are most commonly older. Younger children are motivated to learn by their desire to be like the older children, and learn by imitating them. Aaron Falbel eloquently describes an older student teaching a younger one to read in his study of the Friskolen 70 school in Denmark:

> Maria (11) and Johanne (9) are using the wood-burning set in the workshop to make Christmas presents for their friends and relatives. Maria has just completed a small, wooden tic-tac-toe game for a cousin of hers in Sweden. The X's and O's are made out of wooden pieces that fit neatly into a finely crafted playing board. She is etching an inscription to the recipient of the gift on the back of the playing board when little Clara (6), who just started at the school at the beginning of the month, wanders by and is drawn into the scene. She watches Maria etch a floral design under her inscription and is fascinated by the strangeness of this smoldering writing instrument. Maria notices the presence of Clara and displays for her the finished product. Clara turns over the board and feels the grooves of the inscription with her fingers. Maria asks her, "Can you read?" Clara shyly shakes her head no. Maria smiles and says "Come," and she motions for Clara to come sit on her lap. Then, very slowly and sweetly, Maria sounds out the words as Clara guides her fingers over the dark-brown letters. Clara is totally absorbed: her face conveys an expression of rapt concentration, her mouth partly open, and her cheek leaning against Maria's arm. The entire episode lasts not much more than a minute. It is so effortless, natural, and unself-conscious that to call it "peer tutoring" would be to debase the beauty of the situation. (Falbel 1989)

At Friskolen 70, students of all ages learn together in one open classroom. The age segregation typical in the school systems of most cultures is often counterproductive.

On MOOSE Crossing, two thirteen-year-olds, Jack and Rachael, have become particularly strong role models for younger children. They are both accomplished programmers, and their imaginative creations are often catalysts for social activities. In particular, Rachael has founded a sub-community she calls "The Gathering" for medieval role playing. Other children are often heard to comment that they wish they could type as fast as Rachael, so they could keep up in the conversation. It's evident that many of them want to be like Rachael in a variety of ways. This benefits the younger children, and also supplies Rachael with some strong positive reinforcement. The community provides members with a ready supply of positive role models.

5.4 Situated, Ubiquitous Project Models

Role models can be a key source of motivation for learning. Jim wanted to be one of the people who has made something, like Neil and cdr. He wanted to be like them, which inspired him to want to take on a new intellectual challenge—and later to bring that same challenge to students at his children's school. Clara wanted to be like Maria, so she developed an interest in learning to read. Jim and Clara both also took interest in the intriguing artifacts they saw around them. The novelty of cdr's "tapes" and Maria's wood carving contributed to their motivation to learn. It's important to note that these examples were not simply unattributed entries in a library. It wasn't just any old wood carving; it was *Maria's* wood carving. The relationship between Clara and Maria was part of what made Maria's wood carving a compelling model for Clara to learn from. On MOOSE Crossing, Mouse (girl, age 8-9) wanted to create Charlie the wiggly caterpillar because she wanted to make something just like her older sister Miranda's wiggly snake. The object was interesting in relationship to her admiration for its owner. Role models and project models are tightly linked. People establish their identity within the community (and their desirability as a role model) in part through the objects they've created. Conversely, objects are considered intriguing in part by virtue of who created them. Examples are situated in a social context.

It's important that these sample projects are not isolated off in a library—they are ubiquitous. Most programming languages and scripting tools come with a separate folder or directory of examples. To see the examples, you must deliberately seek them out. On MOOSE Crossing, every object in the system can function as an example. In this respect, MOOSE is rather like the World Wide Web. You can usually view the HTML source for any document. When you want to know how to do something yourself, you can remember something you saw which uses that technique, and look at the HTML source code. The benefit comes not from the large number of examples (though this doesn't hurt), but from the fact that as you are going about your daily business using the Web or interacting on MOOSE Crossing, you are developing an ever richer vocabulary of models you can refer back to.

The community of MOOSE Crossing is of course multiple orders of magnitude smaller than that of the World Wide Web. It's akin to comparing the impact of new architectural forms in a small town and a big city. An innovative creation in a small town will be noticed by a high percentage of the population, and is likely to have a strong influence on the town in the future. An innovative creation in a city might be seen by more people total, but by a lower percentage of the population, and is likely to have a less discernible impact on future designs in the area.

Since the set of examples to learn from is the set of all things everyone in the community has ever made, those examples reflect the interests of the community. That set of examples is not static; it grows and evolves as the community grows. A single, centralized author of a set of examples could never reflect the community's interests and needs as well as the decentralized process of using everyone's creations as examples to learn from.

A particularly good example of sample projects that are ubiquitous and strongly associated with their creators are people's pets. On MOOSE Crossing, it's possible to make an object follow its owner around through the virtual world. This seemingly minor feature has proven to be unexpectedly important. Recall the conversation between Rachael and Storm referred to in Chapter 4. Rachael (the experienced user) is showing Storm (the new user) around the virtual world. As they are exploring, Rachael is being followed by her pet pig Rally and her pet cat Clover. This conversation ensues:

```
Rachael has arrived.
Rally says, 'Hello Clover'
Rally arrives, following Rachael.
Clover arrives, following Rachael.
Rachael says, 'I suggest n'
```

[Rachael is suggesting they continue their explorations by going north. "n" is usually allowed as an abbreviation for north.]

```
Storm says, 'how do you make animals?'
Rachael says, 'Well, it depends if you want a new type of animal or
   one that already exists.'
Storm says, 'new type'
Storm says, 'i'd like an animal to follow me around'
```

Experienced users are almost all being followed by a pet (or several pets). New members immediately encounter these pets, and usually decide they want one too. Pets that follow their owners are a compelling example for this particular community. The more broadly applicable design principle is the value of sample projects that are constantly visible, immediately associated with their creators. This relatively subtle design decision has catalyzed the process of engaging new users in making things. Compelling sample projects made by peers enhance motivation for learning to make things. Making those sample projects more visible (people's pets are everywhere all the time) creates additional opportunities for this positive effect. A pet that follows you around is also very prominently associated with you, which leverages on the benefits of linking role models and project models.

5.5 Emotional and Technical Support

One school of research in technology and education aims to apply artificial intelligence techniques to give the learner support. Typically the system makes a model of what the learner understands, and tries to anticipate and correct for common misconceptions. Existing systems have had limited success in this modeling process. However, even if the modeling were much more successful, they would still suffer from a deeper flaw: they fail to account for the fundamentally social nature of learning. Asking for help, receiving help, and giving help are all social acts which help to build networks of relationships. Help is not merely information. Getting help from the computer is not the same as getting it from a person. The relationship between the tutor and the tutee is an essential component of the process.

Peer tutoring provides a great deal of positive reinforcement for both the tutor and the tutee. After Rachael (girl, age 13) helps Cindy (girl, age 11) to finish her animal shed and posts an advertisement about it for her, they have this conversation:

```
Cindy says, 'Thanks so much...what can I do to hellp you?
Cindy says, 'You help me too much...'
Rachael says, 'well...let me send it, and then we can talk!'
Cindy says, 'K. Thanks again'
Rachael says, 'oh, no I don't. It's all in the line of duty!'
Rachael says, 'I was and am helped...now I help others!'
Cindy grins
Cindy says, 'You certainly do'
```

It's true that Cindy needed technical assistance and received it, but this unusually self-reflective snippet of conversation reveals some of the richness of the social context in which giving and receiving help takes place. Exchanging help is part of the social bond between Cindy and Rachael. Cindy knows not only that she can get the answer if she has difficulty, but that she can get the answer from a caring friend who enjoys helping her. The emotional support Rachael provides is inseparable from the technical support. Rachael has received positive reinforcement for both her generosity and her technical skill. When explicitly asked what she wants in return, she suggests that they just spend some time together. Cindy is more than happy to do so. The girls talk for 40 minutes afterwards, until it's time to go have dinner. They talk about their ideas for MOOSE Crossing projects, about school, about their friends. They exchange secret nicknames that only their closest friends use. Receiving help from someone you would tell your secret nickname is clearly very different from receiving help from a computer program or a schoolteacher.

Most of the time in most schools, adults teach and children learn. Learning from peers is refreshing because it equalizes the power relationships. In fact, sometimes those relationships are reversed—adults often learn from kids. The first time Steven (man, age 26) connected to MOOSE Crossing, Pete (boy, age 11) immediately teleported over to say hi and offer help:

```
Pete says, 'hi!'
Steven says, 'hi Pete'
Steven says, 'this is my first time here.'
Pete says, 'okay! want help'
Steven says, 'sure. sound great!'
Pete says, 'do you want to make something?'
Steven says, 'okay'
Pete says, 'wait a minute'
Pete drops MOOSE.
Steven says, 'alright'
 Pete says, 'type 'pet moose''
```

[Steven types: pet moose]

```
Steven pets the moose.
MOOSE crosses the crossing forth.
Steven says, 'i like it!'
Pete says, 'try typing 'tel north main street'.'
```

"Tel" is short for "teleport." They both follow Pete's suggestion, and rematerialize at North Main Street, where they see this description:

```
You're in a quiet residential neighborhood of Our Town. The road here
ends in a circle surrounded by houses of all different colors and
sizes. To make a home for yourself here type 'build'.
Obvious exits: ..down.........Magic Subway Station
               ..south........Main Street

Steven says, 'that was fun'
Pete says, 'what do you think?'
Steven says, 'how do i build a room?'
Pete says, 'type build'
Steven says, 'okay'
```

Steven types "build" and sees this message:

```
A team of mooses and barn owls walk and fly in, carrying saws,
electric drills, and hammers.  They hammer and saw and bang so fast
you can hardly see them!  In seconds, you have a new house.  They
attach a sign with your name over the door.
Exit from North Main Street (#208) to Steven's Room (#1842) via
   {"Steven"} created with id #1815.
Exit from Steven's Room (#1842) to North Main Street (#208) via
   {"out"} created with id #1843.
Your home has been set to this room.
You or anyone else may enter this room by simply typing your name
   here.  You may leave that room by typing 'out'.
An owl riding on a moose's antelers says 'Welcome to the
   neighborhood!'  The animals pick up their tools and hurry off to
   their next job.
```

Pete sees:

```
Steven builds a new room here.

Pete says, 'type tel Steven's room'
```

They both do so. They could actually have gotten there by just typing "Steven," but teleporting works as well. They both see:

```
Steven's Room
Welcome to your new home! To decorate it, type 'describe here as "A
pink house with purple windows a green carpet and silver dots on the
ceiling."'  Unless of course you don't like pink....
Obvious exits: ..out.........North Main Street

Steven says, 'that was easy'
Pete says, 'so, what do you want to build?'
Steven says, 'how about a pet?'
Pete says, 'why not a dog?'
Steven says, 'a dog it is'
 Pete says, 'alright! first type 'create #5 named' and put a name.'
```

Steven types: `create #5 named ruff`
He sees: `You now have ruff with object number #1087 and parent generic thing (#5).`

```
Steven says, 'okay'
Pete says, 'press the pencil icon'
Steven says, 'okay. what button do i press then?'
Pete says, 'well, in object, type the name, click on script, and type
   pet.'
Steven says, 'okay'
Steven says, 'i've now got another window open'
Pete says, 'now type 'on pet this' in it, and press return.'
Steven says, 'okay'
Pete says, 'next, try typing 'emote says "arf" ' and press return'
Steven says, 'got it'
Pete says, 'now, type 'end', press save, and close that.'
From Steven: ruff says arf
```

Austina logs on, pages Steven and Pete hello individually, and then joins them.

```
Austina teleports in.
Austina says, 'hi!  Nice to meet you!'
Pete says, 'hi'
Pete says, 'try petting ruff'
Austina smiles
```

Austina types: `pet ruff`

```
ruff says arf
Austina grins.  "Nice dog!"
```

This interaction is entirely typical. Pete helps another adult, Nicole (woman, age 29), later that same day. Similarly, Hermes (boy, age 9) helps Jamal (man, age 18) not long after.

It's remarkable to watch an eleven-year-old teaching a twenty-six-year-old. When I visited with children participating in an educational MUD called MariMUSE (now called Pueblo) at the Longview School in Phoenix, Arizona in March 1994, I asked a group of children what they liked best about it. One child replied that getting to tell adults how things work was his favorite part. The reversal of normal roles is a powerful experience for many children. Such role reversals are often observed not just in MUDs but in children working with computers in general.

The role reversal is most powerful when the child and adult are in the same room. The kids on MariMUSE were particularly pleased by being able to help their teachers with technical problems. Interacting with others online, the role reversal seems less strange, and in fact may not even be noticed. That is what happened in this particular instance for Pete. He was not even aware that Steven is an adult. When asked about it later via MOOmail (email internal to MOOSE Crossing), he replied:

```
Hi! Anyway, no I did'nt know Steven was an adult. It seemed like
talking to anyone else on moose crossing. well, bye!
```

Pete hadn't given the matter any thought one way or the other. Steven, on the other hand, thought about it a great deal, but came to no conclusion. It's easy to find out whether someone on MOOSE Crossing is a kid or an adult, but neither Steven nor Pete was aware of this fact at the time, and neither chose to ask the other. When asked via email how old he thought Pete was, Steven replied:

```
You know, I was curious about that very fact. I really have no idea!
At times I thought he was a grad student at the Media Lab - the way
he instructed me on how to create my room and my pet, the way he
checked my progress. But at other times, I thought he might just be 8
or 10 years old - when I had more specific programming questions, he
didn't seem to understand.

Another strange issue was the duration of time (long pauses) between
Pete's replies to my questions. If he was a grad student, maybe he
was working on something else at the same time? If he was a kid, was
he working on homework? Having trouble typing? Building objects?
Maybe the server was slow?

Recently I've been trying to learn as much as I can about programming
in MooseCrossings (hanging out in the treehouse) so *I* can actually
help the kids (instead of the other way around).

And one final note, I'm coming to think that the most important
aspects of MooseCrossings are the code of conduct and the
teacher/student relationship that develops between players.
```

A cartoon that appeared in the New Yorker in the early 1990s showed a dog sitting at a keyboard, and bore the caption "on the Internet, no one knows you're a dog." It became an instant classic, and an instant cliché. It's also true that on the Internet, no one may guess that you're eleven. This is particularly beneficial for children whose intellectual level deviates significantly from the level expected for their age. You don't appear slow—people just guess that you're younger. You don't appear frighteningly precocious-—you just seem older, and have the opportunity to socialize with older children on a more equal basis than is possible face to face. This is the case for Hermes (boy, age 9), a boy who is so exceptionally bright that people often react to him oddly. As a fourth grader, he is already taking math classes at a local university. His father Howard (who is also an active MOOSE Crossing member) comments that:

```
The nice thing about MC is that it's an outlet for his social
interests as well as his writing skills (and his logical/mathematical
ability as he gets into programming). Plus it's a perfect father-son
activity, especially since he's been much more interested lately in
entering into "my" world -- the Net, programming, writing, etc....
Even at a gifted school he has learned to be wary of adults'
reactions to him. There's often a strong undercurrent of
discomfort/disapproval when he talks or acts above his age-range, and
he is old enough to sense it and to start looking for places &
activities where age is less of an issue.
```

On the Net, Hermes can interact with others more free of prejudice than he can in real life. It's worth noting that the invisibility of age online causes problems as well as bringing benefits—on the first MUD Hermes tried before finding MOOSE Crossing, he was almost immediately propositioned. The MOOSE Crossing Code of Conduct strives to remedy that problem. Howard was delighted to find a place for Hermes that is safer.

Returning to Pete's encounter with Steve, two other things are noteworthy. First, the entire interaction—from the time Pete greeted Steven through the time Steven had built a room, made a dog, and programmed his first script—took a total of eighteen minutes. Second, this was only Pete's second day on MOOSE Crossing. He had just learned everything he taught Steven the previous day. Sometimes the best teachers are not experts, but learners only one step ahead of you who are excited about sharing what they themselves just learned.

In addition to learning from people who just learned the same thing recently, members of MOOSE Crossing also seek advice from locally recognized experts. Generally recognized experts include myself, MIT undergraduate Austina Vainius (who has been working on the MOOSE Crossing project through MIT's Undergraduate Research Opportunities Program or "UROP"), and Rachael. It's interesting to note that the ranks of the recognized experts on MOOSE Crossing has come to include one of the older children; it is not limited to adults. Experts are consulted whenever a difficult question arises. Support from experts is complementary to support from peers. Some children are reluctant to ask an expert for help. Once when Rufus (boy, age 12) didn't know the answer to a question Christopher (boy, age 7) asked him, Rufus asked Christopher, "maybe as in I'l ask the next ranger I see for some help or no way?" Unlike Pete who was unaware and uninterested in whether someone is an adult, Rufus clearly notices this fact and treats adults and peers differently. He will rarely ask for help from an adult, and sometimes won't accept it even when it is offered. For these children, the availability of peer support is particularly beneficial. Other children are eager to interact with the experts as much as possible. Still others like Pete are blissfully unaware of who are experts and who are not. A child's attitude towards authority figures is a deeply personal, psychological issue. Having a diversity of sources of help on MOOSE Crossing helps to meet the needs of people with varied personal styles.

Liza (girl, age 10-11) has a more balanced view. Liza is one of the children who comes to the Media Lab each week for the MOOSE Crossing after-school program. I interviewed her on video tape, and asked:

> Amy: How is it different getting help from another kid versus from an adult?
>
> Liza: It depends on who the kid is and who the adult is.... It's not very different.... Some adults just do all the work for you. Some adults let you do the work. Some kids do all the work for you. Some kids just tell you and leave. Some kids stay around and help you more and socialize.
>
> Amy: Which kind do you like best?
>
> Liza: I like the socializing kind better than just telling you what to do. It's much better!

MOOSE Crossing is particularly conducive to the "staying around and socializing" sort of help that Liza prefers.

Since experts are authority figures, people are often eager to show them completed projects. The design of MOOSE Crossing deliberately attempts to reduce the dominance of adult authority—it's not a place where you need to impress the teacher. However, many kids still take pleasure in asking for approval from authority figures.

Barbara Rogoff contrasts three models of learning: adult-run (where adults transfer information to children), child-run (where children discover things on their own, with adult support only when requested), and a community of learners (where everyone is participating in shared activities, and learning is a process of transformation of participation). Newcomers to a community-of-learners approach may not immediately understand the nature of the learning taking place there. In a school using this model at which Rogoff was a participant-observer, parents (called "co-opers") volunteer three hours per week at the school. New co-opers often don't initially understand the activity going on, because they're trying to interpret the activity going on there in terms of either child-run or adult-run models of instruction. Rogoff writes that:

> In fact, one indicator of alignment with the philosophy of a community of learners in a school seems to be regarding oneself as a learner, continually. Adults who have a learning attitude find, as one co-oper reported, "one of the things that's been nice for me has been having kids, first graders, come time and say 'don't worry about it, I'm gonna show you how to do it!'" (Rogoff 1994)

The reversal of typical roles can be energizing for adults as well as kids. For Rogoff, it's an important feature of such environments that both adults and children see themselves as learners. On MOOSE Crossing, I originally anticipated that adults would volunteer time to help the children. In fact, the children more often help the adults. The interaction between Pete and Steven is typical. Most kids have more time to devote to MOOSE Crossing than adults and more genuine interest, and therefore have greater expertise. Almost all members of MOOSE Crossing of all ages have at some point taken on a teaching role, and that is an important part of the learning experience.

A wide variety of factors have come into play in this discussion of technical and emotional support, including age, attitude towards authority, and the nature of the personal relationship between the helper and the helpee. Asking for, receiving, offering, and providing help are not simply exchanges of information. They are social acts that take place in the context of networks of relationships. Community structures that reorient the typical patterns of those relationships can change people's feelings about giving and receiving help. When I asked people on MOOSE Crossing's "social-issues" mailing list how getting help is different on MOOSE Crossing versus in school, Rachael replied:

```
From:     Rachael
To:       *Social Issues
One thing I'd like to mention about helping other people here, is
that just about everyone I've met at least seems like they _want_ to
help you, and aren't just obligated. They don't seem upset about
helping you out. That's different from school, when a lot of people
feel kinda upset when they help you.
```

```
Also, it is a lot easier here. Most people (including myself) will
drop all but the most important projects or at least set up a date
when someone needs help.

Rachael
```

A community of students in a traditional classroom have a different set of social relationships than the community of MOOSE Crossing. In some cases, students may be reluctant to ask for help because they know the teacher is grading their performance. In other cases, a teacher may be burdened with having too many students and not have enough time to help each student individually. A particularly harried teacher might slip into a brusque tone of voice when demands are made on him or her. These are just stereotypes—every classroom is a unique community with its own patterns of interaction. The broader point is that typical patterns of interaction among community members affect how comfortable each participant is in asking for help or offering it. In turn, the patterns of asking for help affect other patterns of interaction of the community. On MOOSE Crossing, these have become mutually positively reinforcing. People generally receive supportive, friendly help, and then are eager to offer it to others. A friendly tone to the place makes people comfortable asking for help, and the ready availability of cheerful help makes the place seem more friendly.

5.6 An Appreciative Audience

Kids are always showing off their projects to others on MOOSE Crossing—often to everyone they meet. For example, one Saturday in October 1996, Hermes was logged on for much of the day. He worked all day on improving his "generic magician," a character class that lets you cast spells. At this time, he was using one of his alternate personas, Nick. On meeting a new member, Jamal (man, age 18), Nick/Hermes immediately showed off what his new scripts can do:

```
Jamal says, 'hi'
```
Nick types: `cc mr`
```
Nick casts a mirror spell!
Everything that touches Nick bounces back as if something was
   shooting at it!
You try to touch him but you feel a LARGE shock and your hand
   fliesbackward as quickly as your arm allows!
Nick says, 'like it?'
Nick grins
Jamal says, 'pretty cool.'
```
Nick types: `cc fb`
```
Nick casts a fireball spell!
The fireball starts a fire that lights up the room!
```

Jamal types: `look Nick`
He sees:

```
is alot like hermes exept he's a little bit taller and a little bit
stronger.He has a peice of rusty metal in his hand. It has a strong
burning aroma and every once - in - a - while it zaps out red beams
of light that singe everything they touch.
He is awake and looks alert.
Carrying:
generic magician

Nick says, 'i've got 4 more'
Nick casts a fgjhgfhjg spell!
Jamal says, 'a what spell?'
Nick says, 'mad typing'
```

Nick goes west, and Jamal sees him leave, followed by his magic harp and pet human Google. Nick pages him "come on, it's west," and Jamal follows. Having shown Jamal a few of his spells, he now wants to show Jamal another one of his projects, his magic subway system. Jamal sees this room description:

```
The Station Master's Office
The office of Hermes, the Magic Subway's Station Master.   It looks
like a bubble surrounded by coral that is floating through time and
space itself.  There is very little furniture...a bed, something that
looks like a huge blob of jello that might be a couch, and a
chandelier that has small glowing fish hanging from it.  They look
like they don't give a darn that they're hanging upsidedown...in fact
they look rather happy up there.
   Obvious exits: ..east.........Closet
                  ..down.........Magic Subway Station
You see Magic School Bus, Magic Harp, and Google here.
```

Jamal types: `look Google`

```
He sees:
A short, brown, shaggy haired humanoid.  He is very tough and strong
for his size.  He has a short-sword at his belt.  He will protect
himself or his owner with it.

Nick says, 'like my room?'
Jamal says, 'The Closet?'
```

Jamal types: `look Magic Harp`

He sees:

```
You see a brand new following object that needs a description!  Type
'help #342' for more information

Nick says, 'what about it?'
Nick says, 'no. my room is here'
Jamal says, 'The Closet's your room?'
Nick says, 'no.here is.your in it'
Jamal says, 'You're the Station Master?'
Nick says, 'yep'
```

Nick types: `become Hermes`

```
Nick freezes momentarily.  You feel a psychic wrench as if a great
   spell is being broken.  The figure before you dissolves into a
   normal human boy.  Hermes is back!
Jamal says, 'Cool. Looks like I'm meeting an important person.'
Hermes says, 'that answer your question'
```

```
Jamal says, 'yeah.'
Hermes says, 'yeah. i made this hole thing...with my dads help'
Jamal says, 'It's pretty impressive.'
Hermes says, 'wanna be a mgician?'
```
Hermes types: `oo magician`
```
One of Hermes's harp strings breaks with a loud twang!
He must have made a mistake.
Hermes hits himself on the forehead and says " D'oh!"
Hermes says, 'I meant 'magician'!'
Jamal says, 'Don't know if I'm qualified. I'm new to all of this.'
Hermes says, 'you don't have to be qualified'
Jamal says, 'okay then.'
```

The interaction began with Nick/Hermes showing off his creations to a newcomer, Jamal. Jamal's appreciation gave Hermes positive reinforcement. The encounter has now made a smooth, natural transition to a tutoring session where Hermes is helping Jamal.

```
Hermes says, 'type parents Hermes'
```

Jamal types: `parents Hermes`
He sees:
```
Hermes(#1354)   generic magician(#1448)   Generic Multiple-
Personality Character(#456)   MOOSE player class(#108)   generic
programmer(#59)  generic builder(#4)   Frand's modified player
class(#141)   generic player(#6)   Root Class(#1)
```
Hermes also looks at his own parents, and sees the same thing.

```
Jamal says, 'okay. I see a list of objects.'
```

Hermes types: `parents Jamal`
He sees:
```
Jamal(#1599)   MOOSE player class(#108)   generic programmer(#59)
generic builder(#4)   Frand's modified player class(#141)   generic
player(#6)   Root Class(#1)
```

```
Hermes says, 'now type chparent me to gm'
```

Jamal types: `chparent me to gm`
He sees:
```
You must give the name of some object.
```

```
Jamal says, 'It says I must give the name of some object.'
```
Jamal looks at Hermes' parents again.
```
Hermes says, 'oooook.type chparent me to generic magician'
```
Jamal tries this, and gets the same response.
```
Hermes says, 'or #1448'
```

Jamal types: `chparent Jamal to #1448`
He sees:
```
Rebuilding Jamal (#1599)
Done.  1 objects rebuilt.
Parent changed.
```

You can't refer to an object by its name unless it is in the same room as you or you are holding it. Jamal needed to refer to generic magician by its number. Hermes has learned something from teaching Jamal. Later in the day, he will similarly help Pete to become a magician. When he helps Pete, he will get the command right the first time.

```
Jamal says, 'done.'
Jamal checks his own parents and Hermes' parents, and sees that they're the same.
Hermes says, 'ok. type cc fb'
Jamal types: cc fb
Jamal casts a fireball spell!
The fireball starts a fire that lights up the room!
Jamal swrls his cloak and smothers the fire!
Hermes says, 'see!?'
Jamal says, 'cool!'
Jamal says, 'thanks.'
Hermes types: cc mr
Hermes casts a mirror spell!
Everything that touches Hermes bounces back as if something was
   shooting at it!
You try to touch him but you feel a LARGE shock and your hand
   fliesbackward as quickly as your arm allows!
Hermes types: cc fb at jamal
Hermes casts a fireball spell at Jamal!
The fireball covers Jamal in red and orange flames!
Hermes swirlshis cloak and smothers the fire!
Hermes types: cc bq at jamal
Hermes covers Jamal with a bright light and sends him to Mars and
   yells after him 'BE QUIET!'
he thinks hard as soon as he hears a tiny 'All right' and brings
   Jamal back to earth.
Hermes types: cc nv
Hermes casts a light spell!
A small glowing sphere appears in his hand.
he see's what he is looking for and turns the light out.
Hermes types: cc armor
Hermes casts a armor spell!
Jamal says, 'You've got a lot of spells'
Hermes says, '6 infact'
Hermes types: cc mr
Hermes casts a mirror spell!
Everything that touches Hermes bounces back as if something was
   shooting at it!
You try to touch him but you feel a LARGE shock and your hand
   fliesbackward as quickly as your arm allows!
Jamal says, 'So what does the Station Master do?'
Jamal types: cc
Jamal casts a  spell!
Hermes says, 'oh...fix the subway when it needs it'
Hermes types:  cc
Hermes casts a  spell!
Hermes says, 'neat!'
Hermes says, 'never thought of that!'
Jamal says, 'was an accident actually.'
Hermes says, 'oh.still ....it suprises people'
```

Jamal accidentally cast a spell with no name. Hermes decides this is a clever trick, and compliments him on it.

```
Hermes says, 'let's go to the subway'
Jamal says, 'okay'
```

They both go down, continuing their tour of Hermes' subway system. They see:

```
Magic Subway Station
Welcome to The Magic Subway Station!  You are deep underground, in a
great hall carved out of solid rock.  You see a regular stream of
travellers emerging from the shimmering magical entrances that line
the west side of the hall.  Most hurry across the room and vanish
into the magical exits on the opposite side, but a few stay to admire
the magnificent interior.  (Typw 'view hints' for suggestions.)

When you are ready to leave, please select your destination and step
   through the magic portal!
  Obvious exits: ..CR...........Crossroads
                 ..EA...........Emerald Apartments
                 ..HC...........Home in the Clouds
                 ..IC...........Mouse's favorite Ice Cream
                 ..NM...........North Main Street
                 ..PI...........Paradise Island
                 ..RF...........Redwood Forest
                 ..ST...........SparkyTown
                 ..TA...........Travel Agency
                 ..XX...........The Station Master's Office

Hermes says, 'type view mural'
```

Jamal types: `view mural`
He sees:

```
At the far north end of the hall, you see  a gigantic three-part
mural, showing the construction of the Subway Station.  The three
parts are called "PLANNING," "CREATION," and "COMPLETION."

Hermes says, 'or view hints'
```

Jamal types: `view hints`
Hermes types the same thing.
They see:

```
A luminous scroll dances through the air towards you and unrolls
before your eyes!  It says:  "You can view north, south, east, west,
up, down, and other things mentioned in those views.  In addition,
you particularly notice a statue in the center of the hall."  The
scroll rolls itself up with a saucy snap and dances away, humming a
spritely tune!
```

Jamal types: `view mural planning`
He sees:

```
There is nothing to view by that name!
```

Jamal types: `view "PLANNING"`
He sees:

`In the left-hand portion of the mural you see a slim boy with tousled brown hair. He is standing in a rough natural cavern, surveying its walls and ceiling by magical light. He holds a slender rowan wand in one hand, and a small singer's harp rests at his feet. The wand and the harp shimmer with a powerful magical aura. There are two more sections of the mural in the center and on the right.`

Hermes types “v n” and sees the same introductory description Jamal saw when he viewed the mural.

Jamal types: `view center`
He sees:
`In the foreground of the center part of the mural you see a slim hand holding a wand. Powerful lightning streams from the wand in a great arc. Ahead of the lightning are the rough rocky walls of a natural cavern, but where the lightning has already passed you see the walls of the present Subway Station taking shape. There are two more sections of the mural to the left and right.`

`Hermes says, 'type v planning'`

Jamal types: `view right`
`He sees:`
`In the center of the rightmost part of the mural a boy is using his wand to create great magical gates and link them to faraway locations. In the background is a shadowy older figure, using his staff to carve columns and create chandeliers and other decorations.`

Hermes types “v planning” and sees the description Jamal saw earlier.

`Jamal says, 'cool. it's you in the mural right?'`
`Hermes says, 'yeah.how'd you guess?'`
`Jamal says, 'the wand and the harp.'`

Hermes types “v creation” and sees the same description Jamal saw on the center wall. Then he types “v completion”, and sees the same description Jamal saw on the right wall.

Jamal types: `look CR`
He sees:
`A shimmering gateway of pure energy, set into the east wall of the Subway Station. It leads to the Crossroads.`

Jamal types: `look HC`
He sees:
`A shimmering gateway of pure energy, set into the east wall of the Subway Station. It leads to the Home in the Clouds.`

`Hermes says, 'type v completion'`
Jamal types “v completion” and sees the same description he saw on the right wall.
`Hermes says, 'the shadowy guy is my dad'`
`Jamal says, 'I guessed he was.'`

Hermes types: `v statue`
He sees:
`In the exact center of the Subway Station, you see a heroic statue of a large moose, standing on his hind legs. He's wearing glasses and agoofy grin, and his antlers seem a bit crooked. He appears to be`

```
staring off into the distance.  In one hand, he is holding a model
sailboat that reflects deep red sparkles.  There's a flying squirrel
sitting on the moose's shoulder, wearing goggles and a leather flying
helmet.

Hermes says, 'type v statue.wonder who they are?'
```
Jamal looks at the statue, and sees its description.
```
Jamal says, 'Is it Rocky And Bullwinkle?'
Hermes says, 'yeah!'
Hermes goes ding!ding!ding!
Jamal says, 'Why are you dinging?'
```
Hermes types: `kick google`
```
Hermes kicks Google hard in the stomach!  Google lands on his butt,
   but bounces right back up!
Google says, Whadya do dat for, ya big louse?!!
Google then pulls out his sword and threatens to hack Hermes into
   bits.
Hermes says, 'game shows do that when you get an answer rite'
Jamal says, 'I see. Thanks... Go easy on Google. Seems kind of
   feisty.'
Hermes says, 'yeah... i will'
Hermes says, 'hey! want me to put a create candy spell?'
Jamal says, 'Sure.'
Hermes says, 'on us that is'
Jamal says, 'As long as it doesn't turn us into candy.'
Hermes says, 'ok. just a sec.click on the pencil, type me in the
   browser and duble click on cast'
```

Having an audience has inspired Hermes to continue to improve his set of spells. He adds a new spell to his magician. Jamal waits a few minutes, checking help messages.

```
Hermes says, 'now type cc cc'
```
Jamal types: `cc cc`
```
Jamal snaps his fingers and your favorite kind of candy appears in
   him hand!
Your mouth waters at the sight of that candy!Yum!Yum!
Jamal says, 'That's pretty cool.'
```

Hermes is unhappy with the incorrect pronouns in the first line of the script's output. He modifies the script and tries it again. The script prints out:

```
Hermes + my pp + snaps your favorite kind of candy appears in + my pp
   + hand!"
Your mouth waters at the sight of that candy!Yum!Yum!
```

There's still an error. Now a mistake in the placement of quotation marks is causing a line of the script not to be correctly evaluated. Hermes modifies the script and tries it again. This time it works:

```
Hermes snaps his fingers and your favorite kind of candy appears in
   his hand!
Your mouth waters at the sight of that candy!Yum!Yum!
```

Here's the final version of the candy script:

```
on cast "cc"
  if context isn't me
    return
  endif
  announce_all my name + " snaps " + my pp + " fingers and your
   favorite kind of candy appears in " + my pp + " hand!"
  announce_all "Your mouth waters at the sight of that candy!  You
   take the candy and gobble it up!  Yum! Yum!"
end
```

Hermes is enjoying being the teacher:

```
Hermes says, 'yeah.hope you can follow my lead'
Hermes says, 'so you can make your own spells'
Hermes types: cc bq
Hermes casts a bq spell!
Hermes says, 'nuts!'
Hermes says, 'it doesn't work'
Jamal says, 'I think I can follow your lead. I have to go soon.   But
   I'll make a spell for next time I'm around.'
Jamal says, 'Did you create the generic magician?'
Hermes says, 'yeah. if you have any ideas for spells tell me.by mail
   or talk'
Jamal says, 'Okay.'
Jamal says, 'Thanks a lot for your help.'
Hermes says, 'don't try cc bq!'
Hermes says, 'it won't work easily'
Hermes says, 'you'ljust get a turkey leg of embarresment!'
Jamal tries to smile, but can't figure out the command.
Jamal says, 'Thanks for the tip.'
Hermes says, 'welcome'
Jamal says, 'I'll talk to you later. I have to go.'
Hermes says, 'by'
Hermes says, 'wait'
Jamal says, 'Yeah?'
Hermes says, 'i'll tell you when it's okay to do cc bq ok?'
Jamal says, 'Okay.'
Hermes says, 'by'
```

Jamal, an MIT undergraduate studying computer science (or "Course 6" in MIT lingo), was impressed with his encounter with Hermes. Asked via email about his impressions of the encounter, he replied:

> I remember meeting Hermes. The deepest impressions left on me from our conversation were his knowledge of programming in Moose Crossing and the ease with which he was able to teach me about what he'd done and what he knew. I even talked to one of my friends later that day about him. It seemed then (and still does) to me that if he continues along his present course he'll be well-prepared for Course 6 by the time he's old enough.

> From what I can tell your MOOSE Crossing MUD is doing exactly what you said it would; allowing kids (and others) to learn by doing and teach others. I think it's great.

During the course of the day, Hermes showed his spells off to three different people: Jamal, Pete, and Miranda. He showed Jamal and Pete how to change their parent object to his character class so they can cast spells too, using his spell scripts. Miranda already had her own character class, which has an answering machine which records things that you hear when you're not connected. Hermes changes his character class to inherit from hers, so people can use both at once.

Whenever anyone finishes a project on MOOSE Crossing, they almost always rush to show it off to their online friends. Positive reinforcement is always available. In fact, while I have been writing this paragraph, Mouse asked me to come see the pool she is building as part of her hotel, and Miranda showed me the program she wrote to be able to change her facial expression/mood separate from changing her description. (I often have a MOOSE Crossing session open in another window while I write.) The availability of a potentially large audience helps motivate people to create things.

Another way kids show off their creations to one another is by placing advertisements in our online newspaper, the MOOSE Herald-Examiner. Here is the advertisements section on the day of this writing:

```
*** ADVERTISEMENTS ***

JACKET -- 10/14/96
~~~!~~~!~~!A COZY JACKET TO SLEEP IN!~~~!~~~!~~~
MOUSE HAS MADE A COZY JACKET! SOUNDS WEIRD BUT IS A GREAT PLACE TO
BUILD A HOME OR A STUDY OR MABEY JUST A ROOM! JUST TEL TO THE JACKET
AND SEE WHAT YOU CAN DO ON A COZY JACKET! IF YOU HAVE ANY QUESTIONS
OR COMENTS PLEASE MAIL MOUSE OR TALK TO HER IN PERSON. THIS IS A
GREAT PLACE TO BUILD ATTICS OR OTHER PLACES WITH LITTLE NOCKS.
REMEMBER: AFTER A MOUTH AND MOUSE DOES NOT DO WELL WITH THE JACKET,
THE WHOLE THING WILL BE RECYCLED! -- Mouse

POTATO -- 10/9/96
I just made a cool potato! You can plant new potatos with it. You can
also mash it, french fry it, make potato chips, and more! When you
use those scripts, it changes the potato's appearance! You can also
fix the potato after you've cooked it. Come see it! -- Miranda

MOUTHPIECE -- 10/09/96
Now, if you want to talk to a bunch of people at once who are in
different rooms, then join mouthpiece (#1684) by typing @addfeature
#1684, and then turn it on by typing "on mp" . Then, you can talk by
typing 'shout <whatever> into mp'. Mouthpiece is public, so anyone
can join. If you want to start your own, then make one...mouthpiece
is also a generic! -- Rachael
```

```
HAVE YOU EVER BEEN ON MOOSE CROSSING WITH NOBODY TO TALK TO? --
9/30/96
Now, whenever you feel like it, you can talk to your
generic_conversation object! With this object, you can ask it yes or
no questions, and whenever you tell it something, it will answer you!
Starting with asking you how you're feeling, this objct will talk to
you for hours on end, and never have to disconnect!  -- Miranda

MOOSE MAGIC -- 9/30/96
~~~~~~~~~~~~NEW MOOSE MAGIC~~~~~~~~~~~~
Ever wish you could du extraordinary stuff, like make people float,
or shoot out fake fireballs? Impress your friends by getting a free
spellbook at the Owl's Nest. Good for magic fights at the SparkArena.
To even further these powers, get an owl, all at the Owl's Nest. You
must find the Owl's Nest at your own risk. -- Rufus

UTOPIA -- 9/23/96
Do you need a break ? well if you do come to UTOPIA the greatest
resort in the land.it has swimming,shop, and a realy cool arcad
come now to UTOPIA . this lovely resort is in red wood forest off
liza s room.(for any more info talk to liza) -- liza

ANSWERING MACHINE -- 9/23/96
I just finished a character class which is an Answering Machine! It
records what people tell you when you're asleep! For more details
mail Miranda.
P.S. Please mail me if you would like to add a room to the travel
agency! -- Miranda
```

The power of having an audience can be seen clearly in the explosion of interest in the World Wide Web in 1995-1996. People's home pages are a form of multimedia self portrait. The tools to make such self portraits (for example, paint programs) have been commonly available (to middle-class Westerners) since the mid to late 1980s. However, it was only when tools to *share* such self portraits with a large audience became commonly available that interest in making them took off (Bruckman 1995).

MOOSE objects, like home pages, help to establish an individual's identity within the community. The first interaction between two members often focuses on admiring one another's creations. With most children being followed by a small entourage of pets, it's easy to see how strongly one's creations affect one's public image. Notice that without being prompted, Jamal examined Google and the magic harp following Hermes. This is typical.

More subtly, people’s creations also affect one's self image—deciding what sort of a thing to make is partly a reflection on the question "what sort of a person am I?" Goffman notes the role that clothes play in defining one's sense of self and role within the community (Goffman 1959). Appadurai studied the same function of objects collected in people's homes (Appadurai 1986). Sherry Turkle takes this analysis a step further, noting how children use computational media to work through deeply personal issues—things you make are even more expressive of who you are than things you choose to wear or buy. Computational media present a particularly rich opportunity for expression, because they can be used to make such a diversity of meanings. The computer is a kind of Rorschach Test (Turkle 1984). Children not only express who they are but also help to shape who they are through their construction activities.

These presentations of self are motivated and shaped by the existence of an audience. Making an object on MOOSE Crossing is a fundamentally social act which exists in relation to an audience. Rachael summed it up in the quote that opens this chapter: "While programing is a lot of fun, I don't think I'd do it, if there wasn't anyone who would apprecitate it."

5.7 Local Community and Online Community

So far in this chapter I have been focusing primarily on the impact of the *online* community on the participants in MOOSE Crossing. The children's experiences are of course also affected by the local, face to face community they are a part of while they participate. The local setting varies. Most children participate from home; however, that may mean they are primarily alone, or are interacting with parents or siblings. Some children participate in classes at school. Some participate in after-school programs. The group I have studied most closely are the children who came to the after-school program I ran at the Media Lab.

On average six children (of the roughly 160 children registered as of March 1997) came to the Media Lab once per week for more than a year to participate in a MOOSE Crossing after-school program. Each session lasted roughly two hours. The exact composition of the group changed slightly over time—older children started bringing younger siblings, and one pair of siblings dropped out.[3] (The reasons for their loss of interest are not clear. MOOSE Crossing,

[3]First, just Miranda (girl, age 10-12) came. She was the first child to work with the MOOSE software. The software was modified and bugs fixed based on her comments and comments of other children to join later. Then her friend Byron (boy, age 10-11) joined. Zoro (boy, age 11-12) joined the group next. The entire time Miranda was participating, her younger sister Mouse (girl, age 8-9) had been watching quietly over her shoulder. Miranda's mother hadn't had a place to leave Mouse while Miranda was at the Media Lab. After several weeks of observing, Mouse decided she was ready to try it too. Mouse's best friend at school is Byron's younger sister Patricia (girl, age 8-9). (In fact, the older children met through the friendship of their younger siblings.) Mouse enthusiastically invited Patricia to join. Patricia never really shared Mouse's enthusiasm, and only came for a few weeks. Approximately a month after she stopped coming, her older brother Byron stopped coming as well. At around that time, Zoro's family decided to send their second child Liza (girl, age 10-11). Next, Miranda invited her best friend from school, Squirrel (girl, age 11) to join. Next, Zoro's family started sending their third child, SpaceRabbit (boy, age 9). Their youngest child Goofy (girl, age 7) joined the group towards the end of the year.

like any activity, is not for everyone.) The size of the group was limited by the number of Macintosh computers we had available for them to use.

All of the children in the after-school program had access only from the Media Lab for the first half of the year. Mid-way through, Miranda and Mouse got access from home as well. Towards the end of the year, I asked both girls to compare using MOOSE Crossing at home versus at The Media Lab. They both said that it's easier to get their questions answered when they are using MOOSE Crossing at the Media Lab than at home. It's easier to ask a question out loud than type it. Additionally, they mentioned that Austina and I always know the correct answers; other children online sometimes do not. However, Mouse mentioned that getting help from Rachael is just as good as getting help from adults—she really knows what she's doing.

One notable difference between home and classroom or after-school program use is the level of activity in the room. The weekly sessions at the Media Lab often get boisterous. One teacher using MOOSE Crossing in her classroom in California called the experience "magical chaos." While it may be easier to get help in an after-school program, Miranda commented that it's quieter when she's working at home, and she can focus more. In contrast, Mouse complained that at home "It's hard to type because Miranda is always there saying 'I want to get on! I want to get on!'" The girls share one computer at home, and the older sibling seems to win control of it more often. At the Media Lab, each girl gets her own machine. The details of the local setting can be substantially different even for two kids using the same computer in the same family setting. That local setting strongly effects a child's experiences.

My impression based on observing Miranda and Mouse over the past year and a half is that having access at home significantly enhanced their enjoyment and enhanced the benefits they got from the project. Children who can participate only once a week may lose interest in a project between weeks. When Zoro (boy, age 11-12) arrives at the Media Lab, he often asks to be reminded of what he was working on the previous week. In contrast, Miranda and Mouse often log on again immediately on arriving home and continue working on their project. When they are excited about a project, they may work on it every day.

The increased rate of progress I observed in Miranda and Mouse after they got access from home can obviously be substantially attributed to their increased time on task. However, there are also other more subtle factors:

- **Engaging with others online**

  When they are at the Media Lab, Mouse and Miranda (and the other children) talk primarily to people in the room, and have fewer conversations online. It's only when they go home that they really start to engage with other children online. Talking with other children from far-off places is good practice for written expression, may open new cultural perspectives, and helps to form a connection to the broader community of members rather than just the local community of their after-school program.

- **Teaching**

  When Miranda and Mouse are at the Media Lab, they primarily work on their own projects and leave it to the adults to help the other children. When they are at home, they often take the time to teach others (as we shall see Miranda do in the next section). Teaching is often as much an educational experience for the teacher as the learner.

- **Invisibility of some social factors**

  Another issue concerns the invisibility of a variety of social factors online. The invisibility of age was discussed in Section 5.5. When Mouse is at the Media Lab, she is usually the youngest child present, and gets treated as such by her peers. Online, other children often guess that she is much older, because of her greater experience with the system. (She told me with pride that Pete (boy, age 11) thought that Mouse was 12). Other factors in addition to age that can sometimes be beneficially invisible online include gender, physical attractiveness, popularity, social class, and having many kinds of handicaps.

- **Spontaneous versus scheduled**

  Finally, the home and after-school program environments differ in how they are scheduled. At home the girls connect only when they are in the mood to do so, and therefore well disposed to benefit from it. The group at the Media Lab meets at a regularly scheduled time. Sessions can be less productive if the child happens at that time to be tired, hungry, sick, or just not in the mood.

Benefits of having access in a more formal group setting (i.e. classroom or after-school program) include:

- **Ease of communication out loud versus in text**

  For most people, speaking is faster and requires less effort than typing. It's easier to ask a question out loud. Putting a question into written words usually requires more reflection. This reflection in some cases may help the questioner to understand the question better; however, the effort required can often be prohibitive, particularly for younger children. Other channels of communication such as body language and intonation can also be useful in getting across a meaning.

- **Someone helping you can see your screen**

  It's much easier to help someone if you can see exactly what they're seeing. It's possible in theory to solve this problem in software by letting one person have a view of exactly what another person's screen looks like. There was not enough time to implement this feature for the MOOSE Crossing project. Consequently, it's easier to help someone who is in the same room.

- **Presence of experts**

  If the community is large enough, there will always be "expert" help available online for difficult questions. MOOSE Crossing is still small enough that this is not always the case. In many group programs organized to date, the adult leaders of those programs have not had the time to develop even a basic understanding of the software. This is understandable—most teachers are overworked. Children in such programs have much more difficulty compared to those in programs where experts are present. Over time, some of the children may develop significant expertise, and be able to step into the expert role. One teacher writes that "We are learning together and I am learning from them as much or more than they are than they are learning from me. The bolder children have a sixth sense about stuff like this and forge ahead. They are becoming all of our tutors." They started out with little local expertise, but developed more over time. Having experts present is a significant advantage.

- **Scheduled vs. spontaneous**

  The advantages of spontaneous time dedicated to an activity were discussed above. Scheduled time also has significant advantages. Time in a formal program has been set aside for that purpose, and is usually not interrupted. Other activities are not competing for the child's attention during that time. Setting aside a regular time for an activity also means that a child is more likely to finish the difficult or boring part of a project. It's natural to set aside a project for a while when you get tired of it, or get mired in a difficult or boring part of it. While a child may never return to a project stuck in such a state if he or she is working on it spontaneously from home, he or she is likely to return to it during the next of a set of regularly scheduled meetings.

A combination of spontaneous and scheduled time has significant advantages. In most of the classrooms in which MOOSE Crossing is currently being used as an organized activity, the children have access once or twice per week during a regularly scheduled time. However, in one notable school, the computers are located in the classroom, and children may use them during breaks in the school day. The classroom has one computer for every two to three children. To save space, the computers are recessed into desks—the surface of the desk is Plexiglas, and the monitor is underneath, at an angle. Many of the children chose to use MOOSE before school starts, during recess, and at other free periods. While not all children take advantage of this opportunity, those that do have become local experts. During the class' regularly scheduled sessions, they often help the other children. The class has been able to accomplish much more than other classes. It's not just a matter of increased time on task, but also of giving students the opportunity to chose to become more involved, and cultivate a genuine, self-motivated interest.

The students in this class see the computer as a tool, and use it fluently and naturally for many tasks. To draw a common analogy, pencils wouldn't have had much of an impact on education if students were scheduled to go to the pencil room for an hour once per week. To be effective, tools need to be plentiful and always available—available to use in organized activities, and during free time.

Interaction in an organized program and interaction during free time are complementary. They each have different benefits, and work particularly well together. Similarly, interaction face to face and interaction over the net are also complementary. This is true in a broader sense of online communities in general. My best relationships facilitated by MediaMOO are with people who I see face to face once or twice a year at conferences (scheduled, face to face), and keep in regular contact with on MediaMOO (unscheduled, over the net). Face to face contact is richer, but online contact is easier to maintain on a regular basis.

A related issue is the relationship between geographically local and remote community. Alan Shaw's MUSIC project uses a computer network to help improve communication in an inner city neighborhood (Shaw 1994). An online community can help to cement relationships within a geographically local community. Children in MOOSE Crossing programs benefit both from increased interaction and collaboration with their local classmates, and from interactions with children in other cities and countries who they would not otherwise have met. Local community and virtual community are not in opposition, but rather for most applications work best together.

5.8 An Extended Example: Lady's First Script

It's worth tracing issues of community support through another extended example: Lady's first script. Lady (girl, age 11) first connected to MOOSE Crossing on a Friday afternoon. Austina helped her to make a dog named Cutey, by making an object which inherits from generic smart dog. Lady returned the next Monday and explored a bit, trying to strike up a conversation with various people. Finally, she joins Miranda, who is in a room called "lab" owned by Jack (boy, age 13). Their conversation is presented here unedited. It took place over a period of 48 minutes.

Jack's room looks like this:

```
lab
You are in a hidden lab.  You don't see any exits, but you doo see a
beam in the back of the room.  Strange robots and potions are on
desks, and non compiled scripts are floating all over!
You see fluffey here.
```

Here's how the conversation begins:

Lady says, 'Hello'
Jack says, 'i made a secret passage :)'
Lady says, 'You did, Wow!'
Miranda says, 'Cool!'
Miranda says, 'Where'
Lady says, 'Yeah, where?'
Miranda smiles
Jack says, 'if you are in my room, just type beam to get here if you are here, type beam to get to Jacks room.'
Jack says, 'it is a script.'
Lady says, 'Cool, how did you do that?'
Miranda says, 'How old are you Lady? I don't think I've met you. Sorry I just left after you joined me.'
Lady says, 'I'm new, I'm 11'
Jack says, 'this is the script: on beam'
[They all type 'beam' to get to Jack's secret room, and 'beam' again to get back.]
Miranda says, 'Hello again'
:)
[The smiley face was announced by Miranda. Everyone in the room sees it without being told who generated it. Technically, this is against the code of conduct all members agree to on joining, but most people don't mind if it's done in moderation in good taste.]
Jack says, 'hi :) sorry about that'
[Jack left to try out the passage. He's apologizing for disappearing suddenly. The others followed him and understood what was going on, so no apology was really necessary.]
Lady says, 'hello again to you too'
my personal favorite is :b [announced by Miranda]
cute huh? [announced by Miranda]
Lady says, 'what is :b'
or :d [Also announced by Miranda. No one ever answers Lady's question. Those are emoticons of someone sticking their tongue out.]
Jack says, 'you can see the script by clicking on the pencal and typing HERE on the box you get when you click on the pencil. Then you click on script, and type beam and hit OK.'
Miranda says, 'ok'
Lady says, 'OK'
Jack says, 'yes :b is cool'
Lady says, 'Wow, but how do you make a script?'

Lady and Miranda were just wandering around saying hello to other kids. Jack uses their arrival as an opportunity to show off his new creation, his beam script. He receives immediate positive feedback from his peers. Lady is impressed. Jack's achievement inspires her to want to learn how to make her own script.

```
Miranda says, 'Cool script! It makes a lot of sense.  Did you make it
   yourself? By the way, I'm the one who announced :b'
Miranda says, 'Well do you have any creatures yet?'
Miranda says, 'Lady?'
Lady says, 'yes'
Jack says, 'yes i made it myself :) lets teach Lady some scripting!'
Miranda says, 'If you want to make a script, you go to the pencal,
   and type your objects name'
Lady says, 'what's the pencal?'
Miranda says, 'But first you should know how to make scripts'
Miranda says, 'So don't do what I just said yet.'
Lady says, 'ok'
Miranda says, 'Why don't you look at the help on scipting?'
Lady says, 'how?'
Jack says, 'i have an idea'
Lady says, 'what is it?'
Miranda says, 'You go to MOOSE, on the top of your screen, (next to
   size and windows) and select 'help''
Jack says, 'how about i walk you through making a magic gem!'
Miranda says, 'I'll help!'
Lady says, 'ok'
```

Jack and Miranda approach the task of teaching Lady how to program enthusiastically. They're teaching technique is not, however, ideal. Instead of asking Lady what she wants to make, Jack suggests a project for her: a magic gem. Throughout the conversation, Jack will tell Lady exactly what to type rather than giving her examples and general principles. Miranda, who has tutored others before, is better at explaining the underlying ideas. Still, a trained teacher could do much better. Peer tutoring is not a panacea. Despite the imperfect nature of the tutoring she's receiving, Lady learns how to script well enough to create a similar script on her own the next day. The children are also developing a friendship as they are learning together—and it's not just Lady who is learning. Jack and Miranda are developing a better grasp of the material they are teaching, and are also learning something about teaching itself.

```
Jack says, 'OK. first, create $thing called gem.'
[Lady types "create gem" which doesn't work.]
Miranda whispers to Jack, 'hi'
Jack whispers to Miranda, 'hi'
```
[Whispering lets you communicate with a specific person without everyone else in the room hearing.]
```
Miranda whispers to Jack, 'What's the gem going to do?'
Lady says, 'how'
Jack says, 'type this: CREATE $thing called gem'
```
[Miranda types what Jack was proposing to check it, and then immediately recycles the gem she has made.]
```
Miranda opens a hidden trash chute.
Jack whispers to Miranda, 'I don't know yet!'
Lady says, 'Oh, cool'
Miranda whispers to Jack, 'how about it sparkles when someone touches
   it'
Jack says, 'good!'
```

```
Jack whisper to Miranda, 'cool!'
```

Team-teaching Lady creates a fun, almost conspiratorial bond between Miranda and Jack.

An interesting problem which arises when trying to teach programming is how to tell people what to type literally and what to fill in with your own content. This is similar to the classic slapstick routine in which the court clerk says, "Repeat after me. I, state your name." The witness replies, "I, state your name." The MOOSE Crossing help system puts in angle brackets things that should be filled in. Jack invents his own solution to this problem by putting words to type literally in capitals and things to fill in in small letters.

```
Lady says, 'So now what do I do?'
Jack says, 'now type DESCRIBE GEM AS "whatever you want!"'
Jack says, 'you do not need to caps it.'
[Lady types: "describe gem as a beutiful white crystal, gleaming in the
   light".  She next looks at the gem to see that it worked.]
Jack whispers to Miranda, 'and we can introduce properities to her by
   making it change color when someone types CHANGE GEM'
Lady says, 'ok, I have the description set'
Miranda says, 'Now, lets say, you want to make it so that when
   someone types 'touch gem' it sparkles'
Miranda says, 'And when you type 'change gem' it changes color.'
Jack grins
Lady says, 'how do you do that, Miranda?'
Jack says, 'drop the gem so we can see it'
Miranda says, 'ok, first you'd have to make a script on it'
[Lady drops the gem.]
Lady drops gem.
fluffey sniffs gem curiously.
Jack says, 'hmm'
Lady says, 'how do you make a script'
Miranda says, 'A script always begins with the word 'on''
Jack says, 'OK, click on the pencil at the top of the screen and type
   in gem. Then check the script circle and type in TOUCH.'
Jack says, 'then hit OK'
Miranda says, 'Tell us when your done'
```

The system design here is less than perfect—Miranda and Jack can't see Lady's screen. A special "last_commands" feature lets you see what someone else has typed (if they allow you), but can not show you the state of another person's windowing environment. The ideal setting for MOOSE Crossing is a combination of spontaneous use and scheduled use in an organized program. Face to face interaction with kids in the same room is complementary to network interaction with kids in far-off places. A local collaborator would be able to see Lady's screen and help her through the basics of creating a script. Luckily, the problem has limited scope—once someone has learned the basics of how to create a script (all of which Lady will learn in this conversation), then he or she can easily receive help on more-advanced topics from others.

```
Jack whispers to Miranda, 'this is cool, i have never tutored anyone
   before!'
Miranda whispers to Jack, 'I have, sometimes it gets frustrating'
```

[Lady fills in the script editor box with the words "`the gem sparkles`" and clicks Save. This does not create a script.]

```
Lady says, 'I think I'm done'
Miranda says, 'Ok, now do you have a script bow that says 'gem:touch'
   on the top?'
Jack says, 'do you see a large text window?'
Miranda says, 'A meant a script box'
Jack says, 'what do you see?'
Lady says, 'I see a picture of a moose, a pencil, a envelope, and a
   question mark'
Jack says, 'OK, you are still in the main window.'
Lady says, 'so what do I do'
Miranda says, 'Go to 'windows' at the top of your screen, and select
   'gem:touch' if it's there'
Lady says, 'ok'
Lady says, 'it's not there!'
Miranda says, 'Then you haven't created the script yet!'
Jack says, 'click on the pencil and you will get a little window. it
   says edit code at the top. type gem in the box and click on
   script. then hit OK. it will say there is not scripts by that name
   ob the gem. click on the button that says "CREATE AS A SCRIPT"'
Lady says, 'but it does have that script there!'
Miranda pages Amy with 'Jack and I are teaching Lady to program!
```

[She receives a message back that Amy has been idle for over an hour.]

```
Jack says, 'having two people tutor one person is a good idea,
   because as one person is typing, the other person is giving
   instructions :)'
Miranda pages Austina with 'Jack and I are teaching Lady to program!
```

Miranda is taking on a traditionally adult role in teaching Lady to program, and she's eager and proud to show this off to the two adult authority figures logged on, Austina and me. Austina and I are trying to get some work done on improving the system design, and on this afternoon have limited time to come by and hang out with the kids, but we do each stop by briefly to appreciate the work completed.

```
Lady says, 'I don't get this'
Jack says, 'OK, if it does have a script, you should be in, right?'
Miranda says, 'What did you do?'
Lady says, 'I think so'
Jack says, 'yes, describe every move you made:)'
Miranda says, 'Sounds good to me!'
Lady says, 'I went to the pencil and clicked, then I typed gem, I
   clicked on script and clicked "OK" after that it took me to the
   script I made.'
Lady says, 'what did I do wrong?'
Jack says, 'good, what does the script you made say?'
Jack says, 'it should not say anything yet...'
Miranda says, 'Ohhh, so now go to 'windows' and click on your
   script!'
Lady says, 'it says the gem sparkles'
Jack says, 'delete that, you have to put it in a diffrent way.'
Miranda whispers to Jack, 'Here comes the confusing part!'
```

```
Jack whispers, 'as if that first part was not confusing enough!'
Lady says, 'I did type the first line ON touch this.'
 Lady says, 'Do I need a second line with more commands?'
Miranda says, 'That is telling the computer, that when you type
   'touch gem' it should do the following.'
Miranda says, 'Do you get it?'
Lady says, 'yes.'
Jack says, 'yes, you need a second line. On the second line type
   ANNOUNCE "the gem sparkles" or type ANNOUNCE "whatever you want"'
Miranda says, 'Ok, now we're going to type what you want the gem to
   do'
Miranda says, 'That will tell the computer that after someone types
   'touch gem'  it should announce <whatever>'
```

Miranda has tutored others before and is more skilled than Jack at explaining the reasons for things rather than just telling Lady what to do.

```
Lady says, 'ok I've done that'
Austina pages Miranda, 'great! can I come over for a bit?'
Miranda says, 'OK, now, you can type 'end' to end your script. you're
   done!'
Jack says, 'on the third line, type end and hit OK. lets test it out,
   but add more lines to the script to make it better.'
[Lady recompiles the script:
  on touch this
    announce the gem sparkles
  end
It compiles successfully.]
Miranda pages Austina, 'Come on over!'
Jack says, 'er...add more lines to the script later. for now type
   end.'
Austina teleports in.
fluffey sniffs Austina curiously.
Fluffy arrives, following Austina.
fluffey sniffs Fluffy curiously.
[Somewhat confusingly, Austina has a dog named Fluffy and Jack has a dog named fluffey!]
Lady says, 'yes, it is saved. Let's try it out now.'
Jack says, 'hi'
[Jack types 'touch gem'.]
the gem sparkles
[Miranda types 'touch gem'.]
the gem sparkles
Miranda says, 'Lady, did you type 'touch gem'?'
Jack says, 'i did!'
Miranda says, 'Hi!'
Jack grins
Lady says, 'Thanks, Jack and Miranda! I have made my first script.'
Austina applauds loudly!!!
Lady says, 'Yes I did'
[Austina types 'touch gem'.]
the gem sparkles
Austina says, 'yay!'
Lady says, 'now Gem works!'
```

Lady gets immediate positive feedback on her project from two peers and one adult. Her teachers are also her audience.

Miranda says, 'You could always add something like telling the person something.'
Jack says, 'but we are not done yet... > :)'
Miranda says, 'To do that you could type in 'tell context <whatever you want>''
Lady says, 'ok.'
Jack says, 'that will tell only the person who touched it somthing, instead of the whole room.'
[Lady adds a line to the script:

```
on touch this
  announce the gem sparkles
  tell context "I am Gem, a smart stone. I am shinny and colorful"
end
```

The script compiles successfully.]
[Lady types 'touch gem'.]
[Everyone sees:] the gem sparkles
[Lady sees:] I am Gem, a smart stone. I am shinny and colorful
Amy pages Miranda, 'cool! Good luck!'
Miranda pages Amy, 'want to join us?'
Jack says, 'Lady, go back to the script window. lets add more.'
Amy pages Miranda, 'sure, but just for a minute'
Lady says, 'ok'
Miranda says, 'Cool.'
Amy teleports in.
Fluffy sniffs Amy curiously.
fluffey sniffs Amy curiously.
Pumpernickel arrives, following Amy.
Fluffy sniffs Pumpernickel curiously.
fluffey sniffs Pumpernickel curiously.
Amy says, 'hi there!'
Miranda says, 'Hi,'
Austina says, 'hi!'
Jack says, 'hi!'
Amy says, 'nice lab, Jack. My kinda place!'
Lady says, 'Hi'
Miranda says, 'Look at Lady's first script!'
Jack grins
Amy says, 'hiya Lady! Nice to meet you!'
Miranda says, 'type 'touch gem''
[Amy types 'touch gem'.]
[Everyone sees:] the gem sparkles
[Amy sees:] I am Gem, a smart stone. I am shinny and colorful
Austina says, 'well, guys, it's been cool...   Congrats on your first script Lady! You're doing great!'
Austina says, 'talk to you all later!'
Austina waves
Jack says, 'bye!'
Amy says, 'very nice!'
Austina goes home.
Fluffy follows after Austina.
Miranda says, 'I'm terribly sorry, I have to go have dinner now, bye!'
Amy says, 'see ya!'
Lady says, 'Jack, you wnt to suggest some to add on my script?'
Miranda has disconnected.
[Amy looks at the gem.]

```
Amy ooohs at the gem.  "Very pretty"
Jack says, 'say OK...'
Jack says, 'er...oK'
Amy hehs
Jack says, 'help dog2'
Jack says, 'aagggh'
```

[Jack now types 'help dog2']
[Lady now looks at 'help dog2' as well.]

```
Amy says, 'great start, Lady!'
Lady says, 'Thanks Jack. I will check the help tomorrow.'
```

[Jack presumably meant to just look at the second part of the dog tutorial for ideas on how to help Lady, but by accident he said it instead of doing it. Lady interprets this as a suggestion that she should read that help message.]

```
Lady says, 'Bye, everyone, I have to go home now.'
Amy waves
Amy goes home.
Pumpernickel follows after Amy.
Jack says, 'your gem?'
Jack says, 'oh well'
[Lady and Jack disconnect.]
```

The next day, Lady connects again and writes another script all on her own, with no help. A few days later, she returns to write four more. She puts these scripts on her dog Cutey (a project she selected for herself, rather than one suggested by her friends.) Here are her scripts:

```
on pet this
  Announce "Cutey rolls over and wags her tail."
end


on feed this
  Announce "Cutey eats the food hungrily, and barks for a thank you."
end


on tickle this
  Announce " Cutey barks, and giggles."
end


on play with this
  Announce" Cutey runs around in circles a few times, while wagging
   her tail."
end


on scratch this
  Announce "Cutey lays down comfortablely enjoying the scratching."
end
```

Lady's learning experience was both community motivated and community supported. She wanted to learn to write a script because Jack had done so. While many people (especially girls) come to think of themselves as the sort who could never do something so high tech, Lady is surrounded by other children writing programs. This helps her to realize that she too can write scripts. Meaningful control over computers is not the privilege of the technical elite—it's within everyone's abilities. Jack and Miranda provided her with project models, role models, technical support, emotional support, and an appreciative audience for her finished work.

Lady's experience would have been very different if, like Uzi, she had logged on later in the evening and found no one else around. Members of the community were an essential part of every aspect of her learning experience.

Real samba schools and The Computer Clubhouse are physical places. People gather there both to work on their projects and to socialize with one another. The architectural space serves as a community center for the members, providing a context for both organized activity and more casual interaction.