Presented at the AERA 1996 Annual Meeting
April 8-12, New York, NY

Kids Learning Engineering Science Using LEGO and the Programmable Brick

Fred G. Martin
MIT Media Laboratory[*]

April 6, 1996

Abstract

With the aim of introducing ideas in engineering science to students at the elementary and high school levels, we have created the "Programmable Brick," a new educational technology, and collaborated with teachers in the design and assessment of curricular plans for introducing the technology to the classroom. Using the Programmable Brick, students explore a variety of concepts, including sensing, control, and systems.

Central to our approach is the belief that children learn most effectively when they are engaged in design, construction, and debugging activities. This paper reports results from a collaboration with three teachers—two elementary, one high school—developing curricular and testing models for introducing these materials to the classroom over the 1994–95 academic year.

---

[*] 20 Ames Street Room E15–319, Cambridge, MA 02139. Author's e-mail address: `fredm@media.mit.edu`. This work was supported by grants from the LEGO Group and the National Science Foundation.

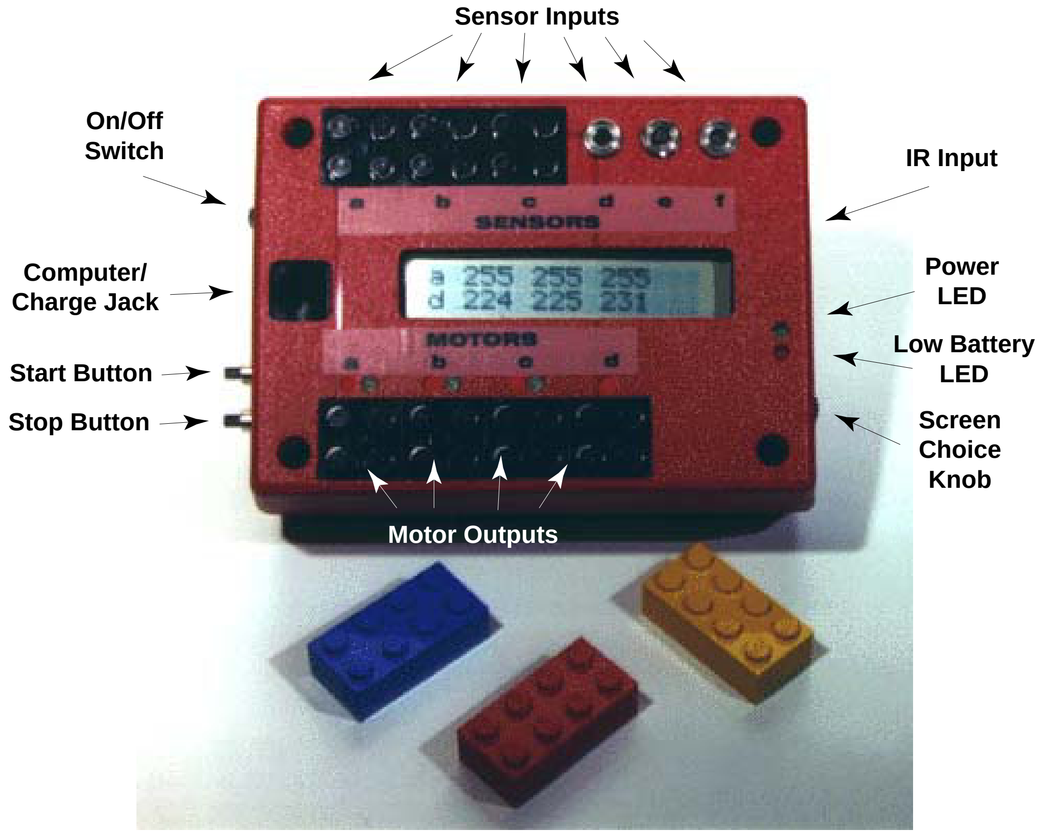


Figure 1: The MIT Programmable Brick, Model 120

Introduction

The Programmable Brick is an extension of LEGO/Logo, the commercial building kit of beams, gears, bricks, motors, sensors, computer interface, and programming language marketed by LEGO Dacta. With LEGO/Logo, children build a variety of mechanical contraptions (e.g., cars, creatures, ferris wheels, and greenhouses), plug them into an interface box that is connected to a desktop computer, and then program the creations to perform various actions.

The Programmable Brick combines the functionality of the desktop computer and the interface to the LEGO motors and sensors into a single brick, about the size of a child's juice box (see Figure 1). The Brick can operate four motors and receive information from six sensors. It has a two-line LCD screen for displaying sensor values and for selecting a program to be run. Two buttons and a knob allow the user to start and stop programs already loaded into the Brick.

To use the Programmable Brick, one connects it to a desktop computer's serial port. Then it is possible to directly run programs, display sensor values, and otherwise interact with the LEGO devices that are connected. Further, one can download a program (or several programs) into the Brick, and disconnect the desktop computer. The Brick retains the programs downloaded into it, and can operate remotely without the desktop computer.

There are several important advantages of the Programmable Brick over the commercial LEGO/Logo system. Because the Brick is small, a project's computation can be built *into the project itself*, rather than having to sit on the desktop. This allows mobile creature-type projects to roam around freely, rather than being tethered to a base station. Since the Brick can actively perform calculations while it is carried around, new categories of projects, like data-taking experiments, are possible. With the sensor display that is built into the Brick, students can observe sensor values in remote locations.

Our research group has developed a series of Programmable Bricks, beginning with an early version in 1987. Previous research with MIT Programmable Bricks is discussed in (Martin, 1988; Bourgoin, 1990; Sargent, Resnick, Martin, & Silverman, 1996). The current version, known as the "Model 120," was used in the research described in this paper.

Framework

As an educational technology, the Programmable Brick is unusual in that was "designed for designers"—the students are not merely *users* of the technology, but become *designers* as they work through problems and express their own ideas with it.

In previous work, the author implemented design-rich environments for teaching engineering science to university undergraduates (Martin, 1994) and studied notions of feedback and control in fifth-graders (Martin, 1988). In the former of these studies, the author developed a workshop-format course in which students received a kit of materials and a specification for a competitive contest challenge. In the latter, the author worked directly with fifth-grade students as they experimented with robotic materials—sensors, motors, and programming.

For the present study, we recognized the critical role that the classroom teacher has in the adoption and use of new technology, and held a week-long summer workshop with a group of twenty teachers (elementary through high school). In the workshop, the teachers worked in teams of two to four persons, developing robots to play in a competitive design challenge. In addition to the project work, discussions were held to develop ideas for classroom projects based on the Programmable Brick materials.

The elementary school teachers decided that a strongly competitive approach would not be suitable for their classrooms. We wanted a classroom framework that would encompass all age groups and appeal to both genders, and collectively decided on a theme we called the "Robotic Park." In the Robotic Park activity, students selected an animal, researched the animal and its habitat, and then implemented LEGO models of these animals, including sensors, actuators, and control programs.

In addition to the animal theme, the Robotic Park framework gave students a specific performance challenge. The LEGO animals were to perform within a four foot square rink, with a light source at one corner and an exit doorway at the opposite corner. Up to three obstacles were to be placed within the rink. Collectively, the Robotic Park event provided the structure for students to explore a number of feedback behaviors, including obstacle avoidance, light seeking, and wall following.

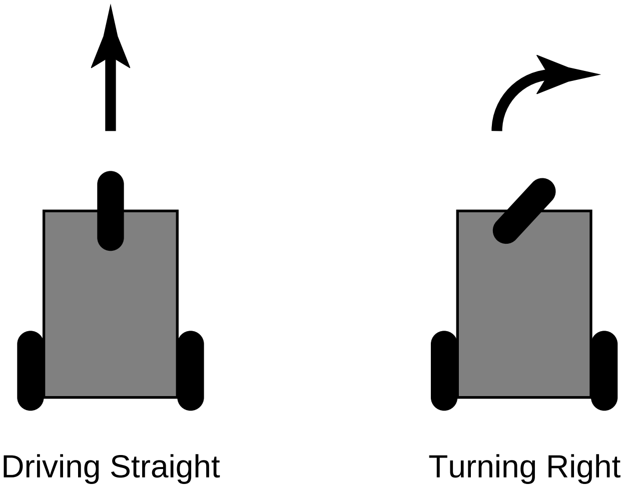
Data Source

Data in this study comes from three classrooms: one fourth-grade classroom, one combined fifth/sixth grade classroom, and one eleventh-grade classroom (at a vocational high school). In the elementary classrooms, the project spanned October 1994 through May 1995; in the high school, the project began in February 1995 and ended in May 1995. In all of the classrooms, students had extensive project time; they worked on the project for several sessions per week of forty-five minutes to two hours each in length.

Data collected includes videotaped interviews before and after the project, observations made from interactions with students during project work, notes during and after classroom sessions, and a photographic record of student work. I visited the classrooms once per week and met with the classroom teachers at the end of the school day during these visits.

Results

Students' project work demonstrated a variety of approaches and understandings. Depending on choices students made at the beginning of their projects, such as what kind of animal they wanted to build, what kinds of sensors it would use, and what behaviors it would exhibit, students encountered problems that led to various challenges in the area of systems engineering.



The standard vehicle base could only drive straight ahead or turn toward the right. It was incapable of turning to the left.

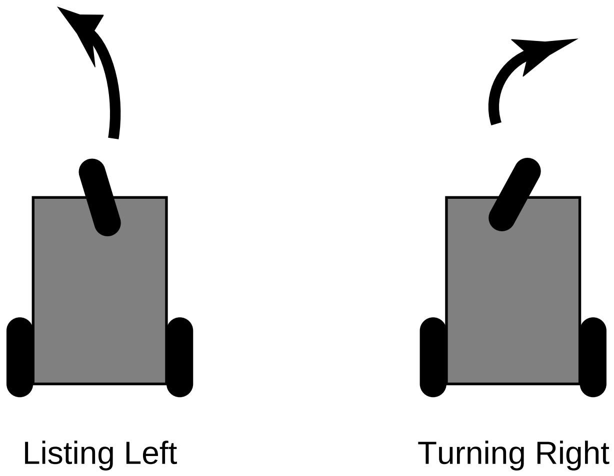
Figure 2: Stock LEGO Vehicle Used for Anchovy Project

The Anchovy

One team created a robotic “anchovy,” and one of the behaviors they wanted it to perform was to track along the face of a wall. The team’s vehicle was adapted from a model given in LEGO plans, which had an unusual maneuverability problem: it could only drive straight ahead or turn to the the right. In order to get the creature to follow a wall on its left-hand side, the students biased its forward movement to the left, so that it could perform repeated corrective movements driving to the right.

Figures 2, 3, and 4 illustrate the concept of the robotic anchovy. In Figure 2, the stock LEGO platform used to build the anchovy project is illustrated. It can drive straight forward or turn to the right. Figure 3 shows the modified platform, which veers gently to the left and also can turn to the right. Figure 4 shows the resulting vehicle in action, advancing and striking the wall, and turning to the right to continue.

Because the standard vehicle had a mechanical limitation, the anchovy designers thus deliberately introduced error into the system in order to get it to perform as desired. This is an unusual solution to the initial problem, demonstrating flexibility in the students’ thinking.



After being modified, the anchovy platform “listed” toward the left, but still could make corrective movements to the right. This was ideal for the task of following along a left-hand wall.

Figure 3: Anchovy Platform After Modification

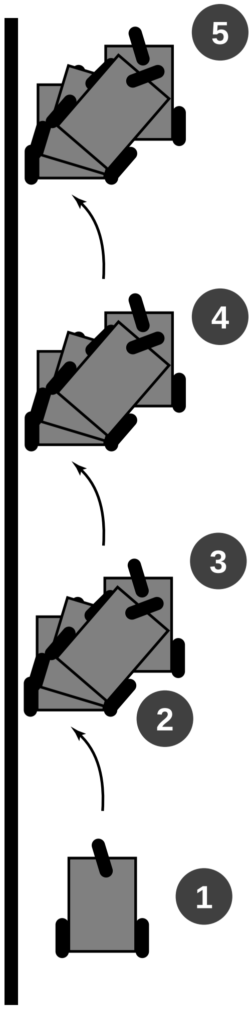
The Turtle

A team building a robot turtle wanted it to act “afraid” of light, and run away from light sources. The mechanism they designed for the turtle robot included a retractable head that could be drawn into the turtle’s shell, to more fully express the turtle’s fear. Figure 5 is a photograph of two of the students who worked on the turtle with a preliminary version of the robot itself.

In order to understand the patterns of light, the students brought the turtle to various positions on the playing table, and studied the sensor readings as the turtle was placed in various positions. For each position, the light sensor reading was measured. The students wrote down the value of the light sensor on a yellow “sticky-note” and placed it on the table where the reading was taken (see Figure 6).

From this experiment, the students gained a sense of how the light sensor performed. In particular, they discovered that the closer the turtle was to the light source, the smaller the value was registered by the light sensor.

This understanding led to the students’ approach for avoiding the light. The method they employed was to have the turtle rotate until the light sensor reading was above a particular threshold (indicating that the turtle was facing the dark) and then drive forward for a while.

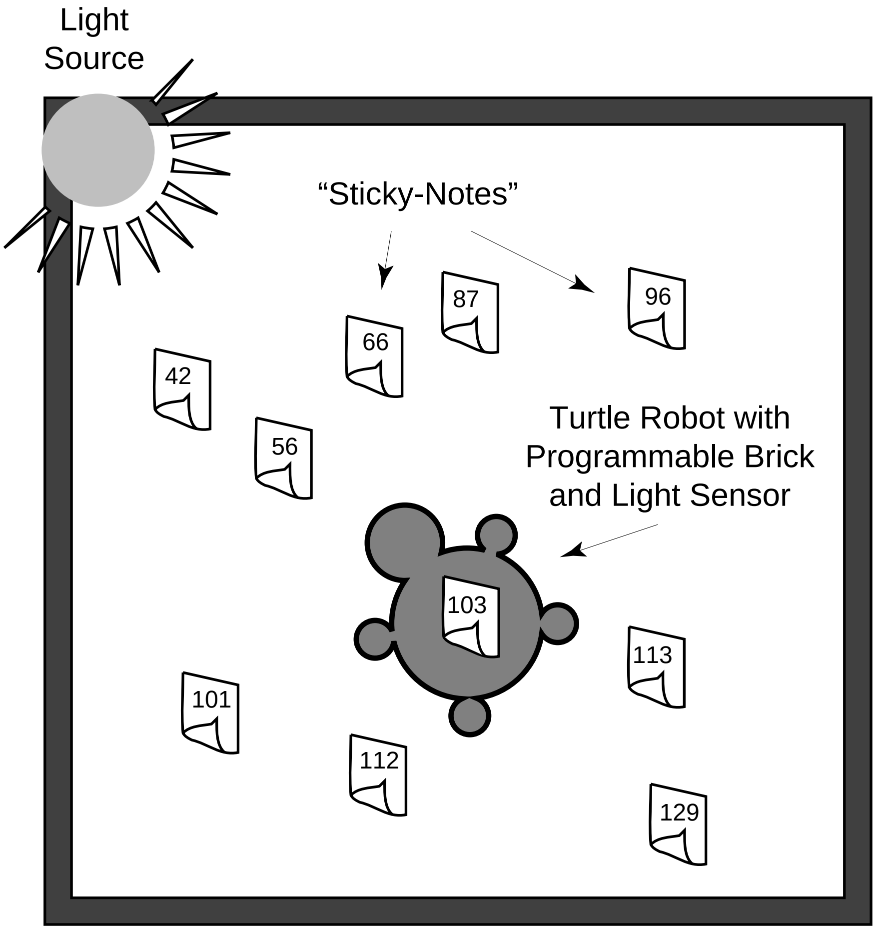


The Anchovy begins at the lower diagram (1), heading toward the wall on the left. After progressing a little, it runs into the wall (2), and then performs its right-turn behavior, which rotates it outward (3). Then it drives forward and toward the left, eventually reaching the wall again (4). This behavior then is repeated (5).

Figure 4: Anchovy Vehicle Following the Wall

photo © 1995 Fred G. Martin
fredm@media.mit.edu

Figure 5: The Turtle Designers Test a Prototype



The team building the robot turtle needed it to act afraid of the light, and move away from the light source. In order to understand the patterns of light that their robot would encounter, the students took a set of light sensor readings, marking each reading with a "sticky-note" placed on the position of the reading.

Figure 6: Light Sensor Field Strength Experiment

Figure 7: The Designers of the Dinosaur Exhibit Their Creation

The Dinosaur and the Jeep

In a project inspired by the film *Jurassic Park*, a team designed a dinosaur which sought out light sources, and a Jeep that carried one. This was an extension of the basic Robotic Park parameters, which employed a fixed light source.

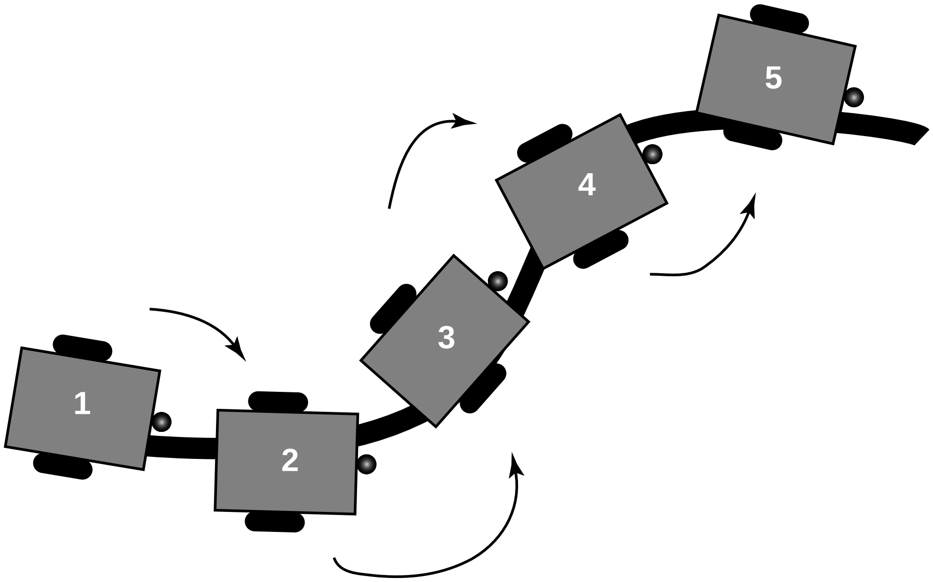
Like the turtle that avoided the light, the dinosaur project required the students to develop an algorithm for finding light. In the case of the dinosaur, light-seeking was necessary, while the Jeep needed to avoid the light.

Figure 7 shows three of students in the dinosaur/Jeep team, with the dinosaur. In the final project implementation, both the dinosaur and the Jeep carried flashlights, so each could “see” the other. The students deliberately gave the dinosaur a slight edge in speed, so it always got its prey.

The Line-Follower

In the high school class, we modified the design challenge to include a line-following segment. One of the goals was to build a machine that could track along a curved line of black tape placed on the playing field surface.

Based on previous work developing robots to perform this task, a “canonical” solution to the problem is shown in Figure 8. Using just one light sensor to detect the line, the solution is to cause the robot to weave back and forth across the line. When the robot crosses completely over the line, it changes direction to go back



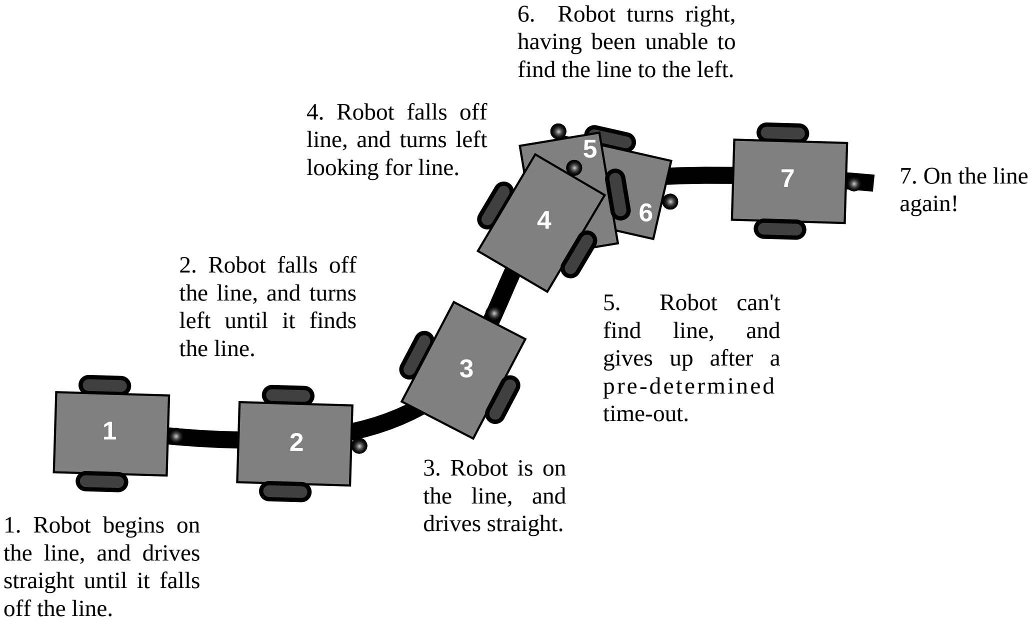
In the "canonical" line-following algorithm, the robot uses a single sensor in a front center position. The robot starts on one side of the line, and begins moving so as to cross the line while moving forward. After crossing all the way over the line so that the line sensor sees the table, the robot changes direction and veers back across the line. This process repeats, giving the robot the appearance of waddling back and forth across the line. Note that when using this algorithm, the robot never drives straight forward; it is always turning across the line one direction or the other.

Figure 8: Canonical Line-Following Algorithm

across it again.

As long as the robot is making forward progress while weaving back and forth, the approach is successful. Many students find this solution counter-intuitive, though, since the robot never drives straight ahead. Many people have the idea that when the robot is directly over the line, it should simply drive straight ahead. The problem with this is that after driving straight ahead and falling off the line, the robot will not know which way to turn to find the line again.

We did not present the canonical answer to the students, preferring them to struggle with the problem themselves. One student in particular, Djonnie, took a special interest in the problem, and devised the following solution (depicted in Figure 9):



Djonnie's line-following algorithm uses the same robot configuration as the canonical follower, but is an approach based more on personal intuition. If the robot is on the line, it drives straight ahead until it falls off the line. After leaving the line, the robot attempts to find the line by turning left. If this succeeds, the robot starts driving straight again. If it fails, after a certain amount of time, then the robot assumes that the line turned the "wrong" way, and spins to the right for twice as long as it tried to turn to the left.

Figure 9: Djonnie's Line-Following Algorithm

1. If the robot is on the line, it should drive straight ahead.
2. After falling off the line, the robot turns left, looking for the line. If it finds it within two seconds, it goes back to step one (and starts driving forward on the line).
3. If the robot does not find the line within two seconds, then it assumes that the line headed off to the right, and spins right for four seconds, and then continues back at step one.

There are several things worth noting about Djonnie's solution. Firstly, it works. It is not as symmetrical as the canonical approach, but it is efficient in its own way. The robot makes fine progress when driving straight and when the line bears to the left (its preferred direction). The corrective action that the robot takes when the line bears to the right is often amusing.

More importantly, Djonnie used what Papert calls *body syntonic thinking* in developing his approach (Papert, 1980). That is, the student thought about how he would solve the problem if he were the robot. As a student who is not particularly academically minded, Djonnie made the connection between the robot as a physical entity and how he would use is body to solve the problem, and leveraged this understanding to create the solution.

Conclusions

In the research reported here, ample classroom time was made available for the activities described. In the elementary classrooms, the children worked for one to two hour sessions, two or three times a week, for the majority of the school year. In the high school classroom, the hours per week were about the same as the elementary classroom, but the duration was about half of the school year.

The elementary school teachers justified the extensive time devoted to this project because of the wide range of activities that the children perform. In addition to the work described in this paper, the children did library research, journal writing, oral presentations, and other activities. The school's art teacher contributed ideas to the students' work, which included a variety of non-LEGO media such as paint, paper mâché, cardboard, and clay. In short, the Robotic Park activity served to integrate content from across the curriculum.

Because of the amount of time made available for project work, many of the students gained a definite sense of mastery over the technological materials. Whereas they struggled at first learning how to use the Programmable Brick and its related materials (motors, gears, sensors, and programming), by the end of the school year it was apparent that the students were fluent with the technology. This sense of mastery expressed itself in the students' feelings of ownership and pride with their accomplishments.

The core part of this paper presented specific technological aspects of the students' work that can be characterized as engineering science learning. All of the students' projects dealt squarely with the concept of algorithm in designing their robots' behaviors. All of the projects involved the idea of sensing, and of thresholding sensor data as a method of interpreting its significance. For many of the students who participated in the project, this was their first experience in designing and building a system that incorporated mechanical, electrical, and computational elements.

It is important not to forget the value of whimsy in the students' work. Especially for the younger students, artistic and theatrical ideas were a large part of the development of the robot projects. The turtle's head retracted into its shell; a crab's claws snapped when struck; an alligator's giant mouth devoured any unsuspecting victims. The Robotic Park activity was successful because it provided a way for students with a wide variety of interests and abilities to contribute to a larger cause.

Acknowledgments

Mariann Hayward and John Bilotta, from the Peace Dale Elementary School, and Lee Rabbitt, from the East Providence Career and Technical Center, contributed many ideas to the successful classroom implementation of this project (as well as being a joy to work with). Wanda Gleason participated in the on-site research at Peace Dale, providing much-valued insight and assistance. Janice Kowalczyk provided greatly appreciated organizational work and many words of inspiration. Brian Silverman co-designed the Programmable Brick technology. Mitchel Resnick contributed to the Programmable Brick project's organization and long-term direction. Seymour Papert provided on-going inspiration for much of the work described herein.

Reference

Bourgoin, M. O. (1990). Children using LEGO robots to explore dynamics. In Harel, I. (Ed.), *Constructionist Learning*. MIT Media Laboratory, Cambridge, MA.

Martin, F. G. (1988). Children, cybernetics, and programmable turtles. Master's thesis, The Massachusetts Institute of Technology, MIT Media Laboratory, 20 Ames Street Room E15–315, Cambridge, MA 02139.

Martin, F. G. (1994). *Circuits to Control: Learning Engineering by Designing LEGO Robots*. Ph.D. thesis, Massachusetts Institute of Technology, MIT Media Laboratory, 20 Ames Street Room E15–315, Cambridge, MA 02139.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.

Sargent, R., Resnick, M., Martin, F., & Silverman, B. (1996). Building and learning with programmable bricks. In Kafai, Y., & Resnick, M. (Eds.), *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*. Lawrence Erlbaum, Mahwah, NJ.

Presented at the AERA 1996 Annual Meeting
April 8-12, New York, NY