JOURNAL OF MATHEMATICAL BEHAVIOR 10, 79–89 (1991)

Learning by "Cheating": Students' Inventive Ways of Using a Boxer Motion Microworld

STEPHEN T. ADAMS
ANDREA A. DISESSA
University of California, Berkeley

This article describes ways in which sixth-grade students using a motion program in a pilot physics class invented methods for working on difficult problems. The students "cheated" by directly manipulating part of the display that also functioned as a working part of the program. In doing so, they found a way to solve problems by solving simpler problems first. This strategy spread in the classroom to become a communal resource for attacking the most difficult problems. The teacher and students negotiated ground rules for using this method productively. Although this episode was not planned, the characteristic of the program that the students exploited was a direct consequence of the design of Boxer, the computer environment in which the program was written. We see this episode as an example of a kind of student-initiated learning that can emerge given a learning-oriented classroom and open technical designs.

Designers of technology for education often seem to pretend that they can design their artifacts and learning activities independently of the cultural and social context in which they are placed. We believe this is neither desirable, nor, in the last analysis, possible. The classroom vignette on which this note is based illustrates not only the general theme of the interdependence of learning artifacts and cultural surroundings, but also some particular interdependencies upon which our experiments in learning with Boxer are based.

INTRODUCTION

When we introduced a motion microworld to sixth-grade students taking a pilot physics course based on Boxer, we included a problem that was impossible to

This research is supported primarily by a grant from the National Science Foundation, NSF–MDR–88–50363, to Andrea A. diSessa. We also gratefully acknowledge a contribution to support students from Apple Corporation, and a contribution of equipment from Sun Microsystems.

This article does not necessarily reflect the views of the National Science Foundation, Apple Corporation, or Sun Microsystems.

The authors are grateful for comments on an earlier draft of this article from David Hammer, Michael Leonard, Don Ploger, and Bruce Sherin.

Correspondence and requests for reprints should be sent to Stephen T. Adams, School of Education, Tolman Hall, University of California, Berkeley, CA 94720.

solve. We were not sure how they would react, but it seemed likely to be an interesting probe of their skill. Perhaps they would not realize it was impossible, work futilely on it, and get frustrated. On the other hand, they might realize it was impossible and not try to solve it. How the students, in fact, dealt with this problem makes an interesting case study of the interactions of their emerging learning culture and the technical aspects of our designs.

The students devised methods for solving the problem by "bending the rules" in different ways. Furthermore, they announced to the teacher that they were "cheating," shared information on how to cheat with other students, and negotiated standards about situations in which it was proper to cheat. In other words, the students adopted goals other than simply "following the rules of the game" as presented to them. We argue that, through this process, the students did not cheat in the sense of circumventing learning, but came to understand the subject matter in a deeper way.

This article describes these events as a way of illustrating one kind of student-directed learning that can occur within the students' computational learning culture. We interpret what happened as a product of mutually compatible interactions among students' competence, their cultural assumptions and classroom dynamics, and the design of the computational representations they were using.

THE NUMBER-SPEED MICROWORLD

NUMBER-SPEED is a microworld implemented in the Boxer computer environment that aims to help students learn about velocity and acceleration in one dimension through activities involving multiple representations of motion (Adams, 1991). The microworld was introduced to a class of 8 talented sixth graders (4 boys and 4 girls) roughly 3 months into a pilot 1-year physics course.

Figure 1a shows the control interface for the microworld. Students use this interface to program two graphical "turtles" to move according to a mathematical representation. The representation is simply a list of positions and a list of speeds. The student is asked to create these lists by setting a triplet of numbers, corresponding to a turtle's initial position, initial speed, and acceleration. For example, the student might specify "0" as the initial position, "4" as the initial speed, and "0" as the acceleration. Pressing a button programs the turtles' motions by generating lists of positions or speeds that correspond to the triplet. The list of speeds generated for the above triplet would be "4 4 4 4 4 4 . . ." and the list of positions would be "0 4 8 12 16 20 . . ." (Figure 1a).

The microworld thus utilizes a discrete model of motion based on lists of numbers where each successive number describes a turtle's motion (position or speed) at successive times. The preceding list of positions would cause a turtle to start at Position 0 and then move to Positions 4, 8, 12, 16, and 20 at successive times (Figure 1b). Note that the microworld allows creating motions with constant acceleration, but not with changing acceleration.

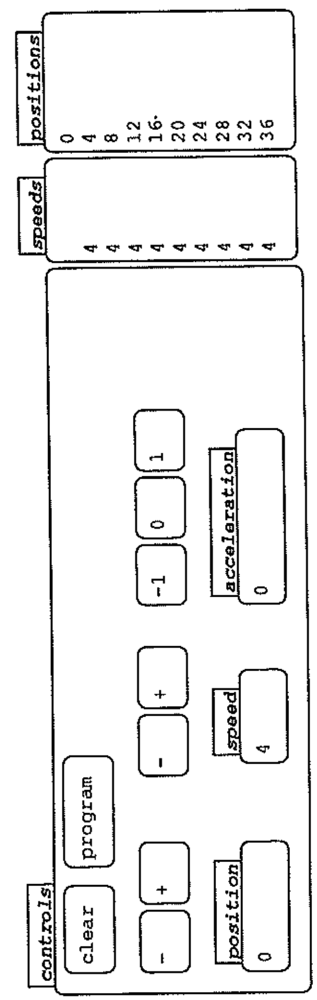

Figure 1a. The student selects "0" for initial position, "4" for initial speed, and "0" for acceleration. Selecting "program" fills in the lists of speeds and positions.

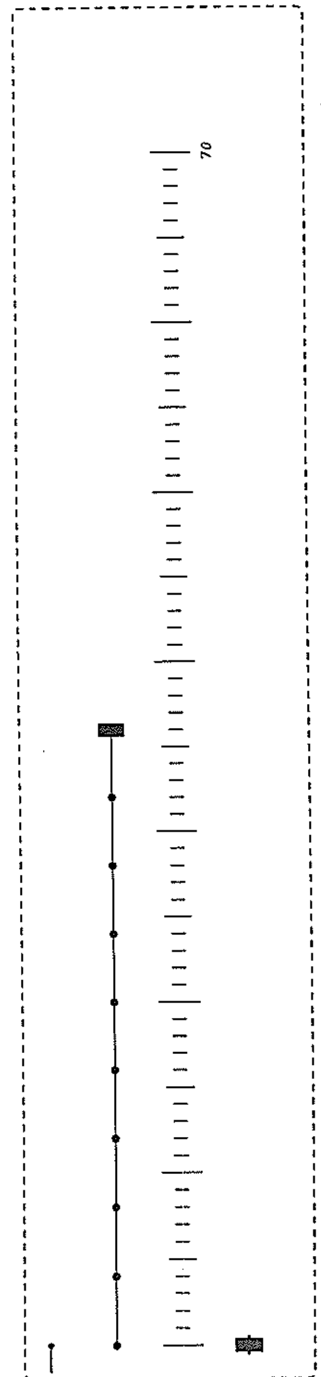

Figure 1b. The turtle moves according to the number sequences in the "speeds" and "positions" boxes. In this case, the turtle moves 4 spaces per unit of time.

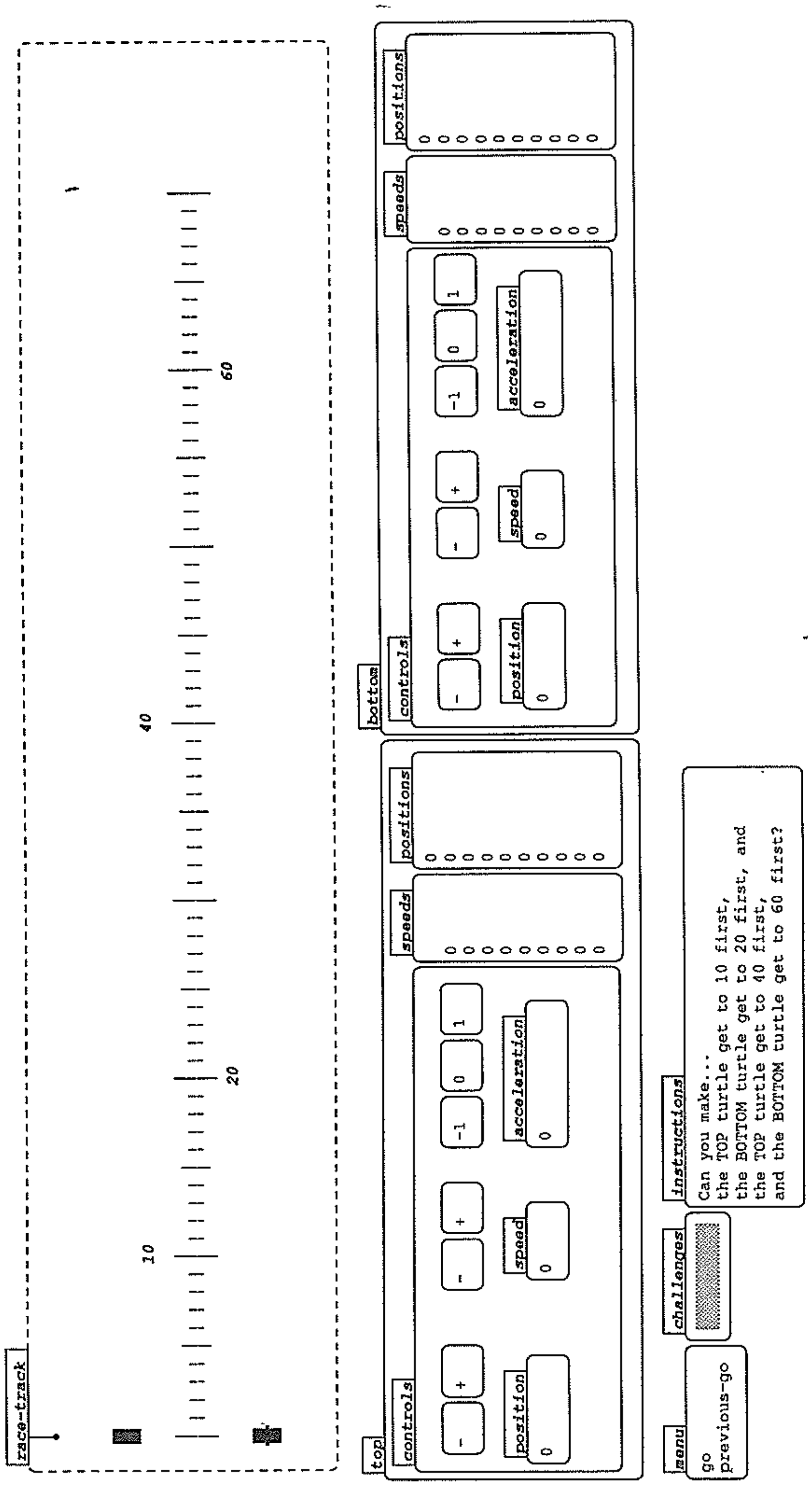

Figure 2. The impossible "three passes" problem.

The "Three Passes" Problem

The "three passes" problem was the first problem in the course that was impossible to solve, and it was part of the last group of problems in the NUMBER-SPEED microworld. In this last group of problems, students were asked to make a pair of turtles move according to specified constraints. The constraints are given as locations that one turtle has to reach before the other turtle.

The "three passes" problem (Figure 2) is to make the turtles pass three times. The student is asked to make the top turtle get to position 10 on a grid first, the bottom turtle get to 20 first, the top turtle get to 40 first, and the bottom turtle get to 60 first. Thus, the turtles must pass each other three times among these four positions.

This challenge is impossible given the microworld's constant acceleration constraint. An algebraic way to see this is that, because the turtles have constant acceleration, the functions that specify their positions are (at most) quadratic in time. Hence, the function that describes the *difference* between the positions of the two turtles must also be (at most) quadratic. The turtles will pass when this difference quadratic is zero. However, because a quadratic can be zero in at most two places, the turtles can pass at most twice.

The day the students were working on these problems, we were videotaping the class. The 2 students who are at the center of our study happened to be working as a pair in the foreground of the video, with the rest of the class partly visible behind. The room was small enough to allow effective recording of public conversations involving any members of the class. The class teacher and an observer were on hand.

Inventing a New Way to Use the Microworld

Students S and C (both boys) doubted that it was possible to create the three passes motion by following the rules. Upon reading the problem, S said:

S: How the heck are we supposed to—I can get the top turtle to 10 first. I can handle the bottom turtle getting to 20 first. I can't handle the top getting to 40 first *and* the bottom turtle getting to 60 first.

S then asked,

S: Can we switch around in the positions box, like change 'em?

Not receiving a response to this question, the pair proceeded to create the motion by using what S later dubbed "the direct method." They typed their own number sequences directly into the velocity-list box, thereby bypassing the triplets for position, speed, and acceleration.

S: [typing numbers directly into the box]: Heck, why not?

C: You're a genius, Scott [not his real name].
S: [to microphone]: We are cheating.
C: It just looks that way.

Figure 3 gives a simplified version of the type of solution they reached. After solving this problem this way, they announced their success and their unorthodox method to the teacher (Ms. K):

C: We did it! We did it! Ms. K, look we did it . . . We cheated, see.

They then returned, unprompted, to the original problem to consider whether it was possible.

C: Why does that make sense? Is there a way to do this without cheating?

The students and the observer then entered a dialogue about whether or not the problem was possible to solve without cheating. The students argued that it was not possible.

C: Yeah, you'd have to change the speeds. You can't do that without acceleration . . . You can't accelerate, not accelerate, accelerate, not accelerate.

In sum, the students solved the three passes problem with the direct method, went back to reflect about whether the problem was possible, and produced explanations why it was not.

We are not certain what exactly accounts for our students' conviction that three passes is impossible under the rules. Our best interpretation is as follows. The lists representing speeds, and the display trace of a turtle's motion, are all visually very "regular." That is, with constant acceleration, they are linear. In such cases, there can be only two interesting regions, where one turtle is going faster and—after the velocities "cross"—where the other turtle is going faster. Now, by arranging the initial position, one can have the initially slower turtle ahead, then passed by the initially faster turtle. Then, when the speed relationship reverses, the now-faster turtle will eventually pass his slower comrade. But, in order to make a third pass, one will have to edit the velocity list so that either the now-slower turtle becomes faster, or vice versa. Roughly speaking, this editing changes a deceleration into an acceleration (or vice versa), which C may be paraphrasing as the impossibility of "accelerating, not accelerating" and so on. Surely our students could not be so articulate about why the problem was impossible. But their recognizing a clear irregularity in the velocity lists that they interpreted as necessary, but impossible for three passes, is, we believe, good intuitive mathematics.

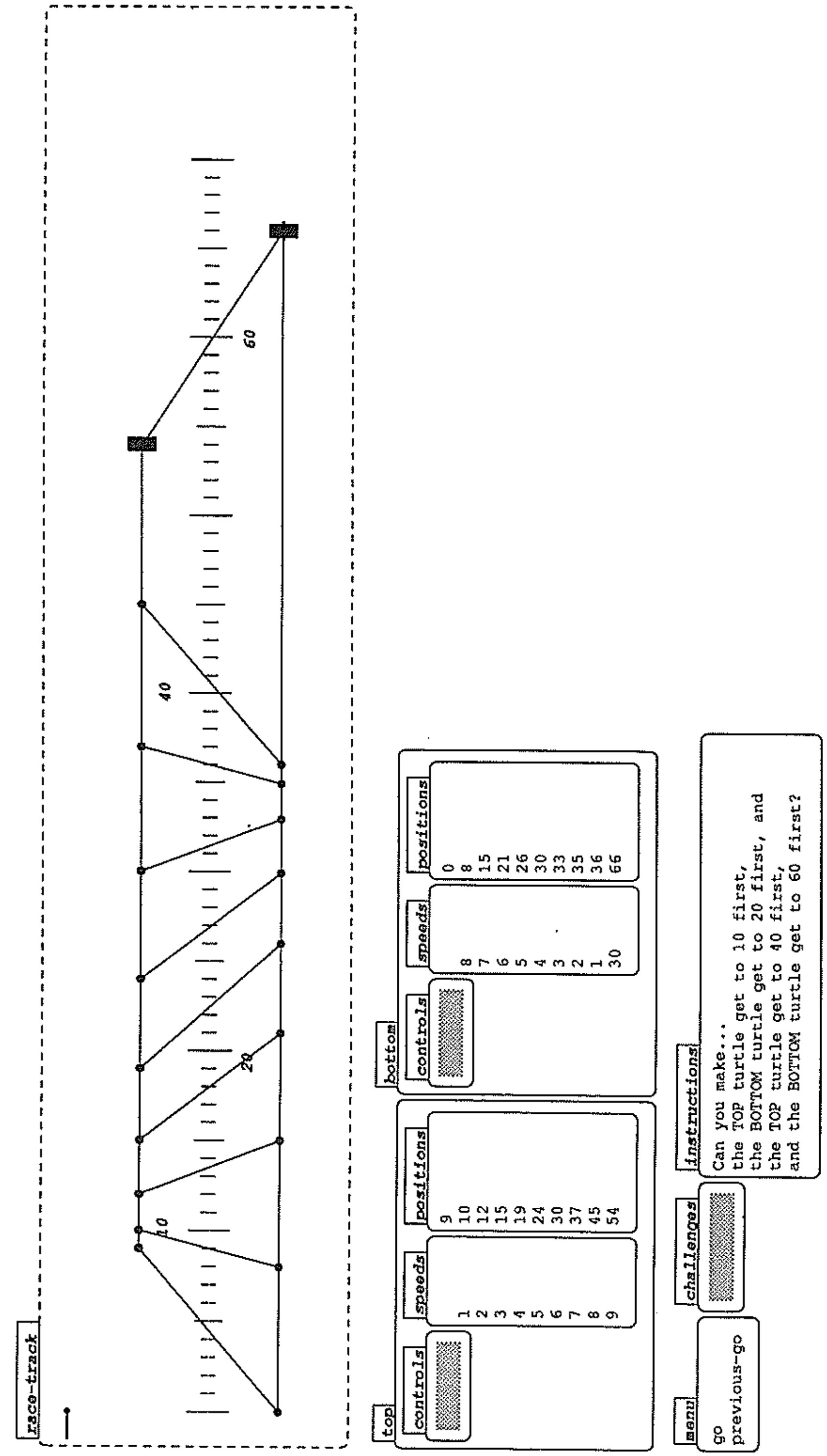

Figure 3. A "fudged" solution to the "three passes" problem. The student has "doctored" the last number of the bottom turtle's number sequences.

CHILD COMPETENCE, MATHETIC CULTURES

Are these students just unruly children working in a classroom where they are not disciplined for cheating? Hardly. We view this as a highly successful learning event that is dependent both on the competence of the children and on a classroom culture that is supportive of learning goals: *mathetic* in the sense introduced by Papert (1980), meaning pertaining to learning.

At the cultural level, we can see children who share authority with the teacher, who are not cowed into lockstep approaches to their work, and who can be proud of an accomplishment they regard as genuine, even if it is reached by an unusual method. They do not constantly ask "What does the teacher want?" but at least as often, "How can we understand and accomplish?" In this class, as will be more evident later, learning goals can dominate over procedural ones. The final event in the episode dramatizes for us how much learning is in the fore. Even after "solving" the problem, the children return to consider it again because they judge, expertly in this case, that there is something more to learn.

We do not intend to paint an idyllic picture of these children. They are frequently rambunctious and mischievous, testing limits set by the teacher. But it is also clear that some of the initiative for learning comes from them. When they are cooperative in the class, it seems to us it is more from a sense of shared enterprise in learning rather than from unselective regimentation.

On a more cognitive level, one can take the students' decision to try to solve the problem by cheating as the invocation of the Polya (1945) heuristic, "solve a simpler problem first." Cheating thus becomes a "way in" for students to begin working on an otherwise intractable problem. Furthermore, the method appears highly successful in this case. Creating a representation of how velocities must change in order to create three passes allowed them to see that it was impossible. This mediated *reductio ad absurdum* (if it is to happen, it must happen in this way, but this is impossible) involved a symbolic version of a Polyan "draw a picture." The velocity list became a "thing to think with" that cued appropriate ways of conceptualizing the problem. The possibility of student-generated representations of this sort is just the kind of thing that a flexible computational medium like Boxer is intended to support. We will return to this important point later.

Classroom Dynamics

We found other ways in which the classroom culture was oriented mathetically rather than toward an authority that dictates the way to do things. Ideas about cheating spread from S and C to the other students. A female student (J) was working with the teacher (K) at a computer adjacent to the one at which S and C were working. J saw and heard that C and S were using the direct method, and J tried using it, too, although she was working on an easier problem. The teacher did not prohibit J from doing this, but proposed reserving the direct method for

more difficult problems. Student S, who was the inventor of the direct method, interrupted J and K's conversation to give a specific prescription for when it was proper to use it, to which the teacher then agreed.

K: [talking to J, who has just "cheated"]: All right. Listen. What you did right there is you cheated. And the thing is, the point is, at the end we can cheat, but right now its pretty ea—I mean—
S: You shouldn't cheat until g and h (the hardest problems).
K: Yeah, g and h are the cheating ones, I think.

We found this to be typical of this teacher's interventions, which were seldom authoritarian, and more often involved negotiating working rules that both the teacher and the students would find reasonable.

In other circumstances, students had opportunities to cheat that defeated the microworld's learning purposes. They did not, as a rule, pursue them. For example, in a different set of problems, students were supposed to watch a turtle's motion and figure out the numbers needed to program it. The numbers were hidden in a closed box. Nothing prevented the students from opening the box to peek, except a message that said "No peeking!" The message was sufficient. In this class, rules were not given ultimate priority. The class neither rigidly followed them, nor indiscriminately broke them.

Flexible Computational Representations

The success of this episode depends on an important quality of the microworld's design. Put simply, the microworld *allowed* the students to program the turtles with the direct method. This possibility reflects, in turn, central underlying properties of Boxer, the medium in which the microworld was implemented.

One of the central representational goals of Boxer is "concreteness" or "naive realism" (diSessa & Abelson, 1986). In particular, Boxer variables do not just hold values, they also display them, giving users a representation to interpret, reason on, and manipulate directly. Variables appear as boxes on the screen, containing text and/or other boxes (including programs and graphics) which the user can directly edit at will.

For this reason, the user can understand the speed boxes of Figure 1 not as just special purpose display devices, but as generic boxes that can be typed into and edited, just like any other boxes. In other words, computationally active parts of a program are a part of the display to the user of the microworld. This gives the user access to modify those parts of the program, and this type of access is simply not available in other programming systems.

The possibility of programming the turtles with the direct method is, thus, a natural consequence of the design of Boxer. The direct method is an idea about the microworld that was waiting to be invented, and its use here is hardly an isolated event. For example, in pilot testing of NUMBER-SPEED, another subject invented a similar idea. D, a 12-year-old boy who had not used Boxer before, encountered difficulty with a similar but easier problem requiring the turtles to pass only two times. D also generated the idea of programming the turtles with the direct method and asked the experimenter how it could be done:

D: Ok. Oh. Twice. Oh God. How do you change the numbers individually?

D later announced he was changing the numbers and proceeded to solve the problem by the direct method, typing directly into the turtle's position box.

D: Now I'm going to go into changing it because I have to.
E: You're going to what?
D: I'm going to change it individually; I can't help myself. I'll slow it down right about, ok . . . [subject types numbers into the position box].

The student was not coached to use the direct method in this context, and it was a surprise for the experimenter, E, that he wanted to use it. Before seeing the microworld, however, D had been shown how to create and edit Boxer boxes. Apparently, with very little experience, he understood the possibility of the direct method as a consequence of Boxer's general properties.

Like S and C, this subject's cheating plausibly served as a stepping stone for understanding the problem. After he solved the problem with the direct method, he was able to go back and solve it according to the rules. Even if such cheating does not directly aid solution attempts, it may provide *affective* support to students while they gain familiarity with the problem.

Boxer is designed to accept and even encourage initiative on the part of students and teachers. The system is designed to be open and inspectable, and every aspect of it is changeable by means of the simplest operations, such as text editing. Although this openness can allow users to defeat the original intentions of instructional designers, we believe it can augment some of the best characteristics of a learning-oriented classroom.

SUMMARY

We have described ways in which students in a pilot Boxer physics class subordinated "following the rules" to working on and thinking about a difficult problem. The students "cheated" by directly manipulating part of the display of a microworld that also functioned as a working part of the program. In doing so, they found a way into a difficult problem by solving a simpler problem, and they constructed a visual representation of the solution that allowed them to see that the more difficult problem was impossible.

This strategy spread in the classroom to become a communal resource for attacking the most difficult problems. The teacher and students negotiated

ground rules for using these new resources productively. Although we did not plan this episode, we see it as an example of a kind of student-initiated learning that can emerge given a learning-oriented classroom and open technical designs.

REFERENCES

Adams, Stephen (1991, April). Learning about motion in one dimension using a numeric computational representation. Paper presented at the annual meeting of the American Educational Research Association, Chicago, Il.

diSessa, Andrea, & Abelson, H. (1986). Boxer: A reconstructible computational medium. *Communications of the ACM, 29*, 859–868.

Papert, Seymour (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

Polya, George (1945). *How to solve it: A new aspect of mathematical method.* Princeton, NJ: Princeton University Press.

The Journal of Mathematical Behavior

EDITOR:
ROBERT B. DAVIS

The Journal of Mathematical Behavior is published tri-annually by Ablex Publishing Corporation, 355 Chestnut Street, Norwood, NJ 07648.

Volume 10, 1991

Individual Subscription: $35.00
Institutional Subscription: $80.00

An additional charge of $15.00 will be made for overseas subscriptions.

Printed in U.S.A. ISSN 0732-3123

THE JOURNAL OF MATHEMATICAL BEHAVIOR
Volume 10, Number 1, 1991

SPECIAL ISSUE:

BOXER!

CONTENTS