Microcomputers in Education: Cognitive and Social Design Principles

Report of a conference *

Edited by

Thomas W. Malone
Cognitive and Instructional Sciences Group
Xerox Palo Alto Research Center

and

James Levin
Laboratory of Comparative Human Cognition
University of California, San Diego

The successful use of microcomputers in education depends critically on the cognitive and motivational processes in learning and the social structure of the educational setting. A number of different groups concerned with these issues have recently tried to specify explicit design principles for using computers successfully in different educational environments. This report summarizes a workshop that brought together people from some of these groups to describe the current state of their groups' efforts and to work toward integrating these efforts into a larger scale statement.

The first part of this report summarizes the short informal presentations made by the workshop participants in order of presentation; the second part describes some examples of well-designed instructional games and articulates several general themes that ran through the conference.

The primary goal of this conference was to specify principles that are actually useful in designing instructional environments. As such, many of the principles discussed here are rough heuristics or rules-of-thumb rather than precisely defined scientific laws. Experienced workers in this field may find many of the principles obvious or well-known, but, in fact, some of the principles that seem the most obvious are the most often violated. It is our hope that this summary will serve both as an introduction to the important issues for those who are not yet familiar with them, and as a stepping stone from which experienced workers can move toward a more powerful set of design principles.

---

*This is the first part of a two-part report of a conference hedl March 12-14, 1981, at the University of California, San Diego, sponsored by the Carnegie Corporation. Part 2 will appear in the next issue of the Bulletin.

List of Participants

Organizers

James Levin
Laboratory of Comparative Human Cognition (D003)
University of California, San Diego
La Jolla, CA 92093

Thomas Malone
Cognitive and Instructional Sciences Group
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

Invitees

John Seely Brown
Cognitive and Instructional Sciences Group
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

Michael Cole
Laboratory of Comparative Human Cognition, D003
University of California, San Diego
La Jolla, CA 92093

Allan Collins
Bolt Beranek and Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

Robert Davis
PLATO Project
College of Education, Curriculum Laboratory
University of Illinois at Urbana-Champaign
1212 West Springfield
Urbana, Illinois 61801

Andrea diSessa
LOGO Project
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Sharon Dugdale
PLATO Project
Computer-based Education Research Laboratory
252 Engineering Research Laboratory
103 S. Mathews Ave.
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Gerhard Fischer
Azenbergstr. 12
University of Stuttgart, Dept. of Computer Science
D-7000-Stuttgart 1
West Germany

Laura Gould
Learning Research Group
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

James Hollan
Naval Personnel Research and Development Center (Code 304)
San Diego, CA 92152

Edwin Hutchins
Naval Personnel Research and Development Center (Code 304)
San Diego, CA 92152

Ted Kahn
Atari, Inc.
1265 Borregas Avenue
Sunnyvale, California 94086

Marge Kosel
Minnesota Educational Computing Consortium
2520 Broadway Drive
Saint Paul, Minnesota 55113

Mark Lepper
Department of Psychology
Stanford University
Stanford, CA 94305

Alan Lesgold
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260

Mark Miller
Central Research Laboratories
Texas Instruments, Inc.
P.O. Box 225936, MS 371
Dallas, Texas 75265

Sponsors

Frederick Mosher
The Carnegie Corporation
437 Madison Ave.
New York, NY 10022

Dave Robinson
The Carnegie Corporation
437 Madison Ave.
New York, NY 10022

Vivian Stewart
The Carnegie Corporation
437 Madison Ave.
New York, NY 10022

SUMMARY OF PRESENTATIONS

Thomas W. Malone
Cognitive and Instructional Sciences Group
Xerox Palo Alto Research Center

Tom Malone described several studies of what makes computer games fun and suggested a framework for using the same features to make computer-based learning environments interesting and enjoyable (see Malone, 1980, 1981). He first described a survey of computer game preferences and two experiments that started with popular computer games and removed features, one at a time, to see which features made the most difference in the appeal of the game. Games studied in this way included "Breakout" and a game called "Darts" designed to teach children about fractions.

Malone then suggested that there are three primary categories of features that make learning fun:

> (1) *Challenge.* In order for an activity to be challenging, it should present a *goal* whose *outcome is uncertain.* Ways of making outcomes uncertain for a wide range of players include (a) having variable difficulty levels (either chosen by the player or determined automatically) and (b) having a number of goals at different levels all embedded in a single environment. These multiple level goals can often be encouraged by score-keeping or speeded responses.
>
> (2) *Fantasy.* Fantasies in instructional activities can make the activities emotionally appealing. They can also provide practical examples and vivid images for the use of the skill being learned. *Intrinsic fantasies* which are intimately related to the skill being learned are hypothesized to be more interesting and more educational than *extrinsic fantasies* which depend only on whether the students' answers are right or wrong. Since there are large individual differences in the fantasies people find appealing, instructional designers should either pick fantasies very carefully or let students choose among several fantasies for a given educational goal.
>
> (3) *Curiosity.* Educational activities can evoke *sensory curiosity* by including audio and visual effects, such as music and graphics. They can evoke *cognitive curiosity* by leading learners into situations in which they are surprised. To be educational, the surprising situations should include information that helps the learners understand the misconceptions that led them to be surprised in the first place.

Jim Levin and Michael Cole
Laboratory of Comparative Human Cognition
University of California at San Diego

Jim Levin described observations of children interacting with computers, both in structured classroom settings and in less structured (by adults) computer clubs (see Levin & Kareev, 1980). In these settings, a critical factor for designing educational uses is the role of social resources, especially peers. People just starting to use computers have many low level problems, such as forgetting to hit the RETURN or ENTER key to terminate a response, typing the lower case letter l instead of the number 1, mistyping, etc. Any one of these problems can prevent further progress, and thus pose major design issues for educational programs. Yet in analyses of video tapes of computer novices, Levin found that few of these problems occurred, largely because several children worked together to use the computer. They interacted cooperatively to detect and correct these low level problems almost immediately.

Having more than one person use a computer for educational purposes also allows novices to divide up the task at higher levels so that each can master part of the required skills before moving on to tackle the rest. The side effect of having joint use is that the amount of cooperative peer interaction is increased through educational computer use, rather than decreased as commonly feared.

Levin described a number of heuristics that the members of LCHC have found important in designing educational computer activities:

(1) *Dynamic computer support:* Educational computer systems should initially take the initiative for large parts of the task to be mastered, but allow the learners to assume responsibility as they progress to expertise.

(2) *Dynamic social support:* Educational systems should enable and encourage interaction and helping among peers and between novices and experts.

(3) *Active/interactive:* Learners should take an active role in the activity.

(4) *Breadth:* Learners should have available a wide variety of educational microworlds (each of which exercise the skills they want to acquire) so that each can find a "world" in which they can become actively involved.

(5) *Power:* Educational activities should allow even novices to create interesting results with relatively little effort.

Michael Cole described how the flexible use of microcomputers in a rich social environment can lead to the creation of zones of proximal development, optimal regions for learning in which a person cannot perform the task to be mastered by him/herself but can with the aid of others. In this state, the individual can then internalize this social support, progressing to expertise.

Alan M. Lesgold
Learning Research and Development Center
University of Pittsburgh

Alan Lesgold emphasized that massive amounts of drill and practice are needed to acquire basic skills, and that the practice opportunities must be motivating, appropriate, and capable of providing immediate feedback. In particular:

(1) *Practice tasks should be appropriate to the child's level of progress.* One way to achieve this might be to use communications networks or plug-in modules to transfer diagnostic information and practice exercises between computers at school and at home.

(2) *Computer-based practice opportunities should provide understandable, productive, and immediate feedback which should take every error in performance into account.*

(3) *In a complex task, the computer can support one aspect of performance in order to allow higher-level practice of a second aspect.* Such "intellectual prostheses" can allow students to exercise advanced subskills (such as planning the structure of a story) before they have mastered all the earlier subskills (such as grammar and punctuation).

(4) *Practice environments should be motivating.* When computer-based games and instructional systems are commonplace, children may take for granted the "cute bells and whistles" that are motivating for them now. Perhaps computer-based instructional systems can be designed to increase the mental discipline and self-motivation of students. If success is to be rewarding, students must be able to recognize their own successes. A child cannot be motivated by success in writing essays, for example, unless he or she has an internal cognitive model of what a successful essay is.

Andrea diSessa
LOGO Project, Massachusetts Institute of Technology

Andy diSessa emphasized the importance of creating complete examples of good educational environments rather than just listing separate principles of instructional design. He described the LOGO programming environment at MIT (Papert, 1980; Abelson & diSessa, 1981; Papert, Watt, diSessa, & Weir, 1979) and related several examples of students' educational experiences with it. DiSessa highlighted the importance of creating educational environments where students have a sense of control, set their own goals, and are able to form deeply personal, long-term links with the material they are learning. He also advocated designing instructional environments in which the concepts to be learned are deeply embedded in the environment itself.

One of the examples diSessa described involved an environment in which students could learn about certain concepts of elementary physics. Instead of the "turtle" used in standard LOGO environments, students in this environment controlled what diSessa called a "dynaturtle". In standard LOGO, students draw lines on the computer screen by telling an imaginary "turtle" to turn in certain directions and move specified distances. In the dynaturtle environment, students control the motion of the turtle by "pushing" it with forces of specified direction and magnitude. The turtle then moves on the screen according to the laws of Newtonian physics as if it were an object on a frictionless surface.

One of the first surprises students have in this environment is that the turtle doesn't always move in the direction they push it. For example, if the turtle is moving upward and the student wants it to change direction and go sideways, he cannot just give it a sideways push. Instead, he must give it a push with a direction and magnitude that completely counteract the upward motion and also impart a sideways motion (see Figure 1).

DiSessa described how this environment could be used to impart an intuitive or phenomenological understanding of elementary mechanics that is very hard to get in traditional learning environments. The difficulty of achieving this kind of understanding from traditional methods is illustrated by the fact that the MIT physics students who played with the dynaturtle did nearly as poorly as the elementary school students tested (diSessa, 1981).

Allan Collins
Bolt Beranek and Newman, Inc.

Alan Collins discussed how computers can be used to create new environments in which there is intrinsic motivation to engage in reading and writing. He observed that only some kids get pleasure from reading, and very few get pleasure from writing. So for most children, the only reason to read or write is to satisfy the demands of teachers and schools. Then he listed a number

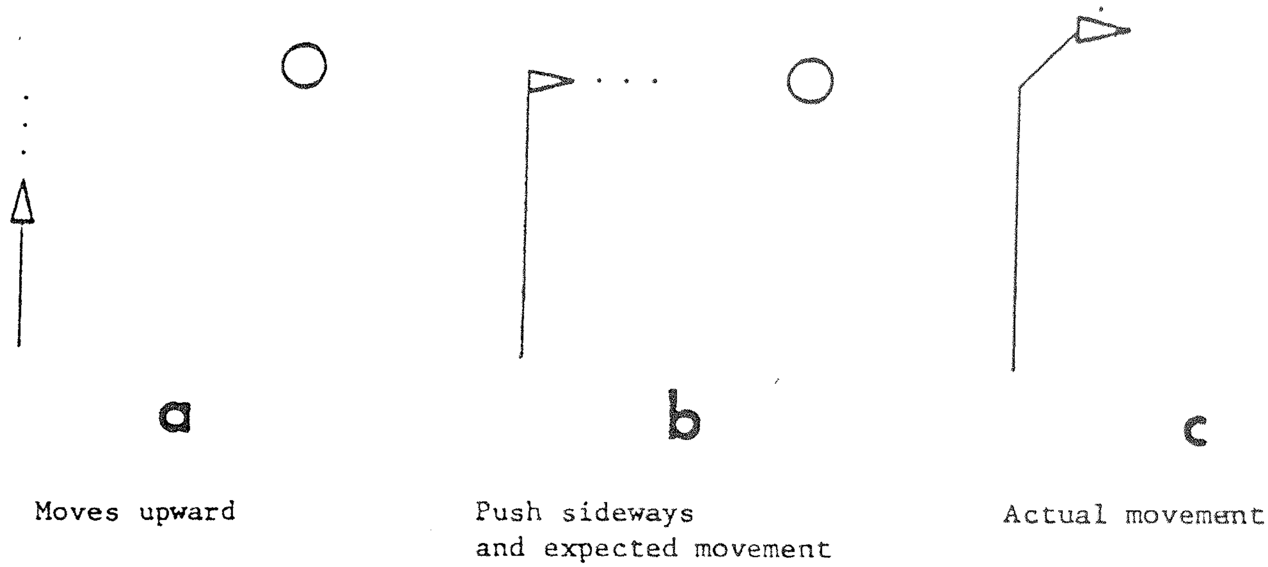

Figure 1a: A bug in a student's conceptions about motion

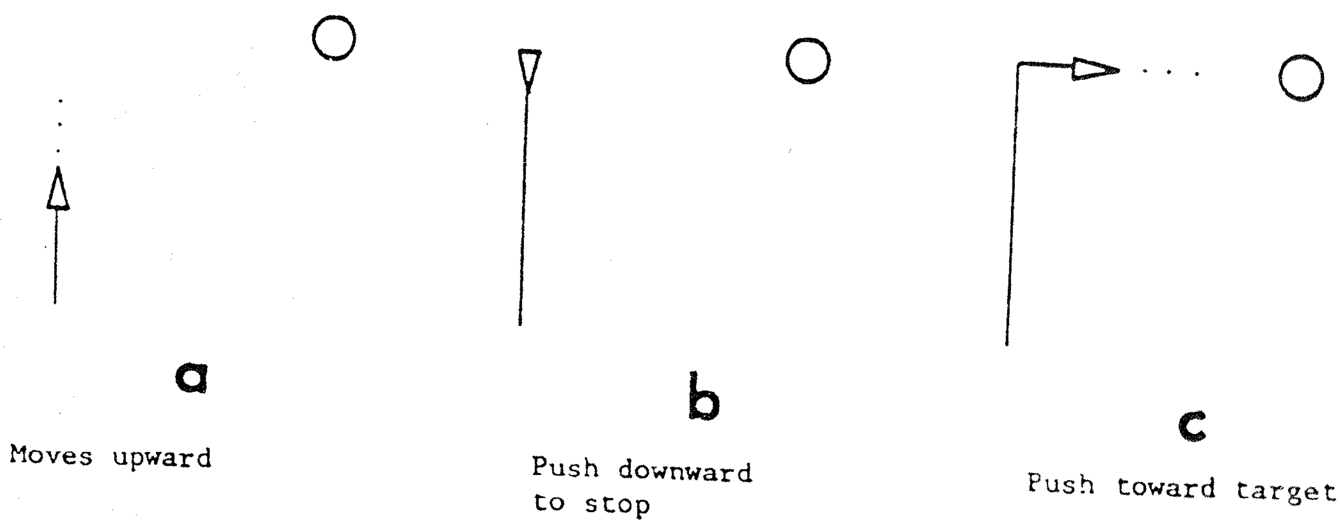

Figure 1b: A Newtonian method of negotiating a corner

of ways that computers can create environments where kids read and write for their own purposes rather than to satisfy a teacher.

For example, there are *game environments* (like Adventure) where kids have to read or write to play the game. Not all kids like games, but for those who do, these environments can lead them to want to be able to do the things other kids are doing, to compete with other kids, and to practice on their own. There are also computer-based *communication environments* like electronic message systems, bulletin boards, and newspapers, where the technology enables kids to communicate with distant friends, with kids they don't know, and with their own classmates. This use of computers is very widespread among adults with message systems and seems to be highly motivating. It provides, in a very natural way, lots of feedback about failures to communicate.

To illustrate the process of creating intrinsically motivating instructional environments, Collins discussed two applications of the principles proposed by Malone. First, he analyzed a motivating arcade game called "Missile Command" in terms of the features Malone described such as fantasy, audio and visual effects, and adjustable difficulty level. Collins also described several additional features of the game including the *hindsight principle.* This principle says that activities are more motivating when you can see at the end of them how you could have done just a little bit better. Collins then used the same features to analyze an educational game called "Textman" proposed by Andee Rubin. This game is a variation of Hangman where players try to select sentences of a particular text (e.g., a paragraph in a suspense story) out of a much larger list of sentences. Describing work he did with Andee Rubin, Collins showed how the Textman game as originally proposed had some--but not very many--of the motivational features of the arcade game. Then he described how the game might be improved by adding motivational features. The plan is to compare the two versions of Textman to see which works best with children.

Robert Davis and Sharon Dugdale
PLATO Project, University of Illinois

Bob Davis suggested three levels for categorizing instructional design principles:

(1) *Micro level* involving the minute details of screen layout, control key meanings, and so forth. This level is usually not noticed unless it is done poorly.

(2) *Obvious or memorable level* involving the themes and instructional content of the material.

(3) *Macro level* involving the structure of the curriculum over the year and its relationship with other aspects of the school and classroom.

Davis also suggested several guidelines for designing meaningful instruction. One technique used in PLATO lessons involved storing examples of students' work for other students to see (Dugdale, 1979). For example, in one instance of a "library lesson" children were asked to divide a square into pieces and then color a specified fraction of it. Successful solutions to the problem could be stored in a library that other children could access. The availability of previous solutions inspired many students to try more and more original or artistic solutions including making patterns and spelling words with the colored portion of the square. Since a correct solution had to equal the specified fraction, these original solutions required a thorough understanding of the notion of equivalent fractions.

In lessons with a "Hall of Fame" (such as the Green Globs game described below), the best solutions are stored for others to see. Since the complete games of high scorers are stored, other students can often engage in a kind of "industrial espionage" to learn the secrets of playing the game well (and incidentally of the skill the game is designed to teach). Thus this technique not only takes advantage of a form of competition, it also allows children to learn from their peers.

Sharon Dugdale discussed the use of "intrinsic models" in designing instructional materials. Intrinsic models are instructional activities in which the student explores and manipulates a working model that provides meaningful and constructive feedback. The mathematics to be learned is intrinsic to the model and is treated as inherently interesting, rather than hidden behind irrelevant themes. The activities are designed to be engaged in by students of widely varying backgrounds and abilities, but students find that the more math they apply, the better they do. Dugdale also suggested that:

> (1) Materials for student use should be designed to engage the student in productive thought and activity rather than to showcase the capabilities of the hardware or the author.
>
> (2) They should draw on the inherently interesting characteristics of the topic rather than trying to hide the topic under a lot of hoopla (e.g., graphics, animations, and music that are unrelated to the task).
>
> (3) They should keep the student interacting and participating rather than passively watching and listening.

Finally, Dugdale described a new example of an activity that illustrates the use of the "intrinsic models" characteristics, as well as some social aspects of courseware design and usage. This lesson, called Green Globs, is described in the synthesis section below.

John Seely Brown
Cognitive and Instructional Sciences Group
Xerox Palo Alto Research Center

John Seely Brown first described an example of how extremely subtle aspects of an instructional environment might have very important effects. The example involved the input format on an early microcomputer version of the Darts game (derived from the original Plato game described below). Careful observation of a number of students playing this game suggested that they learned to control the position of the arrow by treating the input as a three argument function (integer, numerator, and denominator) without any understanding at all of the meaning of fractions. For instance, they might have learned that increasing the third number makes the arrow go down without understanding anything about how the denominator specifies the number of equal parts in a whole.

In the version of the game in which this phenomenon was observed, students had to type a carriage return after each of the three parts of the answer. One conjecture is that these explicit delimiters between parts of the answer might have heightened the tendency for students to see the answer as three separate entities rather than as a single mixed number with meaningful parts. This conjecture is supported by the fact that the phenomenon was not observed in a later microcomputer version of the game that used the conventional implicit delimiters (as in the original Plato version): whole number, (space), numerator, (slash mark) denominator.

Next Brown discussed a detailed set of principles for "coaching" students in informal learning environments. These principles are explicitly encoded in a computer-based coaching system for an arithmetic game (see Burton & Brown, 1979). They assume that the student takes turns playing the game against another student or against the computer. The computer constructs a model of the student's skills and weaknesses by observing when the student misses good moves in the game. This student model is based on a set of Issues which students can learn about from playing the game. The principles include the following (for a more detailed discussion, see Burton & Brown, 1979):

(1) Before giving advice, be sure the Issue used is one in which the student is weak.

(2) When illustrating an Issue, only use an Example (an alternative move) in which the result or outcome of that move is dramatically superior to the move made by the student.

(3) If a student is about to lose, interrupt and tutor him only with moves that will keep him from losing.

(4) Do not tutor on two consecutive moves, no matter what.

(5) Do not tutor before the student has a chance to discover the game for himself.

(6) Do not provide only criticism when the Tutor breaks in! If the student makes an exceptional move, identify why it is good and congratulate him.

(7) After giving advice to the student, offer him a chance to retake his turn, but do not force him to.

(8) Always have the Computer Expert play an optimal game.

(9) If the student asks for help, provide several levels of hints.

(10) If the student is losing consistently, adjust the level of play.

(11) If the student makes a potentially careless error, be forgiving. But provide explicit commentary in case it was not just careless.

Laura Gould
Learning Research Group
Xerox Palo Alto Research Center

Laura Gould described a system named TRIP for animating algebra word problems (see Gould and Finzer, 1981). TRIP is intended for students who have mastered the mechanics of algebra but have difficulty translating the English text of the problem into suitable algebraic expressions. TRIP provides an environment where students can develop an intuitive grasp of time-rate-distance problems and their algebraic representations. The system supplies a helpful graphical interface by means of which students construct a diagram of the problem using high-resolution pictures of places, travellers, speedometers, odometers, and clocks (see Figure 2). Once the diagram is judged by the system to be correct, the system asks the student to make a rough guess of the answer. Then the travellers, meters, and clocks all move together producing an animated representation of the problem. When the state specified by the student's guess is reached, the action stops, and the student gets to see the result of the guess. A record of each successive guess and its consequences is kept in a table from which students induce algebraic expressions and finally an equation for computing the answer.

Gould pointed out some of the successful and unsuccessful aspects of the system in an attempt to discover the underlying design guidelines. The user interface for the system, although apparently quite complicated, was successfully and easily controlled even by students with low confidence levels. This seemed to be due to (1) the extensive help facility (accessed by a large "Help" button always visible on the screen), (2) the fact that students started with a nearly blank screen and built up for themselves a complex representation of the problem, and (3) the consistent use of the same functions in different parts of the system. TRIP was also succesfully integrated into an existing classroom curriculum. This was aided by (1) involving teachers in the early design phases, (2) tailoring the computer curriculum to mesh well with what was going on in the class, (3) individualizing the problems by choosing easy or hard numbers based on each student's ability, and (4) collecting feedback from the students using a "gripe" facility.

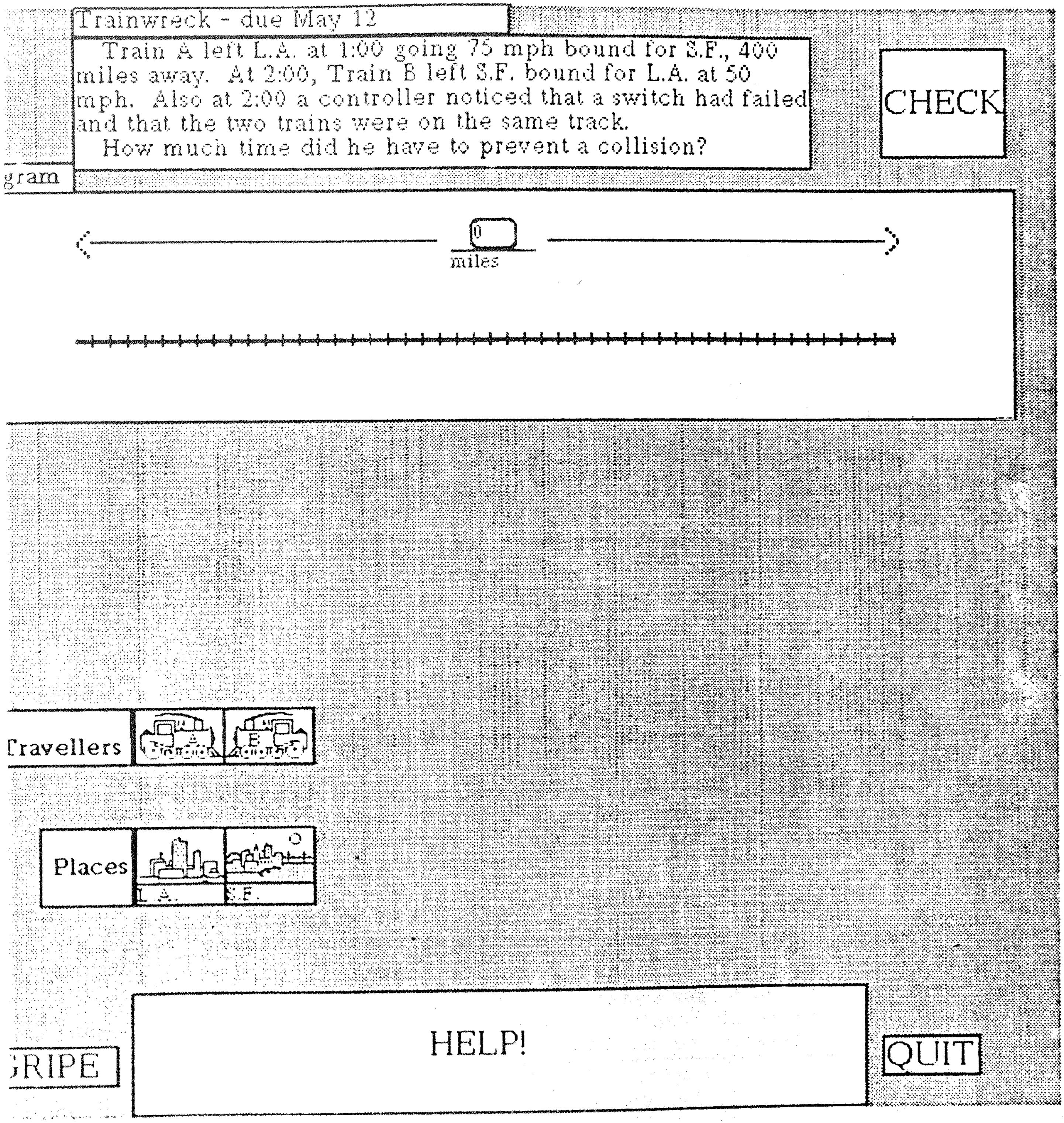

Figure 2a: An initial screen configuration in the TRIP system.

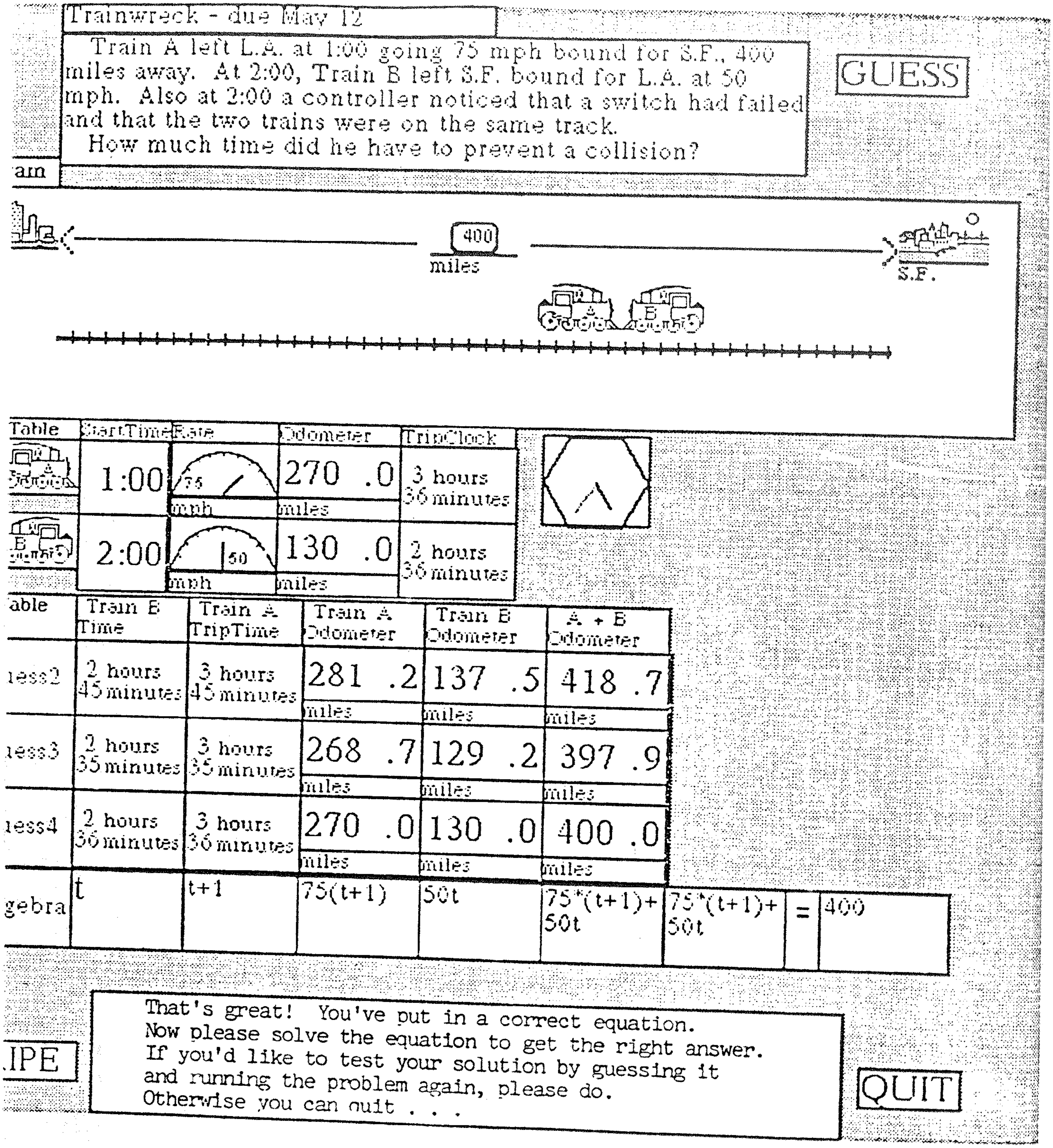

Figure 2b: A final screen configuration in the TRIP system

Students who used the TRIP system learned to make good diagrams of algebra word problems, but they continued to have difficulty actually solving the problems, especially when constructing the "guess table" and forming algebraic expressions. This seemed to be due in part to the fact that the system did too much of the work for them--it automatically constructed table headings and ran the clocks and odometers so that the students were not required to compute these values. The students could presumably have profitted from doing more of this work themselves. They could also have been aided by a help system that included substantive coaching in algebra, rather than one that just provided help in how to use the system.

Marge Kosel
Minnesota Educational Computing Consortium

Marge Kosel suggested that an instructional program designer can be compared to a sculptor working with a piece of stone. Creativity, knowledge of the media, and consideration of the audience are parts of both processes. Just as it would be difficult to define all the components needed to create a piece of sculpture, there are distinctive qualities in the design of a microcomputer program that are brought out by the designer and the subject. As in sculpting, however, certain sub-skills of expert performance can be defined.

Kosel then summarized a number of these principles using the three levels suggested earlier by Davis:

(1) The first level can be called the *invisible level* since it is not noticeable unless poorly designed. This level can be defined through explicit rules or checklists such as the following (see MECC, 1980, for more examples): (a) Avoid crowded displays. (b) Vary the way the text is presented, using boxes and lines. (c) Avoid jumping scrolling of a screen of text. (d) Let users signal when they are through reading; do not use standard timing loops.

(2) The second level, which can be called the *memory level,* is what the user remembers after completing the program. This level includes concepts or themes used to enhance learning, design of user controls in the program, effective use of graphics, color, and sound, and attention to the use of appropriate reading levels.

(3) The third level, which may be called the *educational level,* is concerned with questions such as whether the material is educationally sound, whether the capabilities of the computer are used in an appropriate way, whether the material will fit in different classroom situations, and what types of support materials are necessary. This level also

examines whether the system will be used by individuals, by small groups of students, or by an entire class.

Each level is integral to the final product, and although guidelines and principles to be followed can be defined at each level, the success of the final product depends ultimately on the creativity of the individual author and the ways in which the subject matter is interpreted on the screen.

Jim Hollan and Ed Hutchins
Navy Personnel Research and Development Center

Jim Hollan and Ed Hutchins presented two examples of microprocessor-based systems they are developing as part of research efforts concerned with the application of microprocessor technology to navy training needs. One system is based on semantic network databases that represent the information students need to learn. In addition to simply viewing the information in the database, students can play a variety of games (e.g., flash-card and twenty questions) with the information to be learned. The games and exploration facilities are independent of particular databases and thus the system can be used for instruction in a wide variety of domains. Automated facilities are available to assist in the construction of new databases. The system is currently being used to teach a large body of information about the characteristics of various kinds of naval equipment.

The other system provides conceptual instruction and simulation-like practice in the use of an important and conceptually difficult piloting tool called a maneuvering board. In this system, the student can view and control continually updated geographical and relative motion depictions of ships maneuvering in proximity to each other.

Hollan and Hutchins also discussed several other issues involved in instructional design:

(1) *Instruction begins with elicitation.* It is only when one has a good cognitive task analysis that one can decide how to employ a particular technology to improve instruction.

(2) *Is the technology appropriate*? In many domains, the computer is not better than other means of fulfilling educational goals, and in those cases it should not be used. In both of the examples described above, the computer provided instructionally superior facilities that were not available in other media.