
The Apple
Journal of Courseware Review

\$5.95

Educator's Resource Guide

Issue 1

Apple II
Microcomputer programs

Communication: The Computer and You

Two Creative Authoring Systems:
Create, Delete or Amend a Lesson
Unique Lesson Formula

Touch Typing, An Instructional Game

Typing Tutor

Math Skill Builders

BASIC for the Computer Novice

Self-Instruction in Programming

Elementary Exercise in Ecology

Simulated Physics Laboratory

A Sail to the New World

Logic and Reasoning in Reading

Comprehensive Reading Power

Computer-Managed Spelling

Sentence Structure Simplified

Clock and Calendar Skills

Reviewer's Guidelines

Journal Index



The Apple Education Foundation

Dear Educator:

Thank you for your interest in the JOURNAL of Courseware Review.

One of the primary goals of the Foundation is to foster excellence in computer-assisted education. Through our publications, the Foundation seeks to establish an impartial source of reference on quality microcomputer courseware. By recognizing programs that demonstrate excellence in design and development, and identifying programs that are effective in learning environment, the Foundation hopes to encourage and set leadership examples for others to follow.

The JOURNAL of Courseware Review, Volume 1, Number 1, (Red) was published in October of 1981. The second publication, called The Apple JOURNAL of Courseware Review, Issue 1 (Blue), was announced in July, 1982. Though the names differ slightly, each publication follows the same format and features critical reviews of selected, commercially available educational programs. The JOURNAL's qualitative articles are written by educators and instructional designers, based upon their own professional knowledge and teaching experience. We encouraged reviewers to identify models of excellence in program design, to highlight innovative techniques, and to comment upon a program's strengths and weaknesses, as well as application to microcomputer technology.

Programs resulting from the Foundation's grant awards will also be announced and reviewed in the Apple JOURNALS as they are completed and become available for distribution.

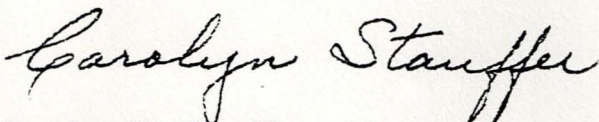
Both JOURNAL issues may be OBTAINED FROM YOUR LOCAL APPLE DEALER. As it takes time to distribute new products, in some areas, you may have to request an order. Ask for:

The JOURNAL of Courseware Review Vol.1, No. 1 (Red)
Product # A2L0061

The Apple JOURNAL of Courseware Review Issue 1. (Blue)
Product # A2L0062

The suggested retail price is \$5.95 per issue. There are NO SUBSCRIPTIONS available and NO PURCHASE ORDERS can be accepted by the Foundation. The JOURNAL is available only through dealers at this time. However, as distribution is established, the JOURNAL will be carried by major book stores.

Sincerely,



Carolyn M. Stauffer
Editor

The Apple Journal of Courseware Review

Editor/Publisher

Carolyn M. Stauffer

Editorial Advisory Board

Dr. Kenneth E. Brumbaugh, Director
Minnesota Educational Computer Consortium

Dr. Judith B. Edwards-Allen, Director
Computer Technology Program

Dr. J. Dexter Fletcher, Executive Director
WICAT Research Institute

Dr. Arthur Luehrmann, Director
Computer Literacy, Inc.

Dr. David Moursund, Professor
Computer Science Department
University of Oregon

Dr. James L. Poirot, Chairman
Department of Computer Sciences
North Texas State University

Dr. Karl Zinn, Research Scientist
Center for Research on Learning and Teaching
University of Michigan

Foundation Staff

G. Gregory Smith, Director

Carolyn M. Stauffer, Manager, Evaluation Centre

Margaret C. Redpath, Grants Administrator

Frances L. Carney, Associate

Consulting Editor

Jude Costello

Consulting Librarian

Ann Lathrop

•

Permission to Reprint

The Apple Journal of Courseware Review articles may be reprinted for publications originating at educational institutions, personal computer clubs, or non-profit associations. Each reprint must carry the following copyright notice on the front page: © 1982, Apple Computer, Inc.

Trademarks of Apple Computer are: Apple and Special Delivery Software.

The Apple Journal of Courseware Review is available through Apple computer dealers and major book stores. No subscriptions.

Table of Contents

Successful Student/Microcomputer Interactions	3
Z.E.S. The Zenith Educational System	8
The Eureka Learning System	13
MasterType	17
Typing Tutor	21
Arcademic Skill Builders in Math	26
Applesoft BASIC Tutorials	31
Programming in Apple Integer BASIC: Self-Teaching Software	35
Elementary, Volume 4	39
Physics CompuLab	43
Search Series: Geography Search	47
MicroSystem80: Critical Reading	51
Comprehension Power	56
Compu-Spell	62
Sentence Diagramming	66
Clock	71
Calendar Skills	73
Guidelines For Journal Review	77
Journal Index of Courseware Reviews	78

Note from the Editor

Our second publication of *The Journal* has a significant addition to its name, adding impact to the Foundation's efforts to foster excellence in computer-assisted instruction. Apple's *Journal of Courseware Review* will expand its efforts to provide its readers with decision-making information on quality courseware selection, creative design, and educational application.

Judging from the responses received following the publication of our first issue, there is a growing number of computer-literate educators seeking reliable answers to questions relating to courseware and who welcome such information from their peers. Until now, few sources existed to guide the educator through the maze of software selection. It was gratifying to have readers mention that *The Journal's* reviews listed measures they could use for identifying effective and reliable programs, thus providing a clearer idea of just how computers could best be employed in the educational environment. *The Journal* outlined maximum standards for quality microcomputer design which enabled readers to develop a more critical eye when searching voluminous courseware directories. Some of the most rewarding comments received were from courseware vendors and publishers who mentioned modifications they had made, based on *Journal* reviewers' suggestions.

It is no small task to develop quality, instructional microcomputer programs. Literally thousands of hours are involved in developing a concept and utilizing the expanded capabilities of electronic-assisted instruction. It is a cooperative venture between educators and developers to produce innovative, workable, quality programs that do, in fact, improve the educational process.

For your interest, we have included in this issue the guidelines sent to invited reviewers to assist them in the evaluation of selected courseware for *The Journal*.

Carolyn M. Stauffer
Editor and Publisher

Successful Student/ Microcomputer Interactions

Jamesine Friend and
James D. Milojkovic

*A closer look at the art of conversation
between student and computer.*

There are many component decisions that contribute to the educational value of an interactive, educational computer program. This article provides the reader with some pedagogical and programming considerations as guidelines for judging the quality of the communication between the student-user and the computer.

Jamesine Friend is an instructional designer with many years' experience in the design and implementation of computer-assisted instruction. As a member of the Stanford University faculty, Friend is associated with the School of Education's Interactive Educational Technology. Her research interests include the development of a theory of computerized testing and the design of intelligent computer-assisted instruction (ICAI).

James D. Milojkovic, a doctoral candidate in the Department of Psychology at Stanford University, is also pursuing a master's degree in Interactive Educational Technology. Milojkovic's research is directed toward the cognitive and motivational impact of microcomputers in education and the social psychology of computer use.

An educational computer program consists of a series of exercises in which the student interacts with the computer. The quality of the program is dependent upon the content of the exercises, the order in which the exercises are presented, and the nature of each interaction between the student and the computer. The first two of these (content and sequence) are the heart of the instructional design problem and are closely related to traditional curriculum design, while the third (interaction) is unique to computerized instruction and includes a number of programming as well as pedagogical considerations. This article focuses on such considerations, and presents some guidelines for teachers and developers in evaluating the effectiveness of student-computer interactions.

Each exchange in an interactive educational program contains four fundamental parts:

1. The computer presents an exercise to the student.
2. The student responds.
3. The computer checks the student's response.
4. The computer provides feedback to the student.

Each of these components will be discussed in turn.

Presentation of the Exercise

Conciseness The statement of exercises should be as short as possible while conveying all necessary information. As a rule, there should be no more than about 100 words per exercise for adults and no more than 20 words for elementary-school children.

Reading Level The reading level should be appropriate for the target population. Professional courseware designers should access readability formulas to ensure the use of appropriate vocabularies. Prior to distribution, students' reading levels should be tested in the classroom; students should be asked to read exercises aloud and then be quizzed on the meaning of what they have read.

Response Mode It must be clear to students how they are to respond. Are they to type their answers, touch the screen with a light pen, or draw on a graphics tablet? A program should require the same kind of responses throughout. If there is a necessary change in mode of response, the instructions must be clear, and there should be as few such changes as possible.

Terminators After a response is typed, most programs require that the students signal the end of their answers by pressing a "terminator" such as the ENTER key or the RETURN key. The use of a terminator should be consistent. Students should not have to use the spacebar in one exercise and the ENTER key in the next. If the student uses a number of different programs, ideally there should also be consistency among those programs. When a terminator is required, students must be given explicit instructions regarding its use. While there is no need for the terminator instructions to be repeated with every exercise, the best programs include a reminder with the first few exercises.

Asking Questions A well-designed program will not mislead students into entering inappropriate responses. The question "Do you know how to multiply 12 times 5?" may result in some young children giving a literal answer like "yes" to mean "Yes, I do know how to multiply those numbers." The direct question is better, e.g., "How much is 12 times 5?" Almost everyone—including young children—understand that they are to answer either *yes* or *no* to the question "Is Georgia one of the New England states?" There is no need for the specific instruction "Type 'yes' or 'no' to answer this question."

Some programs allow abbreviations of commonly used answers ("Y" for "yes," and "T" for "true," and so on). If most exercises are yes/no or true/false questions, this can be quite a time-saver, especially for children who cannot type well. However, if very few of the questions are of this binary form, the time saved may be lost by the child who has to read and puzzle over the instruction "Type 'Y' for 'yes' or 'N' for 'no'."

Multiple-choice exercises or simple yes/no questions, do not ordinarily need specific instructions since all but the very youngest children are accustomed to this kind of exercise. Indexed multiple-choice exercises are overused in educational programs, undoubtedly because the answers are easy for the program to check. It is often the case, however, that little learning takes place with multiple-choice exercises in comparison with what can be gained from more freely constructed responses. Consider

the example in Fig. 1. Contrast this with the more open-ended question in Fig. 2.

Checking the student's answer for such open-ended questions is more of a programming problem, but the extra programming effort usually results in much greater learning gains.

There are two cases when multiple-choice exercises are clearly indicated. One is when the response would require a longer answer than an inexperienced typist could comfortably enter. The second is when the nature of the task is one of discrimination (Fig. 3), and the questions cannot be rephrased to require a constructed response without basically altering the nature of the exercise.

Screen Layout—Text The layout of an exercise on the screen can often affect the student's understanding of what is expected of him, as well as his attitude toward the program and the computer. The video-screen display should be aesthetically pleasing and arranged to make the task of solving the problem as clear as possible. For greatest readability, text should be displayed double spaced in both upper- and lowercase letters. It should not be crowded at the top or to one side of the screen. It is essential that all words be spelled correctly, that sentences be grammatically correct, and that text be properly punctuated. One would think this obvious, except for the large number of violations in programs that developers apparently considered to be finished.

Screen Layout—Graphics Graphics should be used sparingly since good pictures are difficult to design and are time-consuming to display. Graphics should be used only if they convey some important information that is not included in the text. If graphics are used, the same information should not be repeated in the text; for instance, if the student is to determine whether there are more dogs or more cats in a picture at the top of the screen, there is no need to display the sentence "Here is a picture of some dogs and some cats." The "readability" of graphics should be field tested to ensure that they contribute to the learning process, rather than interfere.

The Student Response

As indicated earlier, student responses may be of several kinds: via the keyboard, a light pen, a graphics tablet or, sometimes, a voice-recognition device. Keyboard responses are by far the most common, and we restrict our discussion to them.

Standard Input Routines The cursor, or answer prompt, is usually displayed where the student's

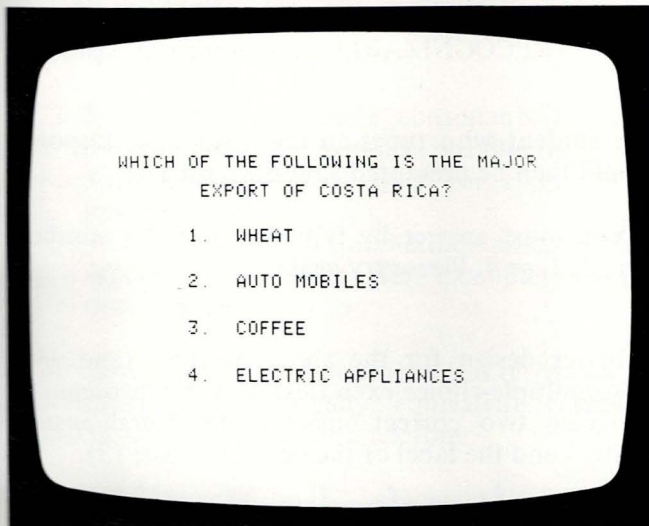


Figure 1

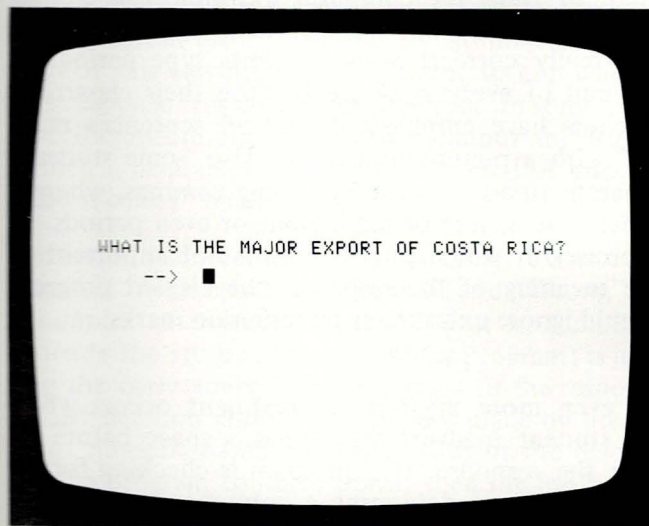


Figure 2

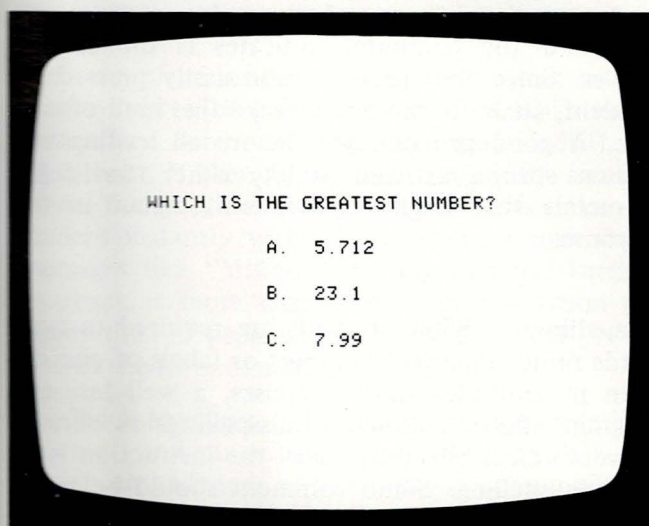


Figure 3

answer is to appear. As the student types, each character appears in place of the cursor, and the cursor moves to the right. Use of a fixed terminator (e.g., pressing the RETURN key) signals the end of the response.

Standard keyboard-response routines are not always applicable. For example, some exercises are best answered by typing characters from right to left, as in the arithmetic problems (Fig. 4).

Erasures Most programs use a DELETE, BACKSPACE or Backarrow key (\leftarrow) to accommodate corrections. Erasures may be character by character, in reverse order, or by entire response. For young children who are typing only one or two letters, it is less confusing for them if the entire response is erased; but for older students, character-by-character erasures are best. In either case, the erasure should be accomplished by eliminating the characters from the screen and moving the cursor (if one is used) back to the beginning of the "erasure".

Timing Responses Many computer programs set a time limit for student responses, a design with both advantages and disadvantages. Unless the time allowed is appropriate for the individual student, it causes frustration. Sophisticated programs do not impose an explicit time limit. They simply record internally the time taken to complete a response and use that information in future decisions about the student's need for more practice. Time limits are generally appropriate either when the material being presented must be learned to a very high level of skill, or as an artificial means of enhancing the interest of the program.

Checking the Student Response

Response Classification The purpose of checking a student response is to determine whether or not it is correct. One simple procedure used to compare the student's response is to precode correct answers and then to classify the response as correct, if it is an exact match, and as incorrect otherwise. In general, this is poor educational-program design since it can never be certain that a mismatch is really incorrect. The student may enter the correct answer in principle but in an inappropriate form. In the Costa Rica example (Fig. 1), if the program expects the answer "3" and the student types "coffee," he is informed his answer is incorrect. In this case, it would be better if the program classified student responses as:

1. CORRECT (an exact match with the precoded correct answer, 3)

2. INCORRECT (any of the numbers 1, 2, or 4)
3. UNRECOGNIZABLE (any other response)

The student who types an unrecognizable response should then be presented a message such as:

You must answer by typing one of the numbers 1, 2, 3, or 4. Please try again.

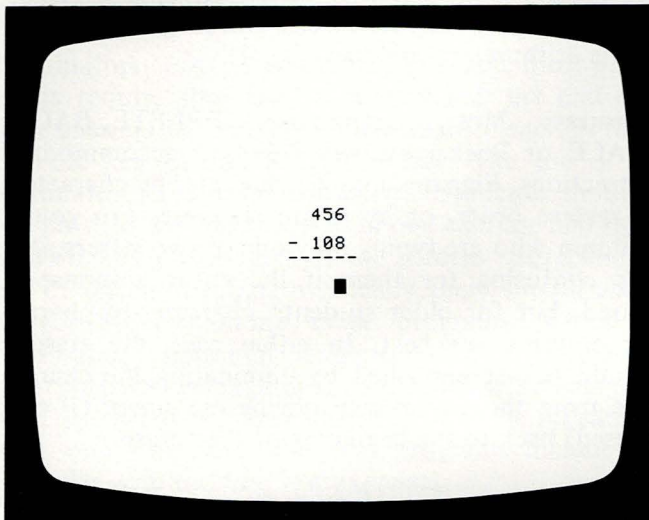


Figure 4

A better design for the above example (and with most multiple-choice exercises) is for the program to recognize two correct answers, the literal answer (coffee) and the label of the correct choice (3).

Students occasionally type unexpected punctuation marks, such as periods at the ends of responses. Note the example in Fig. 5. From the student's point of view, the computer was indicating to him that the answer was wrong when what was typed was really correct! Some students type periods at the end of every response because their classroom teachers have emphasized that all sentences must end with a punctuation mark. Also, some students separate items in a list by typing commas, whereas others use spaces or semicolons or even periods. In exercises in which punctuation is not important to the meaning of the response, the elegant program should ignore extraneous punctuation marks.

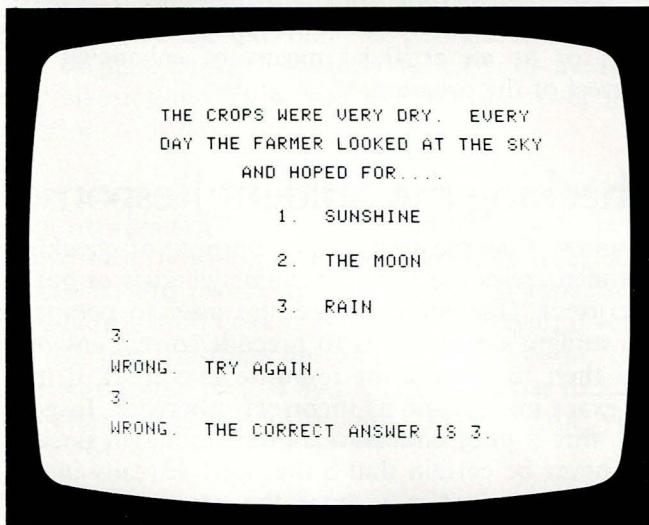


Figure 5

An even more mysterious treatment occurs when the student inadvertently types a space before or after the response. If a program is checking for an exact match to determine a correct response, a response like "car " will not match "car" since one string of characters contains a space that the other does not. However, the display on the video screen usually looks just the same, so the student is unable to discern the difference between his response and the word the computer indicates is the correct answer. Since the spacebar is so easily pressed by accident, students can easily make this kind of mistake. A good program will ignore all leading and trailing spaces, as well as any other "invisible" characters that might inadvertently occur in the response.

Misspellings When students are required to type words rather than just numbers or labels of choices given in multiple-choice exercises, a well-designed program allows reasonable misspellings as correct answers unless one purpose of the instruction is to improve spelling. Some comment should be made about the correct spelling, without penalizing the student by classifying the response as incorrect.

A good program will take account of the more common spelling errors:

1. Substitution of double consonants for single and vice versa (for instance, apealling).
2. Substitution of one vowel for another (e.g., inadvertant).
3. Transposition of letters—either misspellings or typographical errors (for example, infromation or beleive).

The most sophisticated programs also recognize substitutions of phonetically equivalent consonant clusters, such as reseive and telephone.

Providing Feedback to the Student

Feedback vs. Reinforcement When the student enters and terminates his response, it is classified. The program usually informs the student of the result of this classification, indicating to him whether his response was correct or incorrect. In some cases, specific information is provided about the way in which a response was wrong. This kind of information is called "feedback".

Informational feedback may be either extrinsic or intrinsic. Messages like "Correct" and "Wrong" are examples of extrinsic feedback; that is, something outside the student (the computer program) is judging the correctness or incorrectness of the response. If the decision about correctness is made by the student himself, based on observation of the reaction of the program to his response, then the feedback is "intrinsic."

Extrinsic feedback is artificially imposed and thus external to the student, whereas intrinsic feedback is internal to the student. Intrinsic feedback is considered more effective.

For example, consider a game of sea rescue where a lifeline has to be projected with a certain student-determined force and angle to a sailor adrift in a small boat. If the lifeline fails to reach the stranded sailor, the graphics display communicates this to the student instantly without the need for superfluous messages like "Missed" or "Try harder." Intrinsic feedback is more characteristic of simulation programs than of programs that provide direct instruction in the form of a dialog or drill.

Feedback Messages Feedback messages should be succinct and unambiguous. Neither graphics nor sound are advised as a means of communicating success or failure—unless they are part of a simulation program such as the sea-rescue example. Graph-

ics and sound can be distracting both to the student and his neighbors, taking time that would be better spent in more direct educational activities.

"Correct" is ordinarily sufficient feedback for correct responses. This may be varied by using words like "right" or "good," but excessively enthusiastic messages like "super" or "fantastic" are more often than not inappropriate. Some successful computer programs simply present the next exercise as soon as a student gives a correct response, providing feedback only for incorrect answers. Students soon learn that the presentation of another exercise signals a correct response.

"Incorrect" for wrong responses is usually sufficient to convey the desired information, especially for questions that have a small number of choices such as yes/no, or true/false questions, and multiple-choice exercises. If there is a greater variety of possible answers, it can be more educational if the student is informed of the nature of the error. Many programs offer students a second chance to respond before providing detailed information about an error; whenever there is a good possibility the student made a simple error that he can correct without extra help, this is a good idea. However, if the student cannot answer correctly, even when given a second chance, he should be told the correct answer. For young children, an overt correction (that is, typing the correct answer after it is shown) will focus their attention on a correct answer that they might otherwise ignore in their eagerness to get on with the next task.

For many students, the very act of producing an answer seems to generate a small but positive self-reinforcement, independent of the feedback from the program. Negative feedback from the program needs to be more forceful than positive feedback, simply to overcome this natural self-reinforcement.

Conclusion

There are many component decisions and features that contribute to the educational value of an interactive education program. We have presented some of the basic considerations as guides for teachers and developers in judging the quality of the interactions between the student and the computer. Developers must learn to gauge the quality of their program design by field testing it with students—observing their performance, testing, and listening to their comments and suggestions. Educators should be selective and voice their requirements for quality standards in microcomputer courseware design.

Z.E.S.

The Zenith Educational System

Avant-Garde Creations
and

The Eureka Learning System

Eiconics, Inc.

*Jerry W. Sprecher and
Jack A. Chambers*

*Introduction and evaluation of two
authoring programs.*

Authoring programs hold considerable promise for expanding the dimensions of electronic-assisted instruction. Such programs provide teachers with a variety of authoring capabilities, lesson assignment control, and convenient student-record management. As an alternative to using packaged curricula with pre-defined sequences, which may or may not

correspond with classroom activities, teachers now have the means of creating supplemental lessons that complement individual or class instruction. With little or no programming knowledge, teachers/authors can be prompted through a series of steps to key in lesson items, comments and reinforcements that result in a complete student lesson.

Authoring systems represent one approach to resolving the current critical shortage of quality, tested, instructional computing materials through enabling educators themselves to develop computer-based lessons. It is generally accepted that authoring systems should: 1) permit educators with limited computer backgrounds to develop lessons; 2) be appropriate for multiple instructional models (i.e., drill and practice, tutorial, simulation, etc.); 3) provide an effective delivery component; 4) include a management system; and 5) support the technical capabilities of the microcomputer. These five components, therefore, provided the basis for the following evaluation of two authoring systems, the Zenith Educational System (Z.E.S.) and the Eureka Learning System (ELS).

Evaluation Procedure

The evaluation procedure was identical for the two authoring systems and utilized a structured presentation approach. As a first phase, instructional laboratory assistants with strong backgrounds in computer-assisted instruction and familiar with the Apple II microcomputer were responsible for familiarizing themselves with each of the authoring systems to be evaluated. In the second phase, the laboratory assistants demonstrated the authoring systems to a group of six educators, having experience in instructional computing, who evaluated the programs using a standard rating form.

The evaluation form contained 97 items to be rated on a scale of 0 to 10 followed by an overall rating of each system.

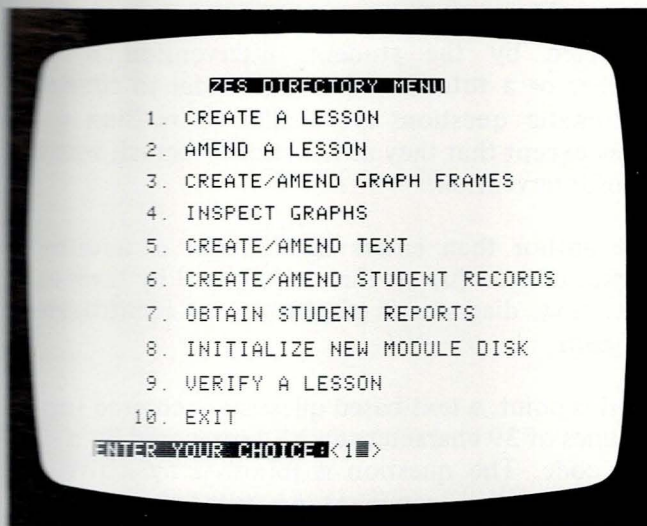


Figure 1

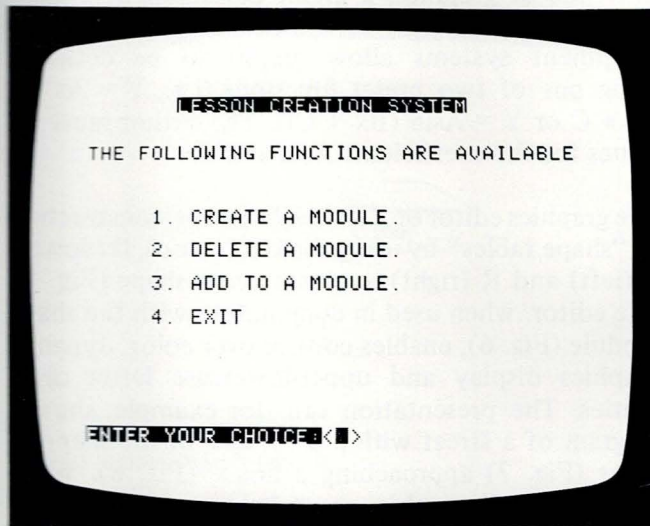


Figure 2

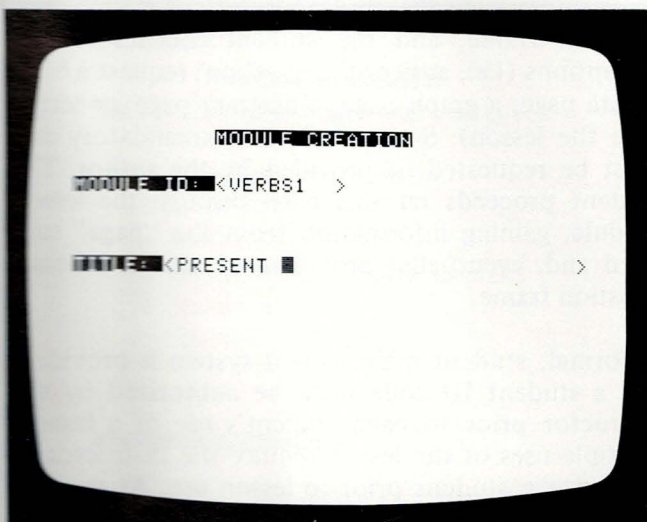


Figure 3

The evaluation was targeted for prospective authors with backgrounds sufficient to understand moderately complex computer systems. Therefore, individuals with limited backgrounds in computing may experience difficulties not reported by the evaluators. Similarly, individuals with more specific knowledge of the particular authoring systems evaluated may have avoided obstacles reported in the evaluation.

Zenith Educational System

Overview

The *Zenith Educational System (Z.E.S.)* is documented in an attractive, professionally printed, loose-leaf binder. In general, the documentation is clearly presented and reasonably easy to follow.

In terms of classification, the *Z.E.S.* system is menu driven in that it provides the lesson author with a list of options at each step. The *Z.E.S.* package consists of three diskettes: 1) the program diskette, which contains the author-oriented (development) programs; 2) the module diskette, which contains the actual student delivery system (lessons); and 3) the graphics diskette, which contains the authoring program for developing graphics for a lesson.

A teacher, as the author, may create, delete or amend a particular lesson. To create a new lesson, the author first enters a lesson name and identification. Following directions, the author can enter a series of frame-oriented presentations. Each frame (single-screen display) consists of a question with up to four anticipated answers (of 19 typed characters each); up to 15 lines of text; a hint; a comment; and additional frames that can include a graphics display, diagram or additional text frames.

As an example of the authoring process, a lesson creation consists of "booting" (starting) the program diskette, selecting the desired option from the menu (e.g., Create a Lesson) and waiting for the program to load (Fig. 1). The author then removes the program diskette and inserts the module diskette. The module diskette provides the options to create, delete or add modules to a lesson (Fig. 2), to create a lesson, a lesson title and identification code (Fig. 3). Questions are then entered into a frame in one of three question modes: normal, revision, or automatic. Nine revision questions can be associated with a normal question. An incorrect response on a normal question can force a branch to the list of revision

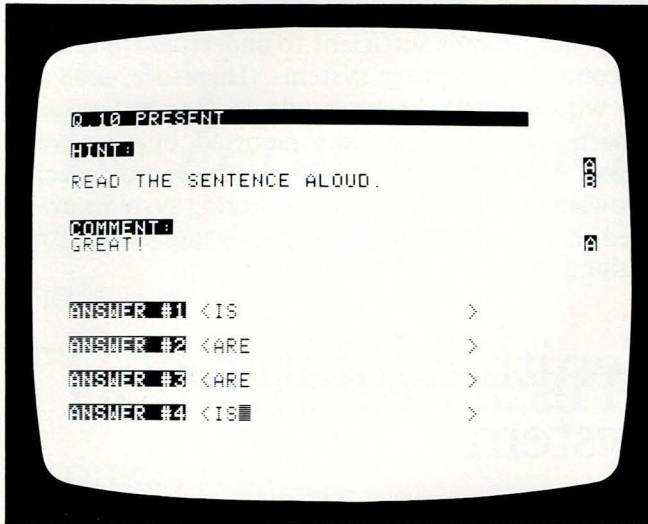


Figure 4

questions. If revision questions are incorrectly answered by the student, intervention by the teacher or a tutor is required in order to continue. Automatic questions are similar to revision questions except that they automatically branch without tutor intervention.

The author then enters the number of additional frames desired (up to three), followed by their type (i.e., text, diagram or graph) and an identification for each.

At this point, a text-based question is entered (up to 15 lines of 39 characters each), terminated by a control code. The question is followed by a two-line hint (Fig. 4), a comment presented for a correct answer, and up to four, 19-character anticipated answers for the question.

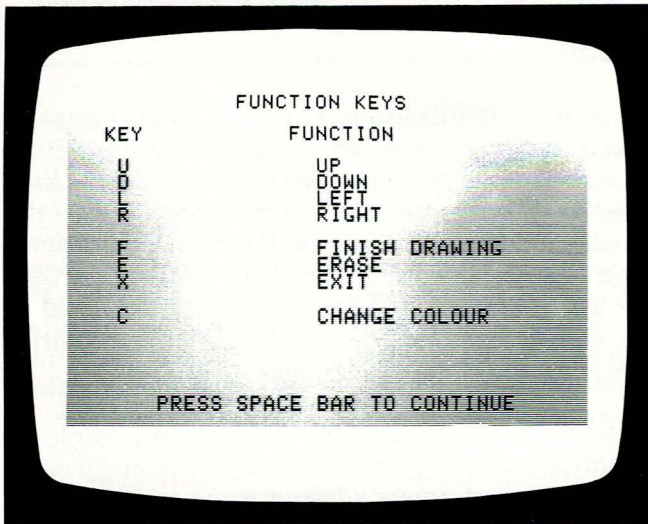


Figure 5

In terms of graphics, a graph development system and a graphics editor are provided. The graph development systems allow graphs to be designed using one of two preset functions (i.e., $Y = Ax^2 + Bx + C$ or $Y = A \sin(Bx + C)$). The author must set values for the coefficients A, B and C.

The graphics editor of Z.E.S. enables the construction of "shape tables" by using the keys U (up), D (down), L (left) and R (right) to create a new shape (Fig. 5). The editor, when used in conjunction with the shape module (Fig. 6), enables control over color, dynamic graphics display and upper-lowercase letter capabilities. The presentation can, for example, show a diagram of a street with a stop sign, or the shape of a car (Fig. 7) approaching a house (Fig. 8). When completed, all graphics must be transferred to the module diskette for use in a particular lesson.

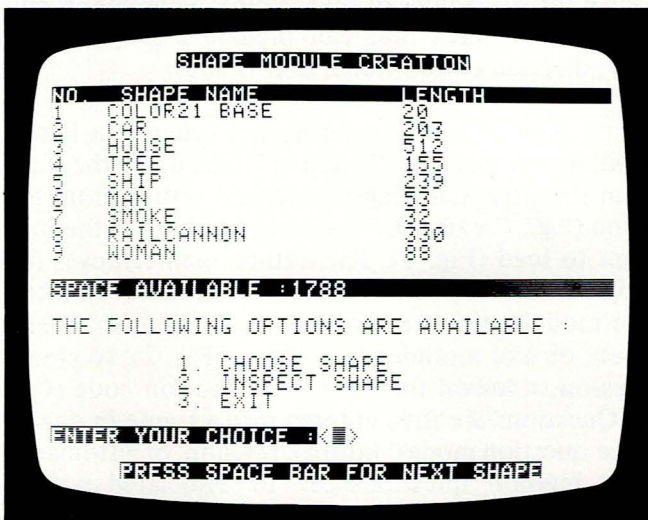


Figure 6

From a student's perspective, a valid student ID code must be entered before beginning a session. Once under way, the student is presented with each question frame, and the student chooses one of six options (i.e., answer the question; request a hint, a data page, a graph page, a diagram page; or terminate the lesson). Some choices are mandatory and must be requested, if provided by the author. The student proceeds on this basis through the lesson module, gaining information from the "page" supplied and, eventually, providing an answer to each question frame.

A formal, student management system is provided, and a student ID code must be authorized by the instructor prior to each student's use of a lesson. Multiple uses of the lesson require the instructor to authorize a student prior to lesson use. At the end of each lesson, the student is permitted to leave a two-line message for the instructor.

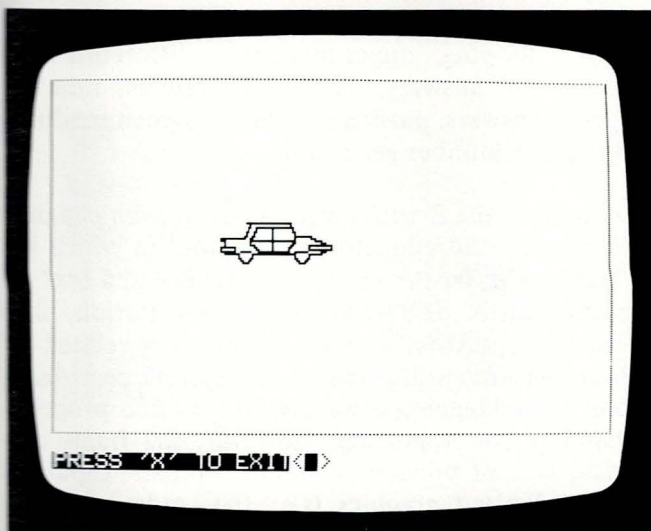


Figure 7

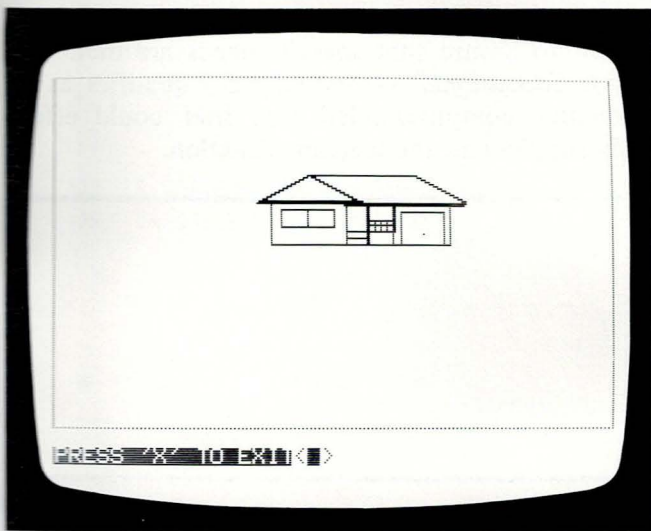


Figure 8

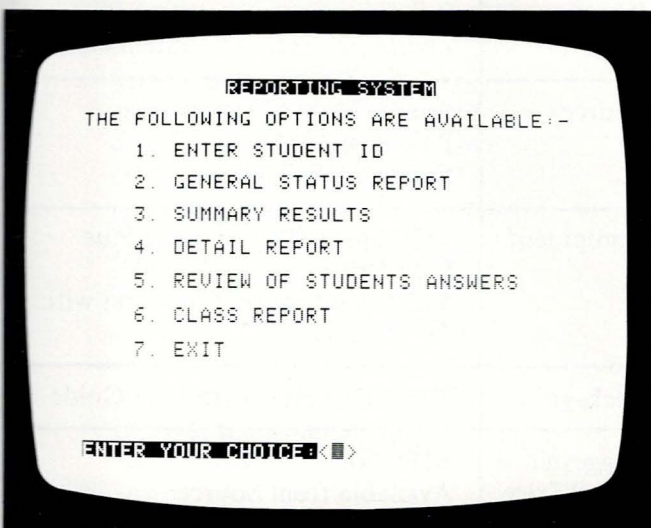


Figure 9

In terms of specific management-systems capabilities, six options and an exit facility are provided (Fig. 9). The first option allows certain selections of a student's activity to be reported. Four student-specific reports may be requested. These include: 1) general status report, containing the current module and question that the student is working on, and messages to/from the tutor; 2) summary results report, containing the number of questions attempted and/or correctly answered, as well as information about hint requests and response time; 3) detailed report, which provides an analysis of the student's performance on a question-by-question basis; and 4) student's answers report, containing the answers supplied by the student for each question.

The Class Report provides information about the class, and contains three additional options. For each option, the student numbers and names are listed along with the requested data, which can include: 1) percentage correct; 2) average response time; or 3) the question taking longest to answer.

Z.E.S. Evaluation

Overall, the evaluator's ratings of the system tended toward the mid-range of the scale. The ratings are discussed in terms of: 1) authoring; 2) delivery; 3) management system; and 4) technical capabilities.

The authoring component of the *Z.E.S.* system was rated highest for clarity and appearance of the documentation. However, during the course of the review, several operational difficulties were encountered while authoring. The "Amend" option, for example, did not work on the demonstration diskette, forcing the use of the "Delete" option and then recreating the entire lesson. Additionally, the system was built for a single disk-drive computer, which required frequent swapping of diskettes for the construction of diagrams. Incorrect swapping of diskettes resulted in a "System Error." A disturbing "End of Data" error was also encountered and, in some cases, exiting from an option required rebooting (or restarting) the system.

Although additional experience with the system may reduce the difficulty level, actual authoring was not found to be as straightforward as the documentation implied.

In terms of the user-information delivery, the orientation to student control rated highly, in that the student can request a hint or diagram for amplification before answering a question. In addition, the student can leave a message for the instructor or tutor, an option for student control. However, the system restriction prohibited a student from repeating a lesson without reinitializing, and thus requiring

intervention by instructors or tutors. This was viewed as a potential problem for drill and practice applications. In addition, the tutor secret code of "shift P" for advancing a student past an incorrectly answered revision question provided only a limited degree of security.

Potential difficulties were also seen in the limit of two lines for a hint, and the reliance upon exact matching for correct answers.

The management system received the highest rated scores in the evaluation. The recordkeeping capability included individual responses to questions, response times and class summaries. All were rated high as essential recordkeeping functions for an authoring system.

The *Z.E.S.* technical strategies, diagram generation and shape table construction provide needed graphics capabilities. This is evidenced by the relative ease with which graphs can be charted through use of the equations provided, and by the availability of some preset shapes. However, the absence of certain basic

capabilities was evident during the review. These included looping, upper-lowercase differentiation, negation of answers, order of answers, number ranges in answers, passing variables between modules and random number generation.

In summary, the *Zenith Educational System* capabilities provide the educator with a tool in which information can be presented in individualized text or graphical form. The strategy of presentation, followed by a question, appears most closely related to skill assessment. In this regard, the system provides a strong recordkeeping capability for *ad hoc* progress assessment and reporting. However, the frame approach, lack of previously mentioned basic capabilities and limited graphics (i.e., the student cannot interact with the graphics) would appear to limit the system's applicability in many instructional strategies. Individual authoring requirements should be carefully researched and evaluated, then compared to the capabilities of the *Zenith Educational System* to ensure that specific needs are met. The *Zenith Educational System (Z.E.S.)* qualifies as an interactive computer-aided tool that could effectively supplement the teaching function.

Cataloging Information

Dewey Decimal Number: 371.3
 ERIC Descriptor: Curriculum Development
 Subject Heading (Sears): Authoring Tools—Computer programs

Source Information

Program	<i>Z.E.S.</i> <i>Zenith Educational System</i>
Source	Avant-Garde Creations P.O. Box 30160 Eugene, OR 97403
Equipment	48K Apple II or Apple II Plus Disk Drive Video Monitor or Television with RF Modulator
Package	Three Diskettes with User Guide
Suggested Retail Price	\$295.00 Available from Source

Eureka Learning System Overview

In order to use the *Eureka Learning System (ELS)* to develop instructional lessons, basic terminology has to be mastered. These include: ENTITIES (concepts you wish to teach), ATTRIBUTES (information associated with those concepts), TYPES (for modification of ENTITIES and ATTRIBUTES), and SENTENCES (material presentation) (Fig. 1). These are briefly defined in the documentation and then presented in terms of example lessons in a "learn-by-doing" approach. The examples contain photocopies of actual computer output with hand-drawn underlining for author-entered information.

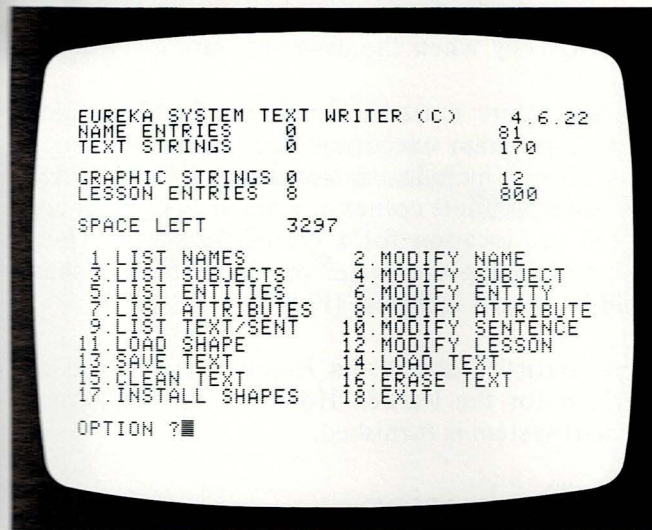


Figure 1

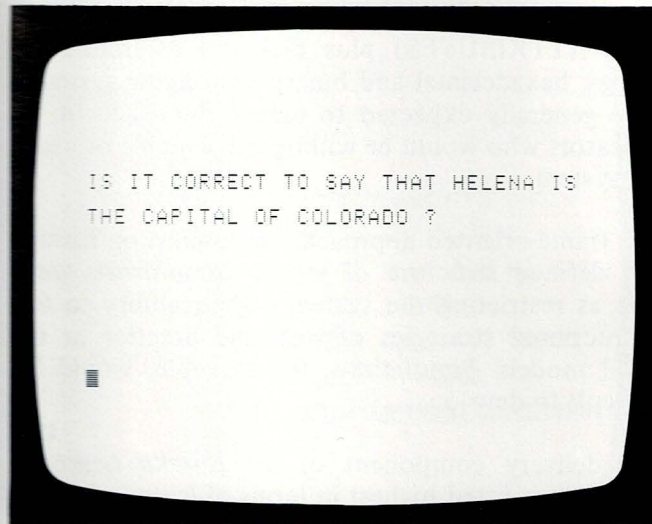


Figure 2

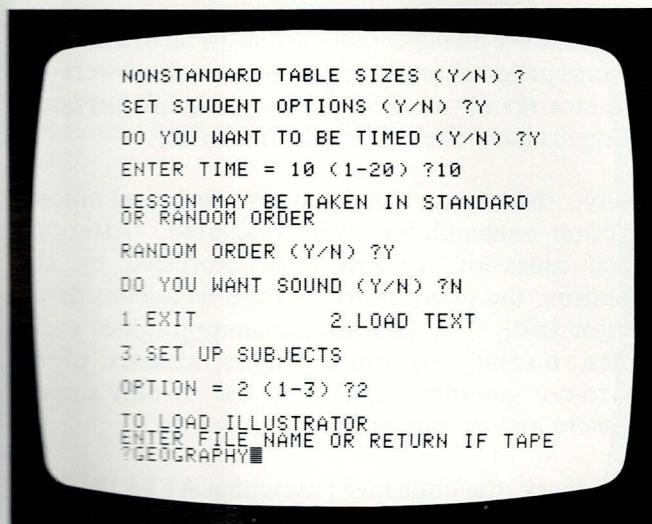


Figure 3

An example lesson using the subject area of geography deals with the ENTITY of "Arizona," which has an ENTITY type of "state," an ATTRIBUTE CODE of "AZL" (among others), an ATTRIBUTE TYPE of "the largest city in" and an ATTRIBUTE VALUE of "Phoenix." With this definition, the author can construct questions called SENTENCES such as "what is (5) the (2) of (1)?" This would be presented to the student as "What is the largest city in the state of Arizona?"

By constructing lists of attributes for an ENTITY and employing multiple SENTENCE (i.e., question) formats, numerous questions can be posed (Fig. 2).

To use the system, the student types RUN and is then asked if he/she wants his answers timed, in what order he wants questions presented, and if he wants sound (Fig. 3). The student can then ask for presentation of the material, practice or evaluation (Fig. 4). The different modes (Figs. 5 and 6) are effected by phrasing questions in a format such as "Is it correct to say _ _ _ _" or "The answer is _ _ _ _." The prompts for the student, such as "press RETURN" are not always presented, and control sequences (i.e., CONTROL E to end) are used only for some functions.

Some graphics capabilities are included in the system. When using "Character Graphics" (i.e., letters, numbers and special characters), each letter or number must be specified in hexadecimal numbers, which are provided in the documentation. The characters are then presented at author-specified X, Y coordinates through the use of graphic SENTENCES (Fig. 7). Similarly, graphic shapes can be entered in paired hexadecimal values.

The student can interact with the graphics display by typing the I, J, K, M keys (Fig. 8) to move the

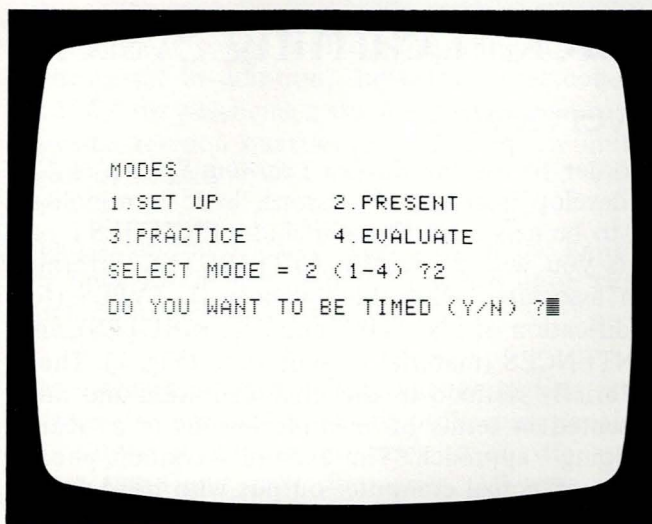


Figure 4

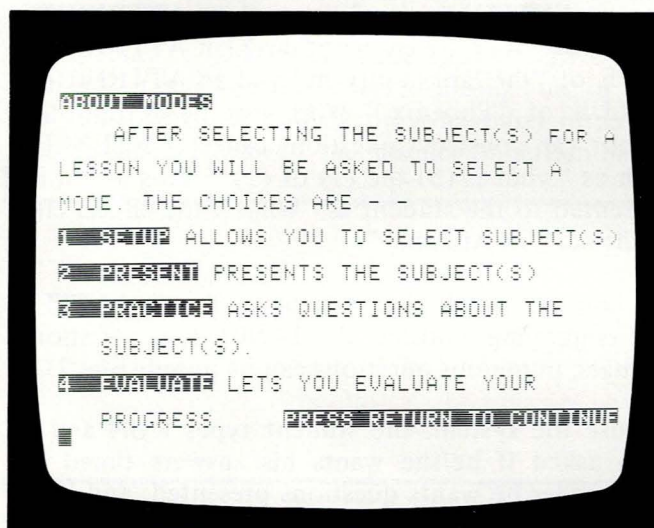


Figure 5

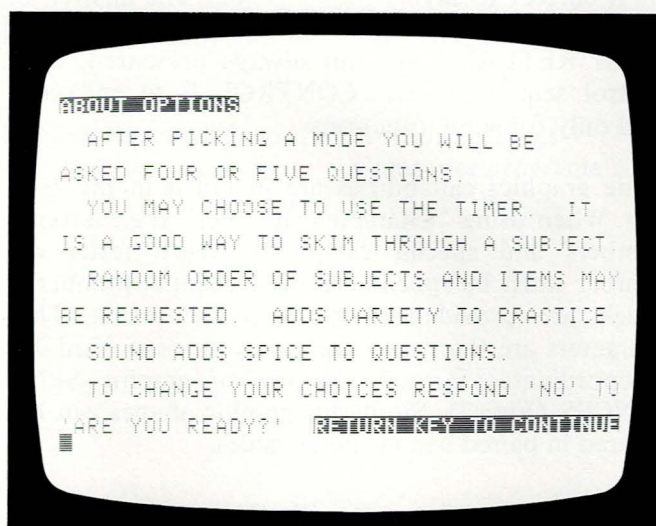


Figure 6

cursor up, left, right or down, and then pressing the RETURN key when the desired location is reached.

This procedure must be completed by the student prior to program execution since run-time instructions do not include this information. In addition, only the upper left corner of a graphics shape is considered the location for a correct response. Therefore, positioning the cursor on the center of a shape could be judged incorrect (Fig. 9).

Some statistics about the lesson performance are provided for the learner. However, no formal management system is furnished.

ELS Evaluation

Overall, the evaluator's ratings of the *Eureka Learning System* tended toward the low end of the scale. In terms of the backgrounds required for educators, the use of system-specific terminology and concepts (e.g., ATTRIBUTES) plus technical references to strings, hexadecimal and binary numbering systems, were generally expected to reduce the spectrum of educators who would be willing and capable of using the system.

The frame-oriented approach, fixed-question format and defined structure of lesson components were seen as restricting the system's applicability to the instructional strategies of drill and practice or tutorial models. Simulations, for example, would be difficult to develop.

The delivery component of the *Eureka Learning System* was rated highest in terms of learner control of the system. However, the dependence on previously established instructions (i.e., cursor control, end, RETURN, etc.) were identified as potential problems.

The lack of a management system to record student answers, progress, correct and incorrect answers for instructor review was identified as a significant issue, tending to limit the system's applicability.

Finally, the area of support for technical micro-computer capabilities was evaluated. Although several areas of concern were expressed by the evaluators, the graphics capabilities were considered a major issue. The lack of dynamic graphics capabilities, inability to rotate shapes, absence of an easy-to-use graphics editor, and use of only upper-case were major areas of concern.

In summary, the concept of matching ATTRIBUTES to ENTITIES was found to be a unique approach in authoring systems. This approach appears to have

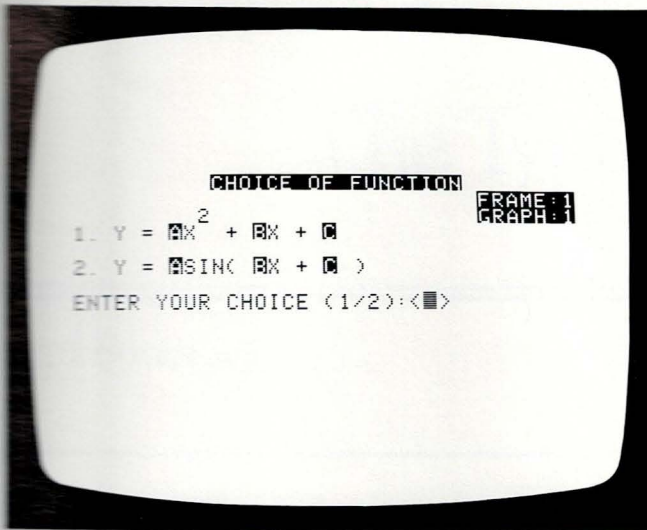


Figure 7

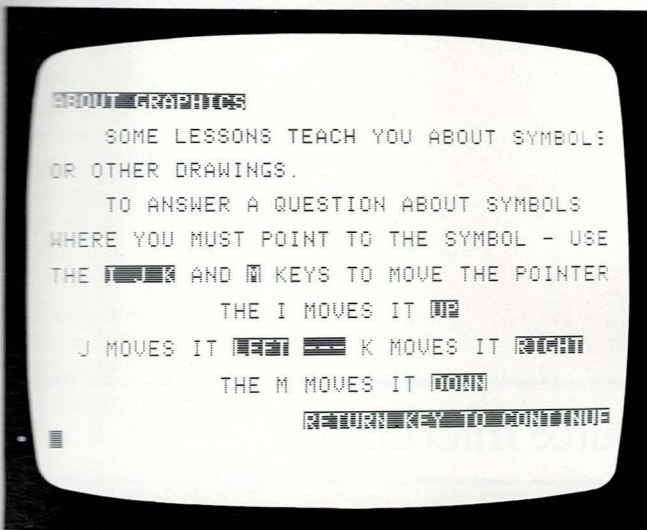


Figure 8

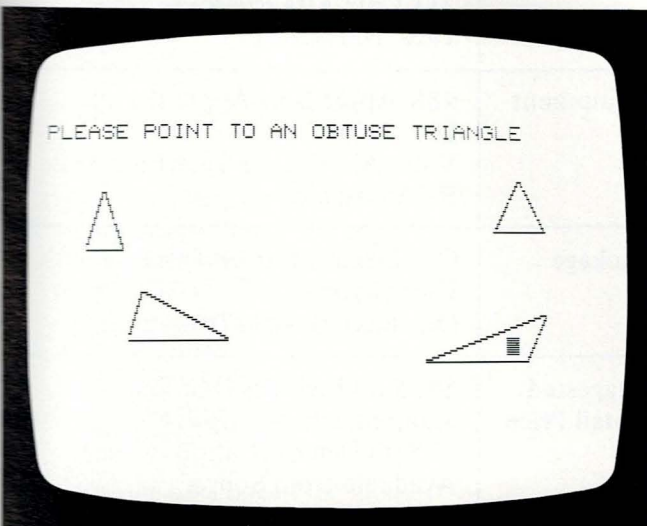


Figure 9

more merit for certain types of instructional strategies than others and could have great value in certain subject areas. Again, we would urge that specific authoring requirements be carefully evaluated and compared to the capabilities of the *Eureka Learning System* to ensure that individual school and instructor needs are met.

Authoring programs hold considerable promise for expanding the dimensions of electronic-assisted instruction. When selecting ANY courseware, care and study should be exercised in the selection of computer-based programs to ensure the enhancement of curricular and instructional goals.

The authors would like to express their appreciation for the assistance and expertise of Brian Nakagawa, Jerry Bowman, John Howard, Julie Kavades and Wayne Krone, all of California State University at Fresno.

Jack A. Chambers, Ph.D., is Director of the Center for Information Processing (CIP) at California State University, Fresno, and Director of the Coordinated Center for Computer-Assisted Instruction in the Cal-State University system.

Jerry W. Sprecher is Associate Director of CIP and Co-Director of the Coordination Center. Chambers and Sprecher are responsible for CAI standards, evaluation and courseware recommendations in the state-wide university system.

Cataloging Information

Dewey Decimal Number: 371.3
 ERIC Descriptor: Curriculum Development
 Subject Heading (Sears): Authoring Tools—Computer programs

Source Information

Program:	<i>Eureka Learning System</i>
Source	Eiconics, Inc. 211 Cruz Alta Road Taos, NM 87571
Equipment	48K Apple II or Apple II Plus Disk Drive Video Monitor or Television with RF Modulator
Package	One Demonstration Diskette with User Guide One Basic System Diskette
Suggested Retail Price	\$495.00 Includes One Year Maintenance and Updates. \$25.00 Demonstration Package Available from Source

MasterType

The Typing Instruction Game

Lightning Software

Fred Bockman

A program that utilizes the unique features of the microcomputer.

The frequent complaint of the dearth of high-quality educational software is becoming a cliché. It's true that there have been so few examples of top quality software that, upon the discovery of an excellent package, one is tempted to stand on the tallest building and shout to the masses, "Here is something worth your money!" Until recently there have been few reasons to shout from the rooftops, but this typing package will send people cheering. It embodies all the qualities of educationally sound software.

Fred Bockman is Director of a Microcomputer Title IV-C Project, dedicated to the Identification and Development of Educational Applications Software (MICRO-IDEAS). Bockman is responsible for direction and implementation of microcomputer workshops and software review for a state-wide consortium composed of 84 Illinois school districts.

MasterType is a drill and practice program used to teach touch typing. This program utilizes a game format to hold student interest, has a number of preprogrammed lessons, and yet, allows a teacher to create lessons.

The program uses an arcade "shoot-'em-up" format. The student is the planetary defender fighting off the deadly invaders from the planet "Lexicon." The invaders are depicted as letters or words in each corner of the screen. The words launch missiles toward the defender's planet in the center of the screen (Fig. 1). The defender must destroy the missiles by typing them on the computer's keyboard. If the defender can destroy a missile near the invading word, the word itself is destroyed, to be quickly replaced by another word. The lesson ends when all 40 invading words have been destroyed.

The missile speed can be controlled by selections at the beginning of the lesson. At faster speeds, the student cannot look at the keyboard, for while he is looking at the keyboard, another word will attack from another direction. In addition, it is necessary to type the word quickly. If it is typed slowly, the missile will have traveled far enough from the attacking word to prevent the destruction of the word itself. It is necessary to destroy the words to complete the lesson (Fig. 2).

This software package includes a manual that contains directions for running the program, descriptions of all options and their educational implications, hints for successful winning of the game, finger charts for the Apple keyboard, suggestions for successful touch typing, extensive directions for making customized lessons, and descriptions of each of the 17 preprogrammed lessons. This manual, while physically small, contains all the information needed to make use of the package in a classroom setting.

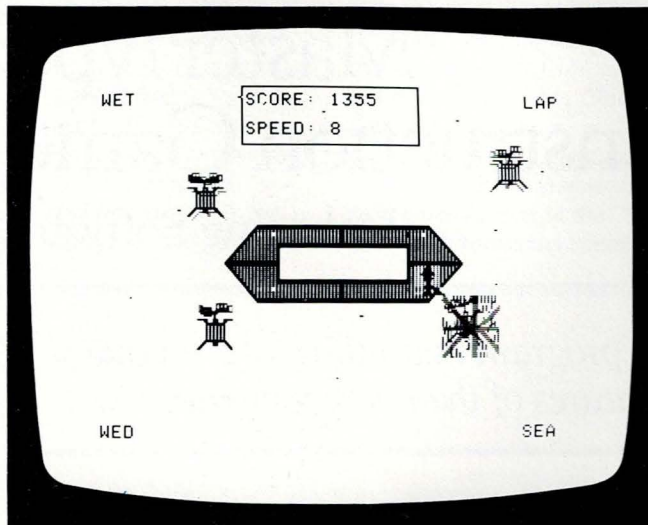


Figure 1

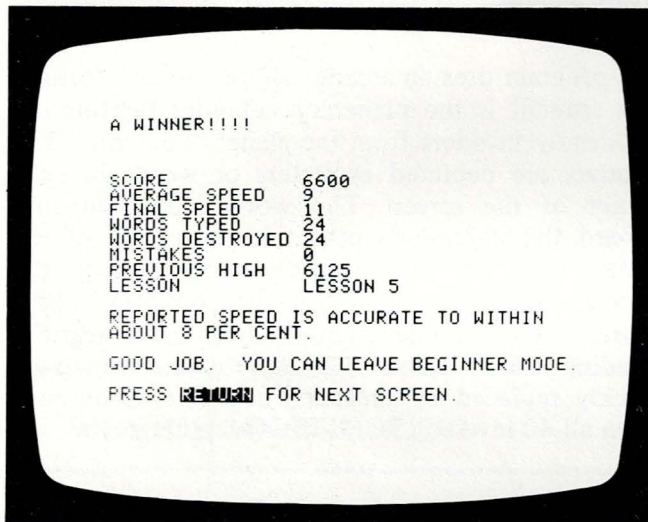


Figure 2

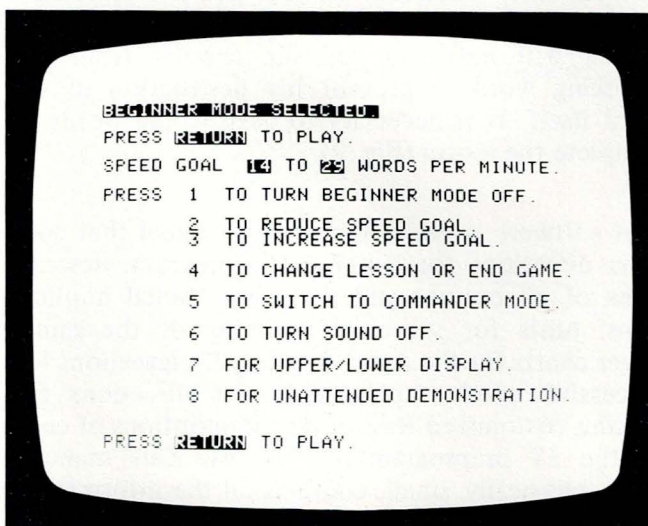


Figure 3

Many software authors forget all they knew of curricular design when they sit down at the keyboard (assuming they know curricular design in the first place). There are certain givens that go into a solid educational package that need to be addressed to create a worthwhile package. These concepts can be broken down into three general areas—educational pedagogy, machine utilization, and appropriateness of delivery.

Educational Pedagogy

Educational pedagogy can be examined in several different ways. Fundamentally, does the package teach something, does it present material in a reasonable sequence, and is it presented in an understandable format? *MasterType* meets these criteria.

Many packages that claim to teach actually present information without regard to whether the student is learning what is being taught. *MasterType* not only presents activities in touch typing, but the very success of the game depends upon the successful learning of the concepts taught. There are two parameters of the program that are under constant student control—the speed at which words are presented and the display of the entered word. The student can select the speed of the lesson at the beginning of each lesson (Fig. 3). The speed corresponds roughly to the number of words per minute the student expects to type. The second parameter, called mode, determines whether the answer as the student types it will be displayed on the screen. In cadet mode, the word is displayed as it is typed (Fig. 4). This allows the student to check his work as he types. In commander mode, the word is not displayed. This causes the student to concentrate on the target words instead of the keyboard or the entry word display. These parameters show an excellent understanding of the way students learn to type, and gradually lead the student away from the crutches that he used as he started.

In addition to the sequences of parameter control mentioned above, the package has 17 preprogrammed lessons. These lessons present the keyboard in a logical and proper order. The lessons follow a pattern of presenting the keys of one row (home, third, bottom, top), then presenting short words using the keys of that row, then longer words using that row, and finally words using letters from all rows already presented. This pattern is repeated for each of the four rows, and finishes with several lessons of hard words.

In addition to the 17 lessons, there is a built-in mini-authoring system that allows a teacher to create lessons (Figs. 5 and 6). This authoring system

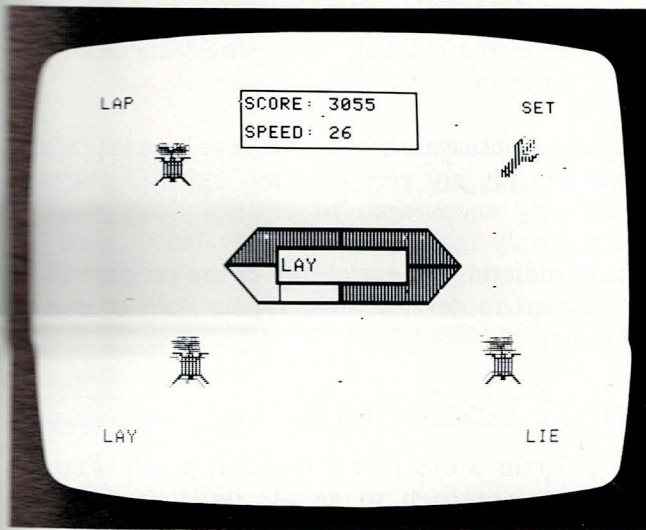


Figure 4

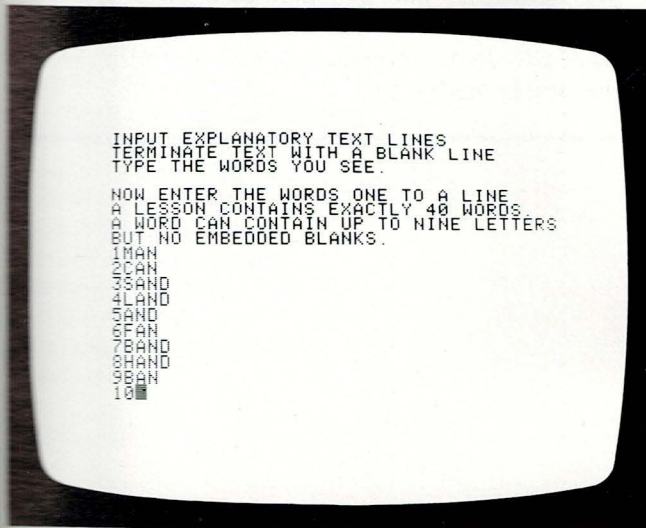


Figure 5

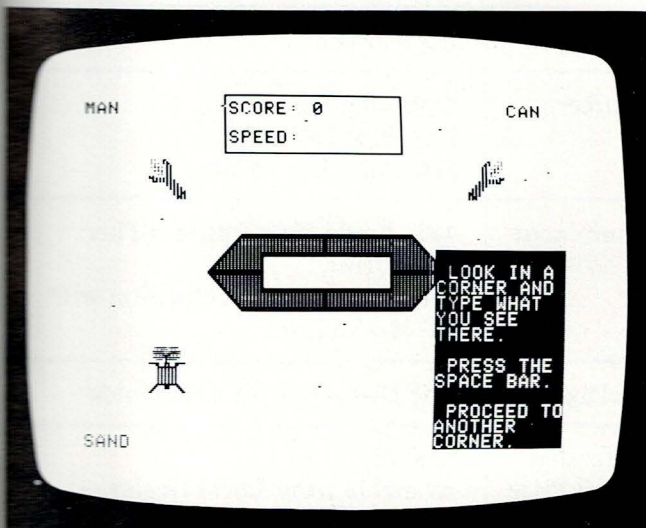


Figure 6

creates lessons consisting of 40 words, each up to nine letters long, with specific directions to the student, up to ten lines of 255 characters each. This mode allows a typing teacher to enter the lessons from whatever text they are using. The lessons are then saved permanently on either the original diskette or on separate diskettes. The use of separate diskettes allows the teacher to create a set of lessons for each chapter of a text to be completed by the student as needed.

The format in which the lessons are presented is certainly appropriate for creating high student interest. The use of an arcade-game format makes the lessons not only fun, but downright addicting. The sound effects, which are removable for those computers without headphones, are reminiscent of the best of the arcade games. The graphics are professionally done, with the visual result being the equal of any of the finest of the current crop of arcade-type entertainment games. The lessons are scored in two ways, by the number of words per minute of typing speed and by a numeric score based on how many words were misspelled and how quickly the words were destroyed. This scoring system should create healthy and worthwhile competition.

Machine Utilization

Many software packages today look as though they were written to run on a teletype terminal—they make no use of the special features of the microcomputer. With the availability of color, graphics, animation, sound, and interactive keyboards on microcomputers, a program should make appropriate use of these factors for improving the motivational aspects. Obviously, it is poor design to use every “whistle and bell” constantly, but wise use of these features can greatly improve a program.

MasterType uses these features to great advantage. The screen layout is well designed, with the important areas of the display remaining at the same location throughout the program. The live keyboard allows the program to evaluate every keypress, and either the RETURN key or spacebar can be used to signify the end of a word. This is a program that could never have been run on a timeshare system. It requires the features of a microcomputer.

The main shortcoming of the program is really a shortcoming of the Apple computer. The program uses the shift key only with punctuation and special symbols. This is because the Apple will not recognize the shift key with alphabetic characters. There are several hardware modifications that can be made to the computer to allow the use of the shift key, but none of these is standardized or approved by

Apple, so the program assumes the machine is unmodified and uses only the standard Apple keys.

Appropriateness of Delivery

Educational games have traditionally been used to present concepts and reinforce learning. Interest centers usually use a game format; vocational, economic, and governmental classes use simulations; and mathematics games have been used for years. It is an interesting comment on the motivational aspects of computer usage that there is some debate concerning the use of game-type educational software. Some students do utilize certain game-type educational software to the detriment of their other studies. It is, therefore, the responsibility of the software designer to make sure that the software package is under teacher control. *MasterType* meets this requirement by presenting the drill in specific lessons. Each lesson should take five to ten minutes for a student to complete. At the end of the lesson, the student's scores are presented on the screen. The student should then record these scores on the

progress sheet (to be supplied by the teacher—there are none in the manual). This provides a clean end to the program.

Drill and practice are often drudgery. Typing practice is traditionally not very exciting, and students must be strongly encouraged to practice. This program will not only make practice enjoyable; it makes it almost addicting. The structure of the program leads the student to develop good typing skills to master the game.

Final Conclusions

This program is one of the few that presents an entirely new approach to an old topic. It is such a good approach that one wonders why it was not thought of earlier. There will be more packages of this groundbreaking nature that will not be repeats of old concepts, but will take new approaches to education. If the new packages meet the standards set by *MasterType*, the educational computing world will be better off by far.

Cataloging Information

Dewey Decimal Number: 652.3
 ERIC Descriptor: Typewriting
 Subject Heading (Sears): Typewriting—
 Computer programs

Source Information

Program	<i>MasterType</i>
Source	Lightning Software P.O. Box 11725 Palo Alto, CA 94306
Equipment	48K Apple II or Apple II Plus Disk Drive, Video Monitor or Television with RF Modulator
Package	One Diskette with User Guide
Suggested Retail Price	\$39.95 Available from Local Dealer

Typing Tutor

Microsoft

Alex May

Older students were intrigued by the hardware; tenth graders, fascinated by the program.

Typing Tutor offers a self-instructional typing course, reminiscent of the recorded lessons once given away with portable typewriters. However, due to the advantages of the micro-processor technology, the student using Typing Tutor will probably learn recommended typing habits much more efficiently than will the student using the records. Practical but personal software such as this program serves to teach specific skills while also introducing the user to the computer.

Documentation

While *Typing Tutor's* documentation does not identify specific objectives in measurable, educational terms, the manual, prepared by Dottie Hall, does discuss the obvious goals of improving typing accuracy and increasing typing speed. An experienced teacher could easily generate objectives for a class or for individual students.

According to the information on the package, *Typing Tutor* is intended for the programmer, the student, the youngster, or the "sometimes" typist. The authors, Dick Ainsworth and Al Baker, don't expect a teacher with a classroom full of micro-computers to use it for instruction. Because of the intended audience, the manual could be improved with a few more charts or illustrations and more explicit instructions.

While the small manual has a clear and informal style, it occasionally presumes a knowledge the user might not possess. For example, the manual has only one diagram of the typewriter keyboard. This diagram carries the title "Diagram of fingers on home keys," but nothing in the drawing identifies the home keys. The accompanying text helps to explain which eight keys serve as home keys and gives a hint about typing some of the other keys. However, the instructions do not tell the student about the "T", "G", and "B", which the left forefinger types or the "Y", "H", and "N", which belong to the right forefinger.

If a few more diagrams were included in the manual or generated by the program onto the screen, beginning typists would benefit. Of course, if *Typing Tutor* is used as supplementary practice along with instruction from a teacher, these shortcomings in the manual do not have much significance.

The information found in the manual does explain how to use the program. Differences between the

computer keyboard and a traditional typewriter keyboard are discussed, along with some precautions to be followed when using the computer for typing practice.

Application

Once the user has read the instructions and RUN *Typing Tutor* on the microcomputer, it will become apparent that this program can truly provide individual attention to the student's needs and abilities. Regardless of the typist's prior skills, *Typing Tutor* is ready. A beginning typist who first uses *Typing Tutor* will spend most of the opening practice sessions working on the "home keys." As skills develop, *Typing Tutor* responds to the improvement and adds other letters to the student's lessons.

A typist with some skills already developed will discover that *Typing Tutor* can quickly diagnose the level of proficiency and adjust to the proper place in the sequence of instruction. Even a typist who needs only to brush up a little will find that this program can help.

Field Test

According to several typing teachers consulted for this review, including a high-school business department chairman, *Typing Tutor's* sequence of instruction follows sound, traditional practices. Home keys, the space bar and the RETURN key provide the basis for the first lessons.

As part of the review process, several students were introduced to *Typing Tutor* and were observed while they were practicing. A sixth grader and a seventh grader with no previous typing experience seemed to grasp, within a few minutes, the information necessary to operate the program. They had no trouble following the directions and proceeding through the lessons.

Typing Tutor provides an initial choice for practice of LETTERS, NUMBERS, or SYMBOLS (Fig. 1). The second choice offers individual keys or a practice paragraph. The paragraph is only a simulation because the program constructs it from the letters mastered in the lessons with individual keys.

Even though the paragraph contains no sentences, the sense of typing a paragraph exists, with spacing between words and a "RETURN" at the end of each line. One typing teacher asserted that these simulated paragraphs are quite appropriate, as many experienced typists complete several pages without error and yet have no idea of the content of what they have typed.

One very noticeable benefit derived from the microcomputer's configuration was that the two typists watched the cursor and the letters on the screen instead of their hands and the keyboard. Students have a tendency to watch their fingers, which interferes with both their typing speed and their ease in memorizing the keys for each finger. The TV screen or monitor may prove a real diversionary plus for *Typing Tutor*.

Older students were intrigued by the hardware, especially the monitor screen. The constant keyboard monitoring performed by the program, coupled with the end of lesson reports, fascinated the tenth graders.

Word Count

One caution, however, is in order here, and the manual does mention it. The Words-Per-Minute (WPM) scores, reported continually through the lesson, do not compensate for errors. This results in inflated scores, which may lead students to unrealistic conclusions as to their speed.

Actually, the WPM scores seem inappropriate during the practice sessions that emphasize mastery of specific letters. These lessons make no attempt to simulate a realistic typing situation, yet they report WPM scores. A continuing controversy among typing teachers centers on the choice of priorities for students. Should they learn speed first without regard for accuracy until they develop keyboard dexterity? Or should they concentrate on accuracy even if speed is sacrificed?

The need to develop speed and accuracy simultaneously is most important, according to the business department chairman who assisted in this review. This seems to be part of *Typing Tutor's* overall goal as well. The program reports an error count even though it does not use errors in its WPM calculations. There is concern that students will lose sight of accuracy in pursuit of speed; students could overreact to WPM numbers.

The students observed for this review leaned towards increased WPM scores. Their relatively high (inflated?) WPM scores cheered them and encouraged them to go even faster. It was felt that the WPM count proved useful with the practice paragraph, but that it only added complexity to the letter practice sessions.

Running the Program

As the above discussion indicates, using *Typing Tutor* is easy and uncomplicated. The instruction

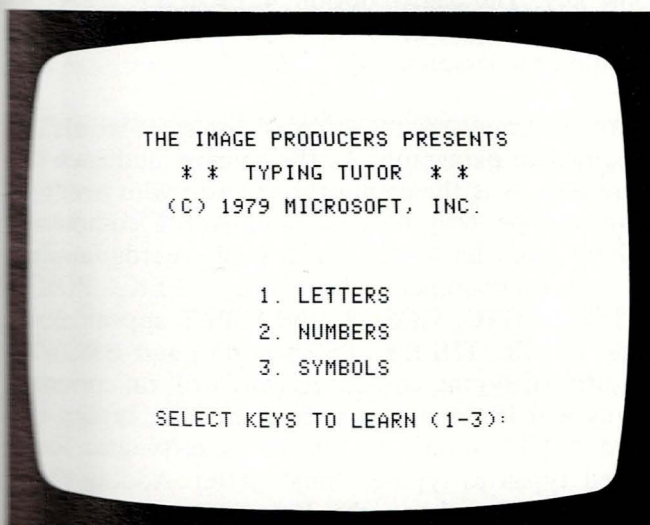


Figure 1

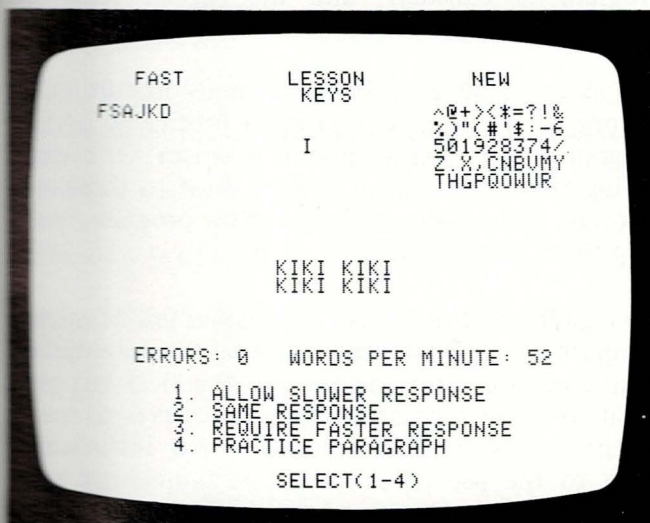


Figure 2

manual and the screens provide enough direction for someone to get started. Once the program is up and running, the user has a choice of (1) LETTERS, (2) NUMBERS, or (3) SYMBOLS. The next screen offers (1) TYPING TUTOR or (2) PRACTICE PARAGRAPH. Choice (1) from this screen gives a series of ten exercises that concentrate on specific keys, beginning with the "Home Keys" and progressing to practice with the rest of the keyboard as the program recognizes that the user has gained proficiency on the current lesson keys (Fig. 2).

Once the user has completed ten exercises in (1) *Typing Tutor*, the second screen again displays a choice of continuing practice on a few keys or moving to (2) PRACTICE PARAGRAPH.

The paragraph lesson provides practice with words arranged in a paragraph format. The practice paragraph appears on the screen typed double spaced (Fig. 3). An underscore beneath the first letter indicates where the user should start, and the user-typed letters appear directly under the words in the lesson. As the user types, the underscore moves to the next space. Whenever a wrong key is struck, the program issues a low-level buzz to warn the user to exercise more care.

As mentioned earlier, this feature of interlinear typing seems to keep the student's attention on the screen and off the keyboard. Successful typing requires attention to the original material, not to the keyboard or one's fingers. The novelty of *Typing Tutor* and the television screen may help beginning typists to develop this important typing habit.

Special Features

Another important feature of the program is the Paragraph Analysis screen (Fig. 4), which indicates the results of the practice session. Information from five data banks is calculated and presented to the user: the number of keystrokes missed, the keys missed, the keyed responses slower than considered acceptable, a percentage score for accuracy, and rate in words per minute. From this information, an individual user or a teacher could plan further work that should lead to improvement.

Once the student has completed several lessons and practice paragraphs, the analysis screen acquires more meaning for instruction with another special feature. Several options available to the user allow modification of the program's responses. If the student wants to work on more letters, selection of 1. ALLOW SLOWER RESPONSE will tell the computer to lower its requirements for determining student proficiency so that new letters can enter the

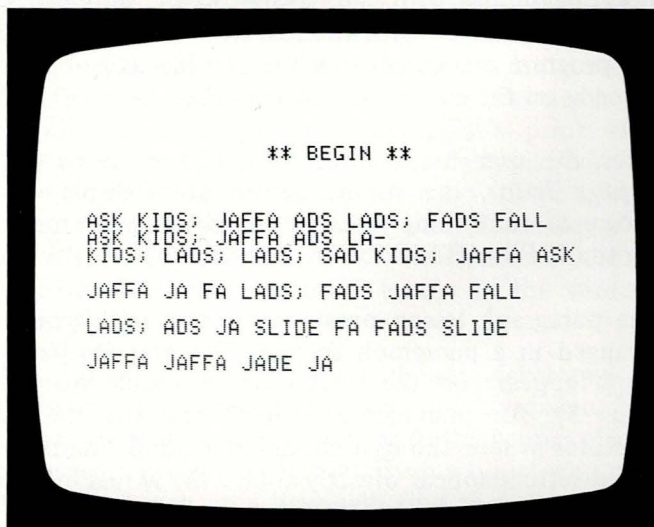


Figure 3

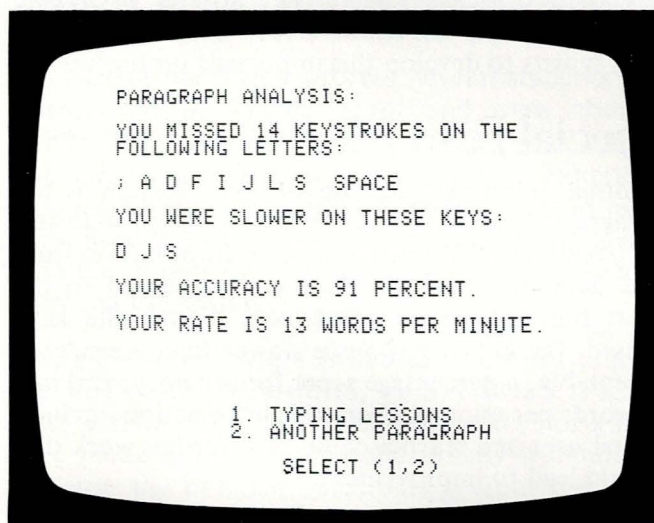


Figure 4

lesson. Choosing 2. SAME RESPONSE leaves the program in its normal operating mode, which the program authors' recommend for most lessons. If the student needs additional practice on specific keys, 3. REQUIRE FASTER RESPONSE instructs the computer to increase its requirements. In effect, this option gives additional practice on each troublesome key. The fourth option, 4. PRACTICE PARAGRAPH, instructs the computer to build a paragraph based on the lesson keys.

A note of interest is the authors' choice of words for the practice paragraph. As the primary audience for these lessons is the computer operator who needs to learn to type, they have included BASIC commands in their word bank along with many words familiar to non-programmers. So PDL, PEEK, POKE, HLOT, GOTO, GOSUB, and INPUT appear along with THEIR, THERE, THESE, MY, and BROWN. Traditional typing courses provide drill on common words and letter patterns. For example, typing the word "THE" should be almost as automatic for a skilled typist as typing a single letter. Adding these BASIC commands to the lessons certainly should benefit a student who intends to type programs.

Conclusions

Overall, *Typing Tutor* seems to fulfill all of its claims and this user's expectations. It adds interest to a repetitious drill and encourages the student to keep trying. The interaction between the user and program is constant, and the regular feedback should keep students motivated.

The skills to be learned are certainly important to anyone who wants to become a proficient BASIC programmer or microcomputer operator. Of course, typing skills have a great deal of value to those not involved with computers at all. This program could help them, too.

Typing Tutor provides an innovative use of microcomputer technology demonstrating the potential of microcomputers for instruction. Practical but personal software such as this program serves to teach specific skills to students while also introducing them to the use of a computer. Individuals who want to learn to type or schools that want to add another dimension to a typing class should consider acquiring this program.

Alex May is Basic Skills Coordinator for New Jersey's Millville Public Schools and is responsible for the implementation and evaluation of all remedial math and reading programs within the district. As the district's Computer Consultant, he supervises all data-processing activities.

May holds bachelor's and master's degrees from Glassboro State College, New Jersey, a linguist's certificate from the Defense Language Institute, Monterey, California, and is currently a Ph.D candidate at the University of Pennsylvania.

Cataloging Information

Dewey Decimal Number: 652.3
 ERIC Descriptor: Typewriting
 Subject Heading (Sears): Typewriting—Computer programs

Source Information

Program	<i>Typing Tutor</i>
Source	Microsoft Consumer Products 400 108th Ave. N.E., Suite 200 Bellevue, WA 98004
Equipment	32K Apple II or Apple II Plus Disk Drive Video Monitor or Television with RF Modulator
Package	One Diskette and User Guide (#2202) Also on TRS-80 Cassette.
Suggested Retail Price	\$19.94 Diskette \$14.95 Cassette Available from Local Dealer

Meteor Multiplication Arcademic Skill Builders

Developmental Learning Materials

Marguerite M. Mason
W. Bruce Mason

A galactic shower of motivational math.

"My two favorite programs that I get to run on the computer are Space Invaders and Meteor Multiplication. I wish my teacher would let me run them more often," remarked a fifth-grade learning disabled student. And a high school special-education student said, "On Fridays, I get to use my coupons to play Meteor Multiplication."

After hearing similar reactions from students of all ages and with significantly different handicaps, the reviewer was curious as to whether this Developmental Learning Tool (DLM) program was just a fast-paced arcade game or an effective learning tool.

What is it about a multiplication program that could entrance a fifth grader to request more time for arithmetic and cause a high school student to invest coupons, earned in a week-long behavior modification program, on a drill-and-practice program when other "pure games" were available?

Meteor Multiplication is one of six packages in the Arcademic Skill Builders in Math Series[®] (TM) published by Developmental Learning Materials. Each math package includes a teacher's manual, a diskette with an instruction sheet, black-line masters and a deck of 52 flash cards. The programs themselves are in arcade format. In *Meteor Multiplication*, eight large colorful meteors, with a single-digit multiplication problem inscribed on each, move toward a star station in the center of the screen, accompanied by appropriate "sci-fi" sounds. Answers appear in the center of the star station. The student aims a laser-type mechanism at the appropriate problem and fires projectiles in an attempt to destroy the meteor (Fig. 1). If any meteor can reach the star station before being disintegrated, the star station explodes in a burst of color and sound. The star station is immediately replaced, and the exercise continues with hits and misses being recorded at the bottom of the screen for the length of time specified by the teacher or until the star station is destroyed for the third time.

The teacher has the capability of controlling the exact stimuli presentation through the use of GAME CONTROL, which is a relatively simple management system (Fig. 2). There are five options presented to the teacher. Skill Level refers to the speed at which the meteors move. Problem Range allows the selection of facts containing multipliers within the ranges of 0-3, 0-6, or 0-9. Run Time sets the length of the session to whole minutes within the range of one to five minutes. Paddle Control allows the teacher to select whether (1) the student has the answer appear on the star station with the student using the Game



Figure 1

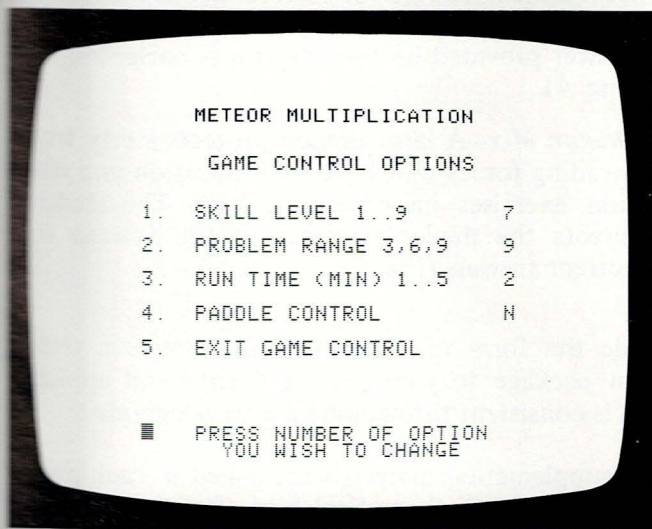


Figure 2

Paddle to aim and fire, or (2) requires the student to use various keyboard options to type in a correct answer, aim and fire the projectiles. One of the most frequent teacher complaints is the inability to turn off the sound; a sixth option to control the use of sound would be greatly appreciated.

One of the most attractive features of Arcademics[®] (TM) is the intent to base the learning activity upon a specified learning theory and then provide the teacher with the information necessary to implement that learning theory within the classroom. Ogden Lindsley's *Precision Teaching*, an offshoot of B. F. Skinner's work, was chosen as a basis for the teaching approach. *Precision Teaching* emphasizes the reduction of target activities to specific, observable tasks; the recording of the frequency of occurrences; and the use of rewards to increase or decrease frequencies as appropriate.

Arcademics[®] (TM) have been designed to provide drill and practice on individual arithmetic operations. Each response by the student is discrete and observable. The management system allows the teacher to specify the exact length of time that the stimuli will be presented, and record the number of correct and incorrect answers. The manual explains how to record these data on a Standard Daily Behavior Chart and to use it for diagnostics. The teacher's manual provides a complete explanation of how to interpret the data, and suggests techniques to improve results. Although this system is available to implement the *Precision Teaching* approach, the teacher has the option to use the diskette as a relatively simple drill and practice module.

Initially my concern centered upon the complexity involved in following the sequence of motions needed to answer each question. However, by experimenting with the variety of options available under Paddle Control and Skill Level, appropriate exercises were found to work effectively with all levels of students, including primary-grade mentally retarded students (on activities such as *Alien Addition*, one of the six DLM packages). Teachers have expressed some concern with options available under Problem Range, since the current system allows the teacher to specify problem components only within preset ranges. While the number sets are logically constructed, they do not follow the pattern taught by most teachers. Teachers tend to prefer to teach the multiplication tables of two, and so forth. While the current approach is reasonable, a minor programming modification to allow the teacher to specify the exact number pool from which to draw would greatly enhance the power of the programs.

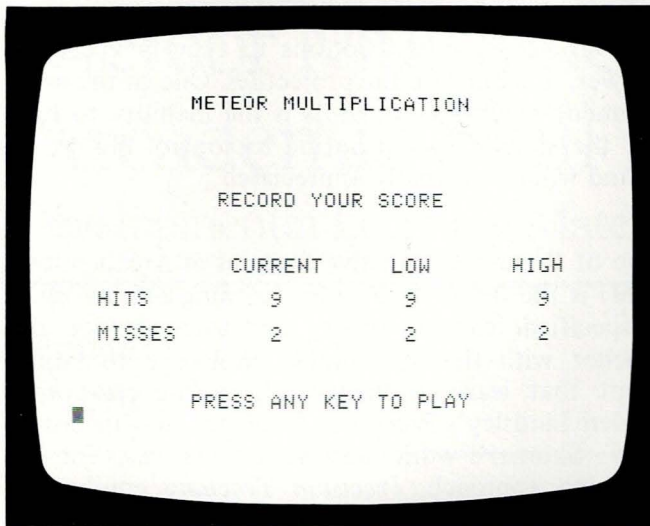


Figure 3

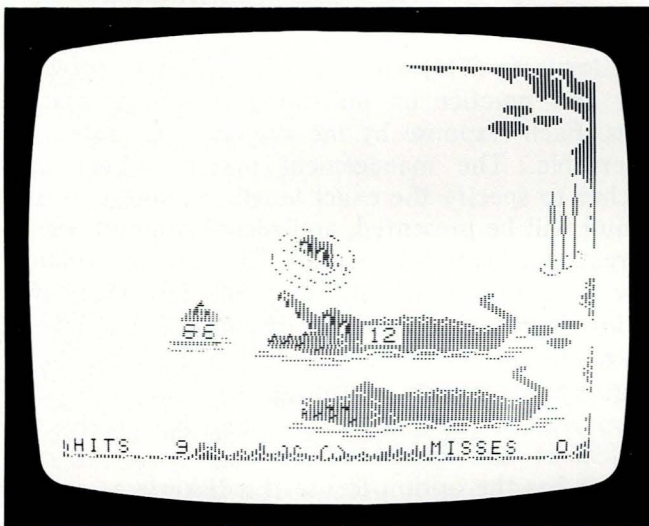


Figure 4

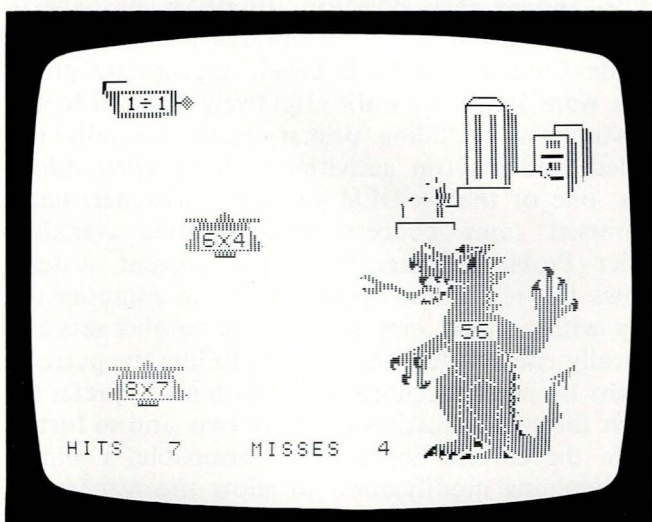


Figure 5

The Run Time option seems to be reasonably precise (about + 3%). However, there is no way to determine the length of a session when the star station is destroyed for the third time, thus terminating the game (Fig. 3). The modification of the program to display exact run time, along with hits and misses, would allow the teacher using *Precision Teaching* to obtain the data necessary for charting.

Meteor Multiplication is one of six program packages in the Arcademic Skill Builders in Math Series® (TM). Three of the six packages present single-type exercises similar to the multiplication package reviewed.

Alien Addition—Invading spaceships, with problems inscribed on them, move down from the top of the screen toward a movable laser that fires with the correct answer.

Minus Mission—Large blobs of a green substance, inscribed with problems, fall toward the bottom of the screen, where a robot fires laser beams for correct answers.

Demolition Division—Tanks, inscribed with problems, move across the screen toward cannons that fire accurately with the right answers.

Two of the packages allow the student to drill on combinations of operations.

Alligator Mix—As apples with addition and subtraction exercises float toward the mouth of an alligator, the student must decide whether the answer provided on the alligator is correct or not (Fig. 4).

Dragon Mix—A large dragon protects a city from invading forces that have multiplication and division exercises inscribed on them. The student directs the dragon's tongue of fire toward the correct answers (Fig. 5).

While the form of challenge and animation varies from package to package, the theory and application is consistent throughout the six programs.

The supplemental materials contained in each package provide both theoretical background and practical aids. The teacher's manual provides an overview and methods of implementing the *Precision Teaching* aspects of the diskette. Unfortunately, the manual does not include instructions on how to run the diskette. The diskette instructions pamphlet provides instructions for all six diskettes; however, the pamphlet's small size has often led to its being misplaced. Initially, the instructions are difficult for students to use; but, once started, the students seem

to have no trouble remembering what to do. Black-line masters provide both exercise sheets and data recording sheets. An additional master with simplified instructions for the student would be welcome. A set of flash cards covering the same concepts (with graphics similar to those on the diskette) help to ease the shortage of computer time and aid the transfer of skills.

DLM states that the program will work with students of all ages who require practice in the basic arithmetic facts. According to the teacher's manual, after extensive drill with this package, the student will be able to:

- correctly write or say answers to the basic-facts problems in the package at 40-90 responses per minute with 100% accuracy without using finger counting, visual aids, or manipulatives.
- correctly compute the basic facts included in the program within individual problems involving more complex processes.
- correctly answer basic facts included in the program in applied situations such as making change, determining distances, etc.

However, DLM's stated purpose #3, to transfer learning to real-life situations, is not directly addressed by the program.

Several strong points do stand out in the series. Foremost is the fact that the series is both complete and

based upon a proven approach to learning. Especially noteworthy is the program's attractiveness for the student based upon the current interest in arcade games. The built-in flexibility to provide teacher control over such variables as content, time, and speed make it a powerful teaching tool, particularly when the teacher follows the suggestions regarding changes in speed and fact range indicated by the patterns that are observed on the Standard Daily Behavior Chart. As illustrated in the teacher's manual, this complete package provides the teacher with an easy method for setting goals for individual students, monitoring their progress, interpreting the data collected, and making modifications to maximize learning.

Our doubts regarding the ability of learning disabled and emotionally mentally-retarded students to deal with small typeface and the pressure of timed exercises with complex sequencing of instructions are no longer serious concerns. The students have been motivated by the program to overcome these challenges.

The value of the series as a form of drill and practice over basic facts is unquestionable. DLM's extensive efforts have been rewarded with a unique combination of state-of-the-art computer programming and proven learning techniques. If you are looking for a set of programs to improve your students' speed and accuracy in basic arithmetic facts, Arcademic Skill Builders® (TM) is well worth considering.

Marguerite M. Mason is Project Director of an Illinois Title 4-C ESEA program, Microcomputer Assistance for Special Students (MASS), and has developed a "Multi-Media Approach to Learning Disabilities" under a grant from the Apple Education Foundation.

W. Bruce Mason, former Administrator and Special Education Teacher, is a microcomputer consultant. The Masons are currently implementing a Federally funded teacher-training project at the Illinois State University.

Cataloging Information

Dewey Decimal: 511
 ERIC Descriptor: Mathematics—
 Computer programs
 Subject Heading (Sears): Mathematics

Source Information

Program	<i>Meteor Multiplication Arcademic Skill Builders</i>
Source	Developmental Learning Materials One DLM Park Allen, TX 75002
Equipment	48K Apple II or Apple II Plus Disk Drive Color Video Monitor
Package	One Diskette, Teacher's Manual, Student Record Sheets, and Flash Cards
Suggested Retail Price	\$39.00 Available from Source

Applesoft BASIC Tutorials Microcomputer Workbook: Apple II Edition

Sterling Swift Publishing Company

Sharmon J. Hilfinger

A simple idea! Use the computer to teach about the computer.

One of the first questions asked by a new user of a microcomputer system is "How do I make it work?" There are usually booklets and manuals supplied with the new system; but to the novice, frequently these add to the confusion. A natural solution is interactive instruction, using the computer in a conversational mode to guide the new user through the learning process.

Objectives

The preface of the *Applesoft BASIC Tutorial Microcomputer Workbook*, authored by James L. Poirot and Donald A. Retzlaff, states that the objectives of the courseware are twofold: 1) to teach the computer novice how to make the computer work, and 2) to "use the computer to teach about the computer," which in this case means using the computer to teach BASIC programming. There is an introductory tutorial on loading a program from a diskette or cassette tape, and an introduction to the Apple keyboard, after which the emphasis of the courseware is on teaching BASIC programming. Because the courseware gives a limited treatment of the first objective (how to make the computer work), the title may be slightly misleading. This is a tutorial in BASIC programming that assumes no prior knowledge of computers or programming.

Users

The workbook does not indicate an intended age group for the courseware. The presentation of ideas does assume that the user has some understanding of variables in mathematics. So, it could be assumed that this is written for someone with at least an introduction to algebra. Age is not necessarily relevant to the use of the program.

Documentation

The workbook and software are completely integrated—one is not useful without the other. The workbook provides a combination of written exercises (describing what the student has discovered) and computer exercises.

The first task of the student is to follow the instructions for Loading a Tutorial From a Diskette, which

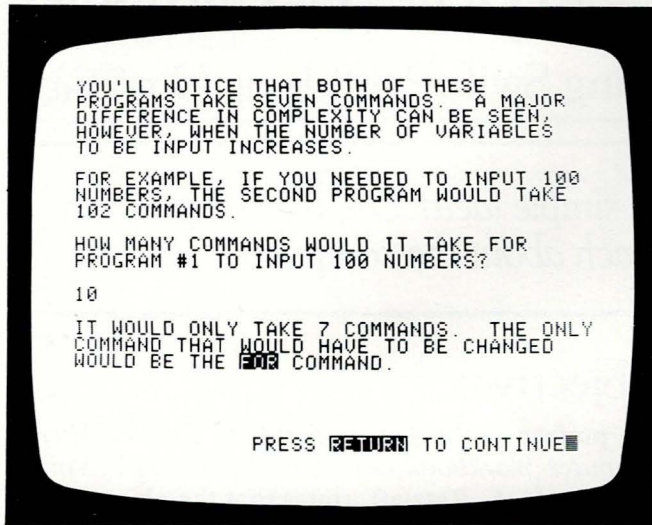


Figure 1

are outlined in a step-by-step procedure in the workbook, starting with "0. Read *all* instructions first." However, when it comes to loading the program, it is not clear that the user must press RETURN in order for the program to load. The intention to have a careful, step-by-step procedure is clear; but the presentation fails by omitting an important step such as "press RETURN".

The writing is generally clear, but the absence of a good copy edit is noticeable. Items such as "Can more than one variable be inputted?" or "variables starting with an alphabetic character" need to be rewritten to eliminate "computerese."

Overall Instructional Design of Product

The *Applesoft BASIC Tutorial Microcomputer Workbook* package contains 12 tutorials plus an Applications module. The concepts are presented on the computer, with the exercises (workbook and computer) presented in the workbook. In working through the tutorials at the computer, one has the feeling that the utilization of computer and that of the workbook are reversed. The concepts are presented briefly and quickly on the computer, and the interaction is supplied by the workbook.

The authors have been careful not to overload the screen with text, so that any given frame is easy to read and one gets a sense of progress in moving from frame to frame (Fig. 1). The rules are presented on the computer in a page-turning format that is an inappropriate use of the computer for covering the amount of material addressed in this product. In order to make the screens uncluttered and readable, concepts are not adequately covered. More than once, two separate concepts are presented in one screen. For example, the concepts of variable evaluation and assignment statement are each introduced on the same screen.

Commands or concepts are used (such as the PRINT statement) before they are explained. This could be attributed to a discovery approach to the material. However, the overall structure of the product does not fit the discovery method. Each lesson in the workbook begins with the statement "DO NOT perform the exercises in this workbook until after you have taken this TUTORIAL." The tutorials cover a number of concepts very quickly. Then, the student is asked to answer questions and perform exercises. The authors missed the opportunity for a more interactive product by not allowing the student to follow exercises in the workbook throughout the tutorial.

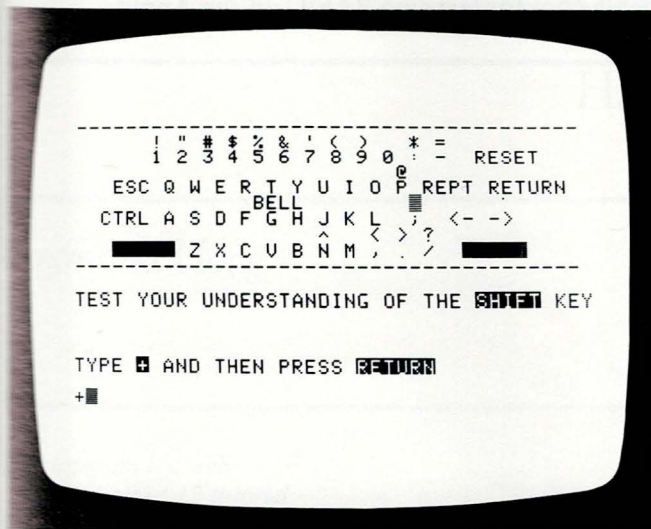


Figure 2

The potential for more interaction within the computer tutorial also exists. For example, in Lesson III: Beginning Programming, the student is shown a program that has the code lines out of order. The LIST command is being introduced. The computer tutorial explains that the LIST command will rearrange the program into its proper sequence. Instead of asking the student to type LIST and watch the computer execute the command, the tutorial is simply a page turner. The next frame shows the program in sequence. The point of using the computer to teach about the computer is lost.

The presentation of programming concepts suffers from a lack of context. Rather than building meaningful programs (even if it would be a simple program to print the student's name, or do a metric conversion), the student learns in a vacuum concepts such as variables, PRINT, and LET. The effect is a disjointed presentation. This, plus the scant development of concepts, makes it difficult to get a feel for the computer environment.

Instructional Design of Programming

The computer programs are very rigid in the responses they will accept. They will read only literally correct answers. For instance, in the section Arithmetic In Basic, the student is asked to type "2 + 4". If the student types "4 + 2", the program response is "incorrect". Similarly, if an answer is preceded or followed by a space, the program indicates an "incorrect" response. Once into a tutorial, there is no way out until all frames have been covered.

These problems are not only irritating, they give the computer novice a poor relationship with the computer. The computer (which in this program is not very smart) is controlling their work environment too tightly.

The program does not break, which is a point in its favor.

Summary

The concept of this courseware (learning about the computer on the computer) is valid, and the topic (teaching BASIC) is appropriate for many computer novices. The execution of the program is disappointing. The best part of the program is the presentation of the computer keyboard (Fig. 2). The screen shows the keyboard configuration and has the student use the special function keys (CTRL, RETURN,

Sharmon Hilfinger, a graduate of Illinois Wesleyan, Bloomington, has a varied background in publishing, both on the marketing and editorial sides. Hilfinger published the "1981 Courseware Market Report" for

Shotwell & Associates and, in her present role as a Vice President of DesignWare, Inc., is responsible for the development and marketing of courseware-development services.

BELL, etc.). This section is appropriately interactive. The workbook does have many discovery exercises to help the student explore the concepts presented in the computer tutorial. But, the lack of an overall context and smooth presentation of concepts makes the workbook more tedious than engaging. It is difficult to recommend the program as a viable tool for learning BASIC on the Apple.

Cataloging Information

Dewey Decimal Number: 001.64

ERIC Descriptor: Programming Languages

Subject Heading (Sears):

1. BASIC (Computer program languages)
2. Programming Languages (Electronic computers)

Source Information

Program	<i>Applesoft BASIC Tutorials Microcomputer Workbook: Apple II Edition</i>
Source	Sterling Swift Publishing Company 1600 Fortview Road Austin, TX 78704
Equipment	32K Apple II or Apple II Plus Disk Drive Video Monitor or Television with RF Modulator
Package	Two Diskettes and Workbook Four Cassettes and Workbook
Suggested Retail Price	\$74.95 Diskette \$74.95 Cassette Available from Local Dealer

Programming in Apple Integer BASIC: Self-Teaching Software

Hayden Microcomputer Program Tapes
Hayden Book Company, Inc.

Raymond Hoag

*An encouraging, self-paced,
programmed instructional course
for the beginner.*

This program is a well-conceived set of 12 tutorial lessons designed to teach Integer BASIC language to the novice. In addition to being an effective means of learning to use the computer, these interactive lessons provide mobility within the program, with appropriately spaced checkpoints to permit review and/or re-entry into material within a given lesson. Periodically the user is questioned to evaluate his or her own relative progress. In the process, the user, as well as the computer, take on new independence.

Pedagogy

With perhaps two or three exceptions, the textual material in *Programming in Apple Integer BASIC: Self-Teaching Software* is presented in concise, accurate, and understandable segments. The segments are well ordered. When the learner encounters the end of a segment (indicated by three colons on a line), he/she is instructed to press the spacebar. Either additional text is presented; or the screen is cleared and new text is presented; or the screen is partially cleared, retaining a relevant example, and then further text on the current topic is presented (Figs. 1, 2, and 3). Graphic commands are always first shown as text, then illustrated, step by step (Figs. 4 and 5).

Probably the most innovative aspect of the lessons, authored by Scott Banks and James S. Coan, occurs when checkpoints are reached. At these appropriate points a multiple-choice question is posed to assess the learner's comprehension. Not only is each right or wrong answer properly responded to, but special responses are permissible at these junctures. If the learner responds by typing a 0 (zero), a help message is printed (Fig. 6). By typing a 9, the learner enters the "let's talk" module. Available options include repeating the current lesson-section, viewing the lesson's table of contents, selecting a new-lesson section, quitting the lesson or continuing where the user left off. By using these options, the learner has a more interactive session and is in greater control of his/her progress through the lessons.

Illegal responses to questions are prohibited. The computer "beeps," rejects the learner's response, and will proceed only when an allowable response is

```

      STEPPING OUT WITH 'FOR...NEXT'
A 'FOR' LOOP DOESN'T HAVE TO COUNT BY
ONE'S.
      USING 'STEP', WE CAN CAUSE A 'FOR' TO
COUNT ANY WAY WE WISH.
10 FOR X=3 TO 10 STEP 2
20 PRINT X,
30 NEXT X
40 END

```

Figure 1

```

      STEPPING OUT WITH 'FOR...NEXT'
A 'FOR' LOOP DOESN'T HAVE TO COUNT BY
ONE'S.
      USING 'STEP', WE CAN CAUSE A 'FOR' TO
COUNT ANY WAY WE WISH.
10 FOR X=3 TO 10 STEP 2
20 PRINT X,
30 NEXT X
40 END

>RUN
3      5      7      9

```

Figure 2

```

      STEPPING OUT WITH 'FOR...NEXT'
A 'FOR' LOOP DOESN'T HAVE TO COUNT BY
ONE'S.
      USING 'STEP', WE CAN CAUSE A 'FOR' TO
COUNT ANY WAY WE WISH.
10 FOR X=3 TO 10 STEP 2
20 PRINT X,
30 NEXT X
40 END

>RUN
3      5      7      9

      THE INITIAL VALUE OF X IS 3, AND IT
ADVANCES BY 2 UNTIL IT EXCEEDS 10.
      SINCE 11 IS GREATER THAN 10, IT CAUSES
THE LOOP TO EXIT.

```

Figure 3

given. The learner is allowed to view the material as long as needed. However, an amusing but attractive feature occurs when a minute or so elapses with no user response. A short alarm is sounded, and the message "Are you there?" is displayed (Fig. 6). This can be startling if the user is in dreamland. But it serves well in keeping the learner's attention focused on the lesson.

Documentation

The lessons are provided by the vendor either on two tape cassettes or two 5 1/4-inch mini-diskettes.

The 12 pages of documentation that accompany the lessons are distributed in a quality three-ring notebook. The documentation includes the preface, a table of contents, instructions for loading the software, a directory of the lessons (including the titles of the major sections of each lesson), an exposition on how to use the learning method employed in the lessons, program examples from the lessons, and a list of companion books in BASIC available through Hayden Book Company.

The lesson directory is a handy guide if one wants to pick out a particular topic to study or review. The learning method used in the lessons is introduced and illustrated in Lesson 1. Having this instruction for using the learning method in written form is handy for reference once the learner has proceeded beyond Lesson 1.

The distribution medium for the set I reviewed was tape cassette. My prior experience with the Apple was exclusively with diskettes. I ran into difficulty in loading the tapes when following the instructions provided. After calling the local distributor, I found that the tapes had to be listened to first in order to determine a "sound leader" prior to each lesson. Once the tape was positioned to this "sound leader," the supplied instructions worked fine.

At the conclusion of the written loading instructions, there is a statement that could easily be overlooked. It encourages learners to experiment, have fun, and try writing some of their own programs. In my opinion this could have been placed in large block letters across a page and reinforced with a set of problem statements to be programmed, correlated to, and referenced by each lesson. Solution lessons could have been provided elsewhere in the program's documentation.

Inclusion of a trainer's guide would enhance the documentation. Suggested ways of incorporating lessons and materials in the learning environment would further the application of the total program.

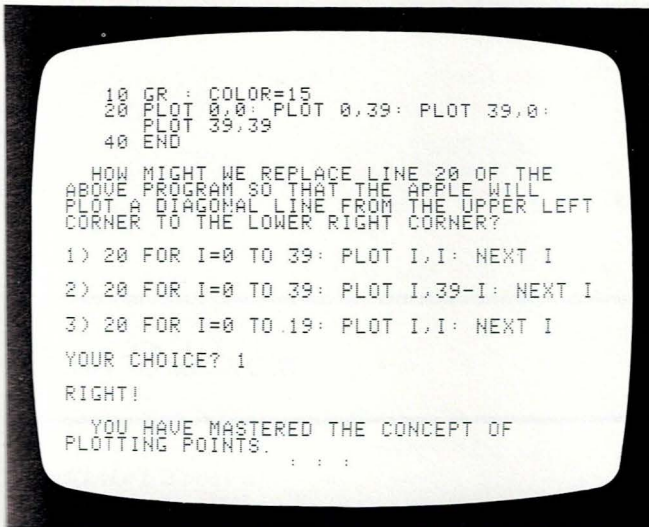


Figure 4

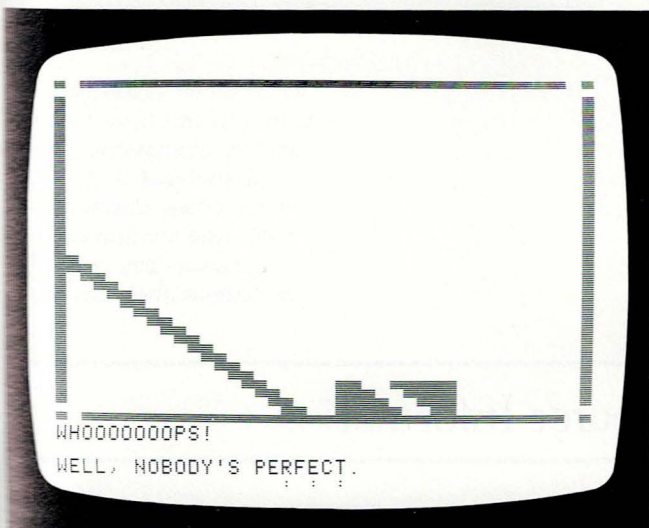


Figure 5

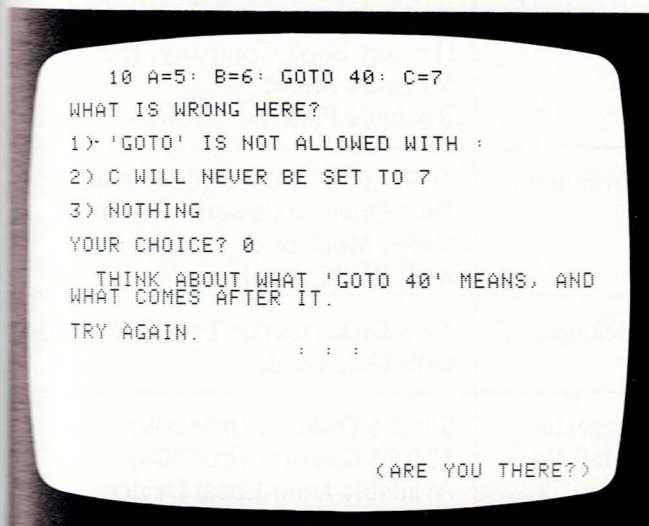


Figure 6

Users

Almost any business could well use these lessons for employee on-the-job computer literacy. Micro-computer systems with these lessons could be checked out for use at home. Employees could learn BASIC and become familiar with the computer operation at the same time. This education, and employee benefit, could hasten the introduction of computerized tasks, and lessen the "awe" that often surrounds the computer, plus increase the business' productivity.

There are several educational formats that could be used. In an elementary school, for example, gifted students could go through the lessons as an independent-study project. Using the computer, they could freely and, I suspect, productively express their creativity.

There would be several ways the lessons could be used in conjunction with the typical BASIC programming class. The lessons could be assigned as a regular class activity. They could be used as a course supplement for those who want further explanation, for those who missed a class and need to catch up, and for those who want to move ahead at a faster pace.

As an alternative to a BASIC programming class, a set of microcomputers using these lessons could be rented for a given time period. Learners would be checked out on the computer and software prior to physically leaving with the systems. Possibly a pre-test could be administered. During the time that the computers are at home, a "hotline" could be established to answer learners' questions. When the microcomputer systems and materials are returned and checked in, the student-learners could be re-tested to assess their increased knowledge of the Integer BASIC language. Appropriate programming tasks could be assigned as homework to promote further mastery of the language.

As the reviewer, I feel that *Programming in Apple Integer BASIC: Self-Teaching Software* is a well-conceived set of tutorial lessons, designed to teach a beginner the Integer BASIC language. The interactive lessons are self-paced, with appropriately spaced checkpoints for review, evaluation, and progress based on mastery.

Dr. Raymond Hoag, Director of Academic Computing at Grand Rapids Junior College, is past president of the Michigan Association for Computer Users in Learning, and the Association for Systems Management, Grand Rapids Chapter. Hoag is the local coordinator for the Association for Educational Data Systems, having authored in-

structional packages for microcomputers through its national workshops. In addition to instructing graduate courses in Computers as a Learning/Instructional Tool, Dr. Hoag originated The Family College at Grand Rapids Junior College, which conducts BASIC microcomputing-programming classes for children and adults.

Cataloging Information

Dewey Decimal Number: 001.64

ERIC Descriptor: Programming Languages

Subject Heading (Sears):

1. Programming Languages (Electronic computers)
2. BASIC (Computer program languages)

Source Information

Program	<i>Programming In Apple Integer BASIC: Self-Teaching Software</i>
Source	Hayden Microcomputer Program Tapes Hayden Book Company, Inc. 50 Essex Street Rochelle Park, NJ 07662
Equipment	16K Apple II or Apple II Plus Disk Drive or Cassette Recorder Video Monitor or Television with RF Modulator
Package	Two Diskettes (or Two Cassettes) with User Guide
Suggested Retail Price	\$39.95 Diskette (#05009) \$29.95 Cassette (#05004) Available from Local Dealer

Elementary, Volume 4

Minnesota Educational
Computing Consortium
MECC

Dotty Calabrese

Another view of problem solving.

Odell Lake is a body of water in the Cascade mountain range of Oregon. It is also the name of a computer simulation designed to illustrate ecological relationships and an awareness of plant and animal life.

Game strategy builds educational content into a format that is entertaining and purposeful. As indefatigable practice aids, each program on this diskette provides a challenge and encourages logic and decision-making.

Although ODELL LAKE was written by a tenth-grade biology teacher in Minnesota, and was originally intended for use at that grade level, the present version of this program has a reading level of 3.2 and is recommended for use with students as young as fourth grade. I have used the program successfully with all ages from second graders through eighth graders and adults. Most of my users have called it one of their favorite programs. When asked for their comments on ODELL LAKE, my students have given replies such as, "Good graphics" and "It's fun".

The educational objectives of ODELL LAKE, taken directly from the accompanying support booklet, are:

1. To simulate the life of a fish and to make the decisions necessary for survival.
2. To understand the food (predator-prey) relationship between fish.
3. To generalize a size relationship between fish.
4. To create an awareness for the plant and animal life forms found in water.

In my experience using ODELL LAKE with students, I believe it fulfills these objectives.

In the ODELL LAKE program the user must role play one of six fish:

1. Whitefish
2. Chub
3. Blueback Salmon
4. Mackinaw Trout
5. Rainbow Trout
6. Dolly Varden

actually found in ODELL LAKE, and make decisions to discover what behaviors lead to survival.

The first time I played ODELL LAKE, I knew nothing about the program and proceeded to play it in

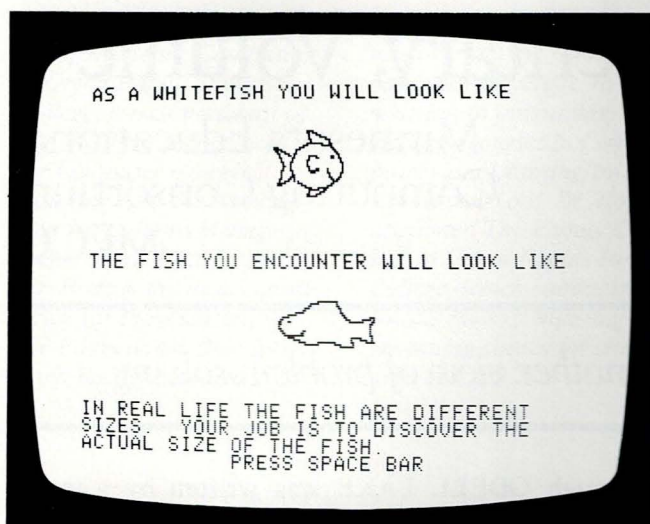


Figure 1

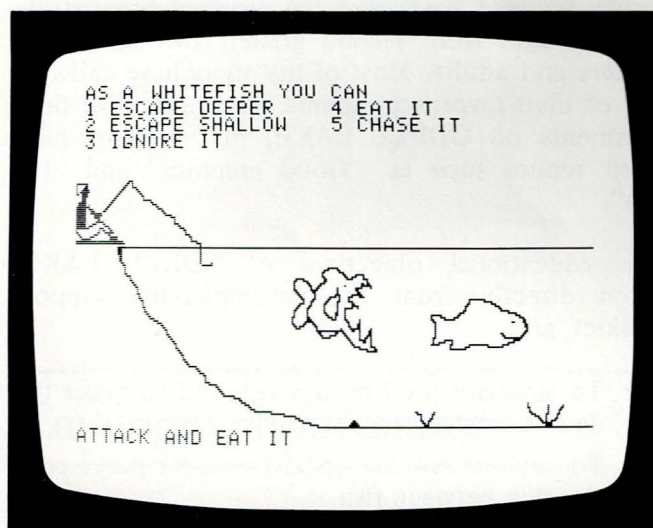


Figure 2

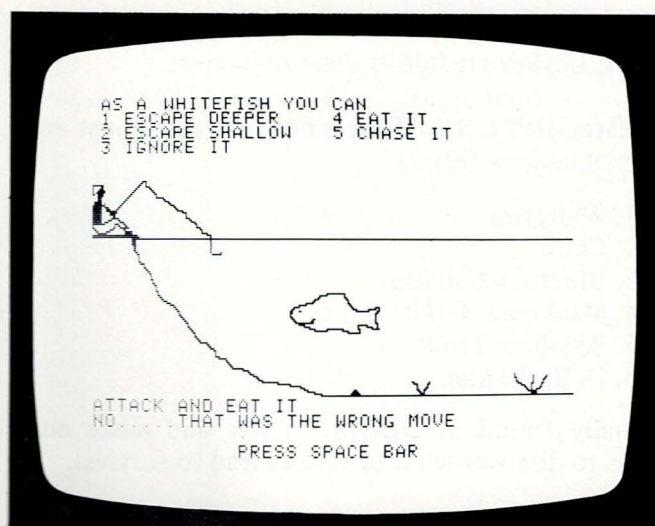


Figure 3

the discovery mode, as it is intended to be used. I read the simple instructions, found them to be clear, and chose to be a Whitefish on my first turn (Fig. 1). In the next frame I encountered a Chub swimming back and forth in front of me. I was given the five choices provided throughout the game:

1. Escape deeper
2. Escape shallow
3. Ignore it
4. Eat it
5. Chase it

Being a believer in non-violence, I chose #3, to ignore it. The screen promptly displayed the message "That fish just ate your lunch!" Obviously, that move in that situation was not the best one for a Whitefish.

In the following frame I encountered a Rainbow Trout. I took a more aggressive stand and chose #4, to try and eat it. The Whitefish on the screen opened its mouth, exposed large teeth, and swam menacingly toward the Rainbow Trout (Fig. 2), only to have the Rainbow Trout remain out of reach and swim away. The screen displayed the message "You cannot kill this fish, and you used a lot of energy needlessly".

The next fish I encountered was a Dolly Varden. I had never heard of such a fish, so I took my chances and decided to choose #4, to try to eat it. As the Whitefish swam toward the Dolly Varden, the Dolly V. turned, opened and closed its jaws repeatedly as it swam toward the Whitefish, and devoured it! The screen displayed the message "No...that was the wrong move" (Fig. 3), followed by, "Remember, a Dolly Varden will eat a Whitefish".

On my next four runs I continued to role play a Whitefish, and was finally able to encounter all five other fish successfully. At that point the next phase of the game loaded automatically. Still role playing the Whitefish, I encountered Animal Plankton, Water Insects, an Otter (Fig. 4), and an Osprey, and successfully ate or escaped from each one. I proceeded to role play other fish, exploring the many possible successful and unsuccessful moves. At one point I was even caught by the fisherman who sits fishing at the edge of the lake (Fig. 5). My only real disappointment with the graphics of ODELL LAKE came when I allowed the Osprey to catch the fish, and found that the Osprey is shown carrying off its prey in its bill rather than in its talons, as in real life. The bird-watcher in me still winces each time I see that frame (Fig. 6).

Several weeks and many uses later the program continues to hold my interest. This also appears to be

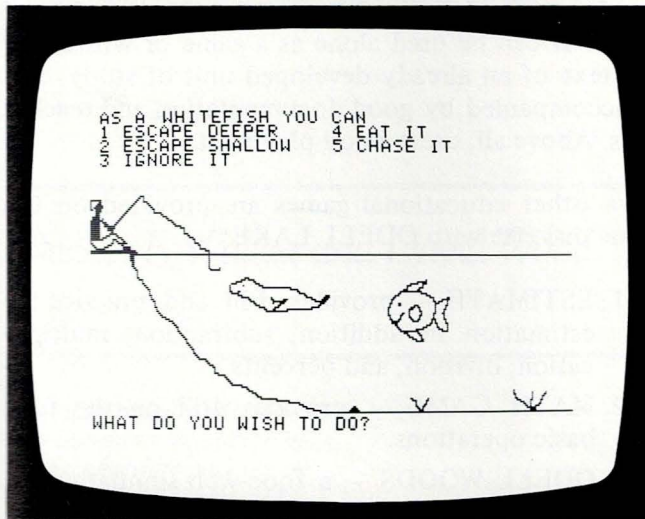


Figure 4

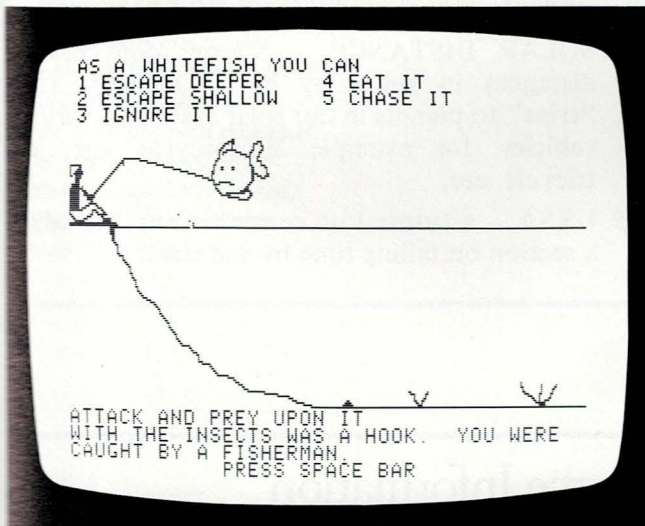


Figure 5

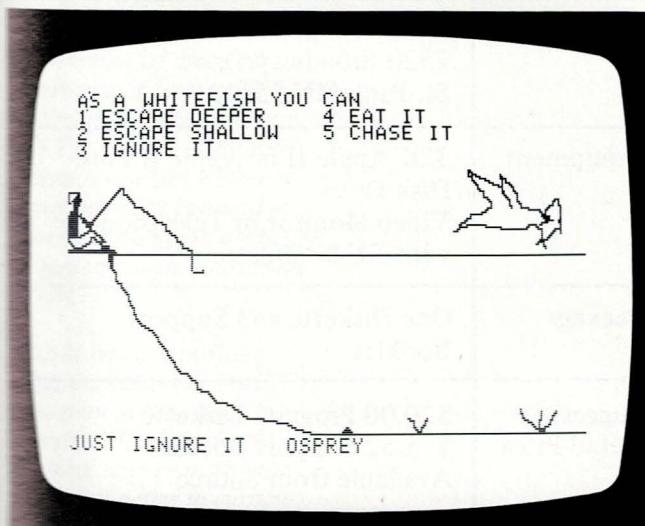


Figure 6

true for the children I teach. Even the ones who have mastered the game repeatedly return by choice to play it.

Among the children with whom I work is a group of fifth and sixth graders who recently took an ecology/biology boat trip out on the San Francisco Bay as part of a unit of study about our geographic area. They dragged nets for bottom dwellers, gathered plankton and other micro-organisms, identified fish and sharks, etc. The ODELL LAKE program has supplemented their unit of study.

I became excited by the possibility of taking a next step and have these students create their own San Francisco Bay game using the ODELL LAKE program as a model or "shell." Much to my dismay, I discovered that the write-protect sold by Apple and used by MECC on the version of ODELL LAKE I obtained does not allow listing of the program. However, a non-write-protect version that can be listed is used within the state of Minnesota and is sold to school systems under contract with MECC. I would encourage teachers with these non-write-protected versions to consider the possibility of a class project using ODELL LAKE as a shell program that students and/or teachers could modify to indicate a local body of water with local fish, mammals, and birds.

Although I know and respect the reasons behind using write-protect, I am concerned by its growing use in educational software. Some of the greatest learning experiences for my young programmers happen while they are playing an existing game, when they ask, "Hey, how did they do that?" and then are able to find out! I also believe that "shell" games in which the teacher or student can access the listing and adapt the program to his or her own situations and interests are proving to be some of the most useful and certainly some of the most versatile educational games available. To my knowledge, both of these possibilities are eliminated by using the existing write-protects on the market. I believe that ODELL LAKE has great potential as a shell game, and hope that MECC will consider developing a package of support materials for this use and market it to the general public with a non-write-protect version of the program.

I consider ODELL LAKE to be one of the best educational programs I have seen to date. It is original, appeals to a wide age group, and holds interest. It contains clear instructions, and all feedback seems carefully worded so as not to leave the user feeling "stupid." ODELL LAKE makes good use of the graphics capabilities of the Apple II microcomputer. It employs the discovery mode of learning, and

Dotty Calabrese received her A.B. in psychology from Cornell University and master in education from Stanford University. Ms. Calabrese is presently an instructor and computer-education specialist at Peninsula School, a noted private, alternative school in Menlo Park, California. Calabrese was one of several educators featured in "Don't Bother Me, I'm Learning!", a one-hour computer educational film shown on PBS in January 1980.

teaches several general concepts of an ecological topic. It can be used alone as a game or within the context of an already developed unit of study, and is accompanied by good documentation and teacher aids. Above all, users enjoy playing it.

Five other educational games are provided on the same diskette with ODELL LAKE:

1. ESTIMATE – provides drill and practice on estimation in addition, subtraction, multiplication, division, and percents.
2. MATH GAME – provides drill on the four basic operations.
3. ODELL WOODS – a food-web simulation involving animals found in northern Minnesota—similar to ODELL LAKE but without graphics, with slightly more complex situations; involves more reading and typing on the part of the student.
4. SOLAR DISTANCE – teaches concepts of distances in space by having students take “trips” to planets in our solar system in various vehicles; for example, motorcycle, car, jet, tricycle, etc.
5. URSA – a tutorial on constellations, including a section on telling time by the stars.

Cataloging Information

Dewey Decimal Number: 500
 ERIC Descriptor: Science Activities
 Subject Heading (Sears): Science—Computer programs

Source Information

Program	<i>Elementary, Volume 4</i>
Source	MECC Publications 2520 Broadway Drive St. Paul, MN 55113
Equipment	32K Apple II or Apple II Plus Disk Drive Video Monitor or Television with RF Modulator
Package	One Diskette and Support Booklet
Suggested Retail Price	\$20.00 Program Diskette \$ 5.52 Support Booklet Available from Source

Physics CompuLab

EduTech

Decker F. Walker and David Wood

Computer-simulated physics experiments.

The experiments selected for computer simulation in this courseware package are those that are considered difficult or impossible to perform in a conventional student laboratory situation. These experiments augment, rather than replace, traditional experiments, and provide students with an interactive experience with physical concepts and phenomena. The reviewers consider it flawed, but definitely a useful introduction to physics.

Decker Walker, Ph.D., is co-director of Stanford University's Interactive Educational Technology (IET) program. A former physics teacher, Walker is professionally involved in the design and evaluation of microcomputer courseware tools.

David Wood is pursuing masters studies in the IET program at Stanford University. Wood, formerly a natural science instructor, has been active in curriculum development and evaluation for 15 years.

EduTech's CompLab is a package of computer-simulated experiments in introductory physics authored by Alan Cromer. The package consists of a teacher's manual, student lab book, and six diskettes. Each diskette holds one of the following programs:

Experiment 1	Statistics
Experiment 2	Parabolic Motion
Experiment 3	Non-Constant Acceleration
Experiment 4	Planetary Motion
Experiment 5	Wave Motion
Experiment 6	Electric Field

Although the author never states what age group the programs are designed for, we feel they would be best used with high school juniors or older.

Description

Most of the programs simulate the phenomena studies in a standard introductory physics experiment. Students carry out the experiment by making observations of the screen and then computing the laws describing the observed motion. These observations usually involve measurements, for example, of wavelength, frequency, or amplitude.

The lone exception to this pattern is Statistics, a program on probability. Students collect data for this experiment by repeatedly sampling a bucket full of pink and blue balls (not supplied in the package). They then enter their data into the computer and create histograms of the relative frequencies of each color. While they do this, the program is randomly generating 600 more such frequencies. After a five-minute delay (no way around it), the computer generates a grand histogram of its own, incorporating all 600 trials.

Programs for experiments 2–5 first present the student with the equation describing the motion to be studied. The student then assigns values to the different variables; and the computer plots out a simulated free fall, projected path, or planetary path using the assigned values. By measuring distances

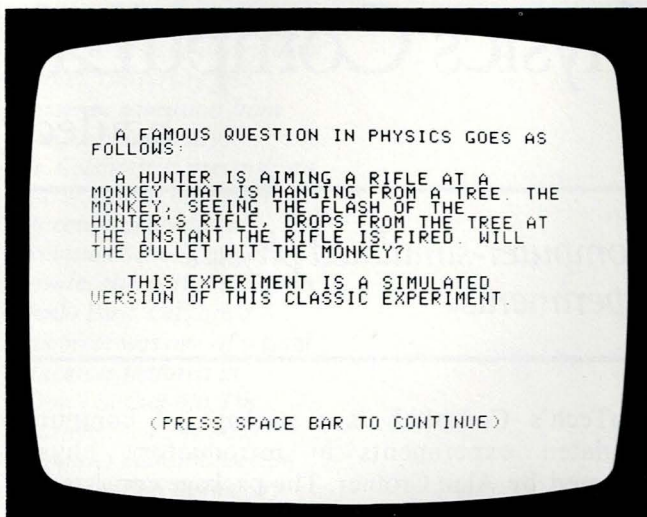


Figure 1

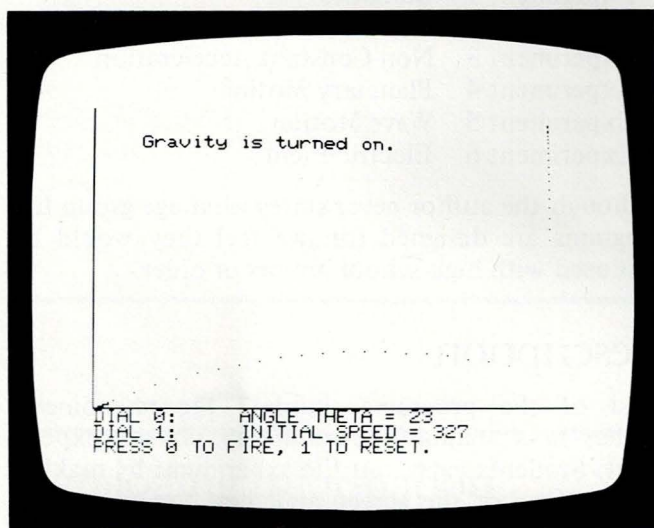


Figure 2

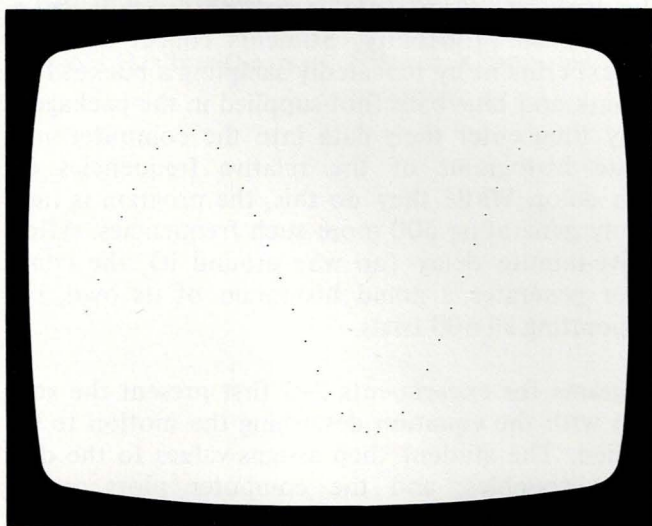


Figure 3

and times, students gather data with which to calculate the coefficients of the equation. The mathematics involved is beginning calculus, and the work is reasonably challenging for a typical student; but the programs are quite usable without this mathematical sophistication.

The program for experiment 6, Electric Field, allows students to map an electric field (or lines of equipotential) by moving a simulated electric charge around with a joystick. A press of the joystick button displays a short line segment indicating the direction (not magnitude) of the field at that point. Since this is how the concept of our electric field is defined in the first place, the program permits a direct simulation of this concept.

Each program ends with an amusing gamelike activity. The best of these is the concluding scene in Parabolic Motion (Fig. 1). That program poses the hunter-monkey problem: if the hunter fires just as the monkey sees him and drops from its branch, will the bullet hit the monkey? Students can control both the velocity of the bullet and the acceleration of gravity in this simulation (Fig. 2). Another interesting activity is the last exercise in Electric Field. Here the student is invited to discover the location of an 'invisible' charged particle by sampling the electric field at various points with a charged probe. Joysticks are needed to manipulate the probe. The final sections of Non-Constant Acceleration and Wave Motion, however, are too confusing visually to be either constructive or amusing.

Technical Quality

The programs all started smoothly and ran flawlessly every time. We were unable to 'crash' or hang up the program with one trivial exception: Projectile Motion has no exit, so the user must turn off the machine to exit. In fact all the programs end suddenly; the only indication a user receives is a prompt at the bottom of the screen.

The programs were, in general, easy to use but could stand some improvement. The directions are generally clear, but they need to be shortened in some cases. Also, there is no way to return to them short of starting over again. Consistency could be improved. For example, in order to exit, students type either \emptyset , End, or press the spacebar, depending on the program. In order to respond, the user usually types both the response and presses the RETURN key, which seems unnecessary. None of these difficulties is substantial enough to constitute a serious problem, however, and one very good feature is that all branching is student controlled.

Graphics are an integral part of the entire packet. In most cases they worked effectively, but not always.

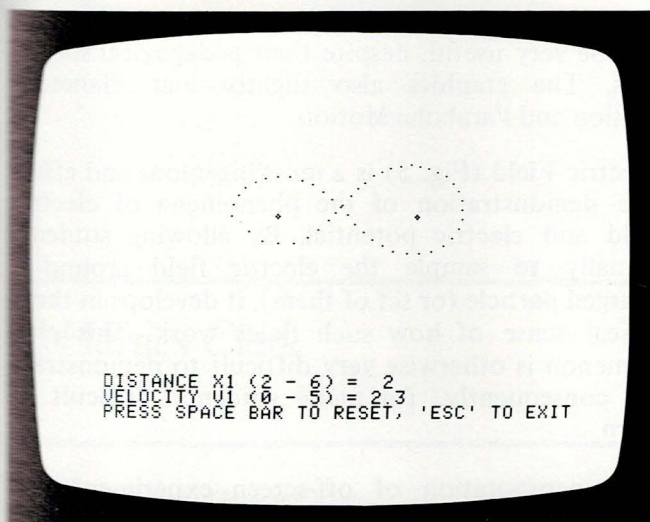


Figure 4

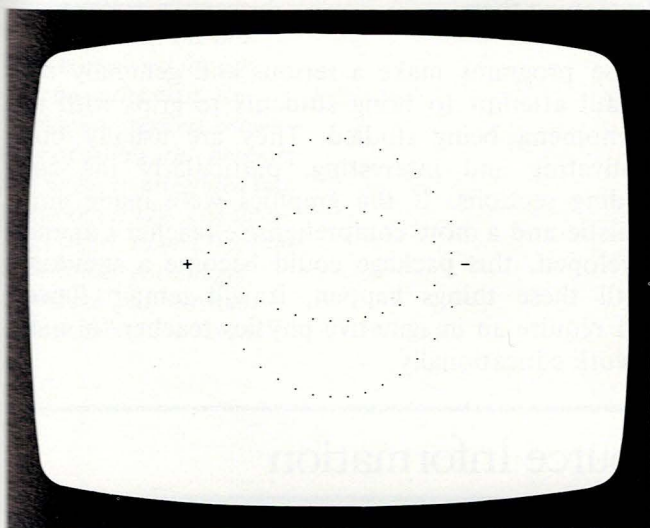


Figure 5

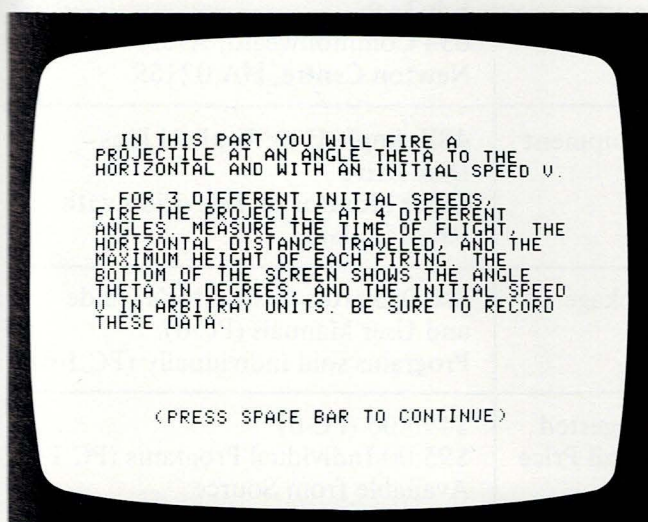


Figure 6

Lines are composed of a series of dots, positioned about six diameters apart from each other. Motion is simulated by generating new dots, each occupying the spot a moving particle would occupy at that time. The relatively slow rate of dot generation combined with the fairly large separations between dots can make both the lines and motion appear highly discontinuous at times—perhaps similar to watching a bird in flight with a stroboscope. It should be said that we viewed all the programs on a black-and-white monitor; perhaps if the lines were different colors, this problem would be less serious. Documentation makes no mention of a color monitor being required or desirable. This technical problem adversely affected both Non-Constant Acceleration and Wave Motion. Wave Motion is too slow and jerky to be believable, an important factor in the educational value of this program. The dot spacing in Non-Constant Acceleration is so uneven that the simulation is barely comprehensible at times.

Quality of Content

The content of the six diskettes is good, standard introductory physics. All the physics is accurate. The phenomena chosen for simulation are essential parts of any introductory classical physics course. They are not a complete set of the important laboratory experiments in introductory physics, of course, but only a selection of six from, perhaps, two dozen standard ones.

Within this self-imposed body of content, the only fault we could find is the absence of applications. A student who experienced only these programs would have no idea of the enormous range of situations to which these phenomena apply. The student will have had exposure only to the basic phenomenon itself. Any teacher using the programs will need to ensure that the student associates what the screen is doing and actual events in the world.

Also, opportunities are missed to help students make a closer connection between the mathematics and the simulated patterns. At no time does an equation and a simulated curve appear on the screen at the same time (Fig. 3). Curves are never labeled, nor are the data tables displayed alongside graphic displays. The relations between numbers, variables, and graphical plots remain hidden.

Instructional Quality

The programs are probably better than other media (text, film, or traditional laboratory) for building an intuitive understanding of the phenomena being studied. They are also flexible enough to be used in many different types of physics classes with differing student ability levels.

In fact, they are rich educationally, offering myriad instructional opportunities to an imaginative teacher. All except Statistics, for example, could be used as visual aids for a lecture on the appropriate topic. Or, students could work in pairs, with one setting goals or targets for the other to hit by adjusting the parameters of an equation (Fig. 4). Or, the teacher could challenge the student to launch a planet whose orbit passed through a specified point. Or, find waves which when superimposed form a certain shape. The possibilities are endless.

Unfortunately, the documentation does not help a teacher discover this potential. The few sections entitled 'teaching notes' are simply commentaries on the pitfalls students commonly encounter. This paucity of information is in marked contrast to the technical domain, where generally complete information and guidance are provided. A teacher who intends to use these programs should, therefore, expect to spend several hours previewing each diskette and deciding what to do with it in class. The documentation also fails to include materials or suggestions for evaluation.

General Conclusions

As a unit this package is good but not great. Some of the programs are better than others: we felt that Parabolic Motion, Statistics, Planetary Motion and Electric Field are all technically sound as well as educationally valuable. Non-Constant Acceleration

and Wave Motion need better graphics before they will be very useful, despite their pedagogical soundness. The graphics also slightly mar Planetary Motion and Parabolic Motion.

Electric Field (Fig. 5) is a most ingenious and effective demonstration of the phenomena of electric field and electric potential. By allowing students actually to sample the electric field around a charged particle (or set of them), it develops in them a real sense of how such fields work. This phenomenon is otherwise very difficult to demonstrate or, consequently, for many students, difficult to learn.

The incorporation of off-screen experimentation into the simulation in Statistics is highly innovative, as is the idea of doing on-screen measurements (Fig. 6). These techniques are sure to be used more in future CAI, and the author is to be applauded for developing them.

These programs make a serious and generally successful attempt to bring students to grips with the phenomena being studied. They are usually both motivating and interesting, particularly the concluding sections. If the graphics were made more realistic and a more comprehensive teacher's manual developed, this package could become a standout. Until these things happen, it will remain flawed and require an imaginative physics teacher to make it work educationally.

Cataloging Information

Dewey Decimal Number: 530
 ERIC Descriptor: Physics Experiments
 Subject Heading (Sears): Physics—Experiments
 —Computer programs

Source Information

Program	<i>Computer-Simulated Physics Experiments</i>
Source	EduTech 634 Commonwealth Ave. Newton Centre, MA 02159
Equipment	48K Apple II or Apple II Plus Disk Drive Color Monitor or Television with RF Modulator
Package	Six Diskettes, Instructor's Guide and User Manuals (PC-0). Programs sold individually (PC 1-6).
Suggested Retail Price	\$495.00 (PC-0) \$95.00 Individual Programs (PC 1-6) Available from Source

Search Series: Geography Search

McGraw-Hill Book Company

W. Hartley Clark

*Simulated voyage to the New World
for gold.*

An understanding of humans and the sea is important to anyone's education. The mating of seamanship and computing in this program has rare value, augmented by the fact that the program is designed for use with an entire class. Requiring no previous knowledge of sailing, this program can be easily integrated into classroom study. The simulated travel, to and from the New World, encourages teamwork and communication among the "crew".

The "Geography Search" unit of the McGraw-Hill Search Series involves the user in the early modern world of sailing and exploration. It assumes no previous knowledge of sailing. Groups of students "sail" ships across the ocean to an imaginary New World where gold is to be found in a ruined city. And then they sail home. The program is designed to engage an entire class of pupils in groups of four or five to take turns every few minutes sailing their own ships in the computer simulation.

A 30-page student's manual (supplemented by a 21-page teacher's manual) plays a key role in the program. At least part of the manual must be mastered, because there are no on-line (screen-displayed) instructions. Each unit in the manual treats a somewhat technical subject, such as wind-direction or water-depth, and each member of a "crew" takes one of these matters as his or her responsibility while "sailing." Each chapter in the manual could consume an hour or more of class time; the whole manual could require more than a week. Only then does the computer enter the picture. On-line instructions (text visible on the screen) might have been more motivating to students; but the class, as a whole, could not read the on-line instructions the way they can read their own copies of the manual. On-line instructions take time, and speed is of the essence in the computer phase of the program because a group, or in some cases, the whole class is taking turns at the computer.

Implementing a sailing exercise on the computer shows originality. It highlights the skills and knowledge of nature a sailor needs, and hence motivates students to learn scientific principles having to do with wind, the compass, weather, and navigation. On the other hand, the program is designed to be used over a period so extended that interest might prove difficult to sustain. The computer phase of the unit itself, following long preparatory reading in the manual, may take two or three class sessions

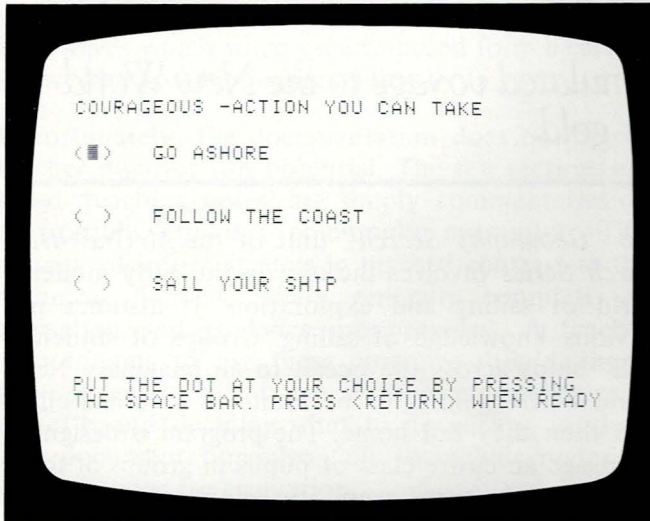


Figure 1

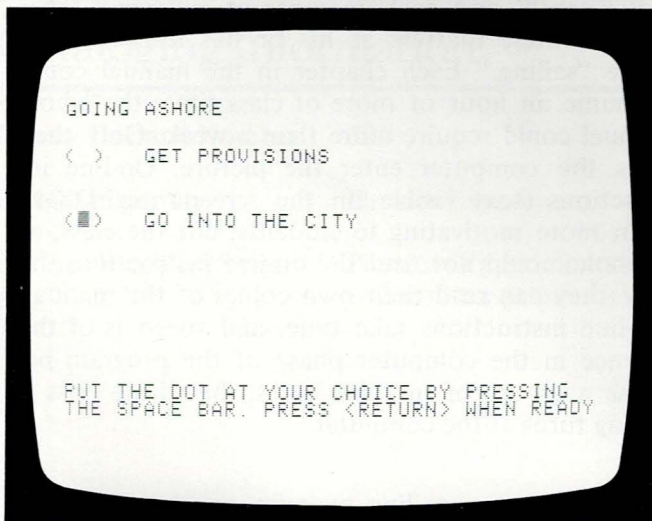


Figure 2

to complete. The computer routine consists of successive sailing days, during which the crew of users for each ship decides on the compass bearing for its course in light of a given wind direction that changes with each user's turn. For possibly up to a dozen plays (days), users may experience little variation in pattern.

The program hedges two ways against the effects of its length. Users may quit and resume where they left off, at a later time. And some text displays of information, vital to user interaction, have no more than ten seconds' duration on the screen. Users cannot cause the text to pause. The "team" concept of play compensates for this haste because team members can specialize in spotting their own assigned information on the furtive screen. Ideally, a team can cross the ocean to the New World in four days (turns); but, as beginners, they will make mistakes and take longer. This is a slow point in the outbound trip that should be shortened.

Life becomes much more interesting for users when they reach the New World because they face decisions about landing (Figs. 1 and 2), taking on supplies, taking on gold, and eventually returning home. Gold is the token object of the voyage, and there are pirates who appear randomly to steal gold from the crew-teams. Although there is considerable verisimilitude in the techniques of navigation in the program, the techniques are crude enough to make it difficult to find one's way home from the New World. This is another slow point in the program that could bear shortening in future editions, along with the outbound trip.

It would help the user, somewhat, to print the location of the New World on the map in the manual or possibly echo its location on the screen when showing the ship's position relative to other teams' ships and to pirates. Admittedly, this would cut into the spirit of discovery that is an objective of the program; but the basic consideration here is that the program is very long and involves, potentially, an entire class. The students will play as beginners, with little or no opportunity for a second try. Complexities that challenge a seasoned user should play less of a role.

Apart from the carefully designed class experience, the program could be played quickly and repeatedly by students using it as a game to see who could get the most gold first; but such use of the program breaks down its verisimilitude. In this program, sailing speed hinges on the sounding of ocean depths; and early modern ships, in fact, had very limited ability to sound depths. If there ever were such a thing as a 1500-foot lead line, it would either break of its own weight if lowered over the side or would

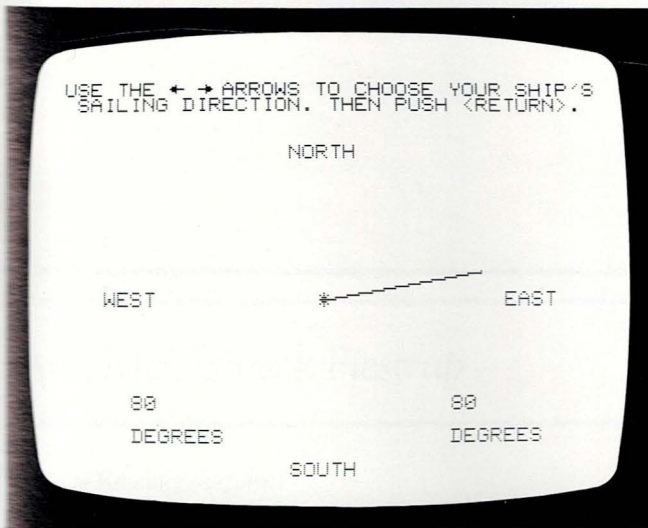


Figure 3

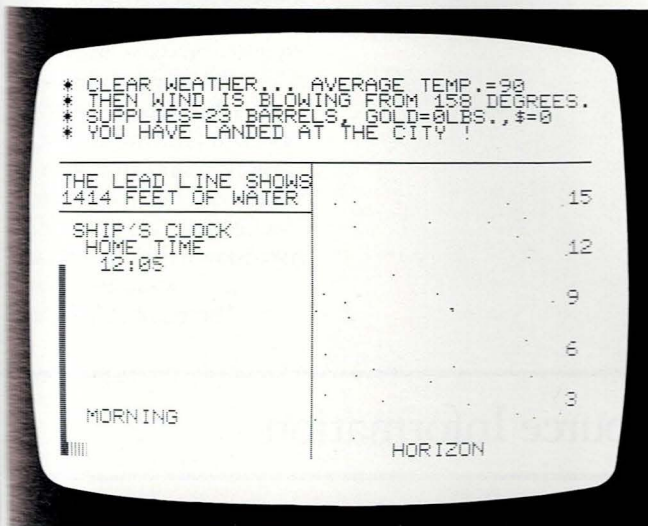


Figure 4

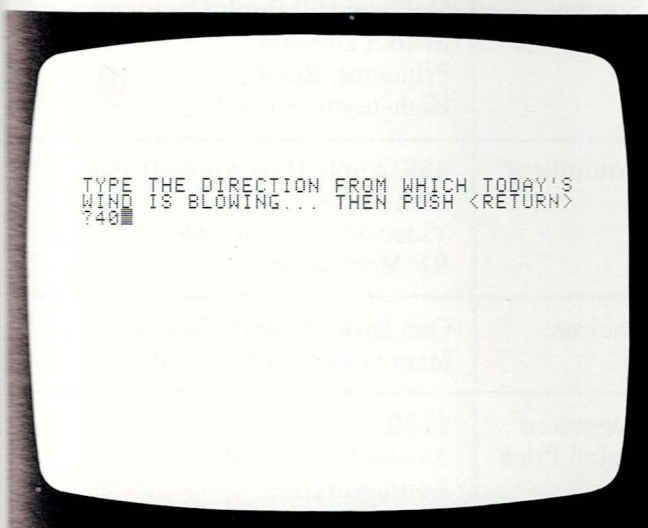


Figure 5

trail behind the ship because of the friction of the line itself in the water and not give a sounding.

The materials do not indicate grade level for the program. It is probably understandable at an elementary level and could easily be used by fourth graders, but some of the navigational concepts might suggest sixth-grade level and above for a sure grasp of technicalities. The student manual is carefully graded for its target users. The program will enhance teaching about natural sea phenomena and about navigational problems, because it sets these facts and principles in the interesting context of sailing (Fig. 3). The success a student has with this computer program can be influenced by his or her mastery of the principles learned from the manual and from the interaction itself. For instance, users sail faster if they understand the wind. Crews starve to death in mid-Atlantic if they start for home without loading redundant provisions in the New World. In many ways the program is a nautical counterpart to MECC's *Oregon Trail* program (reviewed in Vol. I, No. 1 of *The Journal*).

The technical quality of the program is smooth and efficient. It is formatted for running on the simplest of Apple II configurations in that it has no color or sound. Information is displayed to the user in an interesting and appropriate way. Some of the special graphics capabilities of the Apple II are exploited, and the overall design of the graphics makes users conscious of the compass and other nautical concepts. There are just a few graphics—the tides and the firmament—that are not integral to the play (Fig. 4). But even these share space on screens that also show vital information.

On the other hand, the manual as well as screen instructions are unclear as to the need to state wind direction in degrees (Fig. 5) and not as directional names like E for east or NNE, meaning north by northeast. The computer accepts "E" as an entry, and sometimes "N"; but the game becomes very frustrating for users who have failed to enter degrees. They flounder for "weeks" before reaching the New World. Also, the program has a tendency to stop and print an error message if a ship persists in sailing south.

Some understanding of humans and the sea is important to anyone's education, and the mating of seamanship and computing in this program has rare value that is augmented by the fact that the program is designed for use with an entire class. I strongly recommend it for classroom use. I would be more enthusiastic about the program if the pace of its computer phase could be quickened by narrowing the program's equivalent of the Atlantic Ocean.

W. Hartley Clark received his Ph.D. in international relations from New York University and is the Frank B. Kellogg Professor of International Relations at Carleton College, Northfield, Minnesota. Acknowl-

edged in Who's Who in America for his extensive research and publications, Dr. Clark has recently authored a series of computer simulations in diplomacy for college-level use.

Cataloging Information

Dewey Decimal Number: 910

ERIC Descriptor: Geography

Subject Heading (Sears):

1. Geography—Computer programs
2. Simulation—Computer programs

Source Information

Program	<i>Search Series: Geography Search</i>
Source	McGraw-Hill Book Company School Division Princeton Road Highstown, NJ 08520
Equipment	48K Apple II or Apple II Plus Disk Drive Video Monitor or Television with RF Modulator
Package	One Diskette with Teacher's Manual and 20 Search Books.
Suggested Retail Price	\$180.00 Available from Sales Representative

MicroSystem80: Critical Reading

Borg-Warner Educational Systems

Ann McCormick Piestrup

An impeccable QWERTY.

Critical Reading performs superbly in every detail. It is easy to use for both the teacher and the student, and appears to be extremely reliable. This major publishing-house effort is a supplemental reading-comprehension program that introduces basic rules of logic and provides practice in reasoning skills. Though it is executed in accordance with accepted educational practice, it is a computerized text curriculum . . . a QWERTY. Read on!

Summary

Critical Reading is a computer-assisted reading program designed to introduce the basic rules of logic and to provide practice in reasoning skills. It is a supplemental reading-comprehension program for middle grade and older students, including adults. *Critical Reading* is a complete package, superbly executed from the clear and complete Teacher's Guide to the hierarchically sequenced lessons and simple management system. The author, Dr. Joseph M. Scandura, is one of the nation's leading authorities on structural learning. Borg-Warner provides a complete service for schools along with this software package, including demonstrations, in-service training, and a software rental option.

Critical Reading is a black-and-white text curriculum adapted for use on a microcomputer. As such, it does not use the full capability of the Apple, nor is it a product designed specifically for use with a computer. The program has been produced with an attention to detail that marks it as a completely professional product.

The lessons feature:

- skill areas necessary for learners—ability to read logically is clearly important and will be increasingly important in the computer age; a math teacher aware of students' needs for logical thinking recommended this program to the language-arts staff.
- making valid inferences, identifying incompatible statements, and recognizing illogical statements in:
 - rule for “or” elimination
 - rule for “all” elimination
 - conditional statements
 - inductive reasoning (also includes determining a missing premise, drawing a conclusion, and identifying statements that do not follow a logical pattern).

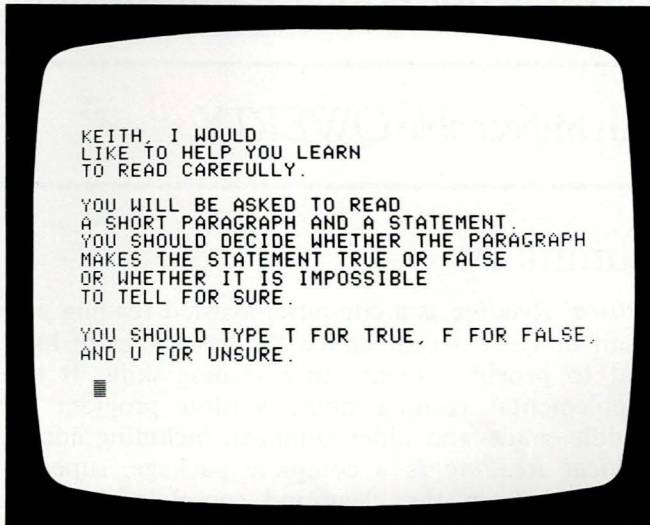


Figure 1

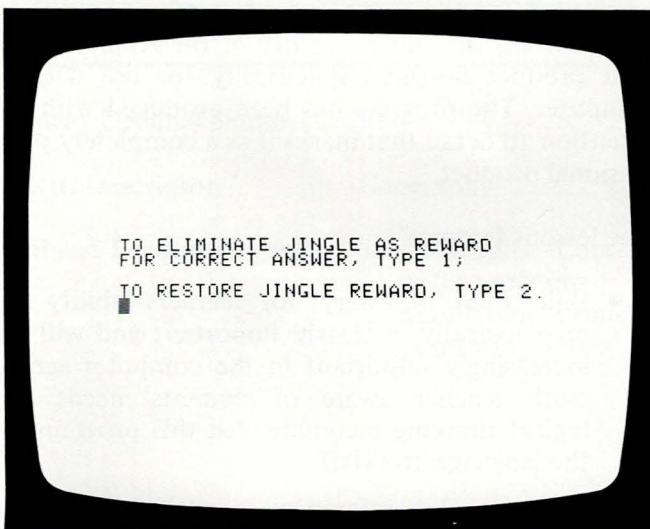


Figure 2

- arrangement in a sequence that allows children to experience success throughout the experience—there is a gradual progression of skills with preparation for each new step (Fig. 1).
- computer-managed and assisted instruction—Apple II presents and scores pretest, branches to appropriate lesson, presents paragraphs with multiple-choice questions, presents progress checks and posttest, transfers scores to individual and class-record lists with no teacher intervention required.
- optional sound—music can be on or off, so class is not disturbed (Fig. 2); some sound feedback remains, which can alert the teacher if a student is having difficulty (teachers have reported this to be a valuable feature in Borg-Warner Educational Systems MicroSystem80 programs; the tones for a series of wrong answers can catch the attention of the teacher, who can diagnose the problem).
- vocabulary level set at third-grade level so that the reasoning skills can be learned by a wide range of students.
- contents suitable for a wide range of users—topics include house color, birds, school classes, dolls, clothing, the zoo, rectangles, clubs, fishing, park rides, detectives, ball games; tested out of school, it held the attention of some junior-high students (but not all!) over Space Eggs and Adventure; the real holding power is the challenge to read logically.
- content without sexism and racism—paragraphs refer to women who are doctors and have expensive cars, a boy who plays with dolls; Maria and Miguel play with Fran and Dick.
- internal consistency—the same keys have the same function throughout all lessons; the same style and approach is used throughout.
- readable quantity of text on the screen for each frame (one paragraph is easier to cope with than pages of text or a pile of dreary cards to flip through); this may be a major advantage of the computer medium for this curriculum—the student sees a manageable bit of work on the screen and no more (Fig. 3); I think students will complete the tasks more readily with the computer's immediate feedback, and spend less time chewing their pencils and watching the clock; they might get even more engaged by other programs, however.
- good timing—lessons load quickly and all at once; text is displayed very quickly; sounds and text appearance do not break the pace (in some computer programs the computer "has all

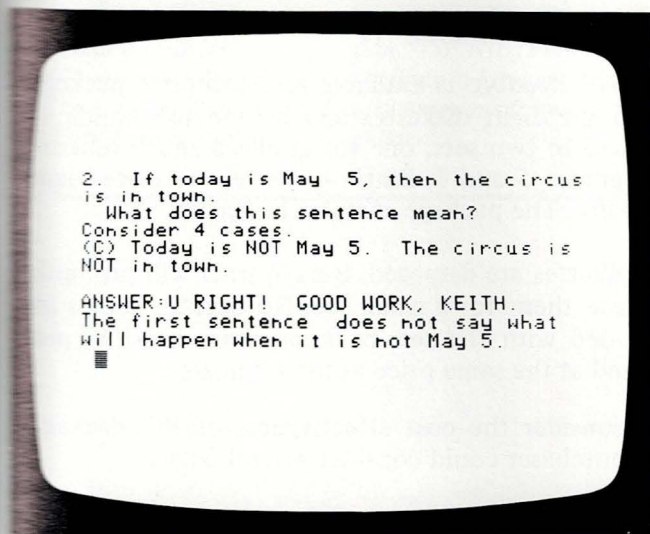


Figure 3

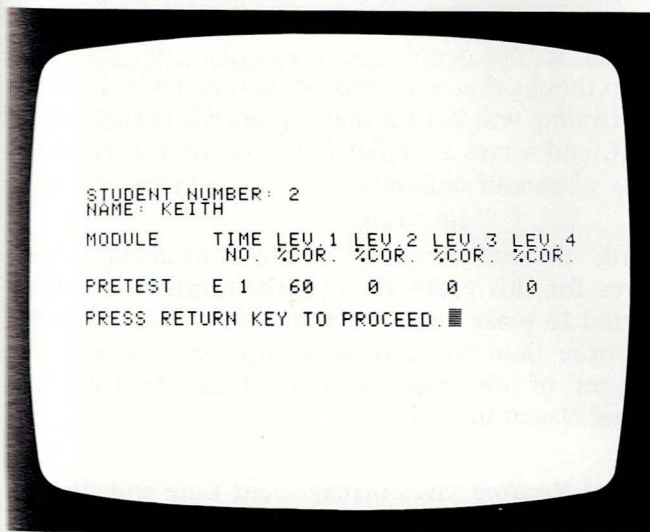


Figure 4

the fun" doing tricks with sound and color while the learner waits without any interaction; in *Critical Reading*, active learning is maximized).

- tight programming—this program runs so smoothly that great care is evident in the programming; memory must have been handled very carefully to allow the management system to occupy the same diskettes as the lessons; the program gives an error code to identify hardware problems (and there was an expert available who solved my problem when I called the hotline); you can count on this program not to crash, or fail, easily.
- a linear sequence—this is not a creative, open experience with lots of choices for the learner; there are no graphics; there is no color; sounds are utilitarian, yet it is a thorough curriculum.

All these features could be delivered with teletypes and computers in use years ago. For that matter, the course could be taught with task cards and manual test scoring. For this reason, and because the program represents a linear curriculum, I call *Critical Reading* an "impeccable QWERTY".

Seymore Papert in his recent book *Mindstorms* (New York: Basic Books, 1980), describes the QWERTY phenomenon. Early typewriters had their keys arranged to avoid jamming; and when the jamming problem was solved, too many typists were used to the arrangement (with the letters q-w-e-r-t-y in the upper row) to make a change. Similarly, it took motion-picture makers years to realize the unique features of the film medium; at first they filmed plays—a QWERTY phenomenon. *Critical Reading* is a computerized text curriculum, a QWERTY, even though it is executed in accordance with accepted educational practice.

I hope that publishers with resources and expertise like Borg-Warner's will consider delivering products truly designed for microcomputer-era learning, and that educators will demand such products. Many of us have a long-term investment in an approach to learning exemplified in *Critical Reading*, but this approach may not fit the computer age. My question is whether teaching *any curriculum* on a computer is appropriate, not whether *Critical Reading* is well designed within its genre. Papert proposes that even very young students use LOGO, a programming language that offers direct experience of the power of a computer to generate dynamic color graphics and thus do logical reading and writing (programming). Students debug their own work within an exploratory environment that they control.

A curriculum based on multiple-choice questions must have right and wrong answers. I object to having a machine emphatically state, "WRONG! THINK!" to elementary students. This wording in the context of a success-oriented program comes across like a tough, well-respected teacher; but a machine lacks the qualities of the caring glance that can go with such admonitions. Precisely because it is anonymous, the computer allows the child to imagine scathing or reassuring glances to go with words like WRONG! This is a key problem that accompanies linear curricula on computers. The computer expects simple answers and judges the student every step of the way. I think that the "right/wrong" model needs reexamination for computer-based learning. As a teacher, I don't want a computer to talk to my students in a way I try to avoid.

The Management System features:

- integration with the lessons so the Apple II indicates progress on lessons as well as tests
- a system for rerouting students within the material with no manual recording of scores
- directions that can be mastered immediately from the screen instructions
- easy access by the teacher for reviewing scores (Fig. 4); the teacher presses a three-key command sequence (holding down three keys won't easily happen accidentally, so students are unlikely to access class records).

The Teacher's Manual features:

- clear instructions for setting up the Apple II microcomputer—from opening the box, plugging in cords, loading diskettes, working through the lessons, reviewing student posttest scores—all in one manual
- all steps illustrated with clear photographs
- format with open space, steps, and cautions highlighted with dark print in the margin; pages broken with photos, bulleted points, shaded charts
- no reference to bits, bytes, boots, bugs, or other computer jargon; the discussion of hierarchical skills sequences is probably jargon, but a type familiar to many educators
- behavioral objectives listed, and the learning sequence explained in detail
- thorough, concise coverage with no redundancy
- separate support materials include: (1) manual describing conceptual development and evalua-

tion of materials for *Critical Reading*, references, author biography, and behavioral objectives; and (2) pamphlet suitable for parents. No student manual is needed—all directions are on the screen, and the procedure is very simple and consistent.

Cost Reduction

Critical Reading is available as a complete package, including eight diskettes and a Teacher's Guide; or divided in two sets, one for grades 3 and 4, and one for grades 5 and 6. Single diskettes are priced individually. The package can also be leased.

If diskettes are damaged, Borg-Warner will promptly replace them for a small fee. No backup copies are supplied with the set; backup copies must be purchased at the same price as the originals.

To consider the cost effectiveness of this package, the purchaser could consider several issues.

First, it is extremely expensive for a major publisher to provide a product with the high quality of *Critical Reading*. This is not a cottage-industry product that may not be supported next year. It is not a game that will lose its interest in a few hours, nor is it a repetitive math drill. It is a complete, logical reading curriculum with a management system. If schools want curriculum products of this quality, backed by a stable company, the cost will be high compared to most computer-store software.

Second, school people spend a similar price for a set of textbooks that will wear out and be revised. *Critical Reading* will last for many years without replacement, and serves a similar function when used with many classes of children.

Third, *Critical Reading* is non-consumable; test scores for this year's class can be printed out, then deleted to make room for next year's class. Records for more than one class or group can be stored on one set of diskettes, so the set can be used by several classes in a school.

Critical Reading saves management time and, therefore, money; the computer performs functions that could be carried out by a teacher aide.

On the other hand, schools might want to consider whether it is optimal to use computer time, which is still fairly scarce in most schools, to present a black-and-white text curriculum on a machine having more dynamic capabilities. Children might be able to learn the critical reading skills taught in this

Ann McCormick Piestrup, Ph.D., studied educational psychology at the University of California, Berkeley, focusing on children's thought and language. She has taught preschool through graduate-level courses, consulted nationwide, and is founder and Chairman of the Board of The Learning Company (TLC) in Portola Valley,

California. Dr. Piestrup was awarded a grant from the Apple Education Foundation for the development of a primary perceptual-skills program and has since directed a National Science Foundation/National Institute of Education (NSF/NIE) project resulting in logic and geometry programs for young children.

package as they learned to program the computer, perhaps using LOGO.

As indicated at the outset, *Critical Reading* performs superbly in every detail. It is easy to use, well managed for the teacher and the student, and appears to be extremely reliable. The program is completely supported by its source, a creditable standard in the industry.

Cataloging Information

Dewey Decimal Number: 372.4
 ERIC Descriptor: Reading—Computer programs
 Subject Heading (Sears): Reading

Source Information

Program	<i>MicroSystem80: Critical Reading</i>
Source	Borg-Warner Educational Systems 600 West University Drive Arlington Heights, IL 60004
Equipment	48K Apple II Plus Disk Drive Video Monitor or Television with RF Modulator
Package	Eight Diskettes with Teacher's Guide (CR-AH)
Suggested Retail Price	\$750.00 Series CR A-H \$350.00 Series CR A-D \$400.00 Series CR E-H Available from Source

Comprehension Power Program

Milliken Publishing Company

Glenn M. Kleiman

*Individualized comprehension skills,
directed toward reading requirements
in grades 4 through 12.*

A major publishing house introduces reading courseware that integrates well with standard reading comprehension curricula. This program aptly demonstrates the use of quality graphic character sets that are clear, attractive, and easy to use.

The Milliken *Comprehension Power Program* is designed to develop 25 comprehension skills, such as finding main ideas, following sequences, making inferences, visualizing, comparing and contrasting, detecting author's purpose, and identifying mood and tone. The stated aims of this package converge well with standard reading comprehension curricula. The full *Comprehension Power* package covers reading levels 4 through 12. All the levels follow the same general format: only the reading materials and the comprehension skills covered change. For each level there are four diskettes, each containing three lessons and a record manager program. Each lesson is based on one reading passage and 20 comprehension questions. There is also a 28-page teacher manual, two duplicating masters for optional record-keeping forms, and a card with instructions for the students.

For review, I examined Level F, which is designed for children at the grade 5 reading level. Level F provides many questions requiring recall of information and details, making inferences, recognizing cause and effect, and reasoning. Other skills are tapped by fewer questions; and some, such as identifying the speaker, visualizing, judging validity, and understanding persuasion, are covered by only one question, or none at all. The range of skills tapped seems appropriate for grade 5. I assume the other skills are covered more extensively in other levels.

The program is perhaps best described by following the experience of a student using it. There are several options that can either be pre-set by the teacher (using the class-manager program described below) or left for the student's choice. The following example assumes that as much as possible has been left to the student's choice. The student types his name, class, and password. He is then presented with a choice of the available stories. The first part of the lesson is vocabulary study. Sentences in which a vocabulary word is underlined appear one at a time, and remain on the screen until the student presses

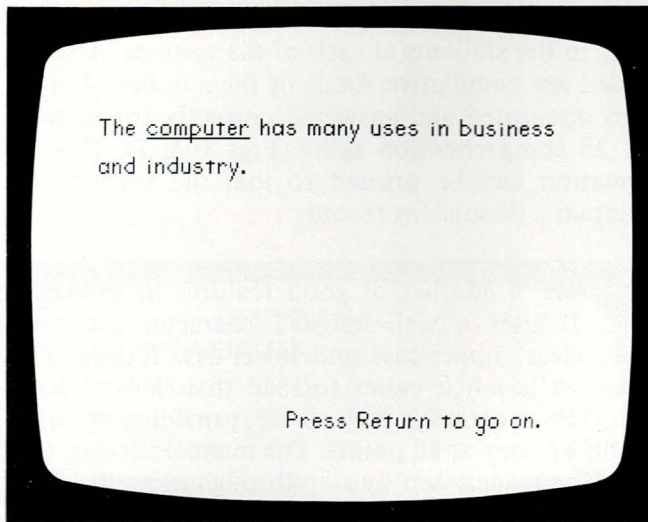


Figure 1

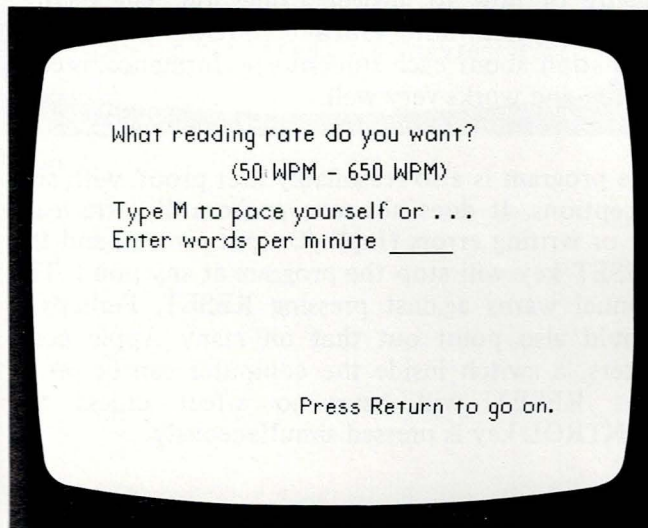


Figure 2

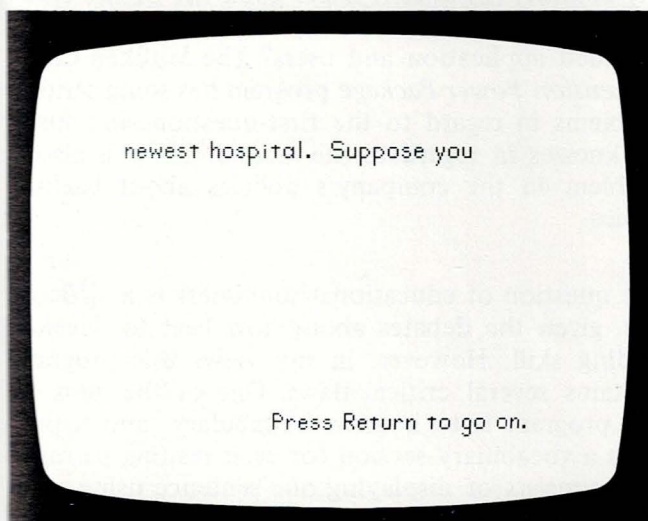


Figure 3

RETURN (Fig. 1). Next the student "previews" the passage. A selected set of sentences is displayed one at a time. Again, the student controls how long each sentence remains on the screen. Previewing is followed by one multiple choice and five true-false questions. Immediate right/wrong feedback is given for each answer. Next, a brief summary of the story appears. The student is then asked to select a reading rate, between 50 and 650 words per minute, or he can choose to pace himself (Fig. 2). Finally, the story itself is presented. The presentation shows one line of about 30 letters at a time (Fig. 3). In a revised version of the package, the student can select a self-paced option in which six to eight lines of text are shown in each display. If a reading rate has been chosen, the lines flash on and off at the set speed. If the self-paced option is chosen, the student presses RETURN to see the next line.

Each passage is divided into ten segments. One to three multiple choice questions are presented after each segment (Fig. 4). When the question is given, the student has the option of rereading the segment before answering. The student gets two chances for each question, and is told the correct answer if he or she is incorrect on both tries. After answering the questions for a segment, the student can end the session or go on to the next segment. The rate of presentation can be changed between segments. If the student stops in the middle of a passage, he will automatically be started there on his next session. At the end of a session, detailed feedback is given. The reading rate, comprehension score, reading index (rate divided by score), number of times the segments were reread, and the number of questions attempted and answered correctly for each comprehension skill are shown. All this information is saved on the diskette so the teacher can access it later.

Each diskette also contains a classroom-manager program. The documentation says this can hold records for up to 100 students in five classes. I did not test these limits. To set up the class manager, the teacher types a special code at the beginning of the program. The teacher then enters his name, and a password. This password is necessary to read or modify the class records. The teacher then types each student's name, which passages are to be assigned, whether or not previewing should be given, and a reading rate. The previewing and reading-rate options can be left to the student. If the reading-rate option is left to the student, he or she can choose self-pacing. The student-pacing option cannot be specified by the teacher.

The class-manager program also allows the teacher to add or delete students, change assignments, and review student progress later (Fig. 5). The records

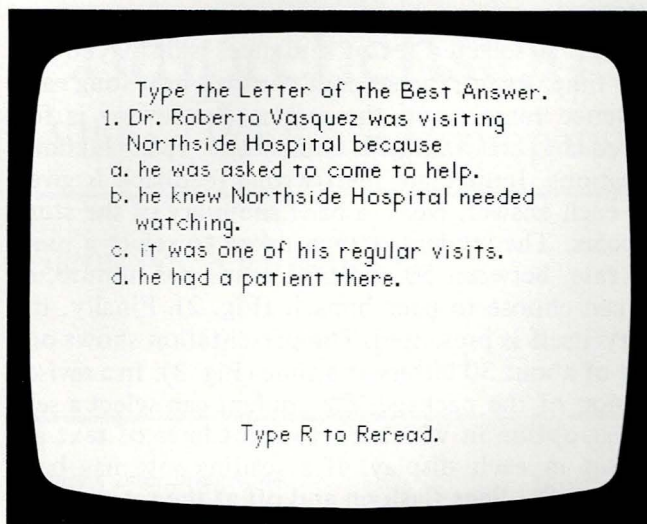


Figure 4

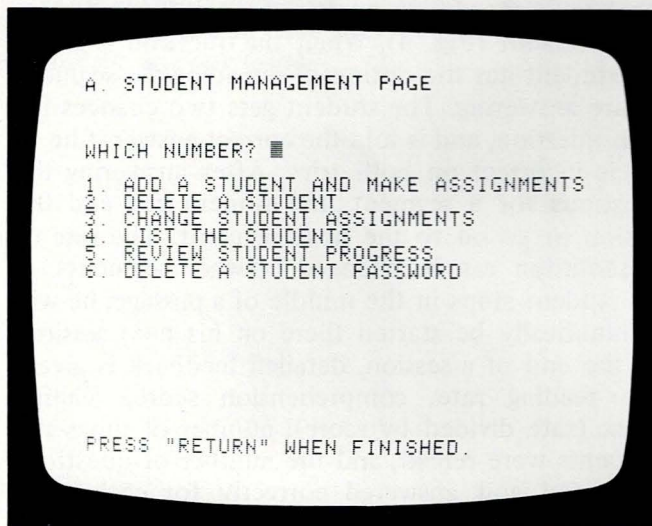


Figure 5

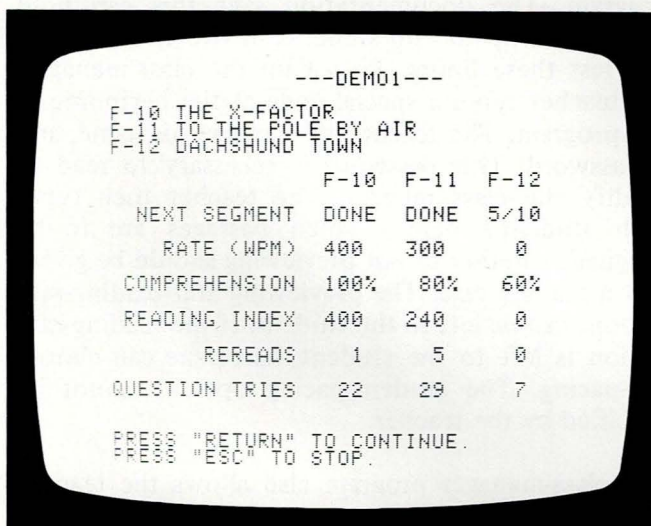


Figure 6

kept are very detailed, including all the information given to the students at each of the sessions. Also recorded are cumulative totals of the number of questions attempted and answered correctly for each of the 25 comprehension skills (Figs. 6 & 7). This information can be printed so that the teacher can maintain a permanent record.

There are a number of good features in this software. It uses a well-designed character set, with large, clear, upper-case and lower-case letters. The letter set is much easier to read than most I have seen. The program is easy to use, providing prompts of the options at all points. The manual is clear, and takes the reader step by step through using the class manager. The program has some flexibility, letting the teacher or student set whether previewing will occur and the rate of presentation. The student also has the option of rereading a segment when he is unsure of how to answer a question. The record-keeping system automatically provides detailed information about each student's performance, is easy to use, and works very well.

The program is also reasonably user proof, with two exceptions. It does not recover from diskette reading or writing errors (Input/Output errors), and the RESET key will stop the program at any point. The manual warns against pressing RESET. Perhaps it should also point out that on many Apple computers, a switch inside the computer can be set so that RESET will have no effect unless the CONTROL key is pressed simultaneously.

Given the number of good features, it is unfortunate that I am unable to recommend the Milliken *Comprehension Power Program*. I consider three main questions in evaluating software: (1) is it educationally sound?; (2) does it make good use of the computer's capabilities?; and (3) is it suitable for the intended application and users? The Milliken *Comprehension Power Package* program has some serious problems in regard to the first question, and some weaknesses in regard to the second. There is also a problem in the company's policies about backup copies.

The question of educational soundness is a difficult one, given the debates about how best to develop reading skill. However, in my view, this program contains several critical flaws. One of the aims of this program is to improve vocabulary, and it presents a vocabulary section for each reading passage. This consists of displaying one sentence using each vocabulary word, with the word underlined. This is, at best, a very minimal introduction to new words.

```

---DEM01---
(<2>) INTERPRETATION
  SKILL      F-10  F-11  F-12  TOTAL
1.MAIN IDEA  1/1   1/1   0/0   02/02
2.INFERENCES 3/3   0/0   0/0   03/03
3.OUTCOMES   0/0   0/0   0/0   00/00
4.CONCLUSIONS 3/3   2/2   0/1   05/06*
5.FIGURATIVE 1/1   0/0   0/0   01/01
6.VISUALIZING 0/0   0/0   0/0   00/00
7.PARAPHRASING 0/0   0/0   0/0   00/00

PRESS "RETURN" TO CONTINUE
PRESS "THE BACK ARROW" FOR PREVIOUS PAGE.
PRESS "ESC" TO STOP.

```

Figure 7

The *Comprehension Power Package* also claims to develop the ability to preview effectively. Previewing is an important reading skill. A good reader can rapidly scan text to pick out main points and get an overview of how the material is organized. The key to previewing skill is learning to be selective and to continue searching for information until a sufficient impression of the material is formed. Presenting sentences to read, as this program provides, does not seem likely to develop previewing skill. Since the sentences are presented one at a time, and the students never see the entire passage, they have no way of knowing the context of what was selected, or why certain sentences were chosen.

This package also claims to “develop the ability to read more rapidly while reading comprehensively”. This is done by setting reading rates. The rate of reading is controlled by presenting one line at a time on the screen for a fixed period. The student must keep up, and cannot reread until the questions appear. There are several problems with this approach. One very severe problem could be remedied. Good readers process text in meaningful units. A common reading problem at the levels addressed by this software is “word-by-word” reading—failure to group words into meaningful phrases. Word-by-word reading is slow; reading meaningful phrases is faster. Unfortunately, the lines of text presented on the screen are not divided into meaningful phrases (Fig. 3). The flashing off of one line and on of the next is disruptive to grouping words together. Presenting meaningful phrase units might encourage faster reading; presenting random units makes reading difficult. I regard this problem to be of sufficient severity that I cannot recommend any computerized reading lesson that presents the text in this way, no matter how good the rest of the package.

Even if meaningful phrase units were presented, the reading situation would still be unnatural. Good readers monitor what they read as they read it. They slow down at important parts of the text, reread words, phrases or sentences of which they are unsure, skim parts they already know, and so on. That is, they actively control how they process the text.

This active control is fundamental to good reading. Presenting a few words at a time does not allow it, and might discourage its development. This is especially true when the reader does not even control how long he can look at each group of words.

My second question in evaluating software is: Does the software take advantage of the computer’s capabilities? This program does in some ways. It automatically keeps detailed records, and it provides immediate feedback for the students. However,

in some ways it does not use the computer well. The vocabulary lessons as presented would be better done on paper, rather than taking up valuable computer time. As noted above, the attempt to use the computer to develop previewing and reading speed skills seems to be misguided.

I also have concerns about the flexibility of the overall package and the company's policies in regard to backup diskettes. The 36 diskettes in the entire package are all basically the same. The only changes are in the content of the passages and the questions. The materials seem to be carefully selected (judging from the sample I have seen). However, many teachers might prefer other passages, more questions, or additional passages at certain levels. An ideal program would allow teachers to enter their own passages and questions in addition to those provided. It would also allow a single class-manager diskette to accumulate student records from many passages, not just the three on the same diskette.

The diskettes are protected from copying. I understand that software producers must protect their investments and that some schools have violated copyrights on programs. I do not believe that schools can expect special consideration. However, Milliken's policies are not adequate for any serious software, and would not be tolerated in the business community. They will replace a damaged diskette for a fee, if returned within 12 months of purchase.

After that, the replacement fee is higher per diskette, if the diskette is "then in stock". The most critical problem is that no backup copies are provided. If a diskette is damaged, the class cannot use those lessons until the diskette is sent to Milliken and replaced. This can seriously disrupt a teacher's plans. The class records are stored on the same diskette as the reading passages. Therefore, each diskette is used a great deal, and the likelihood of accidental damage is high. I believe the appropriate policy, now followed by many software companies, is to provide a backup of each diskette. If one copy is damaged, the backup copy can be used while waiting for the replacement (which is made available at a small charge). This policy makes it feasible to use copy-protected diskettes, without constantly risking the disruption of lesson plans by a damaged diskette.

To summarize, I cannot recommend this package. In my opinion, the programs are not educationally sound in several critical ways. Some aspects of the program do not make good use of the computer's capabilities. I also find the company's policies about backup copies to be unacceptable. Despite these flaws, *a number of things were done right*. The record manager is effective and easy to use, the letter set is excellent, and the program has some flexibility and is easy for students to use. I hope a future version will build upon these strengths and correct the weaknesses.

Dr. Glenn M. Kleiman received his graduate degree in psychology from Stanford University in 1977. Prior to becoming a partner in *Teaching Tools: Micro-computer Services*, a Palo Alto-based software consulting and development firm, Kleiman completed exten-

sive research at the Center for the Study of Reading at the University of Illinois, Champaign/Urbana campus. Dr. Kleiman has also been a faculty member at the University of Illinois, the University of Toronto, and the Ontario Institute for Studies in Education.

Cataloging Information

Dewey Decimal Number: 372.4
 ERIC Descriptor: Reading
 Subject Heading (Sears): Reading—Computer programs

Source Information

Program	<i>Comprehension Power Program</i>
Source	Milliken Publishing Company Instructional Communications Technology, Inc. 1100 Research Company St. Louis, MO 63132
Equipment	48K Apple II or Apple II Plus Applesoft ROM Card Disk Drive, Video Monitor or Television with RF Modulator
Package	Four Diskettes per Level, with Teacher's Guide. Level D-E-F (12 Diskettes) Level D-H-I (12 Diskettes) Level J-K-L (12 Diskettes)
Suggested Retail Price	\$150 per Level \$425 for Three-level Combination Set. Available from Local Dealer.

Compu-Spell

Edu-Ware Services, Inc.

David McCarthy

An electronically assisted spelling bee.

Compu-Spell is an instructional course that uses perceptual principles and positive reinforcement to teach spelling. An infinitely patient silent partner for the user, this program provides spelling practice in a variety of textual formats.

Compu-Spell, a technically sophisticated, computer-managed sequence of spelling drills, contains thousands of appropriate words for students in grades 4–8 and those at a “secretarial” level. This system of spelling practice and memorization model are evaluated from three points of view: the teacher, the student, and from the standpoint of educational psychology.

Teacher

For the teacher, the *Compu-Spell* program contains a comprehensive procedure to enter data for up to 60 students and track their individual progress. For each student-user the system maintains a record of location and progress within the system, and the eleven most recently misspelled words.

The *Compu-Spell* system contains extensive provisions for modifying its contents. However, the user documentation for doing so is relatively poor. It is not designed to fit the needs of a teacher who has had little or no training in the computer field. The directions and procedures for working through the programs are loaded with confusing “computerese” terminology and are not clearly structured. For example, the documentation makes use rather freely of words like “booting,” “initialization,” and “routines.” Such terminology is not, as yet, part of the working vocabulary of most teachers.

The system options are not actually changed on the diskette until the system option program is exited. Hence it is vital for users to exit the system using the appropriate menu selections and not by pressing RESET, etc. If, accidentally or otherwise, this happens, the system options should be rechecked before proceeding further.

Some attempt has been made to direct the user through the system documentation by providing guidelines that give directions on proceeding through the various menus, sub-menus, and sub-sub-menus. For example, selecting the choice “view/change general options” from the “main menu” takes the user to a sub-menu that lists six other

Dr. David McCarthy completed his graduate work in education at the University of California and is now a member of the faculty at Lower Columbia College in Longview, Washington. McCarthy spent the 1980–81 academic year on sabbatical study at the Computer and Information Departments at both the University of Illinois and the University of Oregon. His comparative research involved the relationships and use of computers in language-arts instruction.

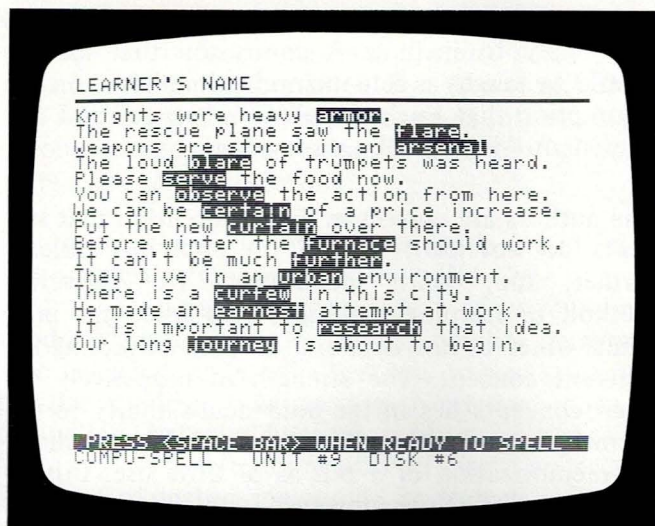


Figure 1

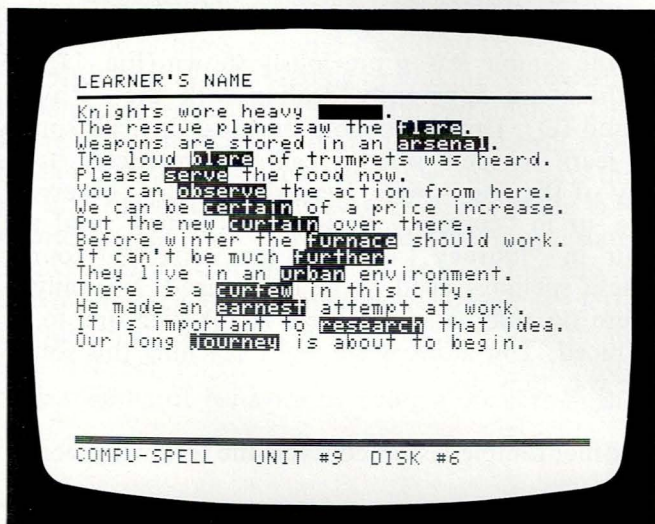


Figure 2

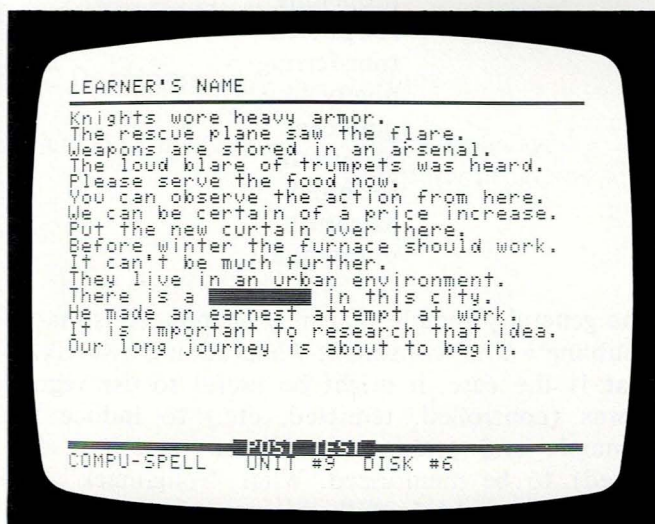


Figure 3

choices: drives, printer, password, mode, posttest display, passing score. By the time I got to the third level, I was uncertain just where I was in the overall design. Some kind of structural map needs to be provided, in addition to simplifying the turgid, technical directions. The designers of this set of instructions might consider the introductory guides to Applesoft BASIC or TRS-80 BASIC as guidelines. Computers-in-education is a complex novelty for nearly every teacher; software designers have got to simplify their directions.

Student

As a student, I am presented with a "page" containing a series of sentences (Fig. 1). The box around a word represents "inverse" (black-on-white) printing. When I press the space bar, the word disappears and the inverse block remains. My task is to type the first word, "armor", in the blank block (Fig. 2). As I type each letter, it automatically appears in the space provided. When I have correctly spelled the word, the inverse representation disappears and I proceed to the next word. If I enter any letter that is not part of the correct spelling of the word, the letter will not print. If I do make an error in typing "armor", it remains in inverse form on the screen, and I will have to type it correctly again before I can exit this lesson—whether my error was in typing, spelling, or because I forgot "armor" and substituted "helmets". In the sentence, "There is a _____ in this town", literally thousands of nouns could fit into this blank, but I'd better remember "curfew" (Fig. 3). That's what I am learning to "spell" (?). Words that are missed will remain until they are spelled correctly. I can, however, press the ESCape key and have the word presented.

After working my way through the complete list of words, I proceed to the posttest. Here the program prints one sentence at a time, with the spelling word deleted. The sentence "Knights wear heavy _____", appears. The program will accept anything I type into the blank, and then replace it with the correct word. Hence, if I type "hats" or "shoes" or "armour", all of which are correct in both logic and spelling, I have missed it. By the time I get near the end of the drill, I am quite likely to have forgotten that there is a "curfew" in this town. There is a river, a park, a fountain, a school, a university. There isn't a curfew; but, once again, I'd better remember what the sentence requires. I consider myself a fair speller and a relatively competent typist, but I was not able to progress through any of the four sample drills and posttests I tried without making errors in typing, cognate substitutions, or logical substitutions. Finally, since I've made errors, I must repeat the entire posttest. The program

does offer the option "quit for now?" And I'll confess, I took it.

Educational Psychology

Finally, it is constructive to examine *Compu-Spell* from the standpoint of educational psychology. A system may be sound technically; it may be easy to operate; it may be interesting to students. However, if it does not present a subject in a way generally congruent with current learning theory, all of its other amenities are ineffectual.

The designers of *Compu-Spell* say that it is based on an "instructional model." This model is at variance with some current concepts of both how we learn to spell and how spelling should be taught. Their first contention is that "spelling is memorization" and that, although many teachers teach it as a rule-generating strategy, most learners do not follow such a strategy in learning to spell. Rather, they contend, the learner uses only *one* method in learning to spell: rote memorization. It seems apparent that the authors are dealing entirely with visual, and perhaps kinesthetic memory, since their program does not allow for either speech or audition. They also claim that the "vagaries and ambiguities" of English necessitate the memorization procedure.

The first problem with this contention is that spelling, like human learning in general, is acquired through the senses. It is multi-sensory, multi-motor, using vision, speech, touch, and audition. Thus, the child develops the ability to speak and write a language after hearing it. S/he speaks by selecting appropriate phonemes (sounds) for the construction of words and applies a motor response for pronunciation. The act of writing and spelling is basically one of encoding phonemes into graphemes. The authors of *Compu-Spell* overlook this basic fact.

The second error in the authors' first contention is that, since the English language is full of "vagaries, ambiguities, and inconsistencies," it would, indeed, be counterproductive to ask the learner to develop such structures.

However, a number of studies have corroborated the relative regularity of the grapheme-phoneme correlations in American English. Roughly 80 percent of the phonemes in words comprising the traditional spelling vocabulary of elementary-school children approximate the alphabetic principle in their letter representations. About 75 percent of vowel phonemes do not represent significant spelling problems. Eighty-two percent of consonant clusters have only one spelling. These facts would suggest two things: it is valuable to teach sound-letter concepts

when teaching spelling, and there are some rules that apply to spelling. A conclusion that spelling should be taught as rote memorization, based on the assumption that English spelling is "vague" and "inconsistent," is an invalid assumption.

The authors are correct in their contention that students do not *learn* to spell by applying "rules". Rather, they learn inductively. The inductive method of learning spelling begins when the individual observes raw data and ends by extracting the inherent concept. The strength of inductively derived concepts lies in the individual's ability to use them in an environmental interaction. The mechanical memorization of a rule is of little use. But inducing a principle through examining specific instances is not the same thing as "rote memorization". I think the authors are confounding the two principles. A brief perusal of a sample lesson or two will suggest the process that is really occurring.

In the sample lesson previously shown (Fig. 1), the authors are apparently dealing with the "schwa" sound (ə). This is probably the most difficult sound to learn to spell, since it can be constructed using any of the five vowels. Hence, the -er in "observe," the -ur in "curtain," the -ear in "earned" and the -our in "journey" all have the same (ə) sound. These spellings need to be memorized. Not only is there no rule to apply, there is not even one to be induced. The authors are right teaching this sound the way they do.

Another sample lesson contains the following words:

controlled
equipping
crestfallen
remitted
propelling
roughneck
transferring
ninety
nineteen
benefited
forfeited
hairbrush
interference

The general principle that may be induced is that of doubling a final consonant when adding a suffix. If that is the case, it might be useful to use regular words (controlled, remitted, etc.) to induce the principle and present exceptions (benefited, forfeited) to be memorized. What "roughneck" and "ninety" have to do with this procedure is difficult to ascertain.

We can conclude from this, then, that our authors are correct in assuming that spelling cannot be learned by memorizing rules. But their conclusion that it must, therefore, be a process of rote memorization, instead of the inductive discovery of concepts, is not an accurate one.

The second and third premises upon which the authors design this program are valid ones: words need to be written in the context of surrounding words; learning behavior should parallel performance behavior, so that the best way to have students practice spelling is by writing. Next they make a distinction between register memory, short-term memory, and long-term memory of taught materials. The goal of instruction should be to have information remain in the long-term memory. True. But the question is "what" should remain in the long-term memory: how to spell the word "curfew", or that "curfew" belongs in the blank?

Lastly, the authors contend that they "precisely accomplish" behavioral shaping by making it impossible to be presented with incorrect reinforcements. They are referring to the fact that, during the drill phase of their lesson, the blank will show only letters that are the correct ones. However, response made is response learned, whether this response is translated into visual representation or not. It would be easy to imagine the student typing (and reinforcing) a letter several times, waiting for the machine to respond. The authors might more "precisely accomplish" their shaping of behavior by being more certain that

errors made by students were fewer, and the results of misconceptions of concepts being taught, rather than misunderstandings of words being sought.

Compu-Spell represents a classical example of the frequent mismatch between technology and pedagogy. Technology says, "Since we can do things, using slide tapes or videodiscs or disappearing words, let's use them to teach". The process should be reversed. Education should say, "This is what we know about how to teach spelling, math, reading, etc. How can technology facilitate the process?"

Compu-Spell is of some value. With proper documentation, probably many of the lessons could be used to teach particular types of words, such as words that are basically learned visually, to certain students who are having trouble with those words. It might be used as a drill to teach irregular words, also. It would, however, need to be integrated selectively into an entire language-arts curriculum that would include saying a word, examining it for graphemic correlations or irregularities, writing the word, checking its accuracy, identifying errors, and restudying difficult words. Its record-management system, once fathomed, may be of little use in such a case. Compared with complex language-arts subjects like reading comprehension or writing, spelling is a relatively simple learning process to analyze and teach. It would seem that computers could facilitate greatly in this domain of learning. Hopefully, a future version of this program will prove this to be true.

Cataloging Information

Dewey Decimal Number: 372.6
 ERIC Descriptor: Spelling
 Subject Heading (Sears): English Language—
 Computer programs

Source Information

Program	<i>Compu-Spell</i>
Source	Edu-Ware Services, Inc. P.O. Box 22222 Agoura, CA 91301
Equipment	32K Apple II or Apple II Plus Disk Drive Video Monitor or Television with RF Modulator
Package	One Diskette and User Guide
Suggested Retail Price	\$29.95 Available from Local Dealer

Sentence Diagramming, Parts of Speech, Usage

Avant-Garde Creations

Stephen Marcus

Remember sentence diagramming? Do you know your declaratives from your imperatives? Your verbs from your interjections?

This language-arts courseware tests usages, parts of speech, types of sentences, sentence diagramming and, in doing so, invites user involvement. The authors, recognizing that all grammarians do not agree, for program simplicity have placed all word classifications in two categories: usages and parts of speech. The result is an English basics program of variety and challenge, which, as the reviewer points out, evokes a certain nostalgia in trying to solve some of those diagramming problems.

Introduction

More and more, testing and instruction in the schools are employing videotext (print that appears on TV screens). There are growing indications that videotext is, in fact, a new medium, shaping language itself, language conventions, and the structure of knowledge. Educators in particular should be concerned with the ways in which videotext will be affecting their students' reading and writing habits. Insofar as teachers themselves are creating and selecting courseware, they need to develop their sensibilities as "information artists".¹ They must realize that, in a rather new way, they are in both "know business and show business".² In the discussion below, there is a particular emphasis on the relationships between what the student sees and what he or she "gets". This seems especially appropriate in the case of language-arts courseware.

Description

In an earlier review of this software, McCarthy³ provided a comprehensive and articulate description of the dynamics of the activities. There seems little reason to generate another; and so the description below follows his closely, with all due credit and respect.

Upon entering the program, the student decides if the computer is to record the session. If not, he or she may practice the exercises in any of the three levels of difficulty, there being 20 sentences in each level. If records are to be kept, the system either creates a file for the student or, if one already exists, begins with exercises from the student's last achievement level.

The first exercise involves identifying parts of speech. A sentence is presented; and, as an asterisk

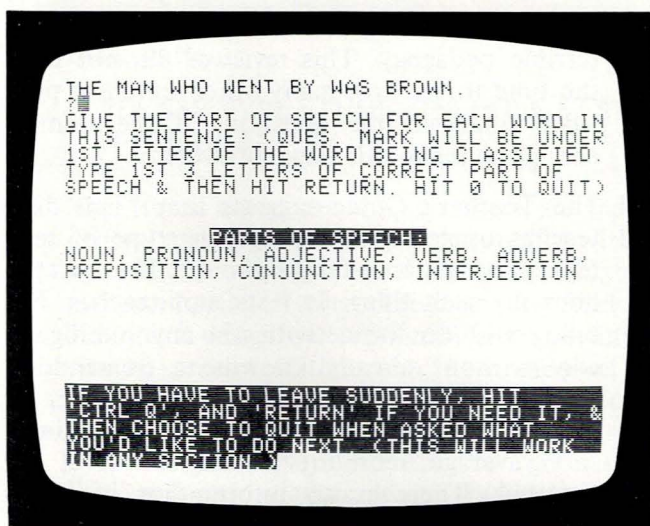


Figure 1

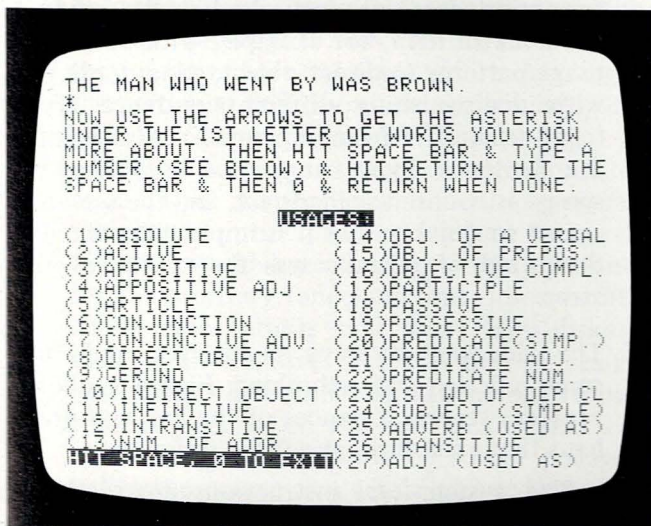


Figure 2

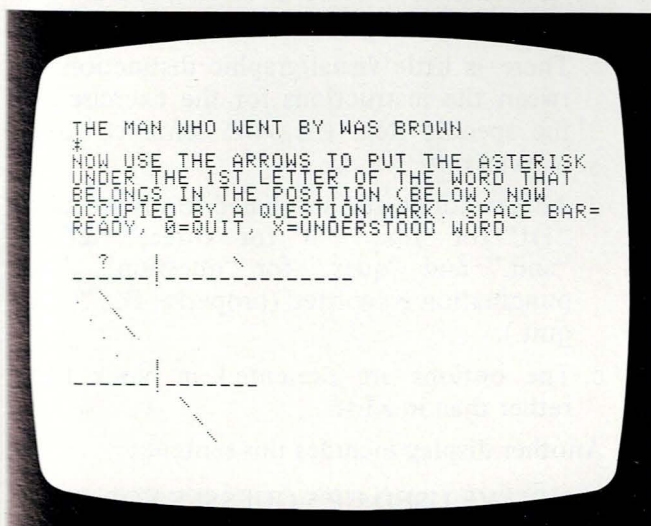


Figure 3

moves from one word to the next, the student identifies the word as either a noun, pronoun, adjective, verb, adverb, preposition, conjunction, or interjection (Fig. 1). Responses are noted as “correct” or “incorrect,” and the second word in the sentence cannot be processed until the first is correctly identified.

Next, the student is presented with the same sentence and a list of 27 usages (Fig. 2). Using the arrow keys, the student moves the asterisk under a word and types the number corresponding to the word’s usage in the sentence. The third activity asks the student to identify the sentence as either declarative, interrogative, exclamatory, or imperative. This is the only activity in which just one answer may be attempted.

The fourth and final activity involves sentence diagramming. The screen displays the sentence again, along with an empty diagramming skeleton (Fig. 3). The student moves the asterisk under the word that is thought to belong in the blank containing the question mark, and presses the spacebar to enter the choice. If the correct answer has been chosen, the computer places the word in the blank and emits a “beep”. The question mark then moves to the next blank space. If the student makes five incorrect responses for a given blank, the correct response appears in the space.

Throughout all this activity—if the student has chosen to have a record kept—the system keeps track of the student’s progress, evaluates it using a point system, and stores the data in tabular form. The teacher can set criterion levels of performance for advancement from one level to the next using two criteria: percent of sentences attempted and percent of correct choices. For example, if the teacher sets both levels at 90%, the student must attempt 18 of the 20 sentences in a level and attain 90% accuracy before being allowed to attempt the next higher level. In succeeding sessions, the student returns to the last level attempted until both criteria are met.

Critique

McCarthy (cited above) concluded his review as follows:

The number of available sentences seems to be the program’s only limitation. It interacts with students in a dynamic, meaningful way. It is sufficiently flexible to allow skipping over or exiting any section which is inappropriate. The formatting and record-keeping components make it a

flexible tool for the secondary or college-level teacher of grammar and composition.

The present reviewer, however, finds several problems with the software and doesn't find the number of sentences to be a particularly important limitation, given the relatively modest scope of the package as a whole. It should be noted at the outset that, overall, this reviewer found both the software and the documentation to be generally sensible and "friendly" insofar as its tone and approach invited user involvement. Many of the criticisms discussed below could be addressed by more careful editing of the material.

Specific criticisms:

1. The first message to appear on the screen (after the logo display) was the following:

IF YOU DIDN'T BOOT THIS DISK BY
TURNING OFF THE COMPUTER & THEN
TURNING IT ON AGAIN, DO IT AGAIN!

This is a particularly unsettling introduction to the program. Aside from the essential confusion the sentence may generate (Do again what you haven't done once?), it seems to be asking the user to do something that doesn't, in fact, work (see below). In addition, as will be discussed at more length below, the use of "&" is an indication of inappropriate videotext design throughout the displays.

2. There are several problems with the documentation. According to the printed Teacher's Guide, "All of Avant-Garde Creations' software is written on DOS 3.2," yet the diskette being reviewed would run only on 3.3. In addition, the user was informed in several places (in the Guide and on the screen) that the diskette could be booted merely by turning the machine off and then on again. This never worked.
3. In the Usage section, the student is to identify the way each word is used in the given sentence. Yet for at least three sentences in Level One, there is no correct response for certain words. The Teacher's Guide notes only one instance where this is the case, rationalizing the omission by referring to an earlier activity in the Parts of Speech section. This is of little or no comfort to the student who may never know how many other instances were marked "incorrect" because the *computer* made the mistake. To ask a student to choose the correct answer from a set of alternatives that does

not provide the correct answer is, in a phrase, terrible pedagogy. This reviewer did not take the time to see how many other sentences provided unanswerable questions. Three seemed enough to make this flaw noteworthy.

4. The Teacher's Guide suggests that "This disk teaches usages, parts of speech, type of sentence, and sentence diagramming". The diskette does no such thing. It tests, not teaches. No one could do the activities in any intelligent way without already knowing a tremendous amount about usage, parts of speech, etc. In addition, the records kept are of a minimal sort: average score at the highest level attempted. There is no information provided about where the student's knowledge was strongest or weakest, i.e., knowledge of diagramming, or of parts of speech, etc. And within each activity, there is no diagnostic information for assessing whether the student needs extra help, for example, with particular usage patterns (indirect object, gerund, etc.) or with distinguishing different sentence types (e.g., declarative from imperative). Also, within the Sentence Type category, a misspelled response is counted as incorrect, and the screen is erased so quickly that it is impossible to tell if that kind of mistake was the source of the error.
5. The videotext displays, themselves, could use some redesign and rethinking. Figure 1, above, provides several instances of the kinds of problems that exist throughout the program.
 - a. The system-level instructions are scattered throughout the display instead of being confined to one area.
 - b. The activity's instructions appear after the test sentence, instead of serving as a context for what is to follow.
 - c. There is little visual/graphic distinction between the instructions for the exercise and the specific item (i.e., the sentence to be worked on).
 - d. Abbreviations and shortened forms are used: "1st" for "first," "3" for "three," "&" for "and," and "ques." for "question". Also, punctuation is omitted (properly: Hit "Ø" to quit.).
 - e. The options are presented in block form rather than in a list.

Another display includes this sentence:

YOU'VE USED UP 5 GUESSES FOR EACH
WORD I
N THIS SENTENCE.

Still another display includes this inconsistent usage pattern:

PLEASE TYPE STUDENTS' 1ST & LAST
NAME HERE.

The items noted above (and there are many others) may seem trivial in isolation; but within the context of instructional methodology, they are examples of what those in the field of videotext design see as significant and undesirable pressures that this new medium is exerting on writing and reading conventions. The limitations of a 40-column, 23-line screen should not excuse inattention to "good English" and good graphic design. Particularly in a language-arts activity, the kinds of inattention noted above contain messages the videotext medium could do without.

6. A broader issue involves the substance of the material presented. The authors indicate that they reviewed several texts before deciding on the particular definitions and diagramming conventions they employed. Thus, while the correct answers in their exercises are consistent or at least comprehensible within generally understood principles, a teacher using this diskette would have to make some allowances for the arbitrary (and sometimes idiosyncratic) choices the authors have made. This diskette, then, is an example of a test that shapes and determines the content of instruction. To use a student's scores on the activities as part of a course grade would not be sound pedagogy unless the course as a whole adopted the specific definitions and conventions employed by the diskette.

Conclusion

The question remains: Is this diskette worth purchasing? This reviewer's answer is: yes. Even in light of the criticisms above, and in spite of the consistent research findings that such drill, practice, and testing do not improve writing skills,⁴ this reviewer felt that the software provided a reasonable and engaging return for the asking price. In comparison with more expensive, extensive, and elaborate software in the area of "English basics," this program provided enough variety and challenge to make it a useful addition to a software library. Care would need to be taken in its incorporation into an instruction and testing program, and the idiosyncracies of the material would need careful monitoring; but on the whole, the software is interesting and useful. Besides, there is a certain nostalgia evoked in trying to solve those diagramming problems.

Notes

1. This formulation is taken from the writings of Robert Johanson, Institute for the Future, Palo Alto, California.
2. This formulation is taken from the work of Aaron Marcus, a consultant in videotext design, Lawrence Berkeley Laboratory, University of California, Berkeley.
3. David McCarthy, *The Computing Teacher*, Vol. 8, No. 7 (Academic Year 1980-81), pp. 40-42.
4. See "Grammar Instruction: What We Know," Anthony R. Petrosky, *English Journal*, December 1977, pp. 86-88; "Do We Need to Teach a Grammar Terminology?" Gary A. Sutton, *English Journal*, December 1976, pp. 37-40.

Stephen Marcus, Ph.D., is Assistant Director of the South Coast Writing Project, one of approximately 80 affiliates of a national network of programs based on the Berkeley/Bay Area Writing Project. He has written articles on computer-assisted writing instruction for the English Journal and Classroom Computer News and conducts workshops in computers and the language arts for student and faculty development programs. Marcus

is also on the Santa Barbara County Task Force on Microcomputers in Education. Presently at the University of California, Santa Barbara, Dr. Marcus works closely with colleagues using computers in freshman English classes and with the UCSB Microcomputer Systems Laboratory. His computer-assisted poetry-writing software, "Compu-poem," will be reviewed in a future issue of The Journal.

Cataloging Information

Dewey Decimal Number: 372.6
 ERIC Descriptor: Language Arts
 Subject Heading (Sears): English Language—
 Computer programs

Source Information

Program	<i>Sentence Diagramming, Parts of Speech, Usage</i>
Source	Avant-Garde Creations P.O. Box 301061 Eugene, OR 97403
Equipment	48K Apple II or Apple II Plus One or Two Disk Drives Video Monitor or Television with RF Modulator
Package	One Diskette and Teacher's Guide
Suggested Retail Price	\$19.95 Available from Local Dealer

Clock and Calendar Skills

Hartley Courseware, Inc.

Barbara Paape

Time and day! An appropriate activity on the computer.

These two programs are designed to assist in the instruction of telling time and mastering the order of weekdays, months and their abbreviations. Since these concepts often prove particularly difficult for students with learning handicaps or perceptual discrepancies, these programs have great value in special-education classes as well as providing a supplement in the regular classroom.

Clock

After seeing a demonstration of Hartley's *Clock* program at the October '81 CUE (Computer Using Educators) Conference in San Jose, California, I was eager to participate when asked to evaluate its use with my students. My 12 students range from first grade to sixth grade, and each has some learning handicap or perceptual lag in one or more of the following areas—motor, visual, or auditory—that interfere with academic learning. Concepts of time are particularly difficult for many of them. The students, who had had only a few weeks' experience with the Apple II computer, were intrigued by the *Clock* program and found it easy to use.

Clock is an innovative program that allows a student to practice telling time in three different ways. At the beginning of the lesson, the program captures the student's attention with, "WATCH WHILE I DRAW THE CLOCK." and the clock begins to appear (Fig. 1). The student types his name, and a series of activity options are displayed (Fig. 2).

After the student types the number of the activity mode of his choice, further refinements for the selected activity appear for modes 1 and 2 (Fig. 3). Unfortunately, the Teacher's Guide states on page 3 that these selections are given for all modes. It is not until page 5 that the teacher learns that these selections are for modes 1 and 2 only.

Setting the clock hands was a favorite activity for my students. They quickly learned that pressing the F key moves the hands of the clock forward, and pressing the B key moves them backward. When a student thinks the clock hands are in the correct position, he then presses the RETURN key. Thus, if a student overshoots the desired position of the hands, he can correct the hand position and not be penalized. Even the youngest student learned to set the clock hands for the HOUR mode. Older students learned to speed the action by using the REPEAT key while holding down the B or the F key, and many students advanced to practice with the smaller

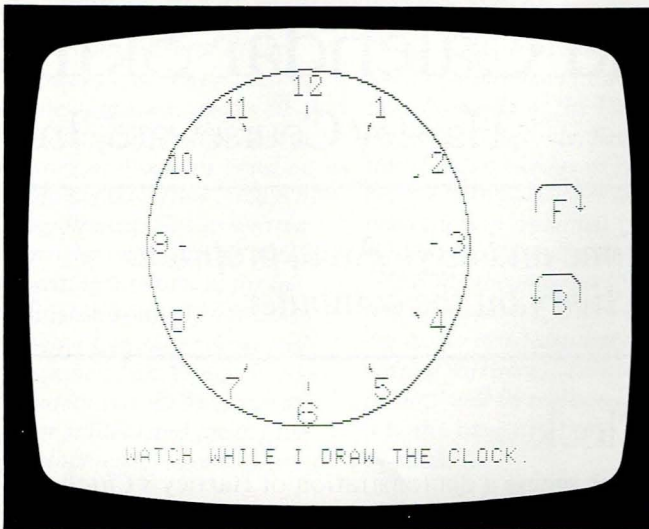


Figure 1

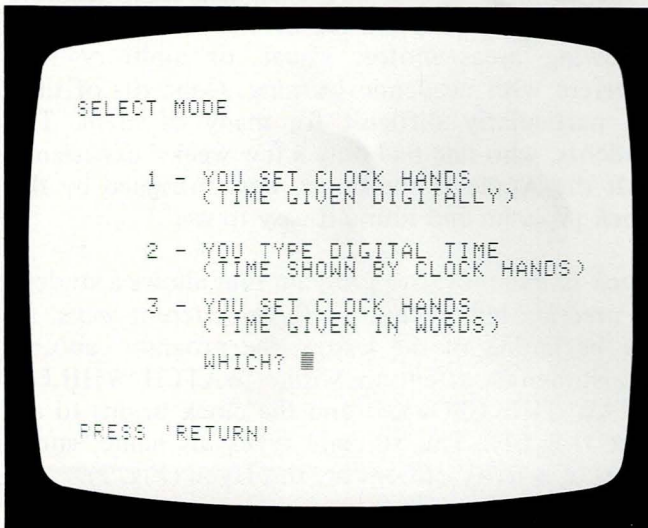


Figure 2

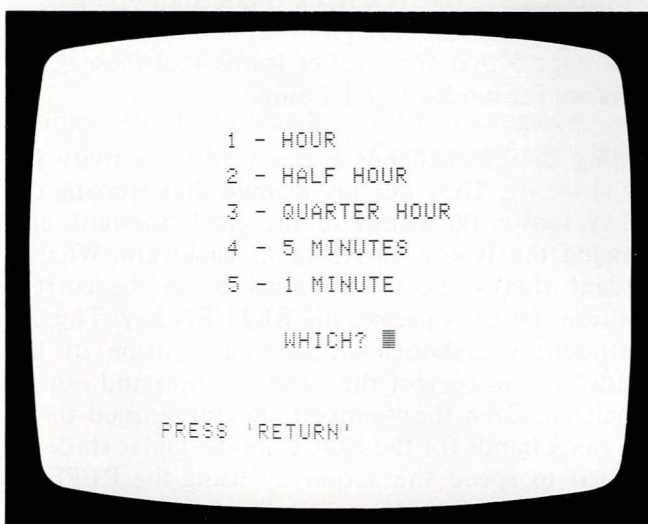


Figure 3

increments of time. As students alternated between setting the clock hands and typing the correct time, it became clear to several of them, for the first time, that at the three-quarter hour, e.g., 1:45, the hour hand is really closer to 2 than to 1. This concept, so logical for adults, can be quite confusing for some children.

Each lesson contains 20 problems, although a student can exit before the lesson is finished. It would have been a more positive experience for my students if the lessons had contained only 10 items. I told some of the students that they could stop after 10 questions. However, as the items were not numbered, the students lost track of the number of questions completed, and several had the feeling they were "quitting". It would have been easier to assign two shorter lessons than only half of a longer lesson. I imagine this might also be a problem in a regular classroom.

When a student enters a correct answer, the computer indicates "CORRECT". After 10 correct responses, the regular clock face changes to a blinking, smiling face (Fig. 4). My students enjoyed this face, but didn't realize, until told, what it meant. A more frequent reward interval probably would have been helpful.

If a student's answer is wrong, a "WRONG - TRY AGAIN" message is shown. The same question remains until the student answers correctly or until the total number of items, or attempts, equals 20. This procedure sometimes caused problems, as some students became quite frustrated and angry when they kept getting a "WRONG - TRY AGAIN" message, and I would have to intervene. I would have preferred that the correct answer be given after the student had made two, or possibly three, attempts to answer the question.

The Teacher's Guide states student errors are recorded into the STUDENT PLANNING file. I found the STUDENT PLANNING option not as helpful as I had hoped. First, I had expected to find "Susie's" records grouped under Susie's name. The scores were filed as the lessons were taken; and if I wanted to find the scores on Susie's last three lessons, I sometimes had to look at records for 30 or 40 lessons. Secondly, while it was quite helpful to see Susie's errors, if she responded to only one item and decided to exit the lesson, the student-planning report told me only that all the items in that lesson were correct. I didn't know whether Susie had completed one, ten, or more items. Lastly, the program is designed to accommodate a printer. The student-planning portion was available without a printer, but

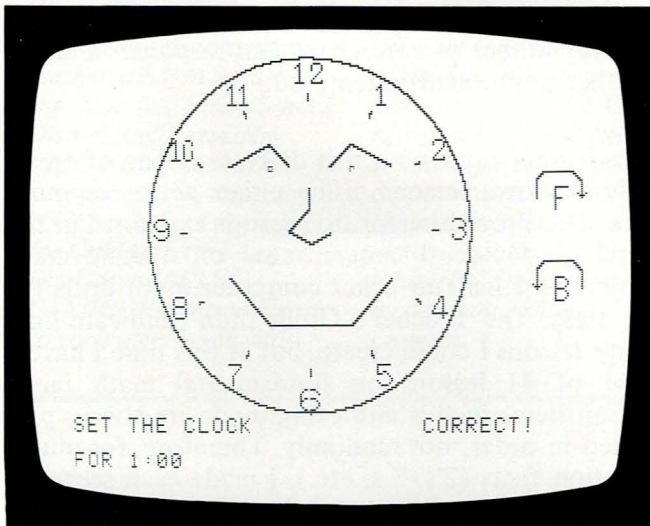


Figure 4

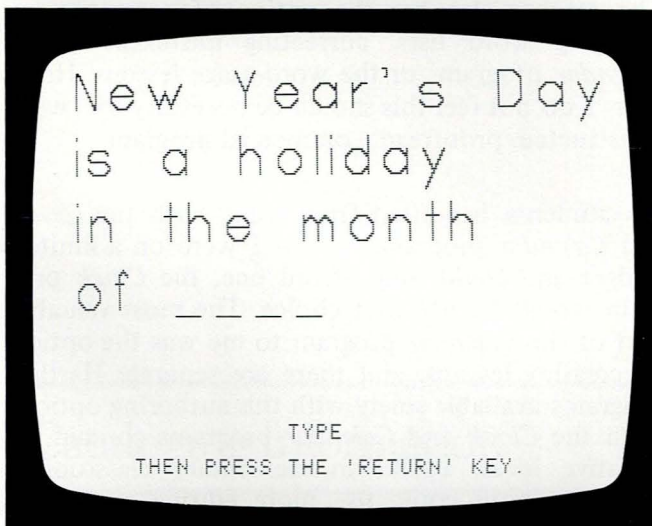


Figure 5

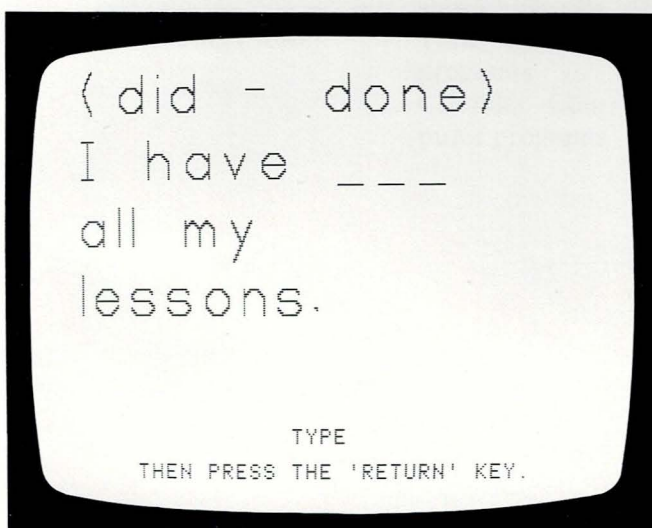


Figure 6

a printer would have simplified finding a particular student's scores.

Objectives for the *Clock* program are clear, and are listed on the last page of the Teacher's Guide under Instructional Skills. The page begins:

Given a diagram of a clock with hands showing "clock time," the student will write the corresponding digital time to the nearest:

- hour
- half-hour
- quarter-hour
- 1 minute

The Teacher's Guide did not identify a user level, but the Hartley catalog indicated that the program was appropriate for use in first through fifth grades. Any teacher will immediately know if his/her students need practice on these skills. I would see the target level as second or third grade in a regular classroom and appropriate for older students in a special-education setting.

Calendar

The concepts presented by Hartley's *Calendar* program, order of days and months, abbreviations, seasons, and major holidays, are important ones for students to learn. However, this program is more difficult for both teacher and students to manage and contains mistakes that can be upsetting to both.

The first difficulty appears after a student types his name. Without being given any list of lessons, he is asked to type the name of the lesson he wants or a "?", which, one learns, will display the catalog. Unfortunately, the names of the lessons in the catalog are listed as CA1, CA2, etc. The student or teacher must go to the last page of the Teacher's Guide to find the objectives or instructional skill for each lesson. The first thing I did was to type a shortened statement of the skill in each lesson and post it on the wall by the computer. Then, at a glance, I could assign the appropriate lesson for each student.

The wording of the questions is clear and easy to read (Fig. 5). One feature I liked occurred in lessons that asked the student to type the name of the month as an answer. The program would accept either JANUARY or JAN. as an answer. In some lessons, the student was asked to type TRUE or FALSE as the answer. I wish there had been the option to type T or F as an alternate correct answer. Unfortunately, in one lesson, the word TRUE was misspelled as TVUE. A student was confused whether he should try to match that spelling or spell the word correctly.

Other, more critical mistakes appeared in Lessons CA8 and CA10. In the first, the computer did not accept the answers of FEB. and NOV. (an option previously allowed) to finish these two statements:

*LINCOLN'S BIRTHDAY IS IN THE MONTH OF
-----.
THANKSGIVING IS A HOLIDAY IN THE
MONTH OF -----.*

Later, the same lesson ended with the statement "*INDEPENDENCE DAY IS A HOLIDAY IN THE MONTH OF -----*". None of the computer keys responded except the RESET key. In Lesson CA10, writing dates the short way, the student is told that:

*6-20-77 IS THE CORRECT WAY TO WRITE
JUNE 10, 1977.
12-25-80 IS THE CORRECT WAY TO WRITE
DEC. 25, 1981.*

There is a note on the last page of the Teacher's Guide indicating that if you have ordered from the old catalog, you will have nine verb-usage lessons on the diskette. The lessons included are on the sheets that are inserted into the booklet. My diskette contained verb-usage lessons, but no sheets were inserted into my Guide. I had to type the word "catalog", find the names of the lessons, US1, US2, etc., and then try each lesson to find the lesson objective. I also typed and posted these lessons on the wall for class use. Each lesson contained sentences with pairs of verbs. Students were asked to type the verb that was correct for the sentence (Fig. 6).

Student planning on this diskette was better thought out than in the *Clock* program. Besides recording a

student's errors, this program provides information on the number of correct student responses and the number of problems attempted.

One further option on this diskette is one of creating one's own lessons, using either words or numbers. The procedure for doing so is explained in the Teacher's Guide. I concentrated on making math lessons as I had no other computer math drills for my class. The Teacher's Guide didn't indicate how many lessons I could create, but at this time I have a total of 41 lessons on fundamental math facts. Lesson items in this authoring mode are always presented in order, not randomly. Therefore, for multiplication facts (2's, 3's, etc.), I made each sentence twice, first in order, then mixed. It was a tedious process, but it was possible to create these lessons without any real knowledge of programming.

The teacher also has the option of correcting or changing word lists, correcting mistakes in the *Calendar* program, or the word-usage lessons. However, I do not feel this should be necessary in a well-constructed, proofread, commercial program.

My students benefited from using both the *Clock* and *Calendar* programs; but if I were on a limited budget and could only afford one, the *Clock* program would be my first choice. The most valuable part of the *Calendar* program to me was the option of creating lessons, and there are separate Hartley programs available solely with this authoring option. Both the *Clock* and *Calendar* programs contain innovative ideas, useful in elementary classrooms. However, both could use more editing, polishing, and proofreading.

Barbara Paape is an elementary and special-education teacher at Eaton School in the Cupertino Union School District, California. Mrs. Paape is a member of Computer-Using Educators (CUE) and the Association for Children with Learning Disabilities (ACLD)/CANHC). Her introduction to computers began with

participation in a 1975-78 Office of Education, Bureau of Handicapped, project to demonstrate the effectiveness of computer-assisted instruction for the handicapped. Paape is also the teacher-consultant on Simon-Says, an Apple Education Foundation model grant authoring program.

Cataloging Information

Dewey Decimal Number: 529
 ERIC Descriptor: Time
 Subject Heading (Sears):
 1. Time-Computer programs
 2. Calendar-Computer programs

Source Information

Program	<i>Clock</i> <i>Calendar Skills</i>
Source	Hartley Courseware, Inc. P.O. Box 431 Dimondale, MI 48821
Equipment	48K Apple II or Apple II Plus Disk Drive Video Monitor or Television with RF Modulator
Package	One Diskette for each program with Teacher's Guide
Suggested Retail Price	\$39.95 <i>Clock</i> \$29.95 <i>Calendar Skills</i> Available from Local Dealer

Apple Courseware Reviews

Again, we wish to extend our appreciation to the courseware developers and publishers who have provided the programs for review in the premiere issues of *The Journal*. As we are all becoming acutely aware, today's world demands computer knowledge. Publishers are taking steps to keep in stride with the changing requirements in electronic-assisted instruction. For their willingness to pioneer new teaching tools and undergo the spotlight of review, *The Journal* acknowledges the program contributions from:

Avant-Garde Creations
 Bertamax Inc.
 Borg-Warner Educational Systems
 COMPress, Inc.
 CONDUIT
 Cook's Computer Company
 CourseWare Magazine
 Creative Publications, Inc.
 George Earl
 Educational Activities, Inc.
 EduTech
 Edutek Corporation
 Edu-Ware Services, Inc.
 Eiconics, Inc.
 M.D. Fulmer Associates
 Grover & Associates
 Hartley Courseware, Inc.
 Hayden Book Company
 J and S Software
 Kyde Tyme Project
 Lightning Software
 Math Software
 Mathware/Math City
 MCE Interpretive Education
 McGraw-Hill Book Company
 Minnesota Educational Computing Consortium
 Micro Music
 Micro Power & Light Co.
 Microsoft Consumer Products

Micro Users Software Exchange
 Milliken Publishing Co.
 Milton Bradley Co.
 Program Design, Inc.
 Programma International, Inc.
 Programs for Learning, Inc.
 Reader's Digest
 Science Research Associates, Inc.
 Society for Visual Education, Inc.
 Special Delivery Software
 Sterling Swift Publishing Co.
 Stoneware Microcomputer Products
 The Learning Company
 University of Illinois
 University of Washington

Future issues of *The Journal* will review commercially available educational courseware for Apple Computer microcomputer systems. When submitting an educational program to *The Journal* for review consideration, include the following information:

- contributor's name, address and phone number
- program name and package information
- system information and requirements
- distribution source
- copyright statements

Send programs to:

The Apple Journal of Courseware Review
 Editor
 20525 Mariani Avenue
 Cupertino, CA 95014

If requested, programs will be returned following their review. *The Journal* will not be held responsible for damage or loss of programs in transit.

Guidelines for Journal of Courseware Reviews

Reviewers are invited to evaluate selected educational microcomputer software that has been submitted to the Foundation's Educational Program Evaluation Centre (EPEC). They are urged to prepare a critique of approximately 1,500 words for publication in *The Journal*. The qualitative reviews are intended to identify models of excellence in program design and development. Our goal is to alert the reader, as a potential microcomputer user, buyer or developer, to the techniques and approaches in program design that promise immediate improvements in the educational process. The reviewers are encouraged to view this as an opportunity to highlight innovative techniques and to comment on the strengths and weaknesses of programs based upon their knowledge of the discipline and/or teaching experience.

To assist in the evaluation of the software, the Evaluation Centre provides the suggested criteria listed below. The list is not intended as an ordered checklist, nor it is necessary for the reviewers to address each item in a category. This accounts for the variety of content. We ask the reviewers to view the list as frames-of-thought to prompt their comments and evaluation. Each reviewer is welcome to expand any point of discussion that emphasizes important aesthetic, pedagogical or economic issues. It is our hope that, through these thoughtful reviews, the Foundation's Evaluation Centre can lead the way in identifying the requirements for maximum courseware design, production and use, and set standards of excellence for a growing computer-literate society.

Review Objectives:

Instructional Design

- Objectives identified/measurable
- Dependent upon prior skills or capabilities
- Activities appropriately sequenced
- Use of instructions
- Originality
- Overall

User level

- Identified
- Recommended

Documentation

- Operating instructions
- Clarity of information
 - Student guide
 - Teacher guide
- Overall

Support of educational process

- Enhances course material
- Improves probability of communication
- Advances principles or concepts
- Activities based upon explicit rules to be learned
- Currency
- Overall

User interest

- Motivating
- Creative
- Adequate/appropriate interaction
- Provides choices/options
- Overall

Programming quality

- User proof
- Sound techniques
- Graphic displays (novelty or integral part)*
- Use of color/sound
- Overall

How critical is subject matter?

Is this an innovative use of microcomputer technology?

Would you recommend this program?

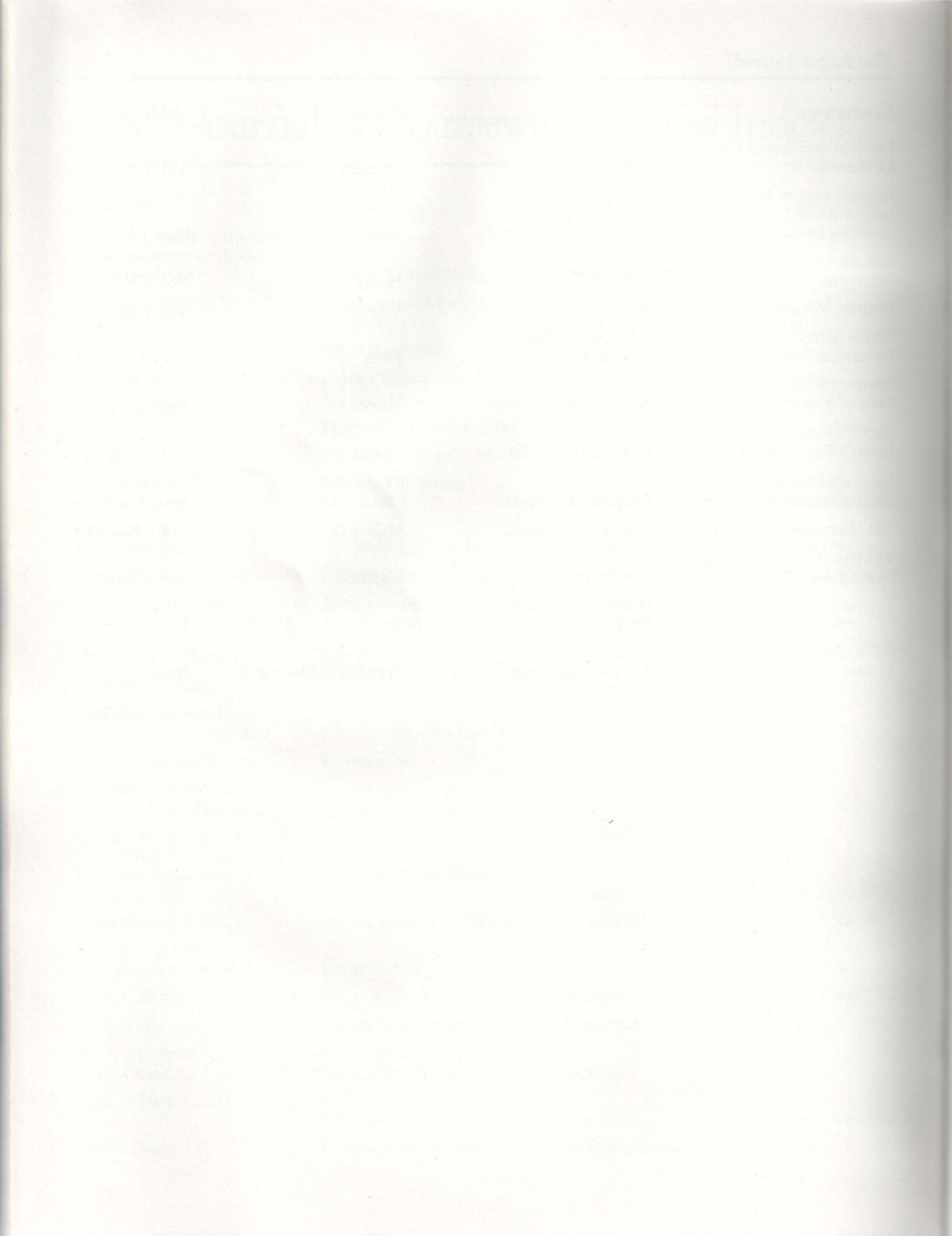
What improvements, if any, could you suggest to strengthen this package in content, program or documentation?

*Reviewers are asked to identify at least four screen displays suitable for reproduction in the finished article. These should be outstanding examples of monitor displays or specific screens that support their comments. The editor reserves the right for final format and screen selection.

Apple Journal of Courseware Review Index

Courseware	Contributor	Reviewer	Publication
Applesoft BASIC Tutorials Microcomputer Workbook: Apple II Edition	Sterling Swift Publishing Co.	Hilfinger	Issue 1
Arcademic Skill Builders in Math	Developmental Learning Materials	Mason	Issue 1
Arnold	Micro Music, Inc.	Wittlich	Vol. 1 No. 1
Calendar Skills	Hartley Courseware, Inc.	Paape	Issue 1
Chemical Equilibrium, Haber	Programs for Learning, Inc.	Petrich	Vol. 1 No. 1
Clock	Hartley Courseware, Inc.	Paape	Issue 1
Comprehension Power Program	Milliken Publishing Company	Kleiman	Issue 1
Compu-Math: Arithmetic Skills	Edu-Ware Services, Inc.	Gilmer	Vol. 1 No. 1
Compu-Spell	Edu-Ware Services, Inc.	McCarthy	Issue 1
Computer Discovery	Science Research Associates, Inc.	Finkel	Vol. 1 No. 1
Elementary, Volume 4	MECC	Calabrese	Issue 1
Elementary, Volume 6	MECC	Goodson	Vol. 1 No. 1
English Basics Part II Concepts in Language Arts	Educational Activities, Inc.	Czechowicz	Vol. 1 No. 1
Eureka Learning System	Eiconics, Inc.	Sprecher & Chambers	Issue 1
Geometry and Measure- ment—Drill and Practice	Special Delivery Software	Kelly	Vol. 1 No. 1
Guinness World Records Microcomputer Programs Fractions, Decimals and Percent	SVE/Society for Visual Education, Inc.	Jerman	Vol. 1 No. 1
Hand-Holding BASIC	Special Delivery Software	Critchfield	Vol. 1 No. 1
How to Evaluate Educational Courseware	Hakansson		Vol. 1 No. 1
Interval Mania	Micro Music, Inc.	Wittlich	Vol. 1 No. 1
MasterType	Lightning Software	Brockman	Issue 1
MicroSystem80: Critical Reading	Borg-Warner Educational Systems	Piestrup	Issue 1
Milliken Math Sequences	Milliken Publishing Company	Forman	Vol. 1 No. 1
Physics CompuLab	EduTech Corporation	Walker & Wood	Issue 1

Courseware	Contributor	Reviewer	Publication
Practicando Espanol Con La Manzana II	CONDUIT	Meredith	Vol. 1 No. 1
Programming in Apple Integer BASIC: Self-Teaching Software	Hayden Book Company	Hoag	Issue 1
Scatter: Unit on Particle Scattering	CONDUIT	Haney	Vol. 1 No. 1
Science, Volume 3	MECC	Caspers	Vol. 1 No. 1
Search Series— Geography Search	McGraw-Hill Book Company	Clark	Issue 1
Sentence Diagramming, Parts of Speech, Usage	Avant-Garde Creations	Marcus	Issue 1
Step By Step—A Beginner's Course in BASIC	Program Design, Inc.	Casella	Vol. 1 No. 1
Successful Student/ Microcomputer Interactions	Friend & Milojkovic		Issue 1
The Microcommunicator	Grover & Associates	McDonald	Vol. 1 No. 1
Tribbles: An Introduction to the Scientific Method	CONDUIT	Koetke	Vol. 1 No. 1
Typing Tutor	Microsoft Consumer Products	May	Issue 1
Z.E.S. Zenith Educational System	Avant-Garde Creations	Sprecher & Chambers	Issue 1



Apple Education Foundation

Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, California 95014