CERL Report X-37 May, 1973

COMPUTER-BASED SCIENCE EDUCATION

DONALD L. BITZER, BRUCE ARNE SHERWOOD,
PAUL TENCZAR

Computer-based Education Research Laboratory

University of Illinois Urbana Illinois

PLATO IV is being developed with support from the State of Illinois, the National Science Foundation, the Control Data Corporation, the Advanced Research Projects Agency, and the Ford Foundation.

Presented at a conference on "Utilization of Educational Technology in the Improvement of Science Education," UNESCO, Paris, 13-16 September 1972

Summary

Students use computers in science education in two ways: they can write computer programs in order to study complex systems and to learn numerical techniques; and they can interact with educational computer programs written by teachers. The first type of use is as widespread as computers, while the second has been severely hampered by the lack of suitable authoring and delivery systems. Our paper concentrates on the latter form of computer-based education and gives examples of materials written by biologists, chemists, mathematicians, and physicists for their students. These materials reflect diverse teaching styles and strategies, including tutoring, simulation or modeling, and drills. By the variety and complexity of these examples we hope to dispel the misconception that the role of the computer is limited to "programmed instruction" or to the presentation of simple multiple-choice questions.

The problems of computer-based education include: (1) The need for an adequate terminal for student use. The common teletype is not adequate in science education---a graphical display terminal is required, a device which can rapidly display line drawings, graphs, and pictures. (2) The need for adequate computing power. A weak computer may only retrieve stored questions and recognize stereotyped responses. To go beyond this simple "teaching machine" function requires enough computing power to generate displays and problems and to recognize open-ended responses. (3) It must be possible for good teachers to author materials without requiring the services of expert computer programmers. This implies the need for a suitable authoring language and system. (4) The cost of computer-based education must become far lower than it has been. Typical costs have been several dollars per hour per student, which does not compete with a human tutor. It was necessary to invent a new technology in order to make progress toward economically viable computer-based education.

One solution to the problems of computer-based education is the PLATO IV system now beginning operation at the University of Illinois. (1) The heart of the student terminal is the plasma display panel, a flat sheet of glass upon which the computer can light up or turn off any of a quarter-million dots (in a 512 by 512 grid) to display text, graphs, and line drawings. The computer can select color photographs to be projected on the back of the transparent panel. For technical reasons discussed in the paper, this display device represents a major advance over previous technology, including the cathode-ray tube. (2) The PLATO system is controlled by a large scientific computer with adequate power and speed to permit the presentation of complex material. The system responds to student input within a fraction of a second. (3) Authors write their own materials in the TUTOR language, which is powerful yet easy to use. Computing power in the PLATO system is used to aid authors in their creative process. (4) When fully implemented it is estimated that capital and operating costs will be $0.50 per student hour at a terminal. Part of the cost reduction is due to a radical restructuring of the way in which the computer itself is operated: in particular, fast electronic memory replaces slow mechanical memory for many important functions, which leads to greatly improved computer utilization.

Appendices to the paper discuss the contrast between large and small computer-based education systems and give an example of the use of the TUTOR language.

"Computer-based Science Education"

Donald L. Bitzer, Bruce Arne Sherwood, Paul Tenczar
Computer-based Education Research Laboratory
University of Illinois, Urbana, Illinois, 61801, U.S.A.

Computers are being used increasingly in science education, both in direct instruction and as calculational tools. Science teachers can prepare computer-based educational materials with which students interact at their own rate, giving the individual student a patient and intelligent tutor which can simulate complex phenomena, drill on basic concepts, and diagnose and treat weaknesses in preparation or comprehension. Students may write their own computer programs and treat problems that transcend the limitations of traditional analytic approaches.

Although computers are used to supplement science education in hundreds of institutions around the world, in only a few schools and colleges have whole courses involved the computer in a major way. Of those few projects which have given birth to complete computer-based courses, fewer still have focussed on the engineering issues of making mass utilization practicable; most projects have rather been directed at exploration and small-scale testing. The PLATO IV system[1] of the University of Illinois is the culmination of a major research and development effort begun in 1959 leading to a viable computer-based education system. More than 1500 hours of computer-based educational materials have been prepared on the PLATO system, including over 20 one-semester courses in diverse fields. This large curriculum base includes lessons representing all the major types of computer utilization and pedagogical styles, and we will use PLATO lesson examples to illustrate the role of the computer in science education.

The lesson examples which follow will help define what we mean by computer-based education: in particular, we hope to dispel the misconception that the role of the computer is limited to the presentation of simple multiple-choice questions. The lesson examples are followed by a discussion of the fundamental problems associated with making computer-based education viable.

The figures which accompany the lesson examples are half-size photographs of the 21 x 21 cm^2 (8.5 x 8.5 in^2) plasma display panel of a student's individual PLATO terminal. The plasma display panel shows orange text and graphics on a black background; negatives are shown here for ease of reproduction. It is important to bear in mind that each student has his own individual terminal (with display screen and typewriter keyboard) and that it is highly unlikely that two students would have the same picture on their screens simultaneously. Indeed, it is unlikely that two students would experience identical presentations of the lesson, since the computer interacts with each student on an individual basis.

LESSON EXAMPLES

Biology

Simulation techniques are used in a genetics course to teach students the laws of inheritance. This computer laboratory allows students to conduct a standard series of fruit fly matings. Each student is presented with stocks of parent flies to examine. Besides flies with normal characteristics, mutant flies can possess such features as white or pink eyes; vestigial or veinless wings; and black, ebony, or striped bodies. The flies are not pre-stored pictures but consist of assembled parts: head, eyes, thorax, wings, and abdomen. Each fly is efficiently coded by a single computer word which specifies the exact type of each body part. This is similar to the biological coding of information in genes located on chromosomes. The computer can construct flies with any combination of normal and mutant characteristics. When the student requests that a mating be made, within seconds the offspring are displayed. Since most mutant characteristics are recessive, they do not appear in the first generation offspring. The student can choose some of the first generation flies as parents for yet another generation (figure 1). Students maintain a scientific logbook of all these experiments so that they can later do statistical tests of hypotheses and hand in the results in a formal laboratory report.

All of the offspring are generated by using random numbers and probabilities based on the Mendelian laws of inheritance. Thus, this computer analog of the real biological system produces thousands of possible outcomes and gives each student his own experiment. A series of conventional fruit fly experiments takes several months. Culture medium must be prepared, bottles sterilized, flies examined at odd hours, etc.. Using the computer, a student can perform the basic experiments of Mendelian genetics in 3 to 4 hours' work. This time compression of the experiments makes the logical flow of the multi-step process much more comprehensible. In addition, this experiment in a conventional course is almost always performed "cookbook" style since not enough time or help is available for the student to go his own way. With the computer, the student can explore various experimental strategies, for it takes only minutes to start over, and help is always available.

The biology lesson just described is one of thirty-five lessons designed to introduce beginning college students to genetics and evolution.[2]

The students spend 4 hours a week with the computer, where all lesson material is presented, followed by a 2 hour discussion period with a human teacher.

Geometry

Pattern recognition is a basic feature of a series of 15 lessons designed to teach informal geometry to junior high school students.[3] This course is designed to give students experience with the facts of plane geometry (symmetry properties, definitions, etc.) before formal proofs are attempted. The students are asked by the computer to construct specific geometric figures by using a set of 8 keys to move a cursor around the screen (Figure 2). When ready, the student can request that the computer "judge" his work. Figure 3 demonstrates a typical sequence of the student-computer interaction. It should be evident that sophisticated "judging" algorithms must be used rather than stored answers since the size, orientation, and even specific type of figure can be arbitrary. Furthermore, when the student has an improper figure, the computer must do more than just tell the student his figure is wrong or the student will become frustrated. The student's attention must be directed to specific features of his construction that are inadequate.

Chemistry

Dialog with the computer supplies the student with information needed to identify chemical unknowns in a qualitative analysis lesson[4]. A student-computer exchange, in the student's natural language, tends to maximize flexibility of approach for the student[5]. Here is an example of a college student's conversation with the computer. (Remember that this dialog actually appears on a display screen, not a teletypewriter. In particular, the spectra are shown on the screen in color.)

Student: What color is it.
PLATO: The solid is white and the liquid is colorless.
Student: What is its melting point
PLATO: The melting point is 19–21°.
Student: What does the nmr look like
PLATO: This is the 60 MHz proton nmr spectrum. For a table of chemical shifts, press DATA. (The spectrum is shown on the screen.)
Student: ir
PLATO: This is the ir spectrum. (The infrared spectrum is displayed on the screen.)
Student: What is the unknown soluble in
PLATO: Please provide more information.
Student: Is it soluble in H_2SO_4
PLATO: The unknown is soluble in cold conc. H_2SO_4.
Student: Elemental Analysis
PLATO: We can give you results from a sodium fusion. To try it press LAB.
Student: Picric Acid Derivative
PLATO: Please forgive . . . . The following words are NOT in my vocabulary . . . picric.
Student: dnp
PLATO: The unknown reacts with 2,4-dinitrophenylhydrazine to form a new substance with a melting point of 249–251°.
Student: Where does it boil
PLATO: The boiling point is 195–200° at 760.
Student: Permanganate Test
PLATO: The unknown reacts slowly with $KMnO_4$.
Student: Tin and HCl
PLATO: Please forgive . . . The following words are NOT in my vocabulary . . . Tin.
Student: Tollens
PLATO: There is no reaction with Tollen's reagent.
The unknown is
Student: Acetphenone
PLATO: SP
Student: Acetophenone
PLATO: OK
For a new unknown press NEXT.

After the student has obtained enough information about the unknown, he can attempt an identification (Figure 4). Shown is a student's misspelled but proper identification. The computer recognizes that the student's answer is correct but that the spelling is inadequate. Spelling algorithms such as this must be a basic part of any educational computer system for, as in this case, if the student were merely told his response is wrong, he could easily spend a great deal of time on the wrong track.

The score shown in Figure 4 refers to an aspect of the lesson in which the student tries to determine the unknown with the least "cost" in chemical and physical tests. Some tests cost more than others: for example, determining the infrared spectrum costs ten points while the melting point costs only one point. The chemist who wrote the lesson wanted to encourage students to make simple tests before using expensive apparatus such as an infrared spectrometer.

This lesson is not meant to replace the organic chemistry laboratory. Rather it is meant to sharpen the intellectual process of formulating questions and interpreting results before the student enters the laboratory. Thus, in a matter of hours, a student can logically identify 5 or 6 unknowns -- often more than the student would identify in a whole semester's work in the laboratory. This is just one of the many chemistry lessons totalling 30 hours taught by computer at the University of Illinois.[5]

Physics

In an introductory mechanics course students are asked to participate actively in the derivation of the basic kinematics equations. In Figure 5a the student has given an algebraically incorrect expression in one step of the derivation: a correct answer is $(v_i + v_f)/2$. In Figure 5b a simple numerical example shows the student the inconsistency of his formula. This procedure shows the student an important method for checking the validity of an algebraic expression, and this numerical substitution method permits the computer to handle appropriately all possible algebraic responses, independent of form. In Figure 5c the student has given a complicated but algebraically correct response and the computer has noted that the response is correct but not in the simplest form. This distinction is made on the basis of the number of arithmetic operations encountered in the numerical evaluation of the student's expression.[7] As in the previous examples, it can be seen that judging student responses by algorithm rather than by comparison with a list of stored answers gives the student great freedom and contributes to heightened interaction.

This example is drawn from a one-semester computer-based mechanics course[8] in which students spend two hours at the beginning of each week studying at a computer terminal. The computer introduces basic concepts, treats applications, simulates phenomena, and tests comprehension. Classroom and laboratory work later in the week build on this solid preparation.

Mathematics

An example of the computer as tool is shown in Figure 6. The student has written a short program to evaluate and plot a parametric function, with an angle "t" as the varying parameter. An attempt has been made to keep the computing language[9] as close to standard algebra as possible to avoid inconsistencies with the natural language of direct instruction.

(The most extensive and successful integration into education of the computer as tool has taken place at Dartmouth College, Hanover, New Hampshire where nearly all students, including non-science students, write computer programs as an integral part of their studies and recreation[10].)

Programming by Children

Young children can be taught the basic elements of programming. First, a series of games teaches the child a set of operations which can be carried out by a little man on the screen. In Figure 7a the child has walked the man, one step at a time, through a maze: in Figure 7b the child learns how to pick up a ball, carry it, and put it down. (A set of 8 keys on the keyboard move the man one step in the 8 basic compass directions. The "plus" key picks up a ball and the "minus" key puts it down.) After learning the basic operations, the child can write a list of operations for the man to carry out, as in Figure 8, and watch the man follow instructions.[11] An important aspect of this exercise is that the child can write an inconsistent program and receive an error message such as "There is no ball here to pick up!" The child enjoys giving directions to the man, and he sees the important aspects of a computer and a program: step-by-step processing, repetitive loops, the concept of an operation, etc..

PROBLEMS AND SOLUTIONS

By example we have defined what we mean by computer-based education. Now it is appropriate to ask what are the basic problems in the field of computer-based education.

Many difficulties have restricted the application of computers in education. Educational computer systems have been too expensive for wide-scale use. The limitations of many existing systems have in some educational circles caused "computer-assisted instruction" to be identified with mere multiple-choice testing or simple drill. (We use the term "computer-based education" in the hope of avoiding this identification.) The authoring of computer-based lessons often has been very difficult, requiring the services of computer programming experts; teachers find themselves shut out of participation in structuring their courses. We will discuss these and other problems facing computer-based education, then report on progress being made in solving these problems.

Display

The physical form of communication between computer and student, or between computer and teacher, is fundamental to all other questions. In most cases, the student's communications device ("terminal") has been some kind of alphanumeric display, usually a teletypewriter. While adequate for some special purposes, its slowness, limited character set, noise, and near inability to draw graphs and diagrams make it a poor medium for the full educational message. The expanded capabilities of a fast graphical display device are almost mandatory for most educational purposes, especially in science education. Such devices have in the past been very expensive, but new technology is rapidly changing this situation. Ideally, the student's terminal should permit photographic image projection, two-way voice communication, pointer inputs, etc., in order to fully engage as many senses as possible---visual, auditory, tactile. At some point, economic considerations force compromise with this ideal, but the basic principle should be kept in mind: education is an extraordinarily difficult human enterprise, and it requires a flexible and powerful medium.

Computing Power

Too often there has been a narrow conception of the role of the computer itself in "computer-assisted instruction". The computer is thought of as a minor control element, choosing and relaying essentially static information to the student and distinguishing among a few standardized replies from the student. For such purposes a small or weak computer is sufficient. But for more general purposes, especially in science education, a powerful computer is required to generate (rather than merely retrieve) material for the student and to process open-ended student replies and questions. How is such power to be paid for? Evidently it must be shared among a large number of users; new ways of organizing such large systems have now made this feasible and economical. There is a hint here of the advantages of a large system over a small system. We return to this point in detail in Appendix A.

Authoring

For a computer-based educational system to be viable and to be accepted by the educational community, it must be relatively easy for good teachers to create computer-based lesson materials. Having a powerful rather than a weak computer at the heart of the system makes it possible for the system to help rather than hinder lesson authors in their creative work. Quality improves and costs drop by eliminating programmers and by placing the author in direct control of his medium. In order to achieve this close coupling to authors, it is necessary to create a suitable author language and authoring environment which strongly minimize the need for special computer knowledge. The ideal is to make the system transparent and responsive to the author as well as to the student.

Cost

Another critical issue is cost. The use of computers in direct instruction will be possible on a wide scale only if this is economically feasible, no matter how great may be the supposed benefits. Typical costs of educational computer systems have been about five to ten dollars per student contact hour, which is adequate to hire a good private tutor. A cost reduction of a factor of ten is required to make it feasible to use computers in education. It is crucial that overall computer system costs be driven as low as possible, while yet implementing enough power and flexibility to be useful. These conflicting requirements have forced the invention and development of completely new technologies. Contrary to much lay and professional belief, the computer technology of the 1960's was incapable of widespread educational application: the costs were too high, even for rather primitive systems. The new technologies include a radically different display device for the terminal, unique telecommunications, and a drastic restructuring of the computer's software to reflect the interactive educational environment.

SOLUTIONS

We have discussed some of the basic problems which have inhibited widespread application of computers in education. Next we will discuss the ways in which these problems have been faced in a particular case---the PLATO IV computer-based education system of the University of Illinois.

Display

In the introduction of this article we gave examples of computer-based educational materials. They were produced on the PLATO IV system, and the figures are photographs of the student's display screen, which is a flat plasma display panel[12], not a television tube. Some discussion of this device is necessary to explain the nature of the student terminal. The plasma panel is a crucial element in making feasible a sufficiently flexible communications medium.

Until recently, the cathode ray tube was normally used for displaying computer-generated graphical and pictorial information to the student. Because a cathode ray tube must be refreshed thirty times per second to maintain the image without objectionable flicker, an expensive external memory device is required in addition to the television apparatus itself. (In the case of home television the image is sustained by the broad-band video channel transmitted continuously from the television station. For individualized use the memory device must be near the display unit, since assigning a video communications channel to each user is prohibitively expensive.) The "storage" cathode ray tube is a television tube with built-in memory due to the special electrostatic properties of its faceplate. This device is better suited to computer-based education, for the computer need transmit the graphical information only once and no refreshing is required. However, the storage television tube does have drawbacks. One major problem is the difficulty of performing a selective erase of a small portion of the display without disturbing the rest of the picture. Selective erase is necessary for many aspects of computer-based education, including erasing and retyping part of an answer, and in animated sequences performed by drawing a figure, pausing, erasing the figure, and redrawing it elsewhere on the screen to give the impression of motion. Other problems include the long period required to erase the entire screen, the need for frequent maintenance adjustments, and the impracticality of superimposing photographic information on the screen.

The plasma display panel was invented at the University of Illinois to solve these problems. Its memory is at the display unit, inherent to the panel. It permits the selective erase of even a single dot without disturbing the rest of the picture. The display is bright, with high contrast, and free of flicker or fading. The panel consists of two sheets of glass on which are deposited 512 horizontal and 512 vertical conductors (the conductors are transparent). Neon gas between the horizontal and vertical conductors can be made to glow as bright dots at the intersections of the 512 by 512 grid of conductors. (The resolution is 2.4 dots/mm .) The simple structure lends itself to low-cost mass production. The organization into a 512x512 grid of dots is ideally suited to addressing by a digital computer. The plasma panel makes possible, at low cost, graphical display capabilities that formerly were available only at prohibitive expense. Moreover, the simplicity of the device makes possible additional economies in the design and operation of the telecommunications and of the computer software.

Because it is flat and transparent, the plasma panel can support a rear projection screen for color slides or movies, selected and driven under computer control, with computer-generated text and graphics superimposed on the plasma panel. This unique combination adds an important dimension to computer-based education. For example, the computer can select a full-color slide of the human heart for a medical student, then superimpose pointers or animated flow markers on the plasma panel to illustrate dynamically the complex action of the organ. Note that transmitting color photographs from the computer would make communications extremely expensive. We do not show an example of a superimposed color photograph in this article because of the difficulty of reproduction.

With this explanation of the display device, it is useful to look again at the photographs of the fruit flies and of the geometry lesson. Note that the usual optical distortions of television are completely absent---the flat display panel with its evenly spaced grid gives a display free of distortion or jitter. The resolution is so fine that a viewer is unaware that the text and graphics are actually composed of individual dots.

The fruit fly picture illustrates another important aspect related to the symbols needed for education. In addition to the standard upper- and lower-case letters, numbers, punctuation marks, etc., lessons in some subject areas require a rather large set of additional symbols. For example, when teaching Russian, the Cyrillic character set is needed. When teaching physics much of the Greek alphabet plus mathematical symbols may be needed. The fruit flies are drawn as appropriately positioned symbols---right wings, left wings, eyes, etc.. This display mode is many times faster than drawing the flies one point at a time. At the beginning of the student session the computer transmits the required special symbol patterns to the terminal---Russian, Greek, or fruit fly parts---and thereafter the computer need only specify which symbols to plot at what screen positions. The PLATO terminal writes 180 symbols per second, each symbol consisting of an 8x16 grid of dots. Similarly, the PLATO IV terminal has enough intelligence to draw the many dots comprising the lines in the geometry lesson simply from endpoint specifications sent by the computer, at the rate of 60 connected lines per second.[13]

There is much more of a technical nature that could be said concerning the nature of a student terminal useful in education, but hopefully the heredity and geometry examples illustrate the basic point: for educational purposes a sophisticated terminal is required. As an exercise, imagine transferring the pedagogical approach of these two examples to a system with typewriter terminals. It would be impossible to preserve the essential aspects of these educational materials, proof that the nature of the student terminal largely determines the possible pedagogical approaches.

The effect of the type of terminal on the range of educational possibilities has been too often underestimated. We have seen interesting pedagogy created following the introduction of each new terminal capability. Other devices under development that have already generated unusual lesson material are a random-access audio device and a touch-panel that permits the computer to recognize where the student is pointing at the display screen. In recognition of our present ignorance of what may prove to be valuable in the future, the PLATO IV terminal has extra input and output connectors for easy attachment of new devices.

Computing power

The need for computing power is well illustrated by the introductory examples. The fruit flies are generated randomly, following the statistical laws of inheritance. No two students will experience the same results, except in the statistical sense. It is the biological algorithm of Mendelian genetics that is programmed. The algorithms of the geometry lesson involve much computation to achieve the accurate pattern recognition of the student's open-ended geometrical response. Understanding the chemistry student's free-form questions requires organized searches of a rather large data base of vocabulary words and basic concepts. All of these aspects of computer-based education require a powerful computer, as opposed to the meager computing requirements for simple multiple-choice materials. Because the memory banks and other non-computational parts of a computer system comprise a major portion of the total system and are similar in cost whether the computational unit is powerful or weak, a weak system can easily be more expensive than a powerful system. The weak system may be capable only of simple programmed instruction or multiple-choice testing which can be done much more cheaply with books and other media. Only a powerful system can, through its enhanced capabilities, justify its cost.

Authoring

The fruit fly lesson was written by a biologist, the geometry lesson by a mathematician, and the lesson on qualitative organic chemistry by a chemist. These authors were able to create these sophisticated materials on their own, without the aid of programmers. This relates directly to the need for strong computing power in the system to lift much of the programming burden from the lesson author, yet place the author in direct control of the medium.

One of the major tasks in building the fruit fly lesson was the generation of the special characters used to assemble a picture of a fruit fly. The biologist drew the characters directly on the screen, then used these characters in his lesson. To create the dialog lesson, the chemist constructed a list of the relevant vocabulary words, stated word synonymy, listed the basic concepts, and listed the corresponding responses. The system took care of transforming the wide range of student responses into forms which would match the basic concepts and yield an appropriate response. The mathematician's task was facilitated by powerful calculational capabilities easily accessible in the system for performing his pattern recognition task. All three authors benefitted greatly from the system's responsiveness, for they could switch in a few seconds from authoring the lesson to testing it as a "student," then back to writing and correcting it. This speed of transition is enormously useful in lesson creation.

All of the PLATO materials are written in the TUTOR language which is specially designed to facilitate the creation of computer-based lessons utilizing graphical display terminals. We give an example of TUTOR programming in Appendix B.

Cost

We have already discussed two important factors which influence cost: the plasma panel makes possible an inexpensive graphical display, and an appropriate authoring procedure enables authors to create their own materials. Another major cost area is the computer itself, and it is appropriate to discuss briefly the novel computer utilization in the PLATO system.[14]

A "time-sharing" computer, which seems to service many users simultaneously, actually serves only one user at a time. The computer services a user for a few thousandths of a second. If the computer manages to finish its work for all the users within a fraction of a second, each user has the illusion of complete control of the machine. In going from one user to another, the computer must save the first user's program and status and load the second user's program and status. This procedure is called "swapping". The swapping of programs and status takes place between the computer's high-speed memory banks and a mechanical, rotating disk or drum of magnetic recording material. Unfortunately, the mechanical speed of these devices is extremely slow compared with the electronic speed of the computer, so that the swapping procedure involves a heavy overhead. The computer is frequently either waiting for a program to work on or involved in the difficult decision of whether to swap or what to swap in order to maximize its overall efficiency. As a result such systems tend to have high computer costs because the computer is doing useful work only a fraction of the time. To put it another way, the computer can handle only small numbers of simultaneous users and the cost per user is proportionately high. Moreover, computer-based educational materials administered by such a system tend to be of a simplistic frame-presentation nature, because the constraints of a slow swapping procedure require that the material be organized in a linear sequence of very short segments. This is a severe limitation: richness of cross-connections is needed to provide quality materials.

One obvious solution would be to keep the students' lessons and individual status information in the computer's memory and avoid swapping. This has almost never been done because even auxiliary bulk computer memory is far more expensive than disk or drum memory. PLATO started from the premise that this scheme should nevertheless be used to improve quality and to improve computer utilization. It is overall performance that matters, and increased memory costs are offset by the elimination of the high swapping overhead, with drastic improvement in quality. While a student on the PLATO system studies his lesson no swapping to disk or drum occurs: the swapping is to a special auxiliary computer memory of extremely high speed (the Control Data Corporation "Extended Core Storage"). To maximize the usefulness of this memory, lessons are shared, with only one copy of a lesson in the memory no matter how many students are studying it. (In disk-swapping systems, students usually have to have their own copy of the lesson, as well as their individual status in that lesson.)

Because the entire lesson is available, corresponding to one or two hours of student study, PLATO lessons usually are quite complex in the interconnections of their parts and rarely resemble the frame-by-frame question-and-answer format so prevalent in the field of computer-assisted instruction. Again we see that, as with the type of terminal, *the system design has important bearing on the styles of possible pedagogy*. This point has been systematically ignored by too many researchers who have thought that questions of system design were minor compared to pedagogical questions, not realizing that the limitations of their systems were distorting their research results. Only if the system is sufficiently powerful as to pose few constraints on possible educational approaches do the details of the system cease to matter.

The result of this restructuring of the computer utilization in the PLATO system is that the computer ceases to be the most expensive part of the computer-based educational system, because a large computer can now run hundreds of terminals rather than tens of terminals. This order-of-magnitude improvement is due mainly to the elimination of swapping, but is partly due to the simplicity of the plasma panel terminals and associated telecommunications equipment. For a discussion of overall costs, see the articles of Reference 1. Total costs including capital and operating costs are estimated at about $0.50 per student hour at a terminal.

SUGGESTED READINGS

There recently appeared a two-part article by science reporter Allen Hammond on the present state of computer-based education in the United States.[15] It discusses the range of uses of computers in education and the current large-scale projects funded by the National Science Foundation.

Dartmouth College has been active in the computational use of computers in education and is the nucleus of a large network of schools and colleges engaged in these activities and utilizing the Dartmouth computer system. (See Reference 10.)

An important center in Europe, directed by Yves Le Corre, is the "Ordinateur pour Etudiants"[16] of the University of Paris, where there has been work in physics and in biology.

The Physics Curriculum Development Project[17] directed by Alfred Bork at the University of California, Irvine, has produced a considerable body of material in physics. Both direct instruction and computing have been introduced into physics courses at Irvine.

A group led by Wallace Feurzeig at Bolt Beranek and Newman, Cambridge, Massachusetts, has created the LOGO language for computational applications that need not be of a numerical nature. This group and several other groups, including one led by Seymour Pappert at the Massachusetts Institute of Technology, have had students of various ages, including young children and college students, write LOGO programs to study mathematics and problem-solving.[18]

There exists a voluminous literature on computers in education, but the field changes so rapidly that publications earlier than 1969 tend to be of little use now. The utilization of computers in education is almost as widespread as computers themselves, so we have cited only some representative projects whose size and committment have permitted the creation of significant quantities of curriculum materials.

APPENDIX A

Large System vs. Small System

There has been much discussion of the merits of large versus small computer-based education systems. As proponents of the large-system concept, it may be helpful to discuss our reasoning on this matter.

It should first be made clear that we are not talking about the question of "centralization" versus "decentralization," which is essentially a different question. If a student or author has the full power of a large system available at any terminal, whether near or far from a large computer, that system is decentralized as far as the user is concerned. Conversely, if a part of the authoring process for a small system must be carried out in a different place, on a special authoring computer system, then a critical part of the operation of the small system is inconveniently centralized.

Years of detailed data collection on the PLATO III system show that average processing and information transfer requirements for a student are remarkably independent of what subject he is studying, the method of presentation, his age, etc.. For example, an elementary-school student working on a simple drill goes through material rapidly but this material requires little processing or display for each interaction. On the other hand, a college student studying complex scientific material thinks a long time between interactions, but this material requires a great deal of processing and display generation for each interaction. The product of interaction rate and computer processing or display requirements per interaction turns out to be approximately the same in both cases. To be specific, averages of approximately 1000 computer processing operations per second and about 15 displayed characters per second ($\sim$ 150 words per minute) characterize our findings. Since the PLATO III system is characterized by processing and display rates of over twenty times these average rates, the observed average requirements presumably reflect physiological constraints.

In the design of a viable system these averages are not the whole story: the peak requirements are just as important. The science student, on the average, thinks for a long time between interactions, but the system must respond instantly so that the student can continue his line of reasoning without interruption. It would be disastrous to force the student to wait a long time for the reply. There is therefore an enormous difference between the average and peak rates. Without going into the details of "queueing theory," it should be clear that only a large system has the necessary reserve power to work rapidly through the huge peak requirements represented by the science student's interactions. Also, statistical fluctuations in the number of students simultaneously requiring service become less and less damaging to system responsiveness the larger the system. Roughly speaking, if N interactions per second are anticipated, the number observed will be $N \pm \sqrt{N} = N(1 \pm \frac{1}{\sqrt{N}})$ due to Poisson statistics; the probability of overload conditions scales like $1/\sqrt{N}$. These factors favor the large system.

An advantage of a large system that is difficult to quantify is that one large computer can perform much more complex tasks than can a group of small computers of comparable aggregate power. Free-form dialog, complicated display generation, rapid extensive calculations, powerful authoring procedures---all of these are essentially out of the reach of the small computer. The reason for this is rather subtle. A time-sharing computer services only one user at a time. During the fraction of a second that the computer is working for an individual user, all of its basic resources are devoted to him: fast memory, processing unit, data transfer channels, etc.. The more powerful are these resources, the more sophisticated will be the service. For example, a large fast memory with high-speed transfer from bulk memory permits operations on a large data base of vocabulary for natural-language dialogs. Unlike the swapping medium (bulk computer memory or disk memory) whose total cost is proportional to the number of users (each of whom needs some average bulk memory allocation), the basic computer resources used during the actual fractional-second processing are not duplicated for each user. The larger the number of users of the system, the more can be paid for basic computer resources to permit more and more sophisticated processing. With a small number of users, a weak processor with small amounts of fast memory and inadequate transfer capabilities is all that can be paid for, at the same cost per user as will buy much more capability in a large system. (One might object that the cost of the central processor is proportional to the number of users and their required number of operations per second. However, the more expensive processors have added capabilities as well as increased speed, so processing requirements do not scale linearly. Also, the processor usually represents only about 10% of the total system cost.)

There are therefore two related but different reasons why many PLATO capabilities could not be duplicated in systems designed to serve a small number of terminals. One is that the peak demand by a student may take an unacceptably long time to process (and cause queuing problems for other students), and the other is that the basic resources may be inadequate to perform some tasks at all (insufficient memory to manipulate a data base, etc.).

Within a factor of two or three, the management and administration of a large system is comparable in cost to that of a small system. Inevitably there is a director, an assistant director, some computer operators, etc., whether the system is large or small. This makes the management of multiple small systems expensive.

There is no way to make many small computers temporarily look like a large computer in order to carry out heavy computational tasks, such as sophisticated analysis of educational data gathered by the system. A large system can handle both the student interactions and standard computing jobs as well. The processing of standard administrative and research computing jobs helps pay for the system, whereas a small system incapable of this performance is purely an add-on expense. A related point is that the management and distribution of a large data base of curriculum materials is best handled by centralizing the storage of these materials. This permits teachers to monitor students' performance at a distance and assures that lesson material can be updated for all students, everywhere in the network.

REFERENCES

1) Bitzer, D. L., R. W. Blomme, B. A. Sherwood, and P. Tenczar, "The PLATO System and Science Education," in Proceedings of a Conference on Computers in Undergraduate Science Education, IIT and Commission on College Physics, 1970. (Available from American Institute of Physics, 335 East 45th Street, N. Y., N. Y. 10017.)
Alpert, D., and D. L. Bitzer, "Advances in Computer-based Education," Science 167, 1582 (1970).
Bitzer, D. L., and R. L. Johnson, "PLATO: A Computer-based System Used in the Engineering of Education," Proceedings of the Institute of Electrical and Electronics Engineers 59, 960 (1971).
Lyman, E., "A Summary of PLATO Curriculum and Research Materials," Computer-based Education Research Laboratory (CERL) Report X-23.

2) Hyatt, G. W., D. C. Eades, and P. Tenczar, "Computer-based Education in Biology," BioScience 22, 401 (1972).

3) Dennis, J. R., "Teaching Selected Geometry Topics via a Computer System," CERL Report X-3a (1969).
Dennis, J. R., "Identification of Pictorial Responses in Computer-based Geometry Instruction," CERL Report X-3b (1971).

4) Smith, S. G., "The Use of Computers in the Teaching of Organic Chemistry," Journal of Chemical Education 47, 608 (1970).

5) Tenczar, P., and W. M. Golden, "Spelling, Word, and Concept Recognition," CERL Report X-35 (1972).

6) Smith, S. G., and J. Ghesquiere, "Computer-based Teaching of Organic Chemistry," in Volume IV of Computers in Chemistry and Instrumentation, ed. by J. S. Mattson, H. C. MacDonald, and H. B. Mark, Jr. (Marcel and Dekker, Inc., New York, in press).

7) Sherwood, B. A., "Judging Algebraic Expressions and Equations,"American Journal of Physics 40, 1042 (1972).

8) Sherwood, B. A., C. Bennett, J. Mitchell, and C. Tenczar, "Experience with a PLATO Mechanics Course," in Proceedings of a Conference on Computers in the Undergraduate Curriculum, Dartmouth College (1971).
Sherwood, B. A., "Free-body Diagrams (a PLATO Lesson)," American Journal of Physics 39, 1199 (1971).

9) This "GRAFIT" computing language, created by B. A. Sherwood, is itself written in the TUTOR language, as are all the lessons discussed in this article.

10) Kemeny, J. G., and T. E. Kurtz, "Dartmouth Time-Sharing," Science 162, 223 (1968).
Nevison, J. N., "The Computer as a Pupil: The Dartmouth Secondary School Project," Kiewit Computation Center Report (1970), Dartmouth College.
Biennial Report 1969-1971, Kiewit Computation Center Report.
Luehrmann, A., "Should the Computer Teach the Student, or Vice Versa?" AFIPS Conference Proceedings 40, 407 (1972).
Also see a sequence of five papers in AFIPS Conference Proceedings 34,649-689 (1969).

11) This pictographic programming language was created by P. Tenczar. It is written in TUTOR.

12) Johnson, R. L., D. L. Bitzer, and H. G. Slottow, "The Device Characteristics of the Plasma Display Element," *IEEE Transactions on Electron Devices*, *18*, 642 (1971).

13) Stifle, J., "The PLATO IV Student Terminal," Proceedings of the Society for Information Display, *13*, 35 (1972).

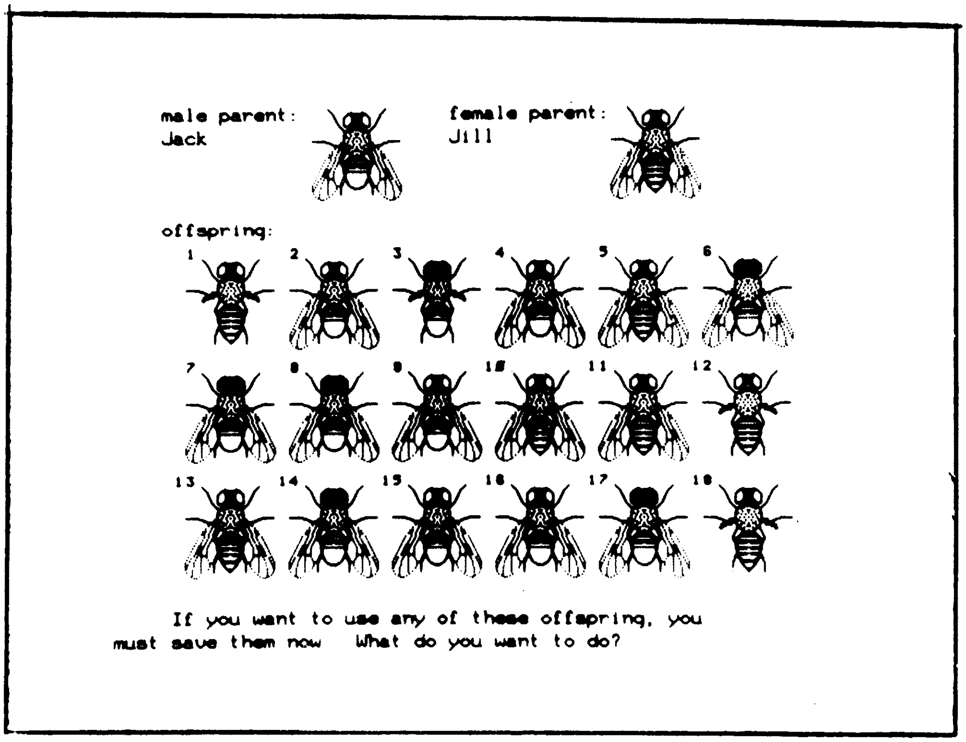
14) Tenczar, P., R. W. Blomme, J. H. Parry, and B. A. Sherwood, "PLATO IV System Software," to be published.
Stifle, J., "PLATO IV Architecture," *CERL Report X-20*, 1972.
Stifle, J., D. L. Bitzer, and M. Johnson, "Digital Data Transmission via CATV," *CERL Report X-26* (1972).
Bitzer, D. L., R. L. Johnson, and D. Skaperdas, "A Digitally Addressable Random-Access Image Selector and Random-Access Audio System," *CERL Report X-13* (1970).

15) Hammond, A. L., "Computer-Assisted Instruction: Many Efforts, Mixed Results," *Science 176*, 1005 (1972).
Hammond, A. L., "Computer-Assisted Instruction: Two Major Demonstrations," *Science 176*, 1110 (1972).

16) Rapport d'activite O.P.E., Laboratoire de l'O.P.E. (Paris-VII), 84p. (1972).
Fiszer, J., "The Use of a Computer in Teaching Biology. Current Trends at the O.P.E. Laboratory," Symposium über Programmierte Instruktion und Lehrmaschinen, G.P.I. (Hochschuldidaktik), Berlin 5-8 April 1972.

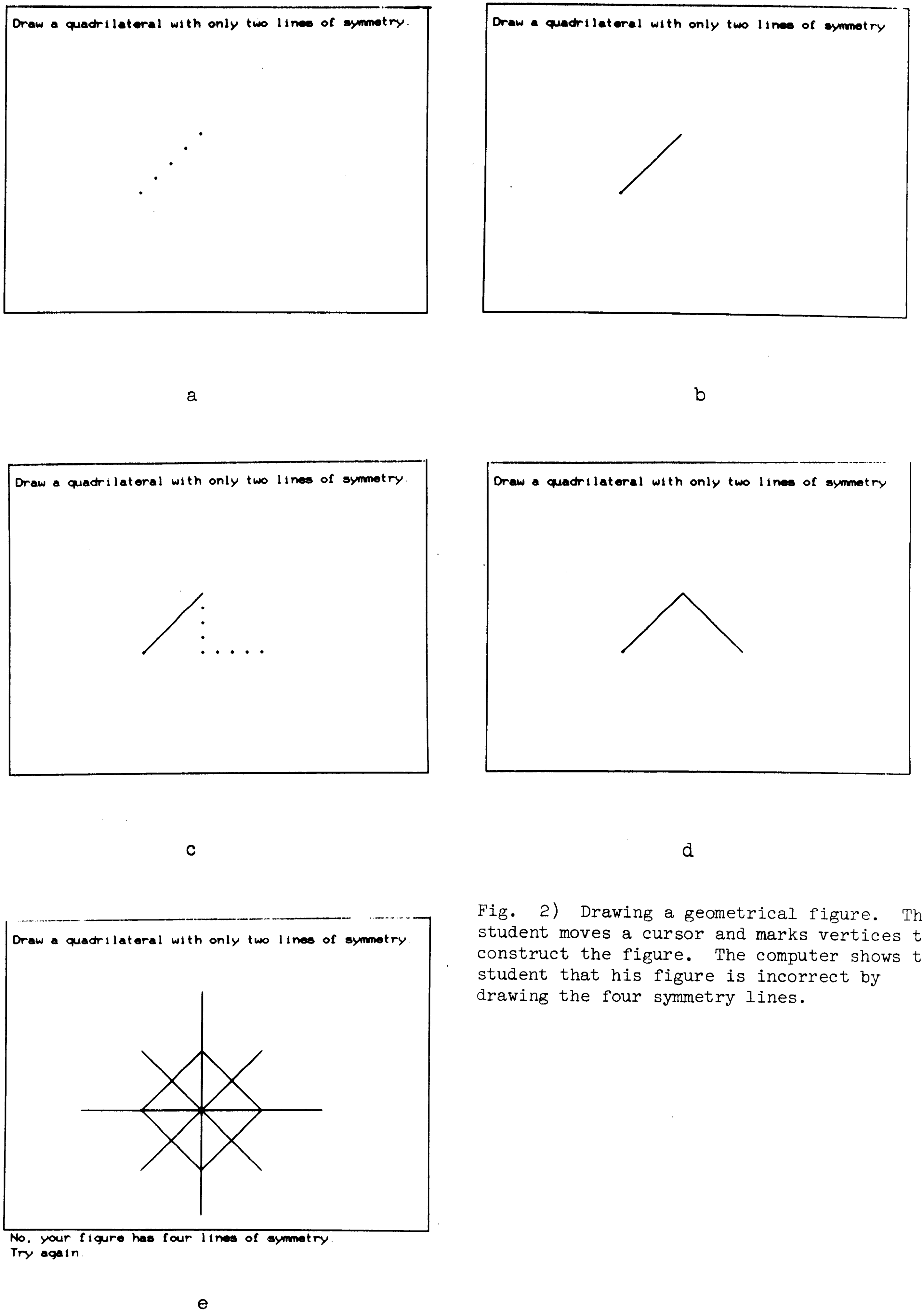
17) Bork, A. M., "The Computer in a Responsive Learning Environment---Let a Thousand Flowers Bloom," and M. Monroe, "Physics Computer Development Project---Computer Assistance in Student Problem Assignments," both in Proceedings of a Conference on Computers in the Undergraduate Curriculum, Dartmouth College, (1971). The Project has produced many internal reports in addition to publications.

18) Feurzeig, W., G. Lukas, P. Faflick, R. Grant, J. D. Lukas, C. R. Morgan, W. B. Weiner, and P. M. Wexelblat, "Programming-Languages as a Conceptual Framework for Teaching Mathematics," in four volumes, Report No. 2165, Bolt Beranek and Newman Inc., Cambridge, Massachusetts (1971).



If you want to use any of these offspring, you must save them now What do you want to do?

Fig. 1) Fruit fly genetics. Some of the offspring have white eyes and/or vestigial wings not seen in the parents. The student records the observed characteristics in his notebook. **This is a half-size negative photograph of the student's display screen.**



Draw a quadrilateral with only two lines of symmetry.

a

Draw a quadrilateral with only two lines of symmetry

b

Draw a quadrilateral with only two lines of symmetry.

c

Draw a quadrilateral with only two lines of symmetry

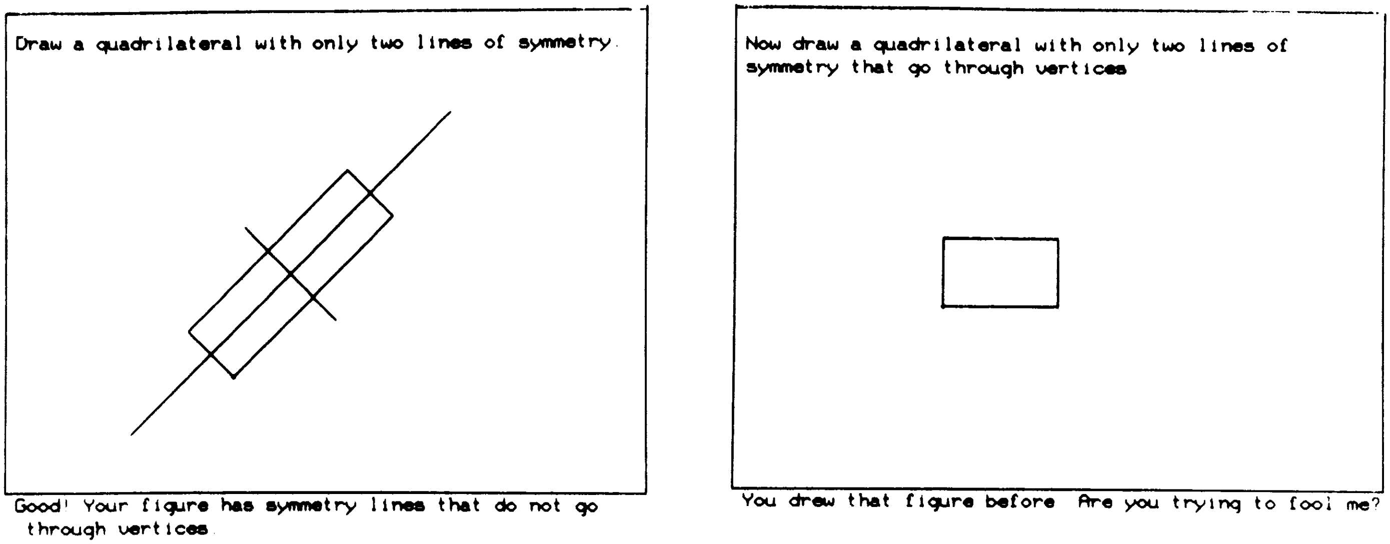
d

Draw a quadrilateral with only two lines of symmetry.

No, your figure has four lines of symmetry.
Try again.

e

Fig. 2) Drawing a geometrical figure. The student moves a cursor and marks vertices to construct the figure. The computer shows the student that his figure is incorrect by drawing the four symmetry lines.



Draw a quadrilateral with only two lines of symmetry.

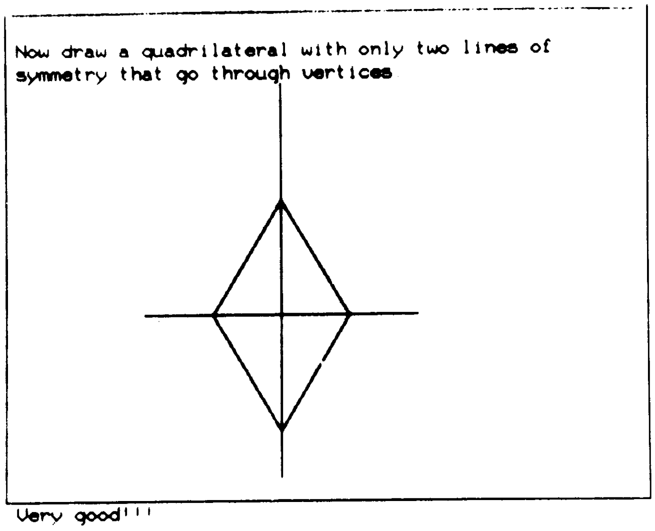
Good! Your figure has symmetry lines that do not go through vertices.

a

Now draw a quadrilateral with only two lines of symmetry that go through vertices

You drew that figure before Are you trying to fool me?

b

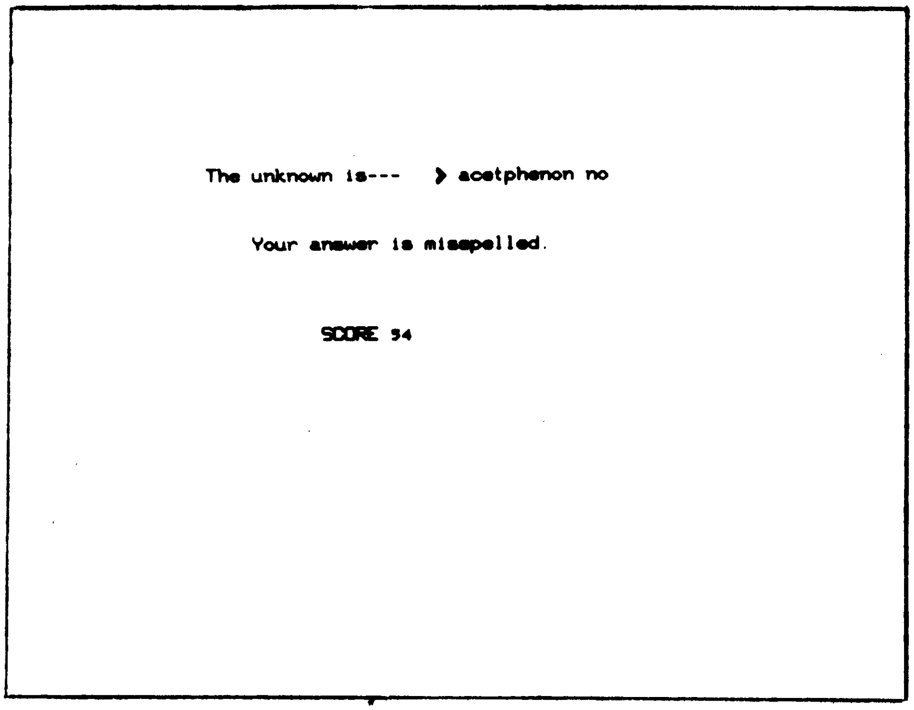


Now draw a quadrilateral with only two lines of symmetry that go through vertices

Very good!!!

c

Fig. 3) Constructing prescribed figures. After succeeding in the first problem the computer chooses an appropriate second problem. The student then incorrectly draws a figure of the same type before finally constructing the desired rhombus. Note that the computer is able to classify the figures independent of size, shape, or orientation.

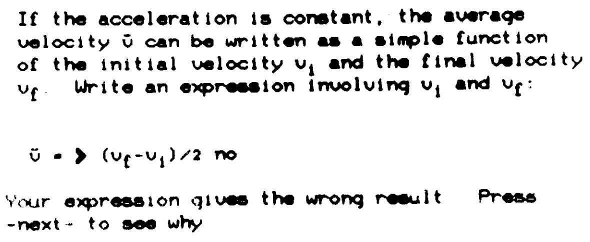


```
The unknown is---   > acetphenon no

    Your answer is misspelled.

          SCORE 54
```

Fig. 4) Qualitative organic analysis. The student has correctly identified the compound but has made a spelling error. The score measures the student's efficiency by charging for chemical tests performed on a scale commensurate with the expense of the test in the laboratory.

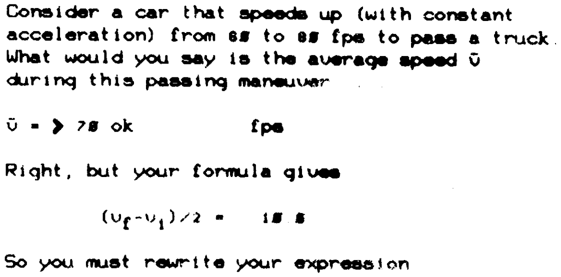


If the acceleration is constant, the average velocity $\bar{v}$ can be written as a simple function of the initial velocity v_i and the final velocity v_f. Write an expression involving v_i and v_f:

$\bar{v}$ = ❯ $(v_f-v_i)/2$ no

Your expression gives the wrong result. Press -next- to see why

a



Consider a car that speeds up (with constant acceleration) from 60 to 80 fps to pass a truck. What would you say is the average speed $\bar{v}$ during this passing maneuver

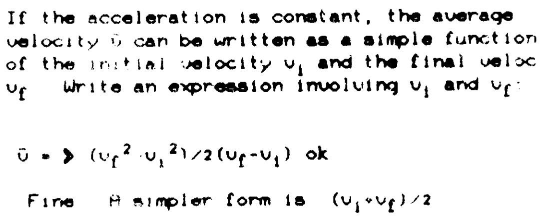
$\bar{v}$ = ❯ 70 ok fps

Right, but your formula gives

$(v_f-v_i)/2$ = 10.0

So you must rewrite your expression

b



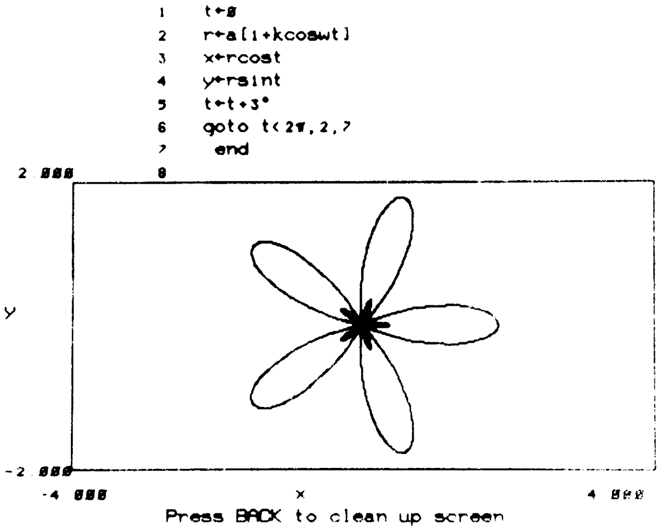
If the acceleration is constant, the average velocity $\bar{v}$ can be written as a simple function of the initial velocity v_i and the final velocity v_f. Write an expression involving v_i and v_f:

$\bar{v}$ = ❯ $(v_f{}^2-v_i{}^2)/2(v_f-v_i)$ ok

Fine. A simpler form is $(v_i+v_f)/2$

c

Fig. 5) Kinematics formulas. The student is shown by example why his formula is invalid. He then gives a valid expression and the computer points out a simpler form. These judgements are made by algorithm, not by searching lists of possible answers.



```
1    t←0
2    r←a[1+kcoswt]
3    x←rcost
4    y←rsint
5    t←t+3°
6    goto t<2π,2,7
7     end
8
```

Press BACK to clean up screen

Fig. 6) Student programming. A mathematics student writes a program to plot a polar function. The angle is "t". On a separate display page the student specified the plotting variables and their bounds and initialized a = .75, k = 1.5, and w = 5.

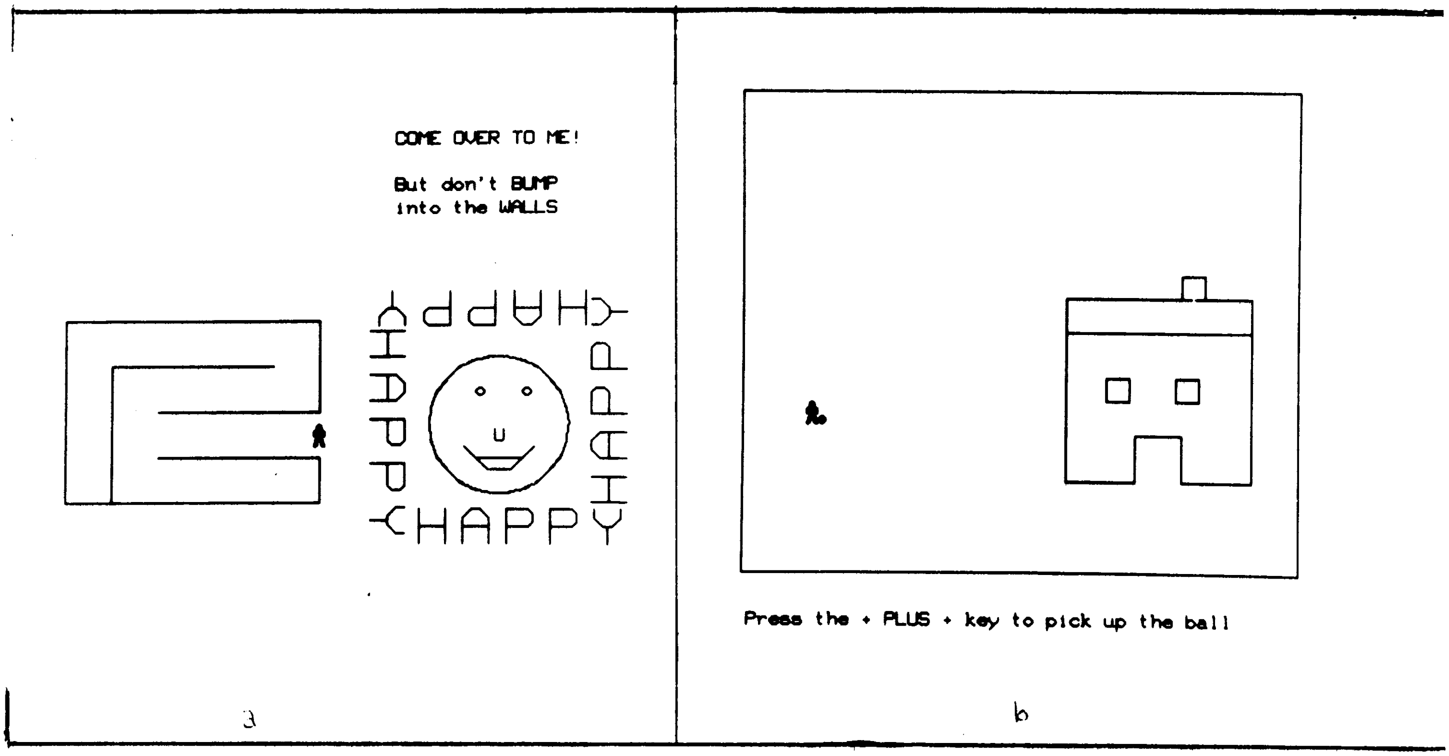


Fig. 7) Games leading to programming by children. The child walked the man through the maze starting from the lower left corner of the maze. Next the child learns to pick up a ball and carry it into the house.

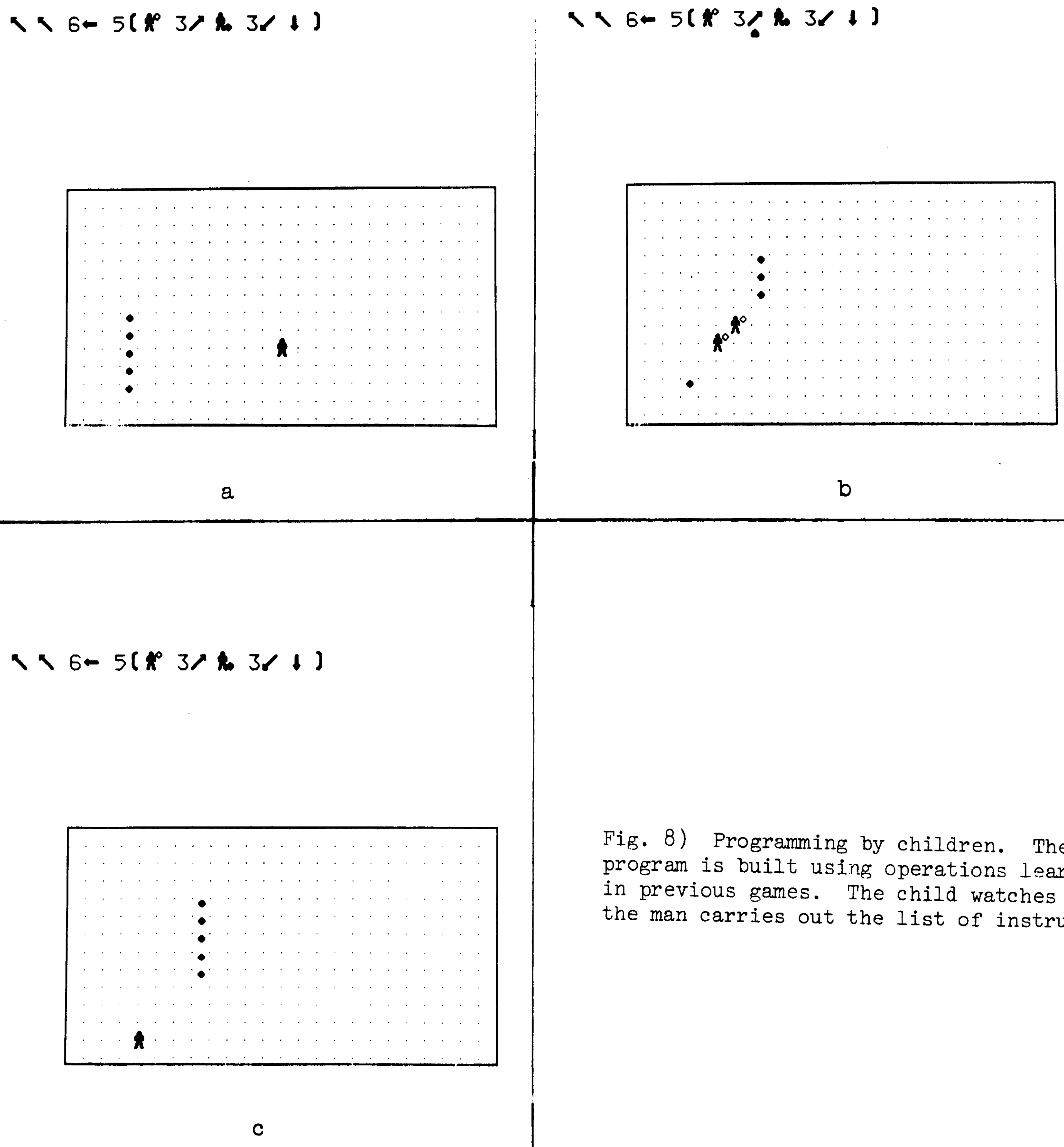
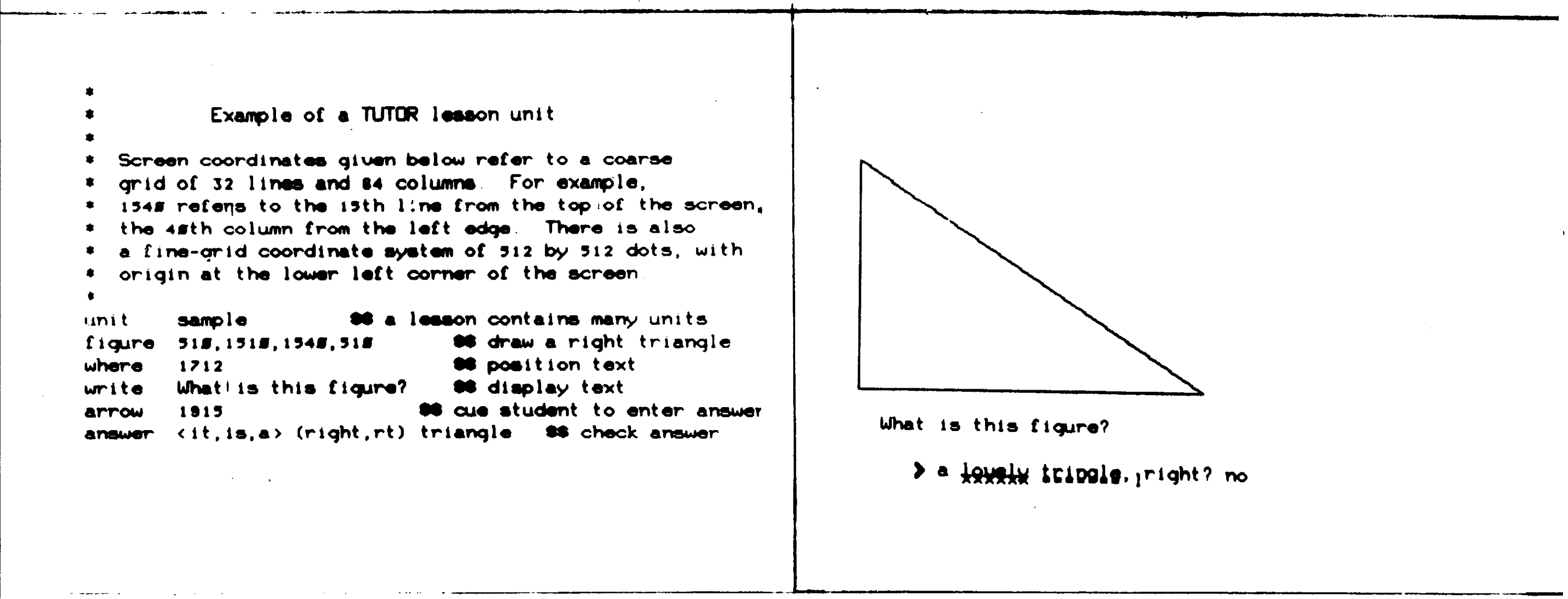


Fig. 8) Programming by children. The program is built using operations learned in previous games. The child watches as the man carries out the list of instructions.

Appendix B

The TUTOR Language

All PLATO materials are written in the TUTOR author language, which was originated by Paul Tenczar in 1967. Fig. 9 gives a simple example of TUTOR programming and its use by a student. Note that the lesson author did not list "tringle" as a possible misspelling of "triangle": the misspelling was detected by the algorithms of the TUTOR "answer" command. In addition to the simple display and judging commands illustrated here, TUTOR has a large repertoire of display, judging, calculational, and branching capabilities which makes possible the complex lesson examples described earlier in this article. Because TUTOR is a full language, not a format for administering standardized items, authors are not restricted to a particular pedagogical strategy or presentation mode. (In fact, some authors have even constructed TUTOR lessons which administer standardized items drawn from a structured data base, so this capability is also available.)



```
*
*         Example of a TUTOR lesson unit
*
*  Screen coordinates given below refer to a coarse
*  grid of 32 lines and 64 columns.  For example,
*  1540 refers to the 15th line from the top of the screen,
*  the 40th column from the left edge.  There is also
*  a fine-grid coordinate system of 512 by 512 dots, with
*  origin at the lower left corner of the screen
*
unit    sample              $$ a lesson contains many units
figure  510,1510,1540,510        $$ draw a right triangle
where   1712                     $$ position text
write   What is this figure?     $$ display text
arrow   1915                  $$ cue student to enter answer
answer  <it,is,a> (right,rt) triangle   $$ check answer
```

```
What is this figure?

  > a ~~lovely~~ tringle, right? no
```

Fig. 9) The TUTOR author language: a lesson unit and how it looks to the student. In the "answer" command the words "it," "is," and "a" are specified to be unimportant, and "right" and "rt" are to be considered synonyms. The reply to the student is much more than a simple "no"--- the word "lovely" is crossed out, the misspelling is underlined, and the word "right" belongs to the left.