

CHIP 98

November 1980

BF
21
C3
V.98

Personal computers and education:
The challenge to schools

James A. Levin

Laboratory of Comparative Human Cognition
University of California, San Diego

and

Yaakov Kareev

Goldie Rotman Center for Cognitive Science
School of Education
The Hebrew University of Jerusalem

This paper would not have been possible without the dynamic support provided by the Laboratory of Comparative Human Cognition or the financial support of The Spencer Foundation. Thanks to Margaret Riel, Denis Newman, Naomi Miyake, Bud Mehan, Tom Malone, Esteban Diaz, Michael Cole, and Marcia Boruta for comments on earlier drafts.

Copyright © 1980 James A. Levin and Yaakov Kareev

Center for Human Information Processing
University of California, San Diego
La Jolla, California 92093

Abstract

Personal computers raise major challenges, both opportunities and threats, for the current educational system as grassroots developments involve inexpensive computers in education in new and often unexpected ways. To explore the implications of these technological innovations, a variety of uses of computers in education are described along with some recent observations of computer use in a range of settings (school, club, home). Techniques for systematically designing educational environments containing personal computers are outlined, illustrated by some examples of using computers in a wide range of educational domains. Finally, some actions are recommended to meet the challenge.

INTRODUCTION

For at least fifteen years, people have been claiming that computers are about to revolutionize education. In 1966, Suppes wrote that, "One can predict that in a few more years millions of school-children will have access to ... the personal services of a tutor as well-informed and responsive as Aristotle. ... This role [of computer-assisted instruction] is scarcely implemented as yet, but ... it cannot fail to have profound effects in the near future." (Suppes, 1966, p. 204) So far, however, there are very few children, if any, who enjoy the privileges of Alexander of Maecedonia through the use of a computer. Why, then, should you continue to read this article, which also argues that revolutionary changes lie ahead of us, due to the impending impact of personal computers on education at all levels?

Previous claims were made on the basis of possible changes that could occur; the changes we describe reflect changes that are occurring. The changes envisioned in the past required that someone (the federal government, private foundations, state governments or local school districts) commit large amounts of money to bring computers into the classroom. The changes described here are based to a large extent on grassroots developments, brought about by the people involved at the basic level of education: teachers, students and parents. Finally, earlier approaches expected the educational effects to occur within the school, and at least implied that some of them may result in the replacement of human teachers by computers. However, more profound effects of personal computers on education may occur outside of schools.

Our predictions about the impending impact of computers on education

are based on the simple observation that it is easy to design highly motivating computer activities with clearly identifiable educational components. The flexibility of computers can assure a wide choice of activities to satisfy many tastes and ability levels. The activities themselves and their contents can be designed to achieve a wide range of educational objectives.

While the potential for intensive use of computers has existed for a long time, the high price of computers or the continuing expense of maintaining communication with a remote central facility proved prohibitive in many cases, and almost always required some benefactor to support the project. Recent advances of microelectronics have drastically reduced the price of computing power and thus reduced the dependence on outside funding. Many parents have taken the initiative themselves to buy personal computers for their homes -- there are even now hundreds of thousands of personal computers in American homes (Flanigan, 1980), and their number is increasing at a rapid rate. Moreover, personal computers are also finding their way into classrooms, sometimes without the official sanction of authorities as teachers bring personal computers in their classrooms on their own. This is one route by which computers are finding their way into schools.

Students attending a university course taught by one of us (JAL) during the spring of 1979 were asked to assess the extent to which computers were being used in the schools of San Diego County. One group of students talked to the school administrators responsible for educational innovation for the county system, and returned with the view that computers were not being used much in classrooms and further that there were no plans for a major expansion of that use. Other students talked to classroom teachers and returned with a different story. The teachers reported a substantial and expanding use of computers in classrooms. How can we reconcile these two accounts? While it is true that the school system as an institution was not

expanding its use of computers very much, it turned out that many teachers were bringing their own personal computers into their classrooms. The price of computers has now dropped to a level where teachers can afford to buy their own computers and integrate them in whatever way they want into their classrooms. This is a grassroots development, in which teachers are taking the initiative and the control over introducing computers into their own classrooms. It is difficult to get a measure of the magnitude of this development, both because it is recent and rapidly changing, and because any statistics collected by the official authorities would be suspect, as personal computers are often introduced into the classroom without the official involvement (or even knowledge, in some cases) of school administrations.

The other half of the grassroots development is the increasing number of personal computers in the homes of students. There are currently more than 1.25 million personal computers, with estimates of 600,000 to be sold in 1980 (Flanigan, 1980). In addition, there are an increasing number of educational programs, marketed directly to owners of these computers. This development is occurring outside the scope of the institutionalized educational process, but the impact of a substantial number of students working with computers in the home will soon have to be acknowledged and adjusted to by schools.

What we see, then, is the spread of a technology which offers a range of highly motivating activities for a decreasing cost. Such a development cannot fail to have a profound effect on education, both in and out of schools. As a result schools face the challenge of integrating computers into their structure to harness their tremendous educational power.

In this paper, we will begin by describing some of the earlier attempts to use computers for education. Then we will outline the range of activities available with current systems, both to explore the range of possible educational uses of personal computers and to gain some insight as to why

computer activities can be motivating. We will suggest a model of educational design for educational computer activities. In the final section we will recommend some actions those involved in education can take to deal with the challenge posed by personal computers.

COMPUTERS AND EDUCATION -- A HISTORICAL SKETCH

Drill and Practice

The earliest yet still most widely used mode of computer assisted instruction (CAI) is the "drill and practice" format. The computer presents instructional material, then asks the student a series of evaluative questions. Drill and practice CAI is an automated version of programmed instruction text books. Both of these were a reaction against the conventional instruction in classrooms, designed to allow personalized instruction and rapid feedback. Depending on the student's answers, the CAI system moves on to new material, asks further diagnostic questions, or presents remedial material. The following is an example of drill and practice (This example is taken from a program called STATES developed by the Minnesota Educational Computing Consortium. The computer prints the words on a computer screen; the student types on a keyboard, and his words are underlined here.):

What is your name? Jim
Lansing is the capital of? Michigan
That's great, Jim!
Charleston is the capital of? South Carolina
No, try again. North Carolina
No, Charlestown is the capital of West Virginia.
The capital of Alaska is? Juno
That's close, Jim.
The exact answer is Juneau.
The capital of Alaska is Juneau.

Number tried	3
Number correct 1st try	2
Number correct 2nd try	0

Early efforts to develop computer assisted instruction were largely oriented towards drill and practice. The Stanford project (Suppes, 1966; Atkinson & Wilson, 1969) developed and tested nationwide systems for arithmetic and reading in the mid to late sixties. The Plato System at the University of Illinois (Bitzer & Skaperdas, 1970) went the furthest in providing a supportive environment for teachers to create drill and practice programs in a subject area.

However, none of these early efforts at CAI have been widely adopted. One of the major factors for the lack of acceptance has been cost. Both the Plato system and the early Stanford system operated on large central computers, with users connected to it by phone lines. The central computers were much too expensive for a school to buy, and even the terminals for these systems were relatively expensive. But the major barrier was the continuing communication costs. Even schools which could raise the startup costs for terminals and training found the continuing expense of phone lines to Illinois or California an intolerable burden.

There was a second major problem with drill and practice approaches. The type of activities provided by them were often un motivating and sometimes downright boring. When given a choice, students prefer a computer game to a straight drill and practice program (Malone, 1980). Lack of variability and similarity to other school tasks are certainly among the reasons for this effect, but a major reason seems to be that the initiative for action in drill and practice environments is almost entirely under the control of the computer program.

Student-initiative Uses of Computers

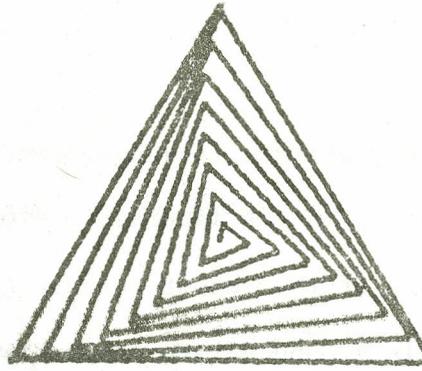
A second major line of development for using computers in education rejected the lack of initiative typical of drill and practice approaches. Instead it created environments in which students had the initiative for actions. In the late sixties, at the time that the "free-school" movement was at its height, Papert and others at M.I.T. started the LOGO Project. Rather than trying to come up with a better way to conduct conventional classroom activities, the LOGO Project provided an environment where learning was achieved through the active process of programming the computer. While drill and practice programs concentrated solely on the acquisition of conventional subject matter, the LOGO approach viewed both learning to program and the acquisition of subject matter as important aspects of computer based activities. Learning to program emerged as a central goal both because it was regarded as an important skill in itself and because it was hoped that it would help develop general problem solving skills. The LOGO Project and others adopting this same educational philosophy explored the potential of computers as a new medium for learning, rather than just as a way to do the same old things more efficiently.

The work of the LOGO Project was named after and based on LOGO, a computer language that children could easily learn and use. More importantly, they restructured knowledge domains (for example, geometry (Papert, 1971), music (Bamberger, 1972), and physics (Abelson, diSessa, & Rudolph, 1974)) so that children using LOGO could easily explore and master them. This restructuring of knowledge makes it relatively easy to write programs that produce interesting results. Below is a program written in LOGO that produces on the computer screen the graphic figures to the right.

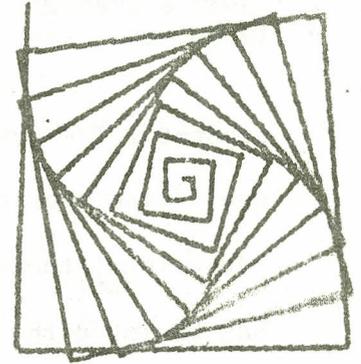
```
TO POLYSPI :STEP :ANGLE
```

Levin & Kareev
12 November 1980

```
1 FORWARD :STEP  
2 LEFT :ANGLE  
3 POLYSPI :STEP+5 :ANGLE  
END
```



POLYSPI 5 121



POLYSPI 5 93

To use POLYSPI one has to supply it with two numbers: a value for the variable STEP, and a value for ANGLE. The program instructs a mechanical turtle (or its equivalent appearing on a TV like screen) to move forward STEP units, then turn left ANGLE degrees. At that point it calls itself again (i.e., recursively) with a new value for STEP (STEP+5) and the same value for the turning angle. Children can learn LOGO and write programs such as POLYSPI, which produce a lot of action for relatively little effort.

Another project sharing LOGO's view of computers and education was the Learning Research Group, started by Kay in the early 1970's at Xerox Palo Alto Research Center. The project set out to provide users in general and children in particular with a powerful tool for fostering learning through active exploration. They developed a computer language for learning, called Smalltalk (Kay, 1977; Learning Research Group, 1976), providing tools for easily producing high quality graphics, animation, and music. The Xerox Learning Research Group was among the first to explore the possibilities of using personal computers for learning, even when the notion of computers inexpensive enough for individuals to own seemed a pipe dream.

Mixed Initiative Uses of Computers for Learning

The two approaches discussed so far can be seen as occupying two ends of a spectrum of underlying educational philosophy. In the drill and practice uses, the computer takes and maintains the initiative in interacting with the

student. The LOGO and Smalltalk languages were designed so that the learner would take the initiative in the interaction to use the computer as an educational tool. Much of the recent research on using computers in education has explored the middle ground between these two extremes, where both student and computer can take the initiative during the interaction. These "mixed initiative systems" (Carbonell, 1970) perform Intelligent Computer Aided Instruction (ICAI). Carbonell's Scholar program (tutoring students in geography) and more recent systems draw upon the methods developed in research in psychology and computer science for representing knowledge, comprehending language, and performing inference.

A prime example of a mixed initiative computer program is the Sophie system developed by Brown and his associates (Brown & Burton, 1975; Brown, Burton, & Zdybel, 1973) to train people to repair complex electronic equipment. The student is presented with a simulated malfunctioning electronic device, and is to discover the malfunction and repair the device. Within this system, the student can make simulated measurements and "replace" suspected components.

Part of the Sophie system builds a representation of the student's current state of knowledge, based on what information the student has gathered and the electrical laws the student has previously mastered. The system can then give guidance to the student as s/he proceeds through the diagnosis process. The computer builds a model of what the student could know about the circuit based on his/her previous actions, and guides the student when s/he makes redundant measurements, generates faulty hypotheses about the trouble, or asks for help. Below is an example of a dialogue between a student (given in Mixed Case Letters) and Sophie (given in UPPERCASE letters). (R22 is a particular resistor in the circuit.) In this dialog, both the student and the computer take the initiative at different points.

What is the resistance of R22?

THE MEASURED RESISTANCE OF R22 IS 25 OHMS.

Replace R22.

R22 REPLACED.

NOW YOU SHOULD TAKE MEASUREMENTS TO SEE IF ANYTHING
ELSE IS WRONG.

Now what is the output?

THE OUTPUT VOLTAGE IS 19.9 VOLTS.

Another example of the use of AI in CAI is the Why system of Collins and Stevens (Collins, 1976; Stevens & Collins, 1977). This system contains a representation of some of the processes embodying Socratic tutoring, in which a student actively learns by answering a series of "leading" questions.

This metaphor of the computer as tutor is based on an educational philosophy in which the computer is neither purely pedant nor purely tool, but instead a jointly active participant in interaction with the learner. A similar metaphor for a mixed initiative interaction sees the computer as a "coach", giving guidance to a learner who is practicing his/her abilities in some domain. The coach metaphor was explored by Goldstein (1977), who designed a program to coach players of a cave exploration game called Wumpus. To play this game well, the student has to learn how to integrate various kinds of uncertain information. The Coach system starts with a complex representation of the Wumpus game and typical student problems, and builds up a representation of the particular student being coached.

This taxonomy of educational uses of computers parallels the more general philosophical positions that have been proposed for education.

Kohlberg and Mayer (1972) describe three "streams of educational ideology": cultural transmission, romanticism, and progressivism. There is a nice mapping of the three positions along the "initiative" dimension (computer-initiative, student-initiative, mixed-initiative) described previously and each of these three general positions. But unlike Kohlberg and Mayer, we are not arguing for the inherent superiority of mixed-initiative systems over computer-initiative or student-initiative systems. Instead, we shall argue for dynamic-initiative systems, that initially provide strong support for novices, and that allow students to acquire more and more control over the domain.

COMPUTERS IN SCHOOL, CLUB, AND HOME ENVIRONMENTS

What are the implications of this dynamic support hypothesis for structuring computer learning environments? We have observed the use of computers by children in three different environments (schools, clubs, and home) in an attempt to discover how children actually interact with personal computers, and to see if a dynamic support notion could allow us to design environments that were fruitful for learning.

Our work itself has been carried out as part of a research project involving problem solving in natural situations; computers were chosen since their use constituted a novel activity where users were likely to face some problems (Levin & Kareev, 1980). We regard the mastering of a new system (the computer in this case) as an important problem solving skill. We are also interested in observing what benefits children might gain from interacting with different computer activities. Some of these activities involved commercially available programs, while others employed programs developed by us. Our observations of children interacting with computers have strongly affected our

thinking, so we will first give an account of them. Following that account we will present our suggestions for how computers can be introduced and used for education.

In all three environments, the computers were relatively small, stand-alone machines consisting of a typewriter-like keyboard and a television-like display screen as the major means for interacting with the computer. (We had Apple II Computers in the school and home environments; Apple II Computers, a PET Computer, and a Terak Computer in the computer club). Except for the PET they all had a disk drive which helped children quickly load new programs. The Apple II computers could use both BASIC and Pascal computer languages; the Terak used Pascal, while the PET used BASIC.

Computers in the Classroom

We were particularly interested to find out how easy it would be to integrate the computer into the normal flow of classroom activities. The computer was placed in a fourth grade class for two week-long periods (Quinsaas, 1980).

The most striking observation was that the introduction of the computer into the classroom did not disrupt the other classroom activities. This was true despite the great deal of enthusiasm and interest surrounding the computer. One fear often expressed is that unless each child in the class has a computer, the other class activities will be hopelessly disrupted. Since this was not true in this classroom, we feel that the way the computer was introduced by the teacher deserves further description. On the first day the whole class was presented with the computer. The available activities were described as well as the steps needed to operate it. In addition the teacher put a sheet of paper detailing the most frequently used commands next to the

computer. It was then placed in a corner of the classroom and defined as an "activity center". The children in this classroom normally rotate through different activity centers during the course of the day, so the computer was added as just another one of them. Each child was assigned a time slot, and the schedule was written on the blackboard so that everyone could see it. From then on, the allocation of time was administered by the children themselves - they watched the clock and claimed their allotted time without the involvement of the teacher. We feel that both the equitable allocation of time and the self regulation principle contributed to the smooth integration of the computer into the class routine.

The children were allowed to work by themselves or to choose a friend with whom to work during their turns. Almost all children selected someone to work with them. Because of this, each child got to spend more time with the computer. But more important, pairs of children working together substantially reduced the number of problems encountered requiring outside help. When the children worked in pairs, a large percentage of the low level problems encountered by one child were immediately solved by the other. For example, in many programs the computer will not act on a typed input until a key labeled RETURN (or ENTER or ACCEPT) is pushed. Inexperienced users often forget to hit this key; they sit there waiting for the computer to respond with increasing frustration. Previous solutions to this problem involved making the RETURN key larger, or coloring it red, or having the computer time a response and inquire the user when too much time elapsed. This problem, along with many other such low level problems, disappeared when children worked in pairs, as one child almost always pointed out the problem or acted to solve the problem as soon as it emerged.

Since teamwork and cooperation helped resolve most of the low level problems encountered, the demands on the teacher's time were minimal. In the

relatively few cases when the pair of children could not resolve a problem, often another child, an "expert" on the particular computer activity engaged in, would come to the rescue. Only in rare circumstances was the teacher's help required.

Educators often express the fear that the introduction of computers into the classroom will adversely affect peer interaction. An often invoked image is that of children sitting before computer consoles, isolated from other children. Our observations showed that computer activities, when properly organized, lead to a considerable increase in cooperative peer interaction, more so than almost any other classroom learning environment.

Throughout our classroom experiments children have showed no signs of "computer anxiety". They related to it as naturally as they would to a telephone or a television set. This stands in marked contrast to the anxiety experienced by many adults in their dealings with computers. Lack of anxiety was true for both boys and girls. Typically there is a gross imbalance in the representation of the two genders in computer related activities. Indicators such as attendance of computer courses, ownership of personal computers, or subscription to computer hobbyist magazines all show approximately a 9:1 ratio of males to females. However we observed no discernable gender differences in amount of motivation or involvement. There was, however, a gender related difference in which activities the children preferred (also found by Malone, 1980). While boys were more attracted by competitive "action" games, girls often preferred games involving writing or music. By providing a wide variety of activities and allowing the children to choose among them, both girls and boys were able to find computer activities that they enjoyed.

The wide variety of activities we provided led to the emergence of "local experts", children who became very adept at performing one activity or another. It was particularly encouraging to observe cases in which those

experts were children who are among the "low achievers" in the class. In two noteworthy examples, a roadrace game and the operation of a lemonade stand, the best performers were two children who were ordinarily at the bottom of the class. Their expertise was recognized by the other children, who came to ask them for advice. We feel that it is very important for any child to excell in some school related activity. The wide variety of activities available on computers greatly increases the chances that most children will indeed be able to achieve that status.

Computers in Clubs

In between the formally structured learning environments of schools and the informally structured ones of homes, there are semi-formally structured settings that children and adults participate in, such as clubs and other voluntary groups led by adults. These are settings where the use of computers is becoming increasingly widespread. One aspect of both clubs and homes that differentiate them from schools is the more flexible interactions allowed. Another difference is that computer activities in clubs and homes must compete with a wide range of alternative activities. Unless the computer activities are sufficiently entertaining children will not engage in them. Entertainment value has long been recognized as an important ingredient for learning environments. However, in school settings it often plays only a secondary role (often accepted only as a necessary evil). In contrast, in less formal environments, entertainment provides the initial motivation necessary for the success of any educational undertaking.

In an attempt to find out how computers could be used for learning and problem solving, we set up two computer clubs for ten year olds, who voluntarily attended over the course of four months. During each club meeting

children were free to engage in any of the computer activities available. A more detailed account of the club activities appears in Levin and Kareev (1980).

As in the school setting, there was a large amount of cooperative interaction among the children in the clubs. Even though most of the computer activities available in the clubs had been originally designed with one user in mind, they were most often used by groups of children. This was true even though the computers were an abundant resource during club sessions (in fact it was not uncommon to see a deserted computer in one end of the room, with several children gathered around another). By working together children were able to divide up the work among themselves thereby dealing with the complexities of new situations in an efficient way. For example, in a game called Harpoon, which required the players to enter two pieces of information, one group smoothly divided up the task so that one child took responsibility for determining one aspect while two others determined the other.

We observed a recurring progression in the way children engaged in computer activities during the clubs. First, when a new computer activity was introduced, a large group of children (four or five) would gather around as an adult demonstrated it. Then one child would claim a turn and sit at the keyboard. This child would often serve as a typist, entering the suggestions of other children and the adult. Next, children would begin to interact with the computer without adult participation, except when they sought help. As the children acquired expertise and the novelty of the activity wore off, the group would grow smaller, often with just pairs of children working together. An expert child might then begin to work alone, increasing the difficulty level of the task when possible. This progression from low-level performance to expertise recurred almost every time we introduced a new computer activity to the club. Once a child became an expert at a particular activity, other

children would turn to him or her for help rather than call an adult. Since there was a relatively large number of activities, expertise was widely distributed among children.

Computer games were the most popular kind of activity in the club. However, most of the games required the practice of skills likely to be useful outside the microworlds of the games. For example, in the Harpoon Game, the children tried to "throw a harpoon at a shark" on the screen. To do so, they had to specify its X and Y coordinates. The harpoon flew to the spot specified, either hitting the shark or splashing into the water. This highly entertaining game provides practice at estimating the number corresponding to a position on a number line, a basic skill for mathematics. The ability to embed basic skills and knowledge in entertaining computer activities allows education to move into less formally structured environments. It also raises the possibility that education in formal environments may be made more motivating without sacrificing its other goals.

Computers in Homes

In an attempt to gain some insight into the issue of how computers are and can be used for education in homes, one of us (Y.K.) conducted a "diary study" of the use of a personal computer in his home. The study involved his two boys (aged 7;6 and 6 at the beginning of the study) and the results reported span a period of seven months. With a computer at home, computing suddenly becomes a free rather than scarce resource. That fact often changes the style of interaction with the computer, as illustrated by one episode involving the two boys.

They had been using a sketching program to create drawings on the computer screen, which they then saved for later viewing and modification. One day,

they had a drawing displayed on the screen, an assortment of dots scattered about. They huddled in front of the computer and carefully moved the cursor (a dot that can be moved to indicate the current position for drawing) from one dot to another. The boys then ran from the living room into their playroom, where they played for a while. Then they ran back to the computer, carefully moving the cursor over next to yet another dot, and again ran to their playroom. This sequence was repeated several times. Finally, when asked what in the world they were doing, they patiently explained that they were playing "Startrek". With the computer as "control room panel", they were "warping" from one star to another, then "beaming down" to the planet to explore (in their playroom).

This "Startrek control panel" example is a good illustration of how computers can be integrated into a larger environment by the child. It is a use that we, the authors, would not be likely to invent, as we have been conditioned to think of access to a computer as a scarce resource. With the computer available on a continuous basis, the boys treated it much in the same way they would treat any other object used for play.

There was a progression of usage in this home, that we believe characterizes a reasonable prediction about usage in other homes as well. Initially there was a large degree of enthusiasm about a variety of computer games available on the computer. Playing was typically intense, but the boys would spend relatively short periods of times with any single game as they were eager to sample others. That pattern changed after a number of days, with longer periods of play devoted to a single game and with much emphasis placed on setting personal records in games where scores were provided. Game playing remained popular throughout the period, but different games were popular at different times. Sometimes the boys would return to games they had not played in months. These cyclical patterns of game playing are very similar to what

one observes with non-computer games, and serve as another indication that children treat the computer as another toy.

The computer neither completely displaced all other play nor did interest disappear as the novelty wore thin. Instead, the boys continued to use it across the seven months of this study, while their patterns of usage shifted. After the initial game usage phase, they also became interested in graphics and music programs. A high point of this usage pattern came several months into the project, when they helped produce a picture and a song appropriate for the Hannukkah holiday. At about this time, each child received his own "floppy disk" to store his pictures on, and a little later they were also allowed for the first time to start the computer and change disks on their own. With that permitted, they became much less dependent on adults in their use of the computer, and could use it even with no adult around.

At this point, they were introduced to a text editor program, which they could use to enter and modify text easily. They enjoyed using this screen editor (the UCSD Pascal Editor) for laboriously entering stories they made up, but mostly for entering "crazy stories" -- text entered by hitting keys randomly on the keyboard. Editing commands to delete, change, and insert text were very popular with the children.

While there was a strong temptation to try and teach the children some programming, we made a conscious decision not to force the issue. They were told that the different activities they were engaged in involved the use of computer programs, and were presented with programs via the use of the 'LIST' command in BASIC. They were then shown how programs could be modified -- initially to change the values of some parameters they did not like in some of the game programs. For example, in a game called Space War, they asked for an increase in the amount of energy allocated by the original program, so they could play longer. In a game involving the operation of a lemonade stand they

asked for a different mix of rainy, sunny, and hot and dry days than that originally available. Only after becoming experts at operating the computer and familiar with low level notions of programming did the children start writing simple programs, for example, ones that filled the screen by printing out strings of characters over and over again.

The computer had an important side benefit: it served as an important aid for language learning. In the beginning, the need to read computer messages and program names, and to type in answers to play games served as a strong motivation for coming to grips with English (a second language for the boys). Later, the text editor and story programs helped them refine their language skills.

Finally, after almost seven months of playing computer games, creating computer graphics and music, editing text, and modifying and writing programs, they came to know one more use for the computer. They were trying to figure out how much money they had earned from baby sitting so that they could plan for an upcoming toy shopping expedition. They were surprised when told that they could use the computer to calculate that amount. They entered the numbers and operations within the BASIC 'PRINT' statement ("PRINT 6.25 + 2.37 - 1.31"), and were pleased to see the computer print out the answer. When told the computer was also similar to a calculator, one of the boys said "No, it's much better than a calculator, since here you may correct your errors".

What we see, then, is that with the computer at home the children engaged in a large number of educationally relevant activities. Not only did they learn how to operate a new system, but they also improved their performance in numerous games (we shall discuss the educational value of such games in the next section), and improved their language skills. Probably most important of all, they became aware of the wide range of uses to which the computer can be put. Finally, it should be noted that the computer did not

become an all encompassing preoccupation with the boys. Interactions with it stabilized at an average of about one hour a day. That time seemed to come mostly at the expense of television viewing time.

We have observed the use of personal computers in three different environments: in a classroom as part of ongoing school activity, in a computer club, and at home. The three environments varied along a number of important dimensions. One was the availability of computing resources: the computers were a scarce resource in the classroom, and the computer club meetings were relatively infrequent. At home, on the other hand, the computer was continuously available, leading to a different style of use. The environments also differed in the kind of interaction between participants. At school, children could consult with some but not with all their peers, and expert advice was not readily available; in the club and at home children could freely interact both with peers and with any expert adults present. Both variables were important in determining the kind of emergent behaviors.

The three environments also had some common features. First, the computers provided much initial support in all three of them. Many of the activities were computer games with the setting and rules determined by the computer (programs), with the participants being explicitly informed about the range of activities available to them. More creative activities such as story making or sketching were also selected with special attention to the availability of clear and easy-to-follow instructions concerning the range of possible actions.

Secondly, children were free to choose any of the available activities. At any point in time they could invoke any program and switch between them. Needless to say, within any activity they were free to determine any aspects dependent on the user.

EDUCATIONAL DESIGN

Preliminary Considerations

In this section we examine the educational potential of computer activities and suggest ways for designing them. Our approach to educational design rests on two related assumptions concerning the educational potential of an activity: (1) The educational potential of an activity is directly related to the individual's interest in engaging in that activity. (2) The educational potential is maximized when the activity is within the individual's "zone of proximal development".

The role of motivation in bringing about learning cannot be overestimated. Children can be forced to engage in activities they do not like, but unmotivated or frustrated children are unlikely to achieve the intended goals of the activity. Educators are well aware of the need for motivating activities; once educational goals are set, much effort is devoted to the question of how to design motivating activities to achieve these goals.

A related question has to do with the kind of activities which are likely to be of maximum benefit to the learner. Some activities may be of minimal value since they enhance a capacity already mastered by the student. Other activities may turn out to require capabilities not at the individual's disposal. Activities of the later type are too difficult and lead to frustration on the learner's part. In between already mastered skills and too difficult tasks lies the zone of proximal development. It includes those activities which the individual cannot perform alone but is capable of performing with some help from the outside environment.

The notion of the zone of proximal development was developed by Vygotsky (1978). The basic idea is that development results from an interplay

between the individual and the environment. The individual possesses certain faculties, and they increase in power as a result of the individual's interaction with the environment. There is a certain region which is beyond the individual's reach when operating alone, but which can be mastered with outside help. The help might provide additional information, point to similarities, differences, or regularities, or indicate what should be done next. The range of conditions within which such help can be utilized by the individual is the zone of proximal development. The zone depends on the individual in question, and changes with the mastery of new faculties. Learning can be optimized if activities are so arranged that the individual operates most of the time within his/her zone of proximal development (Laboratory of Comparative Human Cognition, 1979).

The concept of zone of proximal development provides a powerful tool for the design of educational activities. The basic idea is not new, of course, and similar notions about the design of a supportive environment for the learner pervade the educational literature. Indeed, a basic assumption underlying the use of teachers and curricula is that such a zone exists, and that activities within it can enhance the achievement of educational goals.

The zone of proximal development is not of a fixed size or location, but is dynamically defined by the amount of needed support available in the outside environment, with the appropriate support leading to a wider zone. While support is necessary during the initial stages of achieving new educational goals, the end result often calls for the learners to be able to achieve independent mastery of the subject matter in question. Therefore, an important aspect of designing educational activities is to build into them a mechanism for gradually withdrawing support, thus letting the individual become increasingly independent in his/her performance. At the same time that mastery of certain performances is achieved, other materials which draw on the newly

acquired capacities are introduced, again in a supportive environment. Support systems should be tuned to take into account the special needs of the individual learner. The design, introduction and withdrawal of supportive systems are major challenges to educators. Individualized instruction can be viewed as a way of producing just the right amount of support for the learner.

We regard the suggestions presented below as valid principles for designing computer-based educational activities both when the users have a computer for themselves and when the computer is a scarce resource to be allocated between a number of users. However, the prototypical situation we have in mind is one where both the computer and human expert support are limited. While the prices of microcomputers have dropped at an astounding rate, economic considerations make it unlikely that schools which decide to integrate computers into their activities will be able to acquire a computer for each child in the immediate future. Our observations lead us to believe that this economic reality may be a blessing in disguise since it is not necessary, and possibly not even desirable, for each child to have his/her own computer. Expert advice is likely to be another limited resource: except for a few well financed experimental projects we expect expert adults (teachers, club leaders, parents) to be able to devote only part of the time necessary to support the new users. Here, too, we do not regard the scarcity of the resource as a debilitating limitation; in fact, there might even be ways of taking advantage of this situation. The design of computer activities should take these constraints into account, though.

Progression Towards Expertise

In this section we present a more detailed account of how related

activities may be combined to provide an environment for progressing to expertise. We start by showing what kind of activities might be necessary for novices to become familiar with a computer and to learn how to program. We have chosen this example since quick mastery of the computer system may be a prerequisite for successful interaction required by many different activities, while programming knowledge provides the students with a powerful tool with which they can further explore the newly acquired knowledge. An example is presented that demonstrates how a simple and seemingly useless game can be the entry point for a large number of educational activities on and off the computer.

Introducing a Computer System

There are two levels of skills involved in using a computer, operating a computer and programming it. Current educational uses of computers vary as to how much knowledge of these two aspects they demand of the users, or expect them to achieve. At one end of the spectrum, some educational applications of computers (drill and practice, for example) only require that the user know how to run an existing program, and not much else. At the other extreme, there are approaches (e.g., LOGO) which put much store in the educational benefits of learning to program. These approaches require both familiarity with the system at hand and knowledge of a programming language.

We believe that learning to program is an important eventual goal of using computers in the educational process, and that as many users as possible should acquire at least some knowledge of the skill. Yet, programming is a complicated skill, which requires mastery of the syntax of a new language which is quite different from natural language. Furthermore, without having a good grasp of the operating system of the computer, any programming efforts will be of little use. For example, suppose one learned some of the rules of a programming language, and wanted to type in a program. One is immediately

confronted with a host of problems, unnoticed by experienced users but debilitating for a beginner. How does one terminate a statement? What about typing errors? What does one do to make a program run once it is typed in? Suppose one likes the program; what should be done so that it can be used on some later occasion? And if the program is stored somewhere for future use, how is it called back? Any one of these problems can frustrate the novice, or even destroy his/her enthusiasm for interacting with the computer. All of these problems have easy solutions, but they depend on the system used and have to be learned. Operating manuals describe the solutions, but they often make for dull reading, and in any event it is difficult to remember all the commands, options, and solutions described. Immediate expert advice is of great help in such situations, but is likely to be scarce. Furthermore, expert help will be under the heaviest demands by each individual user at the same time, shortly after the computer is introduced. Thus, approaches which call for immediately teaching novices to program may result in frustration and resentment when put into use in realistic everyday conditions.

One way to ease novices into full-blown use of computers is by initially employing simple computer activities which can be easily used to produce interesting results. Many different kinds of such activities are currently available, but all of them have a number of features in common. First, the computer provides the environment in which the activity takes place. This environment consists of graphic or verbal material presented on a television-like screen, and often accompanied by sound effects. Second, the user is actively engaged through the use of some input device(s) such as a typewriter-like keyboard, game paddles, joysticks, or a lightpen. In addition to providing the environment, the computer almost always serves as a record keeper, provides prompts for action when necessary, and modifies the environment in response to the user's actions. In many activities, the

computer also plays the role of another active participant.

Our observations of children interacting with computers at home, in the classroom, and in clubs all clearly indicate that such computer activities are easy to learn and highly motivating. There are, of course, some which are too difficult for children at certain ages, and some are downright boring. Yet, the variety of existing programs (there are literally hundreds of commercially distributed computer games and other activities) assures the availability of interesting and easy to use activities for beginners of all ages.

Computer activities such as these are a good introduction to the computer for a number of reasons. First, as noted above, people enjoy operating them. Thus motivation to use the computer is achieved in a natural and effortless way. Secondly, since appropriately chosen computer activities are easy to use they put minimal demands on the scarce resource of an expert's time. Novices can operate the computer themselves when all they have to do is run existing programs. In other words most of the necessary initial support is provided by the computer itself. Thirdly, computer activities provide initial familiarity with the operating system. While interacting with the computer the users become familiar with the outlay of the keyboard and with how it is used to type in answers. They learn the functions of the special keys such as "return," "backspace," "delete," "escape," or "reset." Once they become bored with one activity they have to learn how to use the computer's mass storage device (cassette player, disk drive) to load in another program. All this knowledge will be very useful when the users learn to program. And it is acquired without strain in a play-like context. Finally, such activities should not only be viewed as a way of introducing novices to programming. Engaging in computer activities can be of great educational value by itself. The activities can be designed in such a way as to present new materials as well as to require

the practice and development of a large number of skills.

Thus, by starting with computer activity programs one can produce users who not only enjoy interacting with computers, but also know how to do it by themselves. They are likely to know how to run programs, to enter data and modify variables at run time, to correct typing errors, and in general be aware of some of the capabilities of the computer as well as its limitations.

After a number of weeks of working with activity programs the users can become skillful at operating the computer. By that time, different people will also have developed different preferences. Individual differences assure enough variation in the preferences for even the most popular activities (Malone, 1980). Even better, after a number of weeks some people will become frustrated with some features of their favorite programs. They may start complaining about them, and with little or no prompting may start wondering about how they may be changed. It is easy to rouse the curiosity of novices by providing a few variations of the same program ("Why are there sounds in this car race but not in the other?"). Dissatisfaction with existing programs and the realization that they can be modified sets the stage for the next phase in introducing computers, the phase of modifying existing programs.

The modification of existing programs serves a number of useful purposes. First, as a phase along the way towards learning how to program it can introduce novices to concepts necessary for programming. A program cannot be modified without knowing some of the programming concepts involved in the modified section. Thus the novice can become acquainted with basic programming tools such as input or output statements, assignment of values, computations, testing conditions and branching, arrays, loops, etc. Now, not every person in a group will learn about all these concepts, but each can master some of them by the time they are ready to do their own programming. As a result each will

be able to contribute something to the group. Moreover, the learners will acquire those concepts which will be relevant for them at a given point in time, when the particular concept helps them overcome a problem they face. These are probably ideal conditions for learning.

The acquisition of basic programming tools and distributed expertise are only part of the potential benefits of a program modification stage. As they have to integrate their modifications into larger programs learners can learn about program structure and design. Their section of the program will have to interact with other sections of it, and they will have to at least partially understand what those other parts do and how they operate. Their sections are almost certain to contain both syntax and logical errors (bugs), thus providing basic training in the art of troubleshooting and problem solving in the computer environment. All this will happen in the context of relatively small changes, which will increase the likelihood that the tasks will be accomplished with minimum pain, or even with pleasure. While debugging a large program is a painful operation, detecting and fixing the last bug in it is always a very rewarding experience; by having learners work on small changes chances are that their bugs will always be "last" bugs, thus minimizing their frustration.

A continuing high level of motivation is a final advantage of this intermediate stage. The learner will certainly be motivated to participate in a project stemming from their own curiosity and interest in improving a computer activity. But there will also be another motivating force providing support for the project. Recall that almost all the learners in a group will be exposed to the activity being modified. They will be able to evaluate, as well as appreciate, any improvements made in it. Thus we can expect social support for projects undertaken by the learners.

Since other learners in the class are likely to be familiar with the

activity in question, the learners involved in modifying any of the programs can discuss their ideas with other members of a group, incorporate new suggestions, and take criticisms into account. Different learners working on similar projects can compare ideas and products, as well as get into larger scale problems involved in integrating their changes of different parts of the activity into one "super-improved" activity. The ability to plan, design, criticize, take criticism into account, or take another person's point of view are all useful metacognitive skills (Brown & DeLoache, 1978), and have been suggested as some of the prime cognitive abilities underlying intelligence. Indeed, the acquisition of meta-cognitive skills has often been mentioned as one of the expected benefits of teaching programming (Papert, 1971). Our claim is that many of these skills can also be acquired (at least to a degree) during the program modification phase. Some, like the ability to take another person's point of view or the ability to modify a course of action as a result of constructive criticism may even develop better during this stage, since it is not clear that novices are very good at criticizing and monitoring their own activities. The modification of existing programs can provide novices with practice in all of these skills.

At this point learners would probably be ready to move to the next phase, programming on their own. Calling it a phase may be a misnomer, since independent programming may come about gradually, with the learners rewriting longer and more complicated segments of programs, which will eventually become independent programs. Here an adult expert can help by showing how certain segments can stand on their own, or be incorporated in a number of different programs, thus emphasizing the independent nature of the learners' products. At a certain point the instructor may do well to point out that some learners have written complete and independent programs, which will probably be a pleasant surprise to the new programmers, as well as a source of inspiration to

their peers. When the learners seem to be close enough to the independent programs phase the instructor have a number of suggestions for programs. These can be brought up with discussions with learners who are ready to proceed in programming but who might seem at a loss for a topic.

The beauty of programming is that most of its benefits will happen almost independently of the actual project involved. Questions of design and planning (ease of usage, flexibility, error free performance), problem solving involved in the actual writing of the programs, and the benefits of debugging will all occur with any project. Many learners will have, of course, their own ideas for programs they would like to write and test out. In such cases the instructor should gracefully (and gratefully) step aside, and resign him/herself to the role of a consultant. With programming activities naturally evolving on the basis of prior experience and actual needs we do not expect learners to lack ideas for projects or to undertake projects which are too grandiose (two problems frequently encountered by current approaches which attempt to introduce learners to programming right from the start). With programming activities emerging in a gradual way we also expect a proliferation of group projects, based on shared interest in common problems. Such projects will further familiarize learners with problems of teamwork, division of labor, and questions of coordinating the work so that separately produced segments fit together. All these are useful skills which are rarely practiced in most educational settings.

Note that a high level of motivation and a sufficient amount of support are ever present concerns in the scheme just described. The novice learns about the computer system by engaging in self-selected computer activities; basic programming notions are acquired through attempts to make the activities even more interesting; full blown programming evolves from such fixes, or comes about as an integral and useful part of other group activities. During the

program use phase the novice is almost completely supported by the computer program itself, with low level help provided by peers, written instructions, or, if all else fails, the local experts. During the second phase the structure of the to-be-modified program provides most of the necessary support, as the learner changes some parts and retains others. Finally the learner is on his/her own, with relatively little support from the computer. Social resources (peers and adults) are, of course, an ever present source of support, to be used by the learner as s/he pleases.

Lateral Educational Design

What we have been discussing so far is vertical educational design: how to help novices progress to expertise in a particular domain. But what if learners do not want to proceed to more advanced phases? What if they only want to play games, thus "completely wasting" their time? In this section we describe some of our ideas concerning how the computer can be integrated in the larger framework of educational activities. We do this in connection with game and other play-like activities, which probably seem most threatening and least relevant for the achievement of traditional goals of education.

One common concern is that the introduction of computers into the classroom will be a distraction. Certainly, when computers are introduced they are novel, attractive, and curiosity arousing. In the first couple of days following their introduction we will have to let the children explore them, and be able to use them for longer periods of time. It will help, though, to initially introduce the computers with only few games available on them. This reduces confusion among the children, and their desire to sample everything available will be confined to only few programs. New computer activities should be gradually introduced. We have tried this approach in the classrooms