

## MICROCOMPUTERS AND THE HANDICAPPED – EXTENSIONS OF THE JOHNS HOPKINS SEARCH

The University's role in personal computing to aid the handicapped has included projects of national and regional scope—the Johns Hopkins Search and the current development of an authoring system for individualizable special education instruction. Earlier and continuing efforts at APL also include projects to help people adapt to or circumvent physical and intellectual limitations. In the latter context, this article relates two recent pilot projects with several intellectually limited and learning-disabled students in two Maryland schools.

### INTRODUCTION

Thinking back on the personal computer revolution, I recall that during the 1977 Baltimore City Fair APL exhibited a prototype 16K microcomputer, programable in Basic, controlled by a Radio Shack keyboard, and housed in a walnut case built at the APL model shop. By November 1980, when the Johns Hopkins Search for Applications of Personal Computing to Aid the Handicapped was announced, Americans had bought nearly 200,000 personal computers. Recently that number was at the 7 million mark and soaring.

We have already seen three generations of personal computers and, before them, a number of specialized and large computers applied to problems in rehabilitation and computer-aided instruction. The Johns Hopkins Search and subsequent applications of the technology cover, in general, the second generation—from the TRS-80 Model I/Apple II to the IBM PC. This article relates to that period, which parallels my introduction to personal computers and my experience with two of them in the classroom while working at The Johns Hopkins University as an M. A. Tuve Fellow during the 1982-83 academic year.

### EVENTS SURROUNDING THE NATIONAL SEARCH

At APL, a team headed by Paul Hazan, a pioneer in microcomputers, has been active in the educational, special educational, and rehabilitative uses of personal computers since 1979. An educational television series on computer literacy, titled *Adventure of the Mind*, which was co-produced with CTI, Inc., was seen by more than one million high school students. In 1980, funding was obtained from the National Science Foundation and the Tandy Corporation to launch the Johns Hopkins Search.<sup>1</sup> Additional support was received from Apple, Atari, the Institute for Electrical and Electronic Engineers, the Association for Computing Machinery, computer clubs, and a host of other organizations in the fields of computing and rehabilitation.

During the Johns Hopkins Search, we benefited from a strong alliance with faculty of the University's Division of Education and the John F. Kennedy Institute led by Gilbert B. Schiffman, Professor of Education. His team was just then introducing microcomputers into its courses, practicums, and pilot programs for disabled students.

Prior to the Johns Hopkins Search, my experience in special education stemmed from working with my third child, now a sturdy 19-year-old, whose intellectual limitations are secondary to Down's syndrome. To better understand and help remedy his disability, I completed a Master's degree program in communicative disorders in 1978. Having worked at APL in public information and personnel for nearly 30 years, I also spoke computerese, albeit with a slight accent. The opportunity to serve on the Johns Hopkins Search team and to spend an additional year on the University's Homewood campus and at the John F. Kennedy School<sup>2</sup> broadened my experience in computers and education and provided new incentives as well as practical lessons.

The value of interdisciplinary cooperation in developing, testing, and matching computers to the learning strengths and styles of students with special needs cannot be overemphasized. Three years ago, few special education teachers knew anything about computers or computer programming. Sizable numbers of computer professionals and amateurs had formed a talented but largely ineffective pool of help for the millions of handicapped individuals who eagerly awaited developments that could assist them in communicating, learning, and operating more effectively at home and in the workplace. Only a few members of the disabled and computer communities were properly allied to do something of significant value for individuals with handicaps. The question of how to make a broad assault on the problems of our nation's 30 million disabled children and adults remained. Paul Hazan envisioned harnessing people's inventiveness in applying personal computers to rehabilitation problems on a national scale. The ubiquitous and inexpensive micro-

computer would become a standard building block in solutions to aid large numbers of individuals with similar needs.

The numbers of disabled individuals by category (e.g., learning disabled—5 million, and mentally retarded—4 million), when viewed as potential buyers and users of standard microcomputers along with peripheral devices, now could be presented to manufacturers of computer equipment as a very substantial market. Several winning entries in the Johns Hopkins Search have succeeded in the marketplace. Nearly all were products of interdisciplinary cooperation between special educators or rehabilitation specialists and computer scientists or programmers.<sup>3</sup>

### THE NEXT STEP: COMPUTERS IN THE CLASSROOM

Following the Johns Hopkins Search, APL's Richard Henle, who is also a member of the computer science faculty of the Johns Hopkins Whiting School of Engineering, proposed extending some of the computer's communications possibilities as an aid to non-vocal students classified as severely or profoundly mentally retarded.

The resulting programs and special interface for the Radio Shack color computer that he and David Ashman fashioned were dubbed "MASH," in keeping with the fact that the program requires special switches to be "mashed" by the student (with apologies to the Maryland Association for the Severely and Profoundly Handicapped). Figure 1 shows the MASH computer configuration as it was set up for teacher/student use.



**Figure 1**—This is the classroom MASH setup. I sat in front of the keyboard with the students, their switches, and attractors to my right. The switches connect to an input/output device, which is the larger black box behind the blue switch. This special APL-built interface provides robotic speech via a computer program, a synthesizer (mounted below the table to reduce television interference), and the small black speaker on the top shelf. The disk drive and its program-pack interface share the Color Computer's single programmable read-only memory port. The teacher can activate the switches from the keyboard, and either the student or teacher can make them light up and ring like a bell.

Before MASH, examples existed of microprocessor programs and special interfaces that could be used by severely motor-impaired individuals as electronic communication boards for speaking or transmitting words, symbols, and sentences. Most of the microprocessor programs and special interfaces involved scanning or direct pointing to pictures and words on a computer screen under the control of special switches or joysticks. Several provided an output of the selected words or ideas as synthesized speech. Henle felt that the effort involved in creating graphics on the screen, even with the powerful drawing commands of the color computer's extended Basic, was inefficient and limiting in terms of the number and clarity of images that could be created and stored.

My own bias in teaching students of low measured intelligence is toward presenting concrete objects rather than abstract representations of the materials and ideas being communicated or learned. Why use books or even full-color pictures when we can talk about real money, food, animals, etc? Therefore, MASH became a means for expression and concept development in which the student pushes a switch (which lights up and sounds a musical flourish) next to the object that he selects. After he pushes the switch, perhaps positioned next to a real banana, the speaker attached to a voice synthesizer speaks "banana," "yellow banana," "I like bananas," or whatever the teacher decides to have the student, aided by the computer, say about the object. The hardware provides for up to 16 switch inputs. The current software is limited to four inputs. In practice, the five students with whom I worked on ten occasions typically, with varying degrees of success, could only indicate preferences for one object rather than another. Therefore, our lessons often involved only two switches.

Before a lesson, indeed in the middle of a lesson, the teacher would set up a four-switch word menu for each child or use one of the menus stored on disk from the previous session. As prompted on the screen, an input menu would read

|                                 |               |
|---------------------------------|---------------|
| Type your first word(s)         | (e.g., apple) |
| Second word(s)                  | (pear)        |
| Third word(s)                   | (orange)      |
| Fourth word(s)                  | (banana)      |
| What shall we name this program | (e.g., fruit) |

whereupon the program "fruit" is automatically stored. By pressing the "D" (for directory) key, the teacher can immediately access a menu of stored programs. The MASH program shows the teacher which word(s) will be spoken by which switch engagement. The four-word sets appear rectangularly on the computer screen, e.g.,

|          |          |
|----------|----------|
| 1 apple  | 2 pear   |
| 3 orange | 4 banana |

The objects themselves are then matched to the position of the number 1, 2, 3, and 4 switches. The switches can be color-coded with large round stick-on labels.

Finally, it was useful for the teacher to be able to actuate the switches from the keyboard in order to stimulate and reward student responses. All of the students, even those with autistic-like behavior, found it difficult to resist a flashing, flourishing, talking switch.

### MY TOP STUDENTS IN HOWARD COUNTY

The behavior of two students deserves special attention. A boy and a girl, each about six years old, had not developed spontaneous speech, although both had been appraised as being more intelligent than their communications development would indicate. They had no serious emotional or physical problems and had made excellent progress in social and personal skills (e.g., hygiene, manners, self-feeding). The girl spoke one or two words and enjoyed trying out initial consonants. The boy was nonvocal but often indicated desires by leading the teacher by the hand, pointing, or vigorously assenting or denying.

Both were fascinated by the computer's ability to speak their names and to respond with sounds to keys typed on the keyboard (when a simple talking typewriter program was loaded into it). They quickly learned that if one switch was pushed (mashed), one word or idea was spoken, and if the other was pushed, a different idea was spoken. Since after two "correct" responses they were rewarded with the food or object of their choice (e.g., an apple or piece of banana), they reliably chose the correct switch in order to obtain their objective, particularly in the case of the apple and banana. After four sessions involving fruit and several other objects and activities, the girl learned to say the word "apple." Thereafter, when given the choice of an apple or a banana, she would first select the apple, hold it up and say "apple," put it down, and then push the "banana switch" to receive the reward that she preferred. In other words, she arranged to receive two real rewards and much teacher approval. Is this not an example of unusual intelligence in a student classified as severely-to-profoundly intellectually limited?

One of the activities that the students enjoyed was pushing one of the switches and finding out what the computer had to say in response. A representative selection might include a four-statement menu like (1) go playground, (2) go P.E. (gym), (3) take walk, (4) ride bike. The reward for having the computer speak any of these two-word sentences twice in a row was that the teacher would take the student to perform the activity of the student's choice. "Ride bike," which resulted in leaving the classroom and getting a ride in the halls on a shiny red tricycle, became a natural favorite, particularly with one mechanically inclined young boy. His behavior in using the MASH system included a desire to listen to the computer with the hand-held speaker held directly to his ear. Although I might offer a reward after two consistent responses, sometimes I would allow him, and one other student with the same inclination, to listen to the words being spoken four or five times, if they so desired, before providing the reward. He listened to "ride bike" dozens of times. After the third day in which select-

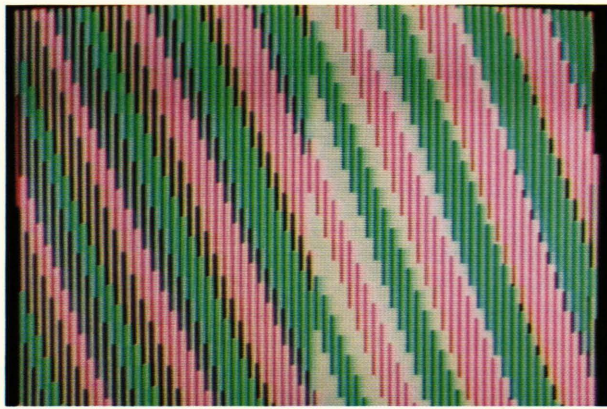
ing an activity such as riding a bike was stressed, he chose "ride bike," listened to it four times, took my hand, and when he saw the tricycle parked outside the room, he excitedly said, "bike" before jumping into the seat. This was the only word I ever heard him speak.

These anecdotal results, from working with students in a classroom of the Cedar Lane School in Columbia, Md., are not presented to suggest that computers offer a new means of encouraging communications or learning or that breakthroughs are apt to occur with any regularity. My limited experience does suggest, however, that computers, like lunchtime, music, and the sandbox, provide an unusually motivating, attention-holding learning environment in which communications and concept development can occur. For the teacher, MASH also provides insight into children's preferred learning styles and underlying strengths and weaknesses.

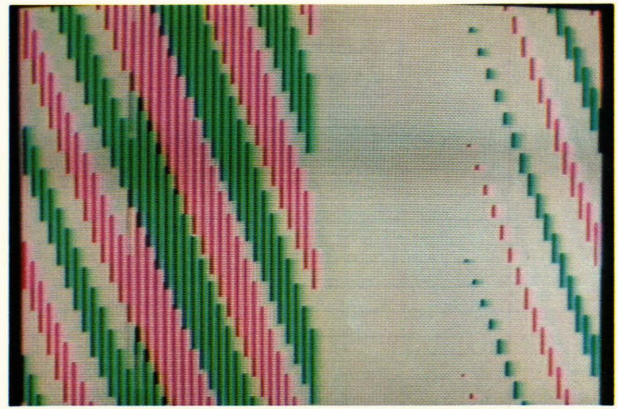
### CREATIVE PROGRAMMING AT THE KENNEDY INSTITUTE

Nearly the same conclusions could be drawn from my other classroom experience during 1982-83. I had the opportunity to work with 10- to 13-year-old, bright (several in the gifted range) learning-disabled students at the John F. Kennedy School. My role was to support the assistant principal with computer-aided instruction and computer programming instruction. He and the other teachers used computers in a variety of ways, including their application to the development of reading and math skills, social studies, and vocational training. However, the class with which I regularly assisted him spent most of the first semester experimenting with LOGO turtle graphics and the second semester learning Basic programming. We met at least once a week for an hour and rarely were less than fully absorbed at the end of the class. The two regular instructors of the 12-member class usually joined us during computer lab. None of us had any previous experience with LOGO. We used the Apple version<sup>4</sup> and usually had five machines available for hands-on work.

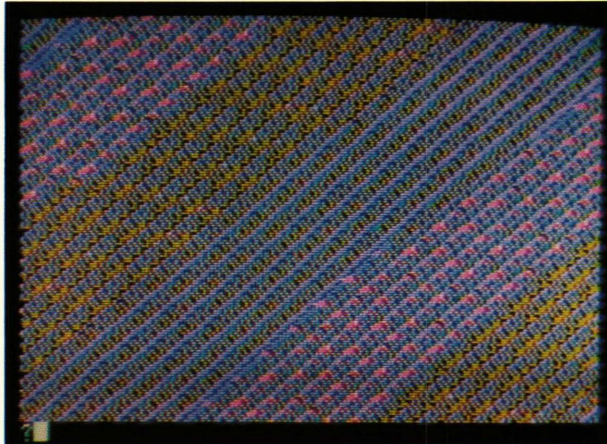
The student teams consisted of either two or three members matched by temperament and potential contribution. The opportunity to provide socialization skills was considered important and, although from a strictly pedagogical point of view two students might be considered ideal, the three-member teams were equal if not greater producers of LOGO procedures. One team consisted of a brilliant strategist who could not be persuaded to worry about details, a worker noted for his common sense and thoroughness, and a good typist who had occasional brilliant insights. Quickly this group and one other leaped ahead of the instructors. Although our reading of the manuals made us better versed in primitive commands and procedures for editing, controlling the borders of the picture, saving and loading programs, and the like, the students were the first to develop original programs—amazing recursions, examples of which are shown in Fig. 2.



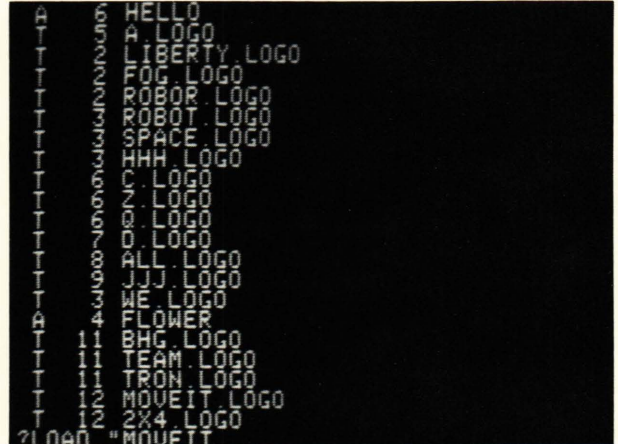
(a)



(b)



(c)



(d)

**Figure 2**—(a) and (b) represent typical intermediate procedures of LOGO graphics saved by students. The procedure titled “Z” in the menu shown in (d) ends with the dense painting shown in (c). As the menu suggests, the students soon nested procedure Z in a still longer and more beautiful graphic “JJJ,” which cannot be shown because it closes with single stars flashing against a black sky.

While we were helping a slower learner draw something interesting on the screen, the advanced groups were creating military tanks and cosmological phenomena that were everchanging in color, position, or phantasmagoric pattern right before their (and, later, our) eyes.

LOGO proved to be a good way to introduce these students to computers. With a relatively small number of mnemonic commands (e.g., RT for right turn, HT for hide turtle), immediate results and pictorial feedback can be obtained. It provides an opportunity to learn the benefits of precise and logical thinking and careful spelling and typing.

LOGO encourages spatial (body) awareness, directionality, and intuitive geometry (how many times must the turtle turn right one degree to be heading in the opposite direction?) and is also a versatile language for writing text on the screen and performing mathematical operations. Its procedural methods provide a good background for Pascal and other powerful languages.

After LOGO, learning Basic was not at first appealing to all members of this class. Individual creativity had to be tethered while a new set of commands was

learned, and Apple graphics in Basic paled beside LOGO's rapidly and easily formed shapes and colors. Fortunately, sound and music were introduced, and the students started to see that math and text operations could be handled more easily in Basic. Built-in functions can immediately calculate sums, products, squares, and square roots. The computer was made to print, sing, talk, count to a million, write quizzes and knock-knock jokes, help keep records, write and correct letters and articles, and even correct spelling. Visions of arcade games leapt into mind. Basic became exciting, too.

Both languages proved ideal and mutually supportive from the point of view of computer literacy. Many procedures (loading, saving, listing, and editing) are similar. Both are powerful, logical, high-level languages that can provide useful or interesting products and learning experiences. And, for learning-disabled students, computers open up an array of career choices. The computer can also be a partner in the process of overcoming disability and can provide compensatory assistance in reading, writing, and calculating. Through its patient insistence on accuracy and logic, it offers close, consistent interaction beyond the

reach of a human teacher in any setting other than one-on-one instruction.

## CONCLUSIONS AND NEW DIRECTIONS

The general conclusion drawn from operational classroom use of 1980-82 microcomputers and programs is that, from the special education teacher's point of view, this technology is both limited and limiting. We need more effective and adaptable programs that we cannot write (or do not have time to write) ourselves;<sup>5</sup> we need computers that support significant advances in graphics, voice technology, and user friendliness. The latter attribute, of course, has not been fully achieved even in such relatively simple tasks as word processing.

At Johns Hopkins, three general approaches have been found to the problem of providing the special education teacher with software that can meet the individual needs of disabled students across the spectrum of disability:

1. Computer education for teachers;
2. Continuing national efforts to stimulate development of teacher-approved, teacher-ready software;
3. Development of a computer authoring system that enables teachers to develop sound and flexible instructional programs quickly without coding.

The first approach is largely the province of the University's Education Division, which has been providing graduate instruction in Baltimore and Columbia and in countless workshops and presentations around the country since 1979. The second, a natural outgrowth of the Johns Hopkins Search, has included the national software search conducted by the Council for Exceptional Children, the nation's largest professional organization in the field of special education. This competition, supported by Johns Hopkins and the Kennedy Institute, attracted 104 entries that were judged for educational merit by the divisions of the Council for Exceptional Children.

Although these two efforts have attracted wide attention and have had considerable impact, the third

approach could provide the revolution required to bring computer-aided instruction to fruition rapidly on a broad scale. Under contract with the federal Department of Education, a team of APL computer scientists, engineers, and programmers has joined forces with the University's special education faculty and graduate students. The goal is an authoring system that will meet requirements specified by teachers and that will be evaluated by teachers every step of the way from concept, through software development, to classroom prototype trials, to eventual acceptance and use by teachers with little or no computer background. On the basis of the total system approach, the Johns Hopkins authoring system project, headed by Paul L. Hazan, has the support of the Maryland school system from the level of superintendent and assistant superintendent down to principals and teachers in schools that have been identified for pilot testing. A thorough description of the educational and technological goals of the system, the system definition, and its over-all design are given in A. Kossiakoff's article, which precedes this one. About a year from now, results of the next chapter in the continuing story of the collaboration effort between APL and the University's Education Division will begin to appear in Maryland's special classes and schools.

## REFERENCES and NOTES

- <sup>1</sup> *Johns Hopkins APL Tech. Dig.* 3, 226-277 (1982).
- <sup>2</sup> Part of the John F. Kennedy Institute, a research and treatment center affiliated with The Johns Hopkins University.
- <sup>3</sup> *Proc. The Johns Hopkins First National Search for Applications of Personal Computing to Aid the Handicapped*, JHU/APL (Oct 1981).
- <sup>4</sup> For younger students, the Terrapin version is recommended. It allows pictures drawn by moving the turtle (cursor) around the screen to be saved as a procedure, i.e., a program that can be saved and brought back. A similar program, "Delta Draw," has simpler commands and the picture-saving feature.
- <sup>5</sup> Incorporating simple procedures or programs within more complicated ones, within still more complicated ones, so that the finished program contains a series of logical but not necessarily predictable events.

**ACKNOWLEDGMENTS**—The author is indebted to Gilbert B. Schiffman, Richard A. Henle, David W. Ashman, the John F. Kennedy School, the Cedar Lane School, and the Howard County, Md., Board of Education for their support of his work with children and computers. Walter G. Berl's editorial contributions are also truly valued.