9361406 /63
PN-AAQ-65
ISN-3592

CENTER FOR INTERNATIONAL EDUCATION

School of Education
University of Massachusetts
Amherst, MA 01003

THE MICROCOMPUTER IN EDUCATION: THREE ACTIVITIES IN LESOTHO

Electronic Aids for Literacy Project

With the Collaboration of

The Lesotho
Distance Teaching Centre

and

The Lesotho
National Curriculum
Development Centre

PN AAQ 656

THE MICROCOMPUTER IN EDUCATION:

THREE ACTIVITIES IN LESOTHO

Electronic Aids for Literacy Project
Center for International Education
University of Massachusetts
Hills House South
Amherst, MA 01003

Prepared for:

Office of Education
Bureau for Science and Technology
U.S. Agency for International Development

1984

PREFACE

This document was produced by the Electronic Aids for Literacy Project of the Center for International Education of the University of Massachusetts under a contract from the U.S. Agency for International Education (AID/DPE-1108-C-00-3062-00).

The document includes contributions from: Stephen Anzalone, Stephen McLaughlin, M.S. Vijay Kumar, and Kathe Conrad.

The activities described were undertaken with the collaboration of the Lesotho Distance Teaching Centre and the National Curriculum Development Centre. A special word of thanks goes to Mr. Kenneth Tsekoa and Mr. Thabo Titso of these institutions for their assistance in organizing these activities.

The Project also wishes to acknowledge the assistance received from the Agency for International Development in Washington and in Maseru, Texas Instruments, Inc., and others. This especially includes the efforts of Ken Dwyer, Barbara Formoso, Julianne Gilmore, and Patsy Layne.

INTRODUCTION

As part of an investigation of educational applications of microprocessor technology in developing countries, the Center for International Education's Electronic Aids for Literacy Project undertook three exploratory activities with microcomputers in the Kingdom of Lesotho. Conducted with the collaboration of the Lesotho Distance Teaching Centre (LDTC) and the National Curriculum Development Centre (NDC), the activities were the following:

1. Conducting for a group of Basotho school children an after-school activity to acquaint them with the language LOGO and observe the possible appropriateness of LOGO as an instructional tool in a developing country.
2. Using a microcomputer with companion software to present to pupils locally-generated spelling lists with words contained in sentences. The effects of this activity were compared to similar activity with the Speak and Spell learning aid, which uses spelling drills without the meaningful context of sentences.
3. Experimenting with an authoring language to devise and test a short instructional module specific for Lesotho.

These activities were conducted in October and November 1983 in conjunction with a twelve-week field test of hand-held electronic learning aids in primary schools. The three activities are discussed separately after the following background information.

Hardward/Software

The equipment used in Lesotho was donated to the Project by the

manufacturer, Texas Instruments, Inc. The hardware was brought to Lesotho and installed on the premises of the LDTC and NCDC. Each institution received the following: a TI-99/4A console; a color monitor; a peripheral expansion box containing cards for 32 K bytes of RAM expansion, a disk drive controller, and an internal disk drive; a speech synthesizer; and one external disk drive. The equipment was obtained in the United States and shipped by air freight to Lesotho. The Project provided electrical current transformers and anti-surge devices.

The activity in Lesotho made use of a variety of software, including: TI LOGO, TI Spell Writer, TI PILOT, and, for demonstration purposes, the TI Touch Typing Tutor, Reading Flight, and pre-release programs for basic skills instruction from TI's PLATO collection. The PILOT activity used the P-Code peripheral and the TI Editor/Filer/Utilities software package.

The Context for Micros in Lesotho

Educators in Lesotho are not unfamiliar with microcomputers. A good deal of interest in microcomputers is to be found, and a number of micros are already being used throughout the country. The equipment coming into the country is obtained from overseas, in the Republic of South Africa, or from a computer retail store in Maseru. Trademarks commonly found in Lesotho are Apple, Osborne, Radio Shack, BBC, ICL, Commodore, and Kaypro.

Microcomputers are a lively subject of conversation among educators in Lesotho as elsewhere. The Science Department at the National University of Lesotho is using micros in instruction and has plans to offer a workshop for secondary school science teachers. Several secondary schools have already obtained microcomputers, and others are making plans to do so. In

November 1983 the LDTC conducted a day long learning workshop entitled, "The Small Computer in Education." The events (described in a brochure in the Annex) were well attended.

LOGO IN LESOTHO

LOGO is the name of an evolving and powerful set of computer languages. Its ease of use, exciting applications, and educational benefits are making it increasingly popular among computer users and educators all over the world. LOGO's development began in 1968 as part of a National Science Foundation sponsored project. A prime contributor to the development of LOGO is Seymour Papert of the Massachusetts Instiutte of Technology's Artificial Intelligence Laboratory. In his book Mindstorms, Papert discusses the LOGO philosophy and its potential contribution to intellectual development.

The specific features of LOGO that render it attractive for education are the following:

- It is a procedural language. LOGO programs are created by combining commands into groups called procedures. These procedures are used as steps in other procedures and build to higher levels of complexity. LOGO allows problem solving by breaking a problem into smaller pieces that are more easily grasped.
- It is interactive: Any LOGO command can be executed or modified by simply typing the command on the keyboard.
- It is powerful. LOGO's lists are functional units that can be transformed in single operations. This makes LOGO a convenient and powerful language for applications involving manipulation of symbols.

The aspects of LOGO that makes it most attractive in terms of its applicability to teach young children are features called Turtle Graphics and Sprites. On a monitor LOGO users create pictures using simple English commands to control a Turtle, which is a representation of a mechanical robot. Turtle Graphics is highly useful both as an introduction to programming and as a foundation for mathematics. With Sprites, the LOGO user brings animated objects to the screen and learns to deal with motion.

Introducing and Testing LOGO in Lesotho

The first experimentation with LOGO in a developing country was conducted in Senegal by a team headed by Papert. Unfortunately, little documentation is available on this experiment. The experiment in Lesotho tried to gain some general insights into using LOGO with school children in a developing country. Through a brief and open-ended activity the experiment in Lesotho tried to answer the following:

1. Are there any immediately observable differences in the way LOGO is learned by children in Lesotho compared to American children?
2. How quickly do Basotho children learn
   (a) the primitive LOGO commands
   (b) LOGO concepts?
3. In independent work, do Basotho children go beyond what is instructed to creative use of the language?
4. What, if any, problems can be observed with respect to
   (a) the English commands
   (b) understanding LOGO concepts?
5. Do Basotho children prefer
   (a) Turtle Graphics to Sprites
   (b) Graphics to non-graphic aspects of LOGO?

Participating Children

The selection of the children to take part in the experiment was done in consultation with the NCOC. Six children (five girls and one boy) all between the ages of 12-13 years were chosen. The selection was based upon age and on the proximity of their residences to LDTC, where classes were conducted. This was necessary because classes were conducted after school and transportation was unavailable.

Instruction

The children received seven sessions of LOGO instruction. This included lectures, hands-on experience and individual projects. Sessions lasted two hours. The pupils were introduced to some basic computer concepts and to the equipment. None had any prior exposure to computers. They were then given a demonstration of the various features and possibilities of both Turtle Graphics and Sprites.

During the first three sessions the pupils were instructed in Turtle Graphics, including the basic commands, editing, and the more complex features such as procedures, combinations of procedures, and recursion. The instruction combined a lecture demonstration with hands-on group work. The children were encouraged to experiment with manipulation of shapes and sizes and helped to solve problems.

The next two sessions were devoted to Sprites. This included introducing concepts of speed and direction. The last two sessions were devoted entirely to hands-on work and independent projects. The pupils demonstrated these projects at the LDTC learning fair mentioned above.

Observations and Results

A rapid and well-grounded understanding of the language and its concepts was observed. This was demonstrated by the pupils generating complex

geometrical shapes, using advanced commands, and exploiting the procedural features of the language by the third session. The hands-on process activities were slowed slightly because only one child was familiar with typing and only one computer was available for this activity.

No language or cultural impediments to learning LOGO could be detected. In fact, one of the researchers, who has had extensive experience teaching LOGO to children in the United States, observed that the Basotho children seemed to learn LOGO at a faster rate than he was accustomed to seeing with American children. This is a subjective and unrealiable response to the progress of a small and possibly unrepresentative sample of children. Nonetheless, it does suggest something about the possibilities for LOGO in a country like Lesotho.

A feedback interview was conducted with the children. The feedback indicated high enthusiasm for LOGO, especially for the Sprites. The high level of enthusiasm was also evident from the fact that the children arrived to work on the computer progressively earlier each day.

The children did not identify any aspect of the language that caused them difficulty. They mentioned a desire to continue with LOGO whenever facilities were available.

SPELLING WITH SPELL WRITER

The microcomputers brought to Lesotho offered the opportunity to compare the instructional routines of electronic learning aids like the Speak and Spell with the more extensive instructional capabilities of the microcomputer. Specifically, it was possible to address one of the concerns raised about aids like the Speak and Spell: that, in dealing with isolated words, they do not provide a meaningful context for learning the words, such as a sentence, paragraph, or story. A micro like the TI-99/4A does have the potential to provide the learner with a wider contextual framework for words presented in an instructional routine. Moreover, the TI-99/4A, like the Speak and Spell, has the facility for using synthetic speech as part of an instructional program. The synthetic speech from the computer's peripheral Speech Synthesizer is referred to as allophonic speech and is produced by stringing together phonemes. The speech in the Speak and Spell is produced through a voice compression technique called linear predictive coding and is substantially more recognizable.

The experiment in Lesotho tried to consider the question: What advantage would a microcomputer's presentation of a locally-generated list of words in sentences have for pupils' mastery of spelling over the more restricted word lists drilled by the Speak and Spell? It was hypothesized that pupils hearing words presented in sentences by the microcomputer would spell more words correctly on an initial try and more words correctly on a second attempt than they would do with words of comparable difficulty asked by the Speak and Spell.

In a test of the effects of the two different presentations of word lists for spelling practice, the six pupils taking part in the LOGO activity served as subjects. All but one of these pupils had completed Standard 7.

Word Lists

The test required two lists of words of comparable difficulty to be "spoken" to the pupils for them to spell. One list was presented by the Speak and Spell through its "Spell It' routine. Here, the device directs the user to spell a word (e.g. "Spell dog"). The second list contained words judged to be "meaningful" for pupils in Lesotho. This list was prepared with the TI Spell Writer program. The words were presented by the microcomputer and accompanied by a sentence with the word in it. For example, the microcomputer might say to the pupil, "Lesotho is a country in Africa. Spell country."

Ideally, the two lists of words would have been drawn from a pool of pre-tested words so as to insure that one list was not more difficult than the other. Resources did not permit taking the time required for pre-testing of words. Instead, a pool of words was taken from spelling lists found in the Rainbow Blue book, a Standard 7 English test written for schools in Southern Africa. The pool contained words like: erosion, continent, blanket, and maize. Fifteen words were put into a word list in a Spell Writer program file. This program allowed the user to enter the correct spelling of a word, a sentence containing the word, and an option to make a phonetic entry to guide pronunciation of the words and sentences.

For the Speak and Spell word list, Level A words in the "Super Stumpers" module were tried. These words are at a level of difficulty such that 50% of all U.S. fifth graders misspell them. The fifteen words pupils were asked to spell were from a pool of thirty Level A words, including parade, vacation, husband, snowy, and wisdom.

An initial trial was conducted in order to get an approximate idea of what level of Speak and Spell difficulty corresponded to Rainbow Blue difficulty. In subsequent trials, the level of difficulty of Speak and Spell words could be adjusted upward or downward. This seemed relatively easy to do, since the level of difficulty of words within and between modules is carefully specified.

Procedures

For the initial trial, the six pupils were presented the fifteen words by the Speak and Spell. They were allowed to press the "Repeat" key if they did not understand the word spoken. The Speak and Spell allows one retry if the word is misspelled. After completing the Speak and Spell test, the pupils were presented the fifteen words and sentences by the microcomputer. If the word is misspelled, the microcomputer spelled the word correctly and allowed two retries (this feature is optional). The difference in the way each device accommodates retries made the tests slightly different.

Results

The pupils proceeded quickly through words presented by the Speak and Spell. Pupils, on the average, had difficulty understanding the pronunciation of two to four words. Repeated pronunciation of the word through the use of the "Repeat" key did not seem to help in recognizing the word. Three of the pupils got more than ten correct on the first try; the others scored three to ten correctly on the first try. The retry did not usually lead to correct spellings.

The pupils did not do as well on the words presented by the microcomputer. Three spelled eight correctly on the first try. The other three pupils spelled only one to three of the words correctly. Almost all of the words were spelled correctly on the first or second retry after seeing the correct spelling of the word.

Although the list of Rainbow Blue words on the microcomputer may have been more difficult and subsequent trails using more difficult Speak and Spell words may have produced different results, the most observable explanation of the difference was the difficulty the pupils had in understanding the microcomputer's allophonic speech. The sentence accompanying presentation of the word seemed to hinder rather than help recognition of the word. Moreover, running through the fifteen words and retries on the microcomputer proceeded very slowly.

There was little reason to continue with subsequent trials. While subsequent trials might have helped to insure that the test words were of comparable difficulty, there seemed no way to disentangle the confounding effects of the poorer quality speech used by the microcomputer. The pupils' expressed their preference for the Speak and Spell presentation and their eagerness to get on with the LOGO activity.

Interpretation

It was not possible to learn what advantages a microcomputer's presentation of a locally-generated list of words, in sentences, had over the more restricted word presentation of the Speak and Spell unless the quality of speech offered by the microcomputer was equal to that in the Speak and Spell. For the moment, the microcomputer did not seem to be a very efficient way of providing opportunities for pupils in developing countries to practice spelling. The advantages of being able to produce word lists for a specific location with the help of a program like Spell Writer were quickly lost with the difficulties posed by the allophonic speech. Better quality speech in a microcomputer would surely exacerbate

the already high cost differential between a microcomputer and an electronic learning aid. The pupils' preference to use the Speak and Spell for spelling and the microcomputer for LOGO was indeed suggestive of a division of labor that would best exploit the relative advantages inherent in each of these two versions of computer technology.

PILOT Authoring Language

PILOT is one of the "authoring" languages that have emerged in recent years to make computers more accessible to non-programmers. Many educators have long wanted to exploit the interactive potential of computer-assisted instruction but were prevented from doing so because they did not know how to program computers. PILOT was developed to circumvent this need to know a complicated computer program language in order to do educational applications.

The original PILOT introduced in the early 1970's consisted of only eight instructions to the computer. Mastery of these commands allowed an educator to write an instructional program in language not much different from everyday speech. The present version of PILOT with its seventeen instructions is more complicated and powerful than the original, reflecting the advancements in computer technology in the last 15 years. Along with adding more capability, PILOT's developers claim to have preserved its capacity to be learned by those with little or no prior programming experience.

PILOT has been adapted for use with both Apple and Texas Instruments microcomputers. According to its developers, TI PILOT is designed "specifically for computer-assisted instruction (CAI)." It can be used to:

1) develop individualized and interactive instructional programs,
2) develop drill-and-practice programs,

3) develop testing programs, and

4) store students' performances and other information.

These applications of TI PILOT can be developed and run on the TI 99/4A using the UCSD p-System.

Both the Apple and Texas Instruments versions of PILOT are versatile systems which feature graphics, varied formatting, color mixing, sound effects, music, moving objects (sprites), and, in the case of the TI version, synthesized speech. Modifying, conditioning and relational functions permit lessons to be specifically tailored to the responses of the learner. Commands can be combined to form more complex instructional programs.

Testing of the usefulness of PILOT for computer-assisted instruction on the TI 99/4A was one of the three microcomputer activities conducted in Lesotho. The purpose of this activity was to determine the feasibility of using TI PILOT to develop and run a CAI module that would teach a specific body of skills and knowledge. This suggested two questions:

1) Could someone who has little or no experience with programming learn TI PILOT instructions well enough to prepare an instructional program, and

2) Would this PILOT module effectively accomplish its learning objectives under actual conditions of use?

Several problems were encountered with the TI PILOT software and with the hardware needed to use it with the TI 99/4A. These problems were serious enough to prevent successful completion of both objectives and the PILOT activity. The problems are discussed below in the order they occurred.

The first step in designing the field test was to select material that would lend itself to CAI objectives. This content was drawn from a manual on bookkeeping and financial management skills for small businesses and

organizations in developing countries.* A section on how and why to keep a cashbook was selected as the specific material around which a TI PILOT module would be designed. While the printed manual would explain basic concepts and methods, the CAI program would supplement these explanations with examples and practice exercises in arithmetic and recording of transactions.

Problems arose when this content began to be inputed into the computer. Programming the module using the TI version of PILOT proved to be exasperating. The limitations of TI PILOT and of the accompanying procedural manuals made the language considerably less user-friendly than expected. Considerable time and effort were required to master enough PILOT to write a relatively straightforward instructional program.

The UCSD p-System (or P-code) needed to run PILOT was one source of difficulty. Creating, editing or storing a PILOT program is done through the UCSD p-System Editor and Filer. These sytems require learning a separate set of commands in addition to the PILOT commands. It was necessary to look back and forth between the p-System and PILOT manuals in order to figure out how to proceed. Attempts by the developers to simplify Editor/Filer commands with such expressions as <ext/eof> to symbolize pressing two keys at the same time often seemed more confusing than simply telling the user what specific keys to press.

The organization of the PILOT manual was another source of difficulty. Procedures are sometimes mentioned but not actually explained until later in the text, leading to errors caused by the user not being aware of the

---

* This manual and selections of text for the module were prepared by June Bourbeau of the Center for International Education. These grew out of work undertaken previously with LDTC.

conditions when the procedure is required. An example of this is the instructions for putting a directory on a diskette in order to save a PILOT program -- a procedure that would ordinarily be needed early in a programming activity. The explanation of this procedure did not come until page 132 of the Appendices.

Some idiosyncracies in TI PILOT were discovered only by trial and error. For example, when the directory of PILOT files is called up, the file name is followed by ".TEXT." But if a file is executed by typing in ".TEXT" after the name, an error message follows. The program automatically adds another ".TEXT" which makes the file impossible to locate. Another bug which slowed mastery of TI PILOT was the tendency for the program to crash without warning when too many lines are added in the Insert mode.

After several weeks of intense work, however, a PILOT program was finally completed, put on a diskette, and subsequently taken to Lesotho for field testing. An appropriate audience of nonformal educators from the LDTC was requested to assess the effectiveness of the PILOT module. It was at this point that the hardware problems were encountered with one of the peripherals for TI 99/4A.

The TI Peripheral Expansion System used in conjunction with the TI 99/4A must be equipped with a P-code card containing the UCSD p-System in order to run PILOT programs. The P-code card installed in the peripheral expansion unit at the test site malfunctioned during the preparation for the demonstration. The computer would either display obvious errors that were not in the program or would not execute the program at all. Efforts to troubleshoot and perhaps correct the fault were unsuccessful and the test of PILOT had to be cancelled. Malfunctions of two other P-code cards on the machine used to

prepare the module suggest some unresolved difficulties with the reliability of this component.

To summarize, unexpected faults in the equipment were the most immediate cause of the failure of the present exercise. Aside from these hardware problems, the version of TI PILOT that was tested showed shortcomings that made it difficult to use and that might limit its appeal. For one thing, the lengthy self-instructional time needed to learn TI PILOT may be unacceptable to many educators. It should be kept in mind, however, that this was a pre-production version which the manufacturer released for testing with the understanding that some problems remained to be resolved.

The inability to test TI PILOT adequately was unfortunate. A system with the expressed intention of broadening access to computers through simplified programming surely deserves careful consideration. It is hoped that there will be continued refinement of TI PILOT for use with existing models of the TI 99/4A as part of this effort to put the power of computers at the disposal of educators.