Software for the People

By Susan Rappaport

THE GAP between the computer "have's" and the computer "have not's" grows as fast as the new technology. A sub-class of information poor will become permanent in our society unless drastic change takes place. This widening space can be partially filled by integrating public access to computers and computing into public library services.

At the New York Public Library, we began a pilot project with the express purpose of putting computers into the hands of people who do not have access to the new technology. With funding by a New York State Outreach grant in February 1983, the library was able to set up sites in six locations providing an Apple IIe, two disk drives, an Amdek color monitor and a dot matrix printer. In addition, approximately 50 pieces of software were provided to be used at each microcomputer site. The locations were selected on the basis of neighborhood and/or age level need, and were located in Harlem (Countee Cullen branch), the Lower East Side (Tompkins Square), the demonstration branch for teenagers (Donnell—Nathan Straus), in the Bronx (Fordham and Wakefield) and in Staten Island (St. George). Since that time and owing to the great success of the project, the number of sites has already been expanded to ten and is about to double again.

The success of public access computing depends on careful selection and evaluation of software. The old computer acronym "GIGO" (garbage in, garbage out) holds true with pre-programmed software packages. Garbage software in equals garbage software out. Therefore the software collections at the New York Public Library (NYPL) sites have always been given the utmost importance. The software determines the effectiveness of the machine.

This article will share the thinking behind the development of the software collection at NYPL with other library systems which are in the process of setting up public access computing. At the conclusion of the article, readers will find NYPL's core list of recommended software titles for an adult public access site. While the list contains software for the Apple II, it can be used as a guideline for other microcomputers.

Two kinds of literacy

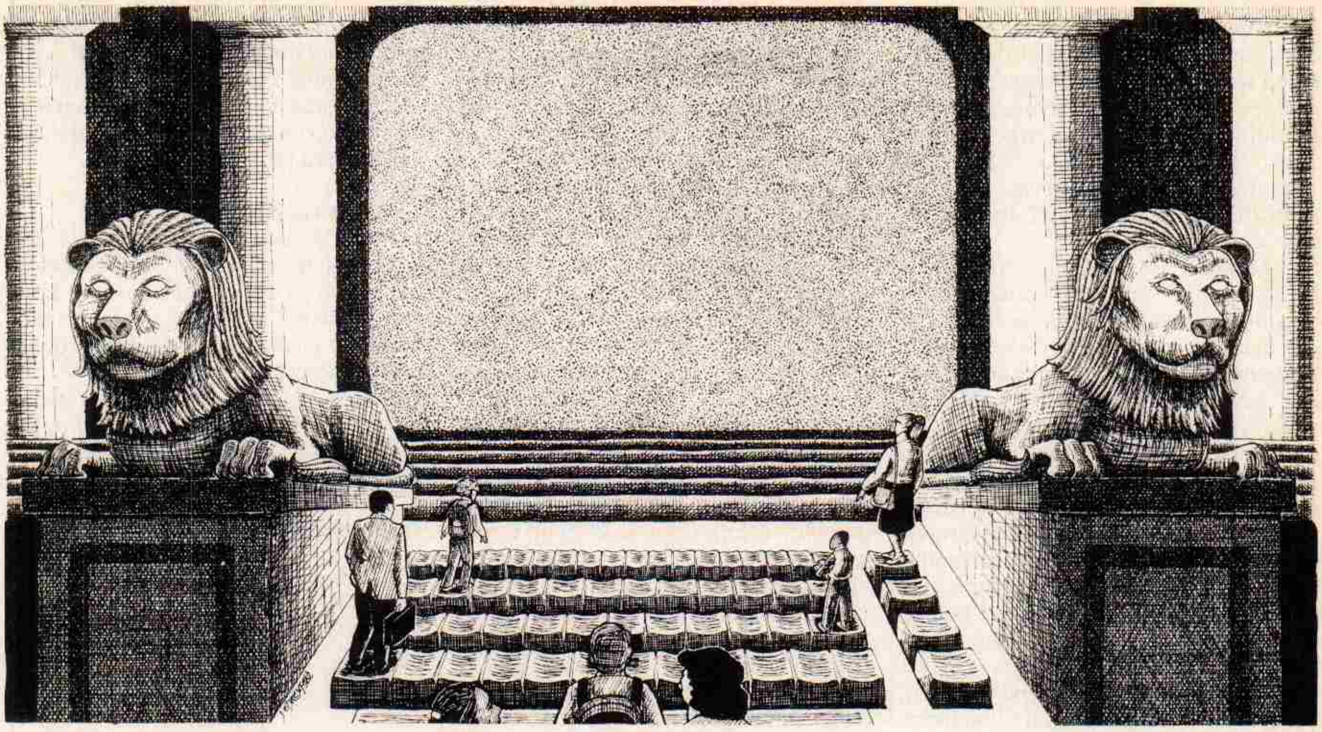
Originally, the main thrust of the NYPL project was computer literacy. This overused term has come to mean a familiarity with exactly what a computer can and cannot do. People who visit these microcomputer sites on a regular basis can develop some real skills depending on where their interests lie. Many adults have used these microcomputers to make themselves more employable by learning word processing, spreadsheets, and database creation. Software to practice basic skills in math and English is also available. Most recently we have linked our Adult Literacy Project with microcomputer facilities dedicated to teaching reading. This newest project marries computer literacy to traditional reading literacy using both books and computers as tools.

Having an idea of what kind of software one is looking for and finding it on the market can be two very distinct functions. The software market can be brutal. Peter McWilliams once referred to the hardware market as a "wilderness." Finding good, appropriate software is like exploring a black hole. First, the sheer quantity is overwhelming, not to mention the constantly changing market. As soon as one directory or bibliography is printed, it is out of date. In a way, the varying quality of products is precisely what makes the whole software field so interesting and challenging. On days of utter frustration, one can take comfort in the fact that if the right software does not exist today, someone will surely develop it tomorrow. On the other hand, a certain amount of stabilization has taken place and some recommendations are possible.

Taking control, languages

The main thing to teach anyone seated at a computer for the first time is that he or she must be in control of the machine. The user will then discover that one of the great joys of working on computers is being in charge of a universe of your own creation. Programming is the epitome of telling the computer what to do. Although most people will not go out and become hotshot pro-

Susan Rappaport is Director of the Microcomputer Project, at the New York Public Library



grammers, it is important to take the mystique out of programming. Logo is an ideal language to use as a first step in this de-mystification process. Logo is an educational language developed at the artificial intelligence lab at MIT by a research team headed by Seymour Papert. Although it is easy to get into at the beginning, it becomes highly complex in its advanced stages. Logo has all the advantages of a sophisticated language in that it is procedural, interactive, and recursive. It has the added advantage of giving immediate feedback which makes debugging easy. This facility with debugging encourages constant evaluation and revision, essential to programming. Ultimately, Logo's forte is its emphasis on problem-solving. It can be used in one of three modes, graphics, music, and text, depending on whatever the user is more comfortable with. Logo is not simply for the use of children, but rather a fine tool for teaching computer skills to people of all ages.

Languages which offer no immediate feedback to be used with more sophisticated patrons are Pascal and BASIC. Pascal is a natural follow-up to Logo. It has many similar characteristics in that it is both procedural and recursive. Like many of the newer languages, it is highly structured and is used to teach programming to most college students. Even though it was originally developed for a mainframe system by Nicholas Wirth in 1970 in Switzerland, it has since been adapted for microcomputers at the University of California at San Diego.

BASIC is the other option for a public access site, only because it is already in the ROM (read only memory) on an Apple and many people are familiar with it. There is a series of three software packages beginning with *The New Step by Step* which teaches BASIC using cassettes in conjunction with hands-on training. A variety of software packages for a variety of machines are available for all three languages, particularly Logo.

Playing programmer

If people are interested in getting a feel for what it is like to be a full-fledged programmer, they could try working on the adventure game of *Zork*. Promoted as recreational software, *Zork* is actually educational in nature. In her book, *The Second Self*, Sherry Turkle tells how she was advised to use the MIT version to see what it was like to be a true computer hacker (p. 224-225). Wandering *Zork*'s underground maze, picking up and dropping tools to figure out various problems, obviously came from the mind of a programmer. *Zork* is almost a physical manifestation of the abstract process.

For people interested in how a computer actually works, there are several pieces of software which help. The earliest to be issued was *Rocky's Boots*. With Rocky Raccoon as host, it manages to teach the building of logic gates in the most charming and playful fashion. Because it has an age range of nine to adult, the teaching of Boolean Algebra is accomplished in a nonthreatening way. The same people have recently produced an even more sophisticated piece of software called *Robot Odyssey* which advances into chip design in order to construct robots. Although less user friendly, there is also *Simulated Computer*, a package which simulates the inner workings of a computer using decimal numbers instead of binary.

Simulations

Simulations are another area where computers have had a great deal of impact. Once again, it puts the user in control of the machine, but in an arena of his/her choice. Perhaps, the most famous of all simulation software packages is *Flight Simulator* where one suddenly finds oneself in the pilot's seat, performing takeoffs, landings, and in the beginning, crashes. Other examples are *Three Mile Island* (a nuclear power plant), *Millionaire* (the stock market), and *Pinball Construction Kit* (pinball). *Pinball Construction Kit* is especially interesting because the player can make things happen which are physically impossible. Changing the level of gravity or putting in black holes are simple manipulations of a joystick.

Business software

Back to the mundane world, adults are frequently interested in business applications. Their main concern is usually with word processing since it resembles the familiar typewriter. Unlike the typewriter, word processors simulate your finished product, complete with total editing facilities. Adults quickly perceive the power of the computer if they are used to typing. At this moment in time, *Homeword* is probably the easiest word processor with which to start. From there, they can move on to *PFS:Write* or *Word Juggler*.

Database systems are the next way

to convince people of the convenience of a computer. *PFS:File* is both simple to use and efficient. It can also be interfaced with *PFS:Write*. By the way, the greatest advantage to buying products by Software Publishing Corporation is their interfacing abilities. *PFS:Report* will spew our reports based on files from *PFS:File*, and *PFS:Graph* can be linked to *Visicalc*.

Mention of *Visicalc*, of course, brings up the third major business application, spreadsheets. Spreadsheets are actually magic ledger sheets with accounting and editing options. *Visicalc* was the first spreadsheet program which actually initiated the whole microcomputer boom. Now, there are many other spreadsheets including *Supercalc* and *Lotus 1-2-3* (for the IBM). *Lotus* was the first of the "integrated software packages," a term meaning software with more than one function. In *Lotus* there is a spreadsheet, graph, and database in one package. Many other integrated packages have arrived on the market mostly for the IBM PC because of the large amount of memory necessary. At the NYPL microcomputer sites, we have only the PFS series available with *Visicalc* which gives the flavor of interfacing.

Other business skills can be practiced on the computer such as typing. *Typing Tutor* is the most comprehensive teaching tool with *Mastertype* running a close second as a drill and practice package. *Typing Tutor* is more serious while *Mastertype* is in game format. For performing accounting on a home scale, there is an alternative to *Visicalc* called *Home Accountant*. Some patrons seem to prefer this less powerful package to *Visicalc*.

Tutorials

Tutorials have been appearing on the market in increasing numbers and fit quite nicely into a public access environment. Novice computer users can teach themselves the essentials from packages such as *Training for the Apple IIe* and *Apple Presents Apple*, a software package which comes with the Apple. *Training for Visicalc* has also proven useful for the first-time user of this spreadsheet program. These teach yourself packages serve as hands-on reference manuals, to be used again and again.

Graphics

One of the major areas of development for the computer lies in graphics capabilities. There are three major categories for graphics programs. For a person totally unfamiliar with the computer, an intuitive approach should be employed. Actual drawing can be done with either a graphics tablet or a joystick. The *Koalapad* has many of the features of professional graphics tablets and yet it is easy to learn and use and inexpensive. *Picture Writer* is a graphics kit to be used with a joystick, which although made for children, is highly recommended for any novice on a computer. The next level of expertise in creating graphics is using utility programs such as *Alpha Plot* or *Graphics Magician*. These packages are already programmed so that the user controls only the final stages of the graphics. But there is a refinement here which might not necessarily exist with a graphics tablet. Now there are more utility packages appearing on the market with specific purposes. *Moviemaker* is an animation system which allows for making digital movies. *Print Shop* provides for making cards, banners, and stationery. The highest level of graphics composition is of course programming, where there is total control of the process from beginning to end. Logo used in the graphics mode will fill that bill. For the more sophisticated, there is *Graforth*, a graphics language which will enable the creation of three-dimensional graphics. In order to get printed copy of what has been produced on the monitor, it is advisable to have available a package which will perform screen dumps such as *Zoom Grafix*.

Music

Music is the other way of making your computer into an instrument. Although the Commodore is more well-known as a music-maker, there are a few packages for the Apple. *Music Construction Kit* allows the user to literally compose music by manipulating notes and other notations with a joystick. *Music Games* is a piece of software designed to train actual music skills such as note recognition and rhythm identification. *Logo* can also be programmed in its music mode to compose songs. *Graforth* also contains a music component.

Looking into the future a bit, there are indications of a kind of software which can be likened to an interactive novel. Examples of this are *Deadline* and *Suspended*. *Deadline* is a mystery novel in which the user as detective must figure out the murderer. *Suspended* is a science fiction extravaganza where the player wakes up in the future in a world of robots. The interactive capacity of the software makes role playing an integral part. Perhaps future books will allow for the main character to be played out on a computer. This joining of reading and computing opens up a whole world of possibilities for a new, creative artform.

In conclusion, public access software collections should provide a showcase for the uninitiated to get hands-on experience to understand exactly what a computer can do and to realize the limitations of the technology as well. Most important of all, software which puts the user in contol of the machine should be emphasized. Technology can then be perceived for what it is, a new information tool which should be available to all.

The software core list

The software core list which follows was developed to teach computer literacy in a public access site. Producers are listed after the name of each software package or set. This list has been geared to use on an Apple II microcomputer and would have to be adjusted for any other kind of computer. Many of the software packages listed are available for other machines.

Programming Languages:

Logo, there are many different versions available
Pascal, Apple
The New Step by Step BASIC, Program Design, Inc.
Step by Step II & III

Computer Logic:

Rocky's Boots, Learning Company
Robot Odyssey, Learning Company
Simulated Computer, Edusoft

Simulations

Flight Simulator II, Sob-Logic
Three Mile Island, Muse
Pinball Construction Kit, Electronic Arts
Millionaire, Blue Chip Software

Tutorials

Apple Presents Apple, Apple
Training for the Apple, Cdex Corporation
Training for Visicalc, Cdex Corporation

Business Utilities

PFS:Write, *PFS:File*, *PFS:Graph*, *PFS:Report*, Software Publishing Corporation
Homeword, Sierra On-Line
Word Juggler, Quark
Home Accountant, Continental Software

Typing

Typing Tutor III, Kriya Systems
Mastertype, Lightning

Graphics

Logo. Many versions available
Graforth, Insoft
Graphics Magician, Penguin
Alpha Plot, Beagle Brothers
Movie Maker, Reston
Print Shop, Broderbund
Zoom Grafix, Phoenix
Picture Writer, Scarborough
Koalapad

Music

Logo
Graforth, Insoft
Music Construction Kit, Electronic Arts
Music Games, Howard W. Sams

Adventure Games

Zork, Infocom
Deadline, Infocom
Suspended, Infocom