ATARI
COMPUTER ENTHUASISTS
of COLUMBUS, OHIO
NEWSLETTER

August 13, 1984

SO MANY BASICS by Charles Brown

Book Reviews
"Elementary Basic"
"From Binary to Baker Street"

ADVANCED SYSTEM FEATURES

Published by
Atari Computer Enthuasists of Columbus, Ohio

for ACE of Columbus membership. Dues are on an annual basis and entitle the members to all club benefits (Newsletter, Disk or Tape of the month, group discounts, etc.). Monthly meetings, in the basement of State Savings, 6895 N. High Street, Worthington, Ohio are open to nonmembers.

Upcoming meeting dates at 7:30 pm are

DATE

August 13
September 10
October 8
November 12
December 4

PRESIDENT
Bill Eckert
6632 Lisamarie Road
Columbus, Ohio 43229
614-891-9785

VICE PRESIDENT
Donald Noble
7536 Northfield Court
Reynoldsburg, Ohio 43868
614-866-4587

MEMBERSHIP CHAIRMAN
Tim Adcock
7544 Satterfield Road
Worthington, Ohio 43885
614-764-9492

TREASURER
Mike Compton
1342 Gumwood
Columbus, Ohio 43229
614-885-3757

DISK LIBRARIAN
Sheldon Wesson
444 North Pearl Street
Granville, Ohio 43823
614-587-2716

CASSETTE LIBRARIAN
Larry Fletcher
12388 Oak Drive
Orient, Ohio 43146
614-877-9530

NEWSLETTER EDITOR
Norman Knapp
1222 Norton Avenue
Columbus, Ohio 43212
614-291-2849

ADDRESS ALL MAIL TO
ACE of Columbus
P.O. Box 849
Worthington, Ohio 43085

SO MANY BASICS

by Charles Brown

As far as I know, there are three Basic languages for the Atari computer. All three are uniquely different, and I am sure each one has it's advantages. The question that I have is why they are not compatible. It would be so nice and convenient if you could switch back and forth between them, but that would be too easy. All three have one major disadvantage, not enough memory. What I mean is that you only can use so much memory for Basic, no matter how much memory you put in your machine. You can only go so far.

First you have Atari Basic. It is a nice and simple language to learn. It has one very major problem. It will not handle string arrays, which is a very important concept in programming. This simple fact makes Atari Basic a very frustrating language to use.

The second Basic is Microsoft Basic, which I like for several important reasons. The first reason is that it will handle string arrays which Atari Basic lacks. The second reason is that it will handle a string array matrix, a very important concept of programming, especially for me. I have used Microsoft Basic to write a data base program which will file and organize information and then easily the resulting data base. Since it is nothing but a string array matrix, you can change your bytes of information in one file and the program will change others in a corresponding file. Microsoft Basic also gives you access to your disk files without having to go to DOS. It also allows you to format your print commands making it easier to make your programs look better, on the screen or the printer. You also have useful programming features like line renumbering, auto line numbering, and tracing of programs. Tracing allows you to follow the running of a program permitting where control is in relation to line numbers. This helps in finding mistakes.

Microsoft Basic was the original Basic written for all home computers. When the different computer companies made their computers, they modified Microsoft Basic to make their own versions of Basic. You might say that Microsoft Basic is the very most Basic of all the versions of Basic. It is too bad that all the different companies could not use just one version. That way if some one wrote a real good program on one type of machine, then all people could take advantage of it, no matter what type of computer they had.

The third Basic is one that I have heard a lot of good things about, Basic XL, which it is the most powerful of the three Basics for the Atari. The ony disadvantage that it has is that it will not handle a string array matrix. For only that reason, I like Microsoft Basic better than Basic XL. Basic XL commands adn ability is similar to Microsoft Basic with some differences. Both Microsoft Basic and Basic XL give you access to the disk files without going to DOSm but Basic XL gives you more capability. One of my favorite Basic XL features is the DIR command. When you key DIR and then hit return, the complete list of disk files in D1 or any other drive you specify is displayed without losing the program in memory. With this feature you do not have to worry about saving your program or having to have a MEM.SAV file on your disk. Another command which Basic XL makes so convenient is the LVAR command which lists the variables in your program and the corresponding line numbers. I would like to have the LVAR command list only specified variables rather than all the variables. This way you could easily read and get the information that you need without looking a lot of unnecessary information. There are a lot of commands used by Basic XL for player missle graphics (some of which are used by Microsoft Basic) which makes Basic XL easier to use than Atari Basic. I feel that Basic XL is the best of the three Basics for the Atari, but I just wich it would handle string array matrices. If it did, I would not need Microsoft Basic.

Each of these languages has its advantages and disadvantages. It all depends on what you want the computer to do. If they wee all compatible with each other it would be so easy for all of us. I guess life can't be that simple. Each person has their own opinion, and I am sure that there are people out there who disagree with me. I did not write this to force my opinion on anyone, just merely to show wome of the unique features of these 3 languages so you can draw your own conclusions.

BOOK REVIEWS

"From Baker Street to Binary"
Henry Ledgard, E. Patrick McQuaid, and Andrew Singer, eds.
McGraw-Hill Book Company, 1983, $10.95.

"Elementary BASIC"
as chronicled by John H. Watson
edited with commentaries by Henry Ledgard and Andrew Singer
Vintage Books * A Division of Random House * New York, 1982, $12.95.

The recently discovered manuscripts of these books penned by Dr. John H. Watson demonstrate the methods and techniques of the first great consulting detective, Sherlock Holmes. Not since publication of the very first case study reported by Dr. Watson, "A Study in Scarlet," where the foundations of modern forensic science were laid, has the general public been shown, in detail, Holme's techniques. In these two volumes, the reader is introduced to computer concepts from simple but fundamental concepts of software and hardware, data and its representation, to more advanced material such as algorithims and computer languages, and esoteric subjects such as artificial intelligence, the Turing machine, and the solvability problem. Each chapter in these two books chronicles a previously unknown case of Sherlock Holmes in which the computer played a key role in solving a case.

Each chapter is divided into two parts, the original manuscipt and the editors' modern interpretation; there are pictures of the Atari 800 and the Apple III on page 117 of "From Baker Street to Binary." The original manuscripts are delightful stories written in the style so characteristic of the original Sherlock Holmes stories. The cases in "Elementary Basic" are dealt with in a more technical way appealing to the reader familiar with Basic and illustrating various aspects of programming in Basic. The editors have chosen to write Holmes' concepts using a generic form of Micorsoft Basic; modified versions of these programs can be run on most any modern computer including the Atari. The Basic XL and Microsoft Basic users will not find in "Elementary Basic" the IF... ELSE... ENDIF and WHILE/ ENDWHILE statements which add so power to these two languages. In this respect "Elementary Basic" is more like Atari Basic than Microsoft Basic, except for string handling.

In his address to Royal Society, Sherlock Holmes explains his programming style, the top-down approach. This programming style requires a clear and complete statement of the problem as well as a basic plan or algorithm before writing the program. The characteristics of the top-down approach are as follows:

1.A program is designed and then constructed according to a conceptual heirarchy. The upper levels of the heirarchy indicate the more general features of the problem, with details and elaborations introduced at the lower levels.

2.Ordinary English is sufficient in the early stages; a computer compatible language, such as Basic, is used only after several refinements.

3.A firm grasp of the broad aspects of a problem at each level is

required before dealing with details at lower levels.

4.Before advancing to a lower level, the solution must be stated precisely.

5.As a new level unfolds, the solution must be verified before proceeding to lower levels.

For lack of space and time, the reviewer refers the reader interested in details to "Elementary Basic" where the top-down approach is applied to the design of one the most useful and elegant of programs, the word processor. There is just one problem with this program; it processes an existing text file. There is no way to construct a text file in the manner of most modern word processors. The reviewer hopes that this is an oversight of the editors or that a piece of the original manuscript has been misplaced rather than a lapse in the logic of the master logician.

The reviewer has several opinions on the top-down method, which is a heated subject when it comes up for discussion in a group of programmers. This is mainly because its proponents make what seem like outrageous claims concerning its usefulness while refusing to concede that there are other successful programming techniques.

Considering briefly the latter point, there is a language, with enthuasistic adherents (free software), which uses the bottom up approach, FORTH. A program written according to the top-down approach, has a structure like a pyramid with the MAIN module as the point. The MAIN module calls other modules as subroutine; each module in turn can call several other modules below it which in turn call others until modules are reached which do not call others. The FORTH programmer starts with several fundamental modules and builds his pyramid from the base up.

A programmer familiar with Basic as well as other languages such as Pascal, C, and FORTRAN can see here a limitation of Basic; Basic is more difficult to program than most other languages in modules which pass values between them. One explanation is that the Basic interpreter maintains a list of variables for the entire program rather than separate lists for different modules. A modular languages, one with true subroutines, like Pascal defines variables valid only for a certain module (local varibles) and those valid for entire program (universal variables. In Basic, all variables all variables are universal. Of course, the Basic programmer can merge merge modules to form a larger program but he has to be more alert concerning variable definition (Basic is sloppy, Pascal etc. are very rigid) and line number usage (none required for Pascal). My concluding remark on the top-down approach is that its use for the Pascal programmer is compulsory while its use is voluntary for Basic programmers, many of whom could profit by its use.

While the emphasis in "Elementary Basic" has been on programming applications, "From Binary to Baker Street" goes beyond programming to consider historical and social aspects of computers. Sherlock Holmes' computer was a mechanical device known as the Analytical Engine conceived of and designed by Charles Babbage, a Cambridge professor of mathematics. Babbage's earlier work on the Difference Engine, supported at considerable expense for years by the British government, was directed toward toward calculation of mathematical tables.

Although Babbage never completed the Difference Engine, an independent implementation was successful. Government support for Babbage's work ceased when he tried to interest the Treasury in sponsoring a new and more powerful design, the Analytical Engine, an automatically sequenced, general purpose calculating machine programmed by punched cards.

These two books reviewed here are the first indication that the Analytical Engine has ever been impemented; the manuscripts having been suppressed until now since there previously has not been a suitable readership for them. The implementation of the Analytical Engine used by Sherlock Holmes has gone beyond that outlined in Babbage's notebooks. The software used by Holmes is considerably more sophisticated than the first programs written for Babbage's Analytical Engine by Lady Augusta Ada Lovelace, after whoh the Department of Defence has named its universal computer language, ADA.

In the traditional Holmesian fashion, Holmes with his unerring logic and Dr. Watson in his bumbling fashion proceed to solve previously undisclosed cases using the Analytical Engine. In the first few chapters of "From Binary to Baker Street" data representation by means of the binary ASCII code, computer memory, data storage, input/output, and CPU operation are explained using a fascinating story of diplomatic intrigue.

After several chapters concerned with computer applications, the shortcomings of the Analytical Engine are summarized. Some of these shortcomings are concerned with inadequate documentation, meaning of keys and complex keystrokes, unfamiliar vocabulary, ease of operator error, difficulty of locating and correcting errors, and the unnecessary costs associated with daily use. Perhaps the most significant of the unnecessary costs arise from the failure of the computers' diverse users to communicate ideas and problems to one another.

Subjects covered in the last few chapters are concerned with operation of the ideal computer (the Turing machine), the solvability of a problem, artificial intelligence, misconceptions held about computers, and their social implications.

These two books are rather unique in that a popular literary format has been successfully used to convey computer concepts and programming techniques. The beginning programmer will not find here everything he needs since specific hardware and software implementations are not covered. The universal concepts treated by these two books are translatable to the user's computer. The delightful and thoughtful stories can be read aside from the editors' explanation of details.

Reviewed by Norman Knapp