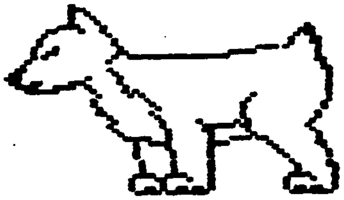

D.A.U.G.

Newsletter

DuPage Apple Users Group

JUNE 1985

JUNE MEETING

When: 7:30 pm, June 11, 1985

Where: Faith United Methodist Church
59th & Fairview, Downers Grove

Topic: Mind Probe Story and Apple Documentation

Other Dates to Remember

June 4: Applesoft SIG (Call 968-3897 for details)
11: Regular meeting
19: DAUG Board Meeting (7 pm, Downers Grove Library)
July 2: Applesoft SIG (Call 968-3897 for details)
9: Regular meeting
17: DAUG Board Meeting (7 pm, Downers Grove Library)
Aug 6: Applesoft SIG (Call 968-3897 for details)
13: Regular meeting
28: DAUG Board meeting (7 pm, Downers Grove Library)
Sept 3: Applesoft SIG (Call 968-3897 for details)
10: Regular meeting
18: DAUG Board meeting (7 pm, Downers Grove Library)

*****

OFFICERS

President:	Bob Konikow,	968-3897
VP/Membership:	Jim Cormack,	960-5691
Secretary:	Bruce Mansfield,	420-1608
Treasurer:	John Sandora,	352-3059
Librarian:	Detlef Adolff,	766-7741
Program Ch:	Ed Danley,	969-4433
Editor:	Dave Dohmeier,	941-1645
Directors:	Chuck Jonah,	985-5497
	Priscilla Walling,	964-4894

*****

Treasurers Report for	May
PREVIOUS BALANCE	1177.61
REVENUE	
MEMBERSHIP	106.00
MONTHLY DOM'S	65.00
TOTAL REVENUE	171.00
EXPENSES	
REFRESHMENTS	11.00
OPERATIONAL EXPENSES	130.00
NEWSLETTER+STAMPS	127.20
RENT	100.00
TOTAL EXPENSES	368.20
NET INCOME	-197.20
ENDING BALANCE	980.41

Ed Danley

*****

HELP LINE

The following members have volunteered to answer questions by phone on the subjects listed. Please be considerate when calling for help. Think through your question; collect all the information; and don't call later than 10 p.m. If you'd like to help, let us know.

Applesoft:	Chuck Jonah, 985-5497
DOS:	Chuck Jonah, 985-5497
Interface:	Rufus Teesdale, 469-8836
Mach Lang:	Chuck Jonah, 985-5497
Modems:	Rufus Teesdale, 469-8836

*****

SOFTWARE REVIEW: SIMPLY PERFECT: Is It Perfect?

Summary

Program: INTEGRATED WP-DB
Purpose: Word processor and
Data Manager
Equipment Req'd: Apple //e with
extended memory, disk
drive and printer
Optional Equip.: Second Drive, Ultraterm
Suggested Retail Price: $189.95
Rating on a scale of 1 to 5: 4.5 (for two functions evaluated)

Publisher:
LJK, Inc.
7852 Big Bend Blvd.
St. Louis, MO 63119

My primary use of my Apple has been word processing, whether for work or as editor of the Dillo. I have had the opportunity to review and tryout numerous word processors (this has been especially true, since the Club has begun reviewing software), however I have usually felt that the time and effort needed to master a new word processor was not worth it. For this reason I have continued to use SuperText, which was the original word processor I learned to use. In fact, I continued to use SuperText (I currently use SuperText Pro) because none of the word processors that I've had the opportunity to try offered me features, in excess of SuperText's, that I felt were worth enough to expend the time to learn. The feature that I most wanted from a word processor was being able to merge information from a data base into a form letter without having to run eighteen other programs. Yes, Wordstar has Mailmerge, Supertext has its Address Book and Form Letter modules and Applewriter has WPL and Quickfile, however each of these required running another program to achieve the mail merge function. Well, the people at LJK, Inc. have come to my rescue and I think to the rescue of many other Apple //e (a //c version is expected shortly), who have, and are continuing to search for an integrated word processing package. While I have been told since early childhood that nothing is perfect, LJK's Simply Perfect is pretty close and given its price of $189, it offers what I consider to be one of the best values in Apple software today. But enough of my biases, what does Simply Perfect offer a perspective purchaser?

To begin with, Simply Perfect requires an Apple //e with an extended 80 column board and at least one disk drive. However, if you want to take full advantage of the data base and spelling checker integration, two disk drives are needed. For those of you who have a hard disk, yes it is possible to write a disk driver to interface SP with different types of disk drives, a hard disk driver for the Xebec drive is presented in an Appendix. While this information did not make much sense to me (my knowledge of assembly language is comparable to my knowledge of Sanskrit), providing this information to the user is most welcome, since most of us envision the day when a gigabyte of on-line storage will be next to our trusty Apple. You might ask why a software vendor provides one with this type information when we all know that it's next to impossible for most software in the private domain to work with anything but standard Apple drives because of the elaborate, but usually ineffective, copy protection schemes found with almost all commercially available software. Well, the answer is that SP is not copy protected. Yes, let me repeat that, SP is not copy protected. So for those of you who have both a dot matrix and letter quality printer you do not have to re-install the control codes, etc, when you use your other printer. In fact, the SP disk contains standard configurations for a number of printers, including Epson, Apple DMP, Okidata 92, Qume and NEC Spinwriter. Should your printer not be included, worksheets are provided for indicating the escape and control codes associated with your printer so SP will be able to fully utilize the capabilities of your printer. One other notable feature is that SP will utilize the Videx Ultraterm board with its various display modes as long as you have an extended 80 column board. As you can tell from the above, SP can be configured in just about any way you could want now and in the future. But what about its word processing, data base, spelling and terminal capabilities.

The first task you are asked to perform with SP is to make a backup copy of your program disk with the COPYA program found on your Apple Master Disk. Upon completing that task you are instructed to place the SP program disk in a safe place and use your newly created backup as your working copy. Or if you would like, I assume LJK will accept your check for $20 or $30 and send you a backup. (Isn't this what most reputable software manufacturers do???). You are next informed how to configure the program to take advantage of your particular computer setup. When configuring SP you are asked for the type of video display you have (while you must have the Apple //e extended 80 card SP will take advantage of the Videx Ultraterm). The next menu sets the unit default assignments. If you have a two drive system all you'll probably need to do is accept the default assignments. Just remember that once SP is loaded the program (the backup you just created) can be taken out of the disk drive and you will not need to access it again, since the entire program (both word processor and data base) are maintained in memory. For this reason your data disk for your Letter Perfect files is located in drive 1. Data Perfect file disks are maintained in drive 2. The reason for this particular configuration is when you later want to merge information from your data base into a letter, the operation is easily performed with this setup. After accepting or modifying this menu, one is shown a display of the printer drivers that are available. If your printer is shown on the menu you would load that driver (the Epson driver is loaded by default when first configuring the program) and answer 'no' if you do not what to change the driver routine. What is contained in the various print driver routines is found in the Appendix. In the Appendix is also found an explanation of what features are supported among the various printers along with blank worksheets if you care to develop your own print driver routine.

After you have configured SP for your particular setup, you can then begin using either the word processor or file manager. Both programs always reside in memory and all that is necessary to switch between them is to select which function you would like to perform from the main menu. In the SP manual the word processing tutorial comes first and this is where we will start. A file called "sample" is stored on the disk which is utilized to lead you through the tutorial. The tutorial instructs you how the various commands in SP are utilized to load and edit this file. For those of you who want to immediately get into using the program all you need to do is select the edit command from the main menu and you will immediately be presented with a blank screen to begin creating text. Available for your document is

39678 bytes, which is equivalent to approximately 11 pages of text (SP was not able to take advantage of the additional 64K available on the extended 80 column board I am using). There are approximately 65 commands which make up Simply Perfect. These range from 'advance by paragraph' (esc A) to the 'underline toggle' (ctrl N). A handy reference card is provided which lists these commands and the escape or control sequence which activates each. Most of these commands are quite logical, i.e., a goto beginning of text is esc B, while a replace is a ctrl R and quiting the editor is a ctrl Q. In all, I was quite impressed with the range and versatility of the commands available within the editor. One of the more useful features is that for most of the commands which could delete major portions of your text, SP requires you to press the '#' key (shift 3) before it will complete the selected deletion operation command available within the editor.

After you've completed composing your text, you can check your spelling and redundancy by exiting your document (ctrl Q), which will return you to the main menu. From the main menu, select the VERIFY option (all you need to do is strike 'V') and SP will begin a count of the number of total and different words contained in your document. If you have a two drive system you should have placed the Spell Perfect disk in drive two and a search of the dictionary will commence. The dictionary search will indicate the number of unlisted words along with the number of different words. After these counts have been completed you are asked what operation (Ignore, Match, Change, Quit) you would like performed on the unlisted words. Except for the 'match' command, the other commands are self explanatory. The match command is useful when you are not sure of the spelling of a word. When an unlisted word is highlighted during a spell checking and your not quite sure what you intended to write, striking 'M' will present you with a list of words which sound like the word highlighted. Any of these words can be selected to replace the highlighted word. In checking the text for some of the articles in this month's newsletter I usually found that about 10% of the words (e.g., DILLO, byte, Stuart, NEC, etc.) contained in the article were unlisted. Not to worry, the basic dictionary can quite easily be updated. First you should have backed-up the dictionary disk, then type the list of words you want added to the dictionary in the editor, return to the main menu and select the "Update" option from the main menu. You will be presented with a number of options (Add, Delete and Backup) from which to select. By choosing Add, the words contained in the editor will be added to your working dictionary. I was not able to determine how many words are on the Spell Perfect disk, nor was there any mention of how many words could be contained on a dictionary disk. Other than this shortcoming, I was very impressed with how easily it was to use the speller in conjunction with documents. No need to boot a spelling checker program and then remember the name of the file you want checked.

The data base management portion of Simply Perfect is called Data Perfect (DP). Data Perfect is considered a file manager in that you are restricted to a single form type within a data file. The structure of DP is similar to that of a file cabinet where each drawer in the cabinet represents a FILE, each folder in the drawer represents one RECORD in the FILE, and in each folder or RECORD are items or FIELDS of information. Each file can contain a maximum of 109K bytes of information. These data are apportioned such that the maximum number of fields per record is no more than 32, with each field's maximum length being 127 bytes and a maximum record length (LRECL) of 511. Additionally, the field name length cannot exceed 12 characters. Each file that you create MUST be maintained on its own disk. That is you can only have one file per diskette, irrespective of the size of your file.

Four types of fields are supported within DP. These are Alphanumeric (e.g., name, city, state), Date (MM/DD/YY, i.e., month, day, year), Numeric (information which is treated arithmetically) and Formulated fields (fields derived from calculations performed on numeric fields).

To create a data file you first select the Database option from the Main Menu. Since this is your first database you would select the 'CREATE DATA BASE' option from the database main menu. This will format your data file disks and store the file structure on the disks. Only one data file can be contained on a data disk. After completing this task you will be requested to provide a filename. The file name cannot be more than 12 characters. You are now asked to specify the field names for the information you want contained within this data file. One of the nice features when creating the file structure is you have complete control over the screen design. You can place fields for each data item anywhere on the screen. Since a data record is restricted to 511 bytes a counter is maintained in the upper right hand corner indicating the number of bytes used in creating the file. After completing the data file structure you can indicate which data elements are to contain default values. For example, if you are creating a mailing list and most of the addresses are in Austin, Texas, you could enter Austin as the default for the city field and TX as the default for the state field. When you have completed organizing your file structure hit "Q" to quit the EDIT mode. You are now shown the profile of the data base you are going to create. If this is satisfactory you press "RETURN" until you are requested to insert a blank disk to contain the data you will be inputting. After the data disk is formatted you are returned to the main menu and select any other option you desire.

If you select FILE MAINTENANCE you would be presented with another menu from which you could select the various maintenance options, such as add, edit, delete, etc. Since you have just created this data file adding records is what you would probably select. After selecting "ADD" the input screen you previously designed would appear and you can begin entering data. Before saving a record to the data file you have a number of options available to insure that you will save the correct information. After adding records to the data file, a number of searching options are available. These options include an individual record, a range of records or all the records. One can also search the file by values contained in a field. For instance, all records with a date between 8/1/84 and 9/1/84. Wildcard searches are also available, with the wildcard either for a block of characters or a single character.

Once you've entered data into your data file you'll probably want to printout the information. Well, SP has a fairly versatile report feature. To generate a report you select the REPORTS function from the main menu. From this menu you can load or create/edit a report. It is also possible to initialize your printer with up to 7 decimal code printer commands. Substantial flexibility is offered with the report writing function of SP. You are able to customize both the header associated with a report along with the data elements contained within the report. The capability to do both subtotals and totals is also available.

Other features contained within the file manager component of SP include reformatting and merging of a data file, creating computed fields within a data file, global updating of fields and last but not least integrating information from the data file with the word processor within SP. The reformatting option allows one to create either a new data base or make blank copies of existing data files. With this function you can add fields, change field lengths or formulas to an old data file. After reformatting an existing data file, the merge capability allows the transfer of data from an existing data file into the newly revised data structure. Besides keying in data elements, SP allows the use of computed fields. This capability provides the user with the ability to define the contents of a field to be dependent on values or conditions in other fields. An example of this is an invoice file, which contains fields for quantity ordered and price per unit. One could define a new field whose value would be the sum of price times quantity. Since SP allows boolean functions (i.e., greater than, equal to, or less than), fields based on whether other fields are true or false are also allowed. Global updating of fields is useful when changes have to be made to records in a file depending upon whether some condition is meet. For example, updating members whose expiration date is the current month. With SP you could search for all records whose expiration date value is equal to SEP84 and then do an automatic update to SEP85 if renewal dues have been paid.

As I mentioned in the first part of this review, a feature I have wanted since I first got an Apple was the ability to do mail merges. That is to combine information from a data file into a form letter. I know that various CP/M and Apple packages have this capability, but none to my knowledge provided this feature in an intergrated package. Well, SP has it and it does work. All one needs to do is indicate the field number of the data element you want incorporated into your form and presto SP will insert data from you data file into the letter or form you've created in the word processing component of SP. Pretty easy and remember you can select only records which meet your criteria to be printed. Try that with AppleWorks!!!!

Before concluding this review, I must mention the "hacker" section included in the documentation. Included in Section 4 of the Appendix is a number of sections detailing how the user can customize SP. Included in this section are discussions on writing a driver so SP can be used with a hard disk (under a DOS 3.3 format), writing a custom print driver, an example modem program so files can be transmitted between users and how information is stored in the Data Perfect module. With regards to the latter, an example BASIC program is provided to familiarize one with what is required to access any data file created with Data Perfect.

You must think by now that I must be being paid by LJK, Inc. to hype their product since all I mentioned in these reviews is the advantages I found using SP. However, there are a few complaints I have since I've begun using this program. My major complaint is with the cursor that is utilized with the //e version of the program. SP employs a solid inverse block cursor to indicate your position on the screen. Unfortunately, this is the same way control characters are displayed. So that if the cursor is over a control character you lose sight of your position on the screen. The other significant problem I've encountered is when using the spelling program extra characters are inserted at the end of the text. This has required that after checking the spelling I return to the editor and delete this redundant text. I would hope that these problems are remedied in future releases. My other lament is that the file manager does not support DIF files. Having this feature would allow one to transfer data contained in a data file to a spreadsheet program and vice versa without having to write your own BASIC program. Again, I hope that this feature will be included in future releases. There are two additional features I would have liked incorporated into SP. One would be the capability to utilize the 128K of additional memory I have with the Applied Engineering extended 80 column board I'm using with the //e. This feature would substantially increase the size of a document that could be contained in memory. I also would have preferred that SP be able to read a clock card instead of having to enter the date each time you boot the program. I imagine that if I devoted additional time to this review I would find other features which I would like implemented. Isn't this what we all find after using any software package we've purchased?

In conclusion I return to the title of this review, "Is it Perfect"? No, but then is anything? However, I consider Simply Perfect to be an outstanding program. In fact I would recommend that any user should consider it as the first package they buy when purchasing their Apple. At a suggested retail price of $189.95 this is an excellent value. With the $50.00 credit to any individual who sends an original commercial program disk and documentation to LJK when ordering Simply Perfect directly from LJK, Simply Perfect is an exceptional value. Consider the costs when purchasing a word processor, file manager and spelling checker individually and you'll find Simply Perfect to be almost 50 percent less. For example, Appleworks, which does not have a spelling checker nor the ability to merge data from its data files into a letter, but does have a spreadsheet, retails for $250, while the comparable PFS series (PFS Write, PFS File, PFS Report) retails for $375.

Stuart Greenfield
Reprinted from the "Apple Dillo" newsletter

*****

MAY MEETING

The first order of business was the annual election of officers as specified by our Bylaws. The election was supervised by the club secretary, Ron Scharping. A list of those elected is given elsewhere in the Newsletter.

Priscilla Walling then described the May Disk of the Month (DOM). Among other programs, it contains the latest version of File Cabinet, one of the first major Public Domain application programs for Apple IIs. The latest version uses ProDOS. We now have in our DOM library ProDOS versions of a Data Base program

(File Cabinet) and a Word Processor (FreeWriter, DOM Vol. 42, Jan '85). There's also a Spreadsheet program (BASICALC, DOM Vol. 23, June '83), but it uses DOS 3.3. The existence of these three programs on separate DOMs suggest an interesting and potentially useful project for a DAUG member to take on. Convert BASICALC to a ProDOS version and then write an appropriate STARTUP program, including use of ProDOS's RAM disk, that will integrate these three important application programs on a special DOM. We can call it DaugWorks.

Ralph Stein then talked about a possible DAUG activity; the preparation of an annotated bibliography of Apple IIc and IIe books and manuals for eventual publication in the Newsletter (See "Proposed Topics for June 1985 Meeting" in the May issue of the Newsletter). He had a "first draft" book list and asked for volunteers to take portions of this list, check it for accuracy, add to it, and supply annotations for a few of the books listed. The annotations should be very short - one or two sentences - covering availablity, thoroughness of coverage, level of expertise required of the reader, and readability. Bruce Mansfield (420-1608) had previously volunteered to assist with organization and coordination of the effort. During the meeting break Ron Consentino (985-8246) volunteered to work on the IIc portion of the list, Dennis Shanahan (834-7213) and Jim Graff (348-3551) agreed to take on the IIe portion. No one as yet has volunteered to assist with the ProDOS/DOS 3.3 and Applesoft portions. We need more volunteers for all of the four list categories. HELP! If you would like to assist or if you think you have useful information call those who are currently involved.

The second half of the meeting consisted of descriptions and demonstrations of four relatively new commercial programs for Apple IIs. DAUG member Ron Consentino covered "NewsRoom", a program which has had very little advertising but never the less has been a best seller according to reports from local stores. It can be used to prepare newsletters or other documents with embedded graphics and various character fonts. Its mode of operation has MacIntosh like features including the use of Icons and Mouse cursor action as an option. A Telecommunication option for file transfer is also included. It lists at $50. Bob Hudson, owner of Computer Patch (2775 Maple Ave., Lisle 420-8861), demonstrated "Mind Wheel" and "Where is Carmen Sandiego?". The former is a sophisticated text adventure characterized as an "Electronic Novel". The latter is an elaborate detective type mystery with a delightful sense of humor, especially via the graphics. The locale is international, so to assist you as the investigator a World Almanac (the actual book) is included. Both sell for under $30. Ron Mireau of Software Centre International (1163 E. Ogden Ave., Naperville 355-7515) described "Dazzle Draw", a graphics preparation program with many advanced features. Unfortunately, Ron could not demonstrate the program fully; Double Hi-Res graphics capability are required as well as a Joy Stick or Paddles. And we goofed! A quick peek under the hood of our new Apple IIe just prior to Ron's presentation lead us to believe that we didn't have the required graphics capability. A closer look after the meeting told us we were wrong about that (Sorry, Ron!). The program lists for under $60. Incidentally, both of the stores represented offer substantial discounts.

Ralph P. Stein

A General Description of the Computer Language FORTH

Try to imagine a computer language which is extensible, structured, compact, transportable, interpreted, is as fast as compiler produced code, and which can be found in the public domain and on almost any micro you could imagine. This is FORTH.

Most of you have probably heard of FORTH by now. I wish to go into some detail about what FORTH is, how useful it is, and what you can expect from it if you learn it.

A program written in FORTH is simply a collection of subroutines, all called by a master subroutine. EVERYTHING is a subroutine, even the final program. Thus, it is possible to have many programs in memory at once. Each subroutine is given a unique name, and is called simply by mentioning the name somewhere, either by typing it at the keyboard or by including it in a definition of another subroutine.

I have said that FORTH is extensible. This means that any command may be altered, replaced, or renamed, and new commands may be added. As an extreme example, if many of your programs will require the use of N dimensional arrays of three A by B dimensional matrices of complex floating point numbers, you can make the necessary commands a permanent part of your FORTH system. Furthermore, it is not a very hard thing to design such a set of commands. On a more usual level, if you don't like FORTH's use of the name " ! " to mean "store a number at an address", there is nothing at all stopping you from renaming the command to "Store". Notice that lower case command names are allowed.

FORTH is a structured language. While there is no GOTO command, you could of course build one. There are, however, all of the usual looping and decision making commands. If you want more, you may add what you want.

FORTH is more compact than any other code I have to compare it to. Each command is translated into a list of addresses and data, which is given a name, and which is interpreted by FORTH. A program written in FORTH is usually about a fourth the size of a comparable BASIC program, and is even smaller when compared to a program produced by a BASIC compiler. A large FORTH program is even smaller than a comparable hand crafted assembly language program, since FORTH does not include any JSR commands in its code.

FORTH cannot be described as either a compiled language or as an interpreted one. It is really an "incremental compiler". At its core is a set of machine code subroutines, on top of which most of the rest of the language is built. The rest of the commands are sets of addresses of either the machine code commands or of each other. Commands are compiled one at a time, and each may be tested before the next is compiled. When executed, their address lists are scanned by an interpreter, which calls the machine language subroutines. The machine code gives FORTH its speed. Any command can be called directly from the keyboard, which gives FORTH its interpretive nature. FORTH commands may also be executed from a disk file, similar to an EXEC file for BASIC.

Each FORTH system contains machine code commands which have the same functions as commands of the same name on other FORTH systems. A FORTH source code can be transferred from one computer to another of a different type, and will run as expected if the machine dependent commands have been adjusted. Usually these are few.

FORTH has been installed in almost all micros and many minicomputers and mainframes. The Apple II owner has a choice of around ten FORTH systems from various companies, and the Mac owner has a choice of at least two so far.

One of the things which is usually mentioned in a discussion about FORTH is its use of postfix notation, or "Reverse Polish Notation". RPN is seen by some as a strong point against FORTH. I have talked with people who simply cannot understand it. Let me just say that if you work with FORTH for more than two weeks, RPN becomes automatic. I no longer notice it.

Another strange thing about FORTH is its use of two stacks, one for data and one for return addresses. The data stack is the preferred way of sending data to and from subroutines. This requires always knowing exactly where each bit of data is at every instant. Again, let me say that after a week or two, the use of the stacks becomes automatic. A FORTH programmer soon learns when to use the stack, and when to use variables.

On the subject of transportability, there are some things which should be cleared up. It might have ocurred to you that if FORTH can be modified, then each FORTH programmer will soon have his own unique FORTH system! Transportability goes out the door! This would be so, if FORTH was not standardized. Fortunatly for FORTH, it IS standardized. There are three FORTH standards, called FORTH-78, of FIG FORTH, FORTH-79, and FORTH-83. FORTH-83 is the newest, and is not found on as many systems as the others, but it is also considered to be a great improvement over the others by most FORTH programmers.

The way to keep your FORTH system standardized and still have it personalized is this: each time you create a permanent change in your FORTH system, record that change in a file, which contains every permanent change you made to your FORTH system. That way, if you recieve a FORTH program from someone else, you can run it on your origional FORTH system, and if you give a FORTH program to someone else, you also give them your file of changes, so that they may run your program on their FORTH system.

None of the FORTH standards include either graphics or floating point arithmetic, so not all FORTH systems have these. Most FORTH systems, however, come with optional extensions to cover these and other areas. These extensions come in files, and can be made a permanent part of your FORTH system if you wish.

Each FORTH system is guaranteed to have an assembler and an editor. Using the built in assembler, it is possible to include assembly language code right in with your FORTH code. Editors vary, some are crude line editors, and some are powerful screen editors.

How useful is FORTH? Forth is useful for about anything you could think of. Remember, if it doesn't do something you want it to do, you can add that to the language. Perhaps the only problem here is that if you don't know other FORTH users, you will have to expand your language on your own.

FORTH is ideal for robotics of all kinds. One of its first uses was to control the large telescopes at observatories. The makers of StarWars used FORTH to control their model starships during filming. FORTH is widely used in assembly lines.

FORTH is fast enough for things such as video games, animation, and real time anything. Anything which FORTH is too slow for may be handled by including an assembly language routine in the program.

FORTH is used for many number crunching tasks. Engineers and scientists use it often, both for its speed and for its versatility. In a laboratory situation, where experiments change often, FORTH is used because it is easy to modify and maintain a FORTH program.

FORTH's interactive nature greatly decreases the time spent programming and debugging. Usually, a FORTH programmer will test each new subroutine before going to the next. Most bugs are caught before they can do damage. This makes FORTH an ideal language for top down programming. Actually, the program is written from the top down, and is tested from the bottom up. This does not mean, however, that hackers are left in the cold. FORTH is a hacker's paradise. Subroutines may be tested by keyboard on the spot, without entering them into a file first.

Public domain versions of FORTH may be found on most computers. These are usually not as complete as a vender's FORTH system (perhaps no graphics and floating point support), but a public domain system is a good way to see if you like FORTH. Contact the Forth Interest Group for information about public domain FORTH systems. Their address is below.

There are three FORTH standards, and a few hybrid FORTH systems. Which you choose is up to you. FIG FORTH (FORTH-78) was the first FORTH standard. Most of the public domain versions are FIG FORTHs. These were developed by the Forth Interest Group. Most FORTH programs have been written in either FIG FORTH or FORTH-79. The FORTH-83 standard is the newest, and most of the future FORTH programs will probably be written in it. As an analogy, compare FIG FORTH to the Apple II, and FORTH-83 to the Mac.

Suppose you have found a FORTH system. Here are some things to check:

Is the editor a full screen editor or a line editor?
Is floating point supported? As an option?
Are graphics routines available?
Does the system have string handling commands?
(These are not hard to build...)
Are there commands which access the host operating system?
Does the system support all your hardware?
(Two drives, printer, etc.)
What happens when you press the reset key?
(Important if the system hangs up.)
Are there bugs?
Can it be returned if you don't like it?

Hopefully, you now have enough interest in FORTH to at least try it out. There are several places to start. First, check a computer store or a library for introductory books about FORTH. A good book for learning FORTH-79 is Starting FORTH, by Leo Brodie. A good book for learning FORTH-83 is Mastering FORTH, by Anderson and Tracy. A great source for any FORTH related material is the FORTH Interest Group, which is a nationwide FORTH user group.

FORTH Interest Group, P.O. Box 8231, San Jose, CA 95155, (402) 277-0668

Brian Dunn

DuPage Apple User's Group
Disk of the Month

VOLUME # 47
JUNE 1985

With summer here it's hard to convince oneself that relaxation and enjoyment are not life's primary purpose. To help you get over the disappointment that this realization might cause, the Disk-of-the-Month for June contains some R&R programs. Presented here for your summer enjoyment are a number of games and demonstrations. H-A-V-E F-U-N!

INSTRUCTIONS

1) BOOT THIS DISK (THE EASY PART). IN ORDER TO DO SO SUCCESSFULLY, YOU'LL NEED A 16K RAMCARD. A GOOD SELF-CENTERING JOYSTICK IS A MUST IF YOU WANT TO PLAY.

2) YOU CAN WATCH THE INTRO, OR PRESS A KEY OR BUTTON TO GET RIGHT INTO THE ACTION.

3) ONCE YOU START PLAYING, THESE ARE THE CONTROLS:
JOYSTICK - MOVES YOUR SHIP UP, DOWN, LEFT, OR RIGHT.
BUTTON 0 - LASER FIRE. HOLDING DOWN REPEATS.
BUTTON 1 - SMART BOMB. YOU'VE GOT THREE TO START WITH.
BUTTON 0 AND BUTTON 1 (AT THE SAME TIME) - HYPERSPACE.

THAT'S THAT. NOW PLAY!

DEFENDER:

JUST BRUN DEFL TO PLAY DEFENDER. THIS PROGRAM LOADS IN THE OTHER SEGMENTS (DEF1, DEF2, AND DEF3), RELOCATES THEM AND BEGINS EXECUTION. YOU'LL NEED A JOYSTICK WITH TWO BUTTONS. BUTTON 0 WILL FIRE A LASER (HOLDING IT DOWN REPEATS) AND BUTTON 1 FIRES A SMART BOMB. PRESSING BOTH BUTTONS WILL SEND YOUR SHIP INTO HYPERSPACE, REAPPEARING WHO KNOWS WHERE. CTRL-S TOGGLES THE SOUND, AND ESC PAUSES. IF YOU HAVE ANY COMMENTS/SUGGESTIONS, FEEL FREE TO CALL.

CUBES:

THIS IS A PROGRAM I WROTE TO 'FIELD TEST' A FEW NEW GRAPHICS TECHNIQUES I'VE BEEN DEVELOPING. NOTICE THE LACK OF FLICKER AND SMOOTH MOTION OF THE BALLS.

EXPLODE:

THIS IS AN APPLESOFT DRIVER FOR AN EXPLOSION ROUTINE (EXPDOT). CHOOSE AN EXPLOSION PATTERN AND WATCH. I USED THE SAME ALGORITHM FOR THE SHIP'S EXPLOSION IN DEFENDER.

LINES:

THIS ALSO IS A DRIVER FOR A MACHINE LANGUAGE PROGRAM. I WROTE THIS JUST FOR FUN. 'GAGDT' IS THE NAME OF THE PROGRAM I USE TO DESIGN ALL MY GRAPHICS (INCLUDING THE STUFF IN DEFENDER).

BOUNCING BALLS:

NOW WE'RE GETTING ANCIENT! I WROTE THIS BEFORE I KNEW ABOUT BLOCK SHAPES AND SO I USED THE HI-RES CALLS INSIDE APPLESOFT (Y'KNOW, HPLOT, HLIN, &C?). I GUESS THIS SHOWS YOU WHAT YOU CAN DO WITH JUST THOSE ROUTINES. PADDLE ZERO CONTROLS THE SPEED. BY THE WAY, THE BOUNCE ALGORITHM I DEVELOPED HERE HAS FOUND ITS WAY INTO CUBES AND DEFENDER (THE LATTER AS ACCELERATION/DECELERATION).

QIX LINES:

BRUN THIS AND WATCH THE LINES. I USE THE APPLESOFT HLIN FOR LINE-DRAWING. THIS IS BEST WHEN SEEN IN COLOR.

GUN SIGHT:

THIS IS ALSO PRETTY OLD (NOTICE THAT HORRIBLE FLICKER!). USE A JOYSTICK TO MOVE THE SIGHT, AND ESC TO ALTERNATE BETWEEN FULL VIEW AND SIGHTED. I KNOW THERE'S A GAME IN HERE SOMEWHERE.

YOU ARE HEREBY GIVEN PERMISSION TO COPY THIS DISK TO YOUR HEART'S DELIGHT. THIS GAME IS NOT COPY-PROTECTED, AND IS NOT BEING SOLD. IT IS PUBLIC DOMAIN. THIS GAME DISC WAS CREATED BY:

JOE HOLT
11033 ROCKWOOD PLACE
EL CAJON, CA 92020

VISIT OUR FRIENDS!

A number of local computer stores support our activity by offering the discounts listed below to those who show their membership cards. Stores that sell merchandise to everybody at a discount are not included.

Byte Shop of Darien, 8105 S Cass Av, Darien (960-1422) -- 10%
Computer Grove, 1121 Warren Av, Downers Grove (968-0330) -- 10%
Computer Junction, St. Charles Rd and Route 83, Elmhurst (530-1125) -- 15%
Computer Junction, 7 S LaGrange Rd, LaGrange (352-4700) -- 15%
Computer Workshop, 1626 W Ogden Av, Downers Grove (971-0004) -- 10% on training and rentals; none on consulting or programming
ComputerLand, 136 Ogden Av, Downers Grove (964-7762) -- 5%
Farnsworth Computer Center, 1891 N Farnsworth Av, Aurora (851-3888) -- 15%
Farnsworth Computer Center, 383 E North Av, Villa Park (833-7100) -- 15%

Frequently a smaller discount is offered if a credit card is used. If a store would like to be included in this listing, please write the editor and give us your discount schedule for Apple-User Group members.

*****

P O Box 294
Downers Grove IL 60515

****

The mailing label to the right is the only notice you will get that your membership is expiring. If you let your membership lapse, you will have to pay another initiation fee of $8, plus your $12 dues, to get back on our list.

8510
[redacted]