

# SYNTAX<sup>1.4</sup>

SEPT. — OCT. 1985

## TABLE OF CONTENTS

FCAUG Notebook.....	2
Letters.....	3,4,5,6
An Overview of CP/M.....	7,8,9
Adam Graphics.....	9,10
Book Reviews.....	11
Program Reviews.....	CP/M 2.2 and Assembler.....12,13
	Softpack 1.....14
	Electronic Word Book.....15
The Fun of Discovery.....	15,16
Programs.....	18,19,20



FIRST CANADIAN ADAM USERS' GROUP  
P.O. Box 547 Victoria Station  
Westmount, P. Q.  
H3Z 2Y6

POSTAGE  
PAID

Mtl. PQ  
Permit  
Pending

## The FCAUG Notebook

Fall has arrived and we are quickly approaching the last few months of what has been a productive first year for FCAUG. We have strived to bring you information which has hopefully helped many readers get more enjoyment out of their Adam. One of our priorities has been to cover material which you, the users have requested in your letters. Since we cannot possibly answer each letter personally, we try to incorporate most questions and suggestions into every issue.

Our plans and objectives for the upcoming year are already being formulated. A number of new and interesting topics are currently being researched. As always, we welcome your findings, ideas, and other contributions. One area where we would particularly like to see more activity is with SmartLogo. If you have used this fine programming tool extensively, why not send us a program or small article on it to be included in SYNTAX.

Some of you have shared knowledge which you have acquired through SYNTAX with other publications to the south. We welcome this but ask that you please state your source and ask that it be mentioned if used in another publication. Our writers work hard to report their findings to you. It is only fair that credit go where credit is due. On a similar note, we once again want to emphasize the importance of identifying the source of programs sent in which are taken from other publications or books. Remember there is nothing wrong in making others aware of the many great programs available as long as we know where they came from.

Some members have expressed interest in communicating with other users in their local area. Due to space restrictions, we will only list a couple here: Richard Smith, 115 Rue Bernard, St. Jean Sur Richelieu, Que. J3B 7X1 and Renald Blanchette of 593 Brassard in Chambly, Que. J3L 4N5. On page 6 is a list of some of the local Adam groups which have been brought to our attention.

Members interested in buying Adam books can contact Joanne Allen, 3142 Athol St., Regina, Sask., S4S 1Y7. Joanne is a partner in a bookstore and a trained librarian. If you are having difficulty in obtaining a particular book, she can help locate it and even special order it for you.

Our first program library is now available. Price is \$10 (includes postage and tape or disk). Briefly, it includes many graphics, games, interest calculation, mailing list, plus all the programs that have been printed in SYNTAX.

SYNTAX is published bimonthly by: First Canadian Adam Users' Group (FCAUG), P.O. Box 547, Victoria Station, Westmount, Que., H3Z 2Y6. Subscriptions: 6 issues for \$20.00. Second class mail registration pending. POSTMASTER: Send address changes and notice of undelivered copies to above address. Return Postage guaranteed. Copyright 1985 by FCAUG. All rights reserved.

## L E T T E R S

Dear FCAUG,

Thank you very much for the copy of SmartBasic on disk. It works fine except that I cannot get the turnkey program to work on it. The disk saves my program under HELLO but it does not run it when I boot SmartBasic. I can live without a greeting program but it does help boost my badly shattered ego. Is there anything I can do about this problem?

Donald G. Miller  
Ottawa, Ontario

This problem affects a very small number of Adam users. The group that have obtained from FCAUG or have made a personal backup copy of SmartBasic on disk. Unfortunately, there's not too much you can do on your own. You need certain software tools and specific knowledge of where to apply them.

When FCAUG originally started to make backups for people (see SYNTAX 1.1 and 1.3), we made a straight tape to disk copy of SmartBasic. However, we have noticed that turnkey programs stored on the disk will not work. Adam continues to look for the HELLO file on the tape drive even though the Basic program is loaded from the disk. The default drive remains tape drive 1 (dl). This wastes time for the user. You have to wait for it to look for a HELLO file (that doesn't exist) on dl which contains absolutely nothing. You then have to change the default drive to the disk by typing any O/S command (like CATALOG) followed by d5.

Before we get to the solution of this problem, let's look at the reason behind it. The problem begins as soon as you load the Basic interpreter (hit reset). When the screen turns black and the cursor is in the top left hand corner, Adam is programmed to look for and execute a HELLO file on dl. If a HELLO file is not found, a small program that prints "Coleco SmartBASIC V1.0" at the top of the screen is executed instead. This is part of the Basic interpreter program. When you boot SmartBasic from disk, even if you have a HELLO file on the disk, it will not act as a turnkey program. However, if you put a HELLO file on a blank tape and place it in tape drive 1 and then boot your Basic from disk, the HELLO file will be read and promptly executed from the tape. Obviously, what controls the turnkey routine is not the name of the program, "HELLO" as much as what drive the interpreter tells Adam to look for this file on.

We have found the place on the Basic tape and consequently where the instruction is stored in memory that tells Basic that it is on dl, when as in this case, it isn't. The Basic interpreter is stored starting at block 2 and continues for 19 blocks. It ends at block 21 on the SmartBasic tape. The second byte in block number 18 (decimal) sets the default drive. Change this byte from its preset value of 08H to 04H and write the change back to the tape and you will have solved the problem. In order to do this, of course, you need the proper tools: a program that can read

blocks from a tape or disk, display the bytes within that block, change them, and then write these changes back to your media (tape or disk). If you don't have this type of program, you can find one in Hinkle's "Hacker's Guide" as advertised in Family Computing magazine. If you want us to do it for you, just send us your Basic disk with \$2.00 to cover the return postage and we'll modify it. Any tape copying program can be modified to do this job. Good news, ours is now available for just \$15.00 including media and postage.

Dear Editors,

First of all, my congratulations on a very fine and extremely valuable publication. The information I've found through SYNTAX has already more than repaid my membership fee. Secondly, a question and request for information. In several past articles, CP/M was mentioned. Could you explain exactly what this is and what it enables you to do? I understand that it is an operating system, but does it allow you to purchase non-Adam software which uses CP/M?

Laurie Pearce  
Toronto, Ont.

Remember when you were told that Adam was Apple compatible? You probably thought, "Great! Now I can use Apple programs!". Wrong. Coleco's definition of Adam compatible pertains to programs, not software. Briefly, this means most Apple Basic programs can only be typed into the Adam and then run. The reason being that, the disk and tape formats are different for the two machines even though the "language" is the same. Here is an analogy. Imagine, someone trying to play a song recorded on a cassette on a record player. Sure the song is the same on both formats but it is the format itself that is important in this case, not the actual contents or data, which prohibit you from listening to the song. Or in our case to read the Apple tape or disk.

Since the operating system, memory maps, and screen width are also different on the two machines, other problems arise. Peeks, pokes, or calls for Apple programs will not work on Adam - even if you type the program onto Adam media. The difference in screen size (Apple has 40 columns) will mean that in order to run Apple graphics programs on the Adam, you have to modify the program so that it will fit on our narrower screen.

What does this have to do with CP/M, you ask? Nothing, but the problem is similar. CP/M programs will run on Adam as long as they are on Adam media. The Basic problem is easy to solve, all you have to do is type the listings in. With CP/M it's a little more difficult. You can, of course, type in the source code for CP/M 2.2 programs but that's a lot of work. The best thing is to get a modem and download the programs you want. Many public domain programs are exceptionally good and what's best is that they are free. (Which reminds me, if you know someone with an Apple, why don't you get him/her to download some Apple Basic

programs to you as well.)

A word on downloading: AdamLink 2 does not support CP/M. What you have to do is download the source code for a CP/M modem program, assemble and load it. Then you can use that program on the Coleco modem to download CP/M programs. The best known public domain modem program is Modem7. It is available from CP/M users' groups and BBS as well as some Adam BBS. See page 6 for the telephone numbers for the Toronto Adam Club's bulletin board or Canada Remote (the best source of public domain software in Canada) if you need a CP/M based modem program.

You can buy very sophisticated CP/M packages. They are usually very expensive, however, and the only source for these types of programs; Westico has just dried up. More on this as the picture unfolds. CP/M is the best thing that ever happened to the Adam, as there are literally thousands of free CP/M 2.2 programs for you to use. This includes language compilers like C, Fortran, Basic plus programs for word processing, spread sheets, etc. The list goes on and on. I urge everyone with an Adam to explore this aspect - you won't be disappointed. As far as your first question is concerned, refer to the CP/M articles in this issue for more information on CP/M in general and the Adam version of CP/M in particular.

Dear FCAUG,

I have an Adam - as do probably most of the people who write to you and I need HELP! I need to know what the different "CALL" commands will do. When I tried to make a program to get Adam to tell me this, it erased my SmartBasic tape. The main thing I want is a program or explanation of CALL, and how to change the screen and background color.

Hugh Johnson  
Lestock, Sask.

Hugh, do you know what a GOTO or a GOSUB is? If you do, you are 3/4 of the way to knowing what a CALL is. If you don't, I'll briefly run through these.

A Basic program is a set of statements within numbered lines. Simple programs are executed in a fixed (numerical) order. The statement with the lowest line number is carried out first, and so on. When a different order is required "jump statements" and "subroutines" are used. These allow the program flow to jump out of its sequential pattern in response to a yes or no, true or false choice. The GOTO sends the computer to a particular line and unless it encounters another GOTO command, it will continue executing line numbers sequentially starting at where the GOTO told it to go. GOSUB does the same thing, but when it encounters a RETURN, the computer jumps back to the line number immediately following the GOSUB statement in the main program. GOSUB's are used to perform a specific chore, if required, and then jump back to the main program flow once this operation has been completed.

RETURNS are always found ending a GOSUB routine. These routines help to make neat and easy to follow programs. Keep this in mind when you use these GOSUB routines. It's easier to lift a small subroutine like this out of a program and then use it in another one, than it is to muck about in a program without GOSUB's trying to find the information you want.

The CALL statement, like the GOSUB statement, is a general command. I say general here in response to your inference that there are different types of CALLs that do different things. You CALL to a certain location and start executing the instructions there until you encounter a RETURN. The point here is that the way a CALL and a GOSUB work is no different. With a GOSUB you go to a particular section of the program. It is the instructions which are at that subroutine that make the GOSUB do what it does. The same applies with CALLs. When you use a CALL you send the program not to a specific line number but to a particular memory location. The program executes the instructions numerically starting from the first memory location and stops when it gets to the symbol for "return" in one of the memory locations. If you check the sound programs in this and earlier issues, you will notice that the last number in some of the lines is 201. This is the equivalent to the return symbol that I am referring to.

As you can see, the only difference between CALL and GOSUB is that GOSUB is used to send the program flow to another section of your program written in Basic and specified by a line number, while CALL sends the program to a section of the program written in machine (or as near to machine as possible) language and specified by a memory location. Page B-7 of the Basic programming manual contains a very brief description of CALL. CALL is really an assembly language statement added for machine language access in Basic. As we have seen, it is the assembly language analog of GOSUB. GOTO also has an analog in assembly language. It is JMP or JP for jump. But let's not stray too far off the main subject. Obviously, you have to know something about machine language to write a subroutine that you CALL to. That's a little beyond the scope of this answer, but at least you know how to get there.

As for your question regarding screen and background colors, this topic has already been dealt with in detail in previous issues of SYNTAX.

Some Local Canadian Adam User Groups and Sources of public domain software

Montreal Adam User Group  
C/O - S. Evans  
7460 Ouimet  
Verdun, Que.  
H4H 2K2

Hamilton-Niagara Adam User Grp  
C/O - B. Zaruk  
#3 - 125 Livingston Ave.  
Grimsby, Ont.  
L3M 4S5

Toronto Adam Club  
3346 Charmaine Heights  
Mississauga, Ont.  
L5A 3C1

Metro Toronto Adam Group  
260 Adelaide St. E., Box 123  
Toronto, Ont.  
M5A 1N1

Adam-Net BBS (assoc. w/ TAC)  
2 Edna Ave.  
Toronto, Ont. M6P 1B5  
(416) 594-2841 (modem)  
Bill Freads - SYSOP

Canada Remote  
4961 Dundas St. w.  
Toronto, Ont. M9A 1B6  
(416) 239-2835 (audio)  
(416) 231-0538 (modem)

## An Overview of CP/M

CP/M is one of the most expensive software packages that you can buy from Coleco. It is also one of the most puzzling for the first time user. Once understood, however, CP/M is the most powerful software package you could ever buy for your computer.

First off, CP/M is not a language like SmartBasic or assembly language. It is an operating system (O/S for short). And what is an O/S you ask? The O/S is the underlying workhorse of your microcomputer. Note that there are other types of O/S's other than CP/M. CP/M is one of many. We'll look at some of the others first. This will make it easier to understand CP/M in the end.

Nothing can hide its presence so well yet be so important to the functioning of a PC like an O/S. Did you know that Adam comes with its own O/S already built-in? In fact, you have been using Adam's O/S (called EOS) ever since you first turned on your unit. So, let's look at this O/S first.

The EOS very quietly controls all the major computer operations: the screen input and output, printer output, modem input and output, and the operation of the disk and tape drives. In other words, the EOS responds to the user (you) when you tell it to put, for example, a certain string of characters to the screen or on tape. When storing to a tape, it is the EOS' job to find space on your tape for your file, open it up, write to it, close it, and then provide this information to the file directory so that the computer can find this file next time you want to look at it. But think what you have to do when you want to store a file. In Basic you type SAVE <filename>. In word processing you press the STORE/GET command key and then follow the required steps to store your file. It is the EOS that takes care of all the details. It does all the hard work.

The EOS consists of a series of control programs (drivers) for the various components in your system. These drivers are written in assembly language to do very specific jobs. So specific, in fact, that a driver written to control the sending of a single character to the screen will work on just one machine - the machine that it was written for. Manufacturers know this and to make a peripheral compatible with another computer they have to know what parameters the driver program is looking for. So you see, it's not only a physical barrier that disallows the mixing of components from one computer to another. Of course, Adam users have pretty well had a closed system. This will change very soon however, so what I'm saying is not just theoretical.

Besides having machine or system dependent O/S's, there are also CPU dependent O/S's. These O/S's are independent of the specific computer being used. Although it may not immediately look like an improvement over machine specific O/S's, CPU dependent O/S's are. You will see why soon. This is the type of O/S that CP/M is. It stands for control program for microprocessors. There are other machine independent O/S's called MS-DOS and UNIX. They do not work on Adam, only CP/M does. They work in more powerful "16 bit"

microprocessors. Adam uses the 8 bit Z80A microprocessor for its CPU.

When microcomputers were just starting out in the 70's, O/S's and CPU's were a lot simpler. One register CPU's and single drives were common. In fact, teletype machines were for the most part adapted to be used as terminals. To this day, CP/M still uses archaic teletype terms to describe devices that would be more logically called by more up-to-date names. As PC systems became more complex and sophisticated, it became increasingly important to have a good O/S and compatability between other machines. What was needed was a standard. Since an O/S is intimately linked to its CPU and Z80, precursor microchips were common. CP/M was developed for use on the "8080 family of CPU's". All computers that use an 8080, 8085, Z80, or any member of the 8080 family can run CP/M. Each improvement in microchip technology upgraded the 8080, but programs (like the CP/M O/S) made to run on the earlier chips could run on the later ones (called upward compatability). Unix and MS-DOS are advanced O/S's developed for more advanced chips. CP/M started it all and is still growing but it can't grow beyond the limitations of an 8 bit CPU environment. Unless you are doing highly sophisticated scientific work, this is not too much of a setback (no matter what MS-DOS and UNIX users tell you).

CP/M quickly became a de facto standard. It wasn't planned but it soon became apparent that if other computers with the same microchip used the same O/S, a standard would be arrived at. This compatability is CP/M's greatest attribute. There are a lot of Z80 computers out there. That means programs written to work on the O/S of one machine will work on a completely different system assuming they are both running CP/M. The O/S "looks" the same from one machine to another. If one machine has Microsoft Basic, for example, not only will the Basic programs run on another machine but so will the Basic itself. Realize that the O/S is the thing that controls the Basic. It is one level higher (in terms of control) than the language.

What this means is that now there is no limit to the amount of programs, languages, and software packages that you can run on the Adam. The disks and their formatting may not be compatible with Adam but the programs are. To get these programs you can get a CP/M based modem program, run it through the Coleco modem, and after connecting up to a CP/M bulletin board, download the public domain programs. That's why CP/M is such a powerful tool.

#### CP/M: Getting a proper fit

Since it was first developed in 1976, CP/M has been constantly modified and upgraded (upward compatability has been maintained). That's why it's known as CP/M 2.2, meaning version 2.2. Besides being upgraded and changed by the company that created it, CP/M has been changed for its use on the Adam. This last statement requires further explanation.

CP/M is an O/S. That we know. If it looks the same on different computers and all these computers are different, then how can CP/M be the same on all these other machines? After all, I did



say that each driver is machine specific... Good point. Well this is how it's done. CP/M doesn't rewrite driver programs, but it can access them and in that way control them. Thus someone has to "fit" the CP/M onto a specific computer. A section of CP/M is called the BIOS. This small section acts as a link between the universal CP/M environment that the user sees and the actual machine specific driver routines that no one but the people who know the specific ports and internal do-dads to connect to know. To make this a little clearer, take the example of an electrical appliance and an attempt to get it to run somewhere in Europe. Non-North American countries use a different current to run their household appliances than we do. This doesn't prevent them from using similar electrical appliances. And if you move and want to use an appliance from here over there, you have to use a voltage transformer. The transformer is placed between the appliance and the wall socket, making the appliance function in a normal matter in a new environment: The BIOS is the interface between the CP/M and the PC, just as the transformer is the interface between the displaced electrical appliance and its new electrical source. Everything else, the utility programs and environment, is the same program from one machine to the next.

#### Adam Graphics

SmartBasic's TEXT mode allows us to print things as NORMAL, INVERSE, or FLASHing characters. Let's see how it does this.

How the character shape is defined: SmartBasic works with the ASCII character set. These are character codes 0 to 127. It contains upper and lower case letters as well as the various punctuation marks. The shapes for all 128 ASCII characters are stored in a ROM in the ADAM. SmartBasic as well as most other programs use this ROM for their character sets. In SmartBasic, the 128 ASCII characters are white on black. This set is repeated as character codes 128 to 255 but this time the characters are black on white.

How the character color is defined: The black on white and white on black info is placed in the color table of VRAM. Each color byte controls one group of eight characters' color pattern. For example, the first byte in the color table gives the foreground and background colors for character codes 0 to 7. Therefore there are  $256/8=32$  color bytes in the color table. To see how to set up a color byte you should refer to the Color Pokes article in issue 1.1. Note, however, that we were changing the color pattern not for one character or even one group of eight characters, but the entire character set (0-127). One small correction in that article should be made in the color to color code chart at the bottom of page 14. The first black entry should be changed to transparent. Normally the border of the screen is black so with a color code of zero, one would get the color of the black border, but if you change the border color then you will see this new color instead.

NORMAL, INVERSE, or FLASH: When you print NORMAL characters to the screen, the VDP places ASCII codes (0 to 127) into the screen

table. If you want INVERSE characters, 128 is added to the ASCII code (giving character codes 128 to 255) and this new code is put into the screen table.

To get FLASHing characters requires BASIC to do a little more work. It needs 2 screen tables in VRAM. Of course only one screen table can be used at a time. BASIC switches the starting address of the screen table to these 2 tables at a constant rate. When a NORMAL or INVERSE character is printed to the screen its character code is really placed in the proper position of both of the screen tables. This way when the screen tables are switched, the characters on the screen appear steady. To get a FLASHing character one of the screen tables will contain the NORMAL character code while the other one will contain the INVERSE character code. This way when the screen tables switch then so does the FLASH character on the screen. All this allows SmartBasic to do some interesting things in black and white (or any other two colors you select) but what if we want to use more than 2 colors?

Experimenting with Adam's character and color set: Included in the program section is a graphics editor. It will let you "draw" new character patterns, change character's foreground and background colors, as well as place these characters on the screen to see how they look. One of the problems with this program is the use of the PRINT statement. Even when you blank out the cursor you still have remnants of it. You will notice that some of the characters put on the screen in this program will mysteriously go white on black. This has something to do with the cursor routine in BASIC and I don't fully understand it yet. The only way I can think of correcting this problem is by poking the character directly into VRAM (using the VWRITE machine language routine). This would require writing the character to both screen tables which are at Video addresses 2048 and 6144. I welcome any other solutions.

The machine code routines in the program are the VREAD and VWRITE routines we have used before. You create new patterns by moving a cursor on an 8 by 8 grid. Pressing the left fire button will turn a dot on while pressing the right one will turn it off. Moving the joystick moves the cursor. When you're finished drawing press the 1 button on the keypad, you will see the 8 decimal bytes that would need to be poked into the pattern table to create your shape. At the bottom of the screen will be a selection menu, the numbers there correspond to the keypad numbers. If you select the color option, it will prompt you for a color set code. Try entering one and you'll be given the character codes it affects. You are then asked to enter the color byte for that color set. The screen option allows you to place any character on a small part of the screen to see how it interacts with other characters.

This program could use some improvements like easier data entry for example, but it should prove useful for both programmers and non-programmers alike. The programmers can get the information they need to create graphics in their own BASIC programs, and the non-programmers can create colorful pictures on the small screen. I hope to use all the tools we have acquired thus far to create an interesting game for the next issue.

## BOOK REVIEWS

The First Book of ADAM the Computer by Arthur Dent, 1984. Blue Ridge Summit, PA. TAB Books. 207pp. including brief index. Paperback. Reviewed by Greg Slade.

Arthur Dent (No, I'm serious, this isn't some take off on the Hitchhikers' Guide to the Galaxy!) has written an interesting and useful, if somewhat confusing book on using and programming the Adam. Dent divides his book into five chapters. The first, "In the Beginning" is a combination advertisement for history of Coleco and the Adam. It has its fascinating moments. (Did you know that Coleco was originally the Coleco Leather Company?) But, again, it's dated in that it assumed that Coleco would stick with the Adam. (Who knows, if things were different, they might have brought out a second generation and called it "Seth"!) Chapters two and three, devoted to setup and word processing, respectively, don't add much to Coleco's documentation which comes with the machine. It's chapter four, "Introduction to Programming with Adam" which makes the book worthwhile. Over half the book is contained in this one chapter. It gives an extensive list of commands available in SmartBASIC, including commands which are used to control the machine language subroutines. But more interestingly, commands and ASCII values for using the game controllers, the control keys and the SmartKeys are provided. The last chapter is on software and is hopelessly optimistic (and hopelessly out of date). But it should give you a good laugh.

Mastering the Coleco Adam by Ken Porter, 1984; New York. Plume Books (New American Library of Canada, 81 Mack Avenue, Scarborough, Ontario, M1L 1M8). 175 pp. including index. Paperback. Reviewed by Greg Slade.

Many books claim to be "easy-to-use", but up until I found Ken Porter's book, I could not find anything which was more helpful than Coleco's copious (even redundant) documentation. But now, however, there is a "how to" book on the Adam designed for the true computer illiterate (or the computerphobe). Porter actually starts with "Getting Adam Home", and surely there can be no book that covers the Adam so carefully. He advances step by step, from unpacking the Adam to plotting circles in hi-resolution graphics (plus explaining the secret of Adam's mysterious "color bleeding"). There are some drawbacks to Mastering the Coleco Adam. First, it was published in August '84, when Coleco was still building the adorable little thing. This means that Porter was unaware of Adam's seemingly dark future on the one hand and the peripherals released in the past year on the other. Second, like most introductions, it doesn't take you very far. (There wasn't much in it that I didn't already know). Nevertheless, if you know someone who's afraid to touch their Adam, or if you know where you can pick up an Adam cheap, and want to bring Uncle Harry into the computer age for Christmas, get them this book as well.

## PROGRAM REVIEWS

### CP/M 2.2 AND ASSEMBLER

One of the good things about CP/M is that not only do you get the operating system but that you get several very useful utility programs with it. You have all the programs necessary to program in 8080 assembly language, copying programs, and programs that allow you to tailor the environment to your specific needs. As you may recall the EOS (Adam's O/S) has no comparable programs available. Another plus for CP/M.

This is not actually all that surprising. Once an O/S like CP/M was created to provide inter-machine compatibility the next thing to work on would have been to increase its ease of use. CP/M has been around for so long that this problem has been eliminated: You get all the key programs as utility programs when you buy CP/M. Many more are very easily available. So here's the kicker. Believe it or not our good friends at Coleco have decided to make a few changes with the standard CP/M utilities. Every CP/M except for Adam's has a "MOVCPM" function and a proper "SYSGEN".

These are programs that are used to work on making changes to the O/S itself. True, not everyone is going to use them at first but they are part of the CP/M package. They have to be there. SYSGEN is used to put the O/S or "system tracks" onto a new disk. It reads the source from one disk and relays it to another. MOVCPM can be used to work on the system and then SYSGEN can take the modified system and place it on a new or the same disk. SYSGEN starts by asking where you want to get your source O/S from. If you press <return>, it skips this procedure. Assuming you have then moved your modified O/S into the proper place in memory, you can just shoot your new O/S to new system tracks, when the second part of SYSGEN asks you to what destination you want to send the program lying in the SYSGEN memory area (hopefully your modified O/S, most often your unmodified one).

That's the way SYSGEN is supposed to work. Instead of skipping the source retrieval phase as above when you press <return>, this SYSGEN leaves the command and returns you to the system. And if that's not bad enough, Coleco has made it hard to modify SYSGEN. (Too hard to get into just right now.) For some reason Coleco has deliberately tried to make any attempt to really get into the system as hard as they possibly could. It certainly seems that they wanted this version of CP/M to be untamperable. And I think I know why. Coleco wanted CP/M purchasers to buy the CP/M based programs available for it from Westico, whose catalog is enclosed with the software package. These programs are great, but are also very overpriced. Since then Westico has dropped out and another source of CP/M software for the Adam is expected to be announced. It appears Coleco wanted buyers to use but not tamper with CP/M. I think that this plan will backfire very soon, especially when more and more people turn their interests to these problems.

The manual has been described as being, alternately, the greatest and the worst. Contradictory? No. It just depends on who you ask.

Someone new to CP/M will go through the first part of CP/M very easily. It tells you how to move files around, get information about your files, and nice housekeeping chores like that. You finish that session saying, "Great, but what can I do with it?". The next part of the book is great for experienced computer users fluent in CP/M and assembly language. It is incomprehensible to the average user. If you don't already know CP/M and assembly language, my advice is to buy and read one of the books that they at least had the courtesy of mentioning at the end of the first part of the instruction manual. (Try Rodney Zaks' CP/M book on Sybex. It puts CP/M into a little better perspective.) If you are already familiar with the subject, then the next kind of book to get is a CP/M assembly language book. This is where you really put CP/M to use. All programs are written in assembly language. When you use your Adam, you program in Basic. Remember, however, that someone had to write SmartBasic in assembly language first. The whole key to CP/M is assembly language programming.

Assembly language programming is made very easy with CP/M. Instead of having to know all the different access ports to make a peripheral do a particular task, all you have to remember are "function numbers". Needless to say, these numbers are the same for all CP/M computers. Since the BIOS is already connected to the right drivers, all you have to do is load one register with the right function number and the desired driver is activated. Fortunately, Coleco didn't make any changes here.

Coleco's version of CP/M 2.2 is not all bad. Many people will get this package just so they could run the myriad of application programs written for CP/M out there. And that's fine too. It just seems that things that should have been there were deliberately removed. I mean, why go through all that trouble just to make it harder for people? OK, enough complaining about Coleco. There are some very redeeming features about this package. Two new commands have been added just for Adam users which are very useful. "ADAM" changes an Adam file to a CP/M file. This is great for assembly language programmers because they can forego using the tedious "ED" command and type their programs with the word processor. And just when I was starting to compliment Coleco, what do you know? "ADAM" doesn't work. This command changes a W/P file into a CP/M file all right, but when you look at your file, no line feeds are placed at the end of each line. That means that each line is written over the top of the next one and so on until the listing ends. Coleco does provide a fix to this problem. Unfortunately, there are some problems with this fix too. First, nowhere in the book are you told where the fix is. Second, when you do figure out that Appendix D: Adam Filter Program, on page D35 is the fix, you realize that you have almost 6 pages of typing to do just to get the thing to work. The other command is "CPMADAM". It, you guessed it, changes a CP/M file into an Adam file.

Even after all this criticism, I still feel that this CP/M 2.2 and Assembler is a good package. No thanks to Coleco though. They have made some very questionable changes in CP/M. But looking on the bright side, I can see that this will give us plenty of ideas for CP/M fix-it tips and articles in future issues of SYNTAX.

## SOFTPACK 1

Manufacturer: E. & T. Software, P.O. Box 821242, Dallas, Texas  
75382-1242, Tel. (214) 340-8913

Price: U.S. \$18.00

SOFTPACK 1 is a package consisting of four programs: Softcheck, Checkbook Reconciler, Checkbook Totalizer and Softmailer. The instructions to each program are also contained on the tape/disk. To familiarize yourself with the programs you'll need a printout.

Softcheck is a personal checkbook manager designed to help you organize your account by monitoring checks, deposits, automatic teller machine withdrawals, interest earned, check fees and other bank service charges. Files created by Softcheck are used with both the Checkbook Totalizer and Checkbook Reconciler.

Softcheck is very easy to use. It is menu driven, so all you have to do is select the option you want and then enter the details of your personal transactions. You start off by specifying the start date, initial balance, fees per check, and file name. Once you've made a few entries to your file, you can save it. As you update your file, you will need to resave it and delete the old file. If you have a very high volume of transactions, it would be best to create separate (monthly, bi-monthly, or quarterly) files for shorter periods. The authors recommend that you save your revised file under a new name each time due to a bug in SmartBasic. I did not encounter any problems when using the same name. You can review your complete file on the screen or obtain a hard copy with the printer at any time. With the Edit/Review option, all data that is incorrectly entered can be modified or cancelled.

The Reconciler program is used to reconcile your current checking file with your bank statement. After entries accounting for all outstanding amounts and interest are made, selecting the Review option provides you with a detailed comparison of the difference that exists and where you must apply it to balance.

The Totalizer allows you to obtain a total by month or of the entire file of any field or category (food, gas, clothing, etc.). All the tax deductible expenses you marked in Softcheck can now be listed and totalled simply by entering "TAX".

Softmailer is a mailing list program that will let you maintain and print a mini database (maximum 120 records) of names and addresses. You specify the search field (first/last name, record number, code, etc.) you want to use when printing. One thing which Softmailer does not do is to alphabetize your entries. It does, however, number each entry automatically.

I found SOFTPACK 1 to represent good value for the money because the programs are well designed and practical. Plus they can be easily modified for your personal needs or other applications.

## Richard Scarry's Best Electronic Word Book Ever

This educational software package, geared for the 5-8 age group, gives a good indication of the high quality products Coleco was capable of delivering had it decided not to pull out of the home computing market so prematurely.

What we have here is the talent and creativity, of world-famous children's writer and illustrator Richard Scarry, brought to life on the screen through the magic of computer animation.

Using the hand controller, the player guides the main character, Lowly Worm, on a delightful journey full of fun and discovery involving words and objects in six different scenarios. Switching to a different environment can be done any time during the game. Each one is a stunning representation of either a farm, park, railroad, construction site, harbor, or a town. Four distinct skill levels are provided. In skill 1, preschoolers can explore the surroundings by guiding Lowly Worm's Apple Car along the road and discover how illustrated words come to life at every stop it makes. In skill 2, the young player must guide Lowly Worm towards the object that matches the one appearing at the corner of the screen. Skill 3 begins to challenge the child's word and object association skills. Here, the object matching the word on the screen must be found. A correct match in both skill 1 and 2 will result in the object coming to life and its name appearing next to it on the screen. Skill level 4 involves a scavenger hunt. A list of twelve words is presented to the young reader who must then search through all six environments in order to find the corresponding object. Switching to another skill level and then returning to level 4, will generate a new list of twelve objects. By playing this learning adventure game, your child can develop object recognition skills and a stronger interest in expanding his or her reading abilities. The detailed colorful graphics, amusing melodies, and the sight of objects coming to life at the child's discretion help to make it as much fun and challenging as educational.

One minor inconvenience with Electronic Word Book is the rather long time it takes to load the next environment when a change is selected. But this might be to the child's advantage giving him or her ample time to adjust to the next situation he/she is about to explore. As an interlude, during loading, the screen shows Lowly Worm taking a spin around a "Game Map" (selection menu). In conclusion, we find Word Book affordable and a good starter in building a family learning software library for any Adam user with very young children.

## T H E F U N O F D I S C O V E R Y

If you've ever looked at the back of any game cartridge instruction manual, you've probably seen it; "The fun of Discovery". But what exactly does it mean? Well, if you play your favorite game long enough you just might find out. You will find that under certain circumstances odd things happen. Now it's debatable as to whether these "peculiarities" are actual bugs or hidden surprises intentionally left there. Nevertheless, they do exist and in this and future issues of Syntax, we'll expose some of the known and not so well-known discoveries. So come and experience the fun of discovery - you may surprise yourself!

We start off with Donkey Kong (cartridge format only). This game was the first Colecovision cart so there has been plenty of time to discover its secrets. And are there ever a lot of things to discover in this cartridge. For just that reason then, we'll only cover a few DK tricks in this issue.

i) the broken ladder jump

This is a fairly well-known trick but it is one of the better ones so, we'll look at it first. Move Mario up the first broken ladder and then bring him back down. Walk him to the left so that his back is almost touching the same broken ladder. Now move him a step or two to the right and jump. What happened to good old Mario? Now depending on whether you own a recent or older version of the cartridge, either of two things will happen. With the more recent version, Mario falls through the bottom of the screen. He lands safely at the beginning of screen 2 with full bonus points awarded. With an earlier cartridge, Mario also falls through the first girder. But this time he falls and completely disappears off the screen. Meanwhile, the barrels keep rolling down. Ten seconds later Mario comes bouncing back into the picture at the top girder right next to Donkey Kong. You then have to get him out of there lickety-split so that he can climb up that last ladder. This is a little more dangerous than on the later version because if you happen to land on the girder at the same time as Donkey Kong is releasing a barrel, you're a goner!

ii) the ladder flip

You don't earn any extra points, or anything for that matter, when you do this one. In fact, you end up a little further away from your goal. Nevertheless, what happens is interesting because it should be impossible and isn't. First, you have to get to the third (elevator) screen. Then get to the top right hand part of the screen, to the tier where the purse is located. Below you is a short ladder. Position Mario right above it. Now wiggle the joystick up and down very quickly. Mario has fallen through the metal floor and ended up on the tier right below where you were. Even better, Mario's OK and ready to go.

And here's one last one from Mr. Do. It's more of a quirk than a beneficial trick but here goes. Try and crush Mr. Do under an apple. What? Yes, that's right except if you do it right it's not suicide but everlasting life. Try dropping 2 adjacent apples and then getting Mr. Do crushed under the first one dropped. You will crush him so that he looks like an accordion. His power ball will remain where it was and he will land right under where the apples land. Mr. Do ends up looking like a pancake, yet play continues. If he's only dropped a bit he can be eaten by a bad guy. But, if dropped from a higher level, Mr. Do becomes impenetrable. This acts as a sort of permanent pause. The action continues but nothing affects Mr. Do. He can't be eaten or harmed in any way by the bad guys. Unfortunately, he will no longer respond to your promptings either. The only way to continue playing is to reset the game. Not a great trick but good for a laugh at least.

Here's a tip. When playing a cartridge look for things like the ladder flip. Keep an eye out for things one on top of each other that are broken somewhere in the middle (i.e. ladders, vines, ropes, etc.). Quite often this continuity is there for a reason.



ADAM  
USERS!

ADAM  
USERS!

# smartBASIC BONANZA!

## 15 programs for ADAM on one cassette

### OUTSTANDING REVIEWS

"smartBASIC BONANZA is the best... You will never spend \$34.95 more wisely."

— Expandable Computer News

"fun... useful... amazingly low cost"

"... worth every cent."

— ADAM Users Club

"... fine programs... well written and appealing."

— AUGment (ADAM Users)

— Computer Entertainer

**DESIGN:** hi res figures

**SOUNDER:** music and sound

**OTHELLO:** the board game

**MANSION:** adventure game

**FINANCE:** budget, metric,

interest projections

**FUGUE:** 3 instrument music

**MAGIC:** amaze your friends

**TYPYR:** tutor, video game

**TRYME:** 2 educational games

**MINIASSEMBLER:** write

machine code

**DISASSEMBLER:** decipher

machine code

**FILER:** database

**LABELS:** make labels from

FILER files

**TENNIS:** pong game

**BREAKOUT:** video game

+SURPRISES

# FANTASY GAMER



## 3 BIG PROGRAMS FOR THE ADAM\*

**THE VISITOR:** Interactive fiction with animated graphics. Your smart but odd companion must rendezvous with its mother ship.

**BOMB SQUAD:** Graphic adventure. Find the terrorists' bombs in time.

**ADVENTURE CREATOR:** Write your own adventure games. Instructions, "framework" program, graphics subroutines, fast machine language parsing routine.

READ THE REVIEW FROM COMPUTER ENTERTAINER

JUNE 1985 (12115 Magnolia Blvd., No. Hollywood, CA 91607)

ADAM FANTASY GAMER (★★1/2/★★★1/2) from *MacLUG CONSULTING* of Canada includes three programs: two text adventures with graphics and an "Adventure Creator," which is the real star of the show. The programs are written in SmartBASIC, but the parser (the portion of the program that analyzes the player's keyboard input) is in machine language. This allows faster analysis of commands and use of more words in the game vocabulary than adventure games written entirely in BASIC.

**The Adventures**  
Ideally, you should play and solve both games before moving on to the Adventure Creator, since working with the Creator reveals many "secrets" of the two adventures. The first game is "Bomb Squad," an intermediate-level adventure in which you must find and deactivate three bombs planted by terrorists in the embassy of Lunaria. The second is a fantasy adventure, "The Visitor," in which you befriend an alien being and help him find his mother ship. Both

adventures include illustrations with some animation. These are not mere decoration, since important clues are revealed graphically. Both games are well written and compare favorably with the few other adventures available for Adam.

### Adventure Creator

The best reason for owning this program is the Adventure Creator, which is well supported with a detailed instruction manual. Creating an adventure from scratch is an involved and difficult task that is made easier with Adventure Creator. The program assumes that you understand BASIC programming, and you will have to put that knowledge to work. The manual guides you through the process of mapping your game, provides a listing of the Creator program, explains the operation and application of the parser, and teaches you the use of sprites to create animated graphics. (For some Adam owners, the section on sprites alone makes the program a worthwhile purchase.) When you have completed your first adventure with the help of the Creator, you will have learned more about programming than you knew before you started. You'll have fun, of course, but the knowledge gained is a lasting benefit of the program. (Solo use; Keyboard; Digital data pack.)  
*Recommended: (MSR \$34.95)*

EACH CASSETTE ONLY  
\$34.95 (US), \$43.95 (CDN)  
Money Order, VISA  
MasterCard (include  
expiry date)

Martin Consulting  
94 Macalester Bay  
Winnipeg, Manitoba  
R3T 2X5 Canada  
(204) 269-3234

ADAM and smartBASIC T.M. Coleco, Inc.

# Snake Pit - by Andrew Wiles

```

100 LOMEM :29000: TEXT: POKE 16953, 32
105 POKE 28000, 58: POKE 28001, 102: POKE 28003, 211
107 POKE 28004, 255: POKE 28005, 201
110 hiscore = 0: bcolor = 1: rcolor = 12: fcolor = 15
120 INPUT "Do you want instructions? (Y/N) : "; an$: an$ = LEFT$(an$, 1)
130 IF an$ <> "Y" AND an$ <> "y" THEN 180
140 PRINT: PRINT SPC(10); "SNAKE PIT"
150 PRINT "Move snake around maze, trying not to hit any walls or your"
155 PRINT "snake body."
160 PRINT "Try to eat white food blocks for 20 points each."
170 PRINT "The snake grows longer with each food block eaten."
180 PRINT: INPUT "Play level 1 or 2 ? : "; an$
185 an$ = LEFT$(an$, 1): IF an$ <> "1" AND an$ <> "2" THEN 180
190 lev = ASC(an$)-48: & get level of play
200 PRINT: PRINT "Press trigger to continue"
210 IF PDL(7) <> 1 AND PDL(9) <> 1 THEN 210
220 & set up variables
230 size = 1000: DIM s(size, 2)
240 score = 0: snakes = 3: food = 10: delay = 80
245 x = 1: y = lev*10: dir = 4
250 & set up screen
260 GR: PRINT " SCORE "; score: PRINT SPC(10); "SNAKE PIT"
270 PRINT " SNAKES "; snakes; : HTAB 20: PRINT "HI "; hiscore
280 COLOR = bcolor: HLIN 0, 39 AT 0: HLIN 0, 39 AT 39
290 VLIN 0, 39 AT 0: VLIN 0, 39 AT 39
300 & place inner barriers
310 FOR i = 1 TO 3*lev: READ x0, y0, x1, y1
315 HLIN x0, x0+3 AT y0: VLIN y1, y1+3 AT x1: NEXT
320 DATA 4,4,34,4,31,34,4,34,8,14,18,24
330 DATA 19,7,16,5,24,18,14,16,15,33,6,20
340 & place snake on screen
350 FOR i = 0 TO 9+(lev-1)*10: s(i, 0) = 1: s(i, 1) = i+1: NEXT
360 head = lev*10: tail = 0: COLOR = rcolor: VLIN 1, lev*10 AT 1
370 & randomly place 10 food blocks
380 COLOR = fcolor
390 FOR i = 1 TO 10
400 x0 = 39*RND(1): y0 = 39*RND(1): IF SCRN(x0, y0) <> 0 THEN 400
410 PLOT x0, y0: NEXT
420 & wait for trigger press to start game
430 VTAB 1: HTAB 16: PRINT "Press trigger"
440 IF PDL(7) <> 1 AND PDL(9) <> 1 THEN 440
450 VTAB 1: PRINT " SCORE "; score
460 & ***main loop***
470 IF food = 10 THEN 500
480 x0 = 39*RND(1): y0 = 39*RND(1): IF SCRN(x0, y0) <> 0 THEN 480
490 COLOR = fcolor: PLOT x0, y0: food = food+1
500 IF score/500 = INT(score/500) THEN 520
510 delay = delay-2: & speed snake up
520 FOR d1 = 1 TO delay: NEXT: p = PDL(5)
530 IF (dir = 4*p OR p = 4*dir) OR NOT (p = 1 OR p = 2 OR p = 4 OR p = 8) THE
N 550
540 dir = p: & change snake's direction
550 IF dir <> 1 THEN 570
560 y = y-1: GOTO 620
570 IF dir <> 2 THEN 590
580 x = x+1: GOTO 620
590 IF dir <> 4 THEN 610
600 y = y+1: GOTO 620
610 x = x-1
620 block = SCRN(x, y)
630 IF block <> bcolor AND block <> rcolor THEN 680
640 snakes = snakes-1: RESTORE: GOSUB 800: IF snakes > 0 THEN 245
650 hiscore = score: VTAB 1: HTAB 16: INPUT "Play again? "; an$
655 an$ = LEFT$(an$, 1): IF an$ = "Y" OR an$ = "y" THEN 240
660 TEXT: END
680 IF block <> fcolor THEN 730
690 COLOR = rcolor: PLOT x, y: s(head, 0) = x: s(head, 1) = y: head = head+1
700 IF head < size THEN 720
710 head = 0
720 score = score+20: VTAB 1: HTAB 9: PRINT score: food = food-1
725 PRINT CHR$(7); : GOTO 470
730 COLOR = rcolor: PLOT x, y: s(head, 0) = x: s(head, 1) = y: head = head+1
740 IF head < size THEN 760
750 head = 0
760 COLOR = 0: PLOT s(tail, 0), s(tail, 1): tail = tail+1
770 IF tail < size THEN 790
780 tail = 0
790 GOTO 470
795 & make crashing noise
800 POKE 28006, 255: CALL 28000
810 FOR i = 240 TO 255: POKE 28006, i: CALL 28000
820 FOR d = 1 TO 150: NEXT d: NEXT i: RETURN

```

```

50 REM Sprites Editor
100 LOMEM ;28100: TEXT: vw% = 28004: bcount% = 28016
103 buf% = 28046: vaddr% = 28013: DIM m(20, 10)
105 POKE 16953, 32
110 FOR i = 0 TO 21: READ dat%: POKE 28000+i, dat%: NEXT
120 DATA 62,29,56,2,62,26,50,115,109,33,142,109,17,0,0,1,1,0,205,0,253,201
125 & VREAD & VWRITE
130 FOR i = 0 TO 7: READ dat%: POKE buf%+i, dat%: NEXT
135 DATA 120,72,72,72,120,0,0,0: & cursor shape
140 POKE bcount%, 8: av% = 14336: GOSUB 570
150 FOR i = 0 TO 4: READ dat%: POKE buf%+i, dat%: NEXT
155 DATA 0,8,0,9,208: & sprite attributes
160 POKE bcount%, 5: av% = 7936: GOSUB 570
180 GOSUB 510
190 GOSUB 460: HTAB 1: VTAB 23: PRINT "0=END 1=PATT'N 2=SCREEN 3=COLOR";
195 FOR delay = 1 TO 200: NEXT
200 op = PDL(13): IF op = 0 THEN TEXT: POKE 16953, 95: END
210 IF op = 1 OR op = 2 THEN 320
220 IF op <> 3 THEN 200
230 GOSUB 460
240 VTAB 21: PRINT: VTAB 21: HTAB 1: INPUT "color set (5 to 32): "; cs
245 IF cs < 5 OR cs > 32 THEN 240
250 PRINT "has char. range of "; 8*(cs-1); " to "; 8*(cs-1)+7
300 INPUT "color byte (0 to 255): "; cb: IF cb < 0 OR cb > 255 THEN 300
310 POKE buf%, cb: POKE bcount%, 1: av% = 8191+cs: GOSUB 570: GOTO 190
320 GOSUB 520: GOSUB 460: VTAB 23: HTAB 1: IF op <> 1 THEN 350
330 PRINT "1=GET OUT 2=CLEAR GRID ";
340 low = 1: hi = 8: GOTO 370
350 PRINT "1=MAIN MENU 2=SELECT CHAR. CODE";
360 low = 10: hi = 19
370 r = 1: c = low
380 POKE buf%, 8*(r-1): POKE buf%+1, 8*c: av% = 7936
385 POKE bcount%, 2: GOSUB 570
390 p = PDL(5): IF PDL(13) = 1 THEN ON op GOTO 480, 190
400 IF PDL(13) = 2 THEN ON op GOSUB 510, 520
410 IF PDL(7) = 1 THEN VTAB r: HTAB c: ON op GOSUB 540, 560
420 IF PDL(9) = 1 THEN VTAB r: HTAB c: ON op GOSUB 550, 560
430 r = r-(p = 1)+(p = 4): r = r+((r = 0)-(r = hi+1))*hi
440 c = c-(p = 8)+(p = 2): c = c+((c = low-1)-(c = hi+1))*(hi-low+1)
450 FOR d = 1 TO 30: NEXT d: GOTO 380
460 VTAB 21: HTAB 1: FOR j = 1 TO 3: PRINT: NEXT j: RETURN
465 & clear lines 21 to 23
480 VTAB 21: HTAB 1: FOR r = 1 TO 8: byte% = 0: FOR c = 1 TO 8
490 byte% = byte%+m(r, c)*2^(8-c): NEXT c: POKE buf%+(r-1), byte%
495 PRINT byte%; " "; : NEXT r
500 POKE bcount%, 8: av% = 8*ch: GOSUB 570
505 PRINT "press any key": GET k$: GOTO 190
510 FOR r = 1 TO 8: VTAB r: HTAB 1: FOR c = 1 TO 8: m(r, c) = 0
515 PRINT CHR$(46); : NEXT c: NEXT r: r = 1: c = 1: RETURN
520 VTAB 22: HTAB 1: PRINT: VTAB 22: INPUT "char. code (32 to 255): "; ch
525 IF ch < 32 OR ch > 255 THEN 520
530 RETURN
540 PRINT CHR$(42); : m(r, c) = 1: RETURN
550 PRINT CHR$(46); : m(r, c) = 0: RETURN
560 PRINT CHR$(ch); : RETURN
570 v1% = av%/256: v2% = av%-256*v1%: POKE vaddr%, v2%: POKE vaddr%+1, v1%
575 CALL vw%: RETURN: & poke video address

```

```

50 & Alan Aben
75 &
100 & High Resolution FLAG
105 & with no blending of colors
110 HGR2: HCOLOR = 11
120 FOR y = 0 TO 191
130 HPLOT 0, y TO 80, y: NEXT y
140 REM white stripes
150 HCOLOR = 3
160 FOR y = 0 TO 30
170 HPLOT 80, y TO 176, y: NEXT y
180 FOR y = 161 TO 191
190 HPLOT 80, y TO 176, y: NEXT y
200 REM blue stripe
210 HCOLOR = 9
220 FOR y = 0 TO 191
230 HPLOT 176, y TO 255, y: NEXT y
240 REM widen black field
250 HCOLOR = 0
260 FOR y = 31 TO 160
270 HPLOT 73, y: HPLLOT 182, y: NEXT y
280 REM circle of stars
290 r = 52: f = 3.1416/180
300 sv = 360/50: HCOLOR = 3
310 FOR d = 0 TO 360 STEP sv
320 a = d*f
330 x = INT(COS(a)*r*.87+.5)
340 y = INT(SIN(a)*r+.5)
350 HPLOT 128+x, 96-y: NEXT d
360 REM freeze display
370 GET x$
380 IF x$ = "" THEN 370
390 TEXT: END

```

```

50 & Alan Aben
75 &
100 & Polygon
105 &
110 TEXT: PRINT " Polygon:": PRINT
120 INPUT " How many sides? "; s
130 PRINT " I'm thinking..."
140 REM array of points
150 DIM c(s, 2)
160 d = 90: r = 75
170 sv = 360/s
180 FOR p = 1 TO s
190 a = d*(3.1416/180)
200 c(p, 1) = 128+INT(COS(a)*r*.87+.5)
210 c(p, 2) = 80-INT(SIN(a)*r+.5)
220 d = d+sv
230 NEXT p
240 c(0, 1) = c(s, 1): c(0, 2) = c(s, 2)
250 REM set up screen
260 HGR: HCOLOR = 9
270 PRINT TAB(8); s;
280 PRINT " Sided Figure"
300 REM draw polygon
310 FOR p = 1 TO s
320 sx = c(p, 1): sy = c(p, 2)
330 FOR v = p TO s
340 dx = c(v, 1): dy = c(v, 2)
350 HPLLOT sx, sy TO dx, dy
360 NEXT v
370 NEXT p
380 FOR t = 1 TO 2000: NEXT t
390 CLEAR
400 GOTO 110

```

# A P E SOFTWARE PRESENTS...

## ELECTRONIC GAME PACK

NOW PLAY EXCITING COMPUTER VERSIONS OF THE FOLLOWING POPULAR GAMES. PLUS ONE NEW ONE ESPECIALLY CREATED FOR THIS PACKAGE. GAMES INCLUDED ARE BATTLESHIP, MASTERMIND, BACKGAMMON, THREE DIMENSIONAL TIC-TAC-TOE, AND ROBOT MINER. THESE ARE ONE PLAYER GAMES THAT LET YOU MATCH WITS WITH ADAM. THE GAMES MAKE EXTENSIVE USE OF ADAM'S SPRITE, COLOR GRAPHIC, & SOUND CAPABILITIES. ALL ARE FULLY DOCUMENTED. WE EVEN INCLUDE A BACKUP UTILITY. TO ORDER YOUR COPY, SEND \$29.95 PLUS \$2.00 FOR POSTAGE (QUE. RESID. ADD 9% SALES TAX)

A P E SOFTWARE  
4756 LALONDE  
PIERREFONDS, QUEBEC  
H8Y 1V2

# EVEN UP THE SOFTWARE GAP!

AEC PROVIDES THE SOLUTION TO YOUR SOFTWARE DILEMMA. ENJOY THE SAME TYPE OF SOFTWARE SERVICE THAT USERS OF OTHER BRANDS HAVE HAD FOR YEARS. WE HAVE MOST ADAM PROGRAMS AND CARTRIDGES AVAILABLE ON TAPE OR DISK FOR YOU TO TRY AT HOME FOR JUST \$15.00 EACH! ONCE YOU FINISH EVALUATING IT, YOU MAY DO WHAT YOU WISH WITH THE TAPE OR DISK. IT COSTS \$10.00 PER YEAR TO JOIN. AND NOW AEC ALSO HAS HARDWARE. FACTORY SURPLUS. 2ND TAPE DRIVE ONLY \$55.00 EA. SEPARATE POWER SUPPLY W/CASE ONLY \$70.00 EA. POSTAGE IS INCLUDED. QUEBEC RESIDENTS ADD 9% SALES TAX. FOR OUR CATALOG AND FURTHER INFO WRITE TO:

ADAM EVALUATION CLUB  
3811 PRUD'HOMME #14  
MONTREAL, QUEBEC  
H4A 3H8

### PERIPHERALS and SUPPLIES for the ADAM

- SUPER CHRISTMAS SPECIAL -

These special prices are valid until end of Dec 85

```

10 & MORE POKES
11 &
12 & J.M. Roy (1985)
13 &
20 TEXT
25 POKE 17529, 0
30 GOSUB 240
35 POKE 17115, 240: TEXT
40 POKE 17059, 5: TEXT
50 x = INT(RND(1)*22)
60 y = INT(RND(1)*19)
70 IF x < 1 THEN x = 1
80 IF y < 1 THEN y = 1
90 POKE 17202, x: TEXT
100 POKE 17201, y: TEXT
110 PRINT CHR$(43)
120 PRINT CHR$(7)
130 HTAB 23: VTAB 20
135 PRINT CHR$(43); " is at"
140 HTAB 25: VTAB 22: PRINT "Col "; x
150 HTAB 25: VTAB 23: PRINT "Row "; y
160 FOR t = 1 TO 1000: NEXT t
170 GOTO 50
180 END
190 POKE 17529, 66
200 POKE 17202, 1: TEXT
210 POKE 17201, 0: TEXT
220 POKE 17059, 0: TEXT
230 END
240 POKE 17059, 3: TEXT
250 POKE 17115, 22: TEXT
260 INVERSE
270 VTAB 5: HTAB 10: PRINT "MORE POKES"
280 NORMAL
290 VTAB 6: HTAB 10: PRINT " "
300 VTAB 15: HTAB 5
310 PRINT "Use Control-C to exit."
320 VTAB 17: HTAB 14: PRINT "and"
330 VTAB 19: PRINT "Enter ";
340 PRINT CHR$(34); : PRINT "Run 190";
350 PRINT CHR$(34);
360 PRINT " to clear screen"
370 VTAB 22: HTAB 5
380 PRINT "Hit any key to start."
390 GET a$
400 RETURN
    
```

5 1/4 INCH DISK DRIVE .....	\$349.99
ADAMLINK DIRECT CONNECT MODEM (300 BAUD)	\$ 99.99
ADAM COMPATIBLE 64K MEMORY EXPANDER .....	\$ 75.99
EXTRA DIGITAL DATA DRIVE .....	\$109.99
TRACTOR FEED .....	\$130.00
ADDRESS BOOK FILER & AUTODIALER .....	\$ 56.99
DAISY WHEELS (Gothic 15, Script 12, Manifold 10)	
Letter Gothic 12, Light Italic 12) each ..	\$ 10.00
PRINTER RIBBON.....	\$ 10.00
ADAM COMPATIBLE DATA PACKS .....	\$ 6.00
(5 for \$28.00 or 10 for \$53.00)	
ADAM MONITOR CABLE .....	\$ 14.99
NO NAME DISKETTES .....	10 for \$ 17.00
FUJI DISKETTES .....	10 for \$ 26.00

### \*\*\*\*\*COLECO SOFTWARE for the ADAM\*\*\*\*\*

CP/M 2.2 & ASSEMBLER (DP,DISK) .....	\$ 79.99
ADAMCALC (DP) SPREADSHEET PROGRAM .....	\$ 49.99
SMARTLOGO (DP) EDUCATIONAL PROGRAM .....	\$ 59.99
SMARTFILER (DP,DISK) ALL PURPOSE FILER ...	\$ 45.00
EXPERTYPE (DP,DISK) ELECTRONIC TYPING AID	\$ 55.00
SMART LETTERS AND FORMS (DP,DISK) .....	\$ 45.99
ELECTRONIC FLASHCARD MAKER (DP) EDUCAT'AL	\$ 45.00
FLASH FACTS-TRIVIA (DP) EDUCATIONAL .....	\$ 25.00
FLASH FACTS-VOCABULATOR (DP) EDUCATIONAL	\$ 25.00
ELECTRONIC WORDBOOK (DP) EDUCATIONAL .....	\$ 48.00
RECIPE FILER (DP,DISK) .....	\$ 45.00
TAX PLANNER .....	\$ 59.99
MARKET MONITOR .....	\$ 95.99
DONKEY KONG (DP) .....	\$ 45.00
DONKEY KONG JR (DP) .....	\$ 45.00
ZAXXON (DP) .....	\$ 45.00
DRAGON'S LAIR (DP,DISK) .....	\$ 50.00
FAMILY FEUD (AVAILABLE SOON) .....	\$ 45.00
JEOPORDY (AVAILABLE SOON) .....	\$ 45.00
***** PERSONAL ACCOUNTANT *****	
3 FINANCIAL PROGRAMS IN ONE.	
SIMULTANEOUSLY POSTS ENTRIES, COMPILES FINANCIAL	
REPORTS, LARGE DATA BASE, BREAKS DOWN	
EXPENSES, AMORTIZATION/PAYMENT SCHEDULES ..	\$ 55.00

Please add 3% for shipping and handling charges.

Send orders to: HI-TEK MARKETING  
244 PENBROOKE CLOSE, S.E.  
CALGARY, ALBERTA, T2A 3P1  
(403) 272-3927