HCJournal™

Home Computing Journal

Intelligent Software for Personal Computers in a Home Environment

VOLUME 1

Decision Mapping

VOLUME 1

CONTENTS

Volume 1 — HCJ At A Glance

We'd like to welcome our many thousands of new readers who are joining us as a result of *Home Computer Magazine* ceasing publication. Undoubtedly, some of you are disappointed that your favorite magazine is no longer available.

But lest you stay disappointed for long, please examine this Volume of *Home Computing Journal*. We are pleased that we have been able to offer you many of the best features of *HCM* in the *Journal*. And because *HCJ* is a quarterly, subscriber-based package consisting of a state-of-the-art laser-printed journal with companion diskette, we're able to go far beyond what a newsstand magazine can offer—even one as excellent as *HCM* was...

Artificial intelligence, database management, high-powered programming aids, realistic simulations, and specialized software for personal investing, task-specific report writing, computer-assisted design, desktop publishing, personal communications, plus entertaining math and logic excursions are just *some* of the projects already on our planning board.

With all of this exciting software, learning opportunity, and new worlds of discovery on the horizon, we hope that you'll plan to stay with us for a long time to come. And, if you like the way we've delivered this first quarterly package of bits 'n' bytes, you can do us a big favor: Because we're not a newsstand magazine, we don't have any cost-effective way of reaching the many thousands of computer users who regularly purchase computer magazines each month from their favorite magazine vendors. So, you can do us—and them—a tremendous service by putting out the word that what *HCM* did best—and a whole lot more to boot—is alive and well...and that *HC Journal* invites one and all to come join us.

Publisher David G. Brader

Editor Roger Wood

Production Chief Norman Winney, Jr.

Program Editor William K. Balthrop

Assistant Program Editor Randy Thompson

Editorial Consultant Gary M. Kaplan

Main Switchboard: Tel. (503) 342-4013
Mail Address: P.O. Box 70248 • Eugene, OR 97401

Home Computing Journal (HCJ) is published quarterly. Subscription, editorial, and inquiry address is P.O. Box 70248, Eugene, OR 97401. For current single-copy and subscription pricing, please see rear cover. All mail directed to the Editor or Letters-to-the-Editor (HCJ I/O) column will be treated as unconditionally assigned for publication and copyright purposes and is subject to edit and comment. HCJ assumes no liability for errors in articles or programs.

PRODUCTION NOTE: This printed journal was produced in camera-ready format with a desktop publishing system employing the Apple Macintosh and LaserWriter, and the following software: MacPaint, MacDraw, and MacWrite from Apple Computer, Inc; ReadySetGo from Manhattan Graphics Corporation; plus Desktop Magic and LaserMagic from World Class Software.

HCJournal™ I/O

Welcome to *Home Computing Journal*'s *I/O* section. Here, you get a chance to provide feedback on our articles and programs, and you pick up tips to solve those software and hardware problems that invariably pop up in the course of home computing. We encourage readers who have questions, comments, or interesting computing ideas to send letters to:

Home Computing Journal
Attn: Letters to the Editor
P.O. Box 70248
Eugene, OR 97401

Monitor Disk Drive Errors

Dear Sir:

I had an experience this weekend that I thought might be of interest to other users of the Commodore 1541 disk drive.

My son's *Winnie the Pooh in the Hundred Acre Woods* program was having difficulty loading some of the screen files, and would sometimes provide an error message indicating a problem with the disk drive.

A little investigation showed that if the disk drive was sitting to the left of my TV (or my Commodore 1701 monitor), there was trouble accessing the drive. When the drive was sitting to the right, there was little or no trouble.

In conclusion, setting the disk drive to the left of a TV or monitor can produce symptoms which mimic alignment problems. Readers might want to check for this type of interference before going to the trouble of having a drive realigned.

Jack Ryan
El Dorado, AR

The problem that you are encountering, Jack, is not directly related to the monitor's left side, but to the monitor itself. You see, computer monitors, television sets, and even audio speakers create a magnetic field when in use. Because the data on a disk is stored magnetically, this field can interfere with your disk drive's operation. It is always advisable to keep your disk drive and your disks away from your television or monitor. This applies to all computers. The fact that your 1541 disk drive prefers the right side merely indicates that your monitor has a stronger magnetic field on the left side—you may yet experience difficulties if you leave the drive close to the monitor.

TI Monitor Cable Info

Dear Sir:

I recently purchased a TV/monitor that has extra video/audio plugs for computer hook-up. The problem is that the TI-99/4A can only be hooked up to the antenna terminals of the set. Is there any way to rewire the 5-pin modulator cable to the video/audio RCA plugs needed on the set? I'm sure there are a lot more people like myself with this problem and I feel that this would be another plus point for the 99/4A.

Tom Hamerly
Fremont, CA

What you are looking for, Tom, is a video monitor cable. Do not try to "rewire" your modulator cable to match; there are several readily available monitor cables that will do the trick. You need a cable with a 5-pin DIN plug on one end, and two RCA plugs on the other (one for audio and one for video). Look at mail order advertisements or your local electronics or computer store. They should have just the thing for less than ten dollars.

PCjr Sound Off

Dear Sir:

I have noticed that many programs which I run from *HCM* diskettes have caused the sound on my PCjr to shut off. The only command which allows the sound to restore without rebooting is SOUND OFF. Additionally, some programs cause the function keys to turn off, such as F2/run, F3/load, etc.

Is there any way to correct this problem and is it common among others with Juniors?

Gregory O. Davis
Livonia, MI

The "problem" you describe with the sound shutting off is because you don't have an external speaker attached to your PCjr, Gregory. If you use an RF modulator connected to a television, the sound will come out of the TV's speaker. If you are using a video monitor, there is another way to hear the sound from these programs. Just hook up an amplifier and speaker to the audio output of the PCjr—it is an RCA jack on the back left of the unit labeled with an "A." You can use the AUXiliary input of your stereo, or some other suitable amplification system. The reason that some of the function keys turn off is that they are reassigned to other functions by the program. After you have used a program, there is no need to totally re-boot. Just type SYSTEM to return to DOS, then type BASIC, and you are returned to a freshly initialized BASIC system with all the usual function key assignments in place.

Hi-Res Screen Dump

Dear Sir:

I have an Apple *IIc*, and wonder if *HCM*'s Screen Dump for Apple computers (V5N5) will work for the hi-res screen. The three programs I added it to all use either HGR or HGR2 for the screen display (two pie charts and a bar graph program). All I get printed out are the text commands located below the HGR image or the text commands that were on the TEXT window before the machine went to HGR2. Is there any way I can get the HGR/HGR2 screens "dumped" to the printer? I hope so, as it would make these programs much more useful to me than they now are.

I have been using the *Run-Day-View* program (V5N4) with my work schedule. Is there some way I could get it to not print the spaces between the hours, so as to get at least one more hour of schedule on the sheet?

David E. Paul
Perryville, KY

David, this screen dump program prints only Apple's text screen. It cannot be easily modified to print the high-resolution (hi-res) screen. Hi-res screen dumps are fairly printer-specific. Some printer interfaces, notably the Grappler from Orange Micro, include several short programs that allow for screen dumps of hi-res graphics for a number of different printers.

To change Run-Day-View to add even an extra half-hour would be a major modification. You could add space on the printout to write several more time entries by deleting line 2750 and adding this line:

```
2765 FOR  IT =1 TO 9 : PRINT
  SPC(32); "!" : NEXT
```

Configuration Confusion

Dear Sir:

In December, 1984, I purchased an IBM PCjr, enhanced model. Since that time I have expanded it to include the following hardware: Racore expansion unit with second disk drive, an extended memory to 512K RAM, two IBM joysticks, Keytronics 5149 numeric keypad, Epson RX80/FT+ printer, and IBM PCjr internal modem.

With the Racore unit (similar to the Quadram you tested), because of software patches it is impossible to run IBM PCjr BASIC (cartridge) with the extended memory, and the TERM (terminal emulation program) with the IBM PCjr internal modem. Also, as your review of the Quadram unit pointed out, it is a real pain to discover what combinations of software work with this system. The problem lies in the fact that IBM PCjr BASIC (cartridge) won't work with the extended memory—to

Special "Hackers Challenge" For Atari 800 Users

The screen access used in the Atari version of *WordWeave* bypasses normal screen I/O by making the screen directly access the contents of the variable A$. In addition, the disk routine accesses the disk directly, bypassing DOS, thus speeding up disk I/O. These techniques work great on the Atari 800XL and Atari 130XE. The scheme, however, does not function properly on older Atari 800s. The first Atari programmer who sends us the modifications to our program, allowing it to work on the model 800, will receive a free Volume of *Home Computing Journal*. We'll then share the fix with other 800 users.

overcome this limitation, you must have the original PC-DOS to run IBM PCjr BASIC (cartridge) and the IBM PCjr internal modem, limiting you to 128K RAM. You must use a modified DOS disk to run programs that will use the extended memory! This requires several different modified DOS disks to run programs—confusing the user.

John Battista
Clearwater, FL

That's right, John, the PCjr applications you mention cannot access the Racore memory. In fact neither Cartridge BASIC on the PCjr, nor BASICA on the IBM PC can access more than 64K of RAM. The Microsoft Booster for the PCjr (see review in HCM V5N5) supports a speeded-up version of BASIC that moves Cartridge BASIC into the faster RAM in the Booster—but it still can only acess 64K. You are also quite correct in saying that you need an original DOS 2.1 disk. This would be a disk that does not have a CONFIG.SYS file that reconfigures how the PCjr accesses its memory. One other note: World Class Software has recently developed a menu-driven "Configurator" package. This program is designed to lead even an inexperienced user through the steps of configuring the PCjr Racore memory expansion, and to easily "tailor make" configuration disks for different applications. The package is available for $89.95 from World Class Software, P.O. Box 70288, Eugene, OR 97401.

TI Card-Trix Tip

Dear Sir:

Card-Trix from HCM V5N5 is dynamite stuff. I'm already using it heavily. We made the following changes which you may wish to incorporate or pass along:

Since 3x5 cards can be purchased in computer setups (continuous and tractor-holed), the vertical ! defining the borders of the cards aren't needed.

We threw a full print statement into 920.

```
:: PRINT#1: "!";TAB(51);"!"
```

and scaled 930 down to `930 GOSUB 950` and left 950 with `950 RETURN`

(I'm sure we could have eliminated 930 and 950, but we didn't have time to play with it.)

We resaved the modified program under a new name, so that if we use the computer 3x5's we can use the modified program, but if we use the regular fan-fold 8-1/2 x 11 paper we have your original program which draws top and bottom lines for cutting.

Alan Weber
Marlboro, NJ

We're glad to hear that you are putting Card-Trix to good use, Alan. It's true that the modifications you provide effectively remove the exclamation points defining a card's vertical borders. (And yes, you could have deleted lines 930 and 950 for the same results.) Of course, to make this alteration useful, one would ideally have an adequate supply of the fanfold 3 X 5 cards that you mention. These cards evidently do exist, but are generally hard to find (and expensive) in many areas.

C-64 LPRINT Wanted

Dear Sir:

My questions concern *HCM*'s screen dumping program for a C-64 (V5N4). As you can see from the enclosed printout, whatever is dumped from the screen is printed on only part of the paper. Is there a way to get a printout which uses the entire sheet? Also, is there a way to place a print command directly into a program (similar to the LPRINT command used with IBM PC's)?

Jonathan C. Jones
Lexington, KY

Jonathan, to answer your first question, it would be difficult to convert the screen dump program to take up more room on a sheet of paper. The main problem is that the C-64's screen is simply not that large! A sheet of standard printer paper is 80 characters across, while the Commodore 64's screen is only 40. As for your second question, there is an easy solution. Use the following code in place of LPRINT:

```
OPEN 4,4:CMD 4:
PRINT "Anything you want to
print"
```

The OPEN command sets up the printer as an output device. The CMD 4 statement tells the computer to send all output to device number 4 (the printer). Now, all PRINT statements will go to the printer—just like an LPRINT. To re-direct output to the screen, use this code:

```
PRINT#4:CLOSE4
```

This turns off the printer and cancels the CMD statement.

MX-80 Parallel Operation

Dear Sir:

My computer set-up consists of a TI-99/4A, peripheral expansion box with disk drive, memory expansion and RS232 interface, a TI (MX-80 modified) printer, and a second external disk drive. The system works extremely well and I am very happy with its performance. However, recently I purchased a parallel cable to enable me to use the printer from the parallel port on the RS232 card.

After hooking up the parallel cable, my troubles started. The system works fine when using either of the two serial ports, but during use of the parallel port the printer would go out of control and print gibberish or uncontrolled form indexes—many times with one line per page. Turning the printer and/or computer off would not reset the system. By turning the system off and leaving it for several hours, it might start working all right, but would soon go into the uncontrolled mode again.

I called a service representative for Epson printers and after describing the problem, was told that I would have to disconnect the serial board inside the printer in order to use the parallel port. I feel that there must be an easier way to solve the problem. Have you ever run across this problem before and is there an easier way to solve it? I would appreciate any help that you can provide me.

One other question: I am having a hard time trying to find an easy way to print columns of figures using the printer and line up the decimal points. Does someone have a suggestion?

Robert Schertz
Bettendorf, IA

Robert, we are surprised that you can get the printer to work at all when connected the way you describe. As long as the serial card is in the printer, we have found that the parallel port does not function properly on the MX-80. The answer to your problem is just as Epson suggests—remove the serial card while using your printer as a parallel device. This is not very hard. First, unplug the unit and remove the paper and the knob that turns the platen (it should pull off easily). Then take out the screws on the bottom and remove the case—be careful not to damage the wiring that connects the LED indicators and selector switches on the top of the unit. Finally, remove the four screws that hold the serial card in place, and gently unplug it by pulling it straight up. When reassembled (without the serial card) your parallel printer should work fine.

As for printing formatted numbers, the easiest way is to use the PRINT USING and IMAGE commands available in TI Extended BASIC (see your Extended BASIC reference manual for details). If, however, you don't have access to Extended BASIC, here is a seven line BASIC subroutine that prints decimal numbers into columns with their decimal places aligned vertically. This routine works with any number of 12 digits or less. To use it, you must first open your printer, or whatever output device you choose, as file #1. Then set the variable N to the number to be printed and GOSUB 1000.

```
1000 NUM$=STR$(N)
1010 DP = POS(NUM$,".",1)
1020 IF DP>0 THEN 1050
1030 DP=LEN(NUM$)+1
1040 NUM$=NUM$&"."
1050 PRINT#1:TAB(13-DP);NUM$
1060 RETURN
```

Add A Third Disk Drive?

Dear Sir:

I am delighted with *HCM*s article, "Adding a Second Drive to the PCjr" (V4V4). The second drive I added per your article works like a champ. I understand, though, that the hardware modification already has the extra logic on board to handle four drives, and I'm considering adding a third drive to my configuration. Can you tell me what changes are needed to the software modification in the

article? The software modification as written only recognizes drives A and B.

Alfred I. Fier
Gaithersburg, MD

There are additional hardware modifications (specifically the addition of more logic chips) necessary for the PCjr to support 3 or 4 drives. If there's enough interest, Alfred, we'll develop a Tech Note for this enhancement. The software fix allowing the PCjr to access the properly interfaced hardware is an easier matter: The main purpose of the software is to alter the equipment status byte which is set by the IBM PCjr's Basic Input/Output System (BIOS) upon power up. The two high-order bits of location 410 (HEX) tell DOS how many disk drives are active. To change the number of drives you simply alter the MODBOOT.BAT file (explained in HCM V4N3): Change the OR BY [410],40 to OR BY [410],80 for 3 drives and to OR BY [410],C0 for 4 drives.

Underlining With Epson

Dear Sir:

I use AppleWorks with an Apple IIc. I am extremely satisfied with it, and I use it for all of my word processing. However, I have an annoying problem with AppleWorks. I am unable to use the underline function.

My printer is a TI-99/4A Impact Printer (Epson MX-80). When I use the printer with the TI computer, I am able to underline perfectly. The procedure is to use an "&" before any material that is to be underlined. The printer manual for the Epson does not give any codes for underlining, so I am not able to use Option 2 on the "underlining" menu.

You are my last hope. I have sent Apple a letter, but they have not answered. Any help that you can give me would greatly appreciated.

Robert J. Field
Skokie, IL

We've experimented with the same configuration you describe, Robert, and have had no success with underlining—although the ImageWriter works just fine. Using the ordinary underline codes available from the Open-Apple-O option does add a line feed before underlining takes place. This adds a line feed between the text and the underline on the MX-80, but works just great on an ImageWriter. We've contacted Apple Technical support, but they were not able to solve the problem. Because newer Epson printers (such as the FX-85) support control codes for printing underlines, you can use option 2 of the underline menu with them.

Productivity Problem Solved

Dear Sir:

I believe that I solved HCM's Snap-Calc challenge (V5N6). My spreadsheet (enclosed) is divided into 4 sections of consecutive columns: Colums 1-4 pertain to college A; Columns 5-8 pertain to College B; and the final 2 sections combined (Columns 9-13) represent two years at a community college, followed by three years at College A (taking figures for years 2, 3, and 4). If you analyze the logic of the spreadsheet from the standpoint of the formulas used in Rows 5, 6, 11, 12, 13, 17, and 18 (see the enclosed logic printout), Row 5 displays the total expendable income of the family by adding the first four rows of Column 1. I used the LAG function of *Snap-Calc* to allow for a 10 percent rise in overall yearly expendable income. This function takes the value of the previous column in the designated row. In this case, LAG 5 accesses the values of Row 5 to use in Row 6. (Because there is no column previous to Column 1, the first value in Row 6, Column 1 is meaningless.) The next value (Column 2) in Row 6 is the result of accessing the number in Row 5, Column 1, multiplying that number by 10 percent, and adding the result to the original number accessed. This sequence carries through the entire sixth row.

Row 11 is the sum of Rows 7 through 10 in Column 1, designating total fundamental expenses. Row 12 allows for an overall yearly increase in expenses of 5 percent, using logic similar to that used in Row 6. Row 13 then finds the difference between expenses and expendable income. If the difference is a negative number, the expense of college is driving the family budget into the red, requiring other sources of income to balance the deficit. Rows 14 through 16 are examples of some of the more common sources of financial aid. Row 17 displays the sum total financial aid, while Row 18 simply adds this value to any deficits in Row 13. If the values of Row 18 are still negative, the student may have to become a member of the working class for a while.

Note that the values in Rows 6 and 12 in Columns 1, 5, 9, and 11 are meaningless values generated by the LAG functions of these rows.

William Kincaid
San Francisco, CA

Well done, William! We couldn't have done it better ourselves. [Reader Note: William's spreadsheet and logic file are reproduced below.]

Atari Sound-On-Sound Echo

Dear Sir:

I am an Atari 800 owner and I find that HCM's clever *Sound-On-Sound* program (V5N5) will not run on my Atari 800.

The published version uses an ATASCII conversion table that is not in the same location in ROM on the 800, so I have included my own fix for this program that POKEs the ATASCII conversion table into RAM. In the keyboard code to ATASCII code conversion table that is POKEd into page six

CONTINUED ON PAGE 32

SNAP-CALC SPREAD SHEET COLLEGE ANSWER

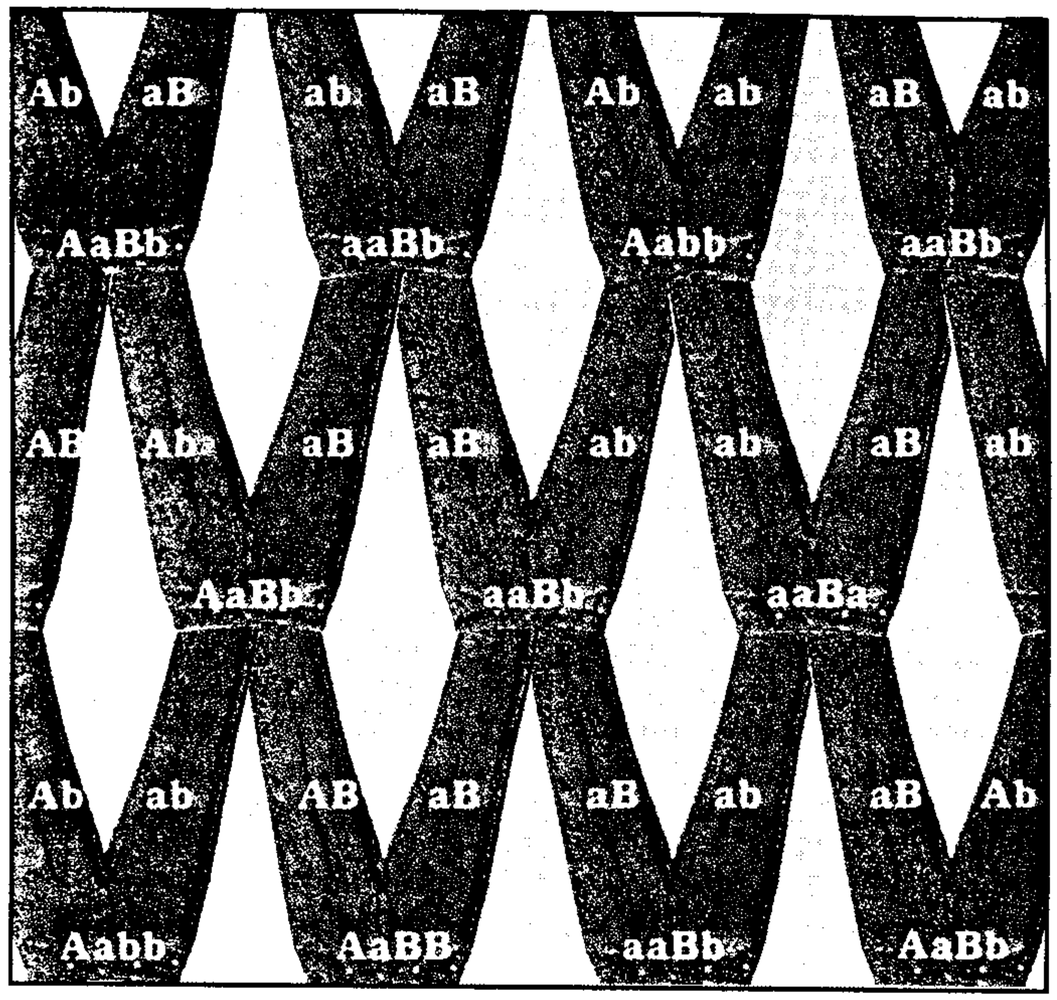
Row	Row Name	1	2	3	4	5	6	7	8	9	10	11	12	13
1	EX / INC / FA	1500.00	0.00	0.00	0.00	1500.00	0.00	0.00	0.00	1500.00	0.00	0.00	0.00	0.00
2	EX / INC / MO	1500.00	0.00	0.00	0.00	1500.00	0.00	0.00	0.00	1500.00	0.00	0.00	0.00	0.00
3	EX / INC / ST	500.00	0.00	0.00	0.00	500.00	0.00	0.00	0.00	500.00	0.00	0.00	0.00	0.00
4	OTHER	500.00	4400.00	4840.00	5324.00	500.00	4400.00	4840.00	5324.00	500.00	4400.00	4840.00	5324.00	5856.40
5	TOT / EX / INC	4000.00	4400.00	4840.00	5324.00	4000.00	4400.00	4840.00	5324.00	4000.00	4400.00	4840.00	5324.00	5856.40
6	NEXT YEAR	0.10	4400.00	4840.00	5324.00	5856.40	4400.00	4840.00	5324.00	5856.40	4400.00	4840.00	5324.00	5856.40
7	TUITION	1432.50	0.00	0.00	0.00	4057.40	0.00	0.00	0.00	594.00	0.00	0.00	0.00	0.00
8	FOOD / HOUS	1098.00	0.00	0.00	0.00	2340.00	0.00	0.00	0.00	400.00	0.00	0.00	0.00	0.00
9	BOOK / SUPL	300.00	0.00	0.00	0.00	300.00	0.00	0.00	0.00	300.00	0.00	0.00	0.00	0.00
10	MISC / PERS	990.00	4011.53	4212.11	4422.72	990.00	8071.77	8475.36	8899.13	800.00	2198.70	4212.11	4422.72	4643.86
11	TOTAL / EXP	3820.50	4011.53	4212.11	4422.72	7687.40	8071.77	8475.36	8899.13	2094.00	2198.70	4212.11	4422.72	4643.86
12	NEXT YEAR	0.05	4011.53	4212.11	4422.72	4643.86	8071.77	8475.36	8899.13	9344.09	2198.70	2308.64	4422.72	4643.86
13	VARIANCE 1	179.50	388.47	627.89	901.28	-3687.40	-3671.77	-3635.36	-3575.13	1906.00	2201.30	627.89	901.28	1212.54
14	SCHSHIPS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	GRANTS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	LOANS / STU	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	TOTAL / AID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	VARIANCE 2	179.50	388.47	627.89	901.28	-3687.40	-3671.77	-3635.36	-3575.13	1906.00	2201.30	627.89	901.28	1212.54

College A (4 Years) | College B (4 Years) | Community College (2 Years) | College A (Last Three Years To Supplement Community College.)

LOGIC FILE

```
LOGIC NAME IS CALCLOG1
TOTAL COLUMN IS 0
LAST COLUMN IS 13
1 IS EX/INC/FA
2 IS WX/INC/FA
3 IS EX/INC/ST
4 IS OTHER
5 IS TOT/EX/INC
5 = 1 + 2 + 3 + 4
6 IS NEXTYEAR
6 = (.1) * LAG 5 + LAG 5
7 IS TUITION
8 IS FOODHOUS
9 IS BOOK/SUPL
10 IS MISC/PERS
11 IS TOTAL/AID
12 IS NEXTYEAR
12 = (.05) * LAG 11 + LAG 11
13 IS VARIANCE1
13 = 5 - 11
14 IS SCHSHIS
15 IS GRANTS
16 IS LOANS/STU
17 IS TOT/AID
17 = 14 + 15 + 16
18 IS VARIANCE2
18 = 13 + 17
```

DESIGNER GENES



Imagine—
Your own experimental laboratory
where you can produce
the one creature
that fits your grand design...

Jeepers, creepers—
where'd you get those peepers?
Jeepers, creepers—where'd you get those eyes?

Has anyone ever said that you have "your mother's eyes"—or, perhaps, "your father's nose?" What do they mean by that? Obviously, your parents can't just hand you an eye or a nose; but most of us probably have a pretty vague idea about how one generation passes on these recognizable traits to its offspring. We may know that it has something to do with genes. Some of us may even have enough Biology 101 under our belts to know how and what genes actually do. But how many can unravel the genetic puzzle enough to predict how specific genes lead to specific traits? Now's your chance with *Designer Genes.*

Working To A Goal

In this "edutainment" program, you direct a series of genetic laboratory experiments, pairing off organisms that you have selected especially for their specific traits. As the head of this laboratory, you must work to attain a goal—breeding and isolating an organism with a definite genetic makeup. For your first assignment, you receive a $100,000 research grant. You can also consult a special Bio-Computer to help you predict or analyze the results of your selective breeding experiments. Your work in the laboratory begins with two organisms: the first generation of parents. Through your understanding of genetics (dominant and recessive genes), you selectively breed these creatures and successive generations until you isolate the desired specimen.

Breaking The Code

Within the nucleus of every living cell are the genes that determine every aspect not only of that cell, but of the entire organism, as well. In other words, each cell in the lifeform contains the same genetic code as every other cell. However, specific genes or combinations of genes in that code control the appearance of specific traits. This is true of every living organism, be it as simple as a single-celled amoeba, or as complex as the human body. When two organisms of the same species mate to form offspring, their respective sets of genetic code recombine into a new code. This process takes place in the interaction between the "germ cells" (egg, sperm, spore, etc.) of each organism.

Genes come in pairs (see Chart 1). Each pair either determines a single trait (a physical feature) in the lifeform or works in conjunction with other gene pairs to determine a single trait. In a simple lifeform, such as a fruit fly, a single gene pair by itself can completely determine a trait (the color of the fly's eyes, for example). In more complex organisms, it may take hundreds or even thousands of genes to completely describe a trait (such as a person's height).

To keep the program simple, we have limited the complexity of the genetic "mixture" in your experimental subjects. On the first level of play, we use a single gene pair to fully describe a physical trait. In the second level, we use two gene pairs, each gene pair describing a separate physical trait. In the third and final level of play, two gene pairs work together in describing a single physical trait.

A Matter Of Priority

Each of the two genes in a pair can be either dominant, or recessive—that is, one can have priority over the other. Unless a recessive gene has a recessive partner, the characteristic it represents remains "invisible," and only the dominant characteristic shows up. When two organisms mate, each of their germ cells contains only one half of any given gene pair. So, as these separated genes recombine in a fertilized ovum to form new pairs, characteristics not visible in the parents may "surface" in the offspring.

With a single gene pair controlling a single trait, there are two possible outcomes for that trait—one that exhibits the dominant characteristic, and one the recessive. (In nature, a combination of dominant and recessive genes in a pair can sometimes work together to produce a *third* characteristic. This program keeps things simple by eliminating this possibility.)

Chart 1.

The 'A' Gene Gene Pair #1	A a	= AaBb
The 'B' Gene Gene Pair #2	B b	

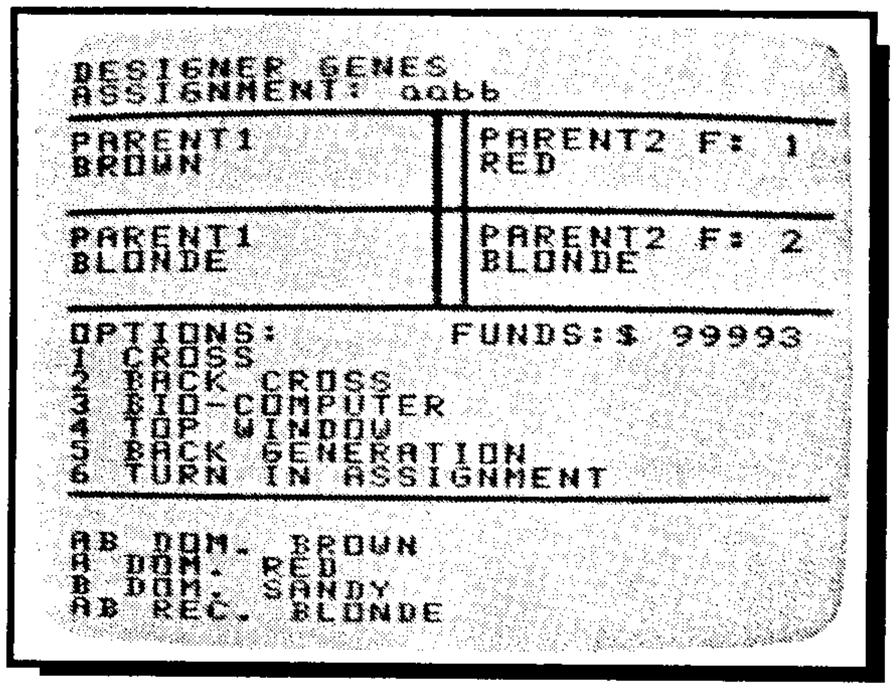


Photo 1. The main game screen features current and past-generation windows at the top, followed by a menu of options, and a description of the genetic combinations.

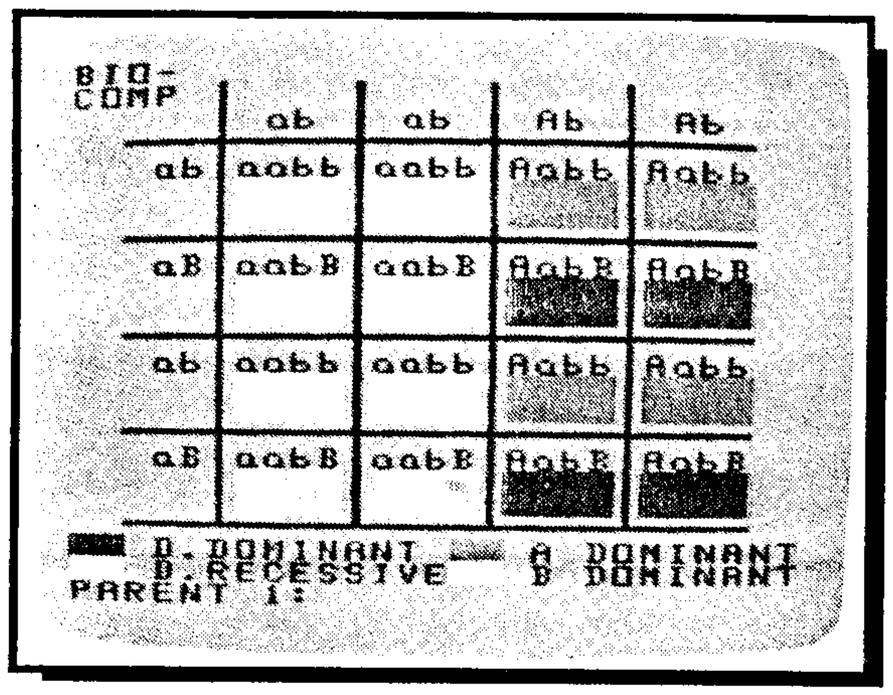


Photo 2. Use the Bio-Computer to predict the genetic makeup and ratios resulting from any hypothetical cross. Simply enter the makeup of the two parents, and the program projects all possible gene combinations of the offspring on the Bio-Computer grid.

Chart 2 illustrates some possible "mixtures" of dominant and recessive genes in a gene pair, and gives the outcome of that mixture. This chart is similar to the grid you will work with on your lab's Bio-Computer. Here, the capital **A** represents the dominant gene, while the lower case **a** represents the recessive gene. The columns marked 1 and 2 contain the genes from one organism (in this case, there is one dominant and one recessive gene). The rows labeled 3 and 4 contain the genes from the second organism (it takes two to tango). This combination of genes in both parents gives us all possible outcomes in the grid. Each cell of the grid represents a possible offspring's genetic makeup.

With the example in Chart 2, an offspring from the two parents has a 25% (one in four) chance of both its genes being recessive (double recessive); and the trait associated with the dominant gene shows up 75% of the time. Try an experiment: Change the first parent—columns 1 and 2—such that both genes are recessive. (Change column 1 to **a**.) Now update the cells of the grid changing one of the capital A's to an **a** in each row of column 1. The ratios of the traits for the offspring now change to 50% dominant and 50% recessive.

To simplify the process of calculating these relationships, the program provides a "Bio-Computer" that simulates the grid we have been studying. On Level 1, with only one gene pair, we use only the top two rows and the left two columns (a 2x2 grid with 4 total cells—as in Chart 2). In Levels 2 and 3, there are two gene pairs at play; so we use all of the grid cells (in a 4x4 grid with 16 cells—see Chart 3).

"Within the nucleus of every living cell are the genes that determine every aspect not only of that cell, but of the entire organism as well."

Playing The Game

When the program first runs, the computer asks if you would like to load a previously saved game. If you have saved a game to disk or tape and wish to continue where you left off, enter **Y**. Follow the system prompts for your machine to enter the file and/or device name of the data file you want to load.

If you do not load a previously saved game, you will be asked to select some options and then enter some parameters to play a new game. The first option is for the level of difficulty. We'll describe the three levels in more detail later. Depending on the level selected, you will be asked to enter the descriptions for a number of dominant and recessive traits. For example, you may designate "blue eyes" as dominant, and "green eyes" as recessive. You will then have the option to let the computer randomly generate both the assignment and the original parents, or you can elect to create your own assignment. If you create your own, you must choose to either let the computer randomly generate the genetic makeup of the original parents, or set up the parents yourself. With your own setup, you must enter a *mixture* of dominant and recessive traits. Trying to "load" the game with an automatic solution will merely bring you back to the previous option.

After you enter the trait descriptions and set up the options as explained above, the computer displays the main game screen. The top line of the display shows the game title. The second line contains the assignment (the problem which you must solve), which consists of a set of genes. You must cross the "critters" (we will refer to a single resultant organism from the sample we are breeding as a critter), until you are sure you have one with the proper genetic code—one that is equivalent to the code in the assignment. Note that the order in which the genes are displayed is not relevant (i.e. **Aa=aA** and **Bb=bB**). Also shown on this main screen is your current operating budget reserve, starting with $100,000.

The Bio-Computer

The Bio-Computer is like a genetic calculator (see Chart 3). You enter the genetic code for two parent critters, and the computer will fill in the grid with all of the possible resultant combinations. It even color codes each cell in the grid, according to the physical description of the traits. If you select the Bio-Computer option from the main game screen, the computer displays a 16-cell grid just like the one in Chart 3. (Chart 3 is much more complex than Chart 2 because of the addition of a second gene pair—the 'B' gene.) You can also select the Bio-Computer by pressing Option E after crossing two critters (see the section on Breeding). When you finish using the Bio-Computer, simply press [ENTER] or [RETURN] at the prompt for the parents genetic code, and you will return to the previous screen.

The Parents

In the upper half of the main game screen, you'll find windows that display the parents from two generations. The parents displayed in the top row of windows can be from a previous, but not the current, generation. The current-generation parents appear in the second row of windows.

In the two windows on the right side of the screen, the number following the letter **F** indicates the generation of the two parents on that row. We use the letter **F** here to indicate "generation" because it is already a common convention in biology textbooks.

You can advance to the next generation with the Cross option from the main game screen menu. This option breeds the two parents from the current generation. Before this takes place, you will also have to specify how many "samples" you want to result from this union (as detailed below under "Crossing Critters").

At the end of the "gestation period," a breakdown of your results will appear on the screen—in terms of how many critters display what characteristics. You then have the option of "keeping" representative samples to become your next generation of parents. If you keep two of the critters created by the cross, then those critters become the current generation parents on the control screen, advancing the current generation by one. The main game screen now shows a reduction in your operating budget due to the deduction of the breeding cost.

You can back up a generation if you discover that you are following a hopeless strain of critters (one that has lost a gene needed to solve the problem), or if you simply need to return to a generation that you now know contains the critter with the proper genetic code to complete the assignment. To back up a generation, simply select the Back Generation option from the main game screen menu. The program asks if you are sure you want to do this, because this option eliminates the current parents (they will no longer exist), making the *previous* generation the *current* generation. If you are sure, enter **Y**.

To place any of the previous generations in the top window, select the Set Generation Top Window option. After selecting this option, the computer asks you to enter a generation. If the generation is legal (exists), then the main control screen reappears with the top windows showing the generation you selected.

The top window indicates which past-generation parents to use during a Back Cross—a cross performed with one current-generation parent and one past-generation parent. (If you choose to keep any offspring resulting from the back cross, the offspring *replaces* the parents in the current-parents window, but it does not advance the generation.) You could also use the top window to simply scan the generations to refresh your memory on what steps were taken to get where you are. The top window may never contain the current generation parents. This is to prevent back crossing a critter with itself (a difficult task at best).

"Because the genes from each parent are selected at random, the same two parents can generate many different offspring."

Crossing Critters

Designer Genes allows you to cross successive generations of critters. In addition, you can cross previous generations with current generations. To produce a critter with the proper genetic code, you need to breed parents to produce offspring. You can accomplish this by using either the Cross or Back Cross options from the main game screen menu.

After you select the Cross or Back Cross option, the bottom half of the screen will clear. You then receive a prompt to enter the number of samples (critters) you would like—the minimum number you can use is 10, maximum is 9999. This is important for determining the accuracy of your experiment. As with any statistical survey, the higher the number of samples, the greater the accuracy. If the selection of offspring were not random, then the offspring produced by the example in Chart 2 would yield 75% dominant and 25% recessive. With the randomness of nature (and your computer), however, you may actually get 68% dominant and 32% recessive. The random sample is closer to a 2 to 1 ratio than the ratio of 3 to 1 predicted by our grid. With larger samples, this margin for error is reduced, yielding more accurate results.

Reading Ratios

When making decisions about the outcome of crossing two parents, you need many samples from which you can calculate ratios. Because the genes from each parent are selected at random, the same two parents can generate many different offspring. You can use the ratio of the total number of critters bred to the number of critters born with a particular physical description to help you discover the parent's genetic make-up.

For example: You are dealing with a single gene pair. You breed two parents and get 100 offspring. Of the 100 offspring, 50% show the dominant trait, and 50% show the recessive trait. After trying several combinations of parents on the Bio-Computer, you arrive at the conclusion that one of the parents must be a double recessive because it shows the recessive trait; and the other parent must have a dominant gene and a recessive gene. So, using the ratio of the offspring, you have determined the exact genetic makup of both parents.

Chart 2.

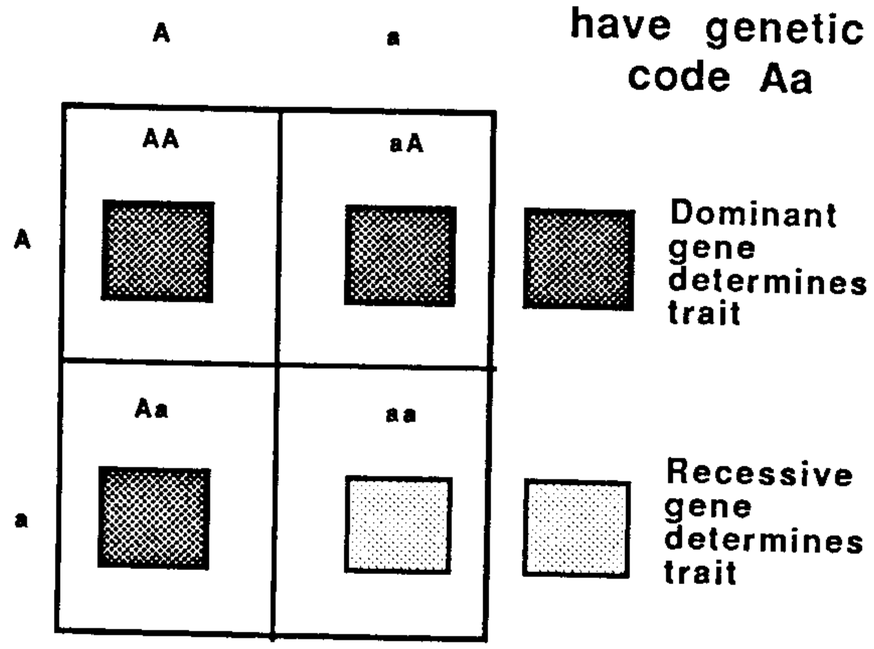
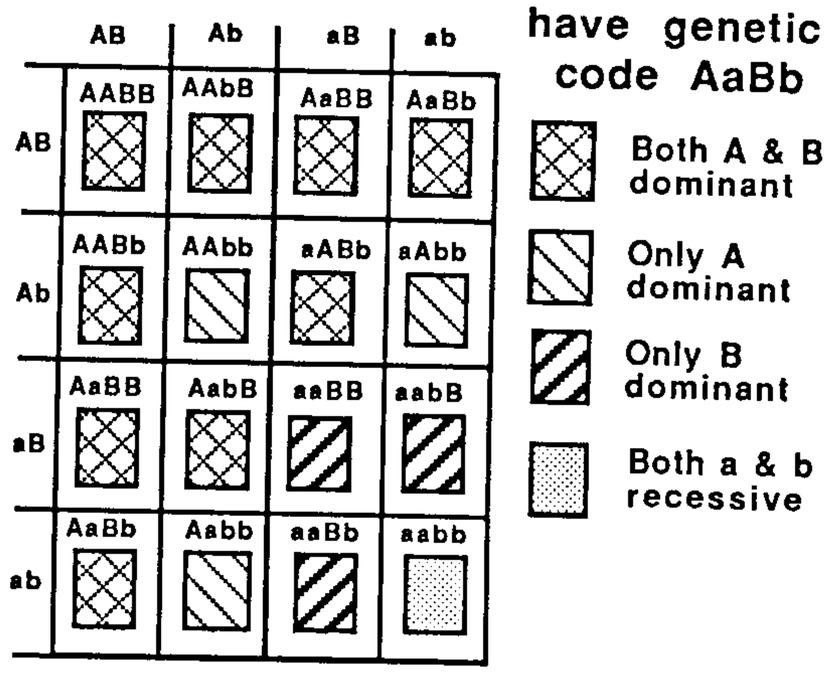


Chart 3.



High School Biology—Level 1

In this first level, you are concerned with a single gene pair and a single trait. The gene we will work with is labeled **A**. A capital **A** indicates a dominant gene, while a lower case **a** indicates a recessive gene. Four possible gene pairs can be made from the dominant and recessive genes: **AA, Aa, aA,** and **aa**. (The two gene pairs **Aa** and **aA** amount to the same thing.)

Level 1 focuses on only one trait. This trait can take one of two decriptions depending on the status of the genes; the dominant and the recessive description. The dominant description of the trait shows up whenever there is a dominant gene present in the pair. The recessive description of the trait shows up only when both genes are recessive (see Chart 2).

Biology Major—Level 2

This level is much more complex than Level 1, because you must now work with two gene pairs and two traits. Each gene pair has absolute control over the outcome of one trait.

After you select Level 2, the computer asks you to enter a dominant and recessive description for each of the two traits. After entering the descriptions, you will be taken through the option screens as described above.

As in Level 1, the first gene pair is called the **A** pair. We will assign the letter **B** to the second gene pair. A capital **B** signifies a dominant gene, and a small **b** indicates a recessive gene. There are four possible pairs of the **B** gene: **BB Bb bB bb**

When asked to enter the code for the parents, you must enter both the **A** gene and then **B** gene (e.g., **AaBb**). The following is a table of the possible combinations of genes with two gene pairs (see also Chart 3):

Chart 4.

This handy form will help you keep track of succeeding generations and results as you conduct your experiments in the genetic laboratory. Keep a supply available for your lab work.

Assignment		Level 1	Level 2	Level 3
'A' GENES	'B' GENES	D:	1 D:	A&B D:
		R:	R:	A D:
		D = Dominant	2 D:	B D:
		R = Recessive	R:	a&b R:

No.	Parents Bred				Breeding Results				Offspring				Notes
	1		2						1		2		
	F	#	F	#	A	B	C	D	F	#	F	#	
1													
2													
3													
4													
5													
6													

AABB AaBB aABB aaBB
AABb AaBb aABb aaBb
AAbB AabB aAbB aabB
AAbb Aabb aAbb aabb

Genetics Engineer—Level 3

In the third level, you still must work with two gene pairs—however, these two pairs together affect only one trait. The combination of the two gene pairs determines the "condition" of the trait. There are 4 possible conditions: (1) both the **A** and the **B** genes are dominant; (2) only the **A** gene is dominant; (3) only the **B** gene is dominant; or (4) neither gene is dominant.

Each of these conditions can result in a different physical characteristic. For example, if the two gene pairs determined the eye color in a rare African frog, these might be your descriptions:

GREEN EYES Both A and B dominant
BLUE EYES A dominant, B recessive
BROWN EYES A recessive, B dominant
WHITE EYES 1Both A and B recessive

As in the other two levels, you are asked to enter a description for each of the 4 conditions listed above.

Turn In Assignment

When you are sure you have the proper genetic code in one of the current parents, select this option to turn in the assignment. The computer asks you to enter the number of the parent that you think contains the proper genetic code. If you are right, the program congratulates you, displays your score, and asks if you want to play again.

If you turn in the wrong answer—a parent with the incorrect genetic code—the program deducts $50,000 from your budget. So if you only have $50,000 or less, you lose the game.

If you have enough money to pay the fine, the computer displays a conciliatory message and asks if you would like to play again. The program then returns you to the main game screen. A word of caution: Because you start with $100,000, and the fine for turning in the wrong answer is $50,000, two wrong answers guarantee the end of the game.

Loading/Saving Games

To save a game that you are unable to complete in one sitting, press the key described in the Control Capsule for the Load/Save option. Your computer then displays a menu with these three options:

1) Load Previous Data
2) Save Data
3) Exit Screen

If you want to quit the game you are currently playing and start a new one which you have previously saved, select the Load Previous Data option. Follow the prompts (if any) for entering the file and/or device name.

To save the current game, select the Save Data option. (Atari users: At prompt, enter **C:Filename** to save to cassette, or **D:Filename** to save to disk.) When the operation is complete, the computer returns to the main game screen.

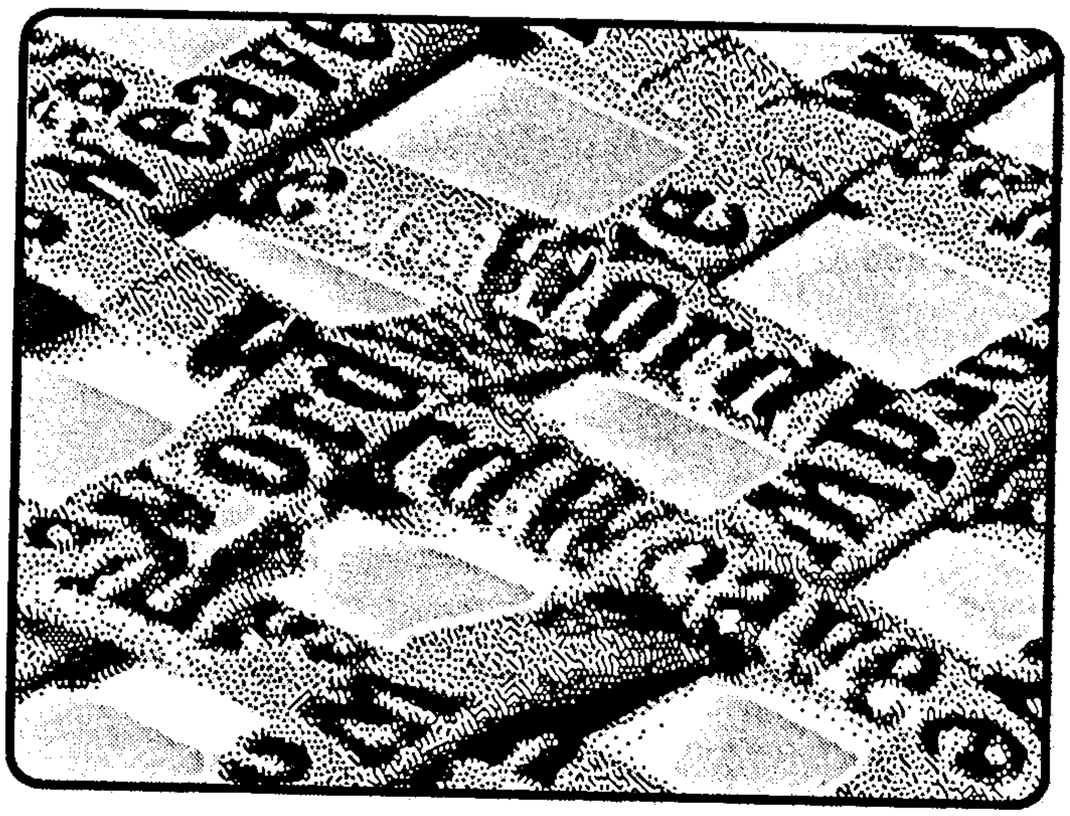
If you made a mistake by selecting this option and do not want to load or save data, select the Exit Screen option to return to the main game screen.

HCJ

CONTROL CAPSULE

Designer Genes

Function	Apple	Atari	C-64	IBM	TI
Load and Save Menu	ESC	CTRL O	←	ESC	FCTN 9



Word Weave

Here's a specialized word processor for creating a branching text stream—be it interactive fiction or a host of practical multiple-path applications. . .

We all know the story:
Rapunzel was locked high in a tower by a wicked witch. Every morning the witch would go to the foot of the tower and cry:
Rapunzel, Rapunzel—let down your golden hair!
Rapunzel would let her long braids drop through the window, and the witch would climb up.
One day the witch demanded that Rapunzel let down her hair as usual. But when Rapunzel came to the window, the witch saw that she had gotten a Mohawk.

Hold it! A Mohawk?! Well, perhaps another cut would do—or an entirely different twist: Rapunzel dumps a pail of dirty water! And then what happens? Ahh, what a tangled plot we can weave—with *WordWeave*...

Woven Text

WordWeave is a structured word processor for creating interactive stories and other multiple-path text applications. (See the sidebar, "What You Can *WordWeave*.") These stories have a variety of surprising turns and numerous endings, because they allow the reader to make decisions as the plot develops. Do you want Rapunzel to let down her hair, or would you prefer she got a Mohawk? From each page, the text branches to a new page—and a new turn of events—depending on the choice of action (see Photo 1).

"Tracing through a woven fabric of text, both the writer and the reader have the freedom to determine how a 'story' may develop, page by page."

With *WordWeave*, all kinds of alternative paths are possible. Tracing through a woven fabric of text, both the writer and the reader have the freedom to determine how a "story" may develop, page by page. In this program, a page consists of one screen of text and the option to branch to 4 different pages.

To use *WordWeave*, load and run the program from disk, and then remove the program disk from the drive—it is no longer required. *WordWeave* operates from one menu containing the following options:

Read
Write
Print
Prepare Disk
Quit

Preparing The Loom

Before you can create a story, you must prepare a *WordWeave* data disk using the menu's Prepare Disk option. The disk that you prepare should be a blank initialized disk. You will need one disk for each text file that you write.

When you choose the Prepare Disk option, the program prompts you for a title and an author. These two items are the first to appear on the screen whenever you choose to Read or Write with *WordWeave*.

Write

Once you have prepared a data disk, you are ready to start weaving your multiple-path text. Before choosing this option, you must place your disk into the disk drive. (If you don't, the program generates an error message and brings you back to the main menu.) As mentioned, the title page appears first. You can now press [RETURN] or [ENTER] (as your machine requires) to turn the page and start writing.

In both the Write and Read options, the screen always displays the page number in the upper-left corner of the screen. The upper-right corner shows the number of options that are available to the reader. Each time you, the writer, add another branch from this page, the number of options increments.

Write mode provides several editing options: You can enter a screen of text, define the different pages that this page may branch to, go back to the previous page, go to any desired page, or return to the menu. There is a different keypress for each of these options. See the Control Capsule for your particular computer for a complete list of these options and keypresses.

When writing a page of text, you have all the standard editing features that your system allows (i.e., insert, delete, etc.). You can even add a header that titles each of the different pages.

A simple keypress (see your Control Capsule) allows you to toggle the Branches menu on and off. This menu appears at the bottom of the screen and displays the various Branch options that are available to the reader. You can specify up to 4 branches, each of which can branch to any page. You can even have a page branch back to itself. You do not have to use all 4 branches—in fact, you don't have to use any. Here is an example of a Branch menu:

```
----------BRANCHES----------
1- Have Rapunzel let down her hair
2- Give Rapunzel a Mohawk
3- Pick a new hairstyle
4- Dump the water
```

To edit the Branches menu, press the proper key to make the menu appear. To add a description to a branch, such as "Have

What You Can *WordWeave*

Not all text generated with *WordWeave* has to be in the form of interactive stories. You can use this program to organize any kind of branching text—fiction or non-fiction. One possible application might be something like a tree-identification guide. Page 1 could ask about the shape of the tree's branches. Are they twisted, straight, multi-limbed...? Depending on the answer, the program will branch to a different page. This new page may ask about the shape of the leaves. Through a series of such logically linked questions, the "story" can lead the reader to a proper identification.

On the subject of foliage, how about *family* trees? Page 1 could profile your own life. Branching from page 1 may bring you to different pages that describe either your mother's or your father's life. And from these pages, you could decide whether to trace the life of your parent's mother or father, and so on. Just as your family grows and branches out, so does the story.

And as well as reaching into the past, this program can also help you anticipate *future* problems and solutions. For example, *WordWeave* is an excellent format for a troubleshooting manual—one that leads you to the proper fix according to which "symptoms" you uncover.

All kinds of practical and creative works are possible with this well-woven program. So, weave on!

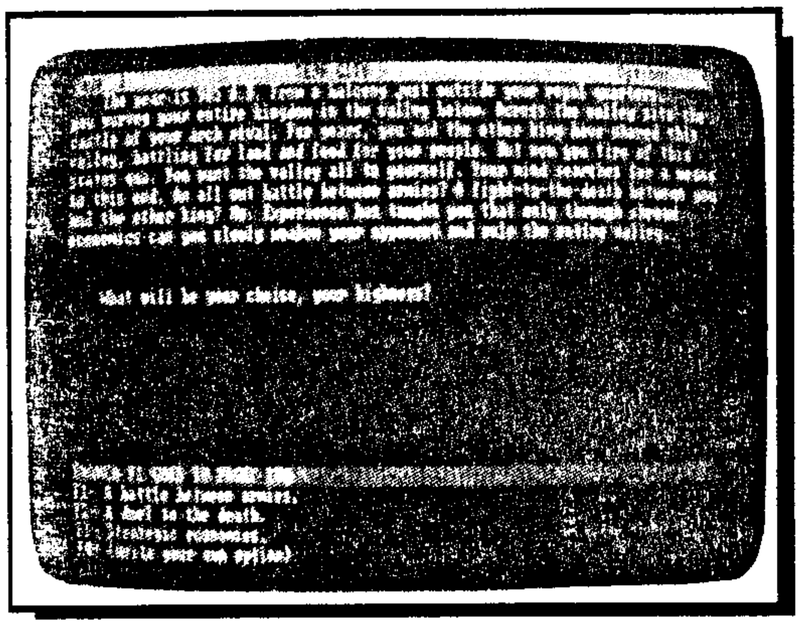


Photo 1: *This screen from the IBM version of WordWeave shows text entered in the main area, with Branch options in the menu window, below. Now in Write mode, the writer is designating a page to which a particular option will branch when a reader selects it from the menu.*

Rapunzel let down her hair," simply press the key corresponding to the branch that you wish to edit and enter your description. To designate which page an option branches to, put the cursor on that Branch line (1-4) and use the "Branch Destination" keypress from the applicable Control Capsule. (Note to TI computer users only: You must press the proper keys while the cursor is scrolling through the top of the Branches menu.) Now the program prompts you for the page number that this branch should go to. If you do not enter a number, the computer defaults to the next available empty page—or, if you have previously entered a number for this branch, it defaults to that page number. Once you enter the number, *WordWeave* immediately takes you to that page. If you wish to "prune" a branch from a page (that is, erase a branch previously set), simply enter a page number of zero when specifying where the branch goes to—effectively truncating that branch from the page.

If, for any one page, you have more then 4 branches in mind, use the fourth branch to go to another page, where you may continue to present alternatives.

If you want to go back to the previous page, press the key specified in your Control Capsule. To go to any page at all, make sure that the Branch menu is not present, then press "Destination" key, enter the desired page number, and away you go. To return to the main menu, press your "Exit" keys.

Now it's time to start writing! Be creative and experiment. For example, try making your story into a maze. Pretend that the pages are rooms and each of the 4 branches are exits—north, south, east, and west. The possibilities go on and on...

Read

Before you choose this option, make sure that the data disk with the text that you wish to read is in the disk drive. The first thing that you see on the screen when you choose the Read option is the title page. Press [RETURN] or [ENTER] to turn the page.

In this mode, you can display the Branches menu, select a Branch option, go back to the previous page, or return to the main menu. There is a different keypress for each of these options. See your computer's Control Capsule for a complete list of these options and keypresses.

From Read mode, you can venture forth into a story's every turn, twist, and pitfall. You control the events by making your own choices as to which direction the story branches. If you don't like the way the plot is going, you can always return to the previous page and make a different decision. If you don't like the way the story is reading, you can always go back into Write mode and change it!

Imagine: You are floating down a calm and peaceful river in Africa, when suddenly you are attacked by a group of violent savages. Startled, you reach for your gun. Just as you steady your aim on one of your spear-waving opponents, you realize that your raft is headed straight for a waterfall. Do you: (1) abandon ship and swim for it; (2) fire at will and worry about the waterfall later; or (3) paddle like heck towards shore and hope these savages are just friendly vegetarians? In Read mode, the choice is yours. You make the decisions; you determine the fate of the story.

Print

With this menu selection, you can print all or part of a *WordWeave* story. When choosing this option, make sure that your data disk is in the disk drive. Because the print routine is set up to print two pages of text per one sheet of fan-fold paper, be sure to begin the printout at the top-of-form. The computer asks you for the starting and ending numbers of the pages that you wish to print. Once you have entered this data, the computer prints each page of text, including the Branches menu, which specifies the page number that each option branches to.

Quit

Select this option to exit the program. Don't worry: You are in no danger of losing data when you exit the program. All of a story's information is kept on your data disk. HCJ

CONTROL CAPSULE

WordWeave

KEY	FUNCTION
Write Mode	
CTRL T	Edit header
RETURN	Exit header
CTRL G	Go to a page
ESC	Go to previous page
CTRL X	Exit to main menu from any page except title page
CTRL B	Toggle Branches menu
CTRL D	Delete character
CTRL I	Insert space
Editing Branches menu	
CTRL B	Toggle Branches menu
CTRL G	Define a branch destination

KEY	FUNCTION
Editing Branches menu (cont.)	
RETURN	Edit branches 1-4
UP & DN ARROWS	
CTRL X	Exit to main menu
Read mode	
CTRL B	Toggle Branches menu
1	Choose Branch 1
2	Choose Branch 2
3	Choose Branch 3
4	Choose Branch 4
ESC	Go to previous page
CTRL X	Exit to main menu from any page except title page

REMARKS:
During Write mode on the Apple version of *WordWeave*, pages are saved as sequential text files on the disk. Each page is a separate file. If you look at the directory of the disk, you find file names such as WT.01 and WT.02 for pages 1 and 2. This process of saving pages as text files continues until the volume is full, giving a total of 51 pages, including the title page. It is therefore suggested that you start with a clean formatted disk for each volume of your *WordWeave* library.

CONTROL CAPSULE

WordWeave

KEY	FUNCTION
Write mode	
CTRL H	Edit header
RETURN	Exit header
CTRL G	Go to a page
ESC	Go to previous page
CTRL B	Toggle Branches menu
CTRL Delete	Delete character

KEY	FUNCTION
Write mode (cont.)	
CTRL Insert	Insert space
CTRL X	Exit to main menu from any page except title page
Editing Branches menu	
CTRL B	Toggle Branches menu

KEY	FUNCTION
Editing Branches menu (cont.)	
RETURN	Edit Branches 1—4
UP & DN CURSOR	
CTRL G	Define a branch destination
CTRL X	Exit to main menu

KEY	FUNCTION
Read mode	
CTRL B	Toggle Branches menu
1	Choose Branch 1
2	Choose Branch 2
3	Choose Branch 3
4	Choose Branch 4
ESC	Go to previous page
CTRL X	Exit to main menu from any page except title page

REMARKS:

During the Prepare Disk option on the Atari version of *WordWeave*, the disk is initialized such that it is dedicated to the WordWeave program alone. Because the disk is actually reformatted so all files on the disk are destroyed, don't use a disk that has anything on it you wish to keep. Only one WordWeave file can be stored on the disk, so don't use the disk for anything except the one WordWeave file. Also, if you look at the directory of the disk it appears to have nothing on it—always place a label on a WordWeave disk so you don't accidentally use it for something else.

Note: Due to this program's unique method of screen memory management, it works only on the Atari 800XL and 130XE. See "HC Journal I/O" for a special "Hacker's Challenge."

CONTROL CAPSULE

WordWeave

KEY	FUNCTION
CTRL P	Print screen
Write mode	
CTRL H	Edit header
RETURN	Exit header
CTRL G	Go to a page
←	Go to previous page
CTRL X	Exit to main menu from any page except title page
CTRL B	Toggle Branches menu

KEY	FUNCTION
Write mode (cont.)	
SHIFT CLR	Erase screen
HOME	Home cursor to top of screen
Editing Branches menu	
F1	Edit Branch 1
F2	Edit Branch 2
F3	Edit Branch 3
F4	Edit Branch 4

KEY	FUNCTION
Editing Branches menu (cont.)	
RETURN	Edit next branch
CTRL G	Define a branch destination
CTRL X	Exit to main menu
CTRL B	Toggle Branches menu
Read mode	
F1	Choose Branch 1
F2	Choose Branch 2

KEY	FUNCTION
Read mode (cont.)	
F3	Choose Branch 3
F4	Choose Branch 4
←	Go to previous page
CTRL X	Exit to main menu from any page except title page
CTRL B	Toggle Branches menu

REMARKS:

The Prepare Disk option on the Commodore 64 version will actually initialize your disk for you, so make sure that the disk you prepare does not contain anything you with to keep.

On this version you also have an option that is not available on the other machines: At any time, you can send a screen dump to the printer by pressing [CTRL] P. The screen dump that *WordWeave* uses is written in machine language.

CONTROL CAPSULE

WordWeave

KEY	FUNCTION
Write mode	
Fn 5	Edit/Exit header
Fn 6	Go to a page
ESC	Go to previous page
Fn 8	Toggle Branches menu
Fn 7	Exit to main menu from any page except title page

KEY	FUNCTION
Editing Branches menu	
Fn 8	Toggle Branches menu
Fn 1	Edit Branch 1
Fn 2	Edit Branch 2
Fn 3	Edit Branch 3
Fn 4	Edit Branch 4
Fn 6	Define a branch destination
Fn 7	Exit to main menu

KEY	FUNCTION
Read mode	
Fn 8	Toggle Branches menu
Fn 1	Choose Branch 1
Fn 2	Choose Branch 2
Fn 3	Choose Branch 3
Fn 4	Choose Branch 4
ESC	Go to previous page
Fn 7	Exit to main menu from any page except title page

REMARKS:

The IBM version of *WordWeave* provides for more text to be entered than any of the other versions. The screen display takes advantage of the 80-column mode for text input and a total of 100 pages is allowed for each story or volume.

Another unique feature in the Write mode allows the cursor to wrap around the screen right to left and bottom to top (or vice versa)—a useful accommodation when editing.

CONTROL CAPSULE

WordWeave

KEY	FUNCTION
Write mode (cursor scroll mode)	
CTRL H	Edit header
ENTER	Exit header
CTRL G	Go to a page
FCTN 9	Go to previous page
CTRL X	Exit to main menu from any page except title page
CTRL B	Toggle Branches menu
CTRL W	Write text (non-memory expansion only)

KEY	FUNCTION
Writing text	
FCTN E	Move up a line
FCTN X	Move down a line
ENTER	Return to write mode (cursor scroll mode) (non-memory expansion only)
Editing Branches menu	
1	Edit Branch 1
2	Edit Branch 2

KEY	FUNCTION
Editing Branches menu (cont.)	
3	Edit Branch 3
4	Edit Branch 4
CTRL G	Define a branch destination
CTRL B	Toggle Branches menu
Read mode	
1	Choose Branch 1
2	Choose Branch 2

KEY	FUNCTION
Read mode (cont.)	
3	Choose Branch 3
4	Choose Branch 4
FCTN 9	Go to previous page
CTRL X	Exit to main menu from any page except title page
CTRL B	Toggle Branches menu

REMARKS:

The TI-99/4A version of *WordWeave* runs differently without memory expansion. The difference is in Write mode. When you first enter Write mode, your cursor circulates across the top of the screen (cursor scroll mode), telling you that it is ready for a keypress. From here you must press one of the keys specified in the Control Capsule to enter any one mode. Specifically, you must press [CTRL] W to enter any text onto the screen. With memory expansion, however, there is no need to press [CTRL] W because you are always editing the screen's text. The memory expansion version takes advantage of a special full-screen editor written in machine language specifically for the *WordWeave* program.

With or without memory expansion, when defining branches, the cursor scrolls through the top of the Branches menu to inform you that it is ready for a keypress. When you choose to define a branch, you must press [CTRL] G while the cursor is scrolling like this. Once pressed, you are prompted for the number of the branch that you wish to define and then the page number that this branch should go to.

DIGITAL DUMPSTER

Can mathematics solve the nuclear waste dilemma? This simulation may provoke many answers—but perhaps not the one you're looking for...

As you climb into the form-fitted seat with an array of knobs, dials, and buttons spread before you like a neon quilt, you suddenly feel a shiver run up your spine. "This is it," you mutter—pressing the activate button. You can actually feel the power surge into the machinery as the sub-modules come to life. Suddenly, a previously blank video screen begins to glow, displaying the greeting, "Welcome to the EPA Nuclear Waste Disposal Simulator." You take a deep breath and try to calm down as the test is about to begin.

How do you handle the disposal of radioactive waste? Who knows? If it's all buried in one place, it may affect the quality of the surrounding ground water. You can't just throw it in the city dump, because it may combine with other unknown wastes to form a gruesome slime monster right out of some science fiction B movie. Lots of proposed solutions exist, but nothing like the *Digital Dumpster*...

What's In A Game?

Digital Dumpster is a simulation game for would-be EPA controllers. Its secret function, however, is to teach algebraic thinking and binary-to-decimal conversion. Before you can dump any wastes, you must first use a binary routing control to pre-condition the "hot dust." Only then can you—as the Digital Dumpster Dispatcher (DDD, for short)—arrange for transportation of the hazardous material.

Your final and most crucial responsibility involves indentifying and deciding where to dump the waste. There are 4 dump sites available. All 4 dump sites are close enough to one another that each affects the other two sites. You must mathematically balance storage among these sites to safely contain the radioactive dust. For each game, the ideal balance always appears as a simple mathematical expression—an expression that produces a whole-number result. By depositing the right "storage number" at each site, you can duplicate this ideal expression—and thus achieve the ideal balance.

Using The Simulator

When you start the program, it presents the title screen and 4 options:

1) Apprentice
2) Field Representative
3) Supervisor
4) Exit Program

After selecting the level of difficulty (explained in detail later), you will go to the simulator screen. This simulator consists of a system of tubes that direct the radioactive waste from the top of the screen (see Photo 1). All waste enters the system through a single tube at the top, which branches into two tubes. Each tube then branches into two more tubes, creating 4 tubes. There are 16 possible outlets for the waste at the bottom of the tubes (numbered 0 to 15). At each level of branching, you control which branch carries the waste by toggling the switches on the left side of the screen. Setting the right combination of switches at the 4 branch levels determines which of the 16 tubes eject the "hot dust." On closer inspection, you start to notice a binary-to-decimal relationship between the switches and the number of the tube that ejects the waste. The switches actually represent a 4-bit binary number.

"... you must first use a binary routing control to pre-condition the hot dust. Only then can you arrange for transportation of the hazardous material."

Bit By Bit

There is a simple method for converting from the binary numbering system to the decimal numbering system (see Chart 1). Start with the "least significant" bit (in this case it is the bottom switch) and assign it a value of one. For each bit after this first bit (bit 0), double the value you assign to it (bit 1 has a value of 2, bit 2 has a value of 4, bit 3 has a value of 8). To find the value of a binary number, simply add the assigned values together for any bits which are turned on (set to 1). For example the binary value 0101 has a decimal value of 5 (bit 0 [1] + bit 2 [4] = 5). If you were to turn the bottom switch (bit 0) to 1, and the switch next from the top (bit 2) to 1, and set the other two switches to 0, then the waste material will be ejected from tube number 5.

If you wish, however, you can ignore all this and merely concentrate on the position of the "gates" within tubes as you flip the switches back and forth—thereby clearing pathways to the proper numbers at the bottom.

Don't Waste The Waste

Digital Dumpster's filtering system of tubes occupies the top two-thirds of the screen display. Immediately below the loading tubes is a ramp that the waste disposal truck uses to get from one tube to another. Below the ramp are 4 bins, with a small box between each bin.

Back the truck up to any one of the numbered tubes using the keys specified in your machine's Control Capsule. You must then select that tube by setting the proper switches—pressing 1, 2, 4, or 8 for the switch you wish to change. (Each switch is set to either 1 or 0; pressing the key for that switch toggles between these two values). Press [ENTER] or [RETURN] (Up Cursor on Atari machines) to load the truck. If you select the wrong tube, the waste spills out onto the ground, contaminating you and all of your co-workers.

Whenever you load waste from a tube, the tube is completely emptied. The only way to get any more from that tube is to press the simulator's reinitialize button to start over (see your machine's Control Capsule). Using this key, however, is not taken lightly with the EPA, so be careful not to waste your waste.

Once a tube has dumped its waste into the truck, the truck can move to any of the 4 bins to unload cargo. Each time you dump into a bin (see Control Capsule), it displays the number of the tube from which your truck received the waste.

Expressing The Goal

Each of the 4 bins at the bottom of the screen can contain a number from 0 to 15. The three small boxes between the bins each contain an operator (+ for addition, – for subtraction, and * for multiplication). The operators are never repeated, so all three operators always appear—but the order in which they appear from one box to the next is always random.

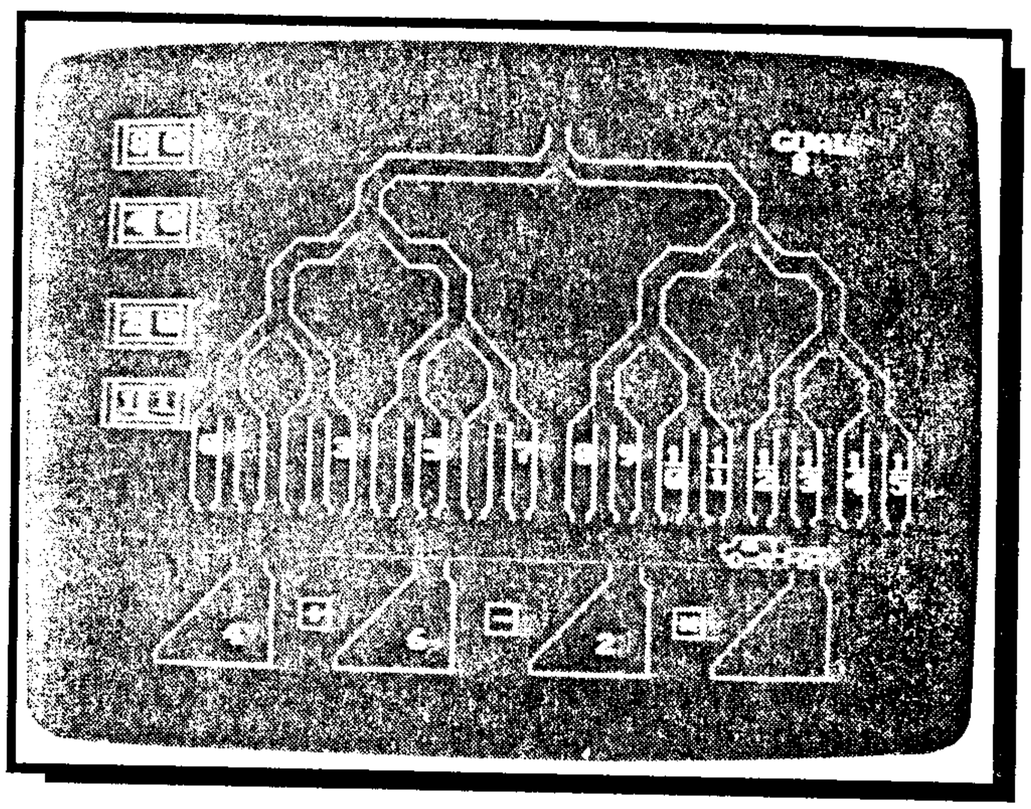


Photo 1: *In this photo from the IBM version of Digital Dumpster, a system of tubes with binary gates provide 16 "mixes" of nuclear waste, each asssigned a numeric value. Placing these gates in the proper position dumps specific numbers, which the truck can shuttle into the bins at the bottom. A correct arrangement of numbers in the bins forms a mathematical expression resulting in the desired "goal" figure shown at the upper right.*

At the beginning of the simulation, you receive an assigned value. This value appears at the top of the screen near the word "GOAL." You must create an expression that produces the value given. To do this, follow the number selecting and dumping procedure outlined above, until each bin contains a number, and the series of numbers and operators forms an expression.

Let's suppose that our first problem has a goal of –12, and a sequence of operators that subtracts, adds, then multiplies (_ _ + * = –12). One possible solution is to subtract 10 from 3 (–7), add 5 (–2), and multiply by 6 (–12).
There is never a single "right answer"—there may be, in fact, several right answers. The simulator recognizes all right answers, because it actually uses the values you place into the bins to do the calculation to see if it produces the proper "GOAL" value.

Levels Of Difficulty

The main difference in play between levels is that in level 1, all three operators are visible; in level 2 only one operator is visible; and level 3 displays no operators at all. Note that the position and type of operators are always random. This means that you need to experiment with several values in order to determine what the hidden operators are. If you guess wrong, your only clue is a prompt stating that your nuclear-waste-balancing expression generates a value too high or too low. To change a value in a bin, simply dump a new value on top of it. To replace a value in the truck before dumping, simply load another value into the truck from the tubes. Because you can dump each number only once, replacing the value previously in the truck removes that value from the game.

Scoring works as follows: Higher scores are possible at higher levels. The highest possible score is 500 on level 1, 750 on level 2, and 1000 on level 3. Each time you load the truck, 10 points subtract from the total. Each time you unload the truck's cargo, another 10 points subtract. If you use the reinitialize numbers key, 100 points subtract from the score.

Note that a player can load a number and then decide to load another number without first dumping the original. This results in a 20-point reduction from the total score, and the loss of the original number.

HCJ

Chart 1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Decimal Value
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Hexadecimal Value
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	Bit 3 / Value Of 8
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	Bit 2 / Value Of 4
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	Bit 1 / Value Of 2
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	Bit 0 / Value Of 1

0 = Bit / Switch Off
1 = Bit / Switch On (add Bit Value to total)

CONTROL CAPSULE — *Digital Dumpster*

KEY	FUNCTION	KEY	FUNCTION
Left Cursor	Move truck left	1	Toggle bit 0
Right Cursor	Move truck right	2	Toggle bit 1
Down Cursor	Dump truck load	4	Toggle bit 2
RETURN	Open loading chute	8	Toggle bit 3
ESC	Return to main menu	0	Reinitialize numbers key

CONTROL CAPSULE — *Digital Dumpster*

KEY	FUNCTION	KEY	FUNCTION
Left Cursor	Move truck left	1	Toggle bit 0
Right Cursor	Move truck right	2	Toggle bit 1
Down Cursor	Dump truck load	4	Toggle bit 2
Up Cursor	Open loading chute	8	Toggle bit 3
CTRL X	Return to main menu	CTRL R	Reinitialize numbers key

CONTROL CAPSULE — *Digital Dumpster*

KEY	FUNCTION	KEY	FUNCTION
Left Cursor	Move truck left	1	Toggle bit 0
Right Cursor	Move truck right	2	Toggle bit 1
Down Cursor	Dump truck load	4	Toggle bit 2
RETURN	Open loading chute	8	Toggle bit 3
Function 5	Return to main menu	Function 3	Reinitialize numbers key

CONTROL CAPSULE — *Digital Dumpster*

KEY	FUNCTION	KEY	FUNCTION
Left Cursor	Move truck left	1	Toggle bit 0
Right Cursor	Move truck right	2	Toggle bit 1
Down Cursor	Dump truck load	4	Toggle bit 2
ENTER	Open loading chute	8	Toggle bit 3
Function 9	Return to main menu	Function 8	Reinitialize numbers key

CONTROL CAPSULE — *Digital Dumpster*

KEY	FUNCTION	KEY	FUNCTION
Left Cursor	Move truck left	1	Toggle bit 0
Right Cursor	Move truck right	2	Toggle bit 1
Down Cursor	Dump truck load	4	Toggle bit 2
ENTER	Open loading chute	8	Toggle bit 3
FCTN 9	Return to main menu	FCTN 8	Reinitiallze numbers key

TI FLASHLIGHT

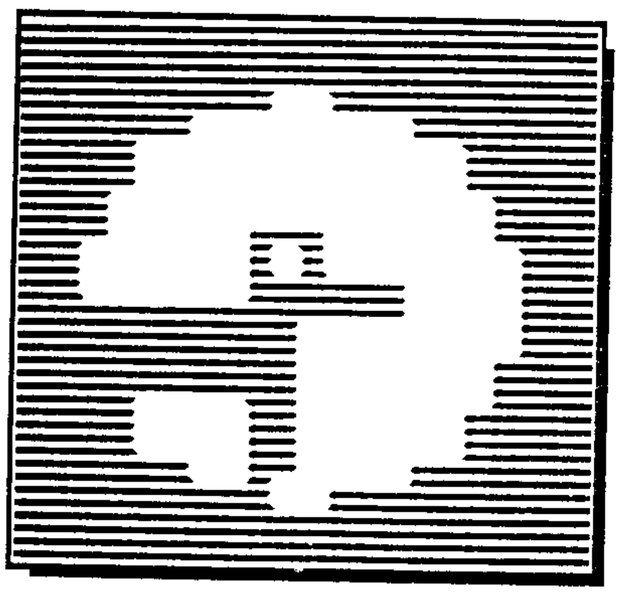
Put the spotlight on your TI-99/4A computer and discover some "hidden" uses for Extended BASIC's sprite graphics.

You wake up hungry, in search of food. Stepping away from the bed, you reach for the flashlight located on your nightstand. Suddenly, you remember the attack dog given to you by your mother at Christmas. He is in your room! You do not dare take another step unless you know exactly where your foot will land. Carefully you turn on the flashlight, hoping to locate the room's light switch. You think to yourself: "If I can find the light switch, I can turn on the light. If I can turn on the light, I can see the dog. And if I can see the dog, I can get out of the room and feed my grumbling stomach. Now, where is that light switch?" Fortunately, you find the switch and are able to feed both yourself and your dog. Once again, another life-threatening situation is safely averted.

Discovering Hidden Objects

The scenario may differ, but the purpose is often the same—find the hidden object. Many computer games employ this principle. Such games present buried treasure for you to discover or hidden bombs that you must defuse before they explode. In the example above, the hidden object is a light switch.

Maybe you have considered writing a similar program yourself. Something that would challenge the player to find certain objects that are hidden on the screen. Well, there is an easy and impressive way to reveal hidden objects using Extended BASIC's sprites. The program presented here, *Flashlight* illustrates this technique.



Here you can see the flashlight shining on the head of the vicious attack dog.

A Flash In The Dark

Flashlight (listed on the opposing page) uses the same basic scenario as the introductory paragraph. Your task is to find the light switch with the aid of a flashlight. Once it's found, you can turn on the light. When you first **RUN** this program, your flashlight beam appears in the middle of the screen. Using the **S**, **D**, **E**, and **X** keys you can move your flashlight left, right, up, and down, respectively. Once you have found the light switch, pressing **O** will turn the light on. The flashlight beam must be shining directly on the light switch for the **O** key to be able to turn the light on.

While searching the room, you may discover other hidden objects. Notice how these objects are revealed. As with a real flashlight, you see that part of the object where the flashlight is aimed. Sprites hold the key to the smooth unveiling of objects that can really add life to a graphic display.

How It Works

Hiding an object is easy enough. Say we have defined a sprite in the shape of a light switch. To hide it on the screen, simply set the sprite's color to that of the screen. Now, when the light switch is placed onto the screen, it's invisible. In *Flashlight*, both the screen and all the hidden objects are colored black.

To reveal objects, this program uses an all too often overlooked feature of Extended BASIC's sprite graphics—sprite priority. A sprite's priority determines whether it is displayed on top of, or underneath other sprites. When two sprites coincide, the sprite with the highest priority comes out on top. The priority of a sprite is determined by the sprite's number—the lower the number, the higher the priority. So, **SPRITE #1** has the highest priority possible, while **SPRITE #28** has the lowest.

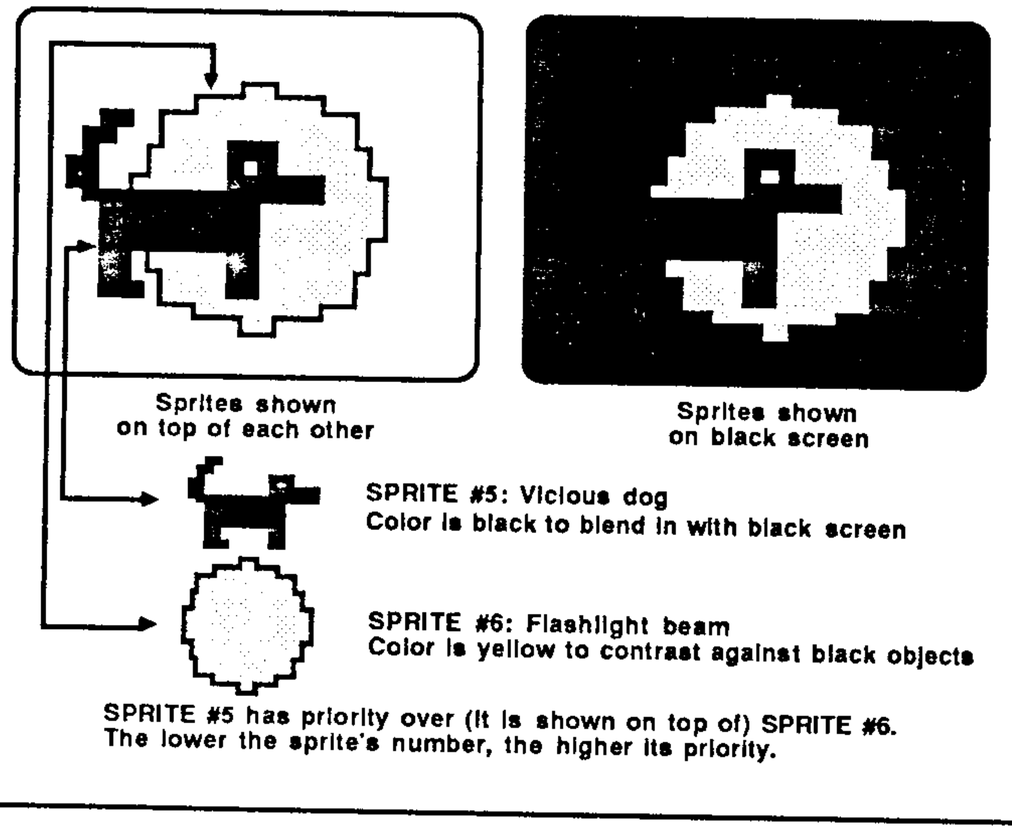
To reveal objects, we give the flashlight beam the lowest priority of all the other sprites on the screen. Now, when the yellow flashlight beam comes in contact with other sprites, it moves under them, providing a more illuminating background.

Figure 1 illustrates this method of revealing sprites. Here, we show how the flashlight beam sprite is covered by the vicious dog sprite. A second picture shows the two sprites overlapping on a blackened screen. Notice that the only part of the dog that is visible is the part with the flashlight beam on it. Because the flashlight beam is yellow, it contrasts against black objects.

To emphasize the sprites, this program uses sprite magnification 4. Line 220 of Flashlight initiates this with the command **CALL MAGNIFY(4)**. This command enlarges all sprites to 4 times their normal size. Because of this, it now takes 4 characters (instead of one) to define a sprite's shape. The sprite shape data is held in the **DATA** statements located at the end of the program.

When you first turn on the light, you may notice the light bulb flicker a bit before actually turning on. We can add this touch of realism fairly easily by alternating the light-bulb sprite's color from black to dark yellow very quickly. Before

Figure 1.



> *"Sprites hold the key to the smooth unveiling of hidden objects that can really add life to a graphic display."*

the light is actually turned on, our light bulb returns to its original color of black.

To turn on the light, all we have to do is change the screen's color from black to yellow with the **CALL SCREEN(12)** command. Because all the hidden sprites are black, they become clearly visible against their new yellow background. The flashlight beam, however, is now invisible as it should be. It is not until you turn the light off that the flashlight beam becomes visible again.

Other Uses

The technique shown here is not limited to revealing hidden objects. How about using a sprite as a window? Other sprites that pass through the window will appear as objects passing by outside the window. How about covering one sprite with several others so that the covered sprite is only visible on certain sections of the screen? There are many great uses for sprites. Try some and see.

HCJ

LINE ANNOTATIONS

Flashlight

Line Nos.	
100-160	Program header
170	Initialize variables
180	Initialize screen
190	Redefine characters
200-210	Initialize sprites
220	Magnify sprites
230	Reset flashlight movement
240	Move flashlight
250	Read keyboard
260	Branch according to key pressed
270	"S" pressed—move left
280	"D" pressed—move right
290	"E" pressed—move up
300	"X" pressed—move down
310-360	"O" pressed—turn on or off light
310	Is switch on?
320	Is flashlight shining on switch
330-340	Toggle switch and flicker light
350	Turn on light
360	Toggle switch and turn off light
370	Sprite data for flashlight beam
380	Sprite data for vicious dog
390-400	Sprite data for "BEWARE..." sign
410	Sprite data for light bulb
420	Sprite data for light switch

KEY VARIABLES

K$ = String of legal keypresses ("SDEXO"); **SWITCH** = Condition of light switch (0=switch is off, 1=switch is on); **Y** = Row velocity of flashlight beam; **X** = Column velocity of flashlight beam; **C** = Flag to detect if the flashlight is shining on the light switch? (0=no, -1=yes); **S$** = Reads character data for sprite shapes; **K** = ASCII of keyboard input; **S** = Keyboard status; **I** and **J** = Loop counters

CONTROL CAPSULE

Flashlight

KEY	FUNCTION
S	Move flashlight beam left
D	Move flashlight beam right
E	Move flashlight beam up
X	Move flashlight beam down
O	Turn on/off light
FCTN 4	Quit program

```
100 ! ***************
110 ! * FLASHLIGHT *
120 ! ***************
130 ! COPYRIGHT 1986
140 ! HOME COMPUTING JOURNAL
150 ! VERSION 1.0
160 ! TI EXTENDED BASIC
170 K$="SDEXO" :: SWITCH=0
180 CALL CLEAR :: CALL SCREEN(2)
190 FOR I=40 TO 63 :: READ S$ :: CALL CHAR(I,S$)::
 NEXT I
200 CALL SPRITE(#1,60,2,90,210,#2,56,2,1,110,#3,52
,2,40,55)
210 CALL SPRITE(#4,48,2,40,23,#5,44,2,160,100,0,24
,#6,40,12,100,100)
220 CALL MAGNIFY(4)
230 X=0 :: Y=0
240 CALL MOTION(#6,Y,X)
250 CALL KEY(0,K,S):: IF S=0 THEN 230
260 ON POS(K$,CHR$(K),1)+1 GOTO 250,270,280,290,30
0,310
270 X=-25 :: GOTO 240
280 X=25 :: GOTO 240
290 Y=-25 :: GOTO 240
300 Y=25 :: GOTO 240
310 IF SWITCH THEN 360
320 CALL COINC(#1,#6,10,C):: IF C=0 THEN 240
330 CALL CHAR(62,"00E0E0E06060E0E0"):: CALL SOUND(
50,-5,0):: FOR I=1 TO 4 :: CALL SPRITE(#2,56,2+9*(
I AND 1),1,110):: FOR J=1 TO 5
340 NEXT J :: NEXT I
350 CALL SCREEN(12):: SWITCH=1 :: GOTO 240
360 CALL CHAR(62,"00E0E0E06060E0E0"):: CALL SOUND(
50,-5,0):: CALL SCREEN(2):: CALL SPRITE(#2,56,2,1,
110):: SWITCH=0 :: GOTO 240
370 DATA 010F3F3F7F7FFFFF,FFFF7F7F3F3F0F01,80F0FCF
CFEFEFFFF,FFFFFEFEFCFCF080
380 DATA 0000003060404C0C0,7F3F3F3F30303800,0000000
00038283F,FFF0F0F030303000
390 DATA FF809C949E929E80,808E8A8A8A8E80FF,FF00EA8
ACF87E500,00E080E0808000FF
400 DATA FF00BBAABB2B2A00,00CEEAAAEACE00FF,FF01B9A
1B121B901,01E181B1A1E101FF
410 DATA 3F07020202040408,0810100804020100,FCE0404
040202010,1008081020408000
420 DATA 0003030303030303,0303000000000000,00E0E0E
0E06060E0,E0E0000000000000
```

TECH NOTE

Mini Memory Battery Replacement

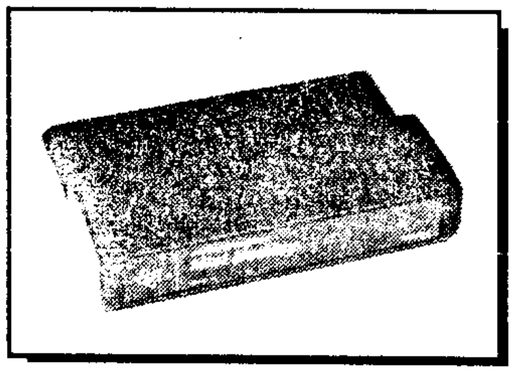


PHOTO 1

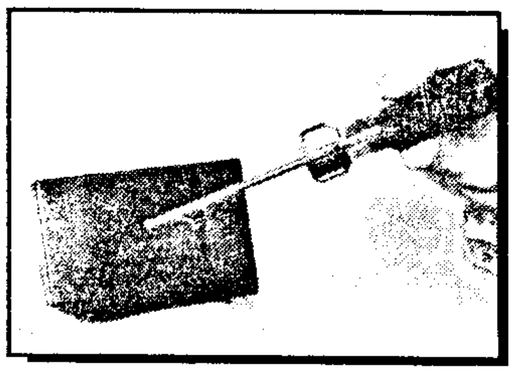


PHOTO 2

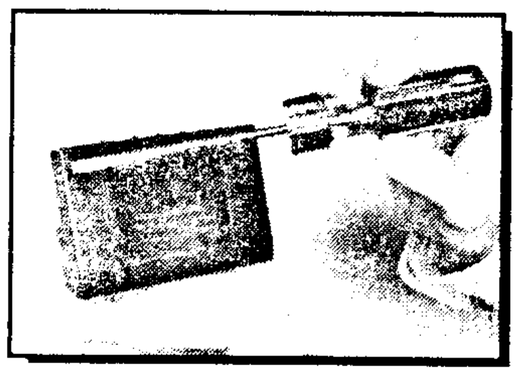


PHOTO 3

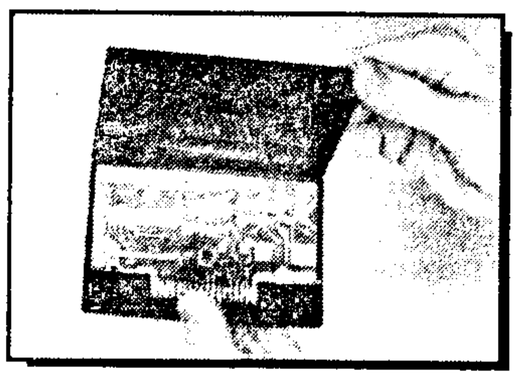


PHOTO 4

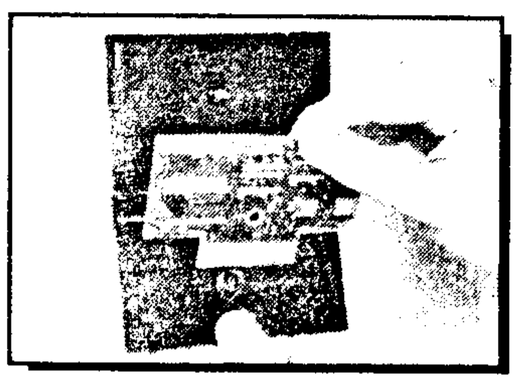


PHOTO 5

Good things never last forever, and so it is with the battery in the TI Mini Memory cartridge. The cartridge (see Photo 1) is battery-powered so that it can retain information stored in it even when the power to the TI-99/4A console is turned off, or the cartridge is removed from the console. Unfortunately, the batteries are not rechargeable, and are not easy to change unless you know the proper procedure. To help get all that you can out of the Mini Memory cartridge, the following describes what you need to change the battery, and how you can complete the task with very few tools.

Tools Required:
- -Light soldering iron (15 to 35 Watts)
- -Copper solder-removing braid or other solder-removal tool
- -Phillips-head screwdriver
- -Small flat-head screwdriver
- -Needle-nose pliers (suggested)

Parts Required:
- -Lithium 3-volt battery (Radio Shack # CR2032)
- -Light wire (18 to 24 gauge, solid) approximately 3"
- -Rosin-core solder

If you have trouble locating the battery listed above, any 3-volt lithium battery around the same size will work. Make sure you have rosin-core solder. Do not use plumber's solder—acid core—because the acid will damage the circuitry. All of the tools and parts listed above can be purchased at any local Radio Shack store, or in most hardware stores.

Step 1. Open the Cartridge

Before you can open the cartridge, you must first remove the screw on the bottom side of the cartridge using a Phillips-head screwdriver (see Photo 2). To open the cartridge, release the two plastic hooks by inserting a flat-head screwdriver into the slots shown in Photo 3. (You do not need to tear the paper label on the cartridge. This label will act as a hinge once the cartridge is opened.) With light pressure, pry the screwdriver toward the center of the module, lightly pulling the two halves apart at the same time. Repeat this in both slots until you are able to open the cartridge as seen in Photo 4. As you open the cartridge, keep your thumb or finger on the dust plate at the mouth of the cartridge. Try not to let this plate or the spring holding it come out, or reassembly will be more difficult.

Step 2. Removing the Circuit Board

Holding the cartridge in one hand with a thumb on the dust plate, pick up the circuit board at the back edge (see Photo 4). Lift up on the back edge until the hole in the center of the board clears the mounting post. Take care not to let the spring under the board get loose (see Photo 5).

Step 3. Removing the Old Battery

Remove the old battery by de-soldering the two posts that hold the battery from the back side. Use solder-removal braid or a similar tool to remove the molten solder as you melt it (see Photo 6). Take care not to leave the soldering iron on the surface of the board for too long, or you may damage the etchings. After clearing the solder away from the posts that hold the battery, pull the posts through the holes in the board. You may have to get hold of a post with a pair of needle-nose pliers and pull it out as you melt the last bit of solder holding the post down. Do not grab the posts or the battery with your fingers until they have cooled down—the soldering iron makes them very hot.

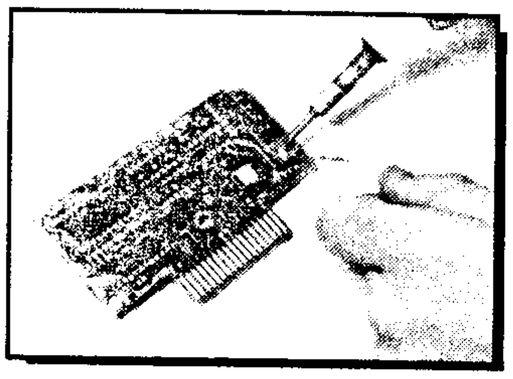


PHOTO 6

Step 4. Installing the New Battery

Cut two pieces of wire—one approximately 1-inch long, and another approximately 3/4 of an inch long. Strip about 1/8 of an inch of insulation from the ends of both wires. Solder one end of the 1-inch wire to the positive side of the battery (see Photo 7). Next solder one end of the 3/4-inch wire to the negative side of the battery. Bend the longer (1-inch) wire over the edge of the battery and insert it into the hole in the circuit board closest to the edge such that only the bare wire is sticking out the other side. The battery should have the positive side facing up when you're looking at the component side of the circuit board (see Photo 8). Bend the wire over that sticks out of the bottom side of the board so it lies flat (needle-nose pliers help here). Solder this lead to the under side of the circuit board (see Photo 9). The solder connection should be smooth and shiny. A bad solder connection may result in intermittent power from the battery and should be re-done. Stick the short wire (3/4-inch) through the other battery lead hole in the circuit board and solder it in place using the same technique as described above.

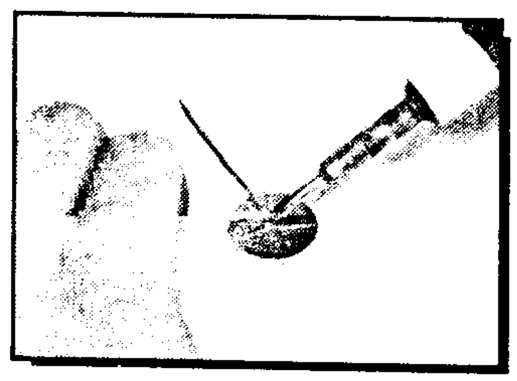


PHOTO 7

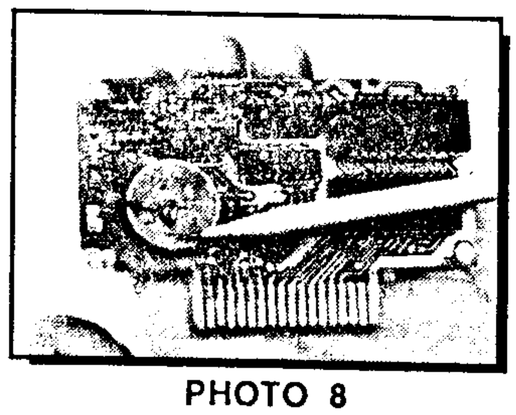


PHOTO 8

Step 5. Installing the Circuit Board

Install the circuit board following the instructions used to remove it (Step 2) in reverse order.

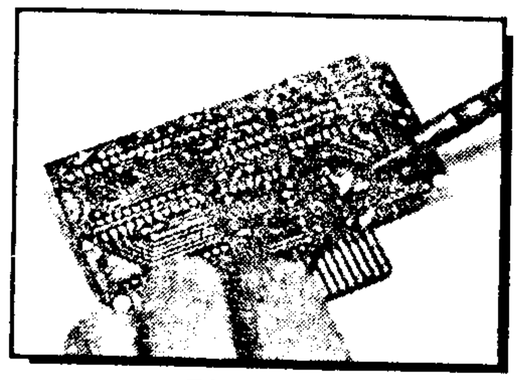


PHOTO 9

Step 6. Closing the Cartridge

Close the cartridge, aligning the dust plate and circuit board as you close the two halves. Snap the mouth of the cartridge closed (see Photo 10). Install the Phillips-head screw, and the cartridge is ready.

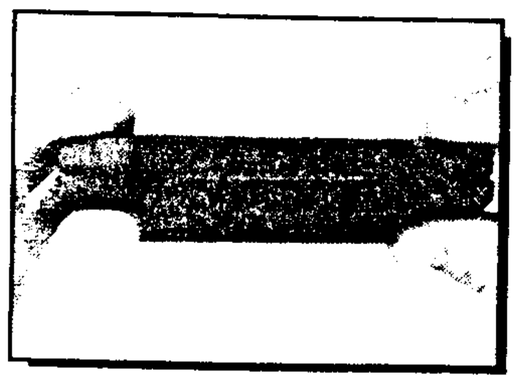


PHOTO 10

WARNING: Please do not attempt this procedure unless you are proficient at soldering electronic components. Improper techniques could result in damage to the circuit board or components. Home Computing Journal, its publisher, and the author assume no liability for unsuccessful project completion or damage to any of your equipment. These instructions are offered as is and readers should proceed at their own risk.

HCJournal™ DIRECTOR

About the HCJ Director and CodeWorks

The directories and instructions on these two pages should help you locate any program file on your HCJ diskette and give you the necessary information to successfully RUN any program. Each system has its own specially designed HCJ Director program that allows menu-driven access to all the HCJ programs in this Volume. If you see a version on your disk written in compiled BASIC, it means that the BASIC code has been processed through a compiler program which greatly enhances the speed of the slower program operations.

Each disk also contains a CodeWorks program that describes which sections of code accomplish specific tasks in the program. Full operating instructions are included in each CodeWorks program explaining how to either view the information on screen, or dump it to your system's printer.

HCJ

Apple II Family

Procedures For Loading Apple II Computers

1. Place the HCJ diskette in your disk drive. If you have more than one disk drive, make sure you insert the diskette in drive 1.
2. Turn on the Apple computer. The HCJ Startup menu will appear, asking if you wish to use the HCJ Director program, go to Applesoft BASIC, or format a ProDOS diskette. (This utility is included for those who wish to use separate data disks with the HCJ programs.)
3. Select HCJ Director from the menu.
4. A list of the programs included on the HCJ diskette will appear on the screen. To **RUN** one of these programs, enter the number corresponding to the program that you wish to **RUN** and press **[RETURN]**.
6. Make sure that the HCJ diskette is still in drive number 1 and press **[RETURN]** again. The selected program will **LOAD** and **RUN** for you. You may press **[ESC]** before pressing **[RETURN]** for the second time, if you change your mind.

Program Name	File Name	Language
1. Startup	STARTUP*	Applesoft BASIC
2. CodeWorks	CODEWORK	Applesoft BASIC
3. Digital Dumpster	DIGDUMP	Applesoft BASIC
4. Designer Genes	GENES	Applesoft BASIC
	GENES.OBJ**	Compiled BASIC
	GENES.DRV	Applesoft BASIC
5. Apple Tech Note (Memory Test)	RAMRX	Applesoft BASIC
6. Enhanced Apple II Sound (Sound Routine Demo)	SOUNDDEMO	Applesoft BASIC
7. WordWeave	WEAVE	Applesoft BASIC

*ProDOS, and ProDOS Utilities, Copyright © 1983-86 Apple Computer, Inc.
** Portions of the Compiled version are Copyrighted Microsoft Corp., 1981,1982,1983 Note: Because the Microsoft Compiler does not allow ProDOS disk access, the compiled version does not support disk options.
Note: The Apple II Family HCJ diskette contains portions of the Apple ProDOS operating system that allow the diskette to be used by itself. It is not necessary for you to have any other operating system to use the HCJ diskette.

Atari 800, 800XL, 130XE

Procedures For Loading Atari Computers

1. Place your DOS 2.5 systems disk in the disk drive 1.
2. Turn on the Atari computer.
3. After **READY** appears in the upper-left corner of the computer screen, insert the HCJ diskette into drive 1.
4. Type **LOAD "D:HCJDIR"** and press **[RETURN]**. After the loading procedure is complete and the cursor returns, type **RUN** and press **[RETURN]**.
5. A list of the programs included on the HCJ diskette will appear on the screen. To **RUN** one of these programs, enter the number corresponding to the program that you wish to **RUN** and press **[RETURN]**.
6. The selected program will **LOAD** and **RUN** for you.

Program Name	File Name	Language
1. CodeWorks	CODEWORK	Atari BASIC
2. Atari Tech Note (DATA Generator)	DATAGEN	Atari BASIC
3. Digital Dumpster	DIGDUMP	Atari BASIC
4. Designer Genes	GENES	Atari BASIC
5. HCJ Director	HCJDIR	Atari BASIC
6. Atari Plotting: Painless Perspective (Ripples)	RIPPLES	Atari BASIC
7. WordWeave	WEAVE	Atari BASIC

Note: Because some programs reconfigure your computer's memory, it is recommended that you press the **[RESET]** key on your computer if you experience problems running a new program.

Commodore 64

Procedures For Loading The C-64

1. Turn on your monitor, printer, disk drive, and any other peripheral connected to your computer.
2. Turn on the C-64 computer.
3. Place the HCJ diskette into the Commodore disk drive known as device number 8.
4. Type **LOAD "HCJDIR",8** and press **[RETURN]**. After the loading procedure is complete and the cursor returns, type **RUN** and press **[RETURN]**.
5. A list of the programs included on the HCJ diskette will appear on the screen. To **RUN** one of these programs, enter the number corresponding to the program that you wish to **RUN** and press **[RETURN]**.
6. Make sure that the HCJ diskette is still in drive number 8 and press **[RETURN]** again. The selected program will **LOAD** and **RUN** for you. You may press ← before pressing **[RETURN]** for the second time, if you change your mind.

Program Name	File Name	Language
1. CodeWorks	CODEWORK	C-64 BASIC
2. Digital Dumpster	DIGDUMP	C-64 BASIC
3. Designer Genes	GENES	C-64 BASIC
	GENES.B64*	Compiled BASIC
4. HCJ Director	HCJDIR	C-64 BASIC
5. C-64 BASID Synthesizer	SYNTH	C-64 BASIC
	BASID	C-64 BASIC
6. WordWeave	WEAVE	C-64 BASIC

*Program compiled using the Abacus Software BASIC 64 compiler.
Note: Because some programs reconfigure your computer's memory, it is recommended that you restart your computer if you experience problems running a new program.

TI-99/4A

Procedures for Loading The TI-99/4A with Extended BASIC

1. Ensure the Peripheral Box is properly connected to the console. Turn on the Peripheral Box.
2. Place the Extended Basic module securely in the machine.
3. Turn on the TI-99/4A computer.
4. Insert the HCJ diskette into drive 1.
5. Strike any key to bring up the first menu, then select Extended BASIC, and The HCJ Director program will automatically RUN.
6. Select the number of the program you wish to use, then press **[ENTER]** and the program will load and RUN automatically.

Procedures for Loading The TI-99/4A with TI BASIC

1. Ensure the Peripheral Box is properly connected to the console. Turn on the Peripheral Box.
2. Turn on the TI-99/4A computer and insert the HCJ diskette in drive 1.
3. Strike any key to bring up the first menu, then select BASIC.
4. To load the BASIC program you wish to use, type **OLD DSK1.***file name* where *file name* is the file name of the program. For example, if you wish to use Digital Dumpster type **OLD DSK1.DIGDUMP** and press **[ENTER]**.

Program Name	File Name	Language
1. Codeworks	CODEWORK	BASIC or Extended BASIC
2. Digital Dumpster	DIGDUMP	BASIC or Extended BASIC
	DIGDUMPX	Extended BASIC
3. Designer Genes	GENES	BASIC or Extended BASIC
	GENESX*	Extended BASIC
4. HCJ Director	LOAD	Extended BASIC
5. TI Flashlight	FLASHLIGHT	Extended BASIC
6. WordWeave	WEAVE	Extended BASIC

*Requires 32K Memory expansion.

CONTINUED

(1536-1727), dummy placeholders (ATASCII 128=INVERSE CONTROL ,) are inserted to fill gaps in the keyboard code sequence, and dummy ATASCII values (148=INVERSE CONTROL T) are given where no ATASCII equivalent of a keyboard code exists (e.g., CONTROL CAPS/LOWR, CONTROL RETURN, CONTROL with the numbers 0-9).

Preston Clark
Spencer, WV

Your method works fine, Preston, but there is a shorter solution—one that identifies the machine the program is running on and accesses the proper ROM locations accordingly. To get the program running on any Atari, including the older 800 models, replace the existing lines listed below:

```
230 GOSUB 1680 : GOSUB 1690 :
GOSUB 1700:GOSUB 1710 :
KEYTAB=64337:IF PEEK(65537)=255
THEN KEYTAB=65278
470 KEY=0 : A=PEEK(53789):
B=PEEK(53775) : IF B=251 OR B=243
THEN KEY=PEEK(A+KEYTAB) :
IF KEY=27 THEN GOSUB 1450:
GOTO 350
590  KEY=PEEK(A+KEYTAB) :
IF KEY=27 THEN GOSUB 1450 :
GOTO 350
840 A=PEEK(764) :
KEY=PEEK(A+KEYTAB) : IF KEY=27
THEN GOSUB 1450 : GOTO 770
```

Looking for AI

Dear Sir:

I have been reading about some new artificial intelligence software such as Turbo Prolog and Expert-Ease for the IBM PC, and XPER for the Commodore 64. It's an exciting prospect to run AI software on a micro, but just how good are these products? Also, although I have a read about this new field,I still don't really understand how it differs from the software I already run on my computer. The general articles on this subject seem too general; and the detailed explanations too detailed. Do you have any recommendations for getting some meaningful hands-on experience in artificial intelligence without getting burned in the wallet?

Ed Wynne
St. Louis, MO

Ed, your questions are indeed timely. Although research and development in artificial intelligence (AI) has been going on for years, only recently has this technology been brought to microcomputers. Needless to say, the many software products that are now available range in actual usefulness from the truly exciting Turbo Prolog from Borland International to products that are simply taking advantage of this latest computer "buzz-word." In upcoming issues of the Journal we will try to help you separate the wheat from the chaff.

Incidently, the way AI programs are created and operate is very different from other types of software you run on your personal computer—including the "intelligent" software we publish in this Journal. Virtually all non-AI programs are structured around algorithmic programming techniques (and suitable implementation languages), and are based on what we call "propositional logic." For number crunching applications and general text manipulations this approach is quite adequate. But when a problem cannot be reduced to a simple matter of finding the correct formula or the right algorithm, AI techniques and languages—structured around "heuristic" rules of thumb, and based on "predicate logic"—come into play. It is in these cases that we need to build an "expert system" to evaluate many different possible answers. Ideally, the system can then "learn" from new observations and facts fed into it.

We have already implemented a development project to bring the exciting concepts of AI to our readers. Watch for it in an upcoming Volume. HCJ