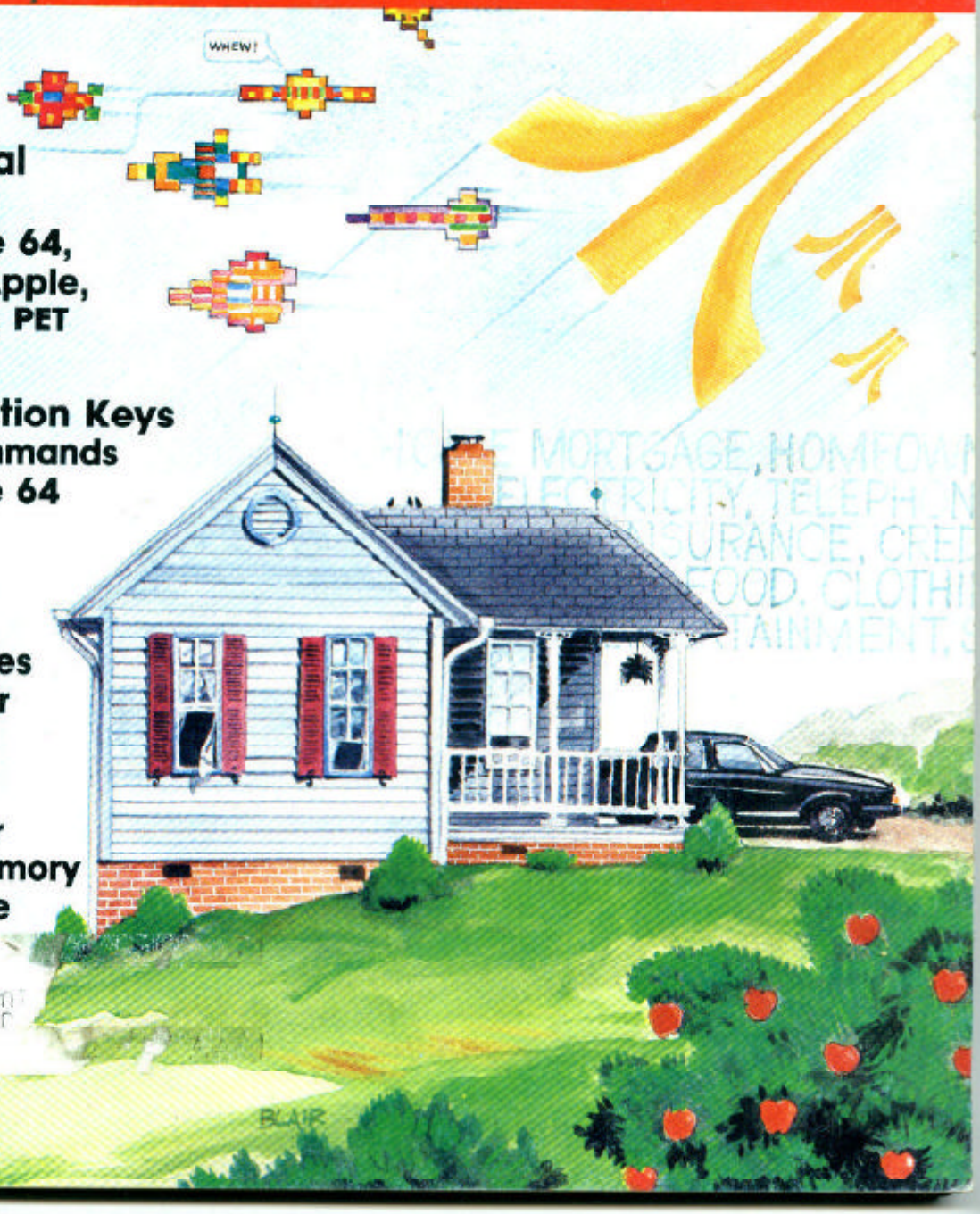
Space Dodger For Commodore 64, VIC, Atari, Apple, And TI

# COMPUTE!

$2.95
May
1985
Issue 60
Vol. 7, No. 5
$3.50 Canada
02193
ISSN 0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing

## SpeedScript 3.0
Our Enhanced Word Processor
Program Inside For Atari




**Home Financial Calculator**
For Commodore 64, VIC-20, Atari, Apple, IBM PC, PCjr, TI, PET

**Dynamic Function Keys**
One-Touch Commands For Commodore 64 & VIC-20

**IBM Disk Rx**
Recover Lost Files On Your PC/PCjr

**Apple IIc**
RAM Disk Mover
How To Turn Memory Into A Disk Drive



0 71486 02193

May 1985 Vol. 7. No. 5

## FEATURES



## REVIEWS



## COLUMNS AND DEPARTMENTS



## THE JOURNAL



**NOTE: See page 117 before typing in programs.**

**TOLL FREE Subscription Order Line**
**800-334-0868 (in NC 919-275-9809)**

**GUIDE TO ARTICLES AND PROGRAMS**

**AP** Apple, **Mac** Macintosh, **AT** Atari, **V** VIC-20, **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16, **P** PET/CBM, **TI** Texas Instruments, **PC** IBM PC, **PCjr** IBM PCjr, **CC** Radio Shack Color Computer. *General interest.


**COMPUTE!** Publications, Inc.
**One of the ABC Publishing Companies:**
**ABC Publishing, President, Robert G. Burton**
1330 Avenue of the Americas, New York, New York 10019
**Address all inquiries to:**
P.O. Box 5406, Greensboro, NC 27403

**COMPUTE! The Journal for Progressive Computing** (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. Send subscription orders or change of address (P.O. form 3579) to **COMPUTE!** Magazine, P.O. Box 914, Farmingdale, NY 11737. Second class postage paid at Greensboro, NC 27403 and additional mailing offices.


ISSN 0194-357X.

*Run the program and press one of the special keys. The corresponding value is printed. Notice that Ctrl, Alt, and the Shift keys are active only while they are pressed. But the Lock keys act as a toggle; pressing them once activates them, and pressing them again turns them off. If you press more than one key, their values are added. For example, holding down Ctrl and Alt displays 4 + 8 = 12.*

*To check for the Caps Lock key, you need to read bit 6. To test a particular bit, you must AND with the bit's value. Bit 6 can be checked by ANDing with 64. Change line 20 as follows:*

```
20 PRINT PEEK(&H17) AND 64
```

*Now the program will check only for Caps Lock. A 64 is displayed when Caps Lock is pressed, and a 0 is displayed when it's not, regardless of the status of the other keys.*

## Commodore Chained Programs

The "64 Paintbox Loader" on page 128 of the December 1984 issue of COMPUTE! is simple and clean, but it seems to be backwards. How does it work?

J. Quinn

*This is an example of a* chained program—*a program which loads and runs another program. Chaining programs with Commodore BASIC isn't too difficult, but it does involve a few tricks.*

*When you use the LOAD command from direct mode, the loaded program goes into memory without running. But if you use the LOAD command inside a program, whatever BASIC program (if any) is in memory after the loading is complete will run automatically. If the loaded program was BASIC, then that new program will begin executing. However, if a machine language program was loaded (with a final ,1 added to the LOAD command), then the BASIC program which requested the LOAD will start again from the beginning. This explains the peculiar construction of the 64 Paintbox Loader.*

*Something unexpected would happen if you used a seemingly more logical construction like this:*

```
10 LOAD "MLGAME",8,1
20 SYS 49152
```

*When this loader program runs, the machine language program MLGAME loads into its proper location, but then the computer tries to restart the BASIC program currently in memory, which is still the loader program. So it loads the MLGAME program again (and again and again and again). The loader program never reaches line 20.*

*Since the variables established by the running BASIC program are kept intact while the new program is loading (unless overwritten by the program being loaded), you can make a small change:*

```
10 IF L=0 THEN L=1:LOAD
   "MLGAME",8,1
20 SYS 49152
```

*When the loader (or any other BASIC program) is first run, all variables are erased, so L equals 0 and the game is loaded. After the LOAD, the program starts again from the beginning, but with variable values retained, so this time L is 1 and the program skips to line 20, which activates the ML program.*

*It is also possible to load one BASIC program from another. With careful planning, you can even run programs that are too large to fit into memory by breaking them into smaller parts and loading each part from the preceding portion. Since BASIC programs always load into the beginning of memory, the second program will overwrite the first. Variables may be erased, depending on how long the programs are. If the original program is larger, all numeric variables will be available for use in the second program.*

*String variables are passed to the second program only if they are dynamic. (Dynamic strings are those that involve some type of operation beyond simple string definition.) To be sure they make it, add a null string to the end of each string variable. Instead of A$="HELLO", use A$="HELLO" + "" to force the computer to store the string in high memory.*

*If the second program is larger, all variables will be lost when it is called by the first, so you must always pay close attention to program length when chaining BASIC programs.*

## Help For Adventurers

I am in need of assistance with the adventure game *Deadline* by Infocom. Do you know where I could write for help?

Rita Miller

*You can try writing to Infocom about any Infocom games. You might also want to contact Shay Addams, publisher and editor of* Questbusters, The Adventure Newsletter, at The Addams Expedition, 202 Elgin Court, Wayne, PA 19087. *Also, you might try writing* Wizards "R" Us, 308 Arrowood, Lake Jackson, TX 77566, *a new club dealing with games.*

## Analog Vs. Directional Joysticks

I was recently dismayed to find that I cannot connect the Wico controllers from our Atari 2600 to our IBM PCjr. The local computer store advised that I need analog controllers. I am confused. What is the difference between controllers, other than planned obsolescence?

David A. Baxter

*It's not planned obsolescence, just two different ways of designing a joystick controller. The joysticks used on the Atari 2600, Atari home computers, Commodore home computers, and Coleco Adam are* directional. *When the stick is deflected, one or two switches are closed, and the joystick returns a value to the computer which corresponds to one of eight directions (up, down, left, right, and the four diagonals). Computer programs check this value to determine the stick's direction, and then move a marker or player accordingly.*

*There is another way of designing a joystick which has been used with the IBM PC, PCjr, Apple II series, and TRS-80 Color Computer. These are analog joysticks. When the stick is deflected, they return a value which corresponds not only to the direction, but also to how far the stick was moved. They are more like paddle controllers on Atari and Commodore machines, with one paddle for the horizontal axis, and one for the vertical. (In fact, you could build an analog joystick for an Atari or Commodore by combining two paddle controllers.)*

*When you move a paddle controller from left to right, it returns a number, say from 0 to 255. That means there are 256 possible horizontal positions. The same type of value is used for the vertical axis. So analog joysticks tell the computer an absolute position—to which spot the joystick is pointing—instead of in which direction the joystick is pointing.*

*Which joystick is best depends upon the application. For a game requiring simple directional information—such as* Pac-Man—*directional sticks are superior, because the action is more positive. On the other hand, analog sticks are preferable for games in which you want to rapidly move an object to a new position on the screen without moving across all the intervening positions (for example, the aiming crosshair in* Missile Command, *although most versions of this game use directional joysticks or trackballs).*

*Because analog joysticks are a little more complicated to manufacture, they cost more. Another drawback is their thumbwheels for adjusting the range of values returned. There are usually two thumbwheels somewhere on the joystick, one for adjusting the vertical values and another for the horizontal values. If either thumbwheel is out of adjustment, the joystick can return wild values that the program can't interpret. A perfectly healthy program can crash with an "Illegal function call" or similar error message, and you might never suspect that it's the fault of the joystick. Some programs circumvent this problem by including routines for calibrating the joystick.*

## Texas Instruments "Cheater"

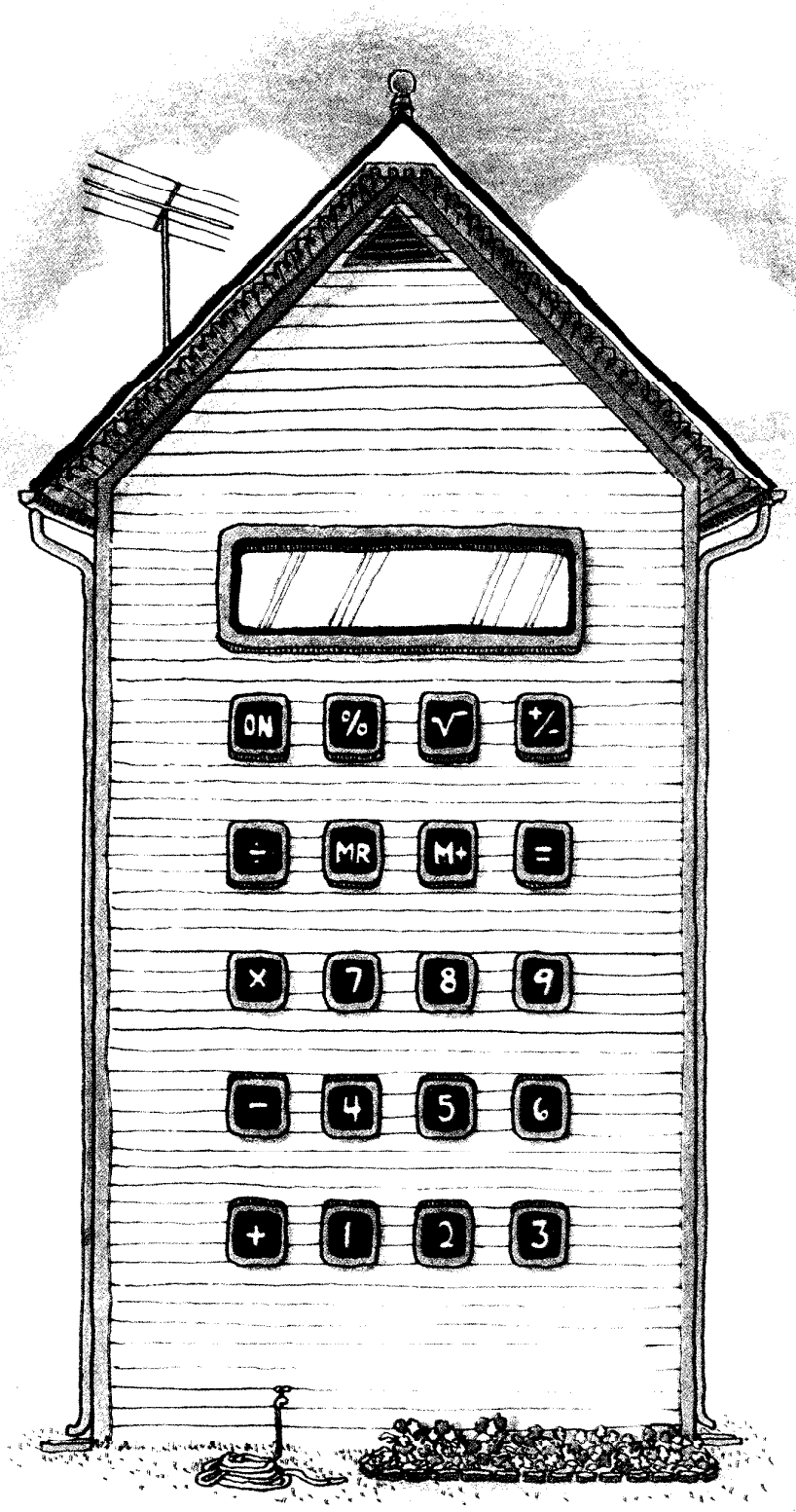
In some of the games for the TI-99/4A, you can change the number of lives or the starting level by following this procedure. Insert the cartridge, and turn on your computer. When the title screen appears, after selecting the game, quickly type "*#*".

David L. Whitlock

# Home Financial Calculator

Patrick Parrish, Programming Supervisor

*Many home budget programs have been published in magazines, but rarely has there been a program integrating as wide a variety of loan and investment calculations as "Home Financial Calculator." It is versatile, easy to use, and flexible. Rapid recalculation features make it an ideal tool for "what if" projections. A calculator mode with memory lets you solve problems not directly supported by the program, and you can pass values generated by one calculation to another. It works on the Commodore 64; VIC-20 (with at least 8K memory expansion); Commodore Plus/4 and 16 (using the 64 version); Commodore PET; Atari 400/800 (with at least 16K for tape and 24K for disk) and XL/XE models; Apple II series; IBM PC and PCjr; and TI-99/4A (regular BASIC) Though not tested on other computers the program is written generally enough to run with trivial modifications on any computer with Microsoft BASIC.*




Investment and loan calculations are readily computerized. In fact, many programs have been written which perform these tasks individually. "Home Financial Calculator" goes a step further by integrating several common financial calculations in a menu-driven package. It also features a calculator mode or scratch pad area where program variables can be manipulated using common mathematical operations.

Program 1 is a general BASIC program that runs without modification on Apple II-series computers, and also on a number of other machines with minor changes No matter what computer you have, type in Program 1. For computers other than the Commodore models you should type a caret (^) for the character shown as an up-arrow (↑). Then add the appropriate lines for your computer from Programs 2–7. As always, save the program before running it for the first time.

*Important: Because Program 1 is a general listing for several different*

*computers, it has no checksum numbers for use with the "Automatic Proofreader." Be extra careful when typing this program, especially the long lines which contain the financial formulas. A mistyped program may still run, but the results it gives could be inaccurate.*

When you run the program, a main menu offers you a choice of Investment or Loan calculations. Type I or L to reach the appropriate submenu.

## Common Variables

Before looking at any calculations, let's consider some basics of the program. Home Financial Calculator uses some parameters or variables repeatedly in the calculations. These variables are *Total* (also referred to as Future Value, Total Owed, etc., depending on the calculation); *Present Value* (principal); *Interest Rate*; *Years*; *Months*; *Number of Periods* (of either compounding, deposits, withdrawals, or payments, depending on the application); *Deposits*; and *Withdrawals*. When in the calculator mode (explained below), you'll reference these eight variables with the single letters T, P, I, Y, M, N, D, and W.

As you work with Home Financial Calculator, the values of the eight variables are preserved until you change them. Whenever the program asks you for an input (for example, Interest), the current value of that variable is displayed (zero if no value has been entered yet). If you want to keep the current value, just press RETURN (or ENTER, depending on your keyboard). Otherwise, enter the new value and press RETURN.

With this feature, Home Financial Calculator makes it easy for you to generate "what if" projections. Simply run the same calculation repeatedly, each time changing a previously entered value. Press RETURN to keep a value, and change only one or two values to see the effect on the final result.

You can also store the current value into the calculator mode's Memory Register or recall a value from the Memory Register. To see how all this works, let's take a look at some calculations possible with Home Financial Calculator.

## Investment Calculations

Here is the Investment submenu that appears when you type I from the main menu:

**1) Future Value with Periodic Interest**
**2) Future Value with Interest Compounded Continuously**
**3) Future Value with Regular Deposits**
**4) Future Value with Cash Flows**
**5) Withdrawal of Funds**
**6) Net Present Value**
**7) Calculator Mode**
**8) Return to Main Menu.**

Determine which option you want and press the appropriate key.

Each option displays screen prompts which ask you to input several values. These values are stored in the eight variables mentioned above: *T* for Total (Future Value), *P* for Present Value (principal), *I* for Interest Rate, *Y* for Years, *M* for Months, *N* for Number of Periods, *D* for Deposits, and *W* for Withdrawals. Of course, not all calculations require you to enter all these values, while others may ask for additional information.

Most calculations can be solved for any *one* of the variables. To solve for a variable, enter an uppercase X at the corresponding input prompt. For example, you could enter values for everything except the Interest Rate, typing X at the Interest Rate prompt. Home Financial Calculator then solves for the Interest Rate.

Remember, however, that the program can solve for only *one* variable during each calculation. If you enter an X at more than one prompt, the program does not have enough information to calculate an answer. Keep this in mind, because the program does not check for potential conflicts.

## Future Value With Periodic Interest

Home Financial Calculator's options are fairly self-explanatory when you run the program, but let's try an example. We'll calculate the future value of an investment drawing periodic interest. This kind of investment could be a savings account, interest-bearing checking account, bonds, or a money market account. Choose this option by entering 1 at the Investment submenu.

After the screen clears, the program asks for the first input—Future Value, which appears with an asterisk (*). Below this is a zero (the current value of this variable in memory; all variables start out with a value of zero). Following this is an input prompt.

The asterisk preceding Future Value means that this is one of the variables you can solve for. (A variable *not* preceded by an asterisk means that variable *cannot* be solved for in that particular calculation, so X would be an illegal response.) If you'd like to calculate the Future Value, enter an X here, and answer all the other prompts with the appropriate values.

Let's calculate the future value of a $1,000 investment drawing 8 percent interest for two years and three months, with four compounding periods each year. Enter an X for *Future Value*, since we'll be solving for this total. Answer *Present Value* with 1000 (the principal you're investing); *Annual Int Rate (%)* with 8 (enter the percentage, not a fraction); *For # Of Years* with 2; *For # Of Months* with 3, and *# Of Periods (Compounding)* with 4. After you enter the last value, Home Financial Calculator figures the *Total Future Value* and displays the answer—$1195.09.

Now suppose you wish to know the future value of the same $1,000 investment if you make 9 percent interest. Choose option 1 on the Investment submenu again and rerun the calculation. Notice how Home Financial Calculator automatically prints the current value of each variable at each prompt. The *Future Value* prompt shows a current value of 1195.09 from the previous calculation. Type an X at this prompt, 9 for Interest Rate, and RETURN at all other prompts to preserve their values. The result should be $1221.71.

The versatility of Home Financial Calculator becomes apparent when you realize how many different ways you can run this calculation. Using this same menu option, you can calculate the initial investment (or present value) necessary to accrue a certain future value with periodic interest; the interest rate necessary to accrue a future value from a present value; or the time (in years and months) it would take to accumulate a future amount from an initial investment with periodic interest payments. Just enter an X for the unknown value you're seeking, and fill in all the other prompts.

## Future Value With Interest Compounded Continuously

Option 2, a variation of option 1, handles investments paying a con-

tinuous interest rate. Like option 1, option 2 can handle a number of calculations—just place an X in the slot you'd like to solve for.

Here, after entering all other parameters, you can calculate the future value of an investment; the initial investment required to reach a certain future value; the interest required to reach a desired future value; or the time required to reach a certain future value at a specified interest rate.

Notice that any variables used in option 1 will be displayed with their current values when running option 2. As mentioned above, the eight major variables in Home Financial Calculator retain their values throughout the program until you change them. This feature is convenient when going from one option to another on the Investment or Loan submenus.

In addition, the values are preserved for use in the calculator mode. For instance, you could compare the effect of continuously compounded interest to periodic interest (option 1) without having to retype the input.

## Future Value With Regular Deposits

If you're interested in setting up an annuity, you'd choose option 3 on the Investment submenu. You can determine the future value of an account (such as a savings account, Individual Retirement Account, college or vacation fund, etc.) with regular deposits where interest is compounded with each deposit.

Option 3 can also tell you the amount of each deposit necessary to accrue a future value; the interest rate needed to provide some future value with regular deposits; or the time it would take to amass a future value with regular deposits.

## Future Value With Cash Flows

Option 4 does a single calculation—it always solves for *Future Value*, so don't enter an X anywhere. It calculates the future value of an investment with yearly cash flows (either positive or negative). The *Annual Interest Rate* you input here is the growth rate on the money you've invested.

As an example, suppose you wish to determine the value of a vacation fund collected over four years. You're asked for the number of years, then for the deposit or withdrawal each year. You deposit $500 in the fund the first year and $200 the second. The third year you are forced to withdraw $300 (entered as −300), and the fourth year, you put in $400. The fund has a growth rate of 12 percent. Its value after four years will be $1,017.34.

A future value determination can also tell you whether an investment is worthwhile. If the future value of all cash flows is positive or zero, the investment is profitable. A negative future value, on the other hand, represents a losing investment.

## Withdrawal Of Funds

If you intend to open an account from which you can regularly withdraw funds, choose option 5. With this option, you can determine the initial deposit required in the account to cover your withdrawals; the amount you can withdraw regularly from this account; the rate of interest you must make on funds in the account; or the period of time over which you can make withdrawals.

## Net Present Value

Option 6 lets you determine the feasibility of a prospective investment by calculating its net present value. Net present value is the current value of all future yearly cash flows to an investment along with any initial cash requirement. The interest rate you input here is the rate of return you require on your investment. A positive net present value indicates a profitable investment, while a negative result signifies a losing investment.

As an example, suppose you have the opportunity to make a $2,000 investment which would return $1,500 the first year, cost you $750 the second year, and return $1,900 the third year. You hope to make 13 percent on your money. With option 6, you determine a net present value of $56.87, representing a profitable investment.

## The Calculator Mode

Option 7 puts you in the calculator mode (also available from the Loan submenu). Calculator mode works very much like a handheld calculator with a single memory. You can type in a value or recall one from a variable by entering its symbol—T(otal), P(resent Value), I(nterest Rate), Y(ears), M(onths), N(umber of Periods), D(eposits), and W(ithdrawals). You can perform simple math on values stored in the Memory Register using reverse Polish notation. And you can use the results in future calculations.

When you enter calculator mode, the calculator command line appears on the screen:

```
V S H R M+ M− M* M/ MR MC
MEM=0
```

Here are the commands:

| | |
|---|---|
| **V** | (View the values of the eight primary variables) |
| **S** | (Store Memory Register into a variable) |
| **H** | (Help prints the command line) |
| **R** | (Return to main menu, exit calculator mode) |
| **M+** | (Add the last input to the Memory Register) |
| **M−** | (Subtract the last input from the value in the Memory Register, and store the result in the Register) |
| **M*** | (Multiply the last input times the value in the Memory Register, and store the result in the Register) |
| **M/** | (Divide the last input into the value in the Memory Register, and store the result in the Register) |
| **MR** | (Memory Recall) |
| **MC** | (Memory Clear to zero) |
| **MEM=** | (Memory Register's current value) |

If you've run through a sample investment calculation, you now have some variables in memory. Enter V in the calculator mode to see them. The screen displays the eight values currently in memory for the eight variables.

To work with one of these variables, enter one of their letters (T, P, I, Y, M, N, D, or W) and press RETURN. Then type M+ to add it to the Memory Register (all variables must be stored in the Register before you can perform any operations on them). Suppose you put the current value for T into the Register and now wish to add $229 to this value. Enter 229, press RETURN, then type M+ and press RETURN. The addition is performed and the result displayed. To store this value back into the T variable, enter S for Store. A prompt appears, requesting the variable in which you intend to store the value. Type T to store the value into the variable T.

You can also use the Memory Register to hold a value not represented by any of the eight variables. To do this, determine a value using the calculator mode and store it into the Memory Register with M+. Then, when you're running a calculation elsewhere in the program, you can substitute this value for any of the eight primary variables by typing MR (Memory Recall) at the appropriate prompt. MR can be used both in the calculator mode and at any prompt where the previous value is displayed.

Finally, option 8 on the Investment submenu returns you to the main menu. Once there, you can perform some loan calculations by typing L.

## Loan Calculations

Here is the Loan calculations submenu:

**1) Regular Loan Payments**
**2) Remaining Loan Liability**
**3) Final Loan Payment**
**4) Single Payment Loan**
**5) Loan Amortization Schedule**
**6) Calculator Mode**
**7) Return to Main Menu**

## Regular Loan Payments

Option 1 handles a number of calculations for equal payment loans. You can figure the principal of a loan; the amount of each regular payment necessary to repay a loan; the annual interest rate on a loan with regular payments; or the term of the loan.

### Remaining Loan Liability

With option 2, you can determine the remaining balance on a loan with regular payments after a number of payments have been made. Enter the principal on the loan, the amount of each payment, the annual interest rate, the number of payments yearly, and the last payment number.

## Final Loan Payment

Option 3 calculates the amount of the final payment on a loan. In many cases, the last payment of a loan will vary from the amount of the regular payment. This option handles situations where the final payment is greater than ("balloon payments") or less than the regular payment.

## Single Payment Loan

Option 4 calculates the amount owed on a loan that is paid off with a single payment. You must input the principal on the loan, its annual interest rate, its term in years and months, and the number of times a year the interest on the principal is compounded.

## Loan Amortization Schedule

Option 5 displays a loan amortization schedule. Enter the principal on the loan, the amount of each payment, the annual interest rate, the term of the loan, and the number of payments yearly. Then enter the period of the year in which the loan began (for instance, 10 for October) and the range in years of the amortization schedule you'd like to examine.

Because of the complexity of these calculations, there may be a delay before the output appears on the screen, especially if you have chosen to look at the latter years in a long-term loan repayment schedule (such as a home mortgage). When the amortization table appears, it displays the payment number, the beginning balance for the period, the amount paid toward the loan principal, the amount paid in interest, and the ending balance. To keep the information from scrolling off the screen, the program shows only a few payment periods at a time. Press RETURN to view another screenful. When the end of a year is reached, the program gives the total amounts paid on the principal and in interest for the year. In addition, when the last period of the loan is reached, the program displays the final payment for the loan.

The last two options on the Loan submenu are the same as those on the Investment submenu.

## Modifying The Program

Home Financial Calculator is written in a modular format for easy modification. For many routines, it uses common input labels (lines 4710–5080) and some output labels (lines 5090–5170). If you want to add an investment or loan calculation routine, choose the labels from these lines that fit your application.

Also, you may wish to add a printer option to the loan amortization schedule. Examine lines 3230–3940. Here, variable D5 (defined in line 150) determines the number of loan payments considered on each screen. Variables S1, S2, S3, and S4 (defined in lines 160–190) format the output horizontally on the screen.

## Program 1: Home Financial Calculator For Apple (General Version)

```
100 DIM V(8)
110 V$="TPIYMNDW"
120 C$="VSHR"
130 C1$="M+M-M*M/MRMC"
140 Q$=""
150 D5=13
160 S1=5
170 S2=15
180 S3=23
190 S4=31
200 GOSUB 5450
210 PRINT "INVESTMENTS OR LOAN
    S"
220 PRINT "(I/L) ";
230 INPUT A$
240 IF A$="I" THEN 270
250 IF A$="L" THEN 2170
260 GOTO 230
270 GOSUB 5450
280 PRINT "INVESTMENTS:"
290 PRINT
300 PRINT "1) FUTURE VALUE WIT
    H PERIODIC INTEREST"
310 PRINT "2) FUTURE VALUE WIT
    H INTEREST COMPOUNDED CONT
    INUOUSLY"
320 PRINT "3) FUTURE VALUE WIT
    H REGULAR DEPOSITS"
330 PRINT "4) FUTURE VALUE WIT
    H CASH FLOWS"
340 PRINT "5) WITHDRAWAL OF FU
    NDS"
350 PRINT "6) NET PRESENT VALU
    E"
360 PRINT "7) CALCULATOR MODE"
370 PRINT "8) RETURN TO MAIN M
    ENU"
380 PRINT
390 PRINT "CHOICE ";
400 INPUT A$
410 A=VAL(A$)
420 IF A<1 THEN 400
430 IF A>8 THEN 400
440 ON A GOTO 470,730,970,1360
    ,1550,1940,450,200
450 GOSUB 4180
460 GOTO 200
470 GOSUB 5450
480 PRINT "FUTURE VALUE WITH P
    ERIODIC INTEREST"
490 PRINT
500 GOSUB 4710
510 GOSUB 4750
520 PRINT "*";
530 GOSUB 4840
540 PRINT "*";
550 GOSUB 4880
560 IF E=4 THEN 580
570 GOSUB 4920
580 GOSUB 4970
590 IF E<>1 THEN 620
600 V(1)=INT(V(2)*(1+V(3)/V(6)
    )↑(V(6)*Y)*100+.5)/100
610 GOSUB 5090
620 IF E<>2 THEN 650
630 V(2)=INT(V(1)/((1+V(3)/V(6
    ))↑(V(6)*Y))*100+.5)/100
640 GOSUB 5120
650 IF E<>3 THEN 680
660 V(3)=INT((V(6)*(V(1)/V(2))
    ↑(1/(V(6)*Y))-V(6))*10000+
    .5)/10000
670 GOSUB 5150
680 IF E<>4 THEN 710
690 V(4)=LOG(V(1)/V(2))/(V(6)*
    LOG(1+V(3)/V(6)))
700 GOSUB 5180
710 GOSUB 5330
```

```
720 GOTO 270
730 GOSUB 5450
740 PRINT "FUTURE VALUE WITH I
    NTEREST COMPOUNDED CONTINU
    OUSLY"
750 PRINT
760 GOSUB 4710
770 GOSUB 4750
780 PRINT "*";
790 GOSUB 4840
800 PRINT "*";
810 GOSUB 4880
820 IF E=4 THEN 840
830 GOSUB 4920
840 IF E<>1 THEN 870
850 V(1)=INT(V(2)*EXP(V(3)*Y)*
    100+.5)/100
860 GOSUB 5090
870 IF E<>2 THEN 900
880 V(2)=INT(V(1)/EXP(V(3)*Y)*
    100+.5)/100
890 GOSUB 5120
900 IF E<>3 THEN 930
910 V(3)=INT(LOG(V(1)/V(2))/Y*
    10000+.5)/10000
920 GOSUB 5150
930 IF E<>4 THEN 710
940 V(4)=INT(LOG(V(1)/V(2))/V(
    3)*100+.5)/100
950 GOSUB 5180
960 GOTO 710
970 GOSUB 5450
980 PRINT "FUTURE VALUE WITH R
    EGULAR DEPOSITS"
990 PRINT
1000 GOSUB 4710
1010 PRINT "*REGULAR DEPOSIT $
    "
1020 C=6
1030 GOSUB 3950
1040 PRINT "*";
1050 GOSUB 4840
1060 PRINT "*";
1070 GOSUB 4880
1080 IF E=4 THEN 1100
1090 GOSUB 4920
1100 GOSUB 4970
1110 IF E<>1 THEN 1140
1120 V(1)=INT(V(7)*V(6)*((1+V(
    3)/V(6))↑(V(6)*Y)-1)/V(3)
    *100+.5)/100
1130 GOSUB 5090
1140 IF E<>3 THEN 1280
1150 V(3)=.99
1160 I=0
1170 T=INT(V(7)*(((1+V(3)/V(6)
    )↑(V(6)*Y)-1)/(V(3)/V(6))
    )*100+.5)/100
1180 TE=ABS(V(3)-I)/2
1190 I=V(3)
1200 IF ABS(T-V(1))<.005 THEN
    {SPACE}1260
1210 IF T<V(1) THEN 1240
1220 V(3)=V(3)-TE
1230 GOTO 1170
1240 V(3)=V(3)+TE
1250 GOTO 1170
1260 V(3)=INT(V(3)*10000+.5)/1
    0000
1270 GOSUB 5150
1280 IF E<>4 THEN 1310
1290 V(4)=LOG(V(3)*V(1)/(V(6)*
    V(7))+1)/(V(6)*LOG(1+V(3)
    /V(6)))
1300 GOSUB 5180
1310 IF E<>7 THEN 710
1320 V(7)=INT(V(1)*(V(3)/V(6))
    /((1+V(3)/V(6))↑(V(6)*Y)-
    1)*100+.5)/100
1330 PRINT
1340 PRINT "REGULAR DEPOSITS R
    EQUIRED:$";V(7)
1350 GOTO 710
1360 GOSUB 5450
1370 PRINT "FUTURE VALUE WITH
    {SPACE}CASH FLOWS"
1380 PRINT
1390 GOSUB 4840
1400 GOSUB 4880
1410 PRINT "CASH FLOW (+/-)"
1420 PRINT
1430 V(1)=0
1440 FOR I=1 TO V(4)
1450 PRINT "CASH FLOW - YEAR #
    ";I
1460 INPUT A$
1470 A=VAL(A$)
1480 V(1)=V(1)+A*(1+V(3))↑(V(4
    )-I)
1490 NEXT I
1500 V(1)=INT(V(1)*100+.5)/100
1510 GOSUB 5090
1520 TE=V(1)
1530 GOSUB 5270
1540 GOTO 710
1550 GOSUB 5450
1560 PRINT "WITHDRAWAL OF FUND
    S"
1570 PRINT
1580 GOSUB 4750
1590 PRINT "*REGULAR WITHDRAWA
    L $"
1600 C=7
1610 GOSUB 3950
1620 PRINT "*";
1630 GOSUB 4840
1640 PRINT "*";
1650 GOSUB 4880
1660 IF E=4 THEN 1680
1670 GOSUB 4920
1680 GOSUB 4970
1690 IF E<>2 THEN 1720
1700 V(2)=INT(V(8)*V(6)/V(3)*(
    1-(1+V(3)/V(6))↑(-V(6)*Y)
    )*100+.5)/100
1710 GOSUB 5120
1720 IF E<>3 THEN 1860
1730 V(3)=.99
1740 I=0
1750 R=INT(V(2)*V(3)/V(6)*(1/(
    (1+V(3)/V(6))↑(V(6)*Y)-1)
    +1)*100+.5)/100
1760 TE=ABS(V(3)-I)/2
1770 I=V(3)
1780 IF ABS(R-V(8))<.005 THEN
    {SPACE}1840
1790 IF R<V(8) THEN 1820
1800 V(3)=V(3)-TE
1810 GOTO 1750
1820 V(3)=V(3)+TE
1830 GOTO 1750
1840 V(3)=INT(V(3)*10000+.5)/1
    0000
1850 GOSUB 5150
1860 IF E<>4 THEN 1890
1870 V(4)=LOG(V(6)*V(8)/(V(6)*
    V(8)-V(3)*V(2)))/(V(6)*LO
    G(1+V(3)/V(6)))
1880 GOSUB 5180
1890 IF E<>8 THEN 710
1900 V(8)=INT(V(2)*V(3)/V(6)*(
    1/((1+V(3)/V(6))↑(V(6)*Y)
    -1)+1)*100+.5)/100
1910 PRINT
1920 PRINT "REGULAR WITHDRAWAL
    S:$";V(8)
1930 GOTO 710
1940 GOSUB 5450
1950 PRINT "NET PRESENT VALUE:
    $"
1960 PRINT
1970 PRINT "INITIAL INVESTMENT
    "
1980 C=1
1990 GOSUB 3950
2000 GOSUB 4840
2010 GOSUB 4880
2020 PRINT "CASH FLOW (+/-)"
2030 PRINT
2040 NV=-V(2)
2050 FOR I=1 TO V(4)
2060 PRINT "CASH FLOW - YEAR #
    ";I
2070 INPUT A$
2080 A=VAL(A$)
2090 NV=NV+A/((V(3)+1)↑I)
2100 NEXT I
2110 NV=INT(NV*100+.5)/100
2120 PRINT
2130 PRINT "NET PRESENT VALUE:
    $";NV
2140 TE=NV
2150 GOSUB 5270
2160 GOTO 710
2170 GOSUB 5450
2180 PRINT "LOANS:"
2190 PRINT
2200 PRINT "1) REGULAR LOAN PA
    YMENTS"
2210 PRINT "2) REMAINING LOAN
    {SPACE}LIABILITY"
2220 PRINT "3) FINAL LOAN PAYM
    ENT"
2230 PRINT "4) SINGLE PAYMENT
    {SPACE}LOAN"
2240 PRINT "5) LOAN AMORTIZATI
    ON SCHEDULE"
2250 PRINT "6) CALCULATOR MODE
    "
2260 PRINT "7) RETURN TO MAIN
    {SPACE}MENU"
2270 PRINT
2280 PRINT "CHOICE ";
2290 INPUT A$
2300 A=VAL(A$)
2310 IF A<1 THEN 2290
2320 IF A>7 THEN 2290
2330 ON A GOTO 2360,2780,2960,
    3120,3230,2340,200
2340 GOSUB 4180
2350 GOTO 200
2360 GOSUB 5450
2370 PRINT "REGULAR LOAN PAYME
    NTS"
2380 PRINT
2390 PRINT "*";
2400 GOSUB 4790
2410 PRINT "*";
2420 GOSUB 5010
2430 PRINT "*";
2440 GOSUB 4840
2450 PRINT "*";
2460 GOSUB 4880
2470 IF E=4 THEN 2490
2480 GOSUB 4920
2490 GOSUB 4970
2500 IF E<>2 THEN 2550
2510 V(2)=INT(V(7)*V(6)/V(3)*(
    1-(1+V(3)/V(6))↑(-V(6)*Y)
    )*100+.5)/100
2520 PRINT
2530 PRINT "AMT OF PRINCIPAL:$
    ";V(2)
2540 GOTO 2760
2550 IF E<>3 THEN 2690
2560 V(3)=.99
2570 I=0
2580 P=INT(V(7)*V(6)/V(3)*(1-(
    (1+V(3)/V(6))↑(-V(6)*Y)))
    *100+.5)/100
2590 TE=ABS(V(3)-I)/2
2600 I=V(3)
2610 IF ABS(P-V(2))<.005 THEN
    {SPACE}2670
2620 IF P<V(2) THEN 2650
2630 V(3)=V(3)+TE
```

```
2640 GOTO 2580
2650 V(3)=V(3)-TE
2660 GOTO 2580
2670 V(3)=INT(V(3)*10000+.5)/1
     0000
2680 GOSUB 5150
2690 IF E<>4 THEN 2720
2700 V(4)=-LOG(1-V(3)*V(2)/(V(
     6)*V(7)))/(V(6)*LOG(V(3)/
     V(6)+1))
2710 GOSUB 5180
2720 IF E<>7 THEN 2760
2730 V(7)=INT(V(3)*V(2)/(V(6)*
     (1-(V(3)/V(6)+1)↑(-V(6)*V
     ))*100+.5)/100
2740 PRINT
2750 PRINT "REQ PAYMENT:$";V(7
     )
2760 GOSUB 5330
2770 GOTO 2170
2780 GOSUB 5450
2790 PRINT "REMAINING LOAN LIA
     BILITY"
2800 PRINT
2810 GOSUB 4790
2820 GOSUB 5010
2830 GOSUB 4840
2840 GOSUB 4970
2850 PRINT "LAST PAYMENT # WAS
     :"
2860 INPUT A$
2870 A=VAL(A$)
2880 FOR J=1 TO A
2890 I=INT(P*V(3)/V(6)*100+.5)
     /100
2900 P=P+I-V(7)
2910 NEXT J
2920 LI=INT(P*100+.5)/100
2930 PRINT
2940 PRINT "LIABILITY AFTER ";
     A;" PAYMENTS:$";LI
2950 GOTO 2760
2960 GOSUB 5450
2970 PRINT "LAST LOAN PAYMENT"
2980 PRINT
2990 GOSUB 4790
3000 GOSUB 5010
3010 GOSUB 4840
3020 GOSUB 5050
3030 GOSUB 4970
3040 FOR J=1 TO V(6)*Y
3050 I=INT(P*V(3)/V(6)*100+.5)
     /100
3060 P=P+I-V(7)
3070 NEXT J
3080 LP=INT(P*100+.5)/100+V(7)
3090 PRINT
3100 PRINT "LAST PAYMENT:$";LP
3110 GOTO 2760
3120 GOSUB 5450
3130 PRINT "SINGLE PAYMENT LOA
     N"
3140 PRINT
3150 GOSUB 4790
3160 GOSUB 4840
3170 GOSUB 5050
3180 GOSUB 4970
3190 V(1)=INT(V(2)*(1+V(3)/V(6
     ))↑(Y*V(6))*100+.5)/100
3200 PRINT
3210 PRINT "TOTAL OWED:$";V(1)
3220 GOTO 2760
3230 C5=0
3240 N5=0
3250 F=0
3260 P1=0
3270 I1=0
3280 GOSUB 5450
3290 PRINT "LOAN AMORTIZATION
     {SPACE}SCHEDULE"
3300 PRINT
3310 GOSUB 4790
3320 GOSUB 5010
3330 GOSUB 4840
3340 GOSUB 5050
3350 PRINT "# OF PAYMENTS YEAR
     LY"
3360 GOSUB 3950
3370 PRINT "ENTER THE PERIOD O
     F THE YEAR IN WHICH THE L
     OAN BEGAN"
3380 INPUT N
3390 NE=N
3400 NP=(V(4)*12+V(5))/(12/V(6
     ))
3410 NY=INT(((N-1)+NP)/V(6)+.9
     9)
3420 PRINT "ENTER THE RANGE OF
      YEARS YOU'D LIKE TO EXAM
     INE (FIRST, LAST)"
3430 INPUT F1,L1
3440 IF L1<=NY THEN 3460
3450 L1=NY
3460 FOR J1=1 TO L1
3470 IF J1<F1 THEN 3490
3480 GOSUB 5390
3490 FOR J=1 TO V(6)-N+1
3500 I=INT(P*V(3)/V(6)*100+.5)
     /100
3510 N5=N5+1
3520 PP=V(7)-I
3530 IF J1<>NY THEN 3570
3540 IF N5<>NP THEN 3570
3550 PP=P
3560 F=1
3570 IF J1<F1 THEN 3600
3580 PRINT N5;TAB(S1);INT(P*10
     0+.5)/100;
3590 PRINT TAB(S2);INT(PP*100+
     .5)/100;Q$;TAB(S3);
3600 P=P+I-V(7)
3610 IF F=0 THEN 3640
3620 P=0
3630 J=V(6)
3640 IF J1<F1 THEN 3670
3650 PRINT I;TAB(S4);INT(P*100
     +.5)/100;
3660 PRINT
3670 I1=I1+I
3680 P1=P1+PP
3690 C5=C5+1
3700 IF C5<>D5 THEN 3770
3710 IF J1<F1 THEN 3770
3720 GOSUB 5330
3730 GOSUB 5450
3740 C5=0
3750 IF J=V(6)-N+1 THEN 3770
3760 GOSUB 5390
3770 NEXT J
3780 IF J1<F1 THEN 3890
3790 IF F=0 THEN 3820
3800 PRINT
3810 PRINT "FINAL PAYMENT :$";
     INT((PP+I)*100+.5)/100
3820 PRINT
3830 PRINT "TOTAL INT PAID IN
     {SPACE}YR ";J1;":$";INT(I
     1*100+.5)/100
3840 PRINT "TOTAL PRINC PAID I
     N YR ";J1;":$";INT(P1*100
     +.5)/100
3850 IF F=1 THEN 3930
3860 IF J1=L1 THEN 3930
3870 GOSUB 5330
3880 GOSUB 5450
3890 C5=0
3900 P1=0
3910 I1=0
3920 N=1
3930 NEXT J1
3940 GOTO 2760
3950 C=C+1
3960 IF C<>3 THEN 3990
3970 PRINT V(3)*100,
3980 GOTO 4000
3990 PRINT V(C),
4000 A$=""
4010 INPUT A$
4020 IF A$<>"" THEN 4040
4030 RETURN
4040 IF A$<>"MR" THEN 4100
4050 PRINT "MEM=";M;"
     {2 SPACES}USE AS VARIABLE
      HERE (Y/N)"
4060 INPUT A$
4070 IF A$="N" THEN 4000
4080 V(C)=M
4090 RETURN
4100 IF A$<>"X" THEN 4130
4110 E=C
4120 RETURN
4130 V(C)=VAL(A$)
4140 IF C<>3 THEN 4160
4150 V(C)=V(C)/100
4160 RETURN
4170 REM CALCULATOR MODE
4180 GOSUB 5450
4190 M5=0
4200 GOSUB 4530
4210 INPUT A$
4220 IF ASC(A$)>57 THEN 4250
4230 T=VAL(A$)
4240 GOTO 4210
4250 FOR I=1 TO 8
4260 IF A$<>MID$(V$,I,1) THEN
     {SPACE}4290
4270 PRINT V(I)
4280 T=V(I)
4290 NEXT I
4300 FOR J=1 TO 6
4310 IF A$<>MID$(C1$,(J-1)*2+1
     ,2) THEN 4330
4320 ON J GOSUB 4580,4600,4620
     ,4640,4660,4680
4330 NEXT J
4340 FOR K=1 TO 4
4350 IF A$<>MID$(C$,K,1) THEN
     {SPACE}4370
4360 ON K GOSUB 4410,4460,4530
     ,4560
4370 NEXT K
4380 IF M5=0 THEN 4210
4390 M5=0
4400 RETURN
4410 FOR I=1 TO 8
4420 PRINT MID$(V$,I,1);"
     {2 SPACES}";V(I)
4430 NEXT I
4440 PRINT
4450 RETURN
4460 PRINT "IN WHAT VARIABLE "
     ;
4470 INPUT A$
4480 FOR I=1 TO 8
4490 IF A$<>MID$(V$,I,1) THEN
     {SPACE}4510
4500 V(I)=M
4510 NEXT I
4520 RETURN
4530 PRINT C$;" ";C1$;" MEM=";
     M
4540 PRINT
4550 RETURN
4560 M5=1
4570 RETURN
4580 M=M+T
4590 GOTO 4690
4600 M=M-T
4610 GOTO 4690
4620 M=M*T
4630 GOTO 4690
4640 M=M/T
4650 GOTO 4690
4660 T=M
4670 GOTO 4690
4680 M=0
```

```
4690 PRINT "MEM=";M
4700 RETURN
4710 PRINT "*FUTURE VALUE $"
4720 C=0
4730 GOSUB 3950
4740 RETURN
4750 PRINT "*PRESENT VALUE $"
4760 C=1
4770 GOSUB 3950
4780 RETURN
4790 PRINT "PRINCIPAL $"
4800 C=1
4810 GOSUB 3950
4820 P=V(C)
4830 RETURN
4840 PRINT "ANNUAL INT RATE (%
     )"
4850 C=2
4860 GOSUB 3950
4870 RETURN
4880 PRINT "FOR # OF YEARS"
4890 C=3
4900 GOSUB 3950
4910 RETURN
4920 PRINT "FOR # OF MONTHS"
4930 C=4
4940 GOSUB 3950
4950 Y=V(C-1)+V(C)/12
4960 RETURN
4970 PRINT "# OF PERIODS (COMP
     OUNDING, DEPOSITS, WITHDR
     AWALS, PAYMENTS) YEARLY"
4980 C=5
4990 GOSUB 3950
5000 RETURN
5010 PRINT "PAYMENTS $"
5020 C=6
5030 GOSUB 3950
5040 RETURN
5050 PRINT "TERM OF LOAN:"
5060 GOSUB 4880
5070 GOSUB 4920
5080 RETURN
5090 PRINT
5100 PRINT "FUTURE VALUE:$";V(
     1)
5110 RETURN
5120 PRINT
5130 PRINT "REQUIRED INVESTMEN
     T:$";V(2)
5140 RETURN
5150 PRINT
5160 PRINT "ANNUAL INT RATE (%
     ) REQUIRED:";V(3)*100
5170 RETURN
5180 V(5)=V(4)-INT(V(4))
5190 V(5)=INT(INT(12*V(5)*10+.
     5)/10)
5200 V(4)=INT(V(4))
5210 IF V(5)<>12 THEN 5240
5220 V(4)=V(4)+1
5230 V(5)=0
5240 PRINT
5250 PRINT "# OF YEARS AND MON
     THS:";V(4);",";V(5)
5260 RETURN
5270 PRINT
5280 IF TE>=0 THEN 5310
5290 PRINT "THIS IS A LOSING I
     NVESTMENT."
5300 RETURN
5310 PRINT "THIS IS A PROFITAB
     LE INVESTMENT."
5320 RETURN
5330 PRINT
5340 PRINT "HIT <RETURN> TO CO
     NTINUE"
5350 A$=""
5360 INPUT A$
5370 IF A$<>"" THEN 5350
5380 RETURN
5390 GOSUB 5450
5400 PRINT "LOAN AMORTIZATION
     {SPACE}SCHEDULE FOR YR ";
     J1
5410 PRINT "PRIN $";V(2);"
     {2 SPACES}RATE ";V(3)*100
     ;"%";"{2 SPACES}PAYM $";V
     (7)
5420 PRINT
5430 PRINT "#{3 SPACES}BEG BAL
     {3 SPACES}PRINC{3 SPACES}
     INT{5 SPACES}END BAL"
5440 RETURN
5450 HOME
5460 RETURN
```

## Program 2: Modifications For Commodore 64, Plus/4, and 16

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
150 D5=6
160 S1=3
170 S2=13
180 S3=21
190 S4=29
3580 PRINT MID$(STR$(N5),2,LEN
     (STR$(N5))-1);TAB(S1);INT
     (P*100+.5)/100;
5450 PRINT CHR$(147)
```

## Program 3: Modifications For Commodore PET

*For PET/CBM models, make the following modifications* in addition to *the changes shown in Program 2.*

```
4010 PRINTCHR$(160);"{3 LEFT}"
     ;:INPUT A$
4020 IF A$<>CHR$(160) THEN 404
     0
5360 GETA$
5370 IF A$<>CHR$(13) THEN 5360
```

## Program 4: Modifications For VIC-20

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
140 Q$=CHR$(13)
150 D5=3
160 S1=3
170 S2=13
180 S3=S1
190 S4=S2
3580 PRINT MID$(STR$(N5),2,LEN
     (STR$(N5))-1);TAB(S1);INT
     (P*100+.5)/100;
5430 PRINT "#{3 SPACES}BEG BAL
     {3 SPACES}PRINC{7 SPACES}
     INT{7 SPACES}END BAL"
5450 PRINT CHR$(147)
```

## Program 5: Modifications For Atari

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
105 DIM A$(10),C$(4),C1$(
    12),V$(8),Q$(1):POKE
    82,0:FOR I=1 TO 8:V(I
    )=0:NEXT I
160 S1=4
170 S2=14
180 S3=22
190 S4=30
3580 PRINT N5;:POKE 85,S1
     :PRINT INT(P*100+0.5
     )/100;
3590 POKE 85,S2:PRINT INT
     (PP*100+0.5)/100;:PO
     KE 85,S3
3650 PRINT I;:POKE 85,S4:
     PRINT INT(P*100+0.5)
     /100;
4260 IF A$<>V$(I,I) THEN
     4290
4310 IF A$<>C1$((J-1)*2+1
     ,(J-1)*2+2) THEN 433
     0
4350 IF A$<>C$(K,K) THEN
     4370
4420 PRINT V$(I,I);"  ";V
     (I)
4490 IF A$<>V$(I,I) THEN
     4510
5450 PRINT CHR$(125)
```

## Program 6: Modifications For IBM PC/PCjr

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

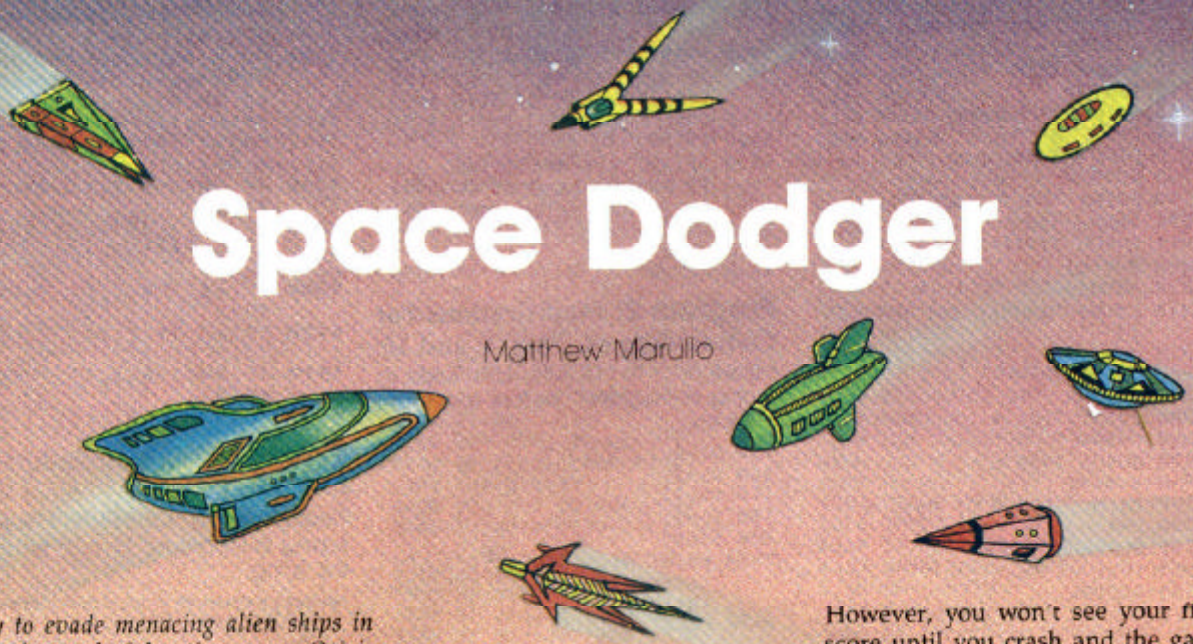
```
90 WIDTH 40:KEY OFF:DEF SEG=0
   :POKE 1047,PEEK(1047) OR 6
   4
160 S1=4
170 S2=14
180 S3=22
190 S4=30
3500 I=INT(P*V(3)/V(6)*100+.5
     )/100:B=I:GOSUB 5470:I$=
     B$
3580 PRINT MID$(STR$(N5),2,LE
     N(STR$(N5))-1);TAB(S1);:
     B=P:GOSUB 5470:PRINT B$;
3590 PRINT TAB(S2);:B=PP:GOSU
     B 5470:PRINT B$;Q$;TAB(S
     3);
3650 PRINT I$;TAB(S4);:B=P:GO
     SUB 5470:PRINT B$;
5340 PRINT "HIT <ENTER> TO CO
     NTINUE"
5450 CLS
5470 TE=0:B$=STR$(B):FOR K=1
     TO LEN(B$):IF MID$(B$,K,
     1)="." THEN TE=K:K=LEN(B
     $)
5480 NEXT K
5490 IF TE=0 THEN RETURN ELSE
      B$=MID$(B$,1,TE+2):RETU
     RN
```

## Program 7: Modifications For TI-99/4A

```
140 Q$=CHR$(13)
150 D5=6
160 S1=9
170 S2=20
180 S3=S1
190 S4=S2
4260 IF A$<>SEG$(V$,I,1)THE
     N 4290
4310 IF A$<>SEG$(C1$,(J-1)*
     2+1,2)THEN 4330
4350 IF A$<>SEG$(C$,K,1)THE
     N 4370
4420 PRINT SEG$(V$,I,1);"
     ";V(I)
4490 IF A$<>SEG$(V$,I,1)THE
     N 4510
5340 PRINT "HIT <ENTER> TO
     CONTINUE"
5430 PRINT " #";TAB(S1+1);"
     BEG BAL";TAB(S2+1);"PR
     INC";Q$;TAB(S3+1);"  I
     NT";TAB(S+1);"END BAL"
5450 CALL CLEAR
```

# Space Dodger

Matthew Marullo



*Try to evade menacing alien ships in this fast, colorful action game. Originally written for the TI-99/4A with Extended BASIC, adaptations have been added for the Commodore 64, unexpanded VIC-20, Apple II series, and Atari. The Commodore, Apple, and Atari versions require a joystick.*

Get ready for a game which demands extremely sharp eye-hand coordination and judgment of time and distance. "Space Dodger" is an addictive test of your physical reflexes.

When you type RUN, there's a brief wait while the program initializes. Then the game opens with your spaceship on the left side of the screen, superimposed over a random starfield. On the right side of the screen is a lineup of several colorful alien ships. When the action starts, the aliens begin moving toward your ship at different speeds. Your job is to avoid a disastrous midspace collision that will turn your vessel into a lump of smoking metal.

To dodge the reckless aliens, you'll have to move up or down. But don't move too far and try to escape the screen—the boundaries are guarded by cuboids (cube-shaped asteriods) zipping along at the speed of light. The cuboids are even more dangerous than the alien ships because they travel too fast to dodge.

## Moving Up The Ranks

The longer you evade the oncoming aliens, the more points you gain. However, you won't see your final score until you crash and the game ends. At that time you're also ranked according to your value to the Space Service: Space Cadet, Corporal, Sergeant, Captain, or Major.

Every time you advance a rank, the game pauses briefly before it continues to the next level. When it restarts, you'll notice the alien ships fly across the screen even faster. Your score adds up faster, too.

But beware—Space Dodger is not as easy as it looks. Chances are you'll play for quite a while before you even advance beyond Space Cadet.

## TI Version

Control your ship with the keyboard: Press the E key to move up, the X key to move down. You can achieve finer control by repeatedly tapping the keys, rather than holding them down.

Space Dodger is one of the fastest BASIC games we've seen for the TI, and it makes good use of the built-in sprite graphics and collision detection.

## Commodore 64/VIC-20 Versions

Plug a joystick into port 2 on the Commodore 64. When the game begins, press the joystick forward to move your ship up, and pull it back to move down.

Joystick controls on the VIC are the same as on the 64. Before loading Space Dodger, be sure to unplug any cartridges. VIC Space Dodger is in

two parts. Program 3 is the loader, which creates the custom characters, then loads and runs Program 4. Type in and save both programs, *using the filename "SD" when saving Program 4.* If you are using tape, save Program 4 on the same tape, immediately after Program 3. Tape users also need to change ,8 to ,1 in line 30 of Program 3.

Both Commodore versions include three additional ranks beyond the lower levels—Lieutenant, General, and Master. The ships are created with multicolor sprites on the 64 and custom multicolor characters on the VIC. Ring modulation and filtering help produce the 64 version's uncanny sound effects.

## Apple Version

Written entirely in machine language, Apple Space Dodger works on any Apple II series computer with any version of Apple DOS. The machine language in Program 4 must be entered with the Apple's built-in machine language monitor. You don't need to understand machine language to enter the program.

To type in Program 5, first enter the monitor by typing CALL −151. The Applesoft prompt (normally a ]) will be replaced by the monitor's prompt, an asterisk (*). To enter a line from the listing, first type in the four-digit hexadecimal number, then type a colon (:) instead of the hyphen shown in the listing. This is the address where you'll enter the rest of the line. Type in the rest of the line after the colon, leaving a space between each two-digit number. After eight numbers, press RETURN and continue to the next line. If you want to review what you've entered to check for accuracy, you can list a block of data by typing the address of the first location in the range, then a period, then the last address, and then RETURN.

When you're done typing the program, save it on disk with this command:

```
BSAVE SPACE.DODGER,A$7000,
L$8AA
```

Because it's difficult to type a listing this long without making errors, we've included a small checksum program (Program 6) which detects typos. To use it, load the machine language program from disk by entering BLOAD SPACE.DODGER, then run Program 6. If you have made a typo, it will tell you where to look to find the mistake.

When Program 5 is error-free, save a copy on disk. Then run it by typing BRUN SPACE.DODGER. Plug in a joystick, and push forward on the joystick to move your ship up, or pull back on the stick to move down. You have a total of three ships in each game.

## Atari Version

Atari Space Dodger works on the 400/800, XL series, and new XE series computers. With a joystick plugged into port 1, you can push the stick forward to move your ship up, and pull back to move the ship down.

The Atari version's multicolored alien ships are created with an unusual implementation of player/missile (P/M) graphics. Ordinarily, the Atari can display a maximum of only four player shapes (or five if you combine the four missiles into an additional player). Each player can be only one color and is limited in width, but can be as tall as the entire screen. But in Space Dodger, one player is used for your ship, and the remaining three players are cleverly combined to make *12 multicolor* alien ships.

The program takes advantage of a technique which allows multiple colors in overlapping players. All three alien players begin at the same horizontal location and are assigned different colors. The P/M shape data is then defined so that visible portions of the underlying alien players can be seen through "holes" in the overlapping players. Thus, each ship is actually a three-colored conglomerate of overlapping shapes. To create the effect of separate ships, the remaining P/M data is filled with zeros to make blank zones between each alien craft.

The result is 12 multicolor ships, but without additional programming tricks, they'd all have to move in unison. Moving one alien player without the others would destroy the carefully arranged multicolor effect.

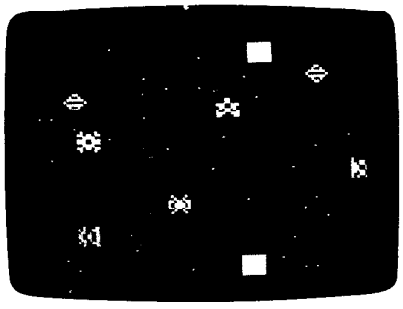
To move the aliens at different speeds and horizontal locations, Space Dodger uses *display list interrupts.* Briefly, the Atari display list is a set of instructions that tells the computer what to display at a given point on the screen as the TV's raster beam sweeps from top to bottom. By manipulating the display list with machine language routines, Space Dodger makes its three overlapped players act like a dozen independently mobile shapes.

The fast, smooth motion of the alien ships is achieved by moving them only during the Atari's *vertical blank interrupt* (the short interval during which the TV's raster beam moves from the bottom of the screen to the top to scan another frame). Naturally, machine language is also needed to make this work.

You can learn more about using both types of interrupts in *De Re Atari,* published by Atari Computers, Inc., as well as *COMPUTE!'s First Book of Atari* and *COMPUTE!'s First Book of Atari Graphics.*

## Program 1: TI Space Dodger

```
100 REM EXTENDED BASIC REQU
    IRED
110 FOR I=0 TO 3 :: READ F$
    (I):: NEXT I
120 DATA SPACE CADET,CORPOR
    AL,SERGEANT,CAPTAIN
130 FOR I=97 TO 103 :: READ
     A$ :: CALL CHAR(I,A$):
    : NEXT I
140 CALL CHARPAT(33,D$)
150 DATA 18247E817E241800,1
    818E7665A99E7C3
160 DATA 42DB3CE7E73CDB42,8
    199DBE7E7DB99A5,4224DBB
    DBDDB2442
170 DATA 23A76BDBDB6BA723,F
    FFFFFFFFFFFFFFF
180 A=15 :: B=25 :: C=12 ::
     D=22 :: E=18 :: F=28 :
    : SC=0 :: T=0 :: LEVEL=
    0
190 CALL CHAR(96,"18247E817
    E241800"):: CALL CLEAR
    :: CALL MAGNIFY(2):: CA
    LL SCREEN(2):: CALL CHA
    R(33,"1"):: CALL COLOR(
    1,2,2)
200 RANDOMIZE :: FOR STAR=1
     TO 35 :: CALL HCHAR(IN
    T(24*RND)+1,INT(32*RND)
    +1,33):: NEXT STAR :: C
    ALL COLOR(1,16,2)
```



*Reckless alien ships hurtling through space make life hazardous in "Space Dodger" (TI version).*

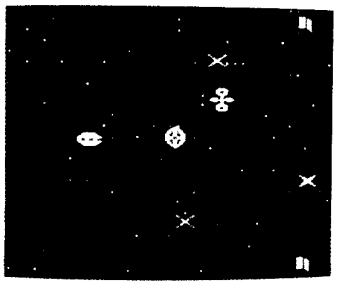
```
210 CALL SPRITE(#1,96,11,75
    ,45,#2,97,INT(6*RND)+3,
    25,180,#3,98,INT(7*RND)
    +3,50,180)
220 CALL SPRITE(#4,99,INT(1
    2*RND)+3,75,180,#5,100,
    INT(10*RND)+3,100,180,#
    6,101,INT(8*RND)+3,125,
    180)
230 CALL SPRITE(#7,102,INT(
    11*RND)+3,150,180,#8,10
    3,16,6,10,#9,103,16,175
    ,10)
240 FOR DELAY=1 TO 650 :: N
    EXT DELAY
250 CALL SOUND(500,110,0,22
    0,0,330,0)
260 CALL MOTION(#8,0,-120,#
    9,0,-120)
270 CALL MOTION(#2,0,-A,#3,
    0,-B,#4,0,-C,#5,0,-D,#6
    ,0,-E,#7,0,-F)
280 CALL KEY(1,K,S):: T=T+1
     :: SC=SC+1+LEVEL*10 ::
     IF T=200 OR T=375 OR T
    =500 THEN 430
290 IF T=550 THEN 460
300 IF S=0 THEN Y=0 :: CALL
     SOUND(1,-7,6)
310 IF K=5 THEN Y=-15 ELSE
    IF K=0 THEN Y=15
320 IF ABS(Y)=15 THEN CALL
    SOUND(-5,1050,3,450,4,-
    6,1)
330 CALL MOTION(#1,Y,0):: C
    ALL COINC(ALL,CC):: CAL
    L POSITION(#1,DR,DC)::
    IF CC=-1 OR DR>192 THEN
     340 ELSE 280
340 CALL DELSPRITE(#2,#3,#4
    ,#5,#6,#7,#8,#9):: CALL
     MOTION(#1,0,0)
350 FOR X=1 TO 10 :: CALL C
    HAR(96,"42008100810081
    42")
360 CALL SOUND(100,-7,X+3)
370 CALL CHAR(96,"000000181
    8000000"):: NEXT X
380 CALL DELSPRITE(#1):: CA
    LL CLEAR :: CALL CHAR(3
    3,D$)
390 FOR V=1 TO 8 :: CALL CO
    LOR(V,16,2):: NEXT V ::
     DISPLAY AT(10,11):"GOO
    D TRY,"
400 DISPLAY AT(12,(29-LEN(F
    $(LEVEL)))/2):F$(LEVEL)
    &"!" :: DISPLAY AT(15,1
    0):"SCORE: "&STR$(SC)::
     DISPLAY AT(18,6):"PLAY
     AGAIN (Y/N) ?"
410 CALL KEY(0,K,S):: IF S=
    0 THEN 410
420 U$=CHR$(K):: IF U$="Y"
    THEN 180 ELSE CALL CLEA
    R :: STOP
430 A=A+10 :: B=B+10 :: C=C
    +10 :: D=D+10 :: E=E+10
    :: F=F+10 :: LEVEL=LEV
    EL+1
440 CALL SOUND(300,440,0,65
    9,0):: CALL DELSPRITE(A
    LL):: CALL SCREEN(10)::
     FOR DELAY=1 TO 300 ::
    NEXT DELAY
450 CALL SCREEN(2):: GOTO 2
    10
460 CALL CLEAR :: FOR U=1 T
    O 8 :: CALL COLOR(U,16,
    1):: NEXT U :: CALL SCR
    EEN(5):: CALL DELSPRITE
    (ALL)
470 CALL CHAR(33,D$):: CALL
     SOUND(2000,131,2,262,2
    ,523,2):: DISPLAY AT(5,
    13):"WOW!" :: DISPLAY A
    T(8,6):"NICE GOING,"&"
    MAJOR!"
480 DISPLAY AT(15,6):"I CON
    GRATULATE YOU" :: DISPL
    AY AT(17,6):"ON YOUR NE
    RVES ---" :: DISPLAY AT
    (19,6):"AND YOUR TALENT
    !"
490 FOR DELAY=1 TO 1000 ::
    NEXT DELAY :: CALL CHAR
    (96,"1818183C243CC3FF")
    :: CALL SPRITE(#1,96,14
    ,95,115)
500 CALL SOUND(1000,440,0,6
    59,0):: FOR DELAY=1 TO
    400 :: NEXT DELAY :: CA
    LL DELSPRITE(#1):: GOTO
     190
```

## Program 2: Commodore 64 Space Dodger

*Version by Kevin Mykytyn, Editorial Programmer*

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
20 C=54272:POKE54296,15
                          :rem 235
30 FORA=1TO8:READB$(A):NEXT:DA
   TA SPACE CADET,CORPORAL,SER
   GEANT,CAPTAIN,MAJOR:rem 160
35 DATA LIEUTENANT,GENERAL,MAS
   TER                    :rem 157
40 FORA=16064TO16319:READB:POK
   EA,B:NEXT              :rem 154
50 FORA=16256TO16319:POKEA+64,
   PEEK(A):NEXT:FORA=16347TO16
   352:POKEA,250:NEXT     :rem 62
60 POKE53276,255:POKE2040,13:F
   ORA=2041TO2045         :rem 247
70 POKEA,251:NEXT:POKE2046,13
                          :rem 217
80 FORA=832TO959:READB:POKEA,B
   :NEXT                  :rem 221
90 POKE53280,15:POKE53281,0:PO
   KE53251,125:GOSUB880
                          :rem 123
100 PRINT"{CLR}{10 DOWN}
    {14 RIGHT}{BLK}SPACE DODGE
    R"                    :rem 230
110 POKE53269,255:POKE53249,12
    5:FORA=90TO255:POKE53248,A
                          :rem 233
120 POKE53250,345-A:IFA=155THE
    NPOKE53249,120:POKE53251,1
    30                    :rem 38
130 POKE55696+(A-24)/8,7
                          :rem 236
140 NEXT:FORTD=1TO800:NEXT:POK
    E53269,0              :rem 125
150 RA=1:POKE253,5:SC=0:L=200:
    POKE53285,7:POKE53286,2:PO
    KE53287,7             :rem 4
160 B=50:FORA=53249TO53261STEP
    2:POKEA,B:B=B+30:NEXT:PRIN
    T"{CLR}":FORA=1TO65
                          :rem 201
170 Q=1024+RND(1)*999:POKEQ,46
    :POKEQ+C,RND(1)*15:NEXT
                          :rem 102
180 POKE53278,0:POKE2047,255:P
    OKE254,0:SYS49152:POKE5326
    9,255                 :rem 3
```



*Commodore 64 "Space Dodger."*

```
190 SC=SC+(PEEK(253)-4)/2:IFSC
    >=LTHENGOSUB330:L=L*3:RA=R
    A+1:GOTO190           :rem 46
200 IFPEEK(53278)<128THEN190
                          :rem 64
210 POKE254,1:POKE54273,4:POKE
    54277,27:POKE54278,0:POKE5
    4276,128:POKE54276,129
                          :rem 207
220 POKE2047,14:POKE53294,8
                          :rem 245
230 FORTD=1TO200:NEXT:POKE5326
    9,127:FORTD=1TO500:NEXT:PO
    KE53269,0             :rem 14
240 POKE56333,129:POKE53274,0
                          :rem 91
250 PRINT"{CLR}":PRINT"{CYN}
    {7 DOWN}{14 RIGHT}SCORE:"I
    NT(SC)                :rem 58
260 IFSC>HSTHENHS=SC       :rem 51
270 PRINT"{3 DOWN}{12 RIGHT}HI
    GH SCORE:"INT(HS)     :rem 166
280 PRINT"{3 DOWN}"SPC(17-LEN(
    B$(RA))/2)"RANK: "B$(RA)
                          :rem 57
290 PRINT"{YEL}{5 DOWN}
    {6 RIGHT}PLAY AGAIN? (UP-Y
    ES DOWN-NO)           :rem 214
300 Q=PEEK(56320):IF(QAND1)=0T
    HEN150                :rem 56
310 IF(QAND2)=0THENSYS832
                           :rem 2
320 GOTO300               :rem 97
330 POKE56333,129:POKE53274,0:
    SYS 65418:POKE53280,2:POKE
    53269,0               :rem 51
340 FORA=53248TO53260STEP2:POK
    EA,40                 :rem 77
350 POKEA-52569,40:NEXT:POKE 2
    52,127:POKE53264,127:POKE5
    3269,255:POKE53263,140
                          :rem 171
360 FORTD=1TO180           :rem 194
370 IF(PEEK(56320)AND16)<>16TH
    ENWAIT56320,16,0:WAIT56320
    ,16,16:TD=180         :rem 155
380 NEXTTD:SYS 49152:POKE53280
    ,15:POKE253,PEEK(253)+1
                          :rem 72
390 POKE53278,0:RETURN    :rem 74
400 DATA0,0,0,0,0,0,0,0:rem 98
410 DATA0,0,0,0,0,0,0,60
                          :rem 153
420 DATA0,60,59,0,236,14,130,1
    76                    :rem 74
430 DATA2,105,128,0,150,0,0,15
    0                      :rem 4
440 DATA0,2,105,128,14,130,176
    ,59                   :rem 126
450 DATA0,236,60,0,60,0,0,0
                          :rem 62
460 DATA0,0,0,0,0,0,0,0
                          :rem 104
470 DATA0,0,0,0,0,0,0,1
                          :rem 106
```

# PROGRAMMING THE TI

C. Regena

# Japanese Characters

I just returned from an interesting trip to Japan. I met several TI-99/4 users and spoke to a Commodore user group at Misawa Air Force Base. I ate all kinds of food and slept on the floor, experiencing real Japanese life that most tourists wouldn't see. I closed my eyes during the drives down narrow streets, but had fun shopping in the crowded stores. There were dozens of computer magazines—much like our magazine racks. I bought several that had interesting program listings to type in. The programming is in BASIC, so I can understand it, but I cannot understand the Japanese articles which tell what the programs do.

I was able to spend only a couple of hours in the Akihabara district of Tokyo seeing all the electronics shops. I would have enjoyed a longer time there, but then I would have just spent more money. Some people collect dolls or other trinkets, but it seems I collect computers. I bought an MSX computer that looks like lots of fun. (See "MSX Is Coming," Parts 1 and 2, COMPUTE!, December 1984 and January 1985.) I decided on a 64K Hitachi MB-H2 because it has a built-in cassette recorder and two cartridge slots (one is reserved for future disk drive expansion). It has one built-in music program that turns the keyboard into an organ. Another built-in program is like the Macintosh's *MacPaint*.

MSX BASIC is Microsoft BASIC with extended graphics commands such as LINE, CIRCLE, and PAINT. The music commands are similar to those in TI BASIC because the computer has the same sound chip found in the TI. MSX computers also use the TI video chip that allows 32 sprites. A graphics key lets you change the keyboard to graphics characters (similar to those on Commodore computers) plus some Japanese Kanji characters. Another key sets the keyboard to Hiragana characters (like our cursive writing), and a SHIFT adds the Katakana characters (comparable to our printing).

## Reprogramming The Keyboard

This brings us to the following program. I've had several inquiries about how to print Japanese or Chinese characters on the TI, or how to change our QWERTY keyboard to a Dvorak keyboard. This program allows the keys to print the Japanese Katakana characters. You would use a similar technique for any other symbols you choose.

Reprogramming the keyboard requires two steps: You need to define the appropriate symbols, then print them on the screen when the corresponding key is pressed. I decided to use CTRL as the key to switch characters because I wanted to keep the English alphabet intact.

To find out what character code is returned for each keypress, you can refer to a chart in the appendix of the *User's Reference Manual* that came with the TI, or you can run a short program. As a key is pressed, the character code is printed on the screen. Notice that if you hold down CTRL while pressing a key, the computer returns a different number than it returns if the key is pressed by itself.

```
110 CALL KEY(0,K,S)
120 IF S<1 THEN 110
130 PRINT K
140 GOTO 110
150 END
```

You'll see that CTRL in combination with the number keys yields values greater than 156. For our special definitions, we are limited to numbers up to 156. Therefore, use SHIFT instead of CTRL for the top row. This means the symbols will be redefined.

## Phonetic Japanese

Each Japanese Katakana symbol represents a syllable. The following chart places the characters in the same order as on the MSX computer keyboard, with a few exceptions on the right side of the keyboard:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| SHIFT | 1 | 2 | 3 | 4 | 5 | a | i | u | e | o |
| CTRL | Q | W | E | R | T | ka | ki | ku | ke | ko |
| CTRL | A | S | D | F | G | sa | shi | su | se | so |
| CTRL | Z | X | C | V | B | ta | chi | tsu | te | to |
| SHIFT | 6 | 7 | 8 | 9 | 0 | na | ni | nu | ne | no |
| CTRL | Y | U | I | O | P | ha | hi | fu | he | ho |
| CTRL | H | J | K | L | ; | ma | mi | mu | me | mo |
| CTRL | N | M | , | . | / | ya | yu | yo | wa | wo |
| SHIFT | + | - | : | < | > | ra | ri | ru | re | ro |

When you run the program, instructions appear and then you can press a key. The Japanese Katakana symbol will appear along with the *romaji* or romanized syllable. You can use the program to practice "writing" Japanese or to learn how to read the symbols. You may want to use these character definitions and placements to expand to a Japanese language program which uses words and phrases.

Once you're familiar with this programming technique, you can change the character definitions to symbols for a different Asian language. Or you can try printing a code, such as Braille. Or you can convert your TI keyboard into a keyboard of graphic shapes.

## Program Explanation

S$ is the array to hold the syllables. To make things easy I just used the character number as the element number in S$. The characters from 128 to 156, however, subtract 128 for the element number. The highest symbol character number is 94, so the DIM statement reserves 94 for the array size. In a larger program you could be more efficient by numbering the elements differently, and you would need only about 50 elements.

Lines 150–350 define the characters for the symbols on the top row of the keyboard (SHIFT and the numbers) with the corresponding syllable sounds.

Lines 370–400 define the characters for the main section of the keyboard (CTRL and the letters), using the DATA statements in lines 420–630. The FOR-NEXT loop goes from character number 128 to 156 and READs first a character definition, then the syllable. Be careful typing the DATA lines. Don't use any extra commas, and don't put a comma at the end of a line. Each DATA statement except the last has three sets of character definitions with syllables.

Lines 540–650 define characters and syllables for the rest of the keyboard. Lines 660–710 print brief instructions. Lines 720–780 detect which key is pressed and accept only valid keys. The IF-THEN statements make sure the keypress is within certain ranges to print a symbol and a corresponding syllable. Line 790 prints the Japanese character. Lines 800–840 print the corresponding syllable, then return to the CALL KEY statement for the next keypress.

If you prefer to save typing effort, you can obtain a copy of this program by sending a blank cassette or disk, a stamped, self-addressed mailer, and $3 to:

*C. Regena*
*P.O. Box 1502*
*Cedar City, UT 84720*

Please specify the title of the program ("Japanese Katakana Characters") and that you need the TI version.

### Japanese Katakana Characters

```
100 REM  KATAKANA
110 DIM S$(94)
120 CALL CLEAR
130 PRINT "CONVERTING THE TI KEYBOA
    RD"
140 PRINT :"TO JAPANESE CHARACTERS"
    :::::
150 S$(32)=" "
160 CALL CHAR(33,"FE0212C1010102")
170 S$(33)="A"
180 CALL CHAR(64,"0404081868080808"
    )
190 S$(64)="I"
200 CALL CHAR(35,"107C4202040400808"
    )
210 S$(35)="U"
220 CALL CHAR(36,"007C1010101010FE")
230 S$(36)="E"
240 CALL CHAR(37,"0808FE1828284808"
    )
250 S$(37)="O"
260 CALL CHAR(94,"0007E0800810102")
270 S$(94)="NA"
280 CALL CHAR(38,"0000380000007E")
290 S$(38)="NI"
300 CALL CHAR(42,"007E0204281020C02"
    )
310 S$(42)="NU"
320 CALL CHAR(40,"107E0408101424A49"
    )
330 S$(40)="NE"
340 CALL CHAR(41,"0004040800810102")
350 S$(41)="NO"
360 REM
370 FOR C=128 TO 156
380 READ C$,S$(C-128)
390 CALL CHAR(C,C$)
400 NEXT C
410 REM
420 DATA 007E027E02027E,YO,0024FE24
    0404080808,SA,606030282420202,TO
430 DATA 00545404040408102,SU,007E04
    0810182442,SU,407C4404040808081,K
    U
```

```
440 DATA 00203C642820201E,SE,002212
    020404081,SO,007E0204180C,MA
450 REM
460 DATA 007E02020408102,FU,3008001
    008003008,MI,00102020445C62,MU
470 DATA 00041408141020 2,ME,0038080
    808087C,YU,207E22222C20202,YA
480 DATA 00102C43,HE,10107C1010549 2
    1,HO,1016FA1212222242,KA
490 DATA 407EC80808101,KE,000020012
    204183,SHI,003C040404047C,KO
500 REM
510 DATA 00202020302020 3F,HI,38007C
    101010202,TE,402C701678040201,K
    I
520 DATA 043808083E08101,CHI,100824
    22224202,HA,007E42441C08081,TA
530 DATA 7E4202020404080 8,WA,003810
    7810101 00C,MO
540 CALL CHAR(47,"7E027E0204040808"
    )
550 S$(47)="WO"
560 CALL CHAR(43,"7C007C0404081 02")
570 S$(43)="RA"
580 CALL CHAR(45,"2424040408080801")
590 S$(45)="RI"
600 CALL CHAR(58,"00282828 2A4C08")
610 S$(58)="RU"
620 CALL CHAR(60,"4040404448506")
630 S$(60)="RE"
640 CALL CHAR(62,"003E424242427E")
650 S$(62)="RO"
660 PRINT :"USE THE SHIFT KEY WITH
    THE"
670 PRINT "TOP ROW OF KEYS,"
680 PRINT "PLUS, MINUS, COLON,"
690 PRINT "GREATER THAN, OR LESS TH
    AN."
700 PRINT :"SLASH IS 'WO'."
710 PRINT :"USE CTRL WITH OTHER KEY
    S.";;
720 CALL KEY(0,K,S)
730 IF S<1 THEN 720
740 IF K>156 THEN 720
750 IF (K=94)+(K=64)+(K=60)+(K=62)+
    (K=58)+(K=47)+(K=45)THEN 790
760 IF (K<128)+(K>43)=-2 THEN 720
770 IF K<32 THEN 720
780 IF (K=34)+(K=39)THEN 720
790 PRINT :TAB(12);CHR$(K);"  ";
800 IF K<128 THEN 830
810 PRINT S$(K-128)
820 GOTO 720
830 PRINT S$(K)
840 GOTO 720
850 END
```



# IBM Personal Computing

Donald B. Trivette

## Titling Your Vacation

Most of us vacation at comfortable places like the beach or the mountains, but my neighbors Don and Judy Getz prefer the extraordinarily *un*-comfortable. One year they spent a month at the Khyber Pass in northern Pakistan; last year they took a boat-bus-train trip up the Amazon and through rural Brazil, Bolivia, and Peru. Just 21 fun-filled days, they say, sleeping on hard beds, drinking bottled water, and coping with South American railway schedules.

When you go to Cochabamba and Cotabambas, you've *got* to take slides and movies (documentary evidence) to show the folks back home what a grand time you've had. And once you've returned, it's useful to title the slides so that six months from now, *you* can tell Cochabamba from Cotabambas. That's why Don called and wanted to know if I had a computer program that would produce professional-looking titles he could photograph directly from the computer screen. I thought I had several programs that would do just that.

## The Missing Mouse

The first program that came to mind was *ColorPaint* for the PCjr. You've probably seen the

# NEWS&PRODUCTS

### 64 Music Program

Brøderbund Software has announced several new packages, including *The Music Shop*, a music composition tool and music synthesizer, for the Commodore 64. The program allows you to create, store, and edit compositions and print out sheet music. The synthesizer can add sound textures. Suggested retail price is $44.95 (disk). Versions for the IBM PCjr and Apple Macintosh are scheduled for this spring.

Also from Brøderbund are *The Ancient Art of War* ($44.95), a new strategy game for the IBM PC and PCjr, featuring 11 built-in war campaigns from the pages of history; *Where in the World is Carmen Sandiego* ($39.95) for Apple II-series computers, a mystery/adventure educational game with color animation and sound effects plus different scenarios involving 30 countries and 10 villains; and *Science Toolkit* ($59.95) for the Apple II series, which turns the computer into a science lab simulator for a variety of applications.

*Brøderbund Software, 17 Paul Dr., San Rafael, CA 94903-2101*
**Circle Reader Service Number 223.**

### Software For Dieters

*The Original Boston Computer Diet*, a personalized weight-loss program, has been released by Scarborough Systems. Developed by Harvard University and Harvard Medical School nutritionists and psychiatrists, the program analyzes weight, height, eating habits, and personality traits to create an individual diet.

Calories and other nutritional values are computed using a database of about 700 items. The program also features a "computer weight-loss counselor" and a cartoon character which offer encouragement and advice.

The program is currently available for the IBM PC/XT/PCjr (with 128K) for $79.95. Versions for the Apple II series and Commodore 64 will be available soon.

Among Scarborough's other recent releases are *Make Millions*, a business simulation adventure by Tom Snyder, available initially for the Macintosh with versions to follow for the Apple II series and IBM PC/PCjr (price not available); and *Build-a-Book*, a program and kit package which allows children to write their own stories, print them out, and bind the finished work as a four-color book. The program is available for $34.95 for the Apple II series, the IBM PC and PCjr, and the Commodore 64. (Additional two-book replacement sets are priced at $19.95).

*Scarborough Systems, Inc., 25 N. Broadway, Tarrytown, NY 10591*
**Circle Reader Service Number 224.**

### Electronic Novels

A series of sophisticated all-text adventure programs has been introduced by Synapse for the PC, Apple, Atari, and Commodore computers. The first titles in the series are *Mindwheel*, a journey into the minds of four deceased people for clues to the Wheel of Wisdom; and *Essex*, an intergalactic search and rescue mission.

Additional novels are underway, including *Brimstone*, a medieval adventure story; *Breakers*, a science-fiction fantasy on the planet Borg; and *Ronin*, a samurai epic. IBM and Apple versions will sell for $44.95, and Atari and Commodore versions will be priced at $39.95.

*Synapse Software, 5221 Central Avenue, Richmond, CA 94804*
**Circle Reader Service Number 225.**

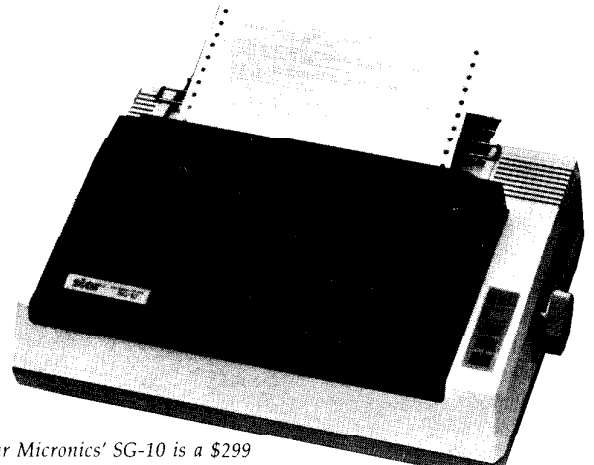
### New Printers

A new line of dot-matrix printers with a wide range of prices and features has been introduced by Star Micronics. These include the SG, SD, and SR series, which combine the Star standard and PC printer lines into one line that is switch-selectable for all personal computers. They are available in two widths (10-inch and 15-inch) and feature near letter quality printing. Prices range from $299 for the 120 characters per second (cps) SG-10 to $799 for the 200 cps SR-15.

Also new from Star are the STX-10, a thermal printer for $199, and the Powertype, a daisywheel printer for $499.

*Star Micronics, Inc., 200 Park Avenue, New York, NY 10166*
**Circle Reader Service Number 226.**





*Star Micronics' SG-10 is a $299 dot matrix printer with speeds of 120 characters per second (cps) in draft mode and 30 cps in near letter quality mode.*