

\$14.95

GAMES COMMODORES PLAY



by Philip W. Dennis and Greg Minter

Based on Games Apples Play by
Mark James Capella and Michael D. Weinstock

Learn programming the fun, enjoyable way . . . by gaming! Included is a vast selection of classic games for your Commodore 64 written in Commodore BASIC. Why make programming hard work?

GAMES
COMMODORES
PLAY

GAMES COMMODORES PLAY

by
Phillip W. Dennis, Ph.D.
and
Greg Minter

Based on Games Apples Play
by
Mark James Capella and Michael D. Weinstock

Cover Art and Illustrations by
Art Huff

 **DATAMOST**
INC. TE

20660 Nordhoff Street
Chatsworth, CA 91311-6152
(818) 709-1202



ISBN 0-88190-121-0

**Copyright © 1984 DATAMOST, Inc.
All Rights Reserved**

This book is published and copyrighted by DATAMOST Inc. All rights are reserved by DATAMOST Inc. Copying, duplicating, selling or otherwise distributing this product is hereby expressly forbidden except by prior written consent of DATAMOST Inc.

The word Commodore and the Commodore logo are registered trademarks of Commodore Business Machines.

Commodore Business Machines was not in any way involved in the writing or other preparation of this book, nor were the facts presented here reviewed for accuracy by that company. Use of the term Commodore should not be construed to represent any endorsement, official or otherwise, by Commodore Business Machines.

Printed in U.S.A.

ACKNOWLEDGMENTS

To my wife, Genie, for all those times she turned her head to see me sneaking up the stairs to the computer room.

Mark James Capella

To Diana and Aaron. Diane for her patience, understanding and the cold dinners she ate while waiting for me to finish just a little bit more. Aaron for the thrill of understanding what it means having a four year old around when things just aren't working correctly.

To Brian Davis for his creative artistic talents that greatly assisted our effort.

To Dave Gordon because he is Dave Gordon, a trusted friend.

Michael D. Weinstock

To both Mom and Dad, who believed in me even when I didn't.

Greg Minter

Table of Contents

Introduction — Using Games to Learn BASIC	9
---	---

Text Games

SCI-FI	23
Biorhythm	39
C-64 Learner	45
Digits	49
Grue Stew	53
IRSMAN	61
Kingdom	67
Reverser	73
Schmoo	77
Transition	81
Twinky	85
Word Scramble	91

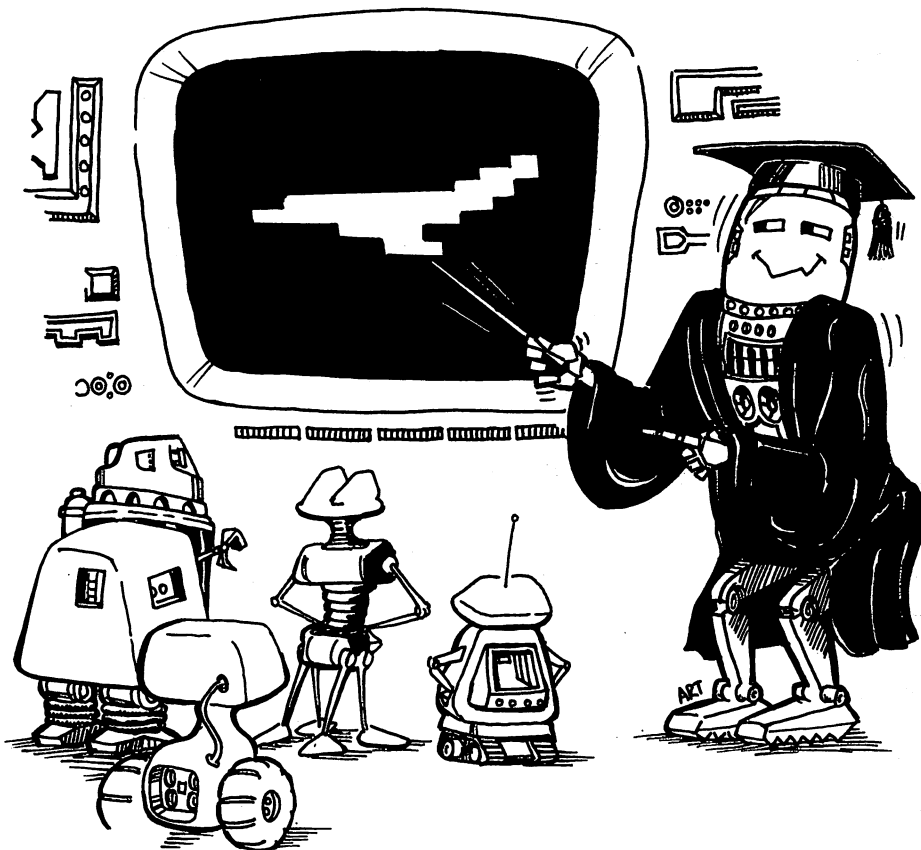
Graphic Action Games

MUBBLE CHASE	97
Air Attack	113
Artist Board	119
Barrel of Fun	123
Block 'em	129
Brain Teaser	133
Brick Wall	139
Connect Five	145
Craps	149
Dragon's Lair	155
Itche	161
Knock Out	167
Leaky Faucet	173
Magic Squares	177
Match the Key	185
Miniature Golf	191
Moving Target	203
Numbers Away	207
Point Attack	215
Robot Chase	219
Safe Cracker	223
Saucer Duels	229
Stardodger	235
Stranded	239
Target	243

INTRODUCTION

Using games to learn BASIC

The Commodore 64 is probably the most popular home computer today, and it's easy to understand why. It has more built-in memory than almost any other home computer on the market, fantastic sound and graphics capabilities, built-in BASIC, and one of the biggest software bases available for any brand of computer. The Commodore 64 is the direct descendant of some earlier Commodore computers, the PET and the VIC-20. Both of these machines were impressive for their time, but both had one major drawback: a small amount of memory for BASIC programs. The Commodore on the other hand, uses about 38K of its memory for BASIC workspace, which is large enough for almost any home computer application.



Most of the commercial software available for the Commodore 64 is written in machine language, which is easy for the computer to understand, but very confusing for the average human being. Back in the sixties and seventies, when the public was just becoming aware of computers, most of the programs were written in BASIC. Machine language programs were few and far between because people were somewhat intimidated by the appearance and complexity of the language. But there were a huge number of great BASIC games written. Programs weren't copy protected either — if somebody created a great game, you could almost bet that a line-by-line listing would appear somewhere soon, usually in one of the popular microcomputer magazines of the day.

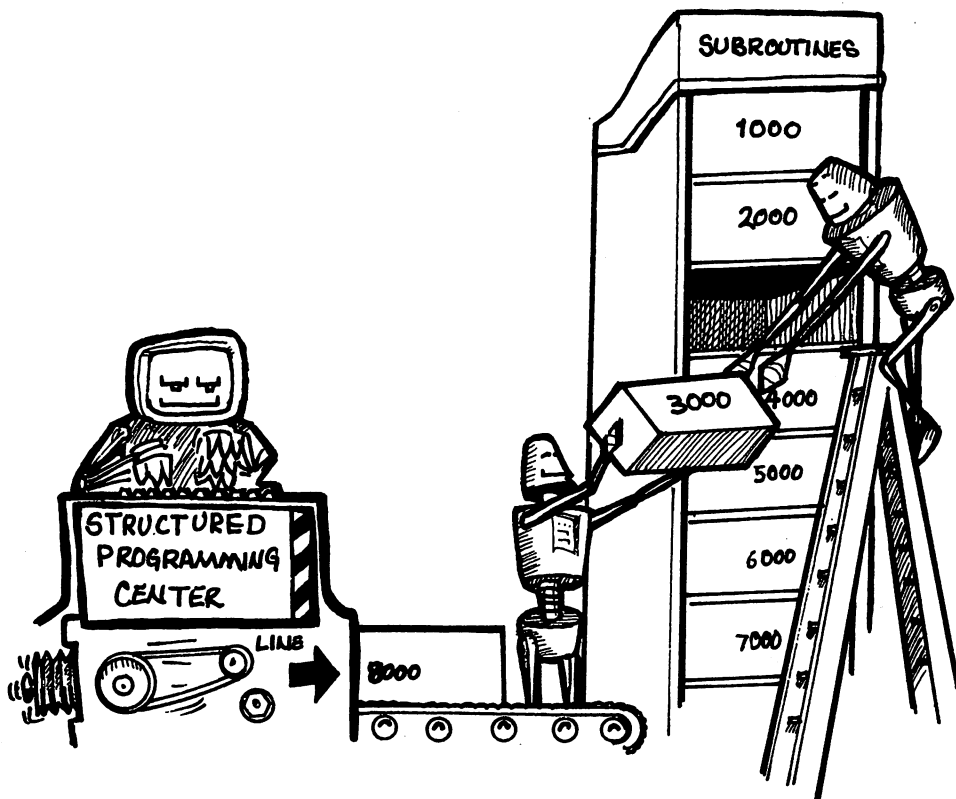
These games were often much slower than their machine language counterparts, but they also had several advantages. Programs created in BASIC on one machine were usually fairly "portable," which means that they were not difficult to transfer from a form that worked on one machine into a form usable by another. It was easy for a programmer to look at one of these listings and see how a game was created, and for him to modify it to fit his own ideas. Graphics were usually slow and not too complex, but the Commodore 64 has changed that quite a bit.

In this book is a collection of some classic BASIC games, written in a straightforward, easy to understand manner. Each of the programs is written using the same general structure, and any experienced programmer will tell you that it always makes it easier to follow the flow of a program if there is a definite structure to it. To compensate for BASIC's general lack of speed, many of the games are designed so that speed is not a factor. Adventure games such as GRUE STEW, word games such as WORD SCRAMBLE, and board games such as CONNECT FIVE all require quick wits instead of quick reactions — and after all, computers should stretch your mind, not your muscles.

Structuring your programs makes it easier to edit, change, and reuse sections of a program. Each game will be prefaced by a short description of some of the more interesting features of BASIC used by that game, which should make it easier to see how the program works. Two games, SCI-FI and MUBBLE CHASE, are accompanied by a line-by-line description of the function of each line. SCI-FI introduces text formatting, input routines, READ statements, and data manipulation. You'll find that these programming concepts are essential in the design of any word game, and learning from actual examples is definitely more beneficial than learning from a dry textbook. MUBBLE CHASE will help you understand some of the basic concepts of computer graphics, including animation, Cartesian coordinates (used to plot points and shapes), collision detection, and more.

All of the programs in this book use the same fundamental structure. Four GOSUB's control the flow of the program. Each target line contains a blank line or an empty REMark. Each subroutine also ends with a RETURN statement. Using this structured technique, you can write the basic structure before you write the inner workings of the program. Below you will find a complete program outline:

```
100 REM *** STRUCTURED PROGRAM ***
110 GOSUB 1000
120 GOSUB 2000
130 GOSUB 3000
140 GOSUB 4000
200 END
1000 REM *** INSTRUCTIONS ***
1990 RETURN
2000 REM *** SETUP ***
2990 RETURN
3000 REM *** PLAY ***
3990 RETURN
4000 REM *** END ***
4990 RETURN
```

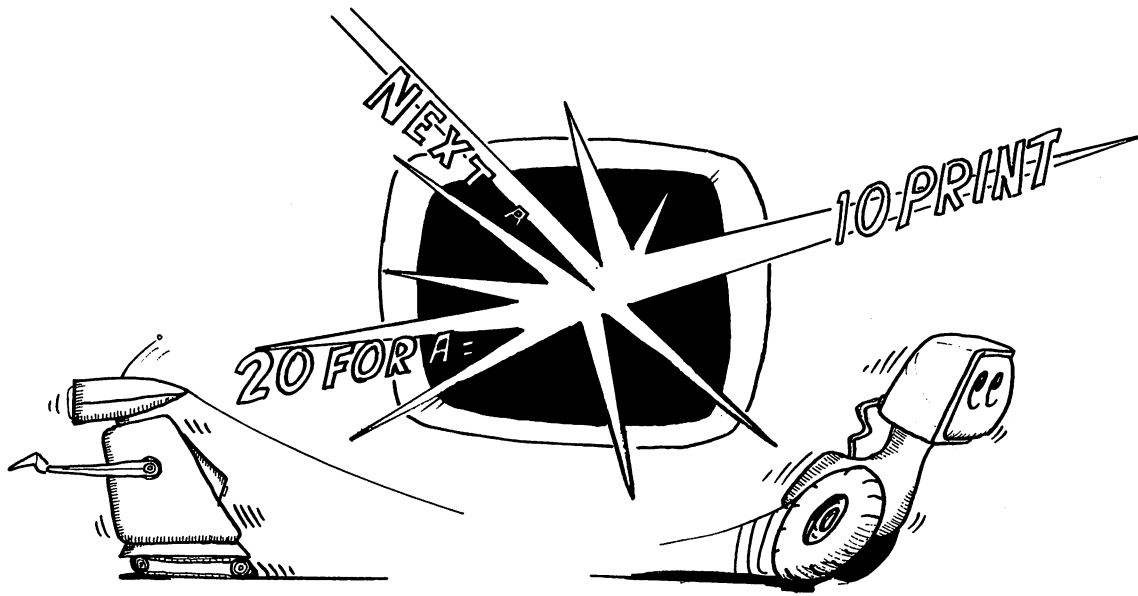


Using a structure like the one above actually makes it easier to write programs, since the programmer can see at a glance where certain programming tasks are being performed. If there is a problem with the way the instructions appear on the screen, the programmer knows that he should look somewhere between lines 1000 and 1990. Clearly demarcated program sections allow you to keep track of "what's going on where" even when programs get very long.

In the short prefaces, you will usually be given some suggestions on experiments to perform which will give you a better understanding of a particular facet of BASIC that was used in the program. You will undoubtedly learn a great deal about BASIC by typing in, studying, and debugging the program. You also have the option of ordering the programs on disk directly from DATAMOST in a ready-to-play state. Either way, you'll agree that this is a truly enjoyable way to learn BASIC!

Several tips may be in order before you start your experimentation:

- Save your game often. There's nothing more frustrating than typing in a program for an hour or two and then having someone disconnect your power supply by tripping over the plug. If you have to pause to think for a minute while entering a program, go ahead and save your program.
- Save your game in several versions. With the Commodore disk drive system in particular, saving a game with the recommended "@0:filename" method may cause you to lose a part of your program under certain conditions. Save your game as GAME.1, then GAME.2, and so forth, until you have a final version. Save this working version as GAME.WRK, and then use the SCRATCH command to remove the old versions from your disk.
- **EXPERIMENT, EXPERIMENT, EXPERIMENT.** Remember that anything inside your computer's memory is in a temporary state and that anything stored on an external storage device, such as a cassette tape recorder or disk drive, is in a permanent state. Once you have saved a program, you can load it back into your computer's memory and change it around as much as you want. These changes will only be made on your saved version if you save it on the same disk or tape under the same file name.



Background Color, Border Color, and Text Color

If you want to make any investment in your Commodore computer system, one of the first things you should consider is a color monitor. (A color TV will also work, but won't give as sharp a picture as the monitor.) Either way, you should keep in mind that the Commodore is really a color machine. A big part of the C-64's reputation is built on its graphic capabilities, so you can't really judge the machine if you're working with a black-and-white TV set or a monochrome monitor.

In many of the programs, you'll see the commands POKE and PEEK. The POKE command is used to place a numeric value into a certain memory location. The PEEK command is used to display the contents of a certain memory location to the screen. The Commodore uses POKES to control the color of the screen and the border, and keyboard commands to control the color of the text itself. Type in the following examples to get an idea of how this works:

```
POKE 53280,1 <RETURN>
```

The border around the screen will change to white.

```
POKE 53280,6 <RETURN>
```

Now the border is invisible. (Actually it is dark blue, the same color as the background color).

POKE 53280,0 <RETURN>

Now the border is black.

POKE 53280,14 <RETURN>

And now it is back to normal.

Now let's change the background color. We'll use the same command, a POKE, but this time we'll be putting the number in a different place:

POKE 53281,0 <RETURN>

Turns the background to black.

POKE 53280,13 <RETURN>

Now it is light green.

POKE 53280,1 <RETURN>

Now it should be white.


POKE 53280,6 <RETURN>

And now it is back to normal.

Before we go on a few things should be said about POKES in general, and about these two POKES in particular. POKES are entered in the form POKE mm,nn. The first number in a POKE (mm) represents a specific memory location. The second number (nn) is the value that is being placed in that location. POKE 53280,1 for example, tells the computer to put the value "1" in memory location 53280. POKEing a number into a specific memory location is often an easy way to get a lot of work done very quickly, but it is also one of the quickest ways to make your program "bomb" or "hang." In general, random POKEing is a risky activity. (It won't physically harm your computer if you POKE around in its memory, but it might ruin your program if you POKE a painful place). Any number in the range of 0-15 may be POKEd into locations 53280 or 53281, since there are 16 background and 16 border colors.

PEEKs on the other hand, can be used with no danger of harming your programs. A PEEK command lets you see what value is in a particular memory location. A PEEK has the form PEEK(nn), where nn is a number

between 1 and 65536 (the number of memory locations in your C-64). PEEKs do not change the contents of memory, they only show you what is at a specific memory location. If you type in PRINT PEEK(53280) when you turn your Commodore on, you will see a 14 appear on the screen, which is the code for light blue.

The color of the text on your screen can be changed merely by pressing different key combinations. When you turn your Commodore 64 on, the screen displays light blue letters on a dark blue screen, which makes for hard reading and squinty eyes. Try this: while holding down the Control key (marked CTRL on the keyboard) press the number 2. Now type in some text. Rather than hard-to-see blue, you have crisp white letters. Using the CTRL key OR the COMMODORE  key in combination with one of the number keys 1-8 will change the color of your text. 9 and 0 change the text to reverse video and back to normal again. Type in the following line:

```
PRINT "{CTRL-1}COMMODORE{CTRL-2}USES{CTRL-3}DIFFERENT  
{CTRL-4}COLORS"
```

{CTRL-#} means to hold down the CTRL key and then press a number while holding it down):

Press RETURN. You will see a multi-colored message.

With a little math, you can figure out that there are over 4000 color combinations that can appear on the screen — 4096 to be exact ($16 * 16 * 16$). With this many combinations, it should be quite a while before you run out.

Error Checking Your Program

The Commodore Operating System will detect a great number of errors, and although it may not seem that way, the messages it produces are designed to help, not infuriate you. Imagine an operating system where no error messages at all were displayed — if everything weren't just right, the program would hang or break, and you'd have no idea at all about what was going wrong. All those ?SYNTAX ERROR IN 10 messages actually make it a lot easier to debug a program than you probably realize. The Commodore BASIC error messages contain two important pieces of information: first, the type of error, and second, the line number at which it occurred. Both parts of these messages are important: imagine if the error message only said ?SYNTAX ERROR. You would have to work through the entire program to

find a single incorrect entry. Or imagine that you got only a line number with no description, like ?ERROR IN 20. You would have to decide whether it was a syntax error, a redimensioned array error, or whatever. All in all, your Commodore 64 is as specific as it can be about what you have to do to your program to make it run.

Use STOP

Put in a STOP command immediately following a line that you don't understand. By seeing what has and what has not been done, you may be able to understand the line's function. If not, put a STOP before the line. Perhaps the absence of a certain attribute will clarify the function of the mystery line.

Detour Lines

If you would like to know what effect skipping a certain line has, put a GOTO right before it. For example, if line 40 were in question, you might type:

```
10 ---  
20 ---  
30 ---  
35 GOTO 50  
40 REM SKIP THIS MYSTERY LINE  
50 ---
```

Program flow would skip right over line 40 to line 50.

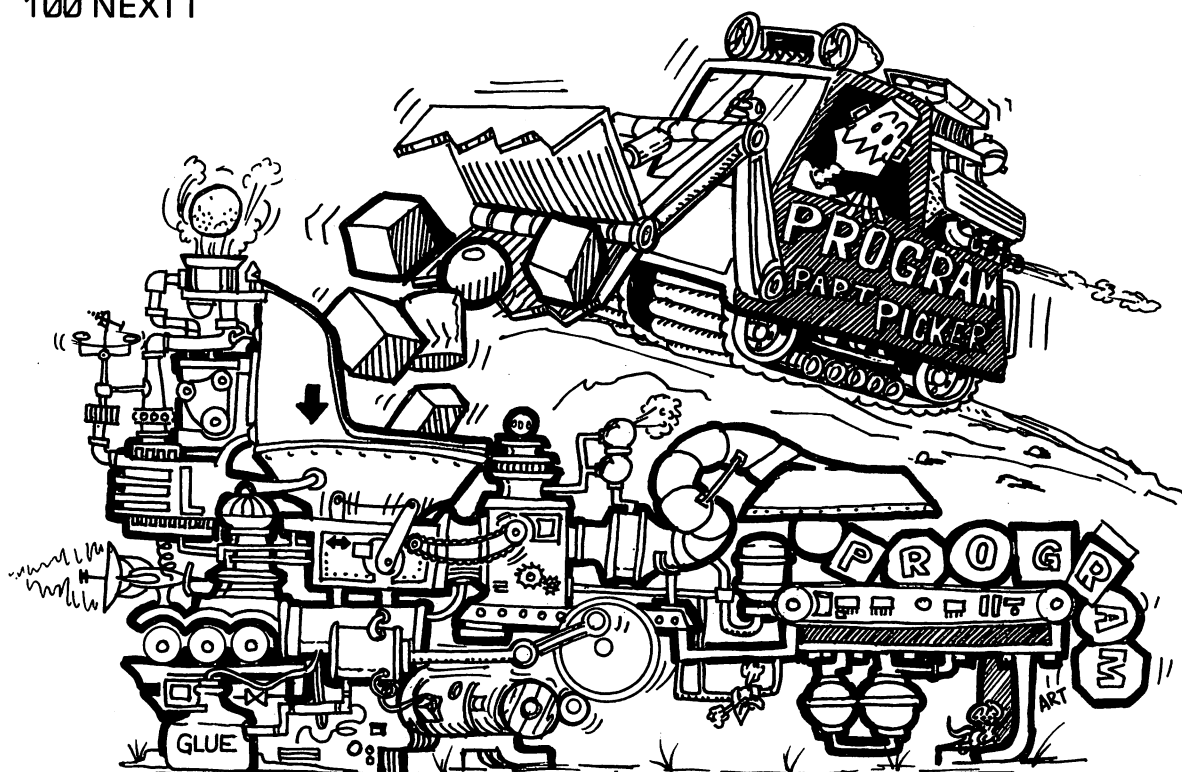
Delete Lines

Rather than using the detour method, you can simply delete a line. This is done by typing in the line number in question and then pressing RETURN. After deleting a line, run the program and make note of any changes that occur. Make sure to reenter the deleted line next time you want to run the program, or load in an unaltered version from cassette or disk.

Isolate Parts of the Program

If you want to run a certain segment of the program, there are three ways to do it. First, you can delete all but the lines in question. This, however, may prove more tedious than expected. Commodore BASIC requires that you delete lines one at a time, and deleting one or two hundred lines to discover the function of three lines is not very efficient. Second, you can add a line at the beginning of the program which directs flow to a specific line, e.g., 10 GOTO 2230 (or whichever line or section is in question). Your program will eventually stop with a ?RETURN WITHOUT GOSUB error, but you will have been able to see the function of the line or section. For the third method, we will use an example. Assume that in the program below, you wanted to isolate lines 40-60:

```
10 PRINT CHR$(147)
20 REM SAMPLE PROGRAM
30 INPUT "WHAT IS YOUR NAME?";N$
40 PRINT "YOUR NAME IS ";N$
50 INPUT "WHAT IS YOUR FAVORITE COLOR?";C$
60 PRINT "GREEN IS MY FAVORITE."
70 PRINT CHR$(147)
80 FOR I= 1 TO 16
90 PRINT N$;C$;
100 NEXT I
```



Copy and run the program above. To isolate lines 40-60, first add line 61 by typing 61 STOP. Now type RUN 40. The computer will start execution at line 40 and STOP at line 61. This is the best method of the three to see what's going on, since the program will stop right at the end of the section you want to see (when it hits the STOP command). After you have seen the results, type in CONT to get the program going again.

After you've experimented for a while, you'll get the hang of writing and debugging your own programs. Learning BASIC with games is both educational and fun, and you will begin to build up a software library of working programs at the same time. So get to it!

For Your Information

Many of the games in this book use a machine language routine to do graphic plotting. Look at the section below, lines 2020-2026 from the program "Leaky Faucet:"

```
2020 PLOT = 49152 : FOR I = 0 TO 24 : READ C : POKE PLOT + I,C
: NEXT I
2021 DATA 32,16,192,132
2022 DATA 253,32,16,192
2023 DATA 152,170,164,253
2024 DATA 24,76,240,255
2025 DATA 32,253,174,32
2026 DATA 158,173,76,170,177
```

In this example, the twenty five data elements are POKEd into twenty five consecutive memory locations, starting with location 49152. If we look at line 3015, we can see an example of this routine being called:

```
3015 SYS PLOT,2,24 .....
```

The first number is the x-coordinate of a point to be plotted on the screen (the column), the second, the y-coordinate (the row). The above command could also have been written like this:

```
3015 SYS49152,2,24
```

The variable PLOT is set to the beginning memory location of the plot routine in line 2020 (PLOT = 49152). Using SYS PLOT instead of SYS49152, however, gives us more information about exactly what machine language routine we are calling. When a SYS PLOT is used, we know that a plot is being done.

If you're interested in machine language, or are already familiar with it, we have included the source code for the plot routine below. The data statements in the BASIC program above are derived from this routine, and both do the same thing to your computer. It's not necessary to understand how this conversion is done, but you might want to take a look at the listing below just to see how machine code is entered:

```

0000-      ; MACHINE LANGUAGE "PLOT" ROUTINE
0000-      ; FOR GAMES COMMODORE'S PLAY
0000-      ; BY PHILLIP W. DENNIS, PHD
0000-      ;
0000-      ; PAGE ZERO
0000-      ;
0000-      TEMP=$FD          ; TEMP (FREE PAGE 0 LOC)
0000-      ;
0000-      ; ROM ENTRY POINTS
0000-      ;
0000-      CHKCOM=$AEFD     ; BASIC CHECK FOR COMMA
0000-      IFIX=$B1AA      ; CONVERT # TO INTEGER
0000-      EVAL=$AD9E      ; EVALUATE EXPRESSION
0000-      PLOT=$FFF0      ; KERNAL PLOT ROUTINE
0000-      ;
0000-      *=$C000         ; SET ORG TO 49152
C000-      ;
C000-      20 10 C0        JSR GETVAL   ; GET COLUMN VALUE
C003-      84 FD          STY TEMP     ; SAVE IT
C005-      20 10 C0        JSR GETVAL   ; GET ROW VALUE
C008-      98            TYA
C009-      AA            TAX           ; PUT ROW IN X
C00A-      A4 FD          LDY TEMP     ; PUT COLUMN IN Y
C00C-      18            CLC           ; CLR CARRY TO INDICATE SET
C00D-      ;              ; CURSOR FOR KERNAL PLOT
C00D-      4C F0 FF        JMP PLOT     ; EXIT VIA PLOT
C010-      ;
C010-      GETVAL
C010-      ;
C010-      ; PARSE VALUE AFTER A COMMA
C010-      ; WILL PRINT SYNTAX ERROR IF COMMA
C010-      ; IS MISSING AND WILL ABORT PROGRAM.
C010-      ;
C010-      20 FD AE        JSR CHKCOM   ; CHECK FOR A COMMA
C013-      20 9E AD        JSR EVAL    ; EVALUATE EXP FOR COLUMN
C016-      4C AA B1        JMP IFIX    ; CONVERT TO INTERGER
C019-      .END

```

Key to the Listings

Within the program listings, you will come across instructions enclosed in braces, like {BLK} or {WHT}. Here is how these special characters should be entered from the keyboard:

The CTRL (ConTRol) key used in combination with other keys produces some of the special characters:

To Get	Press these keys
{BLK}	Hold down the CTRL key and press the 1 key.
{WHT}	Hold down the CTRL key and press the 2 key.
{RED}	Hold down the CTRL key and press the 3 key.
{CYN}	Hold down the CTRL key and press the 4 key.
{PUR}	Hold down the CTRL key and press the 5 key.
{GRN}	Hold down the CTRL key and press the 6 key.
{BLU}	Hold down the CTRL key and press the 7 key.
{YEL}	Hold down the CTRL key and press the 8 key.
{RVS ON}	Hold down the CTRL key and press the 9 key.
{RVS OFF}	Hold down the CTRL key and press the 0 key.

The Commodore key (in combination with other keys) also produces some of the special characters. This key is located at the bottom left-hand corner of the keyboard:

To Get	Press these keys
{ORNG}	Hold down the Commodore key and press the 1 key.
{BRWN}	Hold down the Commodore key and press the 2 key.
{LT RED}	Hold down the Commodore key and press the 3 key.
{GRY 1}	Hold down the Commodore key and press the 4 key.
{GRY 2}	Hold down the Commodore key and press the 5 key.
{LT GRN}	Hold down the Commodore key and press the 6 key.
{LT BLU}	Hold down the Commodore key and press the 7 key.
{GRY 3}	Hold down the Commodore key and press the 8 key.

The functions key are listed in the following manner:

To Get	Press these keys
{F1}	Press the f1 key.
{F2}	Press the f2 key.
{F3}	Press the f3 key.
{F4}	Press the f4 key.
{F5}	Hold down the shift key and press the f1 key.
{F6}	Hold down the shift key and press the f2 key.
{F7}	Hold down the shift key and press the f3 key.
{F8}	Hold down the shift key and press the f4 key.

These characters will also appear:

To Get	Press these keys
{CLR}	Hold down the shift key and press the CLR/HOME key.
{HOME}	Press the CLR/HOME key.
{CSU}	Hold down the shift key and press the cursor-up/cursor-down key.
{CSD}	Press the cursor-up/cursor-down key.
{CSL}	Hold down the shift key and press the cursor-left/cursor-right key.
{CSR}	Press the cursor-left/cursor-right key.

In addition to these characters, actual graphic characters may appear in game listings. If this is the case, then a special section at the beginning of the listing will describe how to produce any such characters.



This program is not a game, rather it is an entertaining collection of short stories which you can make as personal as you want. You are given the chance to enter your own data. As with MUBBLE CHASE in the graphics programs section, this program will be described line by line.

10 REM stands for REMark. Any comments, numbers, symbols, expletives, etc. may follow a REM. In this case, the REMark is used to help highlight the program name. The computer, in effect, ignores the material which follows a REM. In this case, the REMark ***** is made.

11-16 completes the title with REM statements including two blank lines for readability.

20 The entire program is controlled by lines 20-50. There are actually two parts to line 20. First, GOSUB 1000 instructs the computer to go to line 1000, and to continue from there until the command RETURN is encountered. The second instruction, REM, is allowed on the same line for only one reason. The colon (:) allows the programmer to include many, not necessarily related, instructions on the same program line. This is usually done to show that certain pieces of a program are so closely related, that to isolate them on separate lines would belie their common purpose or association. The REMark: INSTS stands for INSTRUCTIONS. This is the programmer's way of telling the reader that the subroutine beginning at line 1000 (GOSUB 1000) is where the INSTRUCTIONS can be found. In this case, the REMark serves to explain the purpose of the subroutine rather than serving the equally important, but more aesthetic, role of lines 10 through line 16.

30 When the subroutine begun by line 20 is completed by the RETURN statement, the program, having completed line 20, drops down to the next sequential instruction, which in this case is line 30. Like line 20, this line first initiates a subroutine (GOSUB 2000), and then informs the reader as to the main emphasis of the subroutine, which is to SETUP the program.

40 This line is identical in function to line 30, except the subroutine it accesses is the PLAY subroutine.

50 END returns you to the immediate mode. Control no longer belongs to the program, but to the programmer.

1000 & 1020 Colons can also be used to create spaces in program listings. They are not the same as REMs, as the computer checks to see if there is any information after the colons. When a program encounters a REM, it automatically drops to the next line number.

1010 This line just indicates what is taking place in the subroutine. In this case, it is INSTRUCTIONS.

1030 POKE 53280,6 and POKE 53281,6 set the border and background screen color respectively. Try changing the values after the commas to some other number in the range 1-15 to get a different perspective on things.

1040 The Commodore 64 allows you to create strings that consist of cursor movements, color keys, and so forth. VT\$ in our case is defined as a {HOME} followed by a number of {CRSR-DWN} movements. So if we print out the first part of VT\$, it would be the same as pressing the CLR/HOME key, and then the CRSR-DWN key several times. We can use this string to position the cursor on a particular row on the screen. Look at the commentary on line 1800 for more detail.

1050 Here we have an example of a nested subroutine. GOSUB 1900 goes to a subroutine that positions and prints a title at the top of the page, then RETURNS to the next line.

1060-1110 These lines will print out exactly what appears between the quotation marks to the screen. Line 1080 uses a PRINT statement followed by no text at all (as the second statement in the line) to put a line space between two paragraphs on the screen.

1800 This line makes use of the string of special characters in line 1040. The command PRINT LEFT\$(VT\$,24) will print 24 characters of the string VT\$, starting with the leftmost character. Since the first character in VT\$ is a HOME, the cursor is placed on the first row in the upper lefthand corner. This is followed by 23 cursor-down moves, which place the cursor near the bottom of the screen. Note that the RETURN is printed out in inverse video so that it stands out.

1810 Waits for the user to press a key to indicate that he has finished reading the screen. If no key is being pressed, the second part of the line will send control back to the beginning of the same line until a key has been pressed.

1890 This RETURN actually sends control of the program back to line 30 at the top of the program, since the line that sent program control to the INSTRUCTIONS routine is still waiting to encounter a RETURN.

1900 This line prints the title "*** SCI-FI ***" across the top of the page whenever it is necessary. Making a job like this a subroutine can save the fingers a lot of typing. Whenever you need to print the title, for example, instead of typing out the whole thing again, you can just use GOSUB 1900. This type of structure has two advantages: first, as mentioned above, you don't have to type as much data in, and secondly, you are assured of getting the same appearance on the screen every time.

2000 & 2002 As stated before, a blank colon (:) is a legitimate means of writing a virtually blank line within the program listing itself.

2001 On this line, the main function of the subroutine is detailed by a REM statement.

2010 This line sets aside sixteen memory locations for the array SO\$. The DIM instruction tells the computer to DIMension a block of memory large enough for sixteen separate values for SO\$ (SO\$(0) through SO\$(15)). Also, the value of SO is set to 0. The number in parentheses is the number of the array variable. The Commodore (and most other microcomputers) start from (0).

2015 This line sets up a one-dimensional array in memory. This table can accommodate up to sixteen separate values of PL\$. Also, PL is set to 0.

2020 & 2025 Are duplicates of lines 2010 and 2015 (but the arrays being DIMensioned are different).

2030 The READ statement is an interesting animal. What it does is find the first available DATA statement and read the first element in it. In this program, the first DATA statement is in line 2050. What happens is this: the

first piece of data (each piece is separated by a comma), ALEXANDER HAIG, is read into (stored at) SO\$. Then a test is done to see if SO\$ is equal to "END". If so, then the program falls through to line 2035. If not, SO is incremented ($SO = SO + 1$), SO\$(1) is assigned the value of the contents of SO\$ (ALEXANDER HAIG), and the process is repeated (GOTO 2030). The program will next READ the second piece of DATA (the Commodore automatically keeps track of which piece of data is next in line), which is RONALD REAGAN, and store it in SO\$(2). After the first seven pieces of DATA on line 2050 have been read, the last DATA element will be encountered. This time the test on line 2030 ($IF\ SO\$ \neq "END"$) will be false, so control will drop through to line 2035. When you understand that SO, SO\$, and SO\$(0) are all different variables you get a gold star.

2035 Once a DATA element has been read, it is no longer "available." Therefore, the first available DATA is not in line 2050, but 2060. It may occur to you that PL stands for PLace. Actually, all of the variables are representative of their meaning.

2040 & 2045 Both of these lines are identical in function to line 2030.

2050 You must make sure that you have the right kind of variable to hold the kind of DATA you are working with. If, in line 2030, you changed all occurrences of SO\$ to SO, then ran the program, you would get a ?SYNTAX ERROR IN 2050 message. If you want to read in special characters and numbers, you must use a string variable (one followed by a dollar sign) to hold it. Variables without a dollar sign can only hold numerical information (including decimal points).

2055-2090 These are all DATA statements. All of the data could have been combined into several long DATA statements. The reason for dividing them into smaller lines was to add clarity.

2095 This line appears to be a nebulous conglomeration of variables. Not so! This line serves a very specific purpose. DEF stands for DEFine. The next question is, what is to be defined? The answer is a FuNction (FN). The FuNction being DEFined is R(X). "R" is the given name of the function. "X" is a variable name which is equal to the $FIRST\ INT(RND(1) * X) + 1$. Each time the FuNction "R (any variable or digit)" is used, the variable inside the parentheses assumes the value of " $INT(RND(1) * (the\ new\ variable\ or\ digit)) + 1$ ". Below is a list which, hopefully, will help you to understand the "DEF FN" statement.

FUNCTION	VALUE OF VARIABLE OR DIGIT	VALUE OF FUNCTION
A(X)=4*X+5	'X' IF NOT STATED, = 0	A(X)=5 (4*0+5)
A(13)	13 = 13	A(13)=57 (4*13+5)
A(Y)	LET US SAY THAT Y = 6	A(Y)=29 (4*6+5)
BS(X)=X*X-22	X = 0	BS(X)=-22 (0*0-22)
BS(17)	17 = 17	BS(17)=267 (17*17-22)
BS(FN A(Y))	FN A(Y) = 29	BS(FN A(Y)) = 819
A(FN BS(17))	FN BS(17) = 267	A(FN BS(17))=1073 (267*4+5)

Here is a program to further illuminate the function of line 2005:

THE PROGRAM

```

10 A = 17 : C = 2.65
20 DEF FN PRY(C) = -A * C
30 PRINT C;: PRINT FN PRY(C)
40 PRINT A;: PRINT FN PRY(A)
50 PRINT FN PRY(FN PRY(C))

```

THE OUTPUT

```

----
----
2.65 -45.05
17 -289
765.85

```

In this little program, line 20 identifies the variable as being "C." Therefore, whenever the function "PRY" is executed, the variable (or digit) within the parentheses is substituted for C. In line 2095, "X" is the variable. If the number five (5) is substituted for X, then the result is $R(5) = \text{INT}(\text{RND}(1) * 5 + 1)$. The reason the DEF FN instruction is used in this program is so that whenever a random number between 1 and any other number is needed, all the programmer needs to write is FN R (any other number), and the random result will be generated.

2100 This sends control to the title subroutine.

2110 This line prints part of the string VT\$ (homing the cursor and moving it down seven spaces).

2120 Prints the rest of the message.

2130 Sets the value of CO to 0.

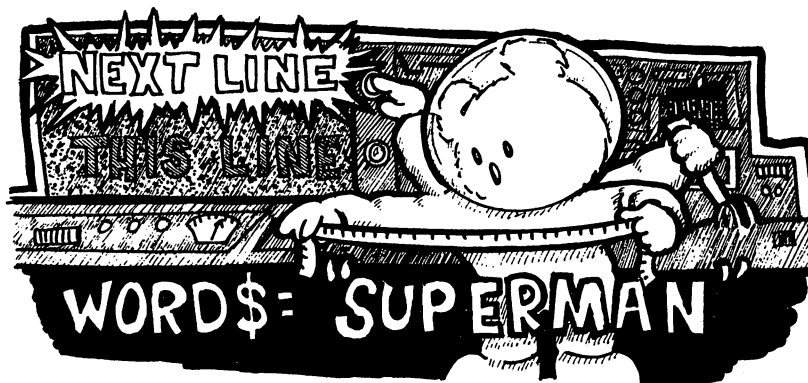
2140-2450 These lines continue to INPUT names of Places that will be Attacked, Names of Monsters, etc.

2900 This sets the value of PT to one (1) each time the subroutine (lines 2900-2999) is executed.

2910 MID\$ states that the computer will look at the MIDDLE of a given word. Which word? All of the remaining pertinent information is in the parentheses. The word or words that are to be checked by MID\$ are contained in WRD\$. Beginning at the lefthand side of WRD\$, the computer will begin the scrutiny at that character plus PT. If WRD\$ contained the phrase "I Love You", and PT equals six, then MID\$(WRD\$,PT) would instruct the computer to begin looking at "I Love You" six characters over from the leftmost character (the "e" in "Love" is the SIXth character). If not specified, the computer will begin the search at the designated location, and continue through to the end of the words. If you want a certain number of characters looked at, just specify that number after you tell the computer where to start. The following program should help:

PROGRAM	OUTPUT
10 A\$ = "SEND MONEY"	--
20 PRINT MID\$(A\$,4,5)	D MON
30 PRINT MID\$(A\$,1,6)	SEND M

In the little program above, the computer is told to search the MIDDLE of A\$, to begin 4 characters from the left and to PRINT the next 5 characters. Likewise, line 30 tells the computer to PRINT the contents of A\$ beginning with character 1, and to continue for a total of 6 characters. You will admit that at the end of most words you will find a blank space. The test (IF MID\$(WRD\$,PT,1) = " ") checks to see if the character at position PT is a blank space () and it checks to see if its line position is greater than 30. The POS(0) command checks to see if the cursor is at a column on the screen whose value is greater than 30 (a regular screen containing 40 columns, numbered 0-39). If the value is greater than 30, the next word is in danger of going off the right side of the screen. To prevent this from happening, an affirmative test result will force the computer to skip to the next line before continuing to print.



2920 There are two steps to this line. First, PT is incremented ($PT = PT + 1$). Second, the size of PT is compared to the character LENGTH of WRD\$. For example, if WRD\$ contained the phrase EARTH WAS ATTACKED, then the LENGTH of WRD\$ would equal the number of characters in that phrase (18). To prevent lines 2910 and 2920 from looping indefinitely, there must be some way of stopping the loop. This is the function of the second half of line 2920. When PT is larger than the length of WRD\$, the program will fall through to line 2990.

2990 This line completes the subroutine started at line 2900.

3000 This is the target of the GOSUB in line 40.

3001 The REMark ***PLAY is used to inform the reader of the main function of the subroutine at line 3000.

3002 A colon used for spacing.

3010-3065 These line instruct the computer to perform various subroutines and these subroutines, in turn, determine which format the story will reflect.

3070 A colon used for spacing.

3080-3085 The computer waits for you to indicate that you have finished reading the story by pressing RETURN. You could change this line to allow the reader to escape from further Sci-Fi literature:

```
3085 PRINT LEFT$(VT$,24): INPUT "HAD ENOUGH";ANS$  
3086 IF LEFT$(ANS$,1) = "Y" THEN END
```

Using the LEFT\$ function to find the first letter of "yes" is standard practice, and most computer enthusiasts are used to answering "Y" and expecting it to work. "Yo," "Yea," or "Yes" will all work. Any response whose first letter is not "Y" will end the program. Instead of END, you could branch to a line at the end of the listings and add any farewell sequence you wished.

3100 This line first clears the screen of text with a (CLR/HOME). Then the function defined at line 2005 yields a RaNDom number between one and five. If the integer generated is one, then the program will branch to 3110. If FN R(5) generates the integer "2" then the program branches to line 3120. If FN R(5) yields 3, 4, or 5, then the program will branch to 3130, 3140, or 3150 respectively.

3110-3150 These lines PRINT the five different attention-getting headlines.

3200 Line 2035 counted the total number of PLaces saved in PL\$. FN R(PL) will yield a RaNDom integer ranging from one to PL. There is the name of a different place saved in PL\$(1), PL\$(2), PL\$(3) . . . PL\$(PL). What this line does is randomly select one of the many PLaces and copy it into the location WRD\$. Then the subroutine beginning at line 2900 checks to make sure that the PLace is not PRINTEd in such a manner that it breaches the right margin.

3290 Control is returned to line 3015, then drops down.

3300 Like line 3100, this one selects a randomly generated integer ranging from one to five FN R(5), and depending on the integer, branches to line 3310, 3320, 3330, or 3340.

3310-3350 These lines serve as continuations of the currently unfolding drama. One of the five rather repugnant actions is stored in the variable WRD\$.

3400 The variable WRD\$ already contains one of the five actions outlined on lines 3310-3350. These lines add the name of one of the many (MO) monsters (MO\$) to WRD\$. Line 2350 counted the number of monsters and stored the name of each one in an MO\$ location. This line RaNDomly selects one of the MONsters and adds its name, plus a trailing blank space (" ") to WRD\$. Each time the subroutine at 2900 is performed, the contents of WRD\$ are PRINTEd and cleared from WRD\$. Each time WRD\$ is emptied, the story grows.

3490 Control returns to the top of the routine.

3500 First, the word "FROM" and a trailing space (" ") are stored in WRD\$. This word is added to the developing story. At the beginning of the subroutine at line 2900, the contents of WRD\$ (FROM) are added to the rapidly developing story.

3510 The HOme base of the MONsters is added to WRD\$, and a period (".") is also added. Once again WRD\$ is emptied (in the subroutine at 2900), and the story sprouts another section.

3590 Control returns to the top of the routine.

3600 From the list of potential heroes, SOmeone will be RaNDomly selected (FN R(SO)). A space (" ") will follow SOmeone's name, and then this material will be added to the story.

3690 Control returns to the top of the routine.

3700 The words "TRIED TO" are added to the text.

3705 One of five "defense methods" is randomly selected (FN R(5)) and added to WRD\$ (at 3710, 3720, 3730, 3740, or 3750). Then the subroutine at line 2900 adds the 'defense method' to the story.

3710-3750 These lines contain the five "defense techniques."

3800 Two more words and a trailing space (BUT THEY) are added to the text.

3805 One of the five responses to SOmeone's "defense technique" is RaNDomly chosen and the text is once again supplemented.

3810-3850 Each line contains one of the five responses to a "defense technique."

3900 There are five ways to announce the changing fortunes of battle. One of these five choices is randomly selected (FN R(5)) GOSUB . . .). The subroutine at line 2900 again prevents overflow on the righthand margin while adding the new material to the story.

3910-3950 These lines contain the five "fortune-reversal lead-ins" mentioned at line 3900.

3090 Sends the program back to the beginning of the routine for another story.

4000 The ON/GOSUB structure deserves explanation. Line 2095 defined the FuNction R(X) to be equal to a RaNDom INTegeR between one and X. The number five is substituted for "X", making R(5) yield a number between one and five. The "ON" statement will cause the program to execute one of the five given subroutines, depending on the value of FN R(5). If the number generated is 3, then line 4000 reads, in effect, ON 3 GOSUB 4030 (the third line number). After one of the five subroutines is performed, the program executes the subroutine beginning at line 2900. Then RETURN completes the subroutine initiated at line 3060.

4010-4050 These five lines contain the five different "attack methods" that SOmeone can employ. The information is stored in WRD\$ until it is PRINTEd (by line 2900).

4100 A two word phrase, "SO THEY ", is stored in WRD\$ for later addition to the story. Next, the subroutine beginning at line 2900 is executed. This

subroutine not only PRINTs the contents of WRD\$, but prevents the contents of WRD\$ from being printed so that they breach the right margin. After one of five "attack results" is added to WRD\$, the subroutine which begins at line 2900 is executed. For all intents and purposes, the RETURN statement at line 4100 completes the program.

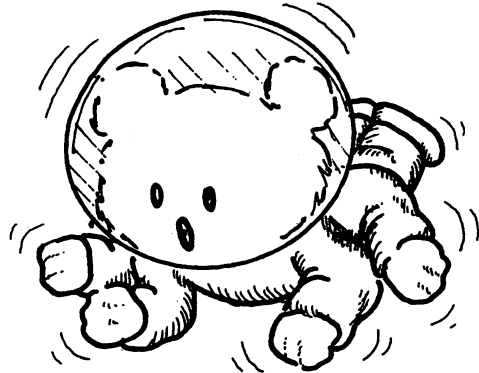
4110-4150 These five line contain the five "result" of the "attack method."

Even as you read these words some of the illustrious characters whose names are stored in the data statements are slipping into obscurity. Put in your own selection of names. This program is really an easy one to change and make your own. Change the data statements and print commands to your own fiendish specifications. You can save the new version under a new name and have both versions on diskette. Remember! What you change in the computer's memory does not change what is stored on the diskette unless you save the new version and give it the same name as the old version. It is sound practice to leave the original version unchanged on a different diskette and call your version SCI-FI V1, SCI-FI V2, etc. Have fun changing things around. It's the best way to learn.

```

10 REM *****
11 REM ***          ***
12 REM ***   SCI-FI   ***
13 REM ***          ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 END
1000 :
1010 REM *** INSTRUCTIONS ***
1020 :
1030 POKE 53280,6:POKE 53281,6
1040 VT$="{HOME}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}"
1050 GOSUB 1900
1060 PRINT "THIS PROGRAM WILL PRODUCE LO
TS OF FUNNY"
1070 PRINT "LITTLE SCIENCE-FICTION STORI
ES FOR YOUR"
1080 PRINT "READING PLEASURE.":PRINT
1090 PRINT "YOU ARE GIVEN THE CHANCE TO
ENTER SOME"
1100 PRINT "PERSONALLY RELEVANT INFORMAT
ION IN HOPES";
1110 PRINT "OF MAKING THE STORIES MORE I
NDIVIDUAL."
1800 PRINT LEFT$(VT$,24);"PRESS {RVS ON}
RETURN{RVS OFF} WHEN READY TO CONTINUE:
{RVS ON} "
1810 GET AN$:IF AN$="" THEN 1810
1890 RETURN
1900 PRINT "{CLR}{WHT}{CSD}{CSD}";TAB(13
);"*** SCI-FI ***":PRINT:PRINT
1990 RETURN
2000 :
2001 REM *** SETUP ***
2002 :
2010 DIM SO$(15):SO=0
2015 DIM PL$(15):PL=0
2020 DIM MO$(15):MO=0
2025 DIM HO$(15):HO=0
2030 READ SO$:IF SO$<>"END" THEN SO=SO+1

```



```

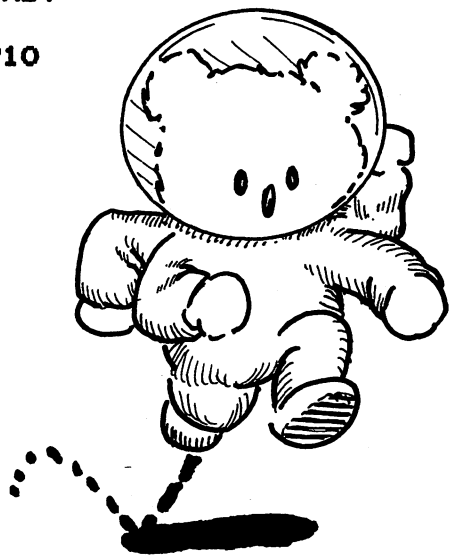
:SO$(SO)=SO$:GOTO 2030
2035 READ PL$:IF PL$<>"END" THEN PL=PL+1
:PL$(PL)=PL$:GOTO 2035
2040 READ MO$:IF MO$<>"END" THEN MO=MO+1
:MO$(MO)=MO$:GOTO 2040
2045 READ HO$:IF HO$<>"END" THEN HO=HO+1
:HO$(HO)=HO$:GOTO 2045
2050 DATA ALEXANDER HAIG, RONALD REAGAN, S
UPER CHICKEN, FATHER GUIDO SARDUCCI
2055 DATA A LITTLE GIRL, AN INTERESTED ON
LOOKER, SUPERMAN, "END"
2060 DATA SYRACUSE, NEW YORK, EARTH, THE UN
ITED STATES, YOUR TOWN, LOS ANGELES
2065 DATA WALLA-WALLA WASHINGTON, THE PRE
SIDENT, THE EASTERN COAST, "END"
2070 DATA LITTLE GREEN MEN, A CROWD OF AN
GRY PEASANTS BEARING TORCHES, BOOGEY MEN
2075 DATA ALIENS, IN-LAWS, SPACE EGGS, FLYI
NG GOOKIES, ZOMBIES, RELIGIOUS FANATICS
2080 DATA ICKY THINGS, "END"
2085 DATA VENUS, MARS, OUTER SPACE, OUT OF
OUR GALAXY, THE MOON, THE 4TH DIMENSION
2090 DATA THE NEGATIVE ZONE, A TIME WARP,
THE STARS, PLUTO, "END"
2095 DEF FN R(X)=INT(RND(1)*X)+1
2100 GOSUB 1900
2110 PRINT LEFT$(VT$,8);"TYPE UP TO 5 NA
MES OF PEOPLE THAT WILL"
2120 PRINT "SAVE THE DAY:":PRINT
2130 CO=0
2140 INPUT "===> ";SO$:IF SO$="END" THEN
2200
2150 SO=SO+1:SO$(SO)=SO$:CO=CO+1:IF CO<5
THEN 2140
2200 GOSUB 1900
2210 PRINT LEFT$(VT$,8);"TYPE UP TO 5 NA
MES OF PLACES THAT WILL"
2220 PRINT "BE ATTACKED:":PRINT
2230 CO=0
2240 INPUT "===> ";PL$:IF PL$="END" THEN
2300
2250 PL=PL+1:PL$(PL)=PL$:CO=CO+1:IF CO<5
THEN 2240
2300 GOSUB 1900
2310 PRINT LEFT$(VT$,8);"TYPE UP TO 5 NA
MES OF MONSTERS THAT WILL";
2320 PRINT "ATTACK:":PRINT
2330 CO=0
2340 INPUT "===> ";MO$:IF MO$="END" THEN
2400

```

```

2350 MO=MO+1:MO$(MO)=MO$:CO=CO+1:IF CO<5
  THEN 2340
2400 GOSUB 1900
2410 PRINT LEFT$(VT$,8);"TYPE UP TO 5 NA
MES OF PLACES THAT"
2420 PRINT "MONSTERS COME FROM:":PRINT
2430 CO=0
2440 INPUT "===> ";HO$:IF HO$="END" THEN
  2500
2450 HO=HO+1:HO$(HO)=HO$:CO=CO+1:IF CO<5
  THEN 2440
2500 RETURN
2900 PT=1
2910 PRINT MID$(WRD$,PT,1);:IF MID$(WRD$
,PT,1)=" " AND POS(0)>30 THEN PRINT
2920 PT=PT+1:IF PT<=LEN(WRD$) THEN 2910
2990 RETURN
3000 :
3001 REM *** PLAY ***
3002 :
3010 GOSUB 3100:REM TITLE
3015 GOSUB 3200:REM PLACE
3020 GOSUB 3300:REM ACTION
3025 GOSUB 3400:REM MONSTER
3030 GOSUB 3500:REM PLACE
3035 GOSUB 3600:REM SOMEONE
3040 GOSUB 3700:REM DEFEND
3045 GOSUB 3800:REM TOO TOUGH
3050 GOSUB 3900:REM FINALLY
3055 GOSUB 3600:REM SOMEONE
3060 GOSUB 4000:REM DEFEND
3065 GOSUB 4100:REM THEY DIED
3070 :
3080 PRINT LEFT$(VT$,24);"PRESS ANY KEY
TO CONTINUE"
3085 GET AN$:IF AN$="" THEN 3085
3090 GOTO 3010
3100 PRINT "{CLR}{CSD}{CSD}"
3105 ON FNR(5) GOSUB 3110,3120,3130,3140
,3150:PRINT "{CSD}{CSD}{CSD}":RETURN
3110 PRINT "*** FLASH! FLASH! FLASH! ***
":RETURN
3120 PRINT "*** BULLETIN!!! ***":RETURN
3130 PRINT "*** ALERT !!! ***":RETURN
3140 PRINT "*** SPECIAL NEWS BULLETIN **
*":RETURN
3150 PRINT "*** TO ALL CITIZENS ***":RET
URN
3200 WRD$=PL$(FNR(PL))+ " ":GOSUB 2900
3290 RETURN

```

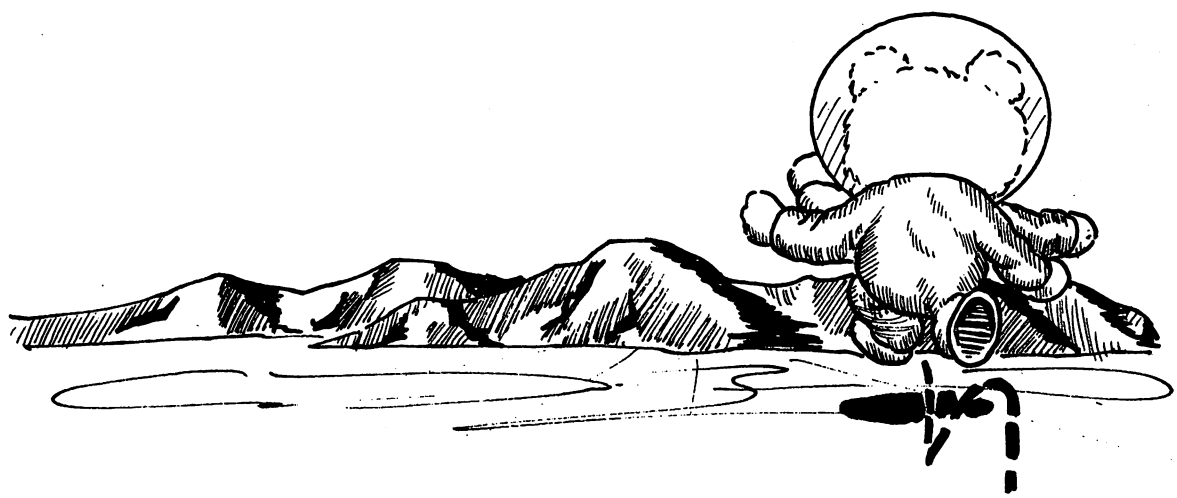


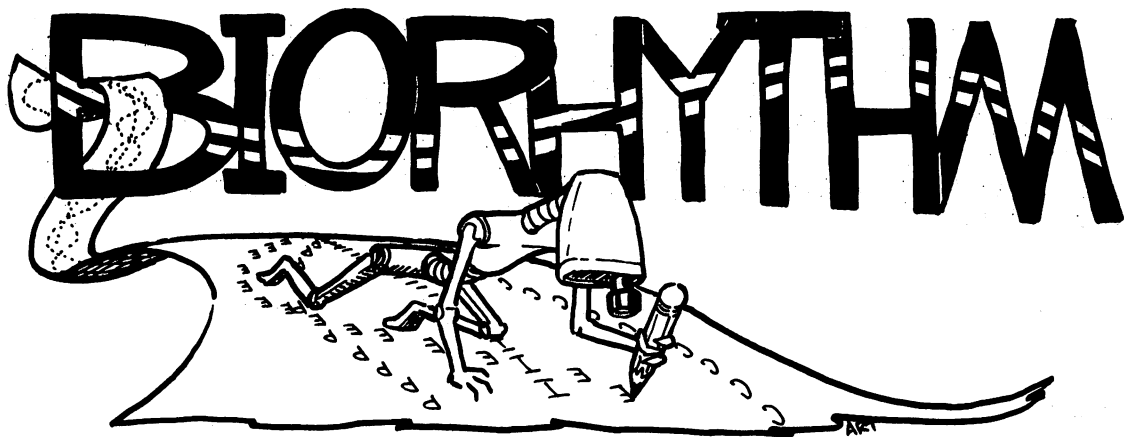
```

3300 ON FNR(5) GOSUB 3310,3320,3330,3340
,3350:GOSUB 2900:RETURN
3310 WRD$="WAS ATTACKED BY ":RETURN
3320 WRD$="WAS EATEN BY ":RETURN
3330 WRD$="IS UNDER THE SPELL OF ":RETUR
N
3340 WRD$="IS BEING INVADED BY ":RETURN
3350 WRD$="IS BEING OVER-RUN BY ":RETURN
3400 WRD$=MO$(FNR(MO))+ " ":GOSUB 2900
3490 RETURN
3500 WRD$="FROM ":GOSUB 2900
3510 WRD$=HO$(FNR(HO))+ ". ":GOSUB 2900
3590 RETURN
3600 WRD$=SO$(FNR(SO))+ " ":GOSUB 2900
3690 RETURN
3700 WRD$="TRIED TO ":GOSUB 2900
3705 ON FNR(5) GOSUB 3710,3720,3730,3740
,3750:GOSUB 2900:RETURN
3710 WRD$="KILL THEM ":RETURN
3720 WRD$="FIGHT THEM ":RETURN
3730 WRD$="HOLD UP A CROSS ":RETURN
3740 WRD$="ATTACK AT DAWN ":RETURN
3750 WRD$="SHOOT THEM ":RETURN
3800 WRD$="BUT THEY ":GOSUB 2900
3805 ON FNR(5) GOSUB 3810,3820,3830,3840
,3850:GOSUB 2900:RETURN
3810 WRD$="WERE TOO TOUGH. ":RETURN
3820 WRD$="KEPT COMING. ":RETURN
3830 WRD$="YELLED AND LAUGHED. ":RETURN
3840 WRD$="SCREAMED FOR MORE. ":RETURN
3850 WRD$="SHOT BACK. ":RETURN
3900 ON FNR(5) GOSUB 3910,3920,3930,3940
,3950:GOSUB 2900:RETURN
3910 WRD$="FINALLY, ":RETURN
3920 WRD$="LATER, ":RETURN
3930 WRD$="THEN... ":RETURN
3940 WRD$="BUT THEN, ":RETURN
3950 WRD$="LUCKILY, ":RETURN
4000 ON FNR(5) GOSUB 4010,4020,4030,4040
,4050:GOSUB 2900:RETURN
4010 WRD$="YELLED AT THEM, ":RETURN
4020 WRD$="DROPPED WATER ON THEM, ":RETN
RN
4030 WRD$="EXPOSED THEM TO MEASLES, ":RE
TURN
4040 WRD$="NUKED THEM, ":RETURN
4050 WRD$="SHOWED THEM RERUNS OF 'I LOVE
LUCY', ":RETURN
4100 WRD$="SO THEY ":GOSUB 2900
4105 ON FNR(5) GOSUB 4110,4120,4130,4140
,4150:GOSUB 2900:RETURN

```

4110 WRD\$="DIED.":RETURN
4120 WRD\$="TURNED INTO LITTLE BROWN LUMP
S.":RETURN
4130 WRD\$="PASSED AWAY.":RETURN
4140 WRD\$="LEFT FOR HOME.":RETURN
4150 WRD\$="VANISHED INTO NOTHINGNESS.":R
ETURN





This program is designed to interpret input, and from it, graph the user's biorhythms. You can trust the results as much as you want, but don't expect the results to be infallible! Biorhythms, though fascinating, are still considered to be unscientific. There are several features of the program worth noting, whether you believe in Biorhythms or not:

Lines 2050-2060 contain the names of the twelve months and the seven days of the week. In later sections, like lines 3290 and 3510, parts of these strings will be cut out to display the appropriate month or day.

In each of lines 3060, 3130, and 3230, the program makes it clear what format the response should be entered under. Little prompts like this help make programs more fun to use and less likely to crash.

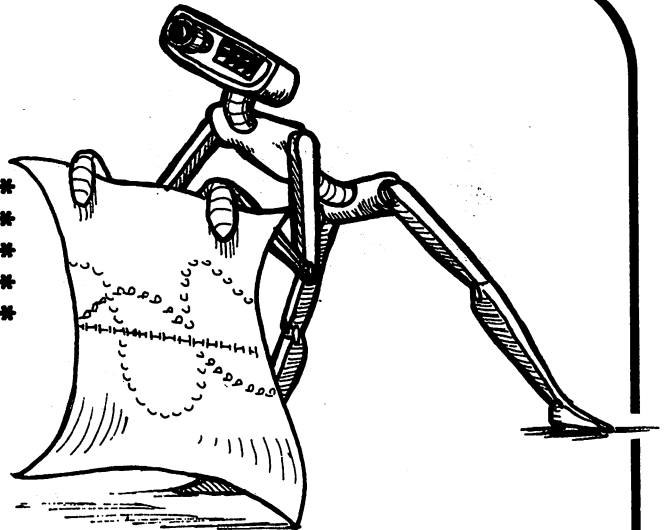
Line 1230 allows you to choose whether you want to see your biorhythms printed out on the screen or on the printer. The channel to the printer is opened in line 3250, right before the actual printing of the graph takes place.

The program uses some complex mathematical functions to determine how the three cycles are plotted on the screen or printer.

```

10 REM *****
11 REM ***                               ***
12 REM ***      BIORHYTHM                ***
13 REM ***                               ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 END
1000 :
1010 REM *** INSTRUCTIONS ***
1020 :
1030 VT$="{HOME}{CSD}{CSD}{CSD}{CSD}{CSD}
} {CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
} {CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
} {CSD}{CSD}{CSD}"
1040 POKE 53280,6:POKE 53281,6
1050 GOSUB 1900
1060 PRINT "A BIORHYTHMIC CALENDAR IS CO
MPOSED OF"
1070 PRINT "THE LETTERS I, E AND P. EAC
H OF THESE"
1080 PRINT "REPRESENTS ONE OF YOUR MEASU
RABLE"
1090 PRINT "BIORHYTHMIC CYCLES.":PRINT
1100 PRINT:PRINT TAB(6);"I=INTELLECTUAL
STATE"
1110 PRINT:PRINT TAB(6);"E=EMOTIONAL STA
TE"
1120 PRINT:PRINT TAB(6);"P=PHYSICAL STAT
E"
1130 GOSUB 1800:GOSUB 1900
1140 PRINT "THIS PROGRAM WILL GRAPH OUT
YOUR UNIQUE"
1150 PRINT "BIORHYTHMIC CYCLES, EITHER O
N THE SCREEN";
1160 PRINT "OR TO A PRINTER.":PRINT
1170 PRINT "PRESS THE SPACEBAR TO PAUSE
THE PRINTOUT";
1180 PRINT "ON THE SCREEN AND AGAIN TO R
ESUME.":PRINT
1190 PRINT "SHOULD I OUTPUT TO:":PRINT
1200 PRINT TAB(8);"S)CREEN"
1210 PRINT TAB(10);"-OR-"
1220 PRINT TAB(8);"P)RINTER"
1230 PRINT LEFT$(VT$,24);"WHICH DO YOU W

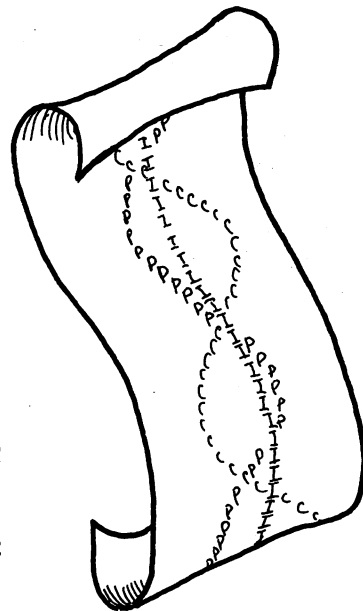
```



```

ANT (S/P) ? {RVS ON} {RVS OFF}";
1240 GET P$:IF P$="" THEN 1240
1250 IF P$="P" THEN PRINT "{CSL}P":GOTO
1790
1260 PRINT "{CSL}S"
1790 RETURN
1800 PRINT LEFT$(VT$,24);"PRESS {RVS ON}
RETURN{RVS OFF} WHEN READY TO CONTINUE:
{RVS ON} {RVS OFF}"
1810 GET AN$:IF AN$="" THEN 1810
1890 RETURN
1900 PRINT "{CLR}{WHT}{CSD}{CSD}";TAB(11
);"*** BIORHYTHM ***":PRINT:PRINT
1990 RETURN
2000 :
2010 REM *** SETUP ***
2020 :
2030 SW=29:PI2=6.28318531
2040 DIM A$(SW),PL(23),EL(28),IL(33)
2050 C$="JANFEBMARAPR MAYJUNJUL AUGSEP OCTN
OVDEC"
2060 W$="SUNMONTUEWEDTHUFRISAT"
2070 FOR I=1 TO SW:A$(I)=" ":NEXT
2080 B=INT(SW/2):A=B+1.5
2090 FOR I=1 TO 33
2100 IL(I)=INT(A+B*SIN(PI2*(I-1)/33))
2110 IF I>28 THEN 2150
2120 EL(I)=INT(A+B*SIN(PI2*(I-1)/28))
2130 IF I>23 THEN 2150
2140 PL(I)=INT(A+B*SIN(PI2*(I-1)/23))
2150 NEXT
2160 ML$=" -.....0.....+"
2990 RETURN
3000 :
3010 REM *** PLAY ***
3020 :
3030 GOSUB 1900
3040 INPUT "WHAT IS YOUR NAME";N$
3050 ER=0
3060 PRINT:PRINT "WHAT IS YOUR BIRTHDATE
?":INPUT "(MM,DD,YY)";M$,D$,Y$
3070 M=VAL(M$)
3075 IF M<1 OR M>12 OR M<>INT(M) THEN PR
INT "INCORRECT MONTH":ER=1
3080 D=VAL(D$)
3085 IF D<1 OR D>31 OR D<>INT(D) THEN PR
INT "INCORRECT DAY":ER=1
3090 Y=VAL(Y$)
3095 IF Y<0 OR Y<>INT(Y) THEN PRINT "INC
ORRECT YEAR":ER=1

```



```

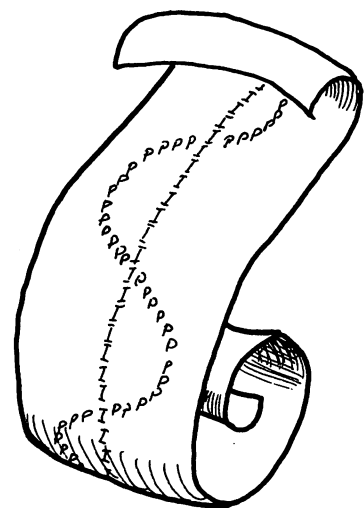
3100 IF ER THEN ER=0:GOTO 3060
3110 IF Y<100 THEN Y=Y+1900
3120 GOSUB 3800:S=J
3130 PRINT:PRINT "WHAT IS THE START DATE
?":INPUT "(MM,DD,YY)";M$,D$,Y$
3140 M1=VAL(M$)
3145 IF M1<1 OR M1>12 OR M1<>INT(M1) THE
N PRINT "INCORRECT MONTH":ER=1
3150 D1=VAL(D$)
3155 IF D1<1 OR D1>31 OR D1<>INT(D1) THE
N PRINT"INCORRECT DAY":ER=1
3160 Y1=VAL(Y$)
3165 IF Y1<0 OR Y1<>INT(Y1) THEN PRINT"I
NCCORRECT YEAR":ER=1
3170 IF ER THEN ER=0:GOTO 3130
3180 IF Y1<100 THEN Y1=Y1+1900
3190 IF Y1>Y THEN 3230
3200 IF Y1=Y AND M1>M THEN 3230
3210 IF Y1=Y AND M1=M AND D1>=D THEN 323
0
3220 PRINT "START DATE BEFORE BIRTHDATE"
:GOTO 3130
3230 PRINT:INPUT "HOW MANY DAYS";Z$:Z=VA
L(Z$)
3235 IF Z<1 OR Z<>INT(Z) THEN 3230
3240 PRINT "{CLR}";
3250 IF P$="P" THEN OPEN 4,4,2:CMD 4
3260 PRINT TAB(10);"BIORHYTHMIC CALENDAR
"
3270 PRINT TAB(14);"--- FOR ---"
3280 PRINT TAB(INT((40-LEN(N$))/2));N$
3290 PRINT TAB(8);"BORN ON, ";MID$(W$,3*N
-2,3);D;MID$(C$,3*M-2,3);Y:PRINT
3300 PRINT TAB(5);"P=PHYSICAL      (23 D
AY CYCLE)"
3310 PRINT TAB(5);"E=EMOTIONAL     (28 D
AY CYCLE)"
3320 PRINT TAB(5);"I=INTELLECTUAL  (33 D
AY CYCLE)"
3330 PRINT:PRINT:PRINT
3340 M=M1:D=D1:Y=Y1:GOSUB 3800:GOSUB 390
0:J2=J1
3350 M=M+1:D=1:GOSUB 3900:LE=J1-J2
3360 IF LE>Z THEN LE=Z
3370 Z=Z-LE
3380 D1=D1-1
3390 F=J-S:E=F+LE-1
3400 FOR K=F TO E
3410 D1=D1+1
3420 PK=K-INT(K/23)*23+1

```

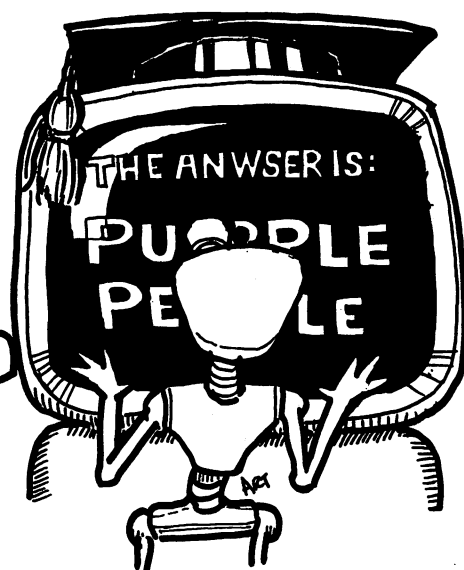
```

3430 EK=K-INT(K/28)*28+1
3440 IK=K-INT(K/33)*33+1
3450 A$(PL(PK))="P"
3460 IF A$(EL(EK))=" " THEN A$(EL(EK))="
E":GOTO 3480
3470 A$(EL(EK))="*"
3480 IF A$(IL(IK))=" " THEN A$(IL(IK))="
I":GOTO 3500
3490 A$(IL(IK))="*"
3500 IF A$(B+1)=" " THEN A$(B+1)="."
3510 IF D1=1 OR K=F THEN PRINT MID$(C$,3
*M1-2,3);Y;ML$
3520 PRINT MID$(W$,3*N-2,3);RIGHT$(" "+
STR$(D1),3);" ";
3530 FOR I=1 TO SW:PRINT A$(I);:A$(I)="
":NEXT:PRINT
3540 IF P$="P" THEN 3570
3550 GET AN$:IF AN$="" THEN 3570
3560 GET AN$:IF AN$="" THEN 3560
3570 N=N+1:IF N>7 THEN N=1
3580 NEXT
3590 D1=1:M1=M1+1:IFM1>12 THEN M1=1:Y1=Y
1+1
3600 IF Z THEN 3340
3610 PRINT TAB(9);ML$
3620 IF P$="P" THEN PRINT#4:CLOSE 4
3630 PRINT "DO YOU WANT TO MAKE ANOTHER
CHART? {RVS ON} {RVS OFF}";:GET AN$
3640 GET AN$:IF AN$="" THEN 3640
3650 IF AN$="Y" THEN 3000
3660 PRINT "{CSL}N"
3790 RETURN
3800 REM FIND DAY OF WEEK
3810 IF M<3 THEN MO=M+10:YO=Y-1:GOTO 383
0
3820 MO=M-2:YO=Y
3830 J=INT(YO/100):K=YO-(J*100)
3840 N=INT((13*MO-1)/5)+D+K+INT(K/4)+INT
(J/4)-2*J+77:N=N-INT(N/7)*7+1
3850 MO=MO-1:J=INT((146097*J)/4)+D+INT((
1461*K)/4)+INT((153*MO+2)/5)
3890 RETURN
3900 REM FIND JULIAN DATE
3910 L=2:IF M<3 THEN L=0:GOTO 3960
3920 IF Y<>INT(Y/4)*4 THEN 3960
3930 IF Y<>INT(Y/100)*100 THEN 3950
3940 IF Y<>INT(Y/400)*400 THEN 3960
3950 L=1
3960 J1=INT((3055*(M+2))/100)-91+D-L
3990 RETURN

```



C-64 Learner



This game is a good demonstration of how “intelligence” is programmed into many computers. This intelligence is actually created with a complex series of “truth tests,” each of which check to see if part of a larger statement is true. The computer begins by being able to identify two objects, a car and a house (line 2060). With each object entered, it becomes harder to stump the computer. The DIM statement in line 2030 sets aside a certain amount of memory for the variables QU\$(1) - QU\$(50), (and for RI, WR, and RA\$ as well). The variables QU\$(1) through QU\$(50) will contain the QUESTIONS entered by the user after the object has tried to guess the object. This program demonstrates how you can write to array tables and how to search arrays for matching strings. You could use the routines in this program to write educational programs and tests.

```

10 REM *****
11 REM ***          ***
12 REM ***C-64 LEARNER ***
13 REM ***          ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 END
1000 :
1010 REM *** INSTS
1020 :
1030 VT$="{HOME}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}"
1040 POKE53280,6:POKE53281,6
1050 GOSUB 1900
1060 PRINT "THIS IS A GAME THAT HAS THE
ABILITY TO"
1070 PRINT "LEARN. THE COMPUTER WILL TR
Y TO GUESS"
1080 PRINT "AN OBJECT THAT YOU SELECT AT
RANDOM.":PRINT
1090 PRINT "WHENEVER YOU STUMP THE COMPU
TER, YOU ARE";
1100 PRINT "ASKED ABOUT THE OBJECT YOU S
ELECTED. BY";
1110 PRINT "COMPILING THIS INFORMATION,
THE COMPUTER";
1120 PRINT "'LEARNS'." :PRINT
1130 PRINT "ENTER THE WORD 'STOP' WHEN Y
OU ARE DONE."
1800 PRINT LEFT$(VT$,24);"PRESS {RVS ON}
RETURN{RVS OFF} WHEN READY TO CONTINUE:
{RVS ON} {RVS OFF}"
1810 GET AN$:IF AN$="" THEN 1810
1890 RETURN
1900 PRINT "{CLR}{WHT}{CSD}{CSD}";TAB(10
);"*** C-64 LEARNER ***":PRINT:PRINT
1990 RETURN
2000 :
2010 REM *** SETUP
2020 :
2030 DIM GU$(50),RI(50),WR(50),RA$(50),W

```

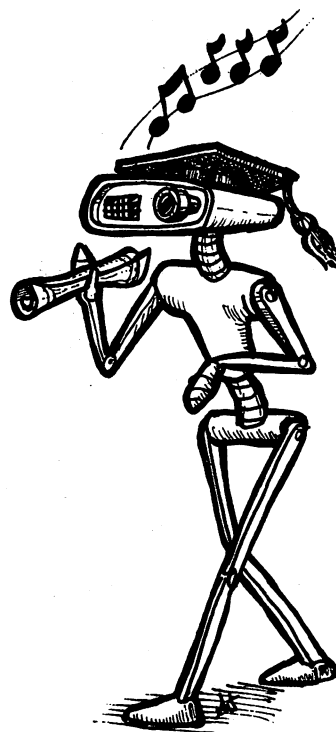


```

A$(50)
2040 QU$(1)="DOES IT MOVE ALONG THE GROU
ND"
2050 RI(1)=0:WR(1)=0
2060 RA$(1)="CAR":WA$(1)="HOUSE"
2070 FR=2
2990 RETURN
3000 :
3010 REM *** PLAY ***
3020 :
3030 GOSUB 1900
3040 LI=1
3050 PRINT "I KNOW OF";FR;"OBJECTS...":P
RINT
3060 PRINT:PRINT QU$(LI);:INPUT AN$:AN$=
LEFT$(AN$,1)
3070 IF AN$="Y" THEN 3100
3080 IF AN$="N" THEN 3200
3090 IF AN$="S" THEN 3990
3095 PRINT "PLEASE ANSWER 'YES' OR 'NO'.
..":PRINT:GOTO 3060
3100 IF RI(LI) THEN LI=RI(LI):GOTO 3060
3105 GU$=RA$(LI):GOTO 3300
3200 IF WR(LI) THEN LI=WR(LI):GOTO 3060
3205 GU$=WA$(LI)
3300 PRINT:PRINT "IS IT A ";GU$;
3310 INPUT TA$:TA$=LEFT$(TA$,1)
3320 IF TA$="S" THEN 3990
3330 IF TA$="Y" THEN PRINT:PRINT "I GOT
IT!!!":FOR PA=1 TO 1000:NEXT:GOTO 3000
3340 PRINT:PRINT:INPUT "WHAT WAS THE OBJ
ECT";NA$
3350 IF FR<51 THEN 3380
3360 PRINT "I CAN'T REMEMBER THAT ONE. M
Y MEMORY SEEMS TO BE FULL."
3370 FOR PA=1 TO 1000:NEXT:GOTO 3000
3380 PRINT:PRINT "WHAT IS A QUESTION THA
T I COULD"
3390 PRINT "USE TO TELL THE DIFFERENCE B
ETWEEN"
3400 PRINT GU$;" AND ";NA$;:INPUT QU$
3410 PRINT:PRINT "FOR ";NA$;" THE ANSWER
IS WHAT";:INPUT YN$:YN$=LEFT$(YN$,1)
3420 IF YN$<>"Y" AND YN$<>"N" THEN 3410
3430 IF AN$="Y" THEN RI(LI)=FR:LI=FR:FR=
FR+1
3440 IF AN$="N" THEN WR(LI)=FR:LI=FR:FR=
FR+1
3450 QU$(LI)=QU$

```

```
3460 IF YN$="Y" THEN RA$(LI)=NA$:WA$(LI)
=GU$
3470 IF YN$="N" THEN RA$(LI)=GU$:WA$(LI)
=NA$
3480 GOTO 3000
3990 RETURN
```





This is definitely a thinking man (or woman's) game! Distantly related to the game Mastermind, you are given clues to help determine a random three number combination in which each number is different from the other two. List lines 2040-2070. This is where the three number combination is defined. Line 2040, $DEF\ FNR(X)=INT(RND(1)*X)$ will produce a random number in the range specified by the value X which is used with the FN R commands in lines 2050-2070. So when line 2050 sets N1 (the first number) equal to FN R(10), the value in parentheses is entered into the function R as stated in line 2040. This will yield a number between 0 and 9. The second number (N2) is generated in the first part of line 2060; the second statement in the line compares it to the (N1). If they are the same, the line is repeated until a new N2 is generated. Line 2070 does the same thing for N3. Now list lines 3100-3130. These lines separate your single three-digit guess into three separate guesses (G1, G2, and G3). Here's how. When any number is converted into an integer, the portion of the number to the right of the decimal point is truncated (cut off). Here are a few examples: INT 3.4 is 3, INT 9.989 is 9, INT 562.3 is 562, and INT 0.3 is 0. Taking line 3100, assume that the guess is 567. 567 divided by 100 is 5.67. When converted to an INTeGer, 5.67 becomes 5 (G1=5). Line 3110 takes 567, subtracts 5×100 , divides the result (67) by 10, and converts 6.7 into the INTeGer. Now G1=5 and G2=6. Lastly, 3120 takes 567 (ANS) and subtracts from it $(5(G1) \text{ times } 100 \text{ plus } 6(G2) \text{ times } 10)$ or 560. The result $(567 - 560)$ is now stored in G3. So now G1=5, G2=6, and G3=7. Statistically, even when you are unlucky, the solution can be derived in no more than seven guesses. You've been challenged, now go to it!

```

10 REM *****
11 REM ***          ***
12 REM ***    DIGITS    ***
13 REM ***          ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 GOSUB 4000
60 END
1000 :
1010 REM *** INSTRUCTIONS ***
1020 :
1030 VT$="{HOME}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}"
1040 POKE 53280,6:POKE 53281,6
1050 GOSUB 1900
1060 PRINT"I WILL THINK OF A NUMBER BETW
EEN 012"
1070 PRINT "AND 987. EACH DIGIT IN THE N
UMBER WILL"
1080 PRINT "BE DIFFERENT FROM THE OTHER
TWO.":PRINT
1090 PRINT "THE OBJECT OF THE GAME IS TO
GUESS THE"
1100 PRINT "SOLUTION IN AS FEW TRIES AS
POSSIBLE."
1110 GOSUB 1800:GOSUB 1900
1120 PRINT "AFTER EACH GUESS I WILL PRIN
T A HINT"
1130 PRINT "LINE AS FOLLOWS:":PRINT
1140 PRINT ">FOR EACH CORRECT DIGIT IN T
HE CORRECT"
1150 PRINT " POSITION, I WILL PRINT AN '
X'.":PRINT
1160 PRINT ">FOR A CORRECT DIGIT WITH AN
INCORRECT"
1170 PRINT " POSITION, I WILL PRINT AN '
O'.":PRINT
1180 PRINT ">FOR TOTALLY INCORRECT DIGIT
S, I WILL"
1190 PRINT " PRINT A '-'.":PRINT

```

```

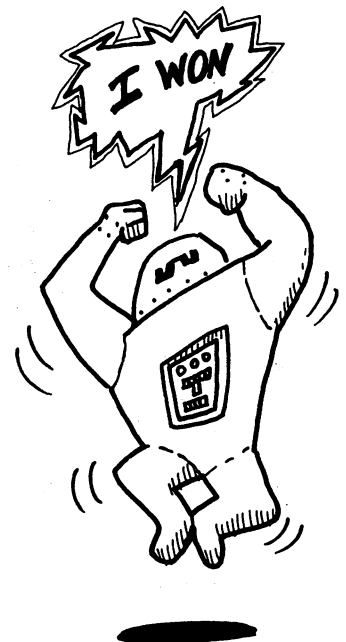
1200 PRINT "PLAY WILL CONTINUE UNTIL YOU
      GUESS THE"
1210 PRINT "NUMBER. TO QUIT EARLY, SIMPL
      Y HIT THE"
1220 PRINT "{RVS ON}RETURN{RVS OFF} KEY
      FOR YOUR GUESS."
1230 GOSUB 1800:GOSUB 1900
1240 PRINT "HERE IS A TABLE TO HELP YOU
      UNDERSTAND"
1250 PRINT "HOW THE HINTS WORK.":PRINT
1260 PRINT "ANSWER      GUESS      HINT LI
      NE "
1270 PRINT "-----"
      ---":PRINT
1280 PRINT " 065          703          0--":
      PRINT
1290 PRINT " 562          463          X--":
      PRINT
1300 PRINT " 918          890          00--":
      PRINT
1310 PRINT " 390          305          X0--":
      PRINT
1320 PRINT " 271          721          X00":
      PRINT
1330 PRINT " 425          780          ----"
1800 PRINT LEFT$(VT$,24);"PRESS {RVS ON}
      RETURN{RVS OFF} WHEN READY TO CONTINUE:
      {RVS ON} {RVS OFF}"
1810 GET AN$:IF AN$="" THEN 1810
1890 RETURN
1900 PRINT "{CLR}{WHT}{CSD}{CSD}";TAB(13
      );"*** DIGITS ***":PRINT:PRINT
1990 RETURN
2000 :
2010 REM *** SETUP ***
2020 :
2030 PRINT "{CLR}";
2040 DEF FNR(X)=INT(RND(1)*X)
2050 N1=FNR(10)
2060 N2=FNR(10):IF N1=N2 THEN 2060
2070 N3=FNR(10):IF N3=N1 OR N3=N2 THEN 2
      070
2990 RETURN
3000 :
3010 REM *** PLAY ***
3020 :
3030 GOSUB 1900
3040 PRINT "OKAY, I'VE GOT A NUMBER...":
      PRINT
3050 AN$="":INPUT "WHAT IS YOUR GUESS";A
      N$:IF AN$="" THEN 3990

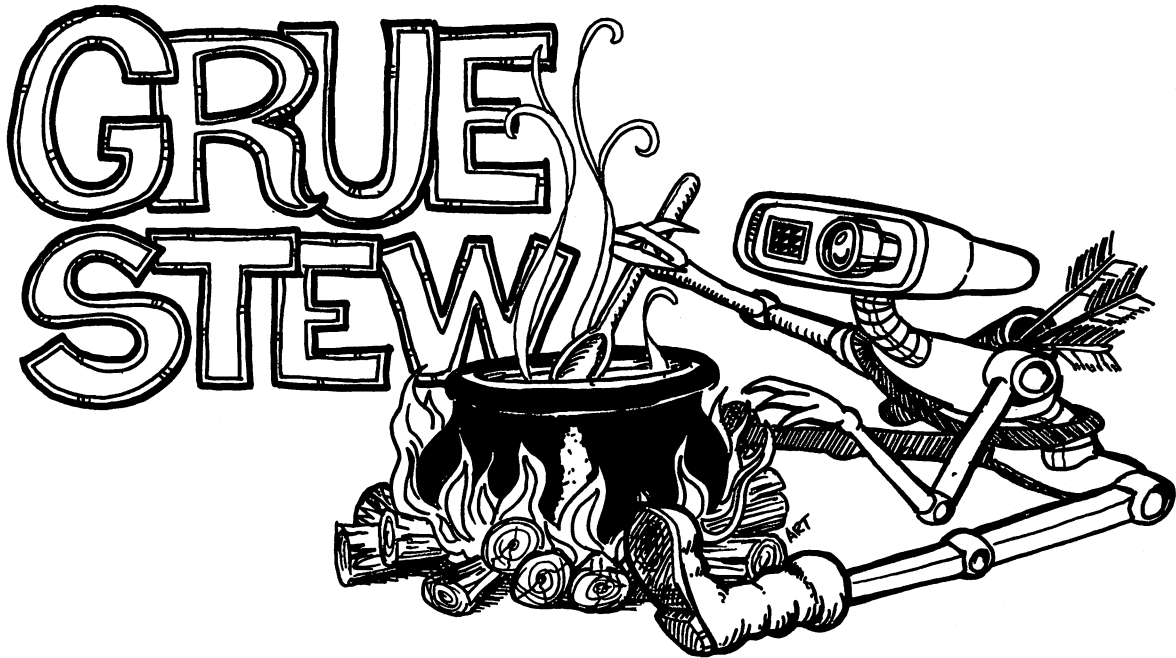
```

```

3060 IF LEN(AN$)=3 THEN 3080
3070 PRINT "TYPE THREE DIGITS PLEASE":GO
TO 3050
3080 AN=VAL(AN$)
3090 IF AN<0 OR AN>999 OR AN<>INT(AN) TH
EN 3070
3100 G1=INT(AN/100)
3110 G2=INT((AN-G1*100)/10)
3120 G3=AN-(G1*100+G2*10)
3130 IF G1<>G2 AND G1<>G3 AND G2<>G3 THE
N 3200
3140 PRINT "TYPE THREE DIFFERENT DIGITS
PLEASE":GOTO 3050
3200 CP=0:CD=0:MI=0
3210 IF G1=N1 THEN CP=CP+1
3220 IF G2=N2 THEN CP=CP+1
3230 IF G3=N3 THEN CP=CP+1
3240 IF G1=N2 OR G1=N3 THEN CD=CD+1
3250 IF G2=N1 OR G2=N3 THEN CD=CD+1
3260 IF G3=N1 OR G3=N2 THEN CD=CD+1
3300 MI=3-CP-CD
3310 PRINT "FOR YOUR GUESS OF ";AN$;", I
HINT ";
3320 IF CP>0 THEN FOR I=1 TO CP:PRINT "X
";:NEXT
3330 IF CD>0 THEN FOR I=1 TO CD:PRINT "O
";:NEXT
3340 IF MI>0 THEN FOR I=1 TO MI:PRINT "-
";:NEXT
3350 PRINT:PRINT:NG=NG+1:IF CP<3 THEN 30
50
3990 RETURN
4000 :
4010 REM *** END ***
4020 :
4030 PRINT "THE GAME IS OVER...":PRINT
4040 IF CP=3 THEN PRINT "YOU GUESSED IT
IN ONLY";NG;" TRIES!"
4050 T$=""
4060 N$=STR$(N1)+STR$(N2)+STR$(N3)
4070 FOR I=1 TO LEN(N$):M$=MID$(N$,I,1):
IF M$<>" " THEN T$=T$+M$
4080 NEXT
4090 IF CP<3 THEN PRINT "THE CORRECT NUM
BER WAS ";T$
4100 PRINT "DO YOU WANT TO PLAY AGAIN? {
RVS ON} {RVS OFF}";
4110 GET AN$:IF AN$="" THEN 4110
4120 IF AN$="Y" THEN RUN
4130 PRINT "{CSL}N"
4990 RETURN

```



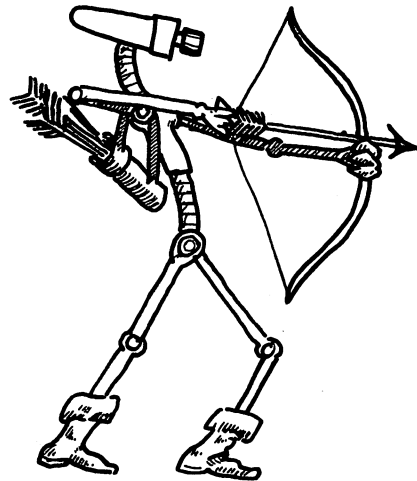


In this game, you are given a perplexing challenge: you must wend your way through a series of caverns in search of the Grue. Unfortunately, you are not spelunking in the most stable of areas — earthquakes can shift the position of the rooms at any time! Although this game does not feature any colorful graphics, it does use some interesting concepts. You may ask “Why do I get different mazes each time I run the program?” Here’s why. Lines 2100-2295 contain descriptions of the room. (Notice how each pair of DATA statements make up a single room description.) Line 2050 reads these DATA statements two at a time (READ A\$,B\$). This structure allows room descriptions that are longer than one-program-line long. Lines 2800 through 2950 assign certain variables random values. Beginning at line 3050, this becomes relevant. One of four messages is printed. Which one it is depends on the RaNDom values of the several variables (EX, P1, P2, B1, B2, and GU). Line 3100 reveals that there is a one in fifteen chance of an earthquake every move. Paraphrased, line 3100 says, “If a RaNDom number between 1 and 15 is equal to 4, then GOSUB 60300 (make a beep), then PRINT ‘EARTHQUAKE,’ then continue with the GOSUB 2900 and GOTO 3000.” As you go through the program, you will notice that the execution of most of the lines depends, either directly or indirectly, on the value of a randomly generated integer.

```

10 REM *****
11 REM ***          ***
12 REM ***  GRUE STEW  ***
13 REM ***          ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 GOSUB 4000
60 END
1000 :
1010 REM *** INSTRUCTIONS ***
1020 :
1030 POKE 53280,6:POKE 53281,6
1040 VT$="{HOME} {CSD} {CSD} {CSD} {CSD} {CSD}
      }{CSD} {CSD} {CSD} {CSD} {CSD} {CSD} {CSD} {CSD}
      }{CSD} {CSD} {CSD} {CSD} {CSD} {CSD} {CSD} {CSD}
      }{CSD} {CSD} {CSD}"
1050 GOSUB 1900
1060 PRINT "IN THIS GAME YOU ARE A BRAVE
      HUNTER."
1070 PRINT "YOU ARE ALSO VERY HUNGRY.  S
      O YOU"
1080 PRINT "DECIDE TO GO 'GRUE' HUNTING.
      A GRUE,"
1090 PRINT "AS EVERYONE KNOWS, IS THE KE
      Y INGREDIENT";
1100 PRINT "IN GRUE STEW.":PRINT
1110 PRINT "YOU ARE GOING TO ENTER A SER
      IES OF"
1120 PRINT "UNDERGROUND CAVES, IN SEARCH
      OF THE"
1130 PRINT "STEW BASE, THE GRUE.":PRINT
1140 PRINT "IF YOU CAN BAG A GRUE AND GE
      T OUT OF"
1150 PRINT "THE CAVES, THEN YOU WILL GET
      YOUR STEW"
1160 PRINT "(AND WIN THE GAME)!!"
1170 GOSUB 1800:GOSUB 1900
1180 PRINT "ONCE IN THE MAZE, YOU CAN EI
      THER MOVE"
1190 PRINT "TO A DIFFERENT CAVERN OR SHO
      OT AN ARROW"
1200 PRINT "INTO AN ADJOINING CAVERN, IN

```




```

HOPES OF"
1210 PRINT "HITTING A FEROCIOUS GRUE.":P
RINT
1220 PRINT "I WILL ASK: MOVE OR SHOOT?,
AND YOU MUST";
1230 PRINT "REPLY 'M' FOR MOVE OR 'S' FO
R SHOOT.":PRINT
1240 PRINT "IF YOU DECIDE TO MOVE, YOU C
AN DO SO IN"
1250 PRINT "ANY OF THE FOUR COMPASS DIRE
CTIONS. WHEN";
1260 PRINT "ASKED WHICH WAY, ENTER 'N' F
OR NORTH,'S'";
1270 PRINT "FOR SOUTH,'W' FOR WEST, OR '
E' FOR EAST."
1280 GOSUB 1800:GOSUB 1900
1290 PRINT "IF YOU DECIDE TO SHOOT, YOU
WILL BE"
1300 PRINT "ASKED 'SHOOT WHICH WAY?', AN
D YOU MUST"
1310 PRINT "REPLY: 'N' FOR NORTH, 'S' FO
R SOUTH,"
1320 PRINT "'E' FOR EAST, 'W' FOR WEST."
:PRINT
1330 PRINT "IF YOU HIT THE GRUE, YOU WIL
L BE TOLD,"
1340 PRINT "AND YOU MUST TRY TO EXIT THE
CAVE.":PRINT
1350 PRINT "BUT...THERE ARE OTHER THINGS
IN THE"
1360 PRINT "CAVES. THERE ARE GIANT BATS
THAT WILL"
1370 PRINT "PICK YOU UP AND DROP YOU ELS
EWHERE."
1380 GOSUB 1800:GOSUB 1900
1390 PRINT "THERE ARE BOTTOMLESS PITS.
IF YOU FALL"
1400 PRINT "INTO ONE OF THESE YOU'LL NEV
ER GET OUT!":PRINT
1410 PRINT "OF COURSE THERE IS THE GRUE
HIMSELF."
1420 PRINT "THOUGH NOT AN AGGRESIVE CREA
TURE, HE"
1430 PRINT "WILL EAT YOU IF YOU COME TOO
CLOSE.":PRINT
1440 PRINT "THERE ARE ALSO QUAKES THAT M
OVE THINGS"
1450 PRINT "AROUND IN THE CAVES (BATS, P
ITS,"
1460 PRINT "THE GRUE, AND THE EXIT!)"

```



```

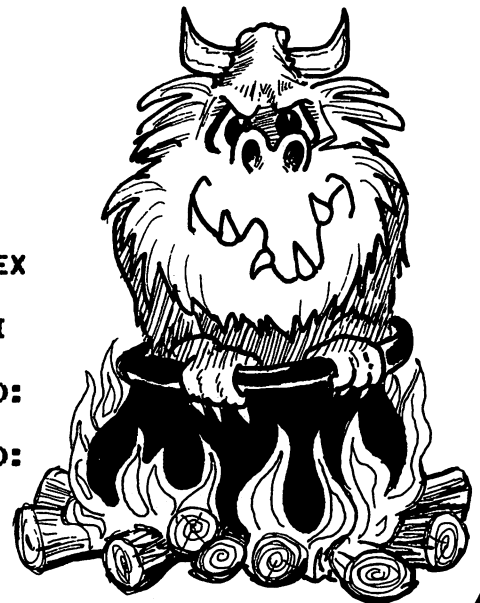
1800 PRINT LEFT$(VT$,24);"PRESS {RVS ON}
RETURN{RVS OFF} WHEN READY TO CONTINUE:
{RVS ON} "
1810 GET AN$:IF AN$="" THEN 1810
1890 RETURN
1900 PRINT "{CLR}{WHT}{CSD}{CSD}";TAB(11
);"*** GRUE STEW ***":PRINT:PRINT
1990 RETURN
2000 :
2010 REM *** SETUP ***
2020 :
2030 GOSUB 1900
2040 DIM RO$(20),TR(20,4)
2050 FOR I=1 TO 20:READ A$,B$:RO$(I)=A$+
" "+B$:NEXT
2100 DATA YOU ARE IN A SMALL ROOM WITH R
OCKS AND DEBRIS SCATTERED EVERYWHERE.
2105 DATA
2110 DATA DUCK YOUR HEAD IN HERE; AS LAR
GE ROCK STALACTITES HANG FROM THE
2115 DATA CEILING.
2120 DATA THE ROOM HERE SLOPES
2125 DATA DOWNWARD.
2130 DATA THIS ROOM IS VERY SMALL; BUT I
THINK WE CAN MAKE IT THROUGH OK.
2135 DATA
2140 DATA THIS IS A VERY LARGE ROOM WITH
A LARGE BOULDER IN THE CENTER OF IT.
2145 DATA
2150 DATA THIS IS THE CENTER OF A NARROW
PASSAGE THAT CONNECTS OTHER ROOMS.
2155 DATA
2160 DATA THIS PASSAGE IS VERY LOW; BUT
IF WE CRAWL WE CAN MAKE IT.
2165 DATA
2170 DATA THIS IS A VERY DIRTY ROOM; IT
WAS PARTIALLY FILLED IN BY THE
2175 DATA LAST EARTH- QUAKE THAT HIT.
2180 DATA THIS ROOM IS ABOUT AVERAGE SIZ
E; BUT IS FILLED WITH A PUNGENT AROMA
2185 DATA THAT IS VERYNAUSEATING.
2190 DATA YOU ARE IN A SMALL
2195 DATA PASSAGEWAY.
2200 DATA YOU ARE IN A SMALL
2205 DATA PASSAGEWAY.
2210 DATA YOU ARE IN A SMALL
2215 DATA PASSAGEWAY.
2220 DATA YOU ARE IN A SMALL
2225 DATA PASSAGEWAY.
2230 DATA YOU ARE IN A SMALL

```

```

2235 DATA PASSAGEWAY.
2240 DATA YOU ARE IN A SMALL
2245 DATA PASSAGEWAY.
2250 DATA A SMALL HOLE IN THE CEILING LETS LIGHT FROM OUTSIDE THROUGH ...
2255 DATA BUT YOU WOULD NEVER FIT THROUGH IT.
2260 DATA SOMEONE HAS LEFT A LIGHTED TORCH ON THE WALL AND IT ILLUMINATES YOUR
2265 DATA PASSAGE.
2270 DATA A RIVULET OF WATER SLOWLY TRICKLES FROM A HOLE IN THE WALL.
2275 DATA
2280 DATA A SMALL HOLE TO YOUR LEFT ATTRACTS YOUR ATTENTION; BUT IT IS TOO
2285 DATA SMALL TO BE OF ANY CONCERN.
2290 DATA YOU ARE IN A LOW DEPRESSION IN THE CENTER OF A MEDIUM-SIZED
2295 DATA ROOM.
2700 DEF FNR(X)=INT(RND(1)*X)+1
2710 FOR I=1 TO 20:F=0
2720 FOR J=1 TO 4
2730 IF FNR(3)=2 OR TR(I,J) THEN 2770
2740 RO=FNR(20):IF RO=I THEN 2730
2750 DI=FNR(4):IF TR(RO,DI) THEN 2730
2760 TR(I,J)=RO:TR(RO,DI)=I
2770 F=F+TR(I,J)
2780 NEXT:IF F=0 THEN 2720
2790 NEXT
2800 YO=FNR(20)
2900 GU=FNR(20):IF GF THEN GU=-1
2910 EX=FNR(20)
2920 B1=FNR(20)
2930 B2=FNR(20)
2940 P1=FNR(20)
2950 P2=FNR(20)
2990 RETURN
3000 :
3010 REM *** PLAY ***
3020 :
3030 PRINT:PRINT RO*(YO)
3040 FOR I=1 TO 4:CO=TR(YO,I)
3050 IF CO=EX THEN GOSUB 60300:PRINT "EX IT NEARBY..."
3060 IF CO=GU THEN GOSUB 60300:PRINT "I SMELL THE GRUE!!!"
3070 IF CO=B1 OR CO=B2 THEN GOSUB 60300:PRINT "FLAP...FLAP...FLAP..."
3080 IF CO=P1 OR CO=P2 THEN GOSUB 60300:PRINT "I FEEL A DRAFT!!!"

```



```

3090 NEXT
3100 IF FNR(15)=4 THEN GOSUB 60300:PRINT
"<<<EARTHQUAKE>>":GOSUB 2900:GOTO 3000
3300 PRINT:AN$="":INPUT "MOVE OR SHOOT";
AN$:AN$=LEFT$(AN$,1)
3310 IF AN$="M" THEN 3400
3320 IF AN$="S" THEN 3500
3330 PRINT "ENTER 'M' OR 'S'":GOTO 3300
3400 PRINT:AN$="":INPUT "MOVE WHICH WAY"
;AN$:GOSUB 3900:IF I=0 THEN 3400
3410 IF TR(YO,I)=0 THEN GOSUB 60300:PRIN
T "YOU CANNOT GO THAT WAY...":GOTO 3000
3420 PRINT "OK...":YO=TR(YO,I)
3430 IF YO=EX THEN WL=0:GOTO 3990
3440 IF YO=GU THEN WL=1:GOTO 3990
3450 IF YO=P1 OR YO=P2 THEN WL=2:GOTO 39
90
3460 IF YO<>B1 AND YO<>B2 THEN 3000
3470 PRINT "BATS HAVE YOU NOW!!!":PRINT
"THEY'RE LIFTING YOU UP!!!"
3480 PRINT "OHHHHH, WHERE ARE WE NOW???"
:YO=FNR(20):GOTO 3000
3500 PRINT:AN$="":INPUT "SHOOT WHICH WAY
";AN$:GOSUB 3900:IF I=0 THEN 3500
3510 IF TR(YO,I) THEN 3540
3520 GOSUB 60300:PRINT "CLUNK!"
3530 PRINT "THE ARROW BOUNCED OFF THE WA
LL":GOTO 3000
3540 IF TR(YO,I)<>GU THEN 3580
3550 GOSUB 60300:PRINT "OUCH!!!"
3560 PRINT "YOU BAGGED A GRUE!!!"
3570 PRINT "NOW TO FIND THE WAY OUT...":
GF=1:GU=-1:GOTO 3000
3580 GOSUB 60300:PRINT "THE ARROW MISSED
THE GRUE!!!":GOTO 3000
3900 AN$=LEFT$(AN$,1):I=0
3910 IF AN$="N" THEN I=1
3920 IF AN$="E" THEN I=2
3930 IF AN$="S" THEN I=3
3940 IF AN$="W" THEN I=4
3950 IF I=0 THEN PRINT "ENTER 'N', 'S',
'E' OR 'W'"
3990 RETURN
4000 :
4010 REM *** END ***
4020 :
4030 IF WL OR GF=0 THEN 4070
4040 PRINT "YOU HAVE REACHED THE EXIT WI
TH YOUR"
4050 PRINT "GRUE!!! YOUR SUPPER WILL BE

```

```
FILLING"
4060 PRINT "TONIGHT FOR SURE!!!":GOTO 49
90
4070 IF WL THEN 4100
4080 PRINT "YOU HAVE REACHED THE EXIT WI
THOUT ANY"
4090 PRINT "GRUE!!! YOU ARE SURE TO STA
RVE...":GOTO 4990
4100 IF WL=2 THEN 4130
4110 PRINT "YOU BUMPED INTO THE GRUE!!!
HE ATE YOU"
4120 PRINT "BEFORE YOU COULD MOVE !!":GO
TO 4990
4130 PRINT "YOU FELL INTO A PIT !!! YOU
FELL A"
4140 PRINT "LOOOOOONG WAY..."
4990 RETURN
60300 :
60310 REM BELL SOUND ROUTINE
60320 :
60330 SD=54272:POKE SD+24,15
60340 POKE SD+5,7:POKE SD+6,0
60350 POKE SD+1,90:POKE SD+4,33
60360 FOR TS=1 TO 100:NEXT
60370 POKE SD+4,0:POKE SD+1,0
60390 RETURN
```





A clever premise overshadows an interesting game. The IR\$man serves as an excellent mathematical teaching device, while being entertaining and challenging. Everyone wants to beat the IRS, now here's your chance! To play, choose a number (we'll call it "X"), and the digits 1 through X will appear. Each time you remove a number from the list, all of the factors of that number (which are still on the list) go to the IR\$man. The object is to garner as much money as possible, while being as stingy as possible with the IR\$man. If you play the number 12, 1-2-3-4-5-6-7-8-9-10-11-12 will appear on the screen. If you begin play by selecting 12, the IR\$man will get 6, 2, 4, 3, and 1 ($6 \times 2 = 12$, $4 \times 3 = 12$, $1 \times 12 = 12$) for a score of 16 to your 12. The board will now look like this: 11 10 9 8 7 5. As you will note, the only remaining number on the list which has a factor is 10 (the remaining factor is 5). Remember, to remove a dollar amount (a number) from the list, there must be a factor to go to the IR\$man. When you remove 10 from the above example, the score will be 22 (12 + 10) for you, and 56 (21 + 11 + 9 + 8 + 7) for the IR\$man. If you choose 6 before choosing 12, the IR\$man scores for $3 \times 2 = 6$, (5 dollars) and 1×6 (1 dollar) for a total of 6 dollars for him and 6 dollars for you. The digits may only be used once, so 6, 3, 2, and 1 are removed from the list. Now when you choose 12, the IR\$man only gets 4 dollars ($4 \times 3 = 12$). Now the score is 18 (12 + 6) for you, and 10 (6 + 4) for the IR\$man. Note that 11 is stuck because the only factors of 11 are 11 and 1. Don't waste the universal factor (1) on just any number. It should be used first to remove the highest prime number from the list. A prime number is one that is only divisible by itself and 1. Examples are 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, etc. To circumvent the loss of 11, choose this number before choosing 6. There are many ways to thwart the IR\$man, but you must really try. Remember that all unused numbers are added to the score of the IR\$man at the end of the game. The maximum score you can achieve when choosing 1 through 12 is 48.

```

10 REM *****
11 REM ***          ***
12 REM ***      IRSMAN      ***
13 REM ***          ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 GOSUB 4000
90 END
1000 :
1010 REM *** INSTRUCTIONS ***
1020 :
1030 POKE 53280,6:POKE 53281,6
1040 VT$="{HOME} {CSD} {CSD} {CSD} {CSD} {CSD}
} {CSD} {CSD} {CSD} {CSD} {CSD} {CSD}
} {CSD} {CSD} {CSD} {CSD} {CSD} {CSD}
} {CSD} {CSD} {CSD}"
1050 GOSUB 1900
1060 PRINT "THIS IS THE GAME OF IRSMAN.
TO WIN, YOU"
1070 PRINT "TRY TO ACCUMULATE MORE MONEY
THAN YOUR"
1080 PRINT "NEMESIS, THE IRSMAN.":PRINT
1090 PRINT "GIVE ME A NUMBER BETWEEN 1 A
ND 50."
1100 PRINT "I WILL DISPLAY A CONSECUTIVE
NUMBER"
1110 PRINT "STRING STARTING AT 1, AND CO
NTINUING"
1120 PRINT "THROUGH TO THE NUMBER YOU SE
LECTED."
1130 PRINT "YOU WILL THEN CHOOSE HOW MUC
H MONEY"
1140 PRINT "(WHICH NUMBER) YOU WANT TO R
EMOVE FROM"
1150 PRINT "THE LIST.":PRINT
1160 GOSUB 1800:GOSUB 1900
1170 PRINT "BUT, AND HERE'S THE FUN PART
, THE IRSMAN";
1180 PRINT "GETS ALL OF THE REMAINING NU
MBERS ON THE";
1190 PRINT "LIST THAT ARE FACTORS OF THE
NUMBER YOU"
1200 PRINT "CHOSE. THAT IS HOW THE IRSMA
N GETS HIS"

```



```

1210 PRINT "MONEY. IF YOU CHOOSE 6, FOR
EXAMPLE,"
1220 PRINT "THE IRSMAN GETS ALL OF THE R
EMAINING"
1230 PRINT "FACTORS OF 6, POTENTIALLY 1,
2, AND 3.":PRINT
1240 PRINT "YOU CANNOT CHOOSE A NUMBER T
HAT HAS NO"
1250 PRINT "REMAINING FACTORS IN THE LIS
T, BECAUSE"
1260 PRINT "YOU MUST ALWAYS PAY THE IRSM
AN.":PRINT
1270 GOSUB 1800:GOSUB 1900
1280 PRINT "WHEN YOU CAN NO LONGER REMOV
E ANY OF THE";
1290 PRINT "REMAINING NUMBERS FROM THE L
IST, THE"
1300 PRINT "IRSMAN CLAIMS ALL OF THE UNU
SED MONEY"
1310 PRINT "(NUMBERS) FOR HIMSELF."
1800 PRINT LEFT$(VT$,24);"PRESS (RVS ON)
RETURN(RVS OFF) WHEN READY TO CONTINUE:
(RVS ON) "
1810 GET AN$:IF AN$="" THEN 1810
1890 RETURN
1900 PRINT "{CLR}{WHT}{CSD}{CSD}";TAB(13
);"*** IRSMAN ***":PRINT:PRINT
1990 RETURN
2000 :
2010 REM *** SETUP ***
2020 :
2030 DIM LI(50)
2040 FOR I=1 TO 50:LI(I)=I:NEXT
2050 INPUT "{CLR}{CSD}{CSD}HOW MANY NUMB
ERS (1-50) IN THE LIST";AN$
2060 AN=VAL(AN$):IF AN>=1 AND AN<=50 AND
AN=INT(AN) THEN 2080
2070 ER$="<<< USE A NUMBER FROM 1 TO 50
>>>":GOSUB 3700:GOTO 2050
2080 NU=AN
2990 RETURN
3000 :
3010 REM *** PLAY ***
3020 :
3030 GOSUB 1900
3040 PRINT LEFT$(VT$,5)
3050 PRINT "HERE IS THE LIST:":PRINT
3060 FOR I=1 TO NU
3070 IF LI(I)=0 THEN PRINT " ";
3080 IF LI(I)<10 THEN PRINT " ";

```



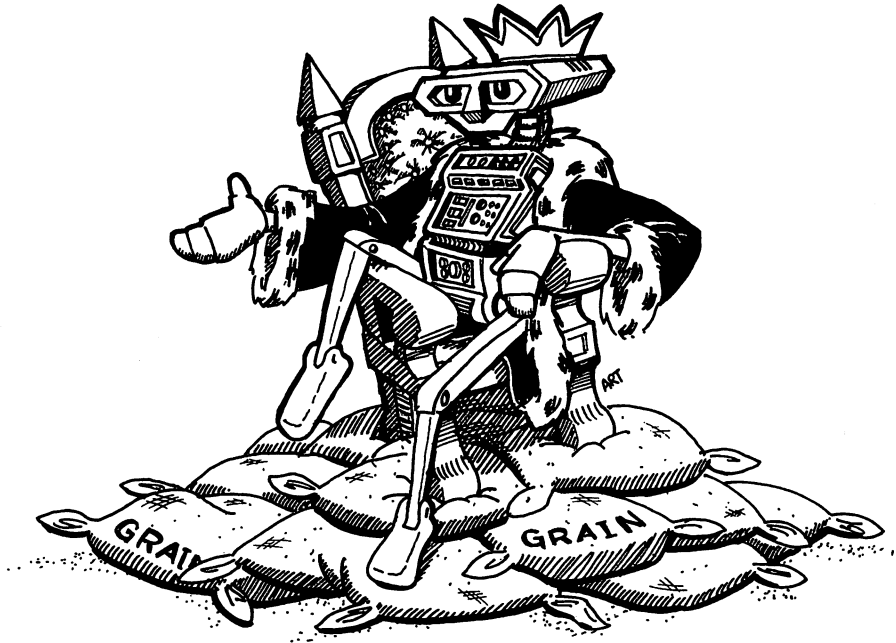
```

3090 IF LI(I) THEN PRINT I;
3100 NEXT I
3110 IF NU=1 THEN 3900
3120 FOR I=2 TO NU:IF LI(I)=0 THEN 3600
3130 FOR J=1 TO I:IF LI(J)=0 THEN 3500
3140 IF J=I THEN 3500
3150 IF LI(I)/J<>INT(LI(I)/J) THEN 3500
3160 PRINT LEFT$(VT$,13)
3170 PRINT "THE SCORE IS:   IRSMAN:";TA
3180 PRINT "                   *YOU*:";YD
3190 AN$="":PRINT LEFT$(VT$,16):INPUT "W
HIGH DO YOU WANT";AN$:AN=VAL(AN$)
3200 IF AN<1 OR AN>NU THEN 3220
3210 IF LI(AN) AND AN=INT(AN) THEN 3230
3220 ER$="THAT IS NOT AVAILABLE!":GOSUB
3700:GOTO 3190
3230 SC=0:IF AN=1 THEN 3280
3240 FOR K=1 TO AN:IF LI(K)=0 THEN 3270
3250 IF K=AN THEN 3270
3260 IF AN/K=INT(AN/K) THEN SC=SC+K
3270 NEXT K
3280 IF SC<>0 THEN 3310
3290 ER$="THAT LEAVES NOTHING FOR THE IR
SMAN."
3300 ER$="YOU CAN'T HAVE IT.
"+ER$:GOSUB 3700:GOTO 3190
3310 LI(AN)=0:YD=YD+AN:TA=TA+SC
3320 FOR K=1 TO AN:IF LI(K)=0 THEN 3340
3330 IF AN/K=INT(AN/K) THEN LI(K)=0
3340 NEXT K
3350 GOTO 3040
3500 NEXT J
3600 NEXT I:GOTO 3990
3700 GOSUB 60300
3710 PRINT LEFT$(VT$,20);ER$
3720 FOR K=1 TO 1600:NEXT
3730 PRINT LEFT$(VT$,20);:FOR K=1 TO LEN
(ER$):PRINT " ";:NEXT
3790 RETURN
3900 PRINT "{CSD}{CSD}OOOOPS, YOU CAN'T
GET ANYTHING...":TA=1:LI(1)=0:GOSUB 1800
3990 RETURN
4000 :
4010 REM *** END ***
4020 :
4025 GOSUB 1900
4030 PRINT "*** THE GAME IS OVER ***"
4040 PRINT
4050 TA=0:FOR I=1 TO NU:TA=TA+I:NEXT:TA=
TA-YD

```

```
4060 PRINT "THE IRSMAN:";TA
4070 PRINT "          YOU:";YO
4080 PRINT "===== ":PRINT
4090 IF TA>YO THEN PRINT "THE IRSMAN IS
THE WINNER!!!"
4100 IF TA<YO THEN PRINT "YOU HAVE BEATE
N THE IRSMAN!!!"
4110 IF TA=YO THEN PRINT "IT'S UNBELIEVA
BLE BUT IT'S A TIE!!!"
4120 GOSUB 60300:GOSUB 60300:GOSUB 60300
4990 RETURN
60300 :
60310 REM BELL SOUND ROUTINE
60320 :
60330 SD=54272:POKE SD+24,15
60340 POKE SD+5,7:POKE SD+6,0
60350 POKE SD+1,50:POKE SD+4,17
60360 FOR TS=1 TO 100:NEXT
60370 POKE SD+4,0:POKE SD+1,0
60390 RETURN
```





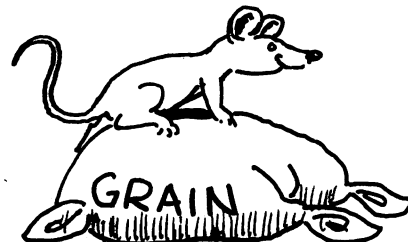
Kingdom

This game is designed to test your leadership ability. You are given a ten year reign, during which you try to guide your kingdom towards health and prosperity. There are certain conditions which are beyond your control, such as the success of the harvest, plagues, or damage by rats. Try to do the best job possible. When the price of acreage is high, (25 or 26 bushels per acre), you may choose to become a land broker instead of a gentleman farmer. That is, you may sell all but one acre of land (you must keep 1), and hope the price of land drops the following year. If the price of land drops by 4 bushels (say, from 26 to 22), you have, in effect made a 4 bushel per acre profit. When the price of land is low, (below 20 bushels an acre), it is recommended that you buy as much land as possible, while retaining enough grain to feed your people and sow your fields. You can easily understand the program by manipulating the beginning values of the variables in line 2010.

```

10 REM *****
11 REM ***          ***
12 REM ***   KINGDOM   ***
13 REM ***          ***
14 REM *****
20 GOSUB 1000:REM INSTS
30 GOSUB 2000:REM SETUP
40 GOSUB 3000:REM PLAY!
50 GOSUB 4000:REM END
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 POKE 53280,0:POKE 53281,0:PRINT"(CL
R)(WHT)";
1015 VT$="{HOME}{CSD}{CSD}{CSD}{CSD}{CSD
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD
}{CSD}{CSD}{CSD}"
1020 PRINT LEFT$(VT$,3);TAB(12);"*** KIN
GDOM ***"
1030 PRINT LEFT$(VT$,6)
1040 PRINT "THIS IS A SIMULATION OF THE
COUNTRY OF"
1050 PRINT "SUMERIA. YOU ARE THE SOVERE
IGN RULER"
1060 PRINT "AND YOU WILL GOVERN FOR TEN
YEARS."
1070 PRINT : PRINT "THE DECISIONS THAT Y
OU MAKE WILL AFFECT"
1080 PRINT "THE LIVES OF HUNDREDS OF PEO
PLE. YOUR"
1090 PRINT "DICTATORIAL SKILLS WILL BE R
ATED ONCE"
1100 PRINT "YOUR REIGN HAS ENDED."
1110 PRINT : PRINT "YOU WILL BE ASKED TO
MAKE SEVERAL KEY"
1120 PRINT "DECISIONS EACH YEAR. EACH OF
THESE"
1130 PRINT "WILL BE EXPLAINED TO YOU."
1140 PRINT LEFT$(VT$,23)
1150 PRINT "PRESS ANY KEY TO CONTINUE"
1160 GET ANS$:IF ANS$="" THEN 1160
1990 RETURN
2000 :
2001 REM *** SETUP

```



```

2002 :
2010 P=95:S=2800:H=3000:E=H-S:Y=3:A=H/Y:
I=5:D=0:Z=0:Q=1
2020 DIM NU$(11): FOR J=1 TO 11: READ NU
$(J):NEXT
2030 DATA FIRST,SECOND,THIRD,FOURTH,FIFT
H,SIXTH
2040 DATA SEVENTH,EIGHTH,NINTH,TENTH,ELE
VENTH
2990 RETURN
3000 :
3001 REM *** PLAY!
3002 :
3005 PRINT "{CLR}";LEFT$(VT$,3);TAB(12);
"*** KINGDOM ***";LEFT$(VT$,6)
3010 Z=Z+1:PRINT:PRINT "HAMURABI, I BEG
TO REPORT TO YOU:"
3020 PRINT "IN THE "NU$(Z)" YEAR,"D"PEOP
LE STARVED."
3030 PRINT "A TOTAL OF"I"CAME TO THE CIT
Y."
3100 P=P+I:IF Q=0 THEN P=INT(P/2):PRINT:
PRINT"A HORRIBLE PLAGUE STRUCK! HALF ";
3110 IF Q=0 THEN PRINT "OF YOUR PEOPLE P
ERISHED."
3120 PRINT
3130 PRINT "THE POPULATION IS"P"
3131 PRINT "THE CITY OWNS"A"ACRES.":PRIN
T"YOU HARVESTED"Y"BUSHEL PER ACRE."
3232 PRINT "RATS ATE A TOTAL OF"E"BUSHEL
S":PRINT"YOU HAVE"S"BUSHEL IN RESERVE."
3250 IF Z=11 THEN RETURN
3400 C=INT(RND(1)*10):Y=C+17
3410 PRINT:PRINT"LAND IS TRADING AT"Y"BU
SHEL PER ACRE."
3411 INPUT"HOW MANY ACRES DO YOU WISH TO
BUY";Q
3420 IF Q<0 THEN PRINT "HAMURABI, YOU CA
NNOT DO THAT. IF YOU WISH TO SELL ";
3421 IF Q<0 THEN PRINT "LAND, BUY 0 ACRE
S FIRST.":GOTO 3410
3425 IF Y*Q=S THEN PRINT"SIR, KEEP AT LE
AST ONE BUSHEL TO PLANT.":GOTO 3410
3430 IF Y*Q>S THEN PRINT"HAMURABI, THINK
AGAIN! YOU ONLY HAVE"
3431 IF Y*Q>S THEN PRINT S"BUSHEL OF GR
AIN.":GOTO 3410
3450 IF Q>0 THEN A=A+Q:S=S-Y*Q:C=0:GOTO
3500
3460 PRINT:INPUT"HOW MANY ACRES WOULD YO

```

```

U SELL";Q
3465 IF Q<0 THEN PRINT"HAMURABI, YOU CAN
NOT DO THAT. IF YOU"
3466 IF Q<0 THEN PRINT"DON'T WANT TO SEL
L, ENTER 0 ACRES.":GOTO 3460
3468 IF A-Q=0 THEN PRINT"YOU MUST KEEP A
T LEAST ONE ACRE OF LAND":GOTO 3460
3480 A=A-Q:S=S+Y*Q:C=0
3500 PRINT:PRINT"OF THE"S"BUSHELS REMAIN
ING, HOW MANY"
3501 INPUT"DO YOU WISH TO FEED YOUR PEOP
LE";Q
3502 IF S-Q=0 THEN PRINT"SIR, KEEP AT LE
AST ONE BUSHEL TO PLANT":GOTO 3500
3505 IF Q<1 THEN PRINT:PRINT"HAMURABI, T
HE PEOPLE WILL STARVE!!"
3506 IF Q<1 THEN PRINT"YOU MUST FEED THE
M SOMETHING.":GOTO 3500
3510 IF Q>S THEN PRINT"HAMURABI, YOU ONL
Y OWN"S"BUSHELS...":GOTO 3500
3520 S=S-Q:C=1
3525 PRINT:PRINT"YOU NOW HAVE"S"BUSHELS"
3530 PRINT:PRINT"OF THE"A"ACRES YOU OWN,
HOW MANY"
3531 PRINT"DO YOU WISH TO PLANT WITH SEE
D?"
3532 INPUT"(PLANT 2 ACRES WITH ONE BUSHE
L)";D
3535 IF D<1 THEN PRINT:PRINT"HAMURABI, Y
OU MUST PLANT SOMETHING SO"
3536 IF D<1 THEN PRINT"THERE WILL BE FOO
D FOR NEXT YEAR":GOTO 3530
3540 IF D>A THEN PRINT "SIR, YOU ONLY HA
VE"A"ACRES.":GOTO 3530
3550 IF D>10*P THEN PRINT"SIR, ONE PERSO
N CAN ONLY WORK TEN ACRES"
3551 IF D>10*P THEN PRINT"YOUR POPULATIO
N OF"P"IS NOT BIG ENOUGH.":GOTO 3530
3553 IF D/2>S THEN PRINT"HAMURABI, THAT
IS TOO MUCH TO PLANT":GOTO 3530
3555 S=S-INT(D/2):C=INT(RND(1)*5)+1
3600 Y=C:H=D*Y:E=0:C=INT(RND(1)*5)+1:IF
INT(C/2)*2=C THEN E=INT(S/C)
3610 S=S-E+H:C=INT(RND(1)*5)+1:I=INT(C*(
20*A+S)/P/100+1):C=INT(Q/20)
3611 Q=INT(10*(2*RND(1)-.3)):IF P<C THEN
D=0:GOTO 3010
3615 D=P-C:IF D>.5*P THEN 3630
3620 P1=((Z-1)*P1+D*100/P)/Z:P=C:D1=D+D
:GOTO 3010

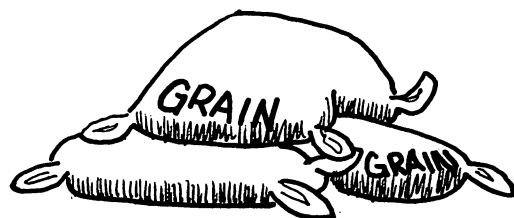
```



```

3630 PRINT:PRINT"YOU STARVED"D"PEOPLE IN
  ONE YEAR!!"
3631 PRINT"YOU HAVE DONE SUCH A MISERABL
  E JOB THAT"
3632 PRINT"YOU HAVE BEEN OVERTHROWN AND
  DEPOSED!"
3633 WL=1:RETURN
4000 :
4001 REM *** END
4002 :
4005 IF WL THEN RETURN:REM ENDS PROGRAM
  IF RULER WAS DEPOSED
4010 PRINT:PRINT"IN YOUR TEN YEARS OF RU
  LE,"P1"%"
4020 PRINT"OF THE POPULATION STARVED PER
  YEAR, ON THE AVERAGE."
4030 PRINT"A TOTAL OF"D1"PEOPLE DIED.":L
  =A/P
4040 PRINT
4050 PRINT"YOU STARTED WITH 10 ACRES PER
  PERSON"
4060 PRINT"AND ENDED WITH"L"ACRES"
4061 PRINT"PER PERSON."
4070 IF P1>33 OR L<7 THEN PRINT"YOU ARE
  A DISGRACE! THE PEOPLE HAVE"
4071 IF P1>33 OR L<7 THEN PRINT"EXILED Y
  OU TO A REMOTE ISLAND.":RETURN
4080 IF P1>10 OR L<9 THEN PRINT"YOU RULE
  LIKE THE AYATOLLAH! MOST OF"
4081 IF P1>10 OR L<9 THEN PRINT"YOUR SUBJ
  ECTS WOULD DANCE ON YOUR GRAVE.":RETURN
4090 IF P1>3 OR L<10 THEN PRINT"YOU COUL
  D HAVE DONE BETTER."INT(P*.8*RND(1))
4091 IF P1>3 OR L<10 THEN PRINT"PEOPLE W
  OULD LOVE TO ASSASSINATE YOU!":RETURN
4095 PRINT"A GREAT JOB!! YOU CAN RULE MY
  COUNTRY"
4096 PRINT"ANYTIME YOU WANT!!!":RETURN

```





This game can be frustrating, challenging, and exciting all at the same time. You are given a list of integers, which you must unscramble using a reversing technique. The list is DIMensioned by line 2030, and values are put into LI(1) through LI(9) in line 2040 using a FOR/NEXT loop. LI(1) becomes 1, LI(2) becomes 2, etc. Line 2050 assigns the random order to the list. Line 3050 does the reversing of the numbers that you select as ANS in line 3040. These lines have several commands grouped together for speed of operation. If you want to dissect them, it is best to rewrite the line as separate statements. You can also print out the variables:

```

3050 MDL = INT ((9 - ANS)/2)
3051 PRINT "MDL IN LINE 3051 ="; MDL
3052 FOR I = ANS TO ANS + MDL
3053 PRINT "I ="; I; " ANS + MDL ="; ANS + MDL
3054 T = LI(I)
3055 PRINT "T ="; T
3056 LI(I) = LI(9 + ANS - I)
3057 PRINT "LI(I) ="; LI(I)
3058 LI(9 + ANS - I) = T
3059 PRINT "T ="; T : NEXT

```

This "expanded" version of line 3050 will let you watch the program pass the variables using "T" as a temporary storage location. Practically all the action takes place in this one line. When you run this revised program, print statements will tell you what happens each time you make a reversal. The reason for combining multiple statements on a single line is execution speed. When you are developing programs, you should have every statement on a

separate line. Once you have the program running you should make two versions. Make a long version with lots of REMs and descriptive variable name, and a short version with combined lines. This will help you modify the program later on. Your own program that you knew by heart a few weeks ago can become a complete mystery if you don't spend enough time on REMs and organization.

```

10 REM *****
11 REM ***           ***
12 REM ***   REVERSER   ***
13 REM ***           ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 GOSUB 4000
60 END
1000 :
1010 REM *** INSTRUCTIONS ***
1020 :
1030 POKE 53280,6:POKE 53281,6
1040 VT$="{HOME}<CSD><CSD><CSD><CSD><CSD><CSD>
}<CSD><CSD><CSD><CSD><CSD><CSD><CSD><CSD>
}<CSD><CSD><CSD><CSD><CSD><CSD><CSD><CSD>
}<CSD><CSD><CSD>":
1050 GOSUB 1900
1060 PRINT "IN THIS GAME YOU ARE GIVEN A
LIST OF"
1070 PRINT "NUMBERS FROM 0 TO 9. THE LIS
T WILL NOT"
1080 PRINT "BE IN SEQUENCE. IT IS YOUR J
OB TO SORT"
1090 PRINT "THE LIST INTO ASCENDING ORDE
R.":PRINT
1100 PRINT "YOU ARRANGE THE LIST BY REVE
RSING THE"
1110 PRINT "ORDER OF IT. YOU INPUT THE S
TARTING"
1120 PRINT "COLUMN THAT IS TO BE REVERSE
D, AND THAT"
1130 PRINT "COLUMN ALL THE WAY THROUGH T
O COLUMN 9,"
1140 PRINT "WILL BE REVERSED."
1150 GOSUB 1800:GOSUB 1900
1160 PRINT "IF YOU HAD THIS LIST:":PRINT
1170 PRINT "POSITIONS: 0 1 2 3 4 5 6 7 8
9"

```



```

1180 PRINT "          -----
--"
1190 PRINT "          LIST: 0 1 9 8 7 6 5 2 3
4":PRINT

1200 PRINT "AND YOU REVERSED IT POSITION
7, IT WOULD";
1210 PRINT "LOOK LIKE THIS:":PRINT
1220 PRINT "POSITIONS: 0 1 2 3 4 5 6 7 8
9"
1230 PRINT "          -----
--"
1240 PRINT "          LIST: 0 1 9 8 7 6 5 4 3
2":PRINT
1250 GOSUB 1800:GOSUB 1900
1260 PRINT "A FINAL REVERSAL AT POSITION
2 WOULD"
1270 PRINT "COMPLETE THE LIST AS THIS:":
PRINT
1280 PRINT "POSITIONS: 0 1 2 3 4 5 6 7 8
9"
1290 PRINT "          -----
--"
1300 PRINT "          LIST: 0 1 2 3 4 5 6 7 8
9":PRINT
1310 PRINT "YOU WIN WHEN THE LIST IS SOR
TED IN"
1320 PRINT "ASCENDING ORDER AS IS SHOWN
ABOVE.":PRINT
1330 PRINT "GOOD LUCK !!!"
1800 PRINT LEFT$(VT$,24);"PRESS {RVS ON}
RETURN{RVS OFF} WHEN READY TO CONTINUE:
{RVS ON} "
1810 GET AN$:IF AN$=""THEN 1810
1890 RETURN
1900 PRINT "{CLR}{WHT}{CSD}{CSD}";TAB(12
);"*** REVERSER ***":PRINT:PRINT
1990 RETURN
2000 :
2010 REM *** SETUP ***
2020 :
2030 DIM LI(9)
2040 FOR I=0 TO 9:LI(I)=I:NEXT
2050 FOR I=0 TO 9:X=INT(RND(1)*10):T=LI(
I):LI(I)=LI(X):LI(X)=T:NEXT
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 GOSUB 1900
3020 PRINT "POSITIONS: 0 1 2 3 4 5
6 7 8 9"

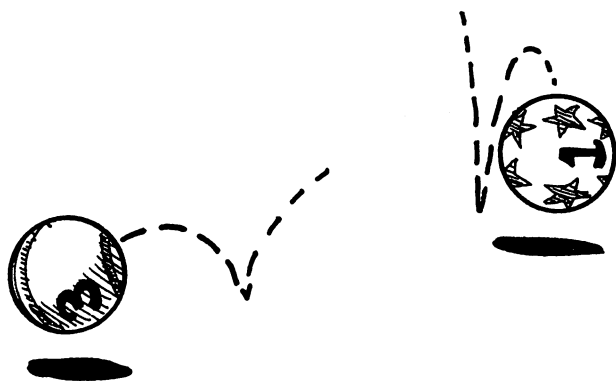
```

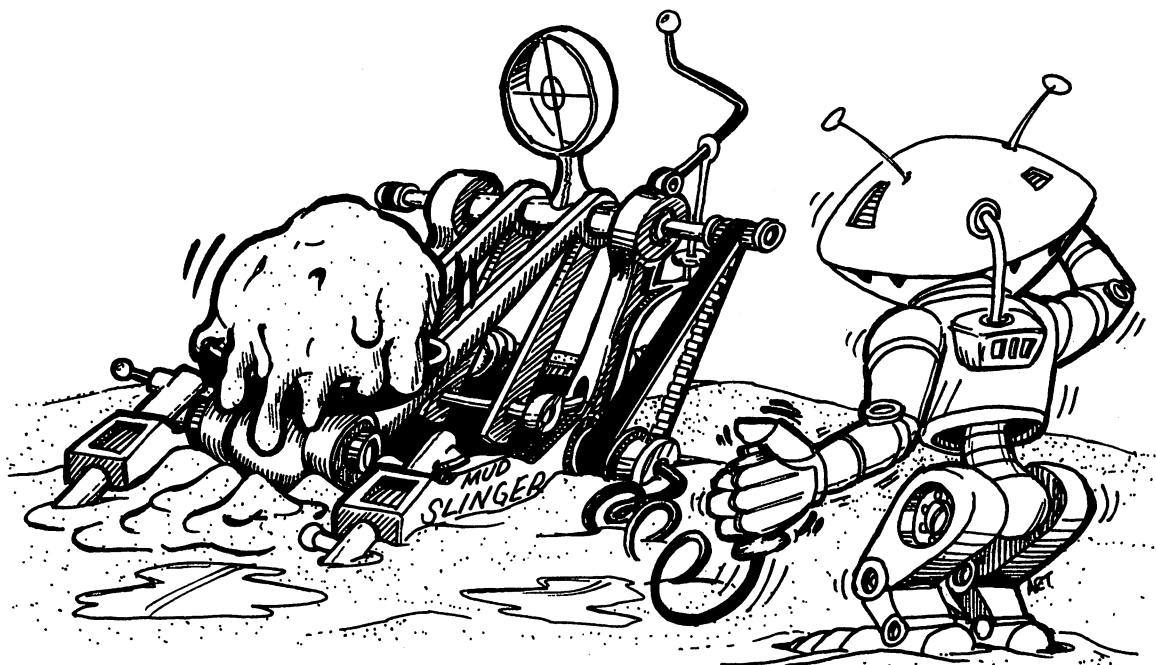


```

3021 PRINT "          -----"
-----"
3022 PRINT "          LIST:";
3030 FOR I=0 TO 9:PRINT LI(I);:NEXT:PRIN
T
3035 FOR I=0 TO 9:IF LI(I)=I THEN NEXT:R
ETURN
3040 PRINT:INPUT "REVERSE AT WHICH POSIT
ION (0-9)";AN$:AN=VAL(AN$)
3041 IF AN$<"0" OR AN$>"9" THEN PRINT "T
YPE A NUMBER BETWEEN 0 AND 9":GOTO 3040
3050 MDL=INT((9-ANS)/2):FORI=ANTOAN+MDL:
T=LI(I):LI(I)=LI(9+AN-I):LI(9+AN-I)=T
3051 NEXT
3060 MOVE=MOVE+1:GOTO3010
4000 :
4001 REM *** END
4002 :
4010 PRINT:PRINT"YOU DID IT!":PRINT:PRIN
T"IT TOOK YOU A TOTAL OF"MO"MOVES."
4020 PRINT
4030 IF MO<11 THEN PRINT "THAT'S SUPER!!
":RETURN
4031 IF MO<14 THEN PRINT "GOOD JOB!":RET
URN
4032 IF MO<17 THEN PRINT "THAT WAS JUST
SO-SO.":RETURN
4033 IF MO<20 THEN PRINT "NOT SO HOT.":R
ETURN
4034 IF MO>19 THEN PRINT "BACK TO SCHOOL
!":RETURN
4040 PRINT"GOOD JOB!"
4990 RETURN

```





Schmoo

Its whimsical name and humorous object camouflage an excellent thinking-man's game. It is hard to imagine how a person could play, and not come away with a better understanding of the Cartesian coordinate system. The premise is that you are trying to splat a mudball onto the mud-loving Schmoo. The elevation at which you aim the automatic mudball slinger determines how far the mudball will travel. The angle at which you shoot will be determined by the coordinates of the Schmoo. Below is a list of coordinates and the angles they represent:

X	Y	ANGLE
		0
12588	0	
17866	17866	45
30910	0	90
5888	-5888	135
0	-9400	180
-25727	-25727	225
-18992	0	270
-34101	34101	315

This chart should help you to understand how various coordinates relate to angles and how best to aim your mudball slinger.

```

10 REM *****
11 REM ***          ***
12 REM ***      SCHMOO      ***
13 REM ***          ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 GOSUB 4000
60 END
1000 :
1010 REM *** INSTRUCTIONS ***
1020 :
1030 VT$="(HOME) {CSD} {CSD} {CSD} {CSD} {CSD}
) {CSD} {CSD} {CSD} {CSD} {CSD} {CSD} {CSD} {CSD}
) {CSD} {CSD} {CSD} {CSD} {CSD} {CSD} {CSD} {CSD}
) {CSD} {CSD} {CSD}"
1040 POKE 53280,6:POKE 53281,6
1050 GOSUB 1900
1060 PRINT "THIS IS THE GAME OF SCHMOO.
IN IT YOU"
1070 PRINT "THROW MUD AROUND IN HOPES OF
HITTING"
1080 PRINT "THE MUD-LOVING SCHMOO.":PRIN
T
1090 PRINT "YOU ARE SITUATED IN THE CENT
ER OF AN"
1100 PRINT "X,Y COORDINATE SYSTEM AT POS
ITION 0,0."
1110 PRINT "THE SCHMOO WILL BE LOCATED S
OMEWHERE ON"
1120 PRINT "THE SAME PLANE. HIS COORDINA
TES ARE"
1130 PRINT "GIVEN TO YOU BEFORE EACH TUR
N.":PRINT
1140 PRINT "YOU HAVE YOUR TRUSTY AUTOMAT
IC MUDBALL"
1150 PRINT "SLINGER WHICH YOU USE TO TOS
S MUDBALLS"
1160 PRINT "AT THE SCHMOO. YOU ENTER THE
ELEVATION"
1170 PRINT "AND THE ANGLE AT WHICH YOU W
ISH TO FIRE"
1180 PRINT "THE MUDBALL. AFTER EACH SHOT
YOU WILL"

```



```

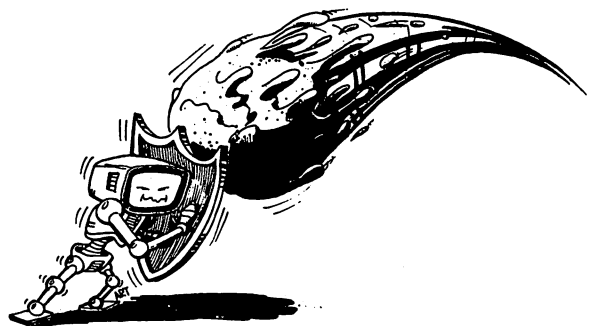
1190 PRINT "BE GIVEN THE COORDINATES WHERE THE MUD LANDED."
1200 GOSUB 1800:GOSUB 1900
1210 PRINT "FOR EXAMPLE, IF THE SCHMOO'S COORDINATES";
1220 PRINT "ARE (-500 , 1000 ) THEN THE SCHMOO IS"
1230 PRINT "ABOUT FIVE HUNDRED FEET TO YOUR LEFT AND";
1240 PRINT "A THOUSAND FEET IN FRONT OF YOU.":PRINT
1250 PRINT "THE ELEVATION FOR THE SHOT WOULD BE"
1260 PRINT "ABOUT 89.3 DEGREES WHILE THE ANGLE WHERE";
1270 PRINT "THE SCHMOO CAN BE FOUND IS ABOUT 333.4"
1280 PRINT "DEGREES.":PRINT
1290 PRINT "THE MUDBALLS ARE LARGE ENOUGH TO MUDDY"
1300 PRINT "THE SCHMOO AS LONG AS THEY LAND AND WITHIN"
1310 PRINT "100 FEET OF HIM.":PRINT
1320 PRINT "NOW THAT YOU KNOW HOW TO MAKE THE SCHMOO";
1330 PRINT "HAPPY, GO GET HIM. GOOD LUCK!"
1800 PRINT LEFT$(VT$,24);"PRESS {RVS ON} RETURN{RVS OFF} WHEN READY TO CONTINUE: {RVS ON} "
1810 GET AN$:IF AN$="" THEN 1810
1890 RETURN
1900 PRINT "{CLR}{WHT}{CSD}{CSD}";TAB(12);"*** SCHMOO ***":PRINT:PRINT
1990 RETURN
2000 :
2010 REM *** SET UP
2020 :
2030 PRINT "{CLR}"
2040 S1=INT(RND(1)*2)*2-1
2050 S2=INT(RND(1)*2)*2-1
2060 SX=(INT(RND(1)*26000)+5000)*S1
2070 SY=(INT(RND(1)*26000)+5000)*S2
2080 CN=3.141592635/180
2990 RETURN
3000 :
3010 REM *** PLAY ***
3020 :
3030 GOSUB 1900
3040 GOSUB 60300
3045 PRINT:PRINT "THE SCHMOO IS AT COORD

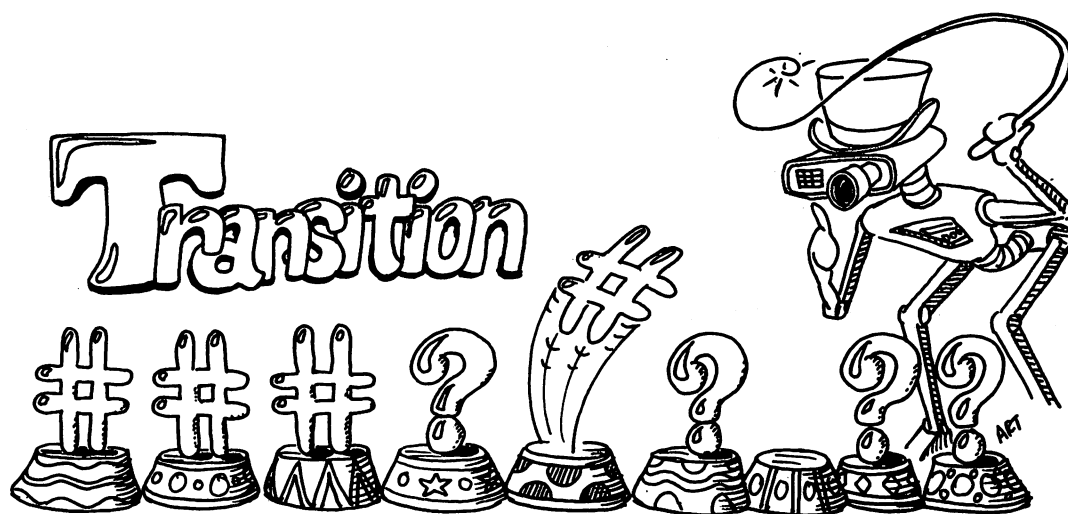
```

```

INATES:":PRINT "(";SX;",";SY;")"
3050 PRINT:PRINT "WHAT ELEVATION FOR THE
MUDBALL":INPUT "SLINGER (1-90)";EL
3060 IF EL<1 OR EL>90 THEN PRINT "ELEVAT
ION RANGES FROM 1 TO 90.":GOTO 3050
3070 IF EL<90 THEN 3100
3080 PRINT "THAT WOULD SHOOT THE MUD STR
AIGHT UP"
3090 PRINT "AND IT WOULD LAND ON TOP OF
YOU.":GOTO 3050
3100 PRINT:PRINT "WHAT ANGLE OF DIRECTIO
N FOR THE"
3110 INPUT "MUDBALL SLINGER (0-360)";AN
3120 IF AN<0 OR AN>360 THEN PRINT "THE A
NGLE RANGES FROM 0 TO 360.":GOTO 3100
3130 DM=ABS(INT(93000*SIN(EL*CN)*COS(EL*
CN)))
3140 XM=DM*SIN(AN*CN)
3150 YM=DM*COS(AN*CN)
3160 DS=SQR((SX-XM)^2+(SY-YM)^2)
3170 PRINT:PRINT "THE MUD SPLATTERED AT
COORDINATES:"
3180 PRINT "(";INT(XM);",";INT(YM);")"
3190 TRY=TRY+1
3200 IF DS>100 THEN 3040
3210 GOSUB 60300:GOSUB 60300:GOSUB 60300
3220 PRINT:PRINT "THAT'S GOOD ENOUGH TO
SPLATTER THE"
3230 PRINT "SCHMO!!!"
3990 RETURN
4000 :
4010 REM *** END ***
4020 :
4030 PRINT:PRINT "YOU SPLATTED THE SCHMO
D IN";TRY;"TRIES.":PRINT
4040 PRINT "DO YOU WISH TO PLAY AGAIN? {
RVS ON} {RVS OFF}";
4050 GET AN$:IF AN$="" THEN 4050
4060 IF LEFT$(AN$,1)="Y" THEN RUN
4070 PRINT "{CSL}N":PRINT "{CSD}{CSD}THA
NKS FOR SPLATTERING THE SCHMO!!!"
4990 RETURN
60300 :
60310 REM BELL SOUND ROUTINE
60320 :
60330 SD=54272:POKE SD+24,15
60340 POKE SD+5,7:POKE SD+6,0
60350 POKE SD+1,50:POKE SD+4,17
60360 FOR TS=1 TO 100:NEXT
60370 POKE SD+4,0:POKE SD+1,0
60390 RETURN

```



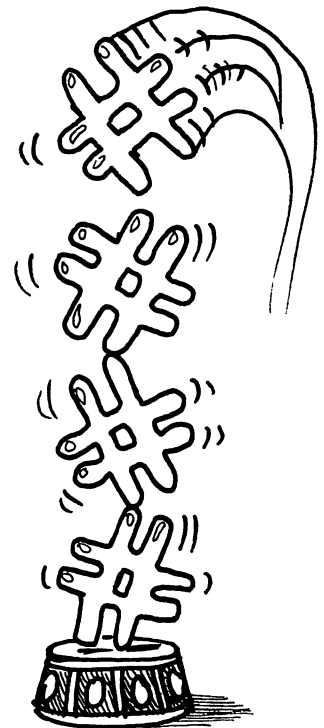


Impossible! It may seem impossible, but it's not. Deriving the key to this challenging game is very satisfying indeed! The object of the game is to transpose this list, # # # # . ? ? ? ? so that it looks like this ? ? ? ? . # # # #. The rules are few. The question marks (?) can only be moved to the left. Pound signs (#) can only be moved to the right. Either sign may be moved during a turn, with the following limitations. A sign may only be moved into the place currently occupied by the period (this space is referred to as the blank or empty space). A move is made by moving directly into an empty space or by jumping over one opposing piece into the empty space. To understand the function of lines in the play section, separate compound lines as you did for REVERSER.

```

10 REM *****
11 REM ***          ***
12 REM ***  TRANSITION  ***
13 REM ***          ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
50 GOSUB 4000
90 END
1000 :
1010 REM *** INSTRUCTIONS ***
1020 :
1030 POKE 53280,6:POKE 53281,6
1040 VT$="{HOME}{CSD}{CSD}{CSD}{CSD}{CSD}
      {CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
      {CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
      {CSD}{CSD}{CSD}"
1050 GOSUB 1900
1060 PRINT "THE GAME OF TRANSITION WILL
PRESENT YOU"
1070 PRINT "WITH A LIST OF NINE DIGITS.
THE LIST"
1080 PRINT "WILL LOOK LIKE THIS:"
1090 PRINT LEFT$(VT$,13);
1100 PRINT TAB(5);"[ 1  2  3  4  5  6  7
      B  9 ]":PRINT
1105 PRINT TAB(5);"[ #  #  #  #  .  ?  ?
      ?  ? ]"
1110 GOSUB 1800:GOSUB 1900
1120 PRINT "THE '#' CHARACTER CAN ONLY M
OVE TO THE"
1130 PRINT "RIGHT, AND THE '?' CHARACTER
CAN ONLY"
1140 PRINT "MOVE TO THE LEFT.":PRINT
1150 PRINT "A MOVE IS MADE BY MOVING TO
AN EMPTY"
1160 PRINT "SPACE, OR BY JUMPING OVER ON
E OPPOSING"
1170 PRINT "PIECE.":PRINT
1180 PRINT "TO MAKE A MOVE, YOU ENTER TH
E POSITION"
1190 PRINT "NUMBER OF THE MOVING PIECE.
TO QUIT,"

```

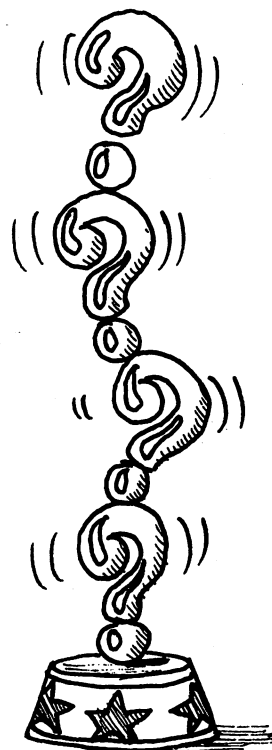


```

1200 PRINT "ENTER ZERO (0). "
1800 PRINT LEFT$(VT$,24);"PRESS {RVS ON}
RETURN{RVS OFF} WHEN READY TO CONTINUE:
{RVS ON} "
1810 GET AN$:IF AN$="" THEN 1810
1890 RETURN
1900 PRINT "{CLR}{WHT}{CSD}{CSD}";TAB(11
);"*** TRANSITION ***":PRINT:PRINT
1990 RETURN
2000 :
2010 REM *** SETUP ***
2020 :
2030 DIM LI(9)
2040 FOR I=1 TO 4:LI(I)=1:LI(10-I)=2
2050 NEXT:NM=0
2990 RETURN
3000 :
3010 REM *** PLAY ***
3020 :
3030 GOSUB 1900
3040 PRINT LEFT$(VT$,8);TAB(5);"[";:FOR
I=1 TO 9:PRINT I;:NEXT:PRINT "]" :PRINT
3050 PRINT TAB(5);"[";:FOR I=1 TO 9:PRIN
T MID$(" . # ? ",LI(I)*3+1,3);:NEXT
3060 PRINT "]" :PRINT:PRINT
3070 IF LI(5) THEN 3090
3080 WL=0:FOR I=1 TO 4:WL=WL+(LI(I)=2)+(
LI(I+5)=1):NEXT:IF WL<=-8 THEN 3990
3090 PRINT TAB(11);"#----> <----?":P
RINT
3100 PRINT TAB(13);"MOVE (0-9): {RVS ON}
{RVS OFF}";
3110 GET AN$:IF AN$<"0" OR AN$>"9" THEN
3110
3120 IF AN$="0" THEN WL=0:GOTO 3990
3130 AN=VAL(AN$):PRINT "{CSL}";AN$
3140 IF LI(AN)=0 THEN ER$="THAT SPACE IS
EMPTY.":GOTO 3220
3150 IF LI(AN)=1 THEN DI=1
3160 IF LI(AN)=2 THEN DI=-1
3170 IF AN+DI>9 OR AN+DI<1 THEN 3210
3180 IF LI(AN+DI)=0 THEN LI(AN+DI)=LI(AN
):GOTO 3250
3190 IF AN+DI+DI<1 OR AN+DI+DI>9 THEN 32
10
3200 IF LI(AN+DI)<>LI(AN) AND LI(AN+DI+D
I)=0 THEN LI(AN+DI+DI)=LI(AN):GOTO 3250
3210 ER$="IT CANNOT MOVE FURTHER."
3220 PRINT LEFT$(VT$,18);ER$
3230 FOR DELAY=1 TO 800:NEXT

```

```
3240 PRINT LEFT$(VT$,18);:FOR J=1 TO LEN
(ER$):PRINT " ";:NEXT:GOTO 3040
3250 LI(AN)=0:NM=NM+1:GOTO 3040
3990 RETURN
4000 :
4010 REM *** END ***
4020 :
4030 PRINT LEFT$(VT$,18);"THE GAME IS OV
ER!!!":PRINT
4040 IF WL THEN 4070
4050 PRINT "YOU GOT STUCK AFTER";NM;"MOV
ES.":PRINT "BETTER LUCK NEXT TIME!
4060 GOTO 4990
4070 PRINT "YOU DID IT!!!":PRINT "AND IT
ONLY TOOK YOU";NM;"MOVES."
4990 RETURN
```





In this game the object is to avoid the relentless pursuit of the horrible Twinky, and to escape from the danger-filled labyrinth. There are a plethora of obstacles which impede your escape. There are twenty squares which relocate you somewhere in the maze. There are twenty squares which cannot be entered. There is one square which contains an extremely sensitive exploding device. If you move onto this space, the ensuing blast will end your perilous journey . . . and your life. In the interest of fair play, you are given a zap gun which will temporarily ward off the Twinky's advance. But don't count on the gun too much or you'll never make it out alive.

```

10 REM *****
11 REM ***          ***
12 REM ***    TWINKY    ***
13 REM ***          ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000
30 GOSUB 2000
40 GOSUB 3000
90 END
1000 :
1010 REM *** INSTRUCTIONS ***
1020 :
1030 POKE 53280,6:POKE 53281,6
1040 VT$="{HOME}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}"
1050 GOSUB 1900
1060 PRINT "THIS IS THE GAME OF TWINKY.
IN IT YOU"
1070 PRINT "PRETEND TO BE A SPACE EXPLOR
ER WHO HAS"
1080 PRINT "LANDED ON A HOSTILE PLANET."
:PRINT
1090 PRINT "CAPTURED BY THE UNFRIENDLY N
ATIVES, YOU"
1100 PRINT "ARE TOSSED INTO A LARGE PRIS
ON ALONG"
1110 PRINT "WITH A FEROCIOUS TWINKY.":PR
INT
1120 PRINT "A TWINKY IS A HORRIBLE CREAT
URE THAT"
1130 PRINT "WILL CATCH YOU AND ABSORB YO
UR BODY INTO";
1140 PRINT "HIS IF HE GETS CLOSER THAN T
WO UNITS"
1150 PRINT "AWAY FROM YOU."
1160 GOSUB 1800:GOSUB 1900
1170 PRINT "IN THE INTEREST OF FAIR PLAY
, YOU ARE"
1180 PRINT "GIVEN A ZAP GUN THAT WILL TE
MPORARILY"
1190 PRINT "CHASE THE TWINKY AWAY." :PRI
NT
1200 PRINT "ALSO, IF YOU CAN MAKE IT TO
THE SPECIAL"

```



```

1210 PRINT "OBJECTIVE SQUARE BEFORE BEIN
G ABSORBED,"
1220 PRINT "YOU WILL BE SET FREE.":PRINT
1230 PRINT "AFTER YOU MOVE, YOU WILL BE
INFORMED OF"
1240 PRINT "YOUR DISTANCE FROM THE OBJEC
TIVE SQUARE"
1250 PRINT "AS WELL AS FROM THE TWINKY."
1260 GOSUB 1800:GOSUB 1900
1270 PRINT "THERE ARE SEVERAL OTHER OBJE
CTS WITHIN"
1280 PRINT "THE MAZE WHICH ARE OF INTERE
ST.":PRINT
1290 PRINT "THERE ARE TWENTY RELOCATION
SQUARES."
1300 PRINT "THESE SQUARES SEND YOU TO SO
ME OTHER"
1310 PRINT "SECTION OF THE MAZE.":PRINT
1320 PRINT "THERE ARE TWENTY IMPREGNABLE
SQUARES"
1330 PRINT "WHICH YOU CANNOT ENTER.":PRI
NT
1340 PRINT "THERE IS ONE SUPER DEADLY KI
LLER SQUARE"
1350 PRINT "WHICH ENDS YOUR ORDEAL QUICK
LY AND"
1360 PRINT "PAINLESSLY.":PRINT
1370 PRINT "THAT'S IT... TRY TO ENJOY IT
!"
1800 PRINT LEFT$(VT$,24);"PRESS {RVS ON}
RETURN{RVS OFF} WHEN READY TO CONTINUE:
{RVS ON} "
1810 GET AN$:IF AN$="" THEN 1810
1890 RETURN
1900 PRINT "{CLR}{WHT}{CSD}{CSD}";TAB(13
);"*** TWINKY ***":PRINT:PRINT
1990 RETURN
2000 :
2010 REM *** SETUP ***
2020 :
2030 GOSUB 1900
2040 DIM MA(15,15)
2050 DEF FNR(X)=INT(RND(1)*X)+1
2060 DEF FNA(X)=.001*INT(X*1000+.5)
2090 :
2100 REM *** BLOCKED
2110 FOR I=1 TO 20
2120 X=FNR(15):Y=FNR(15)
2130 IF MA(X,Y) THEN 2120
2140 MA(X,Y)=1

```

```

2150 NEXT
2190 :
2200 REM *** RELOCATION
2210 FOR I=1 TO 20
2220 X=FNR(15):Y=FNR(15)
2230 IF MA(X,Y) THEN 2220
2240 MA(X,Y)=2
2250 NEXT
2290 :
2300 REM *** SUPER KILL
2320 X=FNR(15):Y=FNR(15)
2330 IF MA(X,Y) THEN 2320
2340 MA(X,Y)=3
2390 :
2400 REM *** OBJECTIVE
2420 XO=FNR(15):YO=FNR(15)
2430 IF MA(XO,YO) THEN 2420
2440 MA(XO,YO)=4
2490 :
2500 REM *** TWINKY
2520 XT=FNR(15):YT=FNR(15)
2530 IF MA(XT,YT) THEN 2520
2540 MA(XT,YT)=5
2590 :
2600 REM *** PLAYER
2620 XP=FNR(15):YP=FNR(15)
2630 IF MA(XP,YP) THEN 2620
2640 MA(XP,YP)=6
2690 :
2900 SP=0:SS=0
2990 RETURN
3000 :
3010 REM *** PLAY ***
3020 :
3030 GOSUB 3600
3040 GOSUB 3900:IF WL THEN 3990
3050 PRINT:SM$="":INPUT "MOVE OR SHOOT (
M/S) ";SM$:SM$=LEFT$(SM$,1)
3060 IF SM$<>"M" AND SM$<>"S" THEN PRINT
"ENTER 'M' OR 'S'":GOTO 3050
3100 PRINT:PRINT "FORWARD, BACKWARD,"
3110 INPUT "RIGHT OR LEFT (F/B/R/L) ";AN
$:AN$=LEFT$(AN$,1)
3120 IF AN$="F" OR AN$="B" OR AN$="R" OR
AN$="L" THEN 3140
3130 PRINT "ENTER 'F', 'B', 'R' OR 'L'":
GOTO 3100
3140 IF AN$="F" THEN X=0:Y=-1
3150 IF AN$="B" THEN X=0:Y=1
3160 IF AN$="R" THEN X=1:Y=0

```

```

3170 IF AN$="L" THEN X=-1:Y=0
3180 IF SM$="S" THEN 3400
3200 X=X+XP:Y=Y+YP
3210 IF X>=1 AND X<=15 AND Y>=1 AND Y<=1
5 THEN 3220
3212 PRINT "MOVE NOT ALLOWED"
3214 PRINT "THAT WOULD TAKE YOU OUT OF T
HE MAZE":GOTO 3500
3220 IF MA(X,Y)<>1 THEN 3230
3222 PRINT "MOVE NOT ALLOWED"
3224 PRINT "THAT SPACE IS BLOCKED"
3226 GOTO 3500
3230 IF MA(X,Y)<>2 THEN 3240
3232 PRINT "YOU'VE BEEN RELOCATED....."
3234 X=FNR(15):Y=FNR(15)
3236 IF MA(X,Y)=1 THEN 3234
3238 GOTO 3230
3240 IF MA(X,Y)<>3 THEN 3250
3242 PRINT "MOVE ALLOWED, BUT..."
3244 PRINT "YOU FOUND THE SUPER KILL SQU
ARE!!!"
3246 PRINT "YOU'VE BEEN KILLED!!!":GOSUB
60300:WL=1:GOTO 3990
3250 IF MA(X,Y)<>4 THEN 3260
3252 PRINT "MOVE ALLOWED"
3254 PRINT "YOU FOUND THE OBJECTIVE!!!"
3256 PRINT "YOU WIN A TRIP OFF THIS PLAN
ET !!!":WL=0:GOTO 3990
3260 IF MA(X,Y)<>5 THEN 3270
3262 PRINT "MOVE ALLOWED":PRINT
3264 PRINT "<<<<< S C H L O O R P ! ! !
>>>>>":GOSUB 60300:WL=1:GOTO 3990
3270 PRINT "MOVE ALLOWED"
3280 MA(XP,YP)=SP:XP=X:YP=Y:SP=MA(XP,YP)
:MA(XP,YP)=6:GOTO 3500
3400 SX=XP:SY=YP
3410 SX=SX+X:SY=SY+Y:PRINT "ZAP--";
3420 IF SX>=1 AND SX<=15 AND SY>=1 AND S
Y<=15 THEN 3430
3422 PRINT "FIZZLE..."
3424 PRINT "THE SHOT LEFT THE MAZE."
3426 PRINT "THE SHOT MISSED THE TWINKY!"
:GOTO 3500
3430 IF MA(SX,SY)=0 OR MA(SX,SY)=2 OR MA
(SX,SY)=3 OR MA(SX,SY)=4 THEN 3410
3440 IF MA(SX,SY)<>1 THEN 3452
3442 PRINT "BLAST!"
3444 PRINT "THE SHOT HIT A WALL!!!"
3446 PRINT "THE SHOT MISSED":GOTO 3500
3452 PRINT "OUCH !!!"

```

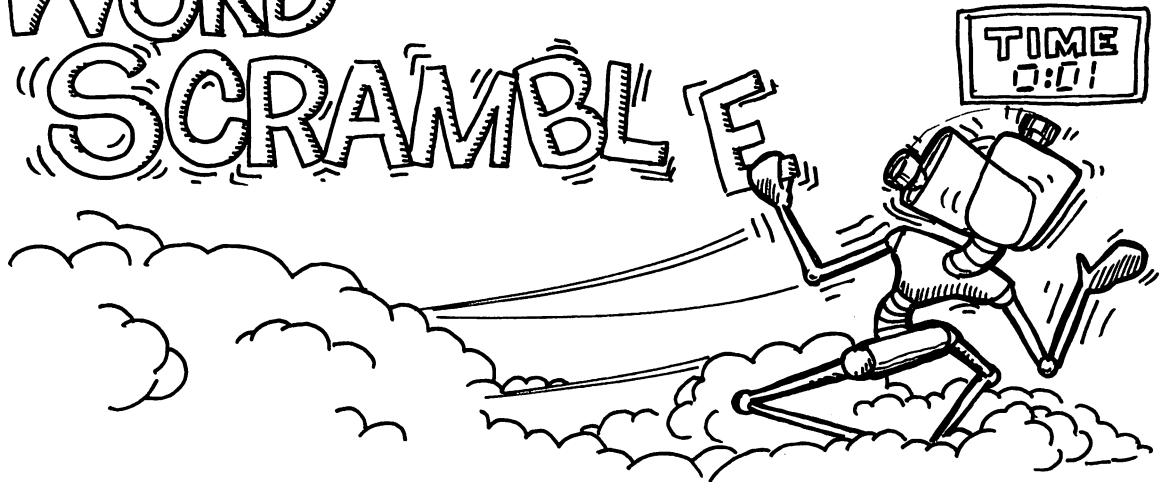
```

3454 PRINT "THE SHOT HIT THE TWINKY"
3456 PRINT "THE TWINKY RETREATS"
3460 MA(SX,SY)=SS:XT=FNR(15):YT=FNR(15):
SS=MA(XT,YT):MA(XT,YT)=5
3500 REM **** TWINKY MOVE LOGIC
3510 GOSUB 3600
3520 PRINT:PRINT "THE TWINKY MOVES ... "
3530 FOR I=1 TO 500:NEXT
3540 GOSUB 3900:IF WL THEN 3990
3550 IF XP<XT THEN X=-1:Y=0:GOTO 3590
3560 IF XP>XT THEN X=1:Y=0:GOTO 3590
3570 IF YP<YT THEN X=0:Y=-1:GOTO 3590
3580 IF YP>YT THEN X=0:Y=1
3590 MA(XT,YT)=SS:XT=XT+X:YT=YT+Y:SS=MA(
XT,YT):MA(XT,YT)=5:GOTO 3000
3600 :
3610 DT=FNA(SQR((XT-XP)^2+(YT-YP)^2))
3620 DO=FNA(SQR((XP-XD)^2+(YP-YD)^2))
3630 PRINT:PRINT "THE TWINKY IS";DT;"UNI
TS AWAY."
3640 PRINT:PRINT "THE OBJECTIVE IS";DO;"
UNITS AWAY."
3690 RETURN
3900 IF DT>=2 THEN 3990
3910 PRINT:PRINT "<<<<< S C H L O O R P
! ! ! >>>>>"
3920 PRINT "YOU'VE BEEN ABSORBED BY THE
TWINKY !!!"
3930 PRINT "YOU LOSE.":GOSUB 60300:WL=1
3990 RETURN
60300 :
60310 REM BELL SOUND ROUTINE
60320 :
60330 SD=54272:POKE SD+24,15
60340 POKE SD+5,7:POKE SD+6,0
60350 POKE SD+1,90:POKE SD+4,33
60360 FOR TS=1 TO 100:NEXT
60370 POKE SD+4,0:POKE SD+1,0
60390 RETURN

```

WORD

SCRAMBLE



For all of you word buffs, here is a game of anagrams geared to any skill level. For those of you who are not familiar with anagrams, they are words that are given in a scrambled fashion. The following list should help.

<u>SCRAMBLED</u>	<u>UNSCRAMBLED</u>	<u>RATING</u>
xob	box	1 (elementary)
tahb	bath	2 (easy)
laott	total	3 (light)
betd	debt	4 (mild)
gindru	during	5 (moderate)
ptles	slept	6 (trying)
spumlie	impulse	7 (tough)
yrtasc	crystal	8 (difficult)
meminscon	mnemonics	9 (hard)
preskulen	spelunker	10 (very hard)

In the actual game, the difficulty factor ranges from one to five. The program does not utilize graphics per se, but there are some interesting points. Line 3050 tells the computer to start printing inverse characters. Instead of printing white on black, the INVERSE is true, black being plotted on white. Lines 3050 - 3070 are responsible for drawing the white border. Try removing some of the inverse commands {RVS ON} and see what effects are produced in your program.

```

10 REM *****
11 REM *** ***
12 REM *** WORD SCRAMBLE ***
13 REM *** ***
14 REM *****
15 REM
16 REM
20 GOSUB 1000:REM INSTS
30 GOSUB 2000:REM SETUP
40 GOSUB 3000:REM PLAY!
50 GOSUB 4000:REM !END!
60 END
1000 :
1001 REM *** INSTS
1002 :
1010 POKE53280,0:POKE53281,0:PRINT "{CLR}
{WHT}";
1015 VT$="{HOME}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}{CSD}
}{CSD}{CSD}{CSD}":COL=646
1016 HT$="{CSR}{CSR}{CSR}{CSR}{CSR}{CSR}
}{CSR}{CSR}{CSR}{CSR}{CSR}{CSR}{CSR}
}{CSR}{CSR}{CSR}{CSR}{CSR}{CSR}{CSR}
}{CSR}{CSR}{CSR}{CSR}{CSR}{CSR}{CSR}"
1020 GOSUB 1800
1030 PRINTLEFT$(VT$,8);"IN THIS GAME, TH
E COMPUTER WILL CHOOSE AWORD AND SHO";
1031 PRINT"W YOU A SCRAMBLED VERSION":PR
INT
1035 PRINT"YOUR PROBLEM IS TO UNSCRAMBLE
THE WORD BEFORE THE ALLOTTED TIME E";
1036 PRINT"XPIRES."
1040 PRINT LEFT$(VT$,24);"PRESS ANY KEY
TO CONTINUE"
1041 GET ANS$:IF ANS$=""THEN 1041
1799 RETURN
1800 PRINT "{CLR}";LEFT$(VT$,4);LEFT$(HT
$,9);"*** WORD SCRAMBLE ***"
1801 RETURN
2000 :
2001 REM *** SET UP
2002 :
2010 GOSUB 1800
2020 PRINTLEFT$(VT$,8);"THE FOLLOWING AR
E AVAILABLE : ":PRINTLEFT$(VT$,12);

```



```

2025 PRINT"      1) VERY EASY"
2026 PRINT"      2) EASY"
2027 PRINT"      3) INTERMEDIATE"
2028 PRINT"      4) HARD      "
2029 PRINT"      5) VERY HARD  "
2030 PRINTLEFT$(VT$,19);:INPUT"WHICH OPT
ION (1-5) :";ANS
2035 IFANS<10RANS>50RANS<>INT(ANS) THENGO
SUB60300:GOTO2030
2100 DIM WO$(100):FORI=1TO100:READWO$(I)
:NEXT
2105 DATACAT,DOG, TREE, SIT, DOOR, BOX, ARM, W
ALL, TEA, PEN, PAD, CUP, PIN, DIG, GOOD, TIE
2106 DATASEA, ARE, HOW, LIP
2110 DATAWARM, WIND, LEAF, BLUE, WAIT, KITE, S
LIP, DRIP, MAZE, PARK, LIFE, GAME, HIGH, DISK
2111 DATARUIN, CARD, MOLE, ARCH, HARD, VERY, D
ATA
2115 DATAPENCIL, LOOSE, NORMAL, TIRED, BEFOR
E, AFTER, BLACK, TARGET, KNOCK, BAGEL, INPUT
2116 DATARETURN, START, ENTRY, GROUND, SHINE
, HORSE, PAPER, GREEN, PHONE
2120 DATAORIGINAL, BEHIND, MAGAZINE, STORAG
E, SCRATCH, COMPUTER, PERSONAL, SOFTWARE
2121 DATAPERFORM, SYSTEM, WINDOW, COMBINE, T
ANGENT, SPECIFY, ANOTHER, EVALUATE, MEMORY
2122 DATAINSIDE, IGNORE, HOWEVER
2125 DATASEQUOIA, MATRIX, COORDINATE, SPACI
AL, DIRECTION, SUBSTANTIAL, CONTINUE
2126 DATASUBSCRIPT, EMULATE, APPROPRIATE, C
ONICAL, DEVELOPMENT, ELEVATION, MECHANICAL
2127 DATAMAGNETIC, TRAJECTORY, STIMULUS, CI
RCUMSTANCE, PROBABILITY, PROJECTION
2200 DIM WR$(15), WS$(15), WC$(15)
2990 RETURN
3000 :
3001 REM *** PLAY
3002 :
3010 WO=INT(RND(1)*20)+(AN-1)*20+1
3020 WL=LEN(WO$(WO)):FORI=1TOWL:WR$(I)=M
ID$(WO$(WO), I, 1):WS$(I)=WR$(I):NEXT
3025 FORI=1TOWL:TX=INT(RND(1)*WL)+1:WS$=
WS$(I):WS$(I)=WS$(TX):WS$(TX)=WS$:NEXT
3030 PRINT"<CLR>"LEFT$(VT$,4);TAB(9);"**
* WORD SCRAMBLE ***"
3040 PRINTLEFT$(VT$,8);"HERE IS YOUR SCA
MBLED WORD : "
3050 PRINTLEFT$(VT$,11);TAB(10);:FORI=1T
OWL+4:PRINT"<RVS ON> ";:NEXT
3060 PRINTLEFT$(VT$,12);TAB(10);:PRINT"<

```

```

RVS ON) ";TAB(10+WL+3) " ";LEFT$(VT$,13);
TAB(10);
3061 PRINT"{RVS ON} ";TAB(10+WL+3) " ";LE
FT$(VT$,14);TAB(10);" ";
3062 PRINTTAB(10+WL+3){RVS ON} ";
3070 PRINTLEFT$(VT$,15);TAB(10);:FORI=1T
OWL+4:PRINT"{RVS ON} {RVS OFF}";:NEXT
3080 PRINTLEFT$(VT$,13);TAB(12);:FORI=1T
OWL:PRINTWS$(I);:NEXT
3090 PRINTLEFT$(VT$,18);TAB(12);:FORI=1T
OWL:PRINT"-";:NEXT
3100 PRINTLEFT$(VT$,21)TAB(1){RVS OFF}{
ENTER A-Z FOR THE LETTER...RETURN WHENYO
U AR";
3101 PRINT"E DONE.)"
3200 WP=1
3205 FORI=250*ANT01STEP-1
3206 IFPEEK(197)<>64THEN3210
3207 IFINT(I/50)*50=ITHENPRINTLEFT$(VT$,
13)TAB(28)"TIME :";INT(I/50);"{DEL} "
3208 NEXT:RETURN
3210 PRINTLEFT$(VT$,18);TAB(11+WP);:GETA
NS$:IFANS$=""THEN3207
3220 IFASC(ANS$)=20THEN3300
3230 IFASC(ANS$)=21THEN3400:REM ?ASC(21)
3235 IFASC(ANS$)=13THENRETURN
3240 IFANS$<"A"ORANS$>"Z"THENANS$="-"
3245 WC$(WP)=ANS$:PRINTANS$
3250 WP=WP+1:IFWP>WLTHENWP=1
3260 GOTO3207
3300 IFWC$(WP)=""THENPRINT"-";:GOTO3320
3310 PRINTWC$(WP);
3320 WP=WP-1:IFWP<1THENWP=WL
3330 GOTO3207
3340 IFWC$(WP)=""THENPRINT"-";:GOTO3420
3410 PRINTWC$(WP);
3420 WP=WP+1:IFWP>WLTHENWP=1
3430 GOTO3207
4000 :
4001 REM *** END
4002 :
4010 FORI=1TOWL:PRINTLEFT$(VT$,13);TAB(1
1+I);WR$(I);
4011 IFWC$(I)=WR$(I)THENPRINTLEFT$(VT$,1
8);TAB(11+I);"{RVS ON}WC$(I);"{RVS OFF}
":WC=WC+1
4020 NEXT:PRINTLEFT$(VT$,21);TAB(1);:IFW
C=WLTHENPRINT"CONGRATULATIONS!";:ZT=1
4021 IFZT=1THENZT=0:PRINT" YOU UNSCRAMBL
ED THE ENTIRE WORD! ":GOTO4040

```



```
4030 PRINT"{CSD}{CSD}THE GAME IS OVER...  
OUT OF"WL"LETTERS, YOU GOT"WC"CORRECT."  
"  
4040 INPUT"DO YOU WISH TO PLAY AGAIN";AN  
S$: IFLEFT$(AN$,1)="Y"THENRUN  
4050 RETURN  
60300 :  
60310 REM BELL SOUND ROUTINE  
60320 :  
60330 SD=54272:FORLS=0TO24:POKESD+LS,0:N  
EXT  
60340 POKESD+1,90:POKESD+5,9:POKESD+15,3  
0:POKESD+24,15  
60350 FORLS=1TO1:POKESD+4,21  
60360 FORTS=1TO25:NEXT:POKESD+4,20  
60370 FORTS=1TO25:NEXT:NEXT  
60375 POKESD+24,0  
60380 RETURN
```



MUBBLE Chase



This is an addicting game in which you try to consume all of the flower points before the Mubble Eaters catch up and consume you. The action takes place on a 9 by 7 grid. There are 48 flower points and three Mubble Eaters (ME's). If you do not complete the mission on the first run, don't worry. This game gives you three "men" per game. If you are unable to eat all the flower points with your first man, you can finish the maze with your second two. It is up to you to safely guide the Mubble to gluttony. This is one of two programs in the book that will be explained line by line. It was chosen because its use of graphics is similar to that used by many of the other games in the collection. If you can understand how this program works, you should be able to figure out how most of the others work as well.

10-14 REM means REMark. Anything may be construed as a remark. In this case, the REM statement is used to allow the program name to be written. Naturally, ***** is not a REMark about the program itself, but the REM statement allows the programmer to construct a title display in a format which is acceptable to the computer.

15-16 Colons may be used to space apart sections of the program just like REM statements. The computer will go to the next line if it encounters a colon on a line by itself.

20 GOSUB 1000 tells the computer to branch to line 1000 and to continue program execution from that point on until a RETURN statement is encountered. When a RETURN is encountered, the program will return to line 20, see if there are any more statements on that line and execute them (in our case there is only a REM), then drop down to line 30. The REM statement is strictly for your benefit — it tells you that the subroutine beginning at line 1000 contains the instructions for the game.

30-50 Are essentially the same as line 20. This small section of the program shows that a good BASIC program has a clearly defined structure. All instructions are included in lines 1000-1999, program setup in lines 2000-2999, game play in lines 3000-3999, and the end routine (i.e., displaying the final score, etc.) is located from 4000 to the end of the program.

60 The END statement ends the entire program after the end routine has been executed.

70-1000 More colons for spacing.

1001 REM *** INSTRUCTIONS This line tells the programmer that the instructions section of the program starts with this line.

1002 Colon for spacing.

1010 POKE 53281,12 sets the background screen color to light grey, and POKE 53280,12 does the same for the screen border. The PRINT "{CLR}" clears the screen and HOMEs the cursor to the upper left corner of the screen. This will show up as an inverse heart-shaped character on your screen.

1020 PRINT TAB(10) uses the TAB command to center the title. The first character to be PRINTed is a (CTRL-1) which sets the letter color to black.

1025 The two PRINT statements add some space between the title and the text to follow.

1030-1050 These lines print what is between the quotation marks on the screen. Note that lines 1030, 1031, 1040, and 1050 all end with a semicolon. A semicolon suppresses the carriage return at the end of the line, and causes the text in the next PRINT statement to continue on the same screen line. 1035, a PRINT all by itself, causes a blank line to be PRINTed on the screen.

1055 Waits for you to press the keyboard to indicate that you have read all the instructions that on the screen. Unlike GET statements in other BASICs, GET statements in Commodore BASIC will go right on to the next line if a key is not being pressed when one is encountered. That is why you'll see a structure like GET X\$: IF X\$ < > (condition) THEN 1055 in a lot of Commodore listings. The IF X\$ < > CHR\$(13) checks to see if the ASCII code for the key being pressed is 13 (a carriage return). Until the RETURN key is pressed, the rest of the line will continue to be executed (GOTO 1055) and the program will go nowhere.

1060 Once a RETURN is entered in 1055, program control falls through to this line. It clears the screen for more text.

1065 PRINTs the title again.

1066 PRINT statements used for blank lines on the screen.

1070-1080 PRINT the text inside quotation marks to the screen.

1085-1095 Construct the diagram of keys that are used in the game to control the movement of the Mubble. Change the values in some of the TAB statements if you can't see how the TABs are being employed.

1100-1105 Waits for INPUT as in line 1055.

1990 The RETURN is placed near the end of the 1000-1999 section. It is best to place RETURNS at the very end of a subroutine so that if any other changes need to be made, they can be entered easily. It also helps keep the structure of the program very clear. Control will return to line 20, where the REM will be encountered. Control will then go to line 30, and the program will continue execution at line 30.

2000 : This line and line 2002 do nothing but add to the readability of line 2001.

2001 REM *** SETUP Tells the programmer that the setup section of the program starts with this line.

2003 SID=54272 Gives the memory location of the SID chip, which is used to generate sound on your Commodore 64. By setting a variable equal to the beginning location for the chip, we can use FOR/NEXT loops to stuff values into it rather than POKEing one value in at a time. It's easier to use a structure like the following than to POKE a value into each of 24 locations one at a time:

```
FOR I = 0 TO 24 : POKE SID+I,C : NEXT I
```

2015 Two arrays are DIMensioned in this line. Let's take a quick look at how multi-dimensional arrays are set up. A DIMension of ME(3,2) will set up a total of six memory locations. The numbers in the parentheses are called subscripts — so ME is said to be a double-subscript variable. This array contains the elements ME(1,1), ME(1,2), ME(2,1), ME(2,2), ME(3,1), and ME(3,2). These elements are arranged in memory something like this:

	<u>COLUMN 1</u>	<u>COLUMN 2</u>
ROW 1	ME(1,1)	ME(1,2)
ROW 2	ME(2,1)	ME(2,2)
ROW 3	ME(3,1)	ME(3,2)



Each of these is actually a different variable — all related, but individual nonetheless. In our example, the ROW number (or first subscript) corresponds to a specific Mubble Eater, while the COLUMN number gives further detail about that Mubble Eater. In our case, for example, ME(1,1) will contain Mubble Eater #1's X coordinate, and ME(1,2) will contain his Y coordinate.

An array actually has an automatic 0 subscript as well, thus creating subscripts like ME(0,1), but these are usually not used. It is permissible to use them, but not recommended. Also, the computer does not actually store arrays in two dimensions, but as a long string. The computer can access part of that string very quickly. The illustration above is just an easy way to visualize the way in which arrays are stored in memory.

The SP(3) DIMension will create a single-DIMensional array containing the elements SP(1), SP(2), and SP(3). Each of these is a unique variable as well.

2020-2026 These lines contain the graphics routine explained at the beginning of the book. PLT is set to the beginning memory location of the block of memory that this routine will be stored in. Line 2020 works like this: the FOR I = 0 to 24 will cause the FOR/NEXT loop to be executed a total of 25 times (0 counts!). Each time the loop is executed, READ C will READ the first DATA element encountered (in this case the 32 in line 2021). The POKE PLT+I,C will put the value C just read into the memory location calculated by the equation PLT + I. The first time through the loop PLT + I will equal 49152, the second time 49153, and so on. This structure lets you read large blocks of data into sequential memory locations.

2030-2032 DIM CO\$(15) DIMensions an array of 15 elements. We will use each of these variables to represent a single color. The FOR/NEXT loop in the second part of the line reads each of the colors (which are entered as string DATA elements in lines 2031-2032) into the array.

2035 This line initializes (sets the initial value of) some variables. PT is set to 0, MU is set to 3, and NM is set to 4. The POKEs set the background and border colors to black.

2041-2049 Line 2041 clears the screen and draws a horizontal white line across the top of the screen. The FOR I = 1 TO 19 STEP 3 will create the same pattern a total of 7 times — a group of 3 lines consisting of a solid black line, followed by two lines that create the blue boxes that make up the playing field. 2048 PRINTs the final horizontal black line, and 2049 PRINTs the horizontal white line across the bottom of the screen.

2050 Draws the number of Mubbles remaining at the bottom of the screen.

2060 Creates a two DIMensional array for the variables MU(1,1) through MU(8,2), as explained above.

2070-2074 These lines probably look scarier to you than they really should. Line 2070 contains a FOR statement, but so does 2071, and we don't get a NEXT until line 2074, and then there are two. What gives? Actually, this programming structure is quite common. The whole construct is known as a nested loop, in that one loop is completely enclosed within another. The outer loop, which is the I-loop, is enclosed by lines 2070 and the second NEXT in line 2074. The purpose of these lines is actually to draw the flower points on the maze, so it may be easier to explain if we attack the problem from that point. The first time through the loop, I = 5. In line 2071, J will be set to 4. So with I=5 and J=4, control drops to line 2072. The first part of the line selects at random a 1 character long portion of C\$ (sets a color). SYSPLT,I,J uses the graphics routine to plot the point 5,4. Next, the line makes sure that we are not in reverse mode, PRINTs C\$ (sets the color), then draws an asterisk ("*"). Control then goes back to line 2071 where J is incremented by the STEP value, which in our case is 3. When this inner loop has been executed enough times so that J=19, the loop will end and the second NEXT in line 2074 will send control back to line 2070, where I will be incremented by 4 and the entire process will be repeated until all the flower points are drawn. While this may seem complex, to create this display using PRINT statements would actually require a greater number of lines and more space in memory.

2080 & 2085 These two lines set up FuNctions which can be called later in the program. Rather than typing out these entire lines every time such information is required, you can type in something like PRINT FN5C(12), and the FuNction defined in line 2080 will perform that calculation with a value of 12 inserted for V. SC and V are variables that may be designated by the programmer, but the DEF FN section must always contain only those reserved words.

2990 Returns control back to line 30 as line 1990 did to 20.

3000 & 3002 Colons used for spacing.

3001 REM statement to indicate beginning of PLAY segment.

3005-3006 Enter the original X,Y coordinates for the Mubble Eaters into the array ME. At the beginning, a Mubble Eater is placed in each corner. ME(1,1) and ME(1,2) define Mubble #1's X and Y coordinates.

3007 Plots the Mubble Eaters using SYSPLT.

3008-3009 3008 figures out the location of the Mubble. The variable NM is the length of the Mubble. 3009 sets the variable MX to 4 (NM) and MY to 22.

3010 Draws the Mubble, using the values in the MU array and a FOR/NEXT loop. The variables MD, HI, and MM are all initialized.

3011 First, MU is decremented by one. Then a test is made (IF MU < 0) and if the condition is found to be true, control will RETURN to line 40 (the beginning of the subroutine).

3015 PRINTs the READY message across the bottom of the screen, and holds it there for a delay of 2000.

3016 Erases the message so the screen won't look cluttered.

3020 Sends program control to the subroutine at 3300, which, as explained by the REM, moves the Mubble.

3025 Checks to see if PT = 48 (if all 48 flower points have been eaten). This would mean that the screen has been cleared, and control would be sent back to the top of the program.

3030 Sends program control to the subroutine at 3400, which, as explained by the REM, moves the Mubble Eaters.

3040 HI is a "flag." Normally HI will equal 0, indicating that no collision has taken place. This will send control back to 3020, and both Mubble and Mubble Eaters will be moved again.

3041-3044 This group of lines makes a raspberry sound if one of the Mubble Eaters catches up to the Mubble.

3045-3051 Erase the old Mubble Eaters and Mubble from the screen.

3005 Sends control back to 3005, giving a fresh playing board (except for flower points already eaten), and starting the next round of play.

3300 Don't fret. Although this line appears to be a confusing conglomeration of variables, there is a definite purpose for this line. Before starting with an explanation, there are two important facts that we must know. One, MX and MY are the X and Y coordinates of the Mubble. Two, the flower points are located at specific intervals. Armed with this knowledge, you have a good chance of understanding what follows. The 48 flower points are each located at an intersection. The X,Y coordinates at these 48 points are:

5,4	9,4	13,4	17,4	21,4	25,4	29,4	33,4
5,7	9,7	13,7	17,7	21,7	25,7	29,7	33,7
5,10	9,10	13,10	17,10	21,10	25,10	29,10	33,10
5,13	9,13	13,13	17,13	21,13	25,13	29,13	33,13
5,16	9,16	13,16	17,16	21,16	25,16	29,16	33,16
5,19	9,19	13,19	17,19	21,19	25,19	29,19	33,19

All 48 of these points have one important thing in common. If you add three to any of the eight different X coordinates, the sum will be an exact multiple of four. Therefore, $MX+3$ divided by 4 will be an integer, and $(MX+3)/4$ will be equal to $INT((MX+3)/4)$. Also, if you add two to any of the six Y coordinates, then the sum will be evenly divisible by three, and $(MY+2)/3$ will equal $INT((MY+2)/3)$. The only times when both $(MX+3)/4 = INT((MX+3)/4)$ and $(MY+2)/3 = INT((MY+2)/3)$ are at one of the forty-eight intersections. If, indeed, the Mubble is at an intersection, then FL equals "yes." If the Mubble is not at an intersection, then FL equals "no."

3301 This line is a test. If FL is equal to "yes" then the computer will skip to line 3320.

3304 Reads the keyboard for the desired direction. If no key is being pressed when the GET comes up, the direction stays the same, and control goes on to line 3320 without changing the Mubble's direction.

3310-3313 These four lines check to see if the last key pressed was an A, K, Z, or J. Note that only if one of these keys was pressed will the variable MD be changed.

3320 Sets the variable X2 equal to the Mubble's X coordinate and Y2 to the Mubble's Y coordinate.

3321-3324 Each of these lines checks the value of MD (Mubble's Direction). In 3321, IF MD equals 1, then X2 is increased by 1 (the Mubble moves

towards the right of the screen). The same tests are made for the other three directions in lines 3322-3324.

3330 This line checks to see if either the X or Y coordinate of the Mubble is out of the range of the maze. If it is, the subroutine will RETURN and wait for an acceptable direction to be entered.

3340 This line makes use of the function FNESC described earlier. If the character at this intersection is not an asterisk ("*"), then the program skips to line 3345.

3341 If the character at that intersection was indeed an asterisk, the point total is incremented by 1. If PT=48, then all the flower points have been collected and control RETURNS to the beginning of the entire loop.

3345-3346 These lines are responsible for plotting the Mubble as it now appears.

3350 These lines define the values X2 and Y2, and then equate MX and MY, the X and Y coordinates for the Mubble, to these two variables.

3355 Ends the Mubble Move subroutine.

3400 Begins the subroutine that controls the movement of the Mubble Eaters. This is the beginning of a loop which will be run through three times.

3401 Defines which of the three Mubble Eaters are to be moved. The first time through the loop it will be Mubble #1, the second time #2, etc.

3402 Uses the same test as in line 3300 to determine if the present Mubble Eater is at an intersection.

3405 If the Mubble Eater is not at an intersection, then the program "drops through" to line 3430.

3407 This line is going to generate a random number in order to determine how the Mubble Eater will make his move. The first part of the line will create the random number, and will use an ON/GOTO structure to decide the appropriate subroutine. If the number is equal to 1, 2, 3, 4, or 5, control drops to line 3410. If the number is equal to 6 or 7, control goes to 3420, and if it equals 8 then control goes to 3430.

3410 Checks to see if the random number generated in line 3407 is greater than .5. If it is, then control drops through to line 3413.

3411-3414 These lines function in much the same way as lines 3310-3313. Various comparisons are made between the X and Y coordinates of the Mubble and the X and Y coordinates of the Mubble Eater. Depending on the relationship between these two coordinates, a direction is assigned to the Mubble Eater.



3420 If the random number generated in line 3407 was either 6 or 7, then a random direction is assigned to the Mubble Eater. The $\text{INT}(4*\text{RND}(0))+1$ will generate a random number between 1 and 4 for the direction.

3430 If the random number generated in line 3407 was equal to 8, control dropped straight through to this line of the program. In this case, the direction for the Mubble will remain the same.

3435-3438 The X and Y coordinates will be adjusted in these four lines, according to the value TD has been assigned. These lines work much the same way as lines 3321-3324, so look at their description above if you don't see what is happening here.

3440 Does more range checking to make sure that the Mubble Eater is within the range of the maze. If it is not, control is sent to line 3490.

3445 Erases the old Mubble position.

3450 Sets the variable $\text{SP}\$(I)$ equal to a random flower point color using the FNCO function defined earlier in the program. If this line is removed, the Mubble Eaters will not replace the flower point that they have crossed over — rather, they will just erase it.

3455 Plots the new Mubble Eater.

3490-3494 These two loops (the I and J loop) check to see if any of the three Mubble Eaters has caught up with the Mubble. Notice that in line 3492, the variable HI is set to 1 if there has been a collision. Remember that in line 3040, that if HI is set to 1 then the program assumes a collision has taken place, and will branch off accordingly.

4000 & 4002 Colons used for spacing.

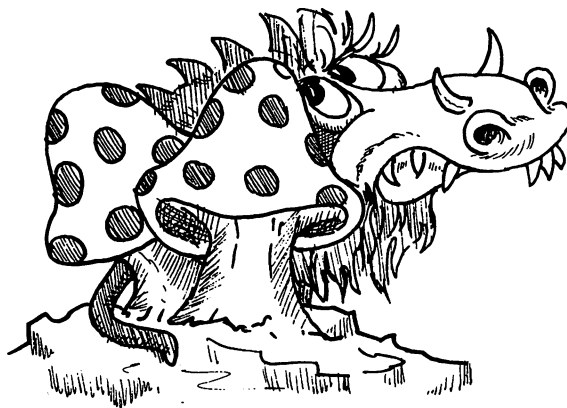
4001 REM *** END is used to indicate that this is the first line of the END subroutine.

4015 Checks the value of PT at the end of the game. If $PT = 48$, then all the flower points were gathered, and the message in that line will be PRINTed. If $PT < > 48$ then control drops through to line 4020.

4020 PRINTs the total flower points gathered if that total is less than 48. That total is incorporated into the middle of the sentence, simply by PRINTing the variable within a PRINT statement.

4990 This line RETURNs control of the program to line 50 (which sent it to line 4000 to begin with). Finally, control drops through to line 60 and the whole program ENDS.

To produce the diamond figure (as in line 2050), press SHIFT-Z.
To produce the circle figure (as in line 3007), press SHIFT-W.



```

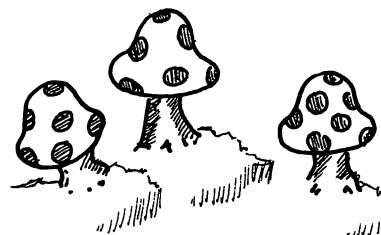
10 REM *****
11 REM ***          ***
12 REM **  MUBBLE CHASE  **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1010 POKE53281,12:POKE53280,12:PRINT"{CLR}"
1020 PRINTTAB(10)"{BLK}*** MUBBLE CHASE
***"
1025 PRINT:PRINT
1030 PRINT"IN THIS EXCITING GAME, YOU CO
NTROL THE MOVEMENT OF THE HUNGRY ";
1031 PRINT"LITTLE CREATURE WE CALL THE
MUBBLE. THE MUBBLE SCURRIESTHROUGH ";
1032 PRINT"A MAZE, TRYING TO EAT UP ALL
OF THE FLOWER POINTS."
1035 PRINT
1040 PRINT"UNFORTUNATELY, THERE ARE THRE
E MUBBLE- EATERS IN THE SAME MAZE, ";
1041 PRINT"WHO WANT TO CATCH AND EAT
THE POOR MUBBLE."
1050 PRINT:PRINT:PRINT"PRESS *RETURN* WH
EN READY TO CONTINUE";
1055 GETX$:IFX$(<>CHR$(13))THEN1055
1060 PRINT"{CLR}"
1065 PRINTTAB(10)"{BLK}*** MUBBLE CHASE
***"
1066 PRINT:PRINT
1070 PRINT"YOU MUST MANEUVER THE MUBBLE
TO THE"
1071 PRINT"FLOWER POINTS AND AWAY FROM T
HE MUBBLE"
1072 PRINT"EATERS. YOU ARE ALLOWED TO L
OSE TWO"
1073 PRINT"MUBBLES, BUT WHEN THE THIRD M
UBBLE IS"
1074 PRINT"EATEN, THE GAME IS OVER."

```

```

1075 PRINT
1076 PRINT"ALSO THE MUBBLE GROWS AS HE C
ONSUMES THE FLOWERS, MAKING IT ";
1077 PRINT"HARDER TO AVOID THE MUBBLE E
ATERS!"
1080 PRINT
1085 PRINT"MOVEMENT OF THE MUBBLE IS CON
TROLLED BY USING THE LETTERS A,Z,J,K."
1090 PRINT
1091 PRINTTAB(16)"UP"
1092 PRINTTAB(16)"A"
1093 PRINTTAB(7)"LEFT - J K - RIGHT"
1094 PRINTTAB(16)"Z"
1095 PRINTTAB(15)"DOWN"
1100 PRINT:PRINT"PRESS *RETURN* WHEN REA
DY TO CONTINUE";
1105 GETX$:IFX$<>CHR$(13)THEN1105
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2003 SID=54272
2015 DIMME(3,2),SP(3)
2020 PLT=49152:FORI=0TO24:READC:POKEPLT+
I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 DIMCO$(15):FORI=0TO15:READCO$(I):NE
XT
2031 DATA"{BLK}","{WHT}","{RED}","{CYN}
","{PUR}","{GRN}","{BLU}","{YEL}"
2032 DATA"{ORNG}","{BRWN}","{LT RED}","{
GRY 1}","{GRY 2}","{LT GRN}","{LT BLU}","
"{GRY 3}"
2035 PT=0:MU=3:NM=4:POKE53281,0:POKE5328
0,0
2041 PRINT"{CLR}{RVS ON}{WHT}
"
2043 FORI=1TO19STEP3
2044 SYSPLT,0,I:PRINT"{RVS ON}{WHT} {RVS
OFF}
{RVS ON} "
2045 PRINT"{RVS ON}{WHT} {CSR}{LT BLU}
{CSR} {CSR} {CSR} {CSR} {CSR}
{CSR} {CSR} {CSR} {CSR}{WHT} "
2046 PRINT"{RVS ON}{WHT} {CSR}{LT BLU}

```



```

{CSR} {CSR} {CSR} {CSR} {CSR}
{CSR} {CSR} {CSR} {CSR} {WHT} "
2047 NEXT
2048 PRINT "{RVS ON} {WHT} {RVS OFF}
           {RVS ON}
"
2049 PRINT "{RVS ON} {WHT}
           "
2050 PRINT "{GRN} * * *";
2060 DIMMU(8,2)
2070 FORI=5TO33STEP4
2071 FORJ=4TO19STEP3
2072 C%=MID$(" {LT RED} {LT GRN} {YEL} {PUR}
{GRN}",1+5*RND(0),1):SYSPLT,I,J:PRINT "{R
VS OFF}"C%*";
2074 NEXT:NEXT
2080 DEFFNSC(V)=PEEK(1024+X2+40*Y2+V)
2085 DEFFNCO(V)=PEEK(55296+X2+40*Y2+V)AN
D15
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3005 ME(1,1)=1:ME(1,2)=1:ME(2,1)=37:ME(2
,2)=1
3006 ME(3,1)=37:ME(3,2)=22:DE(1)=1:DE(2)
=4:DE(3)=3
3007 FORI=1TO3:SYSPLT,ME(I,1),ME(I,2):PR
INT "{WHT} ";:SP$(I)=" ":NEXT
3008 FORI=1TONM:MU(I,1)=I:MU(I,2)=22:NEX
T
3009 MX=NM:MY=22
3010 FORI=1TONM:SYSPLT,MU(I,1),MU(I,2):P
RINT "{GRN} *";:NEXT:MD=1:HI=0:MM=0
3011 MU=MU-1:IFMU<0THENRETURN
3015 SYSPLT,15,24:PRINT "{WHT} <<< READY >
>";:FORI=1TO2000:NEXT
3016 SYSPLT,15,24:PRINT " ";
3020 GOSUB3300:REM MOVE MUBBLE
3025 IFPT=48THENRETURN
3030 GOSUB3400:REM MOVE MUBBLE EATERS
3040 IF HI=0 THEN 3020
3041 POKESID+24,15:POKESID+5,9:POKESID+6
,0
3042 FORI=15TO0STEP-1:POKESID+1,10+40*RN
D(0):POKESID+24,I
3043 POKESID+4,33:FORPA=1TO10:NEXT
3044 POKESID+4,32:NEXT:POKESID,0:POKESID
+1,0
3045 FORI=1TO3:SYSPLT,ME(I,1),ME(I,2):PR

```

```

INTSP$(I);:NEXT
3050 FORI=1TONM:SYSPLT,MU(I,1),MU(I,2):P
RINT" ";:NEXT
3051 SYSPLT,2,24:PRINTCHR$(20)CHR$(20);
3055 GOTO3005
3300 FL=((MX+3)/4)=INT((MX+3)/4)AND((MY+
2)/3)=INT((MY+2)/3)
3301 IFNOTFLTHEN3320
3304 GETK$:IFK$=""THEN3320
3310 IFK$="A"THENMD=2
3311 IFK$="K"THENMD=1
3312 IFK$="Z"THENMD=4
3313 IFK$="J"THENMD=3
3320 X2=MX:Y2=MY
3321 IFMD=1THENX2=X2+1
3322 IFMD=2THENY2=Y2-1
3323 IFMD=3THENX2=X2-1
3324 IFMD=4THENY2=Y2+1
3330 IFX2<1ORX2>37ORY2<1ORY2>22THENRETUR
N
3340 IFFNSC(0)<>42THEN3345
3341 PT=PT+1:IFPT=48THENRETURN
3342 T=4+INT(PT/12):IFT<>NMTHENNM=T:MU(N
M,1)=X2:MU(NM,2)=MY
3345 SYSPLT,MU(1,1),MU(1,2):PRINT" ";:SY
SPLT,X2,Y2:PRINT"{GRN}*";
3346 FORI=1TONM-1:MU(I,1)=MU(I+1,1):MU(I
,2)=MU(I+1,2):NEXT
3350 MU(NM,1)=X2:MU(NM,2)=Y2:MX=X2:MY=Y2
3355 RETURN
3400 FORK=1TO2:MM=MM+1:IFMM>3THENMM=1
3401 I=MM:TX=ME(I,1):TY=ME(I,2)
3402 FL=((TX+3)/4)=INT((TX+3)/4)AND((TY+
2)/3)=INT((TY+2)/3)
3405 IFNOTFLTHEN3430
3407 ON1+8*RND(0)GOTO3410,3410,3410,3410
,3410,3420,3420,3430
3410 IFRND(0)>.5THEN3413
3411 IFMX<TXTHENDE(I)=3:GOTO3430
3412 IFMX>TXTHENDE(I)=1:GOTO3430
3413 IFMY<TYTHENDE(I)=2:GOTO3430
3414 IFMY>TYTHENDE(I)=4:GOTO3430
3420 DE(I)=INT(4*RND(0))+1
3430 X2=TX:Y2=TY:TD=DE(I)
3435 IFTD=1THENX2=X2+1
3436 IFTD=2THENY2=Y2-1
3437 IFTD=3THENX2=X2-1
3438 IFTD=4THENY2=Y2+1
3440 IFX2<1ORX2>37ORY2<1ORY2>22THEN3490
3445 SYSPLT,TX,TY:PRINTSP$(I);

```




```

3450 SP$(I)=CO$(FNCO(O)):T=FNSC(O):IFT=8
7THENT=151
3451 IFT=90THENT=122
3452 SP$(I)=SP$(I)+CHR$(T)
3455 SYSPLT,X2,Y2:PRINT"{WHT}o";:ME(I,1)
=X2:ME(I,2)=Y2
3490 FORI=1TO3:TX=ME(I,1):TY=ME(I,2)
3491 FORJ=1TONM
3492 IFTX=MU(J,1)ANDTY=MU(J,2)THENHI=1:G
OTO3495
3493 NEXT
3494 NEXT
3495 NEXT:RETURN
4000 :
4001 :REM *** END
4002 :
4010 PRINT"{CLR}":PRINT"THE GAME IS OVER
!!!"
4015 IFPT=48THENPRINT"YOU GOT ALL THE PO
INTS (YOU WIN!!)":RETURN
4020 PRINT"YOU SCORED"PT"POINTS.....GOOD
EFFORT"
4990 RETURN

```



Air Attack



This game requires good timing. You are the pilot of a B19 Bomber trying to sink enemy ships. The graphics in this game are simple but effective. To illustrate this point, run the program. The plane is black, the ship is grey, and the water is dark blue. In order to see how these graphics are created, hit the RUN-STOP key. Type LIST 2060-2065. These two lines use the machine language graphics routine set up earlier in the program (loaded with DATA statements in line 2020) to draw the blue water. Try deleting line 2060 to see what is happening. Your ship will appear to "float." The plane is drawn in line 3120 and the ship is drawn in line 3300. Try using different Commodore characters to change the shape of the plane into a U.F.O. bomber or a rocket ship. The various sections of the program are identified by REMarks. If you would like to experiment with any of these routines, first save the correct version to disk, then change the version that is still in your computer's memory. If you wish to save a modified version of the program, make sure it has a different name than the original, e.g., AIR ATTACK2 instead of AIR ATTACK.

To produce the plane (as in line 3120) press these keys:

CTRL-0, CMDR-7, 2 spaces, CTRL-2, 2 CMDR-O's, 1 space, CTRL-1, CTRL-9, SHIFT-English pound sign, CTRL-0, 1 space, 1 {CSD}, 7 {CSL}, CTRL-9, SHIFT-English pound sign, 1 space, minus sign, 2 spaces, minus sign, 1 space.

To produce the ship (as in line 3300) press these keys:

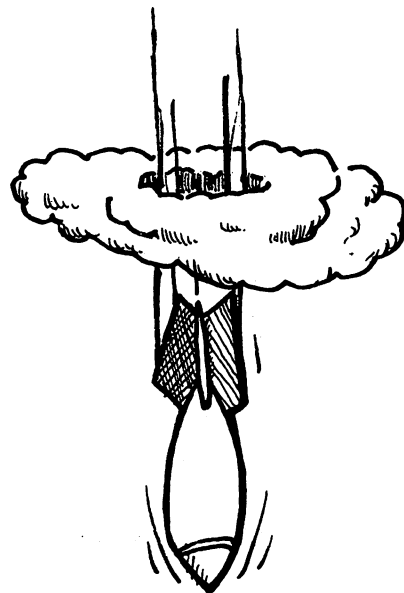
CMDR-8, 3 spaces, CMDR-A, CTRL-R, 1 space, CTRL-0, CMDR-S, 1 {CSD} 6 {CSL}, CMDR-asterisk, CTRL-9, 5 spaces, CTRL-0, SHIFT-pound sign.

The characters in 3180 and 3230 can be made with a CMDR-I.

```

10 REM *****
11 REM ***          ***
12 REM **   AIR ATTACK   **
13 REM ***          ***
14 REM *****
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1030 POKE53281,14:POKE53280,14:PRINT"{CLR}"
1040 PRINTTAB(11)"{BLK}*** AIR ATTACK **
*"
1050 PRINT:PRINT:PRINT
1060 PRINT"IN THIS GAME YOU ARE A FIGHTER PILOT. YOU SCORE BY HITTING ONE ";
1065 PRINT"OF THE ENEMY SHIPS WITH ONE OF YOUR BOMBS AND SINKINGIT."
1070 PRINT
1080 PRINT"TO DROP A BOMB, SIMPLY PRESS ANY KEY ON THE KEYBOARD. YOUR SCORE ";
1085 PRINT"FOR HITTING A SHIP WILL DEPEND ON WHICH PART OF THE SHIP YOU HIT."
1090 PRINT
1100 PRINT"IF YOU HIT THE LOWER DECK, YOU SCORE 10 POINTS. IF YOU HIT THE ";
1105 PRINT"UPPER DECK, YOU SCORE 20 POINTS. IF YOU HIT THE SMOKE-STACK ";
1106 PRINT"YOU HAVE DONE VERY WELL, AND ARE REWARDED WITH 30 POINTS."
1110 PRINT:PRINT
1120 PRINT"PRESS *RETURN* TO CONTINUE";
1125 GETX$:IFX$<>CHR$(13)THEN1125
1130 PRINT"{CLR}"
1135 PRINTTAB(11)"{BLK}*** AIR ATTACK **
*"
1140 PRINT:PRINT:PRINT
1145 PRINT"YOU HAVE AN ARSENAL OF 15 BOMBS. THE SPEED OF EACH SHIP WILL ";
1150 PRINT"VARY, SO MAKE EVERY SHOT COUNT!"
1155 PRINT:PRINT:PRINT"GOOD LUCK!!!"

```



```

1160 PRINT:PRINT:PRINT"PRESS *RETURN* TO
CONTINUE";
1170 GETX$:IFX$<>CHR$(13)THEN1170
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2020 PLOT=49152:FORI=0TO24:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 SL=15
2050 PRINT" {CLR} ":POKE53281,14:POKE53280
,14
2060 SYSPLT,0,22:PRINT" {RVS ON} {BLU}
";
2065 SYSPLT,0,23:PRINT" {RVS ON} {BLU}
";

2090 AX=0: SX=33: SS=1
2110 DEFFNCD(X)=PEEK(55296+FX+40*FY)AND1
5
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3040 SYSPLT,0,24:PRINT" {WHT} {RVS OFF} SH
OTS LEFT: "SL" {CSL} "TAB(18)"SCORE: "TS" {C
SL} ";
3050 GOSUB3090:REM PLANE
3060 GOSUB3130:IFSL=0THENRETURN
3070 GOSUB3270:REM SHIP
3080 GOTO3050
3090 :
3100 REM LINE 3110 CHECKS TO SEE IF THE
PLANE IS AT -1. IF SO, THE
3101 REM PLANE IS OFF THE SCREEN. AX
REVERTS TO 33, THE RIGHT HAND
3102 REM SIDE OF THE PICTURE. FROM THERE
THE CYCLE BEGINS AGAIN.
3110 AX=AX-1:IFAX<0THENAX=33:SYSPLT,0,0:
PRINT" {LT BLU} {CSD} {CSL} {CSL} {CSL}
{CSL} {CSL} {CSL} {CSL} ";
3115 SYSPLT,AX,0
3120 PRINT" {RVS OFF} {LT BLU} {WHT} {B
LK} {RVS ON} {RVS OFF} {CSD} {CSL} {CSL} {CS
L} {CSL} {CSL} {CSL} {CSL} {RVS ON} - - {RVS

```

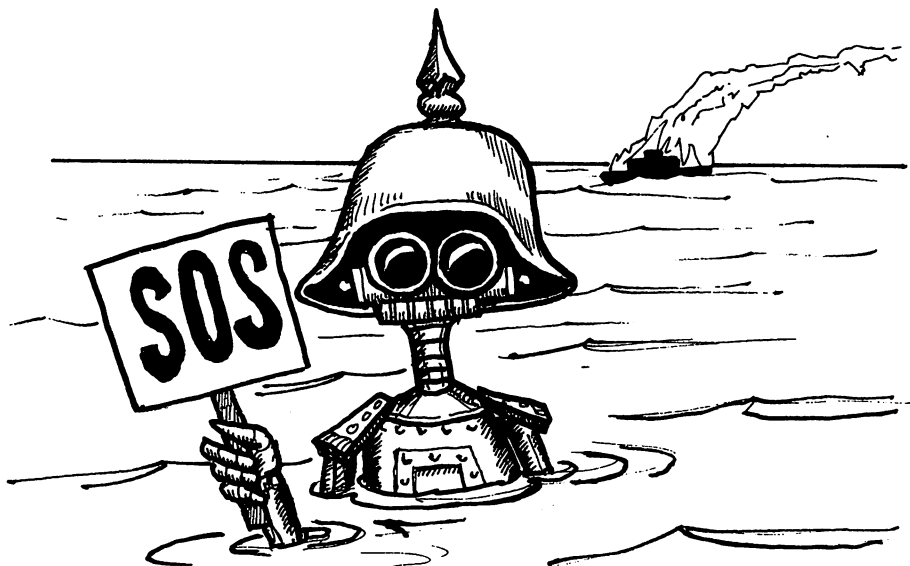


```

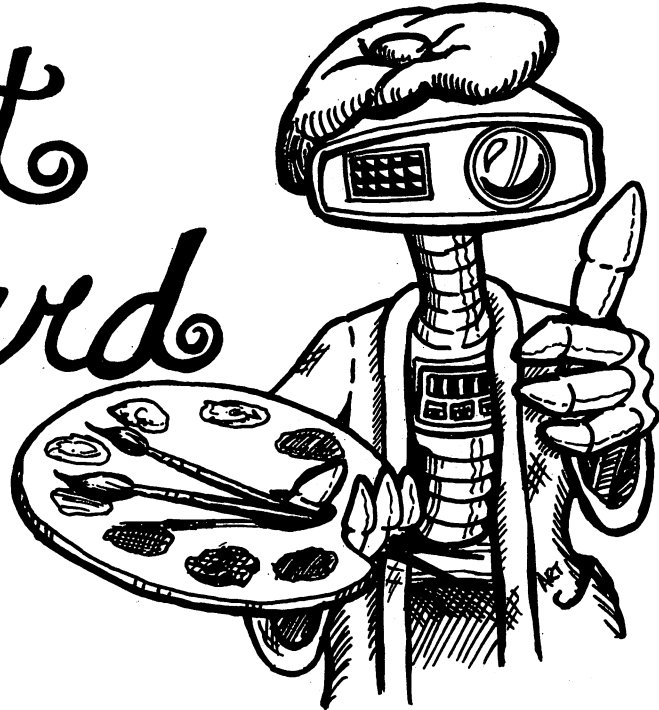
OFF) ";
3125 RETURN
3130 IF FF<>0 THEN3160
3140 GETX$: IFX$=""THENRETURN
3150 FF=1:FX=AX+3:FY=2:BC=1
3160 BC=BC+1: IFBC<2THENRETURN
3165 SYSPLT,FX,FY:PRINT" {RVS OFF} {LT BLU
} ";
3166 BC=0
3170 FY=FY+1
3175 CO=FNC0(0)
3180 IFCO<>15ANDCO<>6THENSYSPLT,FX,FY:PR
INT" {YEL} ";;:RETURN
3190 IFCO=6THEN3230
3200 TS=TS+INT(22-FY+.5)*10: IFFX=5X+4THE
NTS=TS+10
3210 SC=0:SYSPLT, SX, 20:PRINT" {RVS OFF} {L
T BLU} {CSD} {CSL} {CSL} {CSL} {CSL} {C
SL} {CSL} {CSL} ";
3220 SX=33:SS=1:SC=0
3230 SYSPLT,FX-2,20:PRINT" {WHT} {RVS ON
} {RVS OFF} {CSD} {CSL} {CSL} {CSL} {CSL} {R
VS ON} ";;:
3235 FORPA=1T050:NEXTPA
3240 SYSPLT,FX-2,20:PRINT" {RVS OFF} {LT B
LU} {CSD} {CSL} {CSL} {CSL} {CSL} ";
3250 FF=0:SL=SL-1
3260 SYSPLT,0,24:PRINT" {WHT} {RVS OFF}SH
OTS LEFT: "SL" {CSL} "TAB(18)"SCORE: "TS" {C
SL} ";;:RETURN
3270 SC=SC+1: IFSC<SSTHENRETURN
3280 SC=0:SYSPLT, SX, 20:PRINT" {RVS OFF} {L
T BLU} {CSD} {CSL} {CSL} {CSL} {CSL} {C
SL} {CSL} {CSL} ";
3290 SX=SX+1: IFSX>33THENSX=0:SS=1+INT(3*
RND(0))
3300 SYSPLT, SX, 20:PRINT" {GRY 3} {RVS
ON} {RVS OFF} {CSD} {CSL} {CSL} {CSL} {CSL} {
CSL} {CSL} {RVS ON} {RVS OFF}";:RETN
RN
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4030 PRINT" {HOME}":PRINT" {BLK}THE GAME I
S OVER":PRINT"YOUR SCORE OF"TS"IS ";
4040 IFTS<25THENPRINT"ROTTEN":RETURN
4050 IFTS<50THENPRINT"BAD!!!":RETURN
4060 IFTS<75THENPRINT"POOR!!!":RETURN
4070 IFTS<100THENPRINT"FAIR ...":RETURN

```

```
4080 IFTS<150THENPRINT"GOOD ...":RETURN
4090 IFTS<250THENPRINT"GREAT !!":RETURN
4100 IFTS<450THENPRINT"FANTASTIC !!!":RE
TURN
4110 PRINT"PERFECT !!!!":RETURN
4990 RETURN
```



Artist Board



To play Picasso, you must have a brush (a joystick). It is always a good idea to play a game before you attempt an analysis. Load the program and type LIST -1000, and you will be able to see where the main sections of the program are located. Lines 1000 to 2000 print the instructions that you see before each run of the program. Next, list lines 3006-3009. Line 3006 produces a small square of each of the Commodore colors, 3007 shows which letter must be pressed to change the color being drawn with, and 3009 gives all the other commands available for use with the program (which are checked for later in lines 3040 to 3090).

Lines 3010-3020 check the position of the joystick to determine which way the line is being drawn. Take a particularly close look at lines 3013-3016. The variables X and Y are the vertical and horizontal position. In line 3013, if the position of the joystick (JX) evaluates to 8, then the new vertical position (NX) is increased by 1. In line 3014, if the joystick is moved down (IF JX=4) then the new vertical position (NX) is decreased by 1 ($NX=X-1$). The same thing is done for the horizontal coordinates in lines 3015 and 3016.

Lines 3105-3120 issue a disk command to save the file (the picture as it presently appears on the screen) under a name that you specify. Lines 3205-3220 allow you to load in a file from the disk in much the same way. Notice that if you choose option "Q" that you encounter a RETURN in line 3070. This sends you back to line 50, which in turn sends you to the END routine beginning at line 4000.

The big cross in line 3017 can be produced with a SHIFT-plus sign.

```

10 REM *****
11 REM ***          ***
12 REM **  ARTIST BOARD  **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,0:POKE53280,12
1010 PRINT"{CLR}{GRY 3}"TAB(10)"*** ARTI
ST BOARD ***"
1025 PRINT:PRINT
1030 PRINT"BY USING ARTIST BOARD AND YOU
R CREATIVE"
1031 PRINT"TALENT, YOU CAN CREATE MOSAIC
-LIKE"
1032 PRINT"PICTURES.":PRINT
1040 PRINT:PRINT"THE JOYSTICK IS USED TO
MOVE THE CROSS"
1041 PRINT"HAIR (YOUR PAINT BRUSH). "
1042 PRINT:PRINT"PLUG THE JOYSTICK INTO
GAME PORT #1"
1043 PRINT
1045 PRINT"THE TOOLS ARE THERE FOR YOU T
O CREATE"
1046 PRINT"SOME COLORFUL MASTERPIECES...
.GO TO IT!"
1050 PRINT:PRINT"YOU MAY SAVE A DRAWING
ONTO DISK FOR"
1051 PRINT"RECALL AT A LATER TIME.":PRIN
T:PRINT:PRINT
1060 PRINT"PRESS RETURN TO BEGIN PAINTIN
G!"
1065 GETANS$:IFANS$<>CHR$(13)THEN1065
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2005 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXT:POKESID+5,9

```

```

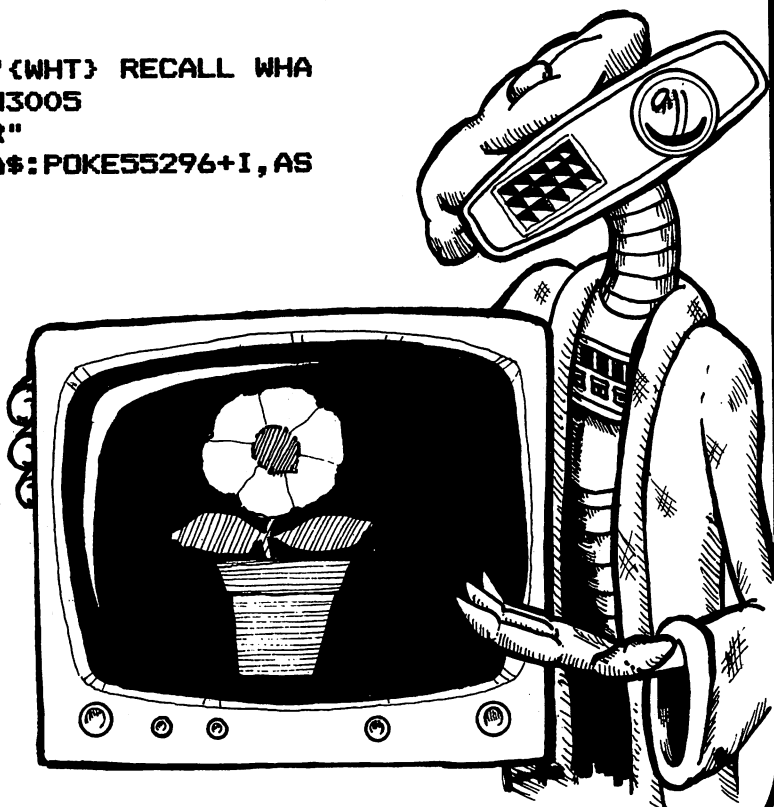
2020 PLOT=49664:FORI=0T024:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,194,132
2022 DATA253,32,16,194
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 CD$="{BLK}{WHT}{RED}{CYN}{PUR}{GRN}
{BLU}{YEL}{ORNG}{BRWN}{LT RED}{GRY 1}{GR
Y 2}{LT GRN}{LT BLU}{GRY 3}"
2130 PDL=49152:FORI=0T063:READC:POKEPDL+
I,C:NEXTI
2131 DATA162,1,120,173,2,220,141,0,193,1
69,192,141,2,220,169
2132 DATA128,141,0,220,160,128,234,136,1
6,252,173,25,212,157
2133 DATA1,193,173,26,212,157,3,193,173,
0,220,9,128,141,5,193
2134 DATA169,64,202,16,222,173,0,193,141
,2,220,173,1,220,141
2135 DATA6,193,88,96
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3003 PRINT"{ORNG}{CLR}";:X=20:Y=10
3004 FORI=1T021:PRINT"{RVS ON}{BLK}
";:NE
XT
3005 SYSPLOT,0,21
3006 PRINT"{RVS ON}{WHT} {BLK} {WHT}
{RED} {CYN} {PUR} {GRN} {BLU} {YE
L} {ORNG} {BRWN} {LT RED} {GRY 1} {
GRY 2} {LT GRN} {LT BLU} {GRY 3} {WH
T} ";
3007 PRINT"{RVS ON}{WHT} A B C D E F
G H I J K L M N O P "
3008 PRINT"{RVS OFF} TYPE LETTER TO
CHANGE COLOR "
3009 PRINT" R=RECALL S=SAVE Q=QUIT
Z=ERASE{BLK}";:POKE56215,1:POKE1943,160
3010 JV=PEEK(56321)AND15:JV=15-JV
3011 JX=JVAND12:JY=JVAND3
3013 IFJX=8THENX=X+1:IFX>39THENX=39
3014 IFJX=4THENX=X-1:IFX<0THENX=0
3015 IFJY=2THENY=Y+1:IFY>20THENY=20
3016 IFJY=1THENY=Y-1:IFY<0THENY=0
3017 SYSPLOT,X,Y:PRINT"{RVS ON} ";:C=PEE
K(55296+X+40*Y)AND15:SYSPLOT,X,Y:PRINT"{

```

```

RVS OFF}{WHT}+";
3018 FORI=1TO30:NEXTI:SYSPLOT,X,Y:PRINT"
{RVS ON}"MID$(C0$,C+1,1)" ";
3020 GETKEY$:IFKEY$=""THEN3010
3040 IFKEY$>="A"ANDKEY$<="P"THENPRINTMID
$(C0$,ASC(KEY$)-ASC("A")+1,1);:GOTO3010
3050 IFKEY$="S"THEN3100
3060 IFKEY$="R"THEN3200
3070 IFKEY$="Q"THENRETURN
3080 IFKEY$="Z"THEN3003
3090 GOTO3010
3100 SYSPLOT,0,23
3101 PRINT"{RVS OFF}"
"
3102 PRINT"{RVS OFF}"
";
3103 FI$=""
3105 SYSPLOT,0,23:INPUT"{WHT} SAVE AS WH
AT FILE";FI$:IFFI$=""THEN3005
3110 OPEN2,8,2,"@:"+FI$+",S,W"
3115 FORI=0TO839:PRINT#2,CHR$(PEEK(55296
+I)AND15);:NEXTI
3120 CLOSE2:GOTO3005
3200 SYSPLOT,0,23
3201 PRINT"{RVS OFF}"
"
3202 PRINT"{RVS OFF}"
";
3203 FI$=""
3205 SYSPLOT,0,23:INPUT"{WHT} RECALL WHA
T FILE";FI$:IFFI$=""THEN3005
3210 OPEN2,8,2,FI$+",S,R"
3215 FORI=0TO839:GET#2,A$:POKE55296+I,AS
C(A$+CHR$(0)):NEXTI
3220 CLOSE2:GOTO3005
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4010 PRINT"{CLR}{WHT}"
4990 RETURN

```



Barrel of Fun



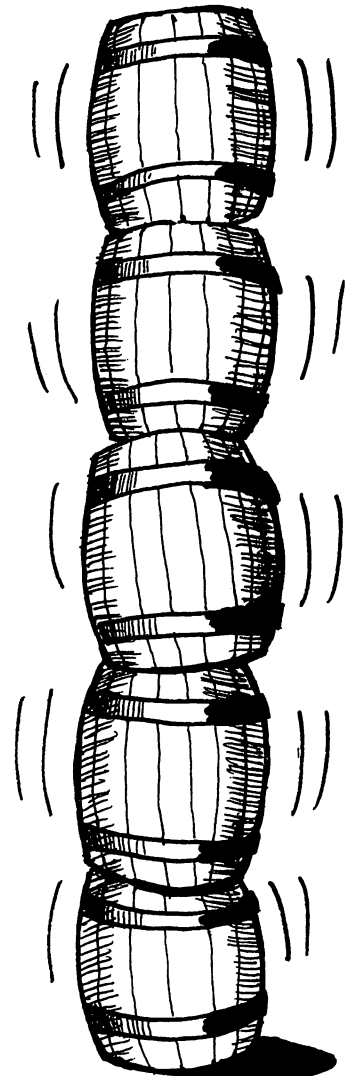
This game has a chance to show off a lot of the colors available on the Commodore 64. In it you are given three rows of colored "barrels" which are stacked on top of each other. The computer then scrambles the barrels and asks you to put them back into order! The problem is, you only have one square in which to move the barrels around. You can move either the rows or columns, but only one at a time. After loading Barrel of Fun, type in LIST -2030. After the machine language routine, line 2030 draws the vertical bars across the screen that separate the rows of barrels. The actual designing of the board is done in lines 2120-2160. Next, list lines 3000-3045. This is the routine that gets input from the keyboard to determine where your next move will be. Line 3030 checks for column rotations and lines 3035-3040 check for horizontal movement. From this section, you will be able to see where certain tasks are performed in the program. Remember that when you don't understand what a line does, try to remove it and see if anything in the program changes. Sometimes, this line by line method is the only way to decipher a program. You'll be surprised at how much you'll learn about programming along the way!

The line in 2034 can be produced by a series of CTRL-C's.

```

10 REM *****
11 REM ***          ***
12 REM ** BARREL OF FUN **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,12:POKE53280,12
1010 PRINT"{CLR}{BLK}";
1020 PRINTTAB(10)"*** BARREL OF FUN ***"
:PRINT:PRINT
1030 PRINT"IN BARREL OF FUN YOU WILL SEE
THREE"
1031 PRINT"COLUMNS OF SIX COLORFUL BARRE
LS. THE"
1032 PRINT"OBJECT OF THE GAME IS TO ALLI
GN THE ROWS";
1033 PRINT"SUCH THAT EACH ROW IS A SOLID
COLOR."
1034 PRINT"THERE ARE ONLY TWO WHITE BARR
ELS. THE"
1035 PRINT"EMPTY SQUARE MUST EVENTUALLY
COMPLETE"
1036 PRINT"THE WHITE ROW."
1040 PRINT:PRINT"THE BOARD OF SQUARES WI
LL BE MIXED UP,"
1041 PRINT"AND YOUR TASK IS TO UNSCRAMBL
E IT. THE ";
1042 PRINT"FINAL PRODUCT SHOULD HAVE ALL
OF THE"
1043 PRINT"SQUARES OF ONE COLOR LINE UP
IN A"
1044 PRINT"HORIZONTAL ROW."
1045 PRINT
1050 PRINT"A SAMPLE OF HOW THE FINISHED
PRODUCT"
1051 PRINT"SHOULD LOOK WILL BE SHOWN TO

```



```

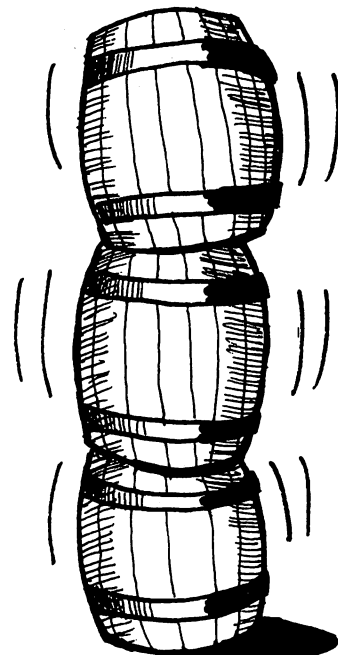
YOU."
1052 PRINT
1060 PRINT"PRESS RETURN TO CONTINUE."
1065 GETANS$:IFANS$<>CHR$(13)THEN1065
1070 PRINT"{CLR}{BLK}";
1071 PRINTTAB(10)"*** BARREL OF FUN ***"
:PRINT:PRINT
1080 PRINT"EACH COLUMN OF SQUARES (COLUM
NS 1,2,3)"
1081 PRINT"CAN BE ROTATED VERTICALLY BY
ENTERING"
1082 PRINT"THE NUMBER OF THE COLUMN YOU
WISH TO"
1083 PRINT"ROTATE."
1090 PRINT:PRINT"YOU CAN MOVE COLORED SQ
UARES INTO THE"
1091 PRINT"EMPTY SQUARE BY USING THE F1
AND F3"
1092 PRINT"KEYS. IF THE COLORED SQUARE
IS TO BE"
1093 PRINT"MOVED TO THE EMPTY SQUARE ON
ITS RIGHT"
1094 PRINT"THEN PRESS F3."
1095 PRINT:PRINT"IF THE COLORED SQUARE I
S TO BE MOVED TO"
1096 PRINT"THE EMPTY SQUARE ON ITS LEFT
THEN PRESS"
1097 PRINT"F1."
1100 PRINT
1101 PRINT"PRESS RETURN TO CONTINUE."
1105 GETANS$:IFANS$<>CHR$(13)THEN1105
1120 PRINT"{CLR}{BLK}";
1121 PRINTTAB(10)"*** BARREL OF FUN ***"
:PRINT:PRINT
1130 PRINT"NOTE THAT IN THE FINAL SOLUTI
ON ALL"
1131 PRINT"HORIZONTAL ROWS MUST BE THE S
AME COLOR."
1132 PRINT:PRINT"IT DOESN'T MATTER WHERE
THE ROW IS"
1133 PRINT"LOCATED, AS LONG AS ALL OF TH
E COLORS"
1134 PRINT"WITHIN THE ROW ARE THE SAME."
1140 PRINT
1141 PRINT"PRESS RETURN TO CONTINUE."
1145 GETANS$:IFANS$<>CHR$(13)THEN1145
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :

```

```

2005 DIMBA(3,6),CO$(6):FORI=0TO6:READCO$(I):NEXT
2006 DATA"{GRY 2}","{BLK}","{RED}","{LT GRN}","{BLU}","{CYN}","{WHT}"
2010 FORI=1TO6:FORJ=1TO3:BA(J,I)=I:NEXT:
NEXT:BA(3,6)=0:BX=3:BY=6
2020 PLOT=49152:FORI=0TO24:READC:POKEPLT+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 PRINT"{CLR}{YEL}"
2034 FORI=0TO18STEP3:SYSPLOT,11,I:PRINT"{RVS ON}-----":NEXT
2040 FORI=1TO3:GOSUB2500:NEXT
2100 SYSPLOT,0,20:FORI=1TO3:PRINT"{RVS OFF}"
":NEXT
2105 SYSPLOT,5,20:PRINT"{BLK}<<< THIS IS THE FINAL PATTERN >>>"
2110 FORPA=1TO2000:NEXTPA
2115 SYSPLOT,0,20:PRINT"{RVS OFF}"
"
2116 SYSPLOT,3,20:PRINT"{BLK}<<< I'M NOW SCRAMBLING THE BOARD >>>"
2120 FORN=1TO20+INT(10*RND(0))
2130 IFBX=3THENRX=2:GOTO2140
2131 IFBX=1THENRX=2:GOTO2140
2132 RX=1+2*INT(2*RND(0))
2140 FORL=1TO1+INT(5*RND(0)):FORM=1TO6:BA(RX,M-1)=BA(RX,M):NEXTM
2141 BA(RX,6)=BA(RX,0):I=RX:NEXTL:GOSUB2500
2145 BA(BX,BY)=BA(RX,BY):BA(RX,BY)=0
2150 I=BX:J=BY:GOSUB2510
2155 BX=RX:I=BX:J=BY:GOSUB2510
2160 NEXTN:RETURN
2200 RETURN
2230 FORN=1TO20+INT(10*RND(0))
2235 IFBX=3THENRX=2:GOTO
2500 FORJ=1TO6:GOSUB2510:NEXTJ:RETURN
2510 PRINTC$(BA(I,J));
2520 FORK=0TO1:SYSPLOT,(I+1)*6,J*3-2+K:PRINT"{RVS ON}      "":NEXTK:RETURN
2990 RETURN
3000 :
3001 :REM *** PLAY

```



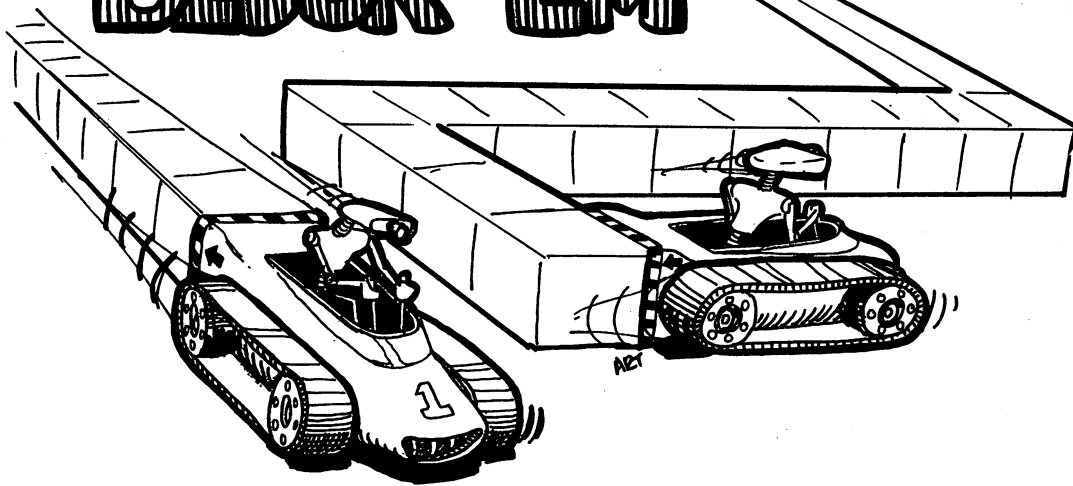

```

3002 :
3010 SYSPLOT,0,20:PRINT" {RVS OFF}
"
3011 SYSPLOT,6,20:PRINT" {BLK} {RVS ON} T
YPE 1 TO ROTATE COLUMN 1 "
3012 PRINTTAB(6)" {RVS ON} TYPE 2 TO ROT
ATE COLUMN 2 "
3013 PRINTTAB(6)" {RVS ON} TYPE 3 TO ROT
ATE COLUMN 3 "
3014 PRINTTAB(6)" {RVS ON} F1 SHIFT LEFT
F3 SHIFT RIGHT";
3020 GETKEY$: IFKEY$="" THEN 3020
3030 IFKEY$="1" OR KEY$="2" OR KEY$="3" THEN 3
100
3035 IFKEY$="{F1}" THEN 3050
3040 IFKEY$="{F3}" THEN 3075
3045 GOTO 3020
3050 RX=BX+1: IFRX=4 THEN 3020
3055 BA(BX, BY)=BA(RX, BY): BA(RX, BY)=0
3060 I=BX: J=BY: GOSUB 2510: BX=RX: I=BX: J=BY
: GOSUB 2510
3065 GOTO 3500
3075 RX=BX-1: IFRX=0 THEN 3020
3080 BA(BX, BY)=BA(RX, BY): BA(RX, BY)=0
3085 I=BX: J=BY: GOSUB 2510: BX=RX: I=BX: J=BY
: GOSUB 2510
3090 GOTO 3500
3100 RX=VAL(KEY$)
3110 FORM=1 TO 6: BA(RX, M-1)=BA(RX, M): NEXT M
: BA(RX, 6)=BA(RX, 0): I=RX: GOSUB 2500
3120 IFRX=BX THEN BY=BY-1: IFBY=0 THEN BY=6
3500 BA(BX, BY)=6: FOR I=1 TO 6
3510 IF BA(1, I)=BA(2, I) AND BA(2, I)=BA(3, I)
THEN NEXT I: RETURN
3520 BA(BX, BY)=0: TRY=TRY+1: GOTO 3020
3990 RETURN
4000 :
4001 : REM *** END
4002 :
4990 RETURN

```



BLOCK'EM



This is a two player game in which you try to draw a line longer than your opponent. If you collide with either the trail of your opponent or one of the borders, you lose. List through line 2026. There is a general purpose sound routine in lines 2001-2026. Line 2015 clears out 25 bytes of memory that the SID chip uses (POKEing each with a 0), then reads in the DATA elements in lines 2021-2026. Lines 2100-2135 create the "countdown" routine. This is just the kind of routine that can be transferred easily from game to game. List lines 3000-3100. Line 3010 creates the inverse yellow circle and the inverse blue diamond that are used as the player pieces. Line 3020 reads the keyboard and checks to see which key is being pressed. Note that if no key is being pressed (IF KEY\$ = ""), the program branches to the NEXT in line 3050, skipping all the IF statements in lines 3030-3043. This program has a lot of routines that perform specific functions. As always, feel free to experiment with the routines if you don't quite understand what each of them does.

In line 3010, the diamond can be produced with a SHIFT-Z, and the circle with a SHIFT-Q.

```

10 REM *****
11 REM ***          ***
12 REM **   BLOCK 'EM   **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1010 PRINT"{CLR}":POKE53281,12:POKE53280
,12
1020 PRINTTAB(12)"{BLU}*** BLOCK 'EM ***
"
1030 PRINT:PRINT:PRINT
1031 PRINT"IN THIS GAME, TWO PLAYERS CON
TROL THE"
1032 PRINT"CREATION OF A LINE."
1035 PRINT
1040 PRINT"THE FIRST PLAYER WHOSE LINE H
ITS A WALL"
1045 PRINT"OR THE OTHER PLAYER'S LINE, L
OSES THE"
1046 PRINT"GAME."
1050 PRINT:PRINT"PLAYER #1      DIRECTION
PLAYER #2":PRINT
1051 PRINT"  W                UP
I":PRINT
1052 PRINT"A      S      LEFT  RIGHT
J      K":PRINT
1053 PRINT"  Z                DOWN
M"
1090 PRINT:PRINT:PRINT"PRESS RETURN WHEN
READY TO CONTINUE"
1095 GETANS$:IFANS$<>CHR$(13)THEN1095
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2005 SID=54272

```

```

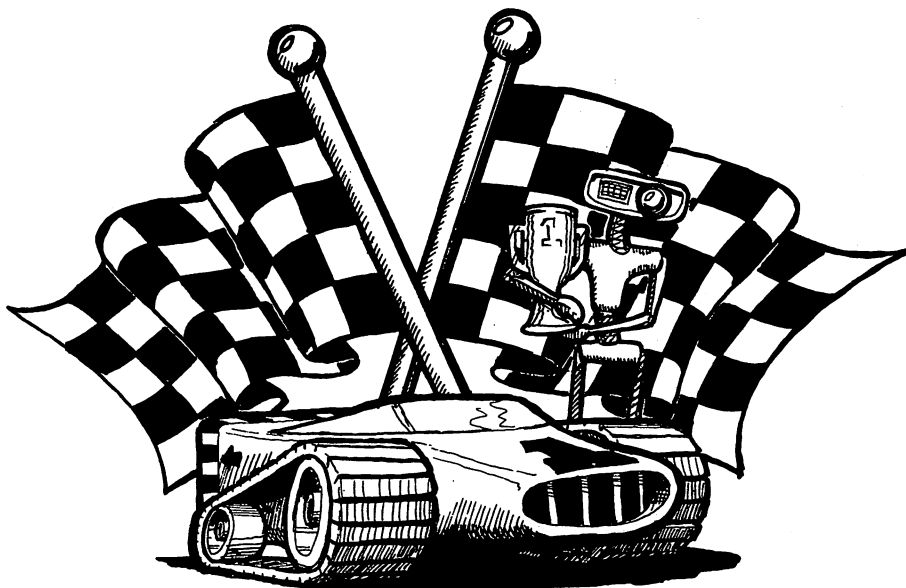
2010 POKE53280,0
2015 FORI=SIDTOSID+24:POKEI,0:NEXT
2016 POKESI+24,15:POKESI+5,88:POKESI+6,1
95
2020 PLOT=49152:FORI=OT024:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2100 PRINT"{CLR}":FORI=5TO1STEP-1
2110 SYSPLT,19,13:PRINT"{RVS ON}{BLK}"I
2130 FORPA=1TO300:NEXTPA
2135 NEXT
2140 SYSPLT,19,13:PRINT"  "
2150 X1=1+INT(5*RND(0)):Y1=6+INT(6*RND(0
))
2151 X2=39-INT(5*RND(0)):Y2=6+INT(6*RND(
0))
2152 D1=4:D2=3
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 SYSPLT,X1,Y1:PRINT"{RVS ON}{BLU}#";
:SYSPLT,X2,Y2:PRINT"{RVS ON}{YEL}#";
3015 NT=5+(MD<35)*(30-INT(MD/15)*15)
3020 FORI=1TONT:GETKEY$:IFKEY$=""THEN305
0
3030 IFKEY$="A"THEND1=3
3031 IFKEY$="S"THEND1=4
3032 IFKEY$="W"THEND1=1
3033 IFKEY$="Z"THEND1=2
3040 IFKEY$="J"THEND2=3
3041 IFKEY$="K"THEND2=4
3042 IFKEY$="I"THEND2=1
3043 IFKEY$="M"THEND2=2
3050 NEXT
3100 OND1GOTO3110,3120,3130,3140
3110 Y1=Y1-1:GOTO3150
3120 Y1=Y1+1:GOTO3150
3130 X1=X1-1:GOTO3150
3140 X1=X1+1
3150 IFX1<0ORX1>39ORY1<1ORY1>24THENWL=2:
RETURN
3155 IFPEEK(1024+X1+40*Y1)<>32THENWL=2:R
ETURN
3200 OND2GOTO3210,3220,3230,3240

```

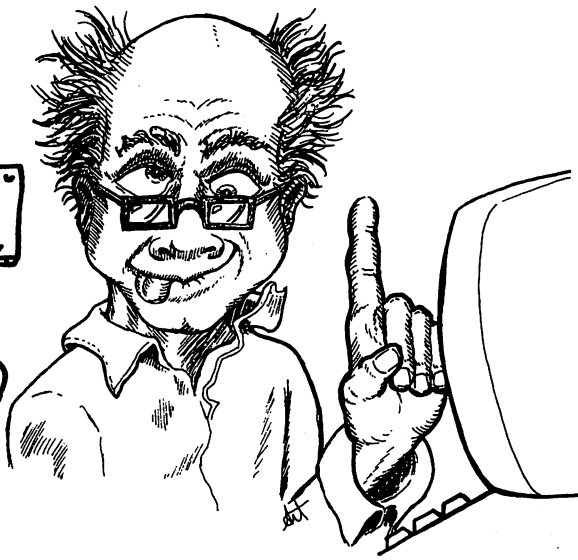
```

3210 Y2=Y2-1:GOTO3250
3220 Y2=Y2+1:GOTO3250
3230 X2=X2-1:GOTO3250
3240 X2=X2+1
3250 IFPEEK(1024+X2+40*Y2)<>32THENWL=1:R
ETURN
3255 IFX2<0ORX2>39ORY2<1ORY2>24THENWL=1:
RETURN
3260 POKESID,0:POKESID+1,90:POKESID+4,33
:FORD=1T050:POKESID+4,32
3300 MO=MO+1:GOTO3010
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4003 POKESID+1,0
4010 POKESID+24,0:PRINT"{HOME}{RVS OFF}{
BLK}THE GAME IS OVER...."
4020 PRINT"PLAYER #"WL"HAS WON THE GAME
!!!"
4990 RETURN

```



BRAIN TEASER



This game is indeed an intellectual challenge. Random guessing will rarely net you the correct solution. It is a good idea to conceptualize how you intend to achieve your goal. This game makes use of some difficult graphic and logic routines, so don't worry if you don't understand what every section does. List through line 3100. Line 3045 reads the keyboard to see which number has been chosen. Line 3076 checks to see if that box is occupied. This is a good example of a flag, a technique used in writing long programs. If the value in $BO(I,J)$ is equal to 1, then that box is occupied. Moving to a square that is not occupied will cause $BO(I,J)$ to evaluate to a -1 and send control to line 3077, then back to 3030 for another number. Line 3080 determines which of the subroutines in lines 3110-3193 the program is sent to. Note that there are nine subroutines in this section, one for each square. This is exactly the type of situation where an ON/GOTO structure is useful — one subroutine for each possible answer. The shapes used on the board are drawn in lines 3500-3525. Experiment with different shapes to give your game board a unique appearance.

The diamond shape may be produced with a SHIFT-Z.

The circle shape may be produced with a SHIFT-Q.

The thick lines in 3520 can be made with a CMDR-I, and with a CMDR-C, CMDR-V combination.

```

10 REM *****
11 REM ***          ***
12 REM **  BRAIN TEASER  **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1010 PRINT"{CLR}{WHT}":POKE53281,14:POKE
53280,0
1020 PRINTTAB(10)"*** BRAIN TEASER ***"
1030 PRINT
1031 PRINT"IN THIS GAME YOU ARE GIVEN A
3 BY 3 GAME";
1032 PRINT"BOARD WITH ONE OCCUPIED SPACE
(THE"
1033 PRINT"CENTER). THE BOARD WILL LOOK
LIKE THIS"
1040 PRINT:PRINT"{BLU}      {RVS ON} {RVS
OFF} {RVS ON} {RVS OFF} {RVS ON} {RVS OF
F}"
1041 PRINT
1042 PRINT"      {RVS ON} {RVS OFF} {RVS O
N}*{RVS OFF} {RVS ON} {RVS OFF}"
1043 PRINT
1044 PRINT"      {RVS ON} {RVS OFF} {RVS O
N} {RVS OFF} {RVS ON} {RVS OFF}"
1045 PRINT
1050 PRINT"{WHT}THE TRICK IS TO MOVE THE
PIECES SO THAT"
1051 PRINT"THE GAME BOARD WINDS UP LOOKI
NG LIKE"
1052 PRINT"THIS:":PRINT
1060 PRINT"{BLU}      {RVS ON}*{RVS OFF} {
RVS ON}*{RVS OFF} {RVS ON}*{RVS OFF}
{WHT}AND{BLU}      {RVS ON} {RVS OFF}
{RVS ON} {RVS OFF} {RVS ON} {RVS OFF}"
1061 PRINT
1062 PRINT"{BLU}      {RVS ON}*{RVS OFF} {
RVS ON} {RVS OFF} {RVS ON}*{RVS OFF}

```



```

{WHT}NOT LIKE {BLU}{RVS ON} {RVS OFF}
{RVS ON} {RVS OFF} {RVS ON} {RVS OFF}"
1063 PRINT
1064 PRINT"{BLU} {RVS ON}*{RVS OFF} {
RVS ON}*{RVS OFF} {RVS ON}*{RVS OFF}
{WHT}THIS:{BLU} {BLU}{RVS ON} {RVS
OFF} {RVS ON} {RVS OFF} {RVS ON} {RVS O
FF}"
1070 PRINT:PRINT:PRINT"{WHT}PRESS RETURN
WHEN READY TO CONTINUE.";
1075 GETANS$: IFANS$<>CHR$(13) THEN 1075
1080 PRINT"{CLR}{WHT}"
1081 PRINTTAB(10)"*** BRAIN TEASER ***"
1091 PRINT
1092 PRINT"YOU MAY ONLY MOVE TO AN OCCUP
IED SPACE"
1093 PRINT"(A SPACE WITH A * ON IT). WH
EN YOU MOVE";
1094 PRINT",CERTAIN SQUARES WILL REVERSE
THEIR"
1095 PRINT"STATE (CHANGE FROM A * TO A B
LANK, OR"
1096 PRINT"VICE-VERSA). "
1099 PRINT
1100 PRINT"IF YOU MOVE TO A CORNER, ALL
OF THE ADJACENT SQUARES REVERSE."
1102 PRINT"IF YOU MOVE TO THE MIDDLE OF
A SIDE, ALL";
1103 PRINT"OF THE SQUARES ON THAT SIDE W
ILL FILL UP";
1104 PRINT", AND IF YOU CHOOSE THE CENTE
R SQUARE,"
1105 PRINT"THAT BOX AND THE FOUR MIDDLE
BOXES WILL ALL BE REVERSED."
1110 PRINT:PRINT:PRINT"{WHT}PRESS RETURN
WHEN READY TO CONTINUE.";
1120 GETANS$: IFANS$<>CHR$(13) THEN 1120
1125 PRINT"{CLR}{WHT}"
1130 PRINTTAB(10)"*** BRAIN TEASER ***"
1131 PRINT
1132 PRINT"HERE IS A QUICK REVIEW OF THE
VARIOUS"
1133 PRINT"MOVES, AND THE RESULTING REVE
RSALS....":PRINT
1140 PRINT"{BLU} {RVS ON}*{RVS OFF} {
RVS ON}*{RVS OFF} {RVS ON} {RVS OFF}
{RVS ON}*{RVS OFF} {RVS ON}*{RVS OFF} {
RVS ON}*{RVS OFF} {RVS ON} {RVS OFF}
{RVS ON}*{RVS OFF} {RVS ON} {RVS OFF}"
1141 PRINT

```

```

1142 PRINT"{BLU}      {RVS ON}*{RVS OFF} {
RVS ON}*{RVS OFF} {RVS ON} {RVS OFF}
  {RVS ON} {RVS OFF} {RVS ON} {RVS OFF} {
RVS ON} {RVS OFF}      {RVS ON}*{RVS OFF}
{RVS ON}*{RVS OFF} {RVS ON}*{RVS OFF}"
1143 PRINT
1144 PRINT"{BLU}      {RVS ON} {RVS OFF} {
RVS ON} {RVS OFF} {RVS ON} {RVS OFF}
  {RVS ON} {RVS OFF} {RVS ON} {RVS OFF} {
RVS ON} {RVS OFF}      {RVS ON} {RVS OFF}
{RVS ON}*{RVS OFF} {RVS ON} {RVS OFF}"
1148 PRINT"{WHT}"
1150 PRINT"THE # DENOTES THE MOVE POSITI
ON AND THE"
1151 PRINT"'*'S DENOTE THE PIECES THAT W
ILL BE"
1152 PRINT"FLIPPED.  THE PIECE AT '# ' WI
LL ALSO BE"
1153 PRINT"FLIPPED.  THE BOARD IS DESIGN
ATED LIKE THIS:":PRINT
1160 PRINT"{BLU}      {RVS ON}1{RVS OFF} {
RVS ON}2{RVS OFF} {RVS ON}3{RVS OFF}"
1161 PRINT
1162 PRINT"      {RVS ON}4{RVS OFF} {RVS O
N}5{RVS OFF} {RVS ON}6{RVS OFF}"
1163 PRINT
1164 PRINT"      {RVS ON}7{RVS OFF} {RVS O
N}8{RVS OFF} {RVS ON}9{RVS OFF}"
1170 PRINT:PRINT"{WHT}PRESS RETURN WHEN
READY TO CONTINUE.";
1175 GETANS$:IFANS$(<>CHR$(13))THEN1175
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2010 DIMBO(3,3):FORI=1TO3:FORJ=1TO3:BO(I
,J)=-1:NEXT:NEXT:BO(2,2)=1
2020 PLOT=49152:FORI=0TO24:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 PRINT"{CLR}":FORI=1TO3:FORJ=1TO3:GO

```

```

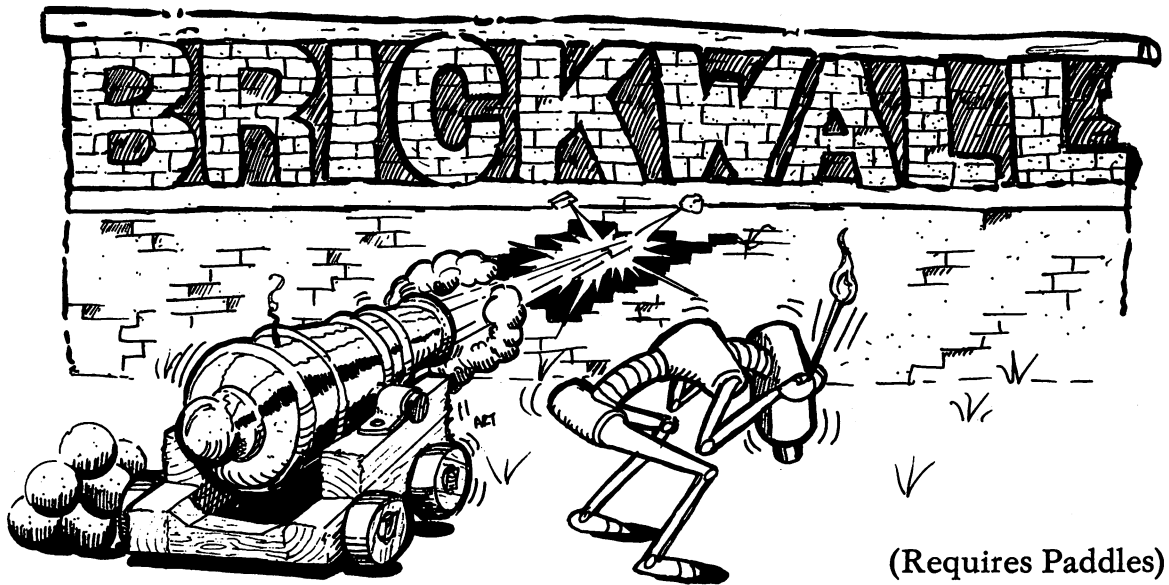
SUB3505: NEXT: NEXT
3020 FORI=1TO3: FORJ=1TO3: IFBO(I, J)=1OR(I
=2ANDJ=2) THENNEXT: NEXT: WL=1: RETURN
3025 FORI=1TO3: FORJ=1TO3: IFBO(I, J)=-1THE
NNEXT: NEXT: WL=0: RETURN
3030 SYSPLOT, 0, 19: FORI=1TO4: PRINT" {RVS O
FF}
"
3031 NEXT
3035 SYSPLOT, 0, 19: PRINT" {BLK}  ENTER A P
OSITION (1-9) OR"
3040 PRINT"  RETURN TO QUIT ==> {RVS ON
} {RVS OFF}{CSL}";
3045 GETKEY$: IFKEY$="" THEN3045
3050 PRINTKEY$
3060 IFKEY$=CHR$(13) THENRETURN
3075 KEY=VAL(KEY$): I=1+INT((KEY-1)/3): J=
KEY-3*(I-1)
3076 IFBO(I, J)=1 THEN3080
3077 PRINT"  MOVE ONLY TO AN OCCUPIED SQ
UARE"
3078 FORPA=1TO1500: NEXT: GOTO3030
3080 ONKEYGOTO3110, 3120, 3130, 3140, 3150, 3
160, 3170, 3180, 3190
3110 I=1: J=1: GOSUB3500
3111 I=1: J=2: GOSUB3500
3112 I=2: J=1: GOSUB3500
3113 I=2: J=2: GOSUB3500: GOTO3020
3120 I=1: J=1: GOSUB3500
3121 I=1: J=2: GOSUB3500
3122 I=1: J=3: GOSUB3500: GOTO3020
3130 I=1: J=2: GOSUB3500
3131 I=1: J=3: GOSUB3500
3132 I=2: J=2: GOSUB3500
3133 I=2: J=3: GOSUB3500: GOTO3020
3140 I=1: J=1: GOSUB3500
3141 I=2: J=1: GOSUB3500
3142 I=3: J=1: GOSUB3500: GOTO3020
3150 I=1: J=2: GOSUB3500
3151 I=2: J=1: GOSUB3500
3152 I=2: J=2: GOSUB3500
3153 I=2: J=3: GOSUB3500
3154 I=3: J=2: GOSUB3500: GOTO3020
3160 I=1: J=3: GOSUB3500
3161 I=2: J=3: GOSUB3500
3162 I=3: J=3: GOSUB3500: GOTO3020
3170 I=2: J=1: GOSUB3500
3171 I=2: J=2: GOSUB3500
3172 I=3: J=1: GOSUB3500
3173 I=3: J=2: GOSUB3500: GOTO3020

```

```

3180 I=3:J=1:GOSUB3500
3181 I=3:J=2:GOSUB3500
3182 I=3:J=3:GOSUB3500:GOTO3020
3190 I=2:J=2:GOSUB3500
3191 I=2:J=3:GOSUB3500
3192 I=3:J=2:GOSUB3500
3193 I=3:J=3:GOSUB3500:GOTO3020
3500 BO(I,J)=-BO(I,J)
3505 PRINTMID$(" {YEL} {BLU}",BO(I,J)+2,1
);
3510 IX=4*I:JX=8*J:FORI2=IXTOIX+3:SYSPLO
T,JX,I2:PRINT"{RVS ON}          ";;NEXTI2
3520 IFBO(I,J)=-1THENSYSPLOT,JX+3,IX+1:P
RINT"{BLK}{RVS ON} = {CSD}{CSL}{CSL} = ";;
RETURN
3525 SYSPLOT,JX+2,IX:PRINT"{BLU}{RVS ON}
* * {CSD}{CSL}{CSL}{CSL}**{CSD}{CSL}{CSL
}* **{CSD}{CSL}{CSL}{CSL}* *";
3540 RETURN
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4005 SYSPLOT,0,21
4010 PRINT"{WHT}{RVS OFF} THE GAME IS O
VER !!!"
4020 IFWL=0THENPRINT" AND YOU LOST... S
ORRY"
4030 IFWL=1THENPRINT" AND YOU WON.... G
REAT!!!"
4040 PRINT"{HOME}";
4990 RETURN

```



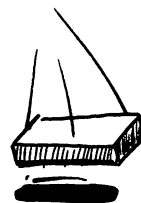
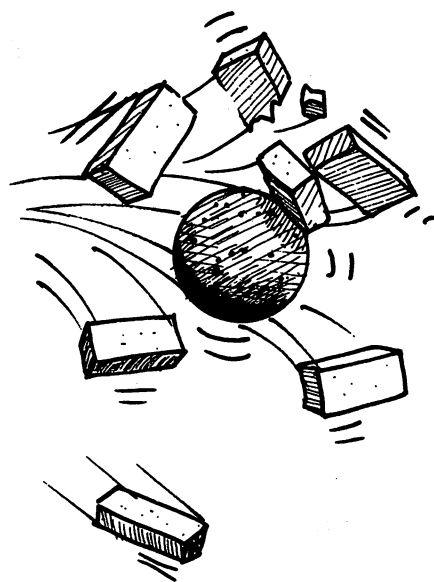
What a challenge! This game tests your quickness and dexterity. The game is similar to some of the arcade games you see. With a little practice, YOU could write a program like this. This program requires that you have a paddle (a joystick won't work) to control the movement of the paddle in the game. Listing lines 2000-2999 will show the setup section of the game, and the various subroutines are labeled so that their functions are more easily understood. Notice that the sections following lines 2100, 2200, and 2300 each perform a function specified by the REMark statement in each of those lines. Although some programmers don't use REM statements, it is usually considered good practice to do so, especially if you are working on a long program or there will be long periods of time between your work sessions. This lets you pick up where you left off with a minimum of confusion about how your program works.

The line in 2220 can be produced with a series of CMDR-I's. The circle can be produced with a SHIFT-Q.

```

10 REM *****
11 REM ***          ***
12 REM **   BRICK WALL   **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53280,14:POKE53281,6
1010 PRINT"{CLR}{CYN}"TAB(12)"*** BRICK
WALL ***"
1020 PRINT:PRINT
1030 PRINT"IN THIS GAME, YOU WILL BE PRE
SENTED"
1031 PRINT"WITH A WALL OF BRICKS AT THE
TOP OF THE"
1032 PRINT"SCREEN, AND A PADDLE AT THE B
OTTOM."
1033 PRINT"THE GAME PADDLE (PORT 2) IS U
SED TO HIT"
1034 PRINT"A ROCK INTO THE WALL OF BRICK
S.":PRINT
1040 PRINT"WHEN THE ROCK HITS A BRICK, I
T WILL"
1041 PRINT"DESTROY IT AND POINTS WILL BE
ADDED TO"
1042 PRINT"YOUR SCORE.":PRINT
1050 PRINT"YOUR MISSION, SHOULD YOU ACCE
PT, IS TO"
1051 PRINT"DESTROY AS MUCH OF THE BRICK
WALL AS"
1052 PRINT"POSSIBLE.":PRINT:PRINT
1053 PRINT"YOU ARE ALLOWED ONLY FIVE MIS
SES."
1054 PRINT:PRINT"PRESS RETURN TO BEGIN P
LAY."
1060 GETANS$:IFANS$<>CHR$(13)THEN1060
1990 RETURN
2000 :

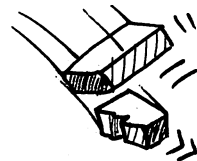
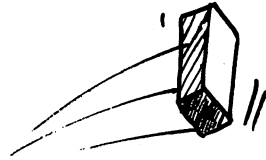
```



```

2001 :REM *** SETUP
2002 :
2010 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXT:POKESID+5,9
2020 PLOT=49664:FORI=0T024:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,194,132
2022 DATA253,32,16,194
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 GOSUB2100
2040 DIMDIR(6):FORI=1T06:READDIR(I):NEXT
:DATA-1.5,-1,-.5,.5,1,1.5
2050 PDL=49152:FORI=0T063:READC:POKEPDL+
I,C:NEXTI
2051 DATA162,1,120,173,2,220,141,0,193,1
69,192,141,2,220,169
2052 DATA128,141,0,220,160,128,234,136,1
6,252,173,25,212,157
2053 DATA1,193,173,26,212,157,3,193,173,
0,220,9,128,141,5,193
2054 DATA169,64,202,16,222,173,0,193,141
,2,220,173,1,220,141
2055 DATA6,193,88,96
2090 RETURN
2100 REM *** DO BACKGROUND
2105 PRINT"(CLR)"
2110 FORI=3T011STEP2:K=(I-1)/2:K=K-2*INT
(K/2)
2115 PRINTMID$("{CYN}{RED}",K+1,1);:FORJ
=0T036STEP4:SYSLOT,J,I:PRINT"{RVS ON}"
";:NEXTJ
2120 PRINTMID$("{CYN}{RED}",2-K,1);:FORJ
=2T038STEP4:SYSLOT,J,I:PRINT"{RVS ON}"
";:NEXTJ
2125 NEXTI
2130 FORI=56256T056295:POKEI,1:NEXT
2190 RETURN
2200 REM *** DRAW PADDLE
2201 SYSPDL:P=(236-PEEK(PDL+258))/6:BU=P
EEK(PDL+261)
2202 IFP<0THENP=0
2203 IFP>34THENP=34
2210 IFP=PPTHENRETURN
2215 SYSLOT,PP,23:PRINT"{RVS OFF}"
";
2220 SYSLOT,P,23:PRINT"{RVS OFF}{WHT}
.....";

```



```

2225 PP=P:RETURN
2300 REM *** MOVE BALL
2305 X2=BX+DI(BA):Y2=BY+BD
2310 IFX2=>OANDX2<=39THEN2315
2311 X2=BX-DI(BA):BA=7-BA
2312 POKESID+1,220:POKESID+4,33:POKESID+
24,15:FORPA=1TO30:NEXT
2313 POKESID+1,0:POKESID+4,32
2315 IFY2=>OTHEN2320
2316 Y2=BY-BD:BD=-BD:BF=1
2317 POKESID+1,200:POKESID+4,33:POKESID+
24,15:FORPA=1TO30:NEXT
2318 POKESID+1,0:POKESID+4,32
2320 IFY2>23THENSYSLOT,BX,BY:PRINT"(RVS
OFF) ";;GOTO3060
2325 CH=PEEK(1024+X2+40*Y2):IFCH<>121THE
N2330
2326 POKESID+1,240:POKESID+4,33:POKESID+
24,15:FORPA=1TO30:NEXT
2327 POKESID+1,0:POKESID+4,32
2328 BD=-BD:BF=0:BA=INT(X2)-INT(PP)+1:PB
=PB+1:IFPB=7THENBD=-2
2329 SYSLOT,BX,BY:PRINT"(RVS OFF) ";;SY
SLOT,X2,Y2:PRINT"(WHT)##";:BX=X2:BY=Y2:G
OTO3040
2330 IFCH=32THEN2340
2331 X3=2*INT(X2/2):SYSLOT,X3,Y2:PRINT"
(RVS OFF) ";;SC=SC+INT(5-(Y2-3)/2)
2332 POKESID+1,10:POKESID+4,33:POKESID+2
4,15:FORPA=1TO30:NEXT
2333 POKESID+1,0:POKESID+4,32
2334 IFSC=>300THEN3065
2335 IFBD<0ORBF=1THENBD=-BD
2336 SYSLOT,11,24:PRINT"(RVS ON){CYN}"S
C;
2340 IFPEEK(1024+BX+40*BY)<>81THEN2345
2341 SYSLOT,BX,BY:PRINT"(RVS OFF) ";
2345 SYSLOT,X2,Y2:PRINT"(RVS OFF){WHT}##
";:BX=X2:BY=Y2:GOTO3040
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3005 SYSLOT,0,24:PRINT"(RVS ON){CYN}
SCORE: 0 BALLS LEFT: ";
3006 POKESID+1,3:POKESID+4,160
3010 FORB=1TO5:SYSLOT,31,24:PRINT"(RVS
ON){CYN}"5-B;
3015 FORJ=1TO100:GOSUB2200:IFBU=255THENN
EXTJ

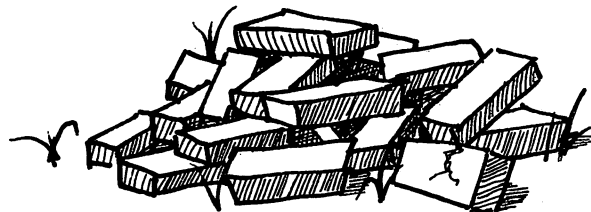
```

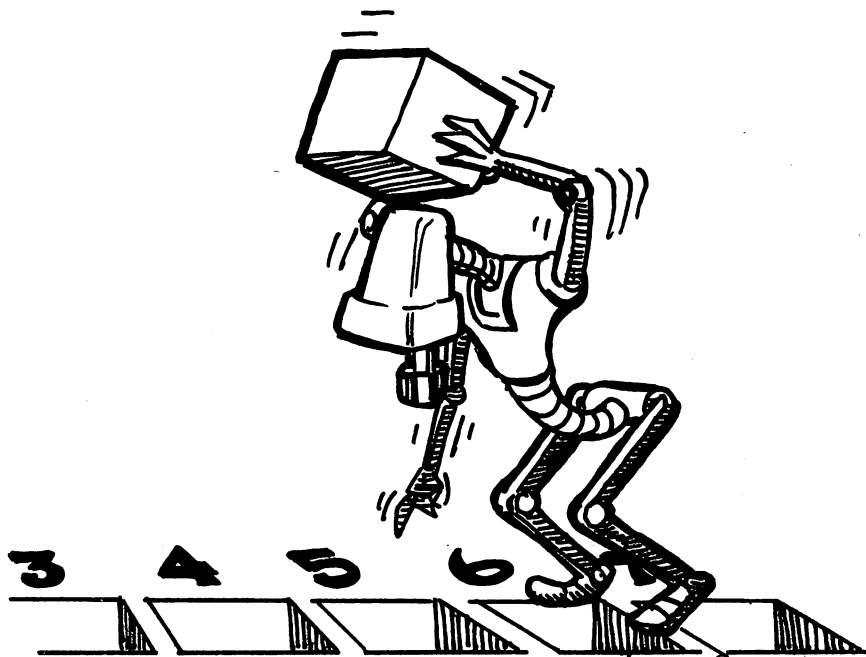


```

3020 BX=10+INT(20*RND(0)):BY=14:BD=1:BF=
0:BA=2+INT(4*RND(0)):PB=0
3030 GOTO2300:REM BALL
3040 GOSUB2200:REM PADDLE
3050 GOTO3030
3060 POKESID+1,250:POKESID+4,33:POKESID+
24,15:FORPA=1T030:NEXTPA
3061 POKESID+1,0:POKESID+4,32
3065 NEXTB:RETURN
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4005 PRINT"{HOME}{RVS OFF}{CYN}THE GAME
IS OVER !!!"
4020 PRINT"YOUR SCORE IS:"SC "(";
4030 IFSC<50THENPRINT"LOUSY)":RETURN
4031 IFSC<100THENPRINT"POOR)":RETURN
4032 IFSC<150THENPRINT"SO-SO)":RETURN
4033 IFSC<200THENPRINT"BLAH)":RETURN
4034 IFSC<225THENPRINT"GOOD)":RETURN
4035 IFSC<250THENPRINT"GREAT)":RETURN
4036 IFSC<275THENPRINT"EXCELLENT)":RETUR
N
4037 IFSC<300THENPRINT"FANTASTIC)":RETUR
N
4038 PRINT"PERFECT)":RETURN
4990 RETURN

```





Connect Five

This classic game requires that you connect five squares either vertically, horizontally, or diagonally. Though not a great challenge, the game is good for your ego. Lines 2000-2017 draw the gameboard. Line 2005 clears the screen, and sets the color to light grey (using the (COMMDR-KEY/8) key combination). If you don't like the color of the board, experiment with the (Commodore-Number) or (CTRL-Number) combinations until you find one you like. Note that the vertical bars are created by printing a series of uniformly spaced inverse spaces. Next, list lines 3000-3025. Although the game asks for a row number, it uses a string variable ANS\$ to collect the information. ANY variable that ends with a dollar sign (\$) is called a string variable, and no mathematical operations may be performed upon it. This is one of the simplest methods of error checking user input. If we had used the variable ANS instead, and somebody accidentally pressed a non-numeric key, we would get a ?REDO FROM START error, and our display would get messed up. By using the VAL function in line 3020, we convert any answer to its numeric equivalent. Line 3021 checks the numeric value to make sure it is between 1 and 7 (the number of columns). If it is not, then 3025 gives you a reminder of what type of input the computer is expecting and sends you back to line 3010 to retrieve it again. You might want to experiment with the VAL command to see how computers alphabetize lists of words.

The diamond can be produced with a SHIFT-Z.
The heart can be produced with a SHIFT-S.

```

10 REM *****
11 REM ***          ***
12 REM **  CONNECT FIVE  **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1010 PRINT"{CLR}"
1015 POKE53281,12:POKE53280,12
1020 PRINTTAB(10)"{BLU}*** CONNECT FIVE
***"
1030 PRINT:PRINT:PRINT
1035 PRINT"THE OBJECT OF THE GAME IS TO
GET FIVE OFYOUR PIECES IN A ROW, ";
1040 PRINT"EITHER VERTICALLY OR HORIZONT
ALLY."
1041 PRINT
1045 PRINT"WHEN IT IS YOUR TURN TO MOVE,
ENTER THE"
1046 PRINT"NUMBER OF THE COLUMN YOU WISH
TO DROP"
1047 PRINT"YOUR PIECE INTO."
1048 PRINT
1050 PRINT"AFTER YOU MOVE, I WILL TAKE A
TURN."
1055 PRINT"THE FIRST ONE TO CONNECT FIVE
IS THE  WINNER."
1060 PRINT:PRINT:PRINT:PRINT"PRESS RETUR
N WHEN READY TO CONTINUE"
1065 GETANS$:IFANS$<>CHR$(13)THEN1065
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2005 PRINT"{CLR}{GRY 3}";:POKE53281,14:P
OKE53280,14
2006 PRINTTAB(6)"{RVS ON}  1  2  3  4
5  6  7  "
2010 FORI=1TO7

```

```

2011 FORJ=1TO2
2012 PRINTTAB(6)"{RVS ON} {RVS OFF}  {R
VS ON} {RVS OFF}  {RVS ON} {RVS OFF}
{RVS ON} {RVS OFF}  {RVS ON} {RVS OFF}
  {RVS ON} {RVS OFF}  {RVS ON} {RVS OFF}
}  {RVS ON} {RVS OFF}"
2015 NEXT
2016 PRINTTAB(6)"{RVS ON}
      ":NEXT
2017 PRINT"{CSU}"TAB(6)"{RVS ON}  1  2
   3  4  5  6  7  "
2020 PLOT=49152:FORI=0TO24:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 P$(1)="***":P$(2)="***"
2040 C(1)=90:C(2)=83
2110 DEFFNCO(X)=PEEK(55296+FX+40*FY)AND1
5
2900 DEFFNX(X)=(X-1)*4+7
2905 DEFFNY(X)=(X-1)*3+1
2920 DEFFNSC(X)=PEEK(1024+X)
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 SYSPLT,0,22:FORI=1TO2:PRINT"
      ":NEXT
3011 P=1:SYSPLT,0,22:INPUT"{BLK}YOUR MOV
E (COLUMN 1-7) ";ANS$
3020 ANS=VAL(ANS$)
3021 IFANS>0ANDANS<BANDANS=INT(ANS)THEN3
030
3022 SYSPLT,0,23
3025 PRINT"PLEASE SELECT A NUMBER FROM 1
TO 7":FORA=1TO1000:NEXT:GOTO3010
3030 AX=FNX(ANS)
3040 IF FNSC(AX+40)=32THEN3050
3045 SYSPLT,0,23:PRINT"THAT COLUMN IS FU
LL...":FORA=1TO1000:NEXT:GOTO3010
3050 FORJ=1TO7:PRINTMID$("{YEL}{LT GRN}"
,P,1);
3051 JX=FNJ(J):FORK=JXTOJX+1:SYSPLT,AX,K
:PRINTP$(P);:NEXTK
3055 IFJ=7THEN3080
3060 LX=FNJ(J+1):IFFNSC(40*LX+AX)<>32THE

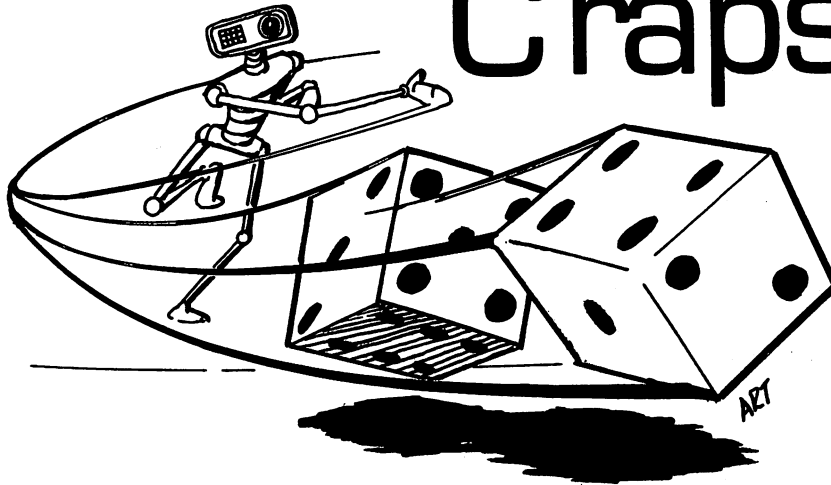
```

```

NJ=7:GOTO3080
3070 FORK=JXTOJX+1:SYSPLT,AX,K:PRINT"
";:NEXTK
3080 NEXTJ
3090 FORI=1TO7:FORJ=1TO3:FORK=JTOJ+4:IX=
FNX(I):KX=FNY(K)
3100 IFFNSC(IX+40*KX)<>C(P)THENK=J+4:NEX
TK:GOTO3110
3105 NEXTK:RETURN
3110 NEXT:NEXT
3120 FORJ=1TO7:FORI=1TO3:FORK=ITOI+4:KX=
FNX(K):JX=FNY(J)
3130 IFFNSC(KX+40*JX)<>C(P)THENK=I+4:NEX
TK:GOTO3140
3135 NEXTK:RETURN
3140 NEXT:NEXT
3150 IFP=2THENP=1:GOTO3010
3155 SYSPLT,0,22:FORI=1TO2:PRINT"
":NEXT
3156 SYSPLT,0,22:PRINT"(BLU)MY MOVE..."
3160 P=2:ANS=1+INT(7*RND(0)):AX=FNX(ANS)
:IFFNSC(40+AX)<>32THEN3160
3170 GOTO3050
4000 :
4001 :REM *** END
4002 :
4010 SYSPLT,0,22:FORI=1TO2:PRINT"
":NEXT
4011 SYSPLT,0,22:PRINT"(BLK)THE GAME IS
OVER !!!":PRINT"THE WINNER IS ...";
4020 IFP=1THENPRINT"YOU!!!{HOME}";:RETUR
N
4030 PRINT"ME !!!{HOME}";:RETURN
4990 RETURN

```

Craps



Craps is a somewhat simplified version of the popular dice game. You are given a \$1500 stake to play with. To stop playing before you run out of money, make a bet of 0 dollars. Let's look into the program. The section that will probably attract your attention first is lines 2340-2393. Each of the four line subroutines in this section of the program draws one of the six sides of a die. Lines 2400-2410 create the sound routine which simulates the turning of the dice. Line 2100 creates a random number for each of the dice (D1 and D2). Line 2330 uses an ON/GOTO structure again, going to the appropriate dice-rolling routine based on the value of the variable DD. Line 2320, by the way, determines what color the dice will be. It creates a string composed of all the possible colors, and then picks a random element of that string for the color value. This is a good one-line routine to use when you want to change the color of an object at random.

The circles for the dice can be produced with SHIFT-Q's.

```

10 REM *****
11 REM ***          ***
12 REM **          CRAPS          **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,12:POKE53280,12
1010 PRINT"{CLR}{BLK}"TAB(13)"*** CRAPS
***"
1030 PRINT:PRINT
1031 PRINT"THIS IS A DICE GAME CALLED **
CRAPS** ."
1035 PRINT
1040 PRINT"TO PLAY, YOU WAGER A PORTION
OF YOUR"
1041 PRINT"MONEY ON A ROLL OF THE DICE.
HERE ARE THE RULES..."
1045 PRINT
1050 PRINT"YOU WIN IF THE FIRST ROLL IS
EITHER 7 OR";
1051 PRINT"11. CONVERSELY, YOU LOSE IF
THE FIRST ROLL IS 2, 3, OR 12."
1060 PRINT
1065 PRINT"IF YOU GET A 4,5,6,8,9, OR 10
ON YOUR"
1066 PRINT"FIRST ROLL, IT IS REFERRED TO
AS YOUR"
1067 PRINT"*POINT*. YOU MUST CONTINUE R
OLLING"
1070 PRINT"UNTIL YOU: 1) ROLL A 7, WHICH
IS CALLED"
1071 PRINT"*CRAPPING OUT* WHEREBY YOU LO
SE, OR 2)"
1072 PRINT"YOU ROLL A NUMBER EQUAL TO YO
UR POINT...YOU WIN!"
1075 PRINT
1080 PRINT"TO QUIT THE GAME BET 0 DOLLAR
S."

```



```

1100 PRINT:PRINT"PRESS RETURN TO BEGIN P
LAY."
1110 GETANS$:IFANS$<>CHR$(13)THEN1110
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2003 POKE53281,0
2005 PRINT"{CLR}";
2010 PRINT"{BLU}{RVS ON}
";
2011 FORI=1TO20:PRINT"{RVS ON} {RVS OFF}
{R
VS ON} {RVS OFF}";:NEXT
2012 PRINT"{BLU}{RVS ON}
";
2015 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXTI
2016 POKESID+5,9
2020 PLOT=49152:FORI=0TO24:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 MNY=1500
2095 RETURN
2200 D1=1+INT(6*RND(0)):D2=1+INT(6*RND(0
))
2210 GOSUB2300:GOSUB2310
2220 GOSUB2300:GOSUB2310
2230 GETKEY$:IFKEY$=""THEN2200
2300 DD=D1:DX=10:GOTO2320
2310 DD=D2:DX=24
2320 PRINTMID$("{RED}{CYN}{PUR}{GRN}{BLU
}{YEL}{LT RED}{ORNG}{LT BLU}",1+INT(9*RN
D(0)),1);
2330 ONDDGOTO2340,2350,2360,2370,2380,23
90
2340 SYSPLOT,DX,10:PRINT"{RVS ON}      "
2341 SYSPLOT,DX,11:PRINT"{RVS ON}      #  "
2342 SYSPLOT,DX,12:PRINT"{RVS ON}      "
2343 GOTO2400
2350 SYSPLOT,DX,10:PRINT"{RVS ON}      #  "
2351 SYSPLOT,DX,11:PRINT"{RVS ON}      "
2352 SYSPLOT,DX,12:PRINT"{RVS ON}      #  "
2353 GOTO2400

```

```

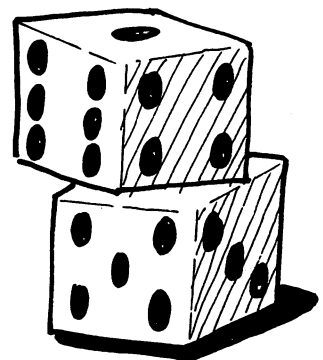
2360 SYSPLOT,DX,10:PRINT"{RVS ON}  #  "
2361 SYSPLOT,DX,11:PRINT"{RVS ON}  #  "
2362 SYSPLOT,DX,12:PRINT"{RVS ON}  #  "
2363 GOTO2400
2370 SYSPLOT,DX,10:PRINT"{RVS ON}  #  #  "
2371 SYSPLOT,DX,11:PRINT"{RVS ON}  #  #  "
2372 SYSPLOT,DX,12:PRINT"{RVS ON}  #  #  "
2373 GOTO2400
2380 SYSPLOT,DX,10:PRINT"{RVS ON}  #  #  "
2381 SYSPLOT,DX,11:PRINT"{RVS ON}  #  #  "
2382 SYSPLOT,DX,12:PRINT"{RVS ON}  #  #  "
2383 GOTO2400
2390 SYSPLOT,DX,10:PRINT"{RVS ON}  #  #  "
2391 SYSPLOT,DX,11:PRINT"{RVS ON}  #  #  "
2392 SYSPLOT,DX,12:PRINT"{RVS ON}  #  #  "
2393 GOTO2400
2400 POKE SID+1,90:POKESID+4,33:POKESID+
24,10:FOR PA=1 TO 80:NEXT
2410 POKE SID+4,32:POKE SID+1,0
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 GOSUB3300:Y=4:GOSUB3305:Y=6:GOSUB33
05
3015 SYSPLOT,3,3:PRINT"{RVS OFF}{GRN}YOU
HAVE"MNY"{CSL} DOLLARS  "
3020 SYSPLOT,2,23:INPUT "{GRY 3}HOW MUCH
WILL YOU BET";ANS$
3021 ANS=INT(VAL(ANS$))
3022 IFANS<0ORANS>MNYTHEN3010
3023 IFANS=0THENRETURN
3025 SYSPLOT,3,4:PRINT"{RVS OFF}{GRY 3}B
ET:"ANS"{CSL}  "
3030 SYSPLOT,2,23:PRINT"(ROLLING...) <PR
ESS ANY KEY TO STOP>";
3040 GOSUB2200:T=D1+D2
3045 GOSUB3300
3050 SYSPLOT,3,6:PRINT"{CYN}ROLLED:"T"{C
SL}  "
3055 IFT>3ANDT<12THEN3060
3056 SYSPLOT,3,7:PRINT"{CYN}YOU CRAPPED
OUT...":ANS=-ANS
3057 FORPA=1TO1500:NEXTPA:GOTO3200
3060 IFT<>7ANDT<>11THEN3065
3061 SYSPLOT,3,7:PRINT"{CYN}YOU WON THAT
TOSS..."
3062 FORPA=1TO1500:NEXTPA:GOTO3200
3065 SYSPLOT,3,7:PRINT"{CYN}YOUR POINT I
S"T"{CSL}  ":PT=T
3066 FORPA=1TO1200:NEXTPA

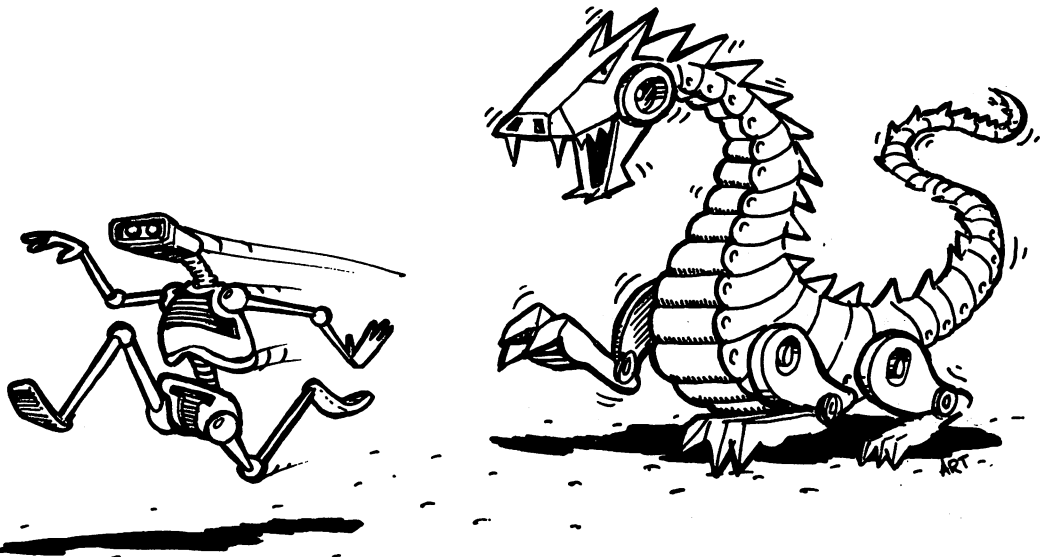
```

```

3070 SYSPLOT,3,4:PRINT"{GRY 3}BET:"ANS"{
CSL} POINT:"PT"{CSL} "
3072 Y=6:GOSUB3305
3073 SYSPLOT,2,23:PRINT"(ROLLING...) <PR
ESS ANY KEY TO STOP>";
3074 GOSUB2200:T=D1+D2
3075 SYSPLOT,3,6:PRINT"{CYN}ROLLED: {CSL
} "T"{CSL} "
3080 IFT<>7THEN3085
3081 SYSPLOT,3,7:PRINT"{CYN}YOU CRAPPED
OUT ..."
3082 FORPA=1TO1500:NEXTPA:GOTO3100
3085 IFT<>PTTHEN3090
3086 SYSPLOT,3,7:PRINT"{CYN}YOU GOT YOUR
POINT !!!"
3087 FORPA=1TO1500:NEXTPA:GOTO3200
3090 SYSPLOT,3,7:PRINT"{CYN}YOU MUST ROL
L AGAIN..."
3091 FORPA=1TO1500:NEXTPA
3092 GOSUB3300:Y=6:GOSUB3305:GOTO3072
3100 MNY=MNY-ANS:IFMNY<=0THENRETURN
3110 GOTO3010
3200 MNY=MNY+ANS:GOTO3010
3300 Y=23
3305 FORK=0TO1:SYSPLOT,1,Y+K:PRINT"{RVS
OFF}
";
3310 NEXT:RETURN
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4005 PRINT"{CLR}{GRY 3}":SYSPLOT,0,10
4020 PRINT"{GRY 3}YOU STOPPED WITH"MNY"D
OLLARS"
4040 IFMNY<100THENPRINT"BETTER STICK TO
THE SLOT MACHINES!":RETURN
4041 IFMNY<500THENPRINT"NOT SO GOOD":RET
URN
4042 IFMNY<1500THENPRINT"NOT BAD AT ALL"
:RETURN
4043 PRINT"THAT'S GREAT! ";
4044 IFMNY>6000THENPRINT"WOW, WHATTA A R
OLL!!!":RETURN
4990 RETURN

```





Dragon's Lair

In this game, a maze is constructed before your eyes but becomes invisible when completed. You must avoid the man-eating dragon and escape the maze — otherwise you will attend the dragon's dinner as the main course! Only a good memory will save you from this terrible fate. Lines 2045-2057 draw the maze the first time around with all the walls in. The rest of the setup section (lines 2062-2999) takes care of knocking out the appropriate walls to create a trail through the maze. As usual, there are a series of IF statements (lines 3015-3033) to test the keyboard. There is one unusual thing about Dragon's Lair, however. You are given a choice of playing in normal or real time modes. Normal time allows you to make one move every time the dragon moves. But if you look at line 3015, you'll see what happens if you play real time. The program branches to the subroutine at line 3700, which plots the dragon's move without waiting for you to make a move! Try it both ways, and see a tough game get even tougher.

The diamond can be produced with a SHIFT-Z.
The circle can be produced with a SHIFT-Q.

```

10 REM *****
11 REM ***          ***
12 REM **  DRAGON'S LAIR  **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53280,14:POKE53281,14
1010 PRINT"{CLR}{BLU}"TAB(10)"*** DRAGON
'S LAIR ***"
1020 PRINT:PRINT
1030 PRINT"YOU WILL BE PLACED IN A MAZE
WITH A MAN-EATING DRAGON."
1040 PRINT:PRINT"YOUR PROBLEM IS TO ESCA
PE FROM THE MAZE"
1041 PRINT"BEFORE THE DRAGON MAKE A MEAL
OF YOU."
1050 PRINT
1051 PRINT"YOU MOVE BY PRESSING: ":PRINT
1052 PRINT"      UP -      'I'"
1053 PRINT"      DOWN -      'M'"
1054 PRINT"      RIGHT -      'K'"
1055 PRINT"      LEFT -      'J'"
1060 PRINT:PRINT:PRINT"PRESS RETURN TO C
ONTINUE."
1065 GETANS$:IFANS$<>CHR$(13)THEN1065
1070 PRINT"{CLR}{BLU}"TAB(10)"*** DRAGON
'S LAIR ***"
1075 PRINT:PRINT
1080 PRINT"YOU MAY PLAY IN ONE OF THE FO
LLOWING TWOMODES:"
1081 PRINT:PRINT"      NORMAL":PRINT"WHERE
THE DRAGON MOVES ONLY AFTER YOU"
1082 PRINT"MAKE A MOVE..."
1090 PRINT:PRINT"      REAL-TIME"
1091 PRINT"WHERE THE DRAGON'S PURSUIT IS
CONTINUAL,";

```

```

1092 PRINT"WHETHER YOU MOVE OR NOT !"
1100 PRINT:PRINT"    WILL YOU PLAY:";PRIN
T
1101 PRINT"N)ORMAL          -OR-          R)EAL-
TIME"
1105 PRINT:INPUT"WHAT IS YOUR CHOICE (N/
R) ";ANS$:ANS$=LEFT$(ANS$,1)
1110 IFANS$<>"N"ANDASN$<>"R"THEN1105
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2005 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXT:POKESID+5,9:POKESID+6,0
2010 DIMMA(19,11,4)
2015 DEFFNR(X)=1+INT(X*RND(0))
2020 PLOT=49152:FORI=0TO24:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 RX=FNR(19):RY=FNR(11)
2040 PRINT"{CLR}";
2045 PRINT"{RVS ON}{WHT}
"

2050 FORI=1TO11
2055 PRINT"{RVS ON}{RVS OFF}{RVS ON}{
RVS OFF}{RVS ON}{RVS OFF}{RVS ON}{RV
S OFF}{RVS ON}{RVS OFF}{RVS ON}{RVS
OFF}{RVS ON}{RVS OFF}{RVS ON}{RVS OF
F}{RVS ON}{RVS OFF}{RVS ON}{RVS OFF}
";
2056 PRINT"{RVS ON}{RVS OFF}{RVS ON}{
RVS OFF}{RVS ON}{RVS OFF}{RVS ON}{RV
S OFF}{RVS ON}{RVS OFF}{RVS ON}{RVS
OFF}{RVS ON}{RVS OFF}{RVS ON}{RVS OF
F}{RVS ON}{RVS OFF}{RVS ON} "
2057 PRINT"{RVS ON}
"

2060 NEXT
2061 PRINT"{BLK}    <<< I'M NOW BUILDING
THE LAIR >>>";
2062 CNT=1
2065 C=0:R=1:L=1:U=1:D=1:IFRX<>19THENR=-
(MA(RX+1,RY,0)>0)
2066 IFRX<>1THENL=- (MA(RX-1,RY,0)>0)
2067 IFRY<>11THEND=- (MA(RX,RY+1,0)>0)

```

```

2068 IFRY<>1THENU=- (MA(RX,RY-1,0)>0)
2069 C=L+R+U+D: IFC=4OR(C>2ANDFNR(10)<3)T
HEN2100
2070 ONFNR(4)GOTO2075,2080,2085,2090
2075 IFRTHEN2070
2078 SYSPLOT,RX+RX,RY+RY-1:PRINT" {RVS OF
F} " ; :MA(RX,RY,0)=MA(RX,RY,0)+1
2079 MA(RX,RY,1)=1:RX=RX+1:MA(RX,RY,3)=1
:MA(RX,RY,0)=MA(RX,RY,0)+1:GOTO2095
2080 IFDTHEN2070
2083 SYSPLOT,RX+RX-1,RY+RY:PRINT" {RVS OF
F} " ; :MA(RX,RY,0)=MA(RX,RY,0)+1
2084 MA(RX,RY,2)=1:RY=RY+1:MA(RX,RY,4)=1
:MA(RX,RY,0)=MA(RX,RY,0)+1:GOTO2095
2085 IFLTHEN2070
2088 SYSPLOT,RX+RX-2,RY+RY-1:PRINT" {RVS
OFF} " ; :MA(RX,RY,0)=MA(RX,RY,0)+1
2089 MA(RX,RY,3)=1:RX=RX-1:MA(RX,RY,1)=1
:MA(RX,RY,0)=MA(RX,RY,0)+1:GOTO2095
2090 IFUTHEN2070
2093 SYSPLOT,RX+RX-1,RY+RY-2:PRINT" {RVS
OFF} " ; :MA(RX,RY,0)=MA(RX,RY,0)+1
2094 MA(RX,RY,4)=1:RY=RY-1:MA(RX,RY,2)=1
:MA(RX,RY,0)=MA(RX,RY,0)+1
2095 IFMA(RX,RY,0)=1THENCNT=cnt+1:IFCNT=
209THEN2200
2097 GOTO2065
2100 RX=FNR(19):RY=FNR(11)
2101 IFMA(RX,RY,0)<>0THEN2065
2102 RX=RX+1:IFRX>19THENRX=1:RY=RY+1:IFR
Y>11THENRY=1
2105 GOTO2101
2200 PRINT" {CLR} {WHT} " ;
2204 PRINT" {RVS ON}
"

2205 FORI=1TO10
2206 PRINT" {RVS ON} {RVS OFF}
{RVS ON} "
2207 PRINT" {RVS ON} {RVS OFF}
{RVS ON} ":NEX
T
2208 PRINT" {RVS ON} {RVS OFF}
{RVS ON} "
2209 PRINT" {RVS ON}
"

2210 PX=1:PY=FNR(11):SYSPLOT,PX+PX-1,PY+
PY-1:PRINT" {RVS ON} {BLU} # " ;
2215 WY=FNR(11):SYSPLOT,38,WY+WY-1:PRINT
" {RVS OFF} " ;
2220 MX=19:MY=WY:SYSPLOT,37,MY+MY-1:PRIN

```



```

T" {BRWN} *";
2320 FORI=1TO19:FORJ=1TO11:MA(I,J,0)=0:N
EXT:NEXT
2400 SYSPLIT,2,24:PRINT" {RVS ON} {BLK} J=
LEFT {LT GRN} K= RIGHT {BLU} I= UP {YEL
} M= DOWN ";
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 GETKEY$
3015 IFANS$="R"ANDKEY$=""THEN3700
3020 IFKEY$=""THEN3010
3030 IFKEY$="K"THEN3100
3031 IFKEY$="M"THEN3200
3032 IFKEY$="J"THEN3300
3033 IFKEY$="I"THEN3400
3040 GOTO3450
3100 IFMA(PX,PY,1) THENX2=PX+1:Y2=PY:GOTO
3500
3110 FORK=-2TO0:SYSPLIT,PX+PX,PY+PY+K:PR
INT" {RVS ON} {WHT} ":NEXT:GOTO3450
3200 IFMA(PX,PY,2) THENX2=PX:Y2=PY+1:GOTO
3500
3210 FORK=-2TO0:SYSPLIT,PX+PX+K,PY+PY:PR
INT" {RVS ON} {WHT} ":NEXT:GOTO3450
3300 IFMA(PX,PY,3) THENX2=PX-1:Y2=PY:GOTO
3500
3310 FORK=-2TO0:SYSPLIT,PX+PX-2,PY+PY+K:
PRINT" {RVS ON} {WHT} ":NEXT:GOTO3450
3400 IFMA(PX,PY,4) THENX2=PX:Y2=PY-1:GOTO
3500
3410 FORK=-2TO0:SYSPLIT,PX+PX+K,PY+PY-2:
PRINT" {RVS ON} {WHT} ":NEXT
3450 POKESID+24,15:POKESID+1,20:POKESID+
4,33:FORPA=1TO100:NEXT
3455 POKESID+4,32:POKESID+1,0:POKESID+24
,0:GOTO3700
3500 SYSPLIT,PX+PX-1,PY+PY-1:PRINT" {RVS
OFF} ";;PX=X2:PY=Y2
3510 SYSPLIT,PX+PX-1,PY+PY-1:PRINT" {RVS
OFF} {BLU} *";:
3700 IFPX=19ANDPY=WYTHENWIN=1:RETURN
3705 IFPX=MXANDPY=MYTHENSYSPLIT,MX+MX-1,
MY+MY-1:PRINT" {RVS OFF} {RED} *";:WIN=0:RE
TURN
3709 ONFNR(4)GOTO3710,3711,3712,3713
3710 IFMX<PXTHEN3720
3711 IFMY<PYTHEN3730
3712 IFMX>PXTHEN3740

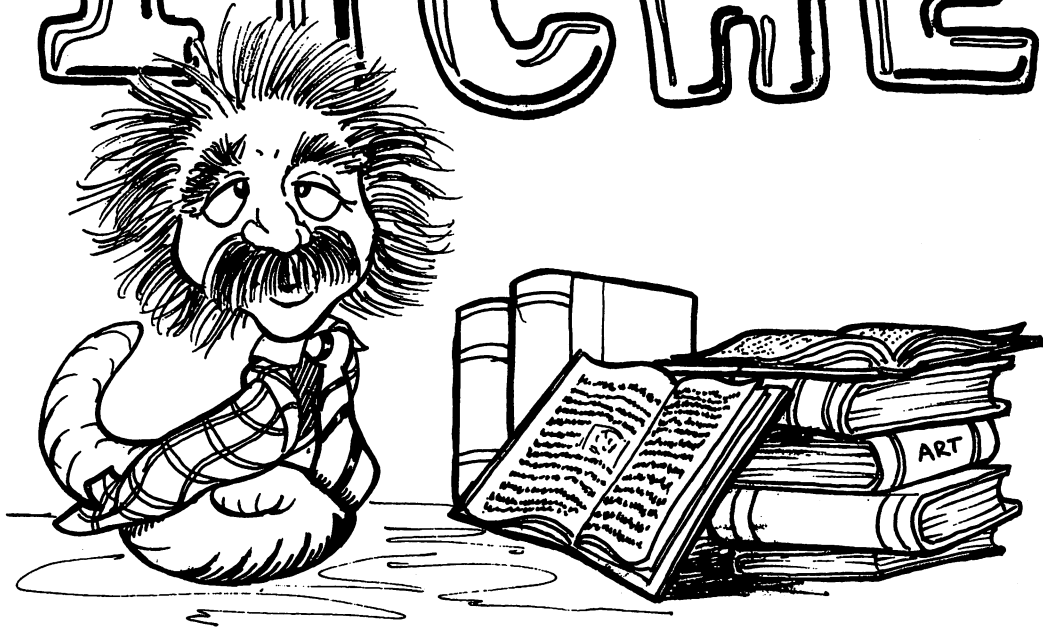
```

```

3713 IFMY>PYTHEN3750
3714 GOTO3710:REM SHOULDN'T GET HERE
3720 IFMX=19THEN3730
3722 IFMA(MX,MY,0)>5THENMA(MX,MY,0)=0:GO
T03726
3724 IFMA(MX,MY,1)=0THEN3730
3726 X2=MX+1:Y2=MY:GOTO3800
3730 IFMY=11THEN3740
3732 IFMA(MX,MY,0)>5THENMA(MX,MY,0)=0:GO
T03736
3734 IFMA(MX,MY,2)=0THEN3740
3736 X2=MX:Y2=MY+1:GOTO3800
3740 IFMX=1THEN3750
3742 IFMA(MX,MY,0)>5THENMA(MX,MY,0)=0:GO
T03746
3744 IFMA(MX,MY,3)=0THEN3750
3746 X2=MX-1:Y2=MY:GOTO3800
3750 IFMY=1THEN3720
3752 IFMA(MX,MY,0)>5THENMA(MX,MY,0)=0:GO
T03756
3754 IFMA(MX,MY,4)=0THEN3720
3756 X2=MX:Y2=MY-1
3800 SYSLOT,MX+MX-1,MY+MY-1:PRINT" (RVS
OFF) ";:MX=X2:MY=Y2
3805 SYSLOT,MX+MX-1,MY+MY-1:PRINT" (RVS
ON) (RED) *";
3810 MA(MX,MY,0)=MA(MX,MY,0)+1
3815 IFPX=19ANDPY=WYTHENWIN=1:RETURN
3820 IFPX=MXANDPY=MYTHENWIN=0:RETURN
3830 GOTO3010
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4005 FORPA=1T01000:NEXT
4010 IF(WIN)THEN4100
4020 PRINT"{CLR}{RVS OFF}{WHT}SORRY, BUT
THE DRAGON GOT YOU...."
4095 RETURN
4100 PRINT"{CLR}{RVS OFF}{WHT}***** YOU
WON ***** (BUT NOW THE DRAGONIS EVEN HU
NGRIER!)"
4990 RETURN

```

ITCHE



In this game, you will build a maze for the maze-loving worm, Itche. Once you have completed the maze, you may instruct Itche to try and solve it. Though not an interactive game, it is an ingenious utilization of graphics. The perimeter of the board is drawn in lines 2006-2008, and the prompts at the bottom of the page are drawn at 2016-2017. Line 2020 waits for the input to be entered. If you choose "P" for example, line 2035 sends control of the program to the routine at line 2200 where the plotting color is set to blue. From there it goes to 2700, and waits for the direction in which plotting is to take place. Choosing "R" for example, would send the program to the routine at line 2750, where Itche's new position (X2) is increased by one over his old position (IX+1). (Note how there is one subroutine for each direction.) Control is then sent to the routine 2790 (if Itche hasn't reached the right border) and is finally returned to line 2700 to check for further movement in the same mode. Can you see what line 2745 does? (Hint: take it out and try to change from PLOT to MOVE or from MOVE to ERASE, or the like.)

The diamond shape can be produced by a SHIFT-Z.

```

10 REM *****
11 REM ***          ***
12 REM **          ITCHE          **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,12:POKE53280,12
1010 PRINT"{CLR}{BLU}"TAB(13)"*** ITCHE
***"
1020 PRINT
1030 PRINT"THIS IS THE GAME OF ITCHE. T
HE ITCHE"
1031 PRINT"IS A SPECIAL WORM WHO LIKES T
O SOLVE"
1032 PRINT"MAZES. YO'J ARE TO CREATE A M
AZE FOR"
1033 PRINT"THE ITCHE TO SOLVE."
1040 PRINT:PRINT"BY USING THE FOLLOWING
INSTRUCTIONS, YOU";
1041 PRINT"WILL BUILD THE MAZE WALLS. U
PON COMMAND";
1042 PRINT"(G), ITCHE WILL WIND HIS WAY
THROUGH THE MAZE."
1050 PRINT:PRINT"G)O TO HAVE THE ITC
HE FIND THE EXIT.";
1051 PRINT "X)IT TO EXIT THE GAME."
1052 PRINT "C)LEAR TO START THE MAZE O
VER."
1053 PRINT "E)RASE TO ERASE A WALL.."
1054 PRINT:PRINT"FOLLOWED BY THESE..."
1055 PRINT
1060 PRINT"M) FOR MOVE U
=UP"
1061 PRINT"P) FOR PLOT R=RIGHT
L=LEFT"
1062 PRINT" D=
DOWN"
1070 PRINT:PRINT"PRESS RETURN TO BEGIN P

```

```

LAY!"
1075 GETANS$: IFANS$<>CHR$(13) THEN1075
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2003 POKE53281,0:POKE53280,12:POKE650,12
B
2004 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXT:POKESID+5,9
2005 PRINT"{CLR}{BLU}{RVS ON}
                                     {RVS OFF}";
2006 PRINT"{BLK}
                                     ";
2007 FORI=1TO20:PRINT"{RVS ON}{BLU} {RVS
OFF}{BLK}
                                     {RVS ON}{BLU} ";;NEXT
2008 PRINT"{RVS ON}
                                     ";
2009 RESTORE:PLOT=49152:FORI=0TO24:READC
:POKEPLT+I,C:NEXTI
2010 DATA32,16,192,132
2011 DATA253,32,16,192
2012 DATA152,170,164,253
2013 DATA24,76,240,255
2014 DATA32,253,174,32
2015 DATA158,173,76,170,177
2016 PRINT"{WHT}{RVS ON}G)O      C)LEAR
X)IT      U)P      D)OWN ";
2017 PRINT"{RVS ON}P)LOT      E)RASE      M)OV
E => L)EFT      R)IGHT{HOME}";
2018 POKE1024+999,160:POKE55296+999,1
2019 C0$="{BLK}{WHT}{RED}{CYN}{PUR}{GRN}
{BLU}{YEL}{ORNG}{BRWN}{LT RED}{GRY 1}{GR
Y 2}{LT GRN}{LT BLU}{GRY 3}":IX=2:IY=1:S
YSLOT,IX,IY:PRINT"{RVS ON}{CYN} ";;CL$=
"":C2$="{RVS ON}{BLK}"
2020 GETANS$: IFANS$=""THEN2020
2030 IFANS$="G"THEN2100
2035 IFANS$="P"THEN2200
2040 IFANS$="M"THEN2300
2045 IFANS$="E"THEN2400
2050 IFANS$="C"THEN2500
2055 IFANS$="X"THEN2600
2060 FR=10:PA=5:GOSUB6000:GOTO2020
2100 REM *** GO ROUTINE
2110 RETURN
2200 REM *** PLOT ROUTINE
2210 CL$="{BLU}":GOTO2700
2300 REM *** MOVE ROUTINE

```

```

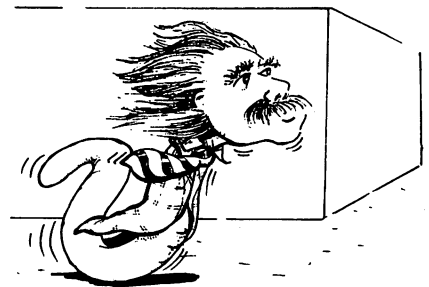
2310 CL$="":GOTO2700
2400 REM *** ERASE ROUTINE
2410 CL$="{BLK}":GOTO2700
2500 REM *** CLEAR ROUTINE
2510 GOTO2000
2600 REM *** EXIT ROUTINE
2610 GOSUB5000:PRINT"DO YOU WISH TO QUIT
?"
2611 GETANS$:IFANS$=""THEN2611
2620 IFLEFT$(ANS$,1)="Y"THENGOSUB4020:EN
D
2625 GOSUB5000:GOTO2016
2700 REM *** DIRECTIONS
2710 GETANS$:IFANS$=""THEN2710
2711 X2=IX:Y2=IY
2715 IFANS$="R"THEN2750
2720 IFANS$="D"THEN2760
2730 IFANS$="L"THEN2770
2740 IFANS$="U"THEN2780
2745 GOTO2030
2750 REM *** RIGHT
2752 X2=IX+1:IFX2<39THEN2790
2754 X2=38:GOTO2785
2760 REM *** DOWN
2762 Y2=IY+1:IFY2<22THEN2790
2764 Y2=21:GOTO2785
2770 REM *** LEFT
2772 X2=IX-1:IFX2>0THEN2790
2774 X2=1:GOTO2785
2780 REM *** UP
2782 Y2=IY-1:IFY2>0THEN2790
2784 Y2=1
2785 FR=200:PA=15:GOSUB6000
2790 PRINTCL$;:IFCL$=""THENPRINTC2$;
2791 SYSLOT,IX,IY:PRINT"{RVS ON} ";
2792 C2$=MID$(C0$,1+(PEEK(55296+X2+40*Y2
)AND15),1)
2795 SYSLOT,X2,Y2:PRINT"{CYN}{RVS ON} "
;:IX=X2:IY=Y2:GOTO2700
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3005 GOSUB5000:PRINT" <<< ITCHE IS NOW S
OLVING THE MAZE >>>"
3010 X2=1:Y2=1:SYSLOT,X2,Y2:PRINT"{GRN}
*";
3015 PRINTCL$;:IFCL$=""THENPRINTC2$;
3016 SYSLOT,IX,IY:PRINT"{RVS ON} ";
3020 X3=0:Y3=1:DIR=2

```

```

3030 XX=X2+X3:YY=Y2+Y3:IFXX=0ANDYY=1THEN
RETURN
3035 IFXX=39ANDYY=1THENRETURN
3040 IFXX<10RXX>38ORYY<10RY>21THEN3045
3041 CO=PEEK(55296+XX+40*YY)AND15:IFCO=0
THEN3100
3045 DI=DI-1:IFDI<1THENDI=4
3050 IFDI=1THENX3=1:Y3=0:GOTO3030
3055 IFDI=2THENX3=0:Y3=1:GOTO3030
3060 IFDI=3THENX3=-1:Y3=0:GOTO3030
3065 IFDI=4THENX3=0:Y3=-1:GOTO3030
3100 SYSPLOT,X2,Y2:PRINT"{RVS OFF}{BLK}
";X2=X2+X3:Y2=Y2+Y3:SYSPLOT,X2,Y2:PRINT
"{GRN}{RVS OFF}*";
3105 CL=CL+1:DI=DI+1:IFDI>4THENDI=1
3110 FR=25:PA=3:GOSUB6000:GOTO3050
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4005 SYSPLOT,X2,Y2:PRINT"{RVS ON}{BLK} "
";X2=X2+X3:Y2=Y2+Y3
4010 FORK=1TO10:SYSPLOT,X2,Y2:PRINT"{GRN
}{RVS OFF}*";
4011 FR=20:PA=20:GOSUB6000
4012 SYSPLOT,X2,Y2:PRINT"{BLK} ";
4013 FR=40:PA=20:GOSUB6000
4015 NEXT
4020 GOSUB5000
4025 IFLEFT$(ANS$,1)<>"Y"THEN4030
4026 PRINT"{WHT}{RVS ON} << ITCHE SAYS Y
OU SPOILED HIS FUN ! >>{HOME}";:RETURN
4030 IFX2<>39THEN4040
4031 PRINT"{WHT}{RVS ON} <<< ITCHE HAS
SOLVED THE MAZE >>>"
4032 PRINT"{RVS ON} HE DID IT IN"C
L"CLICKS...{HOME}";:RETURN
4040 PRINT"{WHT}{RVS ON} <<< ITCHE CANN
OT SOLVE YOUR MAZE >>>"
4041 PRINT"{RVS ON} HE IS STUCK AT
THE BEGINNING..{HOME}":RETURN
5000 REM CLEAR BOTTOM TWO LINES
5005 SYSPLOT,0,23:PRINT"{WHT}{RVS ON}
"
5010 PRINT"{RVS ON}
";
5020 SYSPLOT,0,23:RETURN
6000 REM PLAY A NOTE
6005 POKESID+1,FR:POKESID+24,15:POKESID+
4,33

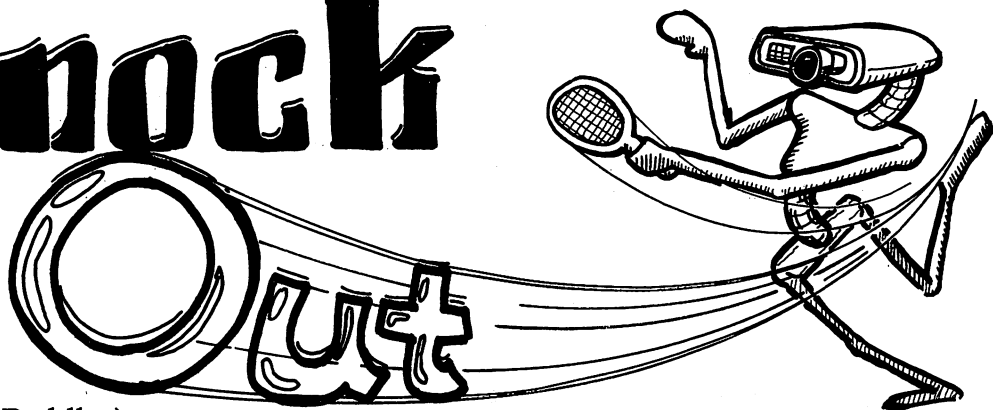
```



```
6010 FORI=1TO10*PA:NEXT I
6020 POKESID+1,0:POKESID+4,32:POKESID+24
,0:RETURN
```



Knock



(Requires Paddles)

This game is similar to the video dinosaur, Pong. Both you and your opponent start with five men. Using paddles, (joysticks won't work) try to maneuver your paddle to deflect the ball into your opponent's men while, at the same time, protecting your own. The winner is the first player to eliminate all of his opponent's men. List lines 2210-2252. These lines, as indicated by the REM statement in 2210, control the movement of the two paddles. Line 2050 draws the five men. Try changing the step value in this line to see how it is set up. Use a value of 5 for the STEP increment to see how many men are produced. The ball moving routine and the various collision routines make up lines 3000-3150. As usual, the best way to see what each line does is to remove it and see how the program is affected.

The vertical bar figure can be produced with a CMDR-K.
The circle can be produced by a SHIFT-Q.

```

10 REM *****
11 REM ***          ***
12 REM **   KNOCK OUT   **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,12:POKE53280,14
1010 PRINT"{CLR}{BLK}{CSD}"TAB(11)"*** K
NOCK OUT ***":PRINT:PRINT
1030 PRINT"*KNOCK OUT* IS A TWO PLAYER G
AME THAT IS";
1031 PRINT"SIMILAR TO PONG.  EACH PLAYER
HAS A"
1032 PRINT"PADDLE WITH WHICH HE TRIES TO
RETURN THE";
1033 PRINT"BALL TO HIS OPPONENT'S COURT.
"
1040 PRINT
1050 PRINT"THE OBJECT OF THE GAME IS TO
KO (KNOCK"
1051 PRINT"OUT) YOUR OPPONENT'S MEN WHIL
E DEFENDING";
1052 PRINT"YOUR OWN. "
1055 PRINT
1060 PRINT"THE FIRST PLAYER TO KNOCK OUT
ALL OF THE";
1061 PRINT"OPPOSITION'S MEN IS THE WINNE
R. "
1070 PRINT:PRINT:PRINT"PRESS RETURN WHEN
READY TO CONTINUE. "
1080 GETANS$:IFANS$<>CHR$(13)THEN1080
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2010 SID=54272:FORI=SIDTOSID+24:POKEI,0:

```

```

NEXT
2015 POKESID+24,15
2020 PLOT=49664:FORI=0T024:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,194,132
2022 DATA253,32,16,194
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2040 PRINT"{CLR}{GRN}";
2050 FORI=4T020STEP3:SYSPLOT,3,I:PRINT"{
BLU}{RVS ON} ";;SYSPLOT,36,I:PRINT"{RVS
ON}{YEL} ";
2055 SYSPLOT,3,I+1:PRINT"{BLU}{RVS ON} "
;;SYSPLOT,36,I+1:PRINT"{RVS ON}{YEL} ";;
NEXT
2130 PDL=49152:FORI=0T063:READC:POKEPDL+
I,C:NEXTI
2131 DATA162,1,120,173,2,220,141,0,193,1
69,192,141,2,220,169
2132 DATA128,141,0,220,160,128,234,136,1
6,252,173,25,212,157
2133 DATA1,193,173,26,212,157,3,193,173,
0,220,9,128,141,5,193
2134 DATA169,64,202,16,222,173,0,193,141
,2,220,173,1,220,141
2135 DATA6,193,88,96
2140 SYSPLOT,0,24:PRINT"{RVS ON}{LT BLU}
"
;
2145 POKE2023,160:POKE56295,14
2146 SYSPLOT,0,0:PRINT"{RVS ON}{LT BLU}
"
;
2150 O0=1:O1=1
2170 FORI=1T0100:GOSUB2200:NEXT
2180 SF=1
2190 RETURN
2200 :
2210 :REM *** PADDLE CONTROL
2220 :
2230 SYSPDL:P0=1+INT(10*PEEK(PDL+257)/11
7):P1=1+INT(10*PEEK(PDL+259)/117)
2240 IFP0=0THEN2250
2241 SYSPLOT,14,O0:PRINT"{RVS OFF} ";;SY
SPLOT,14,O0+1:PRINT" ";
2242 O0=P0:SYSPLOT,14,O0:PRINT"{YEL}# ";;
SYSPLOT,14,O0+1:PRINT"{YEL}# ";;
2250 IFP1=01THEN2260

```

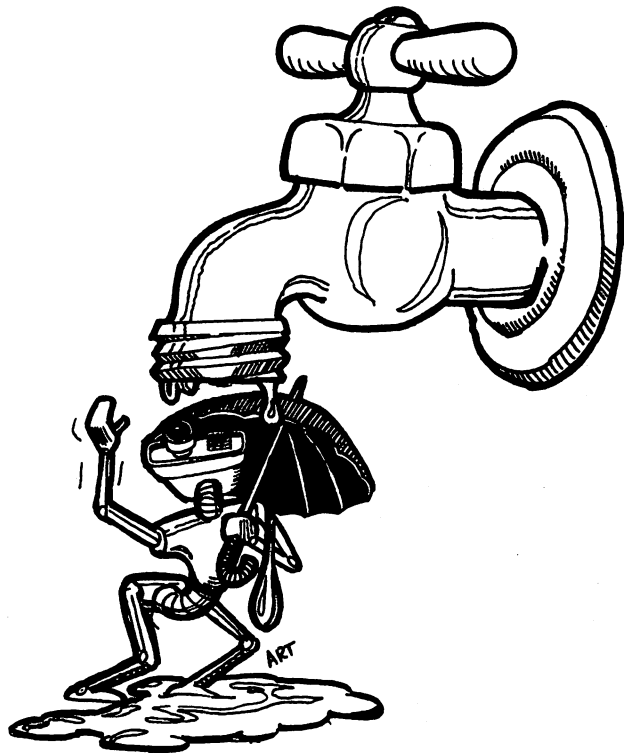
```

2251 SYSPLOT,25,01:PRINT" {RVS OFF} " ;:SY
SPLOT,25,01+1:PRINT" ";
2252 01=P1:SYSPLOT,25,01:PRINT" {BLU} {RVS
ON} " ;:SYSPLOT,25,01+1:PRINT" {BLU} {RVS
ON} " ;
2260 RETURN
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3060 L1=6:L2=6
3070 BX=16+INT(10*RND(0)):BY=5+INT(15*RN
D(0)):BD=-1+2*INT(2*RND(0))
3071 BS=INT(5*RND(0))-2:IFBS=0THEN3071
3075 SYSPLOT,BX,BY:PRINT" {WHT} {RVS OFF} "
";
3080 X2=BX+BD:Y2=BY+BS:GOSUB2200
3085 IFPEEK(1024+BX+40*BY)=81THEN3090
3086 POKESID+1,60:POKESID+4,32:FORPA=1TO
20:NEXT:POKESID+4,33:POKESID+1,0
3087 IFBY<10RBY>23THENBY=BY+BS:BS=-BS
3088 BX=BX-BD:BY=BY-BD:BD=-BD:BS=-BS:SYS
PLOT,BX,BY:PRINT" {WHT} {RVS OFF} " ;
3089 GOTO3080
3090 IFX2>=0ANDX2<=39THEN3100
3091 BD=-BD
3092 POKESID+1,90:POKESID+4,32:FORPA=1TO
20:NEXT:POKESID+4,33:POKESID+1,0
3095 GOTO3080
3100 IFY2>=1ANDY2<=23THEN3110
3101 BS=-BS
3102 POKESID+1,90:POKESID+4,32:FORPA=1TO
20:NEXT:POKESID+4,33:POKESID+1,0
3105 GOTO3080
3110 SC=PEEK(1024+X2+40*Y2):IFSC=32THEN3
150
3111 IFSC<>160THEN3125
3112 YY=1+3*INT(Y2/3)
3113 SYSPLOT,X2,YY:PRINT" {RVS OFF} " ;:SY
SPLOT,X2,YY+1:PRINT" ";
3114 POKESID+1,40:POKESID+4,32:FORPA=1TO
20:NEXT:POKESID+4,33:POKESID+1,0
3115 IFX2=3THENL1=L1-1
3116 IFX2=36THENL2=L2-1
3117 IFL1=0ORL2=0THEN3500
3120 GOTO3150
3125 POKESID+1,60:POKESID+4,32:FORPA=1TO
20:NEXT:POKESID+4,33:POKESID+1,0
3130 IFX2<>25THENBD=-BD:BS=(Y2-00)+(Y2-0
0<1)*2:GOTO3080

```

```
3140 BD=-BD:BS=(Y2-01)+(Y2-01<1)*2:GOTO3
080
3150 SYSPLOT,BX,BY:PRINT" {RVS OFF} " ;:SY
SPLOT,X2,Y2:PRINT" {RVS OFF} {WHT}#";:BX=X
2:BY=Y2:GOTO3080
3500 RETURN
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4005 POKESID+24,0
4010 PRINT" {HOME} {BLK} {CSD} THE GAME IS O
VER !!!"
4020 PRINT" THE WINNER IS PLAYER #"2+(L2=
0)
4990 RETURN
```


Leaky Faucet



This program is not a game per se, but rather a unique demonstration of some of the graphic capabilities of your Commodore 64. The game simulates a dripping faucet. The program allows you to position the faucet on the screen so that you can fill the lower half of the screen with "water" as quickly as possible. List lines 3600-3990. Line 3600 scans the keyboard, checking to see if any of the function keys have been pressed. If the f1 key is pressed, line 3605 will send the program to line 3700. From immediate mode, type GOTO 3700. At the top of your screen, you will see the first digit disappear in some of the numbers. The CHR\$(20) performs the same function as pressing the INST-DEL key. Type in GOTO 3650 from immediate mode to see what happens if the f3 key is pressed. If the f7 key is pressed (line 3608), the program drops through to line 3610, returns to the top of the program, and finally arrives at the END in line 60.

Line 3030 reads: HOME, CTRL-1, CTRL-0, 2 {CSR}, SHIFT-asterisk, CMDR-R, SHIFT-asterisk, 1 {CSD}, 5 {CSL}, 3 SHIFT-asterisk. CMDR-E, SHIFT-asterisk, SHIFT-I.

Line 3040 reads: 4 SHIFT-asterisk, 1 SHIFT-I, . . . , CMDR-Z, CMDR-X.

The circle is produced with a SHIFT-Q.
Lines 3652-3, the line is a CMDR-P.

```

10 REM *****
11 REM ***          ***
12 REM ** LEAKY FAUCET **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,6:POKE53280,14
1010 PRINT"(CLR)"TAB(8)"(WHT)*** THE LEA
KY FAUCET ***"
1020 PRINT:PRINT:PRINT
1030 PRINT"THIS IS NOT A GAME, BUT AN"
1031 PRINT"EXTREMELY ENTERTAINING GRAPHI
CS DEMO."
1032 PRINT:PRINT"WE HOPE YOU ENJOY IT !"
1033 PRINT:PRINT:PRINT
1034 PRINT"PRESS RETURN TO CONTINUE."
1035 GETANS$:IFANS$<>CHR$(13)THEN1035
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2005 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXT:POKESID+5,9
2020 PLOT=49152:FORI=0TO24:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 NU=100
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :

```

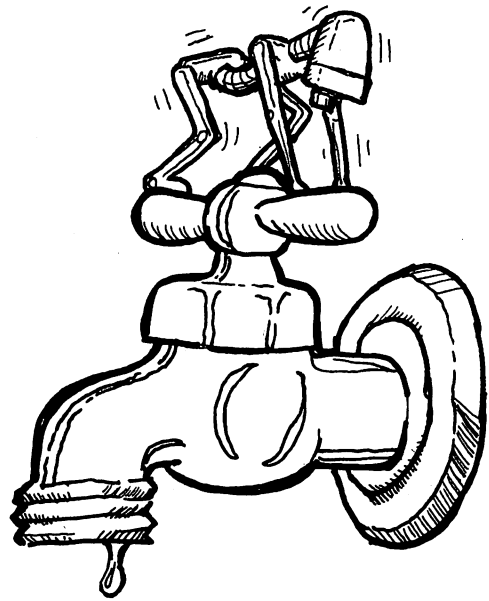


```

3010 PRINT "{CLR}"
3015 SYSPLOT,2,24:PRINT "{RVS ON}{CYN} F1
LEFT {WHT} F3 RIGHT {YEL} F7 START AGAI
N ";
3020 FORI=1TONU:PRINTMID$("{YEL}{BLK}{BR
WN}{WHT}{CYN}{LT GRN}{GRN}",1+INT(7*RND(
0)),1);
3021 SYSPLOT,1+INT(38*RND(0)),10+INT(14*
RND(0)):PRINT "{RVS ON} ";:NEXT
3030 PRINT "{HOME}{BLK}{RVS OFF}{CSR}{CSR
}—{CSD}{CSL}{CSL}{CSL}{CSL}{CSL}—
"
3040 PRINT "— | {CSD}{CSL}{CSL} —"
3110 I=5:RS=0
3120 SX=I:SY=4
3121 POKESID+24,15:POKESID+1,15:POKESID+
4,65
3122 FORPA=1TO10:NEXTPA
3123 POKESID+4,64:POKESID+1,0:POKESID+24
,0
3130 SYSPLOT,SX,SY:PRINT "{RVS OFF}{LT BL
U}#";:GOSUB3600
3135 S2=SY+1:IFS2>23THEN3200
3140 IFPEEK(1024+SX+40*S2)<>32THEN3200
3145 SYSPLOT,SX,SY:PRINT "{RVS OFF} ";:SY
=S2:GOTO3130
3200 LR=2*INT(2*RND(0))-1
3205 IFPEEK(1024+SX+LR+40*SY)<>32ANDPEEK
(1024+SX-LR+40*SY)<>32THEN3500
3215 IFPEEK(1024+SX+LR+40*SY)<>32THENLR=
-LR:GOTO3215
3220 S2=SX+LR:IFS2<10RS2>38THEN3500
3225 IFPEEK(1024+S2+40*SY)<>32THEN3500
3230 SYSPLOT,SX,SY:PRINT "{RVS OFF} ";:SX
=S2
3231 SYSPLOT,SX,SY:PRINT "{RVS OFF}{LT BL
U}#";:GOSUB3600
3235 IFSY<23THENIFPEEK(1024+SX+40*(SY+1)
)=32THEN3135
3240 GOTO3220
3245 SYSPLOT,SX,SY:PRINT "{RVS OFF} ";:SY
=S2:GOTO3130
3500 IFRS<>1THENSYPLOT,SX,SY:PRINT "{RVS
ON}{LT BLU} ";:GOTO3120
3510 GOTO3010
3600 GETANS$
3605 IFANS$="{F1}"THEN3700
3607 IFANS$="{F3}"THEN3640
3608 IFANS$="{F7}"THENRS=1
3610 RETURN

```

```
3640 IFI=36THENRETURN
3645 I=I+1
3650 PRINT" {HOME} {RVS OFF} {BLK} ";
3651 PRINTCHR$(148)
3652 PRINTCHR$(148) "-"
3653 PRINTCHR$(148) "-"
3654 PRINTCHR$(148)
3690 RETURN
3700 IFI=5THENRETURN
3745 I=I-1
3750 PRINT" {HOME} {CSR} "CHR$(20)
3751 PRINT" {CSR} "CHR$(20)
3752 PRINT" {CSR} "CHR$(20)
3753 PRINT" {CSR} "CHR$(20)
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4990 RETURN
```





In this game you try to alphabetize a scrambled list of letters. One square is left open so that you may move a letter into it. The computer will scramble the completed version approximately 125 times. You must unscramble the letter matrix in as few tries as possible. Remember, practice makes perfect! Before we look at the graphics, it seems appropriate to mention that proper graphics alignment is not a matter of trial and error, but entails working with graph paper and drawing the figure which is to be drawn on the screen. After figuring which coordinates will be filled in with what color, the writing of the program can take place. The nice pictures that you see on the screen are more the result of painstaking work than the result of sheer brilliance. Getting back to the program itself, let's look at some of the graphics. Note how each letter has its own subroutine: lines 2220-2223 draw the letter A, 2230 to 2233 draw the letter B and so forth down to the letter O. Remember that keeping your subroutines well partitioned makes it easier to follow the logic of your program. This may not seem important when you're writing the program, but if you ever have to go back and look at a program after being away from it for some time, you'll be glad that you structured it well.

In lines 2200 and following, the horizontal bars are produced with CMDR-I's, CMDR-O's, or CMDR-U's, depending on the thickness and position of the bars.

```

10 REM *****
11 REM ***          ***
12 REM **  MAGIC SQUARES  **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,1:POKE53280,1
1010 PRINT"{CLR}{BLU}"
1020 PRINTTAB(9)"*** MAGIC SQUARES ***"
1030 PRINT:PRINT:PRINT:PRINT
1031 PRINT"IN THIS GAME, YOU HAVE A 4 BY
4 GAME-"
1032 PRINT"BOARD. THE BOARD CONTAINS THE
LETTERS A-O."
1035 PRINT
1040 PRINT"THE OBJECT IS TO HORIZONTALLY
"
1041 PRINT"ALPHABETIZE THE SCRAMBLED GAM
E-BOARD."
1042 PRINT"YOU CAN MOVE A PIECE SIDEWAYS
OR UP AND"
1043 PRINT"DOWN, AS LONG AS THE EMPTY SQ
UARE IS NEXT TO IT."
1045 PRINT
1050 PRINT"YOUR PROGRESS WILL BE MONITOR
ED, AND YOU";
1051 PRINT"WILL BE TOLD HOW YOU ARE DOIN
G."
1055 PRINT:PRINT:PRINT
1060 PRINT"PRESS RETURN WHEN READY TO CO
NTINUE"
1070 GETANS$:IFANS$<>CHR$(13)THEN1070
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2005 DEFFNX(X)=(X-1)*8+4:DEFFNY(X)=(X-1)
*5+1

```

```

2010 DIMB(4,4):FORI=1TO4:FORJ=1TO4:K=K+1
:B(J,I)=K:NEXT:NEXT
2015 DIMDIR(4,2):FORI=1TO4:READDIR(I,1),
DIR(I,2):NEXT:DATA1,0,0,1,-1,0,0,-1
2020 PLOT=49152:FORI=0TO24:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 PRINT"{CLR}{BLU}";
2035 PRINTTAB(3)"{RVS ON}
"
2040 FORI=1TO4:FORJ=1TO4:PRINT"  {RVS O
N} {RVS OFF}          {RVS ON} {RVS OFF}
      {RVS ON} {RVS OFF}          {RVS ON} {R
VS OFF}          {RVS ON} {RVS OFF}"
2041 NEXT
2045 PRINT"  {RVS ON}
"
2050 NEXT
2055 GOSUB2100:SYSPLT,0,22:PRINT"  {BLK}
<< I'M NOW SCRAMBLING THE BOARD >>"
2056 SX=4:SY=4:SC=100+INT(50*RND(0)):FOR
K=1TOSC
2058 D=1+INT(4*RND(0)):PX=SX+DI(D,1):PY=
SY+DI(D,2)
2059 IFPX<10RPX>4ORPY<10RPY>4THEN2058
2060 B(SX,SY)=B(PX,PY)
2061 J=SX:I=SY:GOSUB2200
2062 B(PX,PY)=16:SX=PX:SY=PY
2065 J=SX:I=SY:GOSUB2200
2070 NEXTK:RETURN
2100 REM *** DRAW THE BOARD
2110 FORI=1TO4:FORJ=1TO4:PRINT"{RED}";:G
OSUB2200:NEXT:NEXT:RETURN
2200 :REM *** DRAW A LETTER
2201 X=FNX(J):Y=FNY(I)
2202 SYSPLT,X,Y:PRINT"{RED}";
2205 IFB(J,I)<8THENONB(J,I)GOSUB2220,223
0,2240,2250,2260,2270,2280:RETURN
2210 ONB(J,I)-7GOSUB2290,2300,2310,2320,
2330,2340,2350,2360,2370:RETURN
2220 PRINT"  "
2221 SYSPLT,X,Y+1:PRINT"  {RVS ON}---{RVS O
FF}--- --{RVS ON}---{RVS OFF} "
2222 SYSPLT,X,Y+2:PRINT"  {RVS ON} ===== {R
VS OFF} "

```

```

2223 SYSPLT,X,Y+3:PRINT" - - "
2225 RETURN
2230 PRINT" "
2231 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS O
FF} {RVS ON} {RVS OFF} "
2232 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} {RVS ON} {RVS OFF} "
2233 SYSPLT,X,Y+3:PRINT" {RVS ON} {RVS
OFF} "
2235 RETURN
2240 PRINT" "
2241 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS O
FF} "
2242 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} "
2243 SYSPLT,X,Y+3:PRINT" {RVS ON} {RVS
OFF} "
2245 RETURN
2250 PRINT" "
2251 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS O
FF} {RVS ON} {RVS OFF} "
2252 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} {RVS ON} {RVS OFF} "
2253 SYSPLT,X,Y+3:PRINT" {RVS ON} {RVS
OFF} "
2255 RETURN
2260 PRINT" "
2261 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS O
FF} "
2262 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} "
2263 SYSPLT,X,Y+3:PRINT" {RVS ON} {RVS
OFF} "
2265 RETURN
2270 PRINT" "
2271 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS O
FF} "
2272 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} "
2273 SYSPLT,X,Y+3:PRINT" {RVS ON} {RVS O
FF} "
2275 RETURN
2280 PRINT" "
2281 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS O
FF} "
2282 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} {RVS ON} {RVS OFF} "
2283 SYSPLT,X,Y+3:PRINT" {RVS ON} {RVS
OFF} "
2285 RETURN

```

```

2290 PRINT"  "
2291 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS O
FF}{RVS ON} {RVS OFF} "
2292 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} {RVS ON} {RVS OFF} "
2293 SYSPLT,X,Y+3:PRINT"  "
2295 RETURN
2300 PRINT"  "
2301 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS
OFF} "
2302 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS
OFF} "
2303 SYSPLT,X,Y+3:PRINT" {RVS ON}{RVS
S OFF} "
2305 RETURN
2310 PRINT"  "
2311 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS
OFF} "
2312 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS
OFF} "
2313 SYSPLT,X,Y+3:PRINT" {RVS ON}{RVS
OFF} "
2315 RETURN
2320 PRINT"  "
2321 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS O
FF}{RVS ON}{RVS OFF} "
2322 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} {RVS ON}{RVS OFF} "
2323 SYSPLT,X,Y+3:PRINT"  "
2325 RETURN
2330 PRINT"  "
2331 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS O
FF} "
2332 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} "
2333 SYSPLT,X,Y+3:PRINT" {RVS ON}{RVS
S OFF} "
2335 RETURN
2340 PRINT"  "
2341 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS
OFF}{RVS ON}{RVS OFF} "
2342 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} {RVS ON} {RVS OFF} "
2343 SYSPLT,X,Y+3:PRINT"  "
2345 RETURN
2350 PRINT"  "
2351 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS
OFF}{RVS ON} {RVS OFF} "
2352 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} {RVS ON} {RVS OFF} "

```

```

2353 SYSPLT,X,Y+3:PRINT" --- --- "
2355 RETURN
2360 PRINT" _____ "
2361 SYSPLT,X,Y+1:PRINT" {RVS ON} {RVS O
FF} {RVS ON} {RVS OFF} "
2362 SYSPLT,X,Y+2:PRINT" {RVS ON} {RVS O
FF} {RVS ON} {RVS OFF} "
2363 SYSPLT,X,Y+3:PRINT" {RVS ON} _____{R
VS OFF} "
2365 RETURN
2370 PRINT" "
2371 SYSPLT,X,Y+1:PRINT" "
2372 SYSPLT,X,Y+2:PRINT" "
2373 SYSPLT,X,Y+3:PRINT" "
2375 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 SYSPLT,0,21:FORI=1TO3
3011 PRINT"
"
3012 NEXT
3015 SYSPLT,0,22:PRINT" {BLK}MOVE WHICH
PIECE ?";:GETANS$:PRINTANS$;
3016 GETANS$:IFANS$=""THEN3016
3017 PRINTANS$:IFANS$>="A"ANDANS$<="O"TH
EN3020
3018 PRINT" LETTER MUST BE A TO O...";:
FORA=1TO1000:NEXTA:GOTO3010
3020 FORK=1TO4:PX=SX+DI(K,1):PY=SY-DI(K,
2)
3025 IFPX<1ORPX>4ORPY<1ORPY>4THEN3040
3030 IFB(PX,PY)=ASC(ANS$)-ASC("A")+1THEN
MO=MO+1:GOTO3100
3040 NEXTK
3045 PRINT" THAT LETTER IS BLOCKED...";
:FORA=1TO1000:NEXTA:GOTO3010
3100 B(SX,SY)=B(PX,PY):J=SX:I=SY:GOSUB22
00
3110 B(PX,PY)=16:SX=PX:SY=PY
3115 J=SX:I=SY:GOSUB2200
3200 K=0:FORI=1TO4:FORJ=1TO4:K=K+1
3210 IFB(J,I)=KTHENNEXTJ:NEXTI:RETURN
3220 GOTO3010
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4010 PRINT"{CLR}{BLK}YOU SOLVED IT!":PRI
NT"IT WAS SCRAMBLED"SC"TIMES."

```



```
4020 PRINT"AND YOU SOLVED IT IN"MO"MOVES  
."  
4030 INPUT "DO YOU WISH TO PLAY AGAIN";A  
NS$  
4040 IFLEFT$(ANS$,1)="Y"THENRUN  
4990 RETURN
```





Match the Key

If you have ever played or even seen Simon, then you should be able to recognize this game. A series of tones will be sounded and one of four corresponding color boxes will be lit up on the screen. Each box will have a specific tone. The four boxes are controlled by the four function keys. There are a number of mnemonic devices which make it easier to recall a long series, but you should try to develop one that works best for you. A very good player can repeat a sequence of twenty boxes, and an expert can repeat a sequence of thirty. Listing lines 3000-3200 will show you the various graphic routines that draw the boxes in the appropriate colors. Feel free to experiment with any lines whose function is unclear.

```

10 REM *****
11 REM ***          ***
12 REM ** MATCH THE KEY **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,12:POKE53280,12:PRINT"(CLR)"
1020 PRINTTAB(9)"(BLK)*** MATCH THE KEY
***":PRINT
1030 PRINT"THIS GAME WILL TEST YOUR MEMORY. YOU"
1031 PRINT"WILL BE SHOWN A SCREEN WITH FOUR COLORED";
1032 PRINT"BLOCKS ON IT. THE COMPUTER WILL LIGHT"
1033 PRINT"UP ONE OF THE BLOCKS, AND SOUND ITS"
1034 PRINT"CORRESPONDING TONE."
1035 PRINT
1040 PRINT"INPUT THE FUNCTION KEY FOR THE LIGHTED"
1041 PRINT"BLOCK. IF YOU ARE CORRECT THE COMPUTER"
1042 PRINT"WILL REPEAT THE SEQUENCE AND ADD ANOTHER";
1043 PRINT"COLOR TO IT."
1045 PRINT
1050 PRINT"AFTER EACH ADDITION, YOU MUST RETYPE THE";
1051 PRINT"ENTIRE SEQUENCE. YOU ARE ALLOWED THREE"
1052 PRINT"MISTAKES PER GAME."
1055 PRINT:PRINT"PRESS RETURN TO BEGIN PLAY."
1060 GETANS$:IFANS$<>CHR$(13)THEN1060
1990 RETURN
2000 :

```

```

2001 :REM *** SETUP
2002 :
2005 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXT:POKESID+5,9
2010 DIMSEQ(50):FORI=1T050:SEQ(I)=1+INT(
4*RND(0)):NEXT
2015 POKE53281,12:POKE53280,12:PRINT"{CL
R}"
2020 PLOT=49152:FORI=0T024:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 FORI=1T019STEP5
2033 FORJ=0T03
2034 SYSPLOT,10,J+I:PRINT"{RVS ON}{BLK}
";
2035 NEXT
2040 SYSPLOT,20,I+1:PRINT"{RVS OFF}{WHT}
F"INT(1+2*I/5):NEXT
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3005 POKESID+24,12
3010 FORI=1T050:SC=I-1-M:SYSPLOT,8,23:PR
INT"MISSES:"M"{CSL}          SCORE:"SC"{CSL
} {HOME}";
3020 FORJ=1TOI:PRINTMID$("{BRWN}{YEL}{GR
N}{BLU}",SEQ(J),1);
3025 Y=5*SEQ(J)-3:SYSPLOT,11,Y:PRINT"{RV
S ON}          "
3026 SYSPLOT,11,Y+1:PRINT"{RVS ON}          "
3027 POKESID+1,60*(5-SEQ(J)):POKESID+4,3
3:FORPA=1T0200:NEXT
3028 POKESID+4,32:POKESID+1,0
3030 SYSPLOT,11,Y:PRINT"{RVS ON}{BLK}
"
3031 SYSPLOT,11,Y+1:PRINT"{RVS ON}{BLK}
"
3035 NEXTJ
3100 FORJ=1TOI
3105 X=0
3110 GETX$:IFX$=""THEN3110
3120 IFX$="{F1}"THENX=1
3121 IFX$="{F3}"THENX=2
3122 IFX$="{F5}"THENX=3

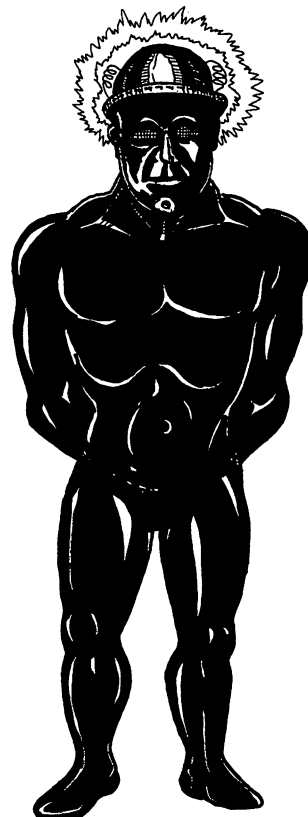
```

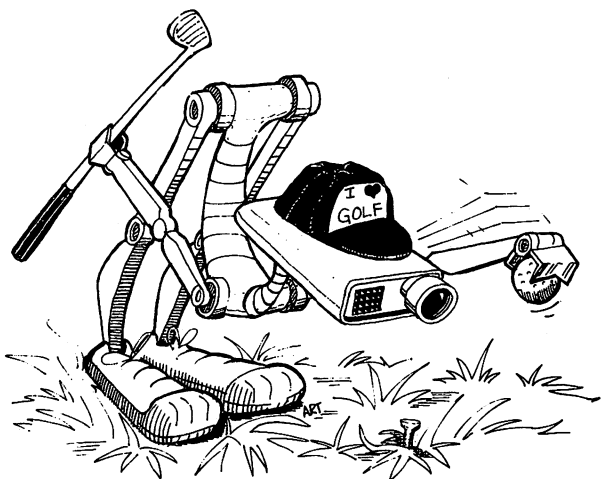
```

3123 IFX$="{F7}"THENX=4
3130 IFX=0THEN3110
3135 PRINTMID$("{BRWN}{YEL}{GRN}{BLU}",X
,1);
3136 Y=5*X-3:SYSPLOT,11,Y:PRINT"{RVS ON}
"
3137 SYSPLOT,11,Y+1:PRINT"{RVS ON}
"
3138 POKESID+1,60*(5-X):POKESID+4,33:FOR
PA=1TO200:NEXT
3139 POKESID+4,32:POKESID+1,0
3140 SYSPLOT,11,Y:PRINT"{RVS ON}{BLK}
"
3142 SYSPLOT,11,Y+1:PRINT"{RVS ON}{BLK}
"
3150 IFX=SEQ(J)THEN3200
3155 POKESID+1,20:POKESID+4,33:FORPA=1TO
400:NEXT
3156 POKESID+4,32:POKESID+1,0
3160 M=M+1:IFM<3THEN3300
3165 RETURN
3200 NEXTJ
3300 FORPA=1TO500:NEXTPA:NEXTI
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4010 PRINT"{CLR}"
4020 FORJ=0TO24
4030 PRINTMID$("{BLK}{WHT}{RED}{CYN}{PUR
}{GRN}{BLU}{YEL}{ORNG}{BRWN}{LT RED}{GRY
1}{GRY 2}{LT GRN}{LT BLU}{GRY 3}",1+INT
(16*RND(0)),1);"{RVS ON}";
4031 SYSPLOT,0,J:PRINT"
";
4032 PRINTMID$("{BLK}{WHT}{RED}{CYN}{PUR
}{GRN}{BLU}{YEL}{ORNG}{BRWN}{LT RED}{GRY
1}{GRY 2}{LT GRN}{LT BLU}{GRY 3}",1+INT
(16*RND(0)),1);"{RVS ON}";
4033 SYSPLOT,0,24-J:PRINT"
";
4034 PRINTMID$("{BLK}{WHT}{RED}{CYN}{PUR
}{GRN}{BLU}{YEL}{ORNG}{BRWN}{LT RED}{GRY
1}{GRY 2}{LT GRN}{LT BLU}{GRY 3}",1+INT
(16*RND(0)),1);"{RVS ON}";
4035 SYSPLOT,0,24-J:PRINT"
";
4036 PRINTMID$("{BLK}{WHT}{RED}{CYN}{PUR
}{GRN}{BLU}{YEL}{ORNG}{BRWN}{LT RED}{GRY
1}{GRY 2}{LT GRN}{LT BLU}{GRY 3}",1+INT
(16*RND(0)),1);"{RVS ON}";

```

```
4037 SYSPLOT,0,J:PRINT"  
";  
4040 POKESID+1,10+2*J:POKESID+4,33:FORPA  
=1TO20:NEXT  
4045 POKESID+4,32:POKESID+1,0  
4050 NEXTJ  
4060 FORI=1TO500:NEXT  
4070 PRINT"{CLR}"  
4080 SYSPLOT,0,12  
4085 IFM<>3THEN4110  
4090 PRINT"{BLK}{RVS OFF}<<< I'M SORRY Y  
OU LOST . . .>>>"  
4095 IFSC<1THENPRINT"YOU GOT THEM ALL WR  
ONG !!!"  
4098 IF SC<50 THEN PRINT"{BLK}{RVS OFF}Y  
OU GOT A TOTAL OF"SC"CORRECT"  
4100 RETURN  
4110 PRINT"{BLK}{RVS OFF}<<< YOU GOT ALL  
50 OF THEM !!! >>>":RETURN  
4990 RETURN
```





Miniature GOLF

This program is an excellent simulation of a game of miniature golf. There are nine holes on the course, which includes water traps, sand hazards, uneven hills, and blocks that you must go around. The graphics are undoubtedly the star attraction of this game, so let's first take a look at how each of the holes is created. Our graphics routine is included in lines 2030-2036. Each hole is placed in a subroutine of its own; the subroutine starting at line 2100 contains the graphic representation of hole #1, 2200 starts hole #2, 2300 hole #3, and so on through 2900 for hole #9. Try changing some of the colors in these routines for some interesting graphic effects. The rest of the graphics, including the movement of the ball, can be found in lines 3000-3999. Your score, as well as your average per hole, is calculated in lines 4025-4026.

The ball (circle) is made with a SHIFT-Q.

The horizontal bars are produced with SHIFT-I's.

The vertical bars are produced with SHIFT-K's.

The small squares (in the upper lefthand corners of the character position) are produced with a CMDR-V.

The small squares (in the upper righthand corners of the character position) are produced with a CMDR-C.

Other characters like CMDR-A, CMDR-S, CMDR-Z, CMDR-X, SHIFT-C, appear also, as in line 2530.

```

10 REM *****
11 REM ***          ***
12 REM ** MINIATURE GOLF **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTS
1002 :
1003 POKE53281,1:POKE53280,1
1010 PRINT"{CLR}{GRY 2}"
1020 PRINTTAB(9)"*** MINIATURE GOLF ***"
1025 PRINT:PRINT
1030 PRINT"WELCOME TO THE CAPELLA COUNTRY CLUB. THIS BEAUTIFUL NINE-HOLE ";
1031 PRINT"MINIATURE GOLF COURSE IS OPEN AND WAITING FOR YOU !"
1035 PRINT
1040 PRINT"YOU SHOULD KNOW THE IDIOSYNCRASIES OF THE COURSE BEFORE YOU BEGIN PLAY."
1041 PRINT
1045 PRINT"TO PUTT THE BALL, YOU HAVE TO INPUT WHICH DIRECTION YOU WANT TO";
1046 PRINT" AIM IT."
1047 PRINT"THERE ARE EIGHT DIRECTIONS, SHOWN BELOW.YOUR BALL IS ASSUMED TO BE AT *."
1050 PRINT:PRINTTAB(17)"2 1 8":PRINTTAB(17)"3 * 7":PRINTTAB(17)"4 5 6"
1060 PRINT:PRINT:PRINT"PRESS RETURN WHEN READY TO CONTINUE"
1065 GETX$:IFX$(<>CHR$(13))THEN1065
1070 PRINT"{CLR}"
1075 PRINTTAB(9)"*** MINIATURE GOLF ***"
1076 PRINT:PRINT
1080 PRINT"THEN YOU MUST INPUT HOW HARD TO HIT THE BALL. THE SPEED SHOULD ";
1081 PRINT"BE SOME NUMBER BETWEEN 0.00 AND 5.00. FOR EXAMPLE, YOU COULD";
1082 PRINT" HIT THE BALL A RELATIVE SPEE

```

```

D OF 3.2."
1085 PRINT
1090 PRINT"IT WILL TAKE A FEW TRIES BEFO
RE YOU GET THE FEEL OF HOW HARD TO PUTT
";
1095 PRINT"THE BALL.":PRINT
1100 PRINT"ALSO, THERE ARE FOUR TYPES OF
HAZARDS ONTHE COURSE. YOU SHOULD ";
1105 PRINT"BE AWARE OF WHATTHEY ARE AND
WHAT AFFECT THEY HAVE ON YOU ";
1106 PRINT"AND YOUR BALL."
1110 PRINT:PRINT:PRINT:PRINT"PRESS RETUR
N WHEN READY TO CONTINUE"
1115 GETX$:IFX$<>CHR$(13)THEN1115
1120 PRINT"{CLR}"
1125 PRINTTAB(9)"*** MINIATURE GOLF ***
1126 PRINT:PRINT
1130 PRINT"BLOCKS: THESE ARE LIKE WALLS.
YOU MUST PUTT AROUND THEM."
1135 PRINT
1140 PRINT"YELLOW TRAPS REPRESENT SAND.
YOUR BALL CANNOT PENETRATE THROUGH ";
1141 PRINT"A SAND TRAP. THE PENALTY FO
R LANDING IN THE SAND IS ONE STROKE."
1145 PRINT
1150 PRINT"WATER: LIKE SAND, THESE BLUE
HAZARDS WILL SLOW AND STOP YOUR BALL.
";
1151 PRINT" THE PENALTY FOR LANDIN
G IN WATER IS ONE STROKE."
1155 PRINT
1160 PRINT"UNEVEN SURFACES: THESE RED HA
ZARDS CAUSETHE BALL TO ROLL IN ";
1165 PRINT"A DIRECTION WHICH ISUNPREDICT
ABLE. THERE IS NO PENALTY ";
1166 PRINT"FORHITTING THIS HAZARD."
1170 PRINT:PRINT:PRINT"PRESS RETURN WHEN
READY TO CONTINUE"
1175 GETX$:IFX$<>CHR$(13)THEN1175
1180 PRINT"{CLR}{GRY 2}"
1185 PRINTTAB(9)"*** MINIATURE GOLF ***
1186 PRINT:PRINT
1190 PRINT"YOU ARE TRYING TO SINK THE BA
LL IN AS FEW TRIES AS POSSIBLE. ";
1191 PRINT"THE HOLE IS THE BLACK SQUARE.
"
1195 PRINT
1200 PRINT"IF YOU HIT THE BALL TOO HARD,
IT WILL JUMP OVER THE HOLE AND ";
1205 PRINT"CONTINUE ROLLING.IT MAY ALSO

```

```

CHANGE DIRECTION, SO BE SURE TO HIT ";
1206 PRINT "THE BALL JUST HARD ENOUGH."
1210 PRINT:PRINT:PRINT "PRESS RETURN WHEN
      READY TO CONTINUE"
1215 GETX$:IFX$(<>CHR$(13))THEN1215
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2010 DIMHA(9,5):FORI=1TO9:FORJ=1TO5:READ
HA(I,J):NEXT:NEXT
2011 DATA 0,0,0,0,2
2012 DATA 1,0,0,0,3
2013 DATA 0,1,0,0,3
2014 DATA 1,1,0,0,3
2015 DATA 0,0,0,1,3
2016 DATA 0,0,1,1,3
2017 DATA 1,0,1,0,3
2018 DATA 0,1,1,0,4
2019 DATA 0,0,1,1,3
2020 DIMDI(8,2):FORI=1TO8:READDI(I,1),DI
(I,2):NEXT
2025 DATA 0,-1,-1,-1,-1,0,-1,1,0,1,
1,1,1,0,1,-1
2030 PLT=49152:FORI=0TO24:READC:POKEPLT+
I,C:NEXTI
2031 DATA32,16,192,132
2032 DATA253,32,16,192
2033 DATA152,170,164,253
2034 DATA24,76,240,255
2035 DATA32,253,174,32
2036 DATA158,173,76,170,177
2040 DIMV(8):FORI=1TO8:READV(I):NEXT:DAT
A-40,-41,-1,39,40,41,1,-39
2050 :
2055 DEF FNCO(X)=PEEK(55296+BX+40*BY+X)A
ND15
2060 DEF FNSC(X)=PEEK(1024+BX+40*BY+X)
2090 RETURN
2100 REM *** HOLE 1
2105 GOSUB2050
2106 PRINT "{HOME}"
2110 PRINTTAB(10)"{BLK}
=====
"
2112 FORI=1TO15
2115 PRINTTAB(10)"{RVS ON}||{GRN}
      {RVS OFF}{BLK}|"
2120 NEXT
2125 PRINTTAB(10)"{BLK}={RVS ON}
=====
      {RVS OFF}="

```



```

2130 PRINT "{HOME} {CSD} {CSD} {CSD} {CSD}" TA
B(19) "{RVS ON} "
2135 BY=16:BX=11+INT(17*RND(0)):SYSPLT,B
X,BY:PRINT "{RVS ON} {GRN}#";
2140 RETURN
2200 REM *** HOLE 2
2205 GOSUB2050
2206 PRINT "{HOME}"
2210 PRINTTAB(8) "{BLK}
"
2212 FORI=1TO5
2215 PRINTTAB(8) "{RVS ON}# {GRN}
{RVS OFF} {BLK}# "
2220 NEXT
2221 PRINTTAB(8) "{RVS ON}# {GRN}
{BLU} {BLK} {RVS OFF}="
2222 PRINTTAB(8) "{RVS ON}# {GRN}
{BLU} {RVS OFF} {BLK}# "
2223 PRINTTAB(8) "{RVS ON}# {GRN}
{BLU} {RVS OFF} {BLK}# "
2224 PRINTTAB(8) "{RVS ON}# {GRN}
{BLU} {RVS OFF} {BLK}# "
2225 PRINTTAB(8) "{RVS ON}# {GRN}
{RVS OFF} {BLK}# "
2226 FORI=1TO6
2227 PRINTTAB(8) "{RVS ON}# {GRN}
{RVS OFF} {BLK}# "
2230 NEXT
2235 PRINTTAB(8) " "{RVS ON}
{RVS OFF}="
2240 BY=17:BX=9+INT(15*RND(0)):SYSPLT,BX
,BY:PRINT "{RVS ON} {GRN}#";
2245 PRINT "{HOME} {CSD} {CSD} {CSD} {CSD}" TA
B(30) "{RVS ON} {BLK} "
2250 RETURN
2300 REM *** HOLE 3
2306 PRINT "{HOME}"
2310 PRINTTAB(20) "{BLK}
"
2312 FORI=1TO5
2315 PRINTTAB(20) "{RVS ON}# {GRN}
{RVS OFF} {BLK}# "
2320 NEXT
2321 PRINTTAB(8) "{RVS OFF} {R
VS ON}" {GRN} {RVS OFF} {BLK}
)# "
2322 PRINTTAB(8) "{RVS ON}# {GRN}
{RVS OFF} {BLK}# "
2323 PRINTTAB(8) "{RVS ON}# {GRN}
{YEL} {RVS OFF} {BLK}# "

```



```

L}>{CSL}>{CSL}      {CSD}>{CSL}>{CSL}>{CSL}>{CSL}
{CSL}      "
2475 SYSPLT,32,10:PRINT">{RVS ON}>{BLK} "
2480 BY=16:BX=8+INT(10*RND(0)):SYSPLT,BX
, BY:PRINT">{RVS ON}>{GRN}>#"
2490 RETURN
2500 REM *** HOLE 5
2506 PRINT">{HOME}>"
2510 PRINTTAB(10)">{BLK}>
=====
"
2512 FORI=1TO15
2515 PRINTTAB(10)">{RVS ON}>||>{GRN}
      >{RVS OFF}>{BLK}>|| "
2520 NEXT
2525 PRINTTAB(10)">{BLK}> ">{RVS ON}>
=====
>{RVS OFF}>" "
2530 SYSPLT,19,6:PRINT">{RVS ON}>{GRN} |
-----| "
2535 SYSPLT,22,4:PRINT">{BLK}>{RVS ON} "
2540 BY=16:BX=11+INT(23*RND(0)):SYSPLT,B
X, BY:PRINT">{GRN}>{RVS ON}>#"
2590 RETURN
2600 REM *** HOLE 6
2605 GOSUB2050
2606 PRINT">{HOME}>"
2610 PRINTTAB(8)">{BLK}>
=====
"
2612 FORI=1TO7
2615 PRINTTAB(8)">{RVS ON}>||>{GRN}
      >{RVS OFF}>{BLK}>|| "
2620 NEXT
2625 PRINTTAB(8)">{RVS ON}>||>{GRN}
      >{BLK}> =====>{RVS OFF}>" "
2630 FORI=1TO7
2635 PRINTTAB(8)">{RVS ON}>||>{GRN}
      >{RVS OFF}>{BLK}>|| "
2640 NEXT
2645 PRINTTAB(8)"> ">{RVS ON}>
=====
>{RVS OFF}>" "
2650 SYSPLT,27,4:PRINT">{RVS ON}>{GRN} >{CS
D}>{CSL} |>{CSD}>{CSL} |"
2655 SYSPLT,20,5:PRINT">{RED}>{RVS ON}
>{CSD}>{CSL}>{CSL}>{CSL}>{CSL}>{CSL}>{CSL}
      >{CSD}>{CSL}>{CSL}>{CSL}>{CSL}>{CSL}      >{CSD
}>{CSL}>{CSL}>{CSL}      "
2660 SYSPLT,30,5:PRINT">{RVS ON}>{BLK} "
2665 BY=16:BX=9+INT(15*RND(0)):SYSPLT,BX
, BY:PRINT">{RVS ON}>{GRN}>#"
2690 RETURN
2700 REM *** HOLE 7

```

```

2706 PRINT" {HOME} "
2710 PRINTTAB (8) " {BLK} ..... "
2712 FORI=1TO4
2715 PRINTTAB (8) " {RVS ON}# {GRN}
      {RVS OFF} {BLK}# "
2720 NEXT
2721 PRINTTAB (8) " {RVS ON}# {GRN}
      {BLK} " {RVS OFF} ..... "
2725 FORI=1TO4
2727 PRINTTAB (8) " {RVS ON}# {GRN}
      {BLK} {RVS OFF}# "

2730 NEXT
2735 PRINTTAB (8) " {BLK} " {RVS ON} .....
      ..... {GRN}          {BLK} {RVS OFF}#
"

2740 FORI=1TO5
2745 PRINTTAB (24) " {RVS ON} {BLK}# {GRN}
      {BLK} {RVS OFF}# "

2750 NEXT
2755 PRINTTAB (24) " {BLK} " {RVS ON} .....
      ..... {RVS OFF} " "
2760 SYSPLT, 9, 7: PRINT" {BLU} {RVS ON}      {
CSD} {CSL} {CSL} {CSL} {CSL}      {CSD} {CSL} {
CSL} {CSL} {CSL}      {CSD} {CSL} {CSL} {CSL}
{CSL} {CSL}      "
2765 SYSPLT, 27, 7: PRINT" {RED} {RVS ON}
      {CSD} {CSL} {CSL} {CSL} {CSL} {CSL} {CSL}
      {CSD} {CSL} {CSL} {CSL} {CSL} {CSL} {CSL}
}      {CSD} {CSL} {CSL} {CSL} {CSL} {CSL} {C
SL} {CSL}      "
2770 SYSPLT, 13, 4: PRINT" {RVS ON} {BLK} "
2775 BY=16: BX=25+INT (13*RND (0)): SYSPLT, B
X, BY: PRINT" {RVS ON} {GRN}# "
2799 RETURN
2800 REM *** HOLE 8
2806 PRINT" {HOME} "
2810 PRINTTAB (7) " {BLK} .....
      ..... "
2812 FORI=1TO6
2815 PRINTTAB (7) " {RVS ON}# {GRN}
      {RVS OFF} {BLK}# "

2820 NEXT
2821 PRINTTAB (7) " {RVS ON}# {GRN}
      {BLK} ..... {GRN}          {RVS OFF} {
BLK}# "
2825 FORI=1TO4
2830 PRINTTAB (7) " {RVS ON}# {GRN}
      {BLK} {RVS OFF}#          {RVS ON}# {GRN}
      {RVS OFF} {BLK}# "
2835 NEXT

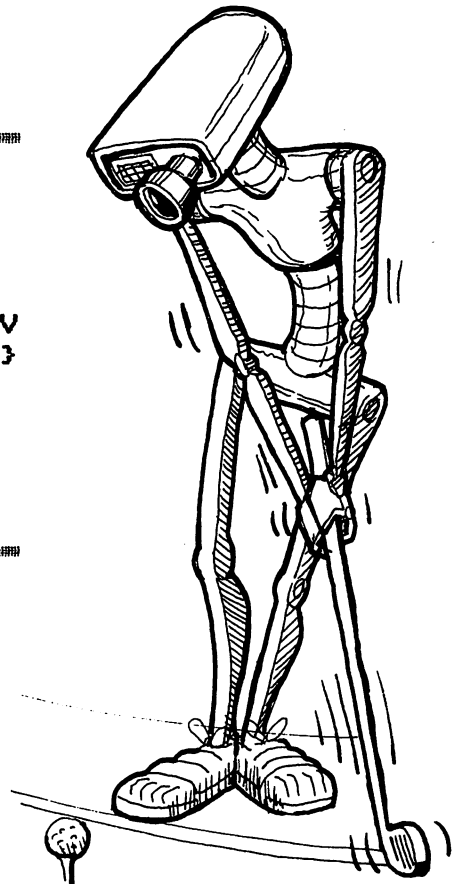
```



```

2840 PRINTTAB(7) " "{RVS ON} .....{RVS
OFF}"      {RVS ON}|| {GRN}      {BL
K}{RVS OFF}|| "
2845 FORI=1TO3
2850 PRINTTAB(26) " "{RVS ON}|| {GRN}
{BLK}{RVS OFF}|| "
2855 NEXT
2860 PRINTTAB(26) " "{RVS ON} .....{RV
S OFF}" "
2865 SYSPLT,28,2:PRINT" {RVS ON} {RED}
"
2866 SYSPLT,29,3:PRINT" {RVS ON}      "
2867 SYSPLT,30,4:PRINT" {RVS ON}      "
2868 SYSPLT,30,5:PRINT" {RVS ON}      "
2869 SYSPLT,32,6:PRINT" {RVS ON}      "
2870 SYSPLT,33,7:PRINT" {RVS ON}      "
2875 SYSPLT,13,3:PRINT" {RVS ON} {YEL}  {
CSD} {CSL} {CSL} {CSL} {CSL}      {CSD} {CSL} {
CSL} {CSL} {CSL} {CSL}      {CSD} {CSL} {CSL} {
CSL} {CSL}      "
2880 SYSPLT,13,10:PRINT" {RVS ON} {BLK} "
2885 BY=16:BX=27+INT(10*RND(0)):SYSPLT,B
X,BY:PRINT" {RVS ON} {GRN}|| "
2890 RETURN
2900 REM *** HOLE 9
2905 GOSUB2050
2906 PRINT" {HOME}"
2910 PRINTTAB(8) " {BLK} .....
..... "
2912 FORI=1TO9
2915 PRINTTAB(8) " "{RVS ON}|| {GRN}
{RVS OFF} {BLK}|| "
2920 NEXT
2921 PRINTTAB(8) " "{RVS ON} .....{RV
S ON}|| {GRN}      {RVS OFF} {BLK}
|| "
2925 FORI=1TO5
2927 PRINTTAB(20) " "{RVS ON}|| {GRN}
{RVS OFF} {BLK}|| "
2930 NEXT
2935 PRINTTAB(20) " "{RVS ON} .....
.....{RVS OFF}" "
2940 SYSPLT,21,2:PRINT" {RVS ON} {RED}
"
2941 SYSPLT,22,3:PRINT" {RVS ON}
"
2942 SYSPLT,23,4:PRINT" {RVS ON}
"
2943 SYSPLT,24,5:PRINT" {RVS ON}
"

```



```

2944 SYSPLT,25,6:PRINT"{RVS ON}
"
2950 SYSPLT,11,6:PRINT"{BLK}{RVS ON} "
2955 SYSPLT,14,5:PRINT"{RVS ON}{GRN}{CS
D}{CSL}{CSD}{CSL}-"
2960 BY=16:BX=21+INT(15*RND(0)):SYSPLT,B
X,BY:PRINT"{RVS ON}{GRN}#"
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 FORHO=1TO9:BC$="{GRN}"
3015 PRINT"{CLR}";
3020 ONHOGOSUB2100,2200,2300,2400,2500,2
600,2700,2800,2900
3030 PRINT"{HOME}{RVS ON}{LT BLU}HOLE NU
MBER:"HO"PAR:"HA(HO,5)"SCORE:"SC
3031 PRINT:PRINT"TRAPS:"":IFHA(HO,1)THENP
RINT"{BLU}WATER"
3032 IFHA(HO,2)THENPRINT"{YEL}SAND"
3033 IFHA(HO,3)THENPRINT"{RED}UNEVEN"
3034 IFHA(HO,4)THENPRINT"{BLK}BLOCKS"
3035 PRINT:PRINT"{RVS ON}{BLK}2 1 8":PRI
NT"{RVS ON}3 # 7":PRINT"{RVS ON}4 5 6"
3036 SYSPLT,0,19:FORI=1TO5:PRINT"
":NEXT
3038 D$="":SYSPLT,0,20:INPUT"DIRECTION (
1-8)";D$:DIR=VAL(D$)
3040 IFDIR>0ANDDIR<9ANDDIR=INT(DIR)THEN3
045
3041 PRINT"THE DIRECTION IS FROM 1 TO 8
...":FORPA=1TO1500:NEXT:GOTO3036
3045 SYSPLT,0,19:FORI=1TO5:PRINT"
":NEXT
3046 D$="":SYSPLT,0,20:INPUT"SPEED (0-5)
";D$:SP=VAL(D$)
3050 IFSP=>0ANDSP<=5THEN3055
3051 PRINT"THE DIRECTION IS FROM 0 TO 5
...":FORPA=1TO1500:NEXT:GOTO3045
3055 UF=0:REM CLEAR UNEVEN FLAG
3056 TF=0:REM CLEAR TRAP FLAG
3100 X2=BX+DI(DI,1):Y2=BY+DI(DI,2)
3110 D2=V(DI):CO=FNCO(D2):IFCO<>5ORFNSC(
D2)<>160THEN3120
3115 SYSPLT,BX,BY:PRINTBC$"{RVS ON} "":B
X=X2:BY=Y2:SYSPLT,BX,BY:PRINT"{RVS ON}{G
RN}#"
3116 BC$="{GRN}":GOTO3900
3120 IFCO=0ANDFNSC(D2)<>160THEN3200:REM
HIT A WALL

```

```

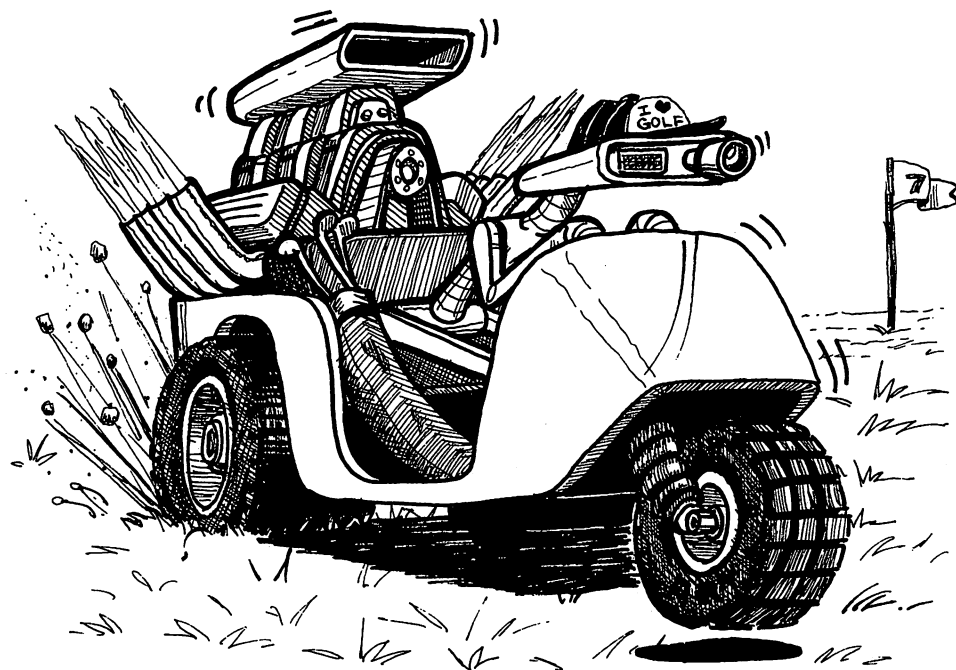
3121 IFCO=6THEN3300:REM WATER
3122 IFCO=7THEN3400:REM SAND
3123 IFCO=2THEN3500:REM UNEVEN
3124 IFCO=0THEN3600:REM THE HOLE
3125 REM WE HIT A BLOCK
3200 IFDI=1ORDI=3THENDI=DI+4:GOTO3100
3201 IFDI=7ORDI=5THENDI=DI-4:GOTO3100
3210 ONDI/2GOTO3220,3240,3260,3280
3220 IFFNSC(D2+1)<>160ANDFNSC(D2+40)<>16
OTHENDI=6:GOTO3100
3225 IFFNSC(D2+1)<>160THENDI=4:GOTO3100
3230 IFFNSC(D2+40)<>160THENDI=8:GOTO3100
3235 DI=6:GOTO3100
3240 IFFNSC(D2+1)<>160ANDFNSC(D2-40)<>16
OTHENDI=8:GOTO3100
3245 IFFNSC(D2+1)<>160THENDI=2:GOTO3100
3250 IFFNSC(D2-40)<>160THENDI=6:GOTO3100
3255 DI=8:GOTO3100
3260 IFFNSC(D2-1)<>160ANDFNSC(D2-40)<>16
OTHENDI=2:GOTO3100
3265 IFFNSC(D2-1)<>160THENDI=8:GOTO3100
3270 IFFNSC(D2-40)<>160THENDI=4:GOTO3100
3275 DI=2:GOTO3100
3280 IFFNSC(D2-1)<>160ANDFNSC(D2+40)<>16
OTHENDI=4:GOTO3100
3285 IFFNSC(D2-1)<>160THENDI=6:GOTO3100
3290 IFFNSC(D2+40)<>160THENDI=2:GOTO3100
3295 DI=4:GOTO3100
3300 IFTFTHEN3310
3305 TF=3:SC=SC+1:REM TRAP FLAG
3310 SYSPLT,BX,BY:PRINTBC$"{RVS ON} ";
3315 BX=X2:BY=Y2:SYSPLT,BX,BY:PRINT"{BLU
}{RVS ON}#";BC$="{BLU}"
3320 GOTO3900
3400 IFTFTHEN3410
3405 TF=3:SC=SC+1:REM TRAP FLAG
3410 SYSPLT,BX,BY:PRINTBC$"{RVS ON} ";
3415 BX=X2:BY=Y2:SYSPLT,BX,BY:PRINT"{YEL
}{RVS ON}#";BC$="{YEL}"
3420 GOTO3900
3500 IFUFTHEN3520
3505 UF=1:REM UNEVEN FLAG, HAVE RE ROLLE
D BALL OFF COURSE YET?...
3510 DI=DI+2*INT(2*RND(0))-1
3515 IFDI=0THENDI=8
3516 IFDI=9THENDI=1
3520 SYSPLT,BX,BY:PRINTBC$"{RVS ON} ";
3525 BX=X2:BY=Y2:SYSPLT,BX,BY:PRINT"{RED
}{RVS ON}#";BC$="{RED}"
3530 GOTO3900

```

```

3600 SYSPLT, BX, BY: PRINT " {RVS ON} {GRN} " :
SYSPLT, X2, Y2: PRINT " {RVS ON} {WHT} "
3601 SYSPLT, X2, Y2: PRINT " {RVS ON} {BLK} " ;
3602 DI=DI+2*INT(2*RND(0))-1
3604 IFDI=0THENDI=8
3605 IFDI=9THENDI=1
3606 SP=SP-.5: IFSP>0THENX2=X2+DI(DI,1): Y
2=Y2+DI(DI,2): GOTO3110
3610 SC=SC+1: GOTO3990
3900 :
3902 IFTF>0THENTF=TF-1: IFTF=0THEN3910
3905 SP=SP-.3: IFSP>0THEN3100
3910 SC=SC+1: GOTO3030
3980 GETX$: IFX$="" THEN3980
3990 NEXTHO: RETURN
4000 :
4001 : REM *** END
4002 :
4005 FORPA=1TO200: NEXTPA
4010 PRINT " {CLR} {CSD} {CSD} {CSD} " ; TAB(9) "
*** MINIATURE GOLF ***"
4015 PRINT: PRINT: PRINT: PRINT
4020 PRINT "THE GAME IS OVER !"
4022 PRINT
4025 PRINT "ON THE PAR 27 COURSE, YOU SHO
T": PRINT "A ROUND OF"SC". THAT IS AN"
4026 PRINT "AVERAGE OF"INT(SC*100/9)/100"
SHOTS PER HOLE. "
4030 PRINT: PRINT: PRINT
4035 PRINT "HOPE YOU ENJOYED THE GAME!"
4990 RETURN

```



Moving Target

(Requires Paddles)

This is a paddle game for one player. The object is to shoot the different colored targets that move across the screen. For the most part, this is a game of dexterity and timing, but there is a little bit of luck involved as well. As you can see, there is a color bar across the bottom of the screen which is set up in lines 2045 (the bar itself), 2050 (the target values), and 2051 (the white bar at the bottom of the screen). Once again, the graphics routines may be found in lines 3000 and following. The section that the REM in line 40 calls PLAY! is actually where all of the WORK is being done (movement of the targets, of the missile launcher, and the missile). As usual, feel free to modify the program to suit your needs.

The circle in line 3547 can be produced with a SHIFT-W.

```

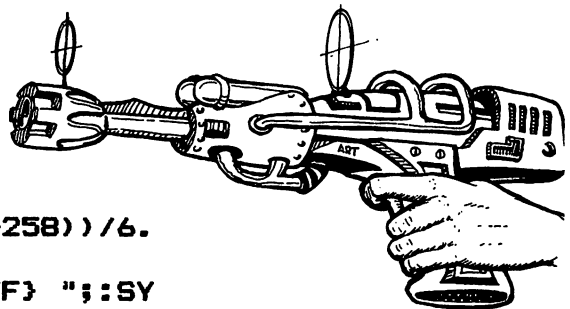
10 REM *****
11 REM ***          ***
12 REM **  MOVING TARGET **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,12:POKE53280,14
1010 PRINT"{CLR}"TAB(10)"{BLU}*** MOVING
    TARGET ***"
1015 PRINT:PRINT
1020 PRINT"IN THIS GAME, YOU CONTROL A M
    ISSILE'S"
1030 PRINT"LAUNCH SITE.  THE LAUNCHER IS
    CONTROLLED";
1031 PRINT"BY PADDLE #1 (PORT 2)..."
1032 PRINT
1040 PRINT"PRESSING THE BUTTON ON THE PA
    DDLE WILL"
1041 PRINT"RELEASE A ROCKET.  TRY TO HIT
    ONE OF THE";
1042 PRINT"THREE MOVING TARGETS ABOVE YO
    U."
1050 PRINT:PRINT"DIFFERENT COLORS ARE WO
    RTH DIFFERENT"
1051 PRINT"AMOUNTS OF POINTS.  SHOOT FOR
    THE BEST SCORE."
1055 PRINT:PRINT:PRINT"PRESS RETURN TO B
    EGIN PLAY!"
1060 GETANS$:IFANS$<>CHR$(13)THEN1060
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2020 PLOT=49664:FORI=0TO24:READC:POKEPLT
    +I,C:NEXTI
2021 DATA32,16,194,132
2022 DATA253,32,16,194
2023 DATA152,170,164,253

```

```

2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 DIMTA(3,4)
2035 FORI=1TO3:TA(I,1)=INT(37*RND(0)):TA
(I,2)=1+INT((4+I+I)*RND(0))
2036 TA(I,0)=2*INT(2*RND(0))-1
2037 TA(I,3)=0:TA(I,4)=1+INT(20*RND(0)):
NEXT
2038 DIMSC(15):FORI=0TO15:READSC(I):NEXT
2039 DATA80,50,40,10,100,20,0,30,0,90,0,
0,0,70,60,0
2040 PRINT"{CLR}";
2045 SYSPLOT,0,22:PRINT"{RVS ON}{CYN}
{GRN} {YEL} {RED} {WHT} {LT
BLU} {LT GRN} {BLK} {BRWN}
{PUR} ";
2050 PRINT"{WHT} 10 20 30 40 50 60
70 80 90 10 ";
2051 PRINT"{WHT}{RVS ON}
";
2055 POKE2023,160:POKE56295,1
2060 CO$="{CYN}{GRN}{YEL}{RED}{WHT}{LT B
LU}{LT GRN}{BLK}{BRWN}{PUR}"
2130 PDL=49152:FORI=0TO63:READC:POKEPDL+
I,C:NEXTI
2131 DATA162,1,120,173,2,220,141,0,193,1
69,192,141,2,220,169
2132 DATA128,141,0,220,160,128,234,136,1
6,252,173,25,212,157
2133 DATA1,193,173,26,212,157,3,193,173,
0,220,9,128,141,5,193
2134 DATA169,64,202,16,222,173,0,193,141
,2,220,173,1,220,141
2135 DATA6,193,88,96
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3020 GOSUB3600
3040 NM=NM+1:IFNM<200THEN3020
3500 SYSPDL:P=INT((256-PEEK(PDL+258))/6.
5):IFP=PPTHEN3520
3510 SYSPLOT,PP,21:PRINT"{RVS OFF} "":SY
SPLOT,P,21:PRINT"{RVS ON}{GRY 3} "":PP=P
3520 IFF<>0THEN3540
3525 SYSPDL:IFPEEK(PDL+261)=255THENRETUR
N
3530 F=1:SX=PP:SY=20
3540 SYSPLOT,SX,SY:PRINT"{RVS OFF} "":SY

```



```

=SY-1
3542 SYSPDL: IFPEEK (PDL+261) <> 255 THEN 3530
3545 IFSY < 0 THEN F=0: RETURN
3546 CH=PEEK (1024+SX+40*SY): IFCH <> 32 THEN
3551
3547 SYSLOT, SX, SY: PRINT "{RVS OFF} {WHT} 0
";: RETURN
3551 Z=(SY>12)+(SY>7)+(SY>2): Z=-Z
3552 PT=PT+SC (PEEK (55296+SX+40*SY) AND 15)
*(4-Z)
3553 F=0: TA(Z, 3)=0: TA(Z, 4)=1+INT (20*RND (
0))
3554 TA(Z, 0)=2*INT (2*RND (0))-1: TA(Z, 2)=1
+INT ((4+I+I)*RND (0))
3560 SYSLOT, 8, 24: PRINT "{RVS ON} {WHT} <<<
YOUR SCORE IS "PT">>>";
3565 RETURN
3600 FOR I=1 TO 3
3602 GOSUB 3500
3605 PRINT "{RVS OFF}";: FOR J=I*5-2 TO I*5-1
: SYSLOT, TA(I, 1), J: PRINT " ";: NEXT
3615 TA(I, 1)=TA(I, 1)+TA(I, 0)
3616 TA(I, 3)=TA(I, 3)+1
3617 IFTA(I, 3) <> TA(I, 4) THEN 3621
3618 TA(I, 3)=0: TA(I, 4)=1+INT (20*RND (0)):
TA(I, 0)=2*INT (2*RND (0))-1
3619 TA(I, 2)=1+INT ((4+I+I)*RND (0))
3621 IFTA(I, 1) < 0 THEN TA(I, 1)=36
3622 IFTA(I, 1) > 36 THEN TA(I, 1)=0
3625 PRINT MID$(CO$, TA(I, 2), 1);
3626 PRINT "{RVS ON}";: FOR J=I*5-2 TO I*5-1:
SYSLOT, TA(I, 1), J: PRINT " ";: NEXT
3630 NEXT I: RETURN
3990 RETURN
4000 :
4001 : REM *** END
4002 :
4010 FOR PA=1 TO 500: NEXT
4020 PRINT "{HOME} {WHT} THE GAME IS OVER !
!!"
4030 PRINT "YOUR FINAL SCORE IS" PT
4990 RETURN

```


Numbers Away

A collection of hand-drawn illustrations including two dice, several numbers (1, 2, 3, 4, 5, 6, 7, 8, 9) in various styles (some 3D, some flat), and a banner with the word 'Away' written on it.

This game is a spin-off from a popular game show. The object is to eliminate as many numbers as you can from the list before you get stymied. To begin, you are given a list of numbers ranging from 1 through 9. A pair of dice are rolled. The total (2-12) must be subtracted from the list. Most of the numbers can be removed from the list in a multitude of ways. If the first number rolled is a nine, there are eight possible ways to total exactly nine. They are:

- | | | | | |
|-------------|-------------|---------|---------|-------------|
| 1, 2, and 6 | 2, 3, and 4 | 3 and 6 | 4 and 5 | 9 by itself |
| 1, 3, and 5 | 2 and 7 | | | |
| 1 and 8 | | | | |

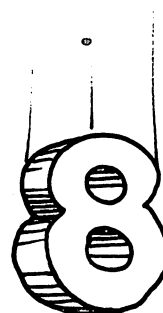
According to the rules, you may remove any of these combinations as long as the total is nine. There are many different strategies, but you can develop your own. Let's look at the program: lines 2125-2169 draw the list of nine numbers. Lines 2340-2392 draw the dice. Once you type the listings into your Commodore, it will be much easier to see what is what, as far as graphics is concerned. Experiment with any lines that you don't understand and note how your changes affect the program overall.

In lines 2340 and following, the dots can be produced with a SHIFT-Q.

```

10 REM *****
11 REM ***          ***
12 REM ** NUMBERS AWAY **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,0:POKE53280,12
1010 PRINT"{CLR}{GRY 3}";
1020 PRINTTAB(10)"*** NUMBERS AWAY ***"
1025 PRINT:PRINT
1030 PRINT"IN THIS GAME, YOU WILL BE PRE
SENTED WITH";
1031 PRINT"A LIST OF NUMBERS BETWEEN 1 A
ND 9."
1032 PRINT
1035 PRINT"A PAIR OF DICE WILL BE ROLLED
, AND THE"
1036 PRINT"TOTAL WILL BE NOTED. YOU MUS
T REMOVE,"
1037 PRINT"FROM THE LIST, A COMBINATION
OF NUMBERS"
1038 PRINT"WHOSE TOTAL MATCHES THE NUMBE
R ON THE DICE."
1040 PRINT:PRINT"FOR EXAMPLE, IF A SEVEN
WAS ROLLED, YOU"
1041 PRINT"COULD REMOVE FROM THE LIST [1
,2,4],"
1042 PRINT"[1,6] [2,5] [3,4] OR JUST PLA
IN [7]."

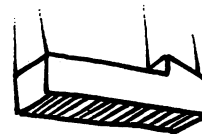
```



```

0 MOVE LEFT";
1056 PRINT"AND F3 TO MOVE RIGHT.  THE NU
MBER YOU"
1057 PRINT"ARE AT WILL BLINK.  TO SELECT
A NUMBER"
1058 PRINT"PRESS F7."
1060 PRINT:PRINT"WHEN YOU SELECT ENOUGH
NUMBERS TO REACH"
1061 PRINT"THE TOTAL ON THE DICE, THE CO
MPUTER WILL";
1062 PRINT"ROLL THE DICE FOR YOUR NEXT T
RY."
1065 PRINT:PRINT"IF YOUR TOTAL GOES OVER
THE NUMBER, THE"
1066 PRINT"LIST WILL BE RESTORED, AND YO
U WILL HAVE";
1067 PRINT"TO TRY AGAIN.  TO GIVE UP, PR
ESS THE '←'KEY."
1070 PRINT:PRINT:PRINT"PRESS RETURN TO C
ONTINUE"
1075 GETANS$: IFANS$<>CHR$(13) THEN1075
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2005 PRINT"{CLR}"
2010 DIMLI(9),L2(9):FORI=1TO9:LI(I)=I:NE
XT
2015 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXTI
2016 POKESID+5,9
2020 PLOT=49152:FORI=0TO24:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 FORI=1TO9:NU=I:GOSUB2100:NEXT
2095 RETURN
2100 PRINTMID$("{RED}{CYN}{PUR}{GRN}{BLU
}{YEL}{LT RED}{ORNG}{LT BLU}",NU,1);:GOT
O2120
2110 PRINT"{BLK}";
2120 ONNUGOTO2125,2130,2135,2140,2145,21
50,2155,2160,2165
2125 SYSPLOT,2,2:PRINT"{RVS ON}  "
2126 SYSPLOT,2,3:PRINT"{RVS ON}{CSR}  "
2127 SYSPLOT,2,4:PRINT"{RVS ON}{CSR}  "

```



```

2128 SYS PLOT,2,5:PRINT" {RVS ON} {CSR} "
2129 SYS PLOT,2,6:PRINT" {RVS ON}      ":RETU
RN
2130 SYS PLOT,6,2:PRINT" {RVS ON}      "
2131 SYS PLOT,6,3:PRINT" {RVS ON} {CSR} {CSR}
} "
2132 SYS PLOT,6,4:PRINT" {RVS ON}      "
2133 SYS PLOT,6,5:PRINT" {RVS ON}      "
2134 SYS PLOT,6,6:PRINT" {RVS ON}      ":RETU
RN
2135 SYS PLOT,10,2:PRINT" {RVS ON}      "
2136 SYS PLOT,10,3:PRINT" {RVS ON} {CSR} {CS
R} "
2137 SYS PLOT,10,4:PRINT" {RVS ON} {CSR}  "
2138 SYS PLOT,10,5:PRINT" {RVS ON} {CSR} {CS
R} "
2139 SYS PLOT,10,6:PRINT" {RVS ON}      ":RET
URN
2140 SYS PLOT,14,2:PRINT" {RVS ON} {CSR} "
2141 SYS PLOT,14,3:PRINT" {RVS ON} {CSR} "
2142 SYS PLOT,14,4:PRINT" {RVS ON}      "
2143 SYS PLOT,14,5:PRINT" {RVS ON} {CSR} {CS
R} "
2144 SYS PLOT,14,6:PRINT" {RVS ON} {CSR} {CS
R} ":RETURN
2145 SYS PLOT,18,2:PRINT" {RVS ON}      "
2146 SYS PLOT,18,3:PRINT" {RVS ON}      "
2147 SYS PLOT,18,4:PRINT" {RVS ON}      "
2148 SYS PLOT,18,5:PRINT" {RVS ON} {CSR} {CS
R} "
2149 SYS PLOT,18,6:PRINT" {RVS ON}      ":RET
URN
2150 SYS PLOT,22,2:PRINT" {RVS ON}      "
2151 SYS PLOT,22,3:PRINT" {RVS ON} {CSR} {C
SR}"
2152 SYS PLOT,22,4:PRINT" {RVS ON}      "
2153 SYS PLOT,22,5:PRINT" {RVS ON} {CSR} "
2154 SYS PLOT,22,6:PRINT" {RVS ON}      ":RET
URN
2155 SYS PLOT,26,2:PRINT" {RVS ON}      "
2156 SYS PLOT,26,3:PRINT" {RVS ON} {CSR} {CS
R} "
2157 SYS PLOT,26,4:PRINT" {RVS ON} {CSR} {CS
R} "
2158 SYS PLOT,26,5:PRINT" {RVS ON} {CSR} {CS
R} "
2159 SYS PLOT,26,6:PRINT" {RVS ON} {CSR} {CS
R} ":RETURN
2160 SYS PLOT,30,2:PRINT" {RVS ON}      "
2161 SYS PLOT,30,3:PRINT" {RVS ON} {CSR} "

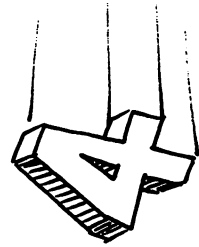
```



```

2162 SYSPLIT,30,4:PRINT" {RVS ON}      "
2163 SYSPLIT,30,5:PRINT" {RVS ON} {CSR} "
2164 SYSPLIT,30,6:PRINT" {RVS ON}      ":RET
URN
2165 SYSPLIT,34,2:PRINT" {RVS ON}      "
2166 SYSPLIT,34,3:PRINT" {RVS ON} {CSR} "
2167 SYSPLIT,34,4:PRINT" {RVS ON}      "
2168 SYSPLIT,34,5:PRINT" {RVS ON} {CSR} {CS
R} "
2169 SYSPLIT,34,6:PRINT" {RVS ON}      ":RET
URN
2200 FORJ=1TO5+INT (5*RND (0) )
2210 D1=1+INT (6*RND (0) ):D2=1+INT (6*RND (0
) )
2220 GOSUB2300:GOSUB2310
2230 NEXT
2300 DD=D1:DX=10:GOTO2320
2310 DD=D2:DX=24
2320 PRINTMID$ (" {RED} {CYN} {PUR} {GRN} {BLU
} {YEL} {LT RED} {ORNG} {LT BLU} ", 1+INT (9*RN
D (0) ), 1);
2330 ONDDGOTO2340,2350,2360,2370,2380,23
90
2340 SYSPLIT,DX,10:PRINT" {RVS ON}      "
2341 SYSPLIT,DX,11:PRINT" {RVS ON}      "
2342 SYSPLIT,DX,12:PRINT" {RVS ON}      "
2343 GOTO2400
2350 SYSPLIT,DX,10:PRINT" {RVS ON}      "
2351 SYSPLIT,DX,11:PRINT" {RVS ON}      "
2352 SYSPLIT,DX,12:PRINT" {RVS ON}      "
2353 GOTO2400
2360 SYSPLIT,DX,10:PRINT" {RVS ON}      "
2361 SYSPLIT,DX,11:PRINT" {RVS ON}      "
2362 SYSPLIT,DX,12:PRINT" {RVS ON}      "
2363 GOTO2400
2370 SYSPLIT,DX,10:PRINT" {RVS ON}      "
2371 SYSPLIT,DX,11:PRINT" {RVS ON}      "
2372 SYSPLIT,DX,12:PRINT" {RVS ON}      "
2373 GOTO2400
2380 SYSPLIT,DX,10:PRINT" {RVS ON}      "
2381 SYSPLIT,DX,11:PRINT" {RVS ON}      "
2382 SYSPLIT,DX,12:PRINT" {RVS ON}      "
2383 GOTO2400
2390 SYSPLIT,DX,10:PRINT" {RVS ON}      "
2391 SYSPLIT,DX,11:PRINT" {RVS ON}      "
2392 SYSPLIT,DX,12:PRINT" {RVS ON}      "
2393 GOTO2400
2400 POKESID+1,90:POKESID+4,33:POKESID+2
4,10:FORPA=1TO80:NEXT
2410 POKESID+4,32:POKESID+1,0

```



```

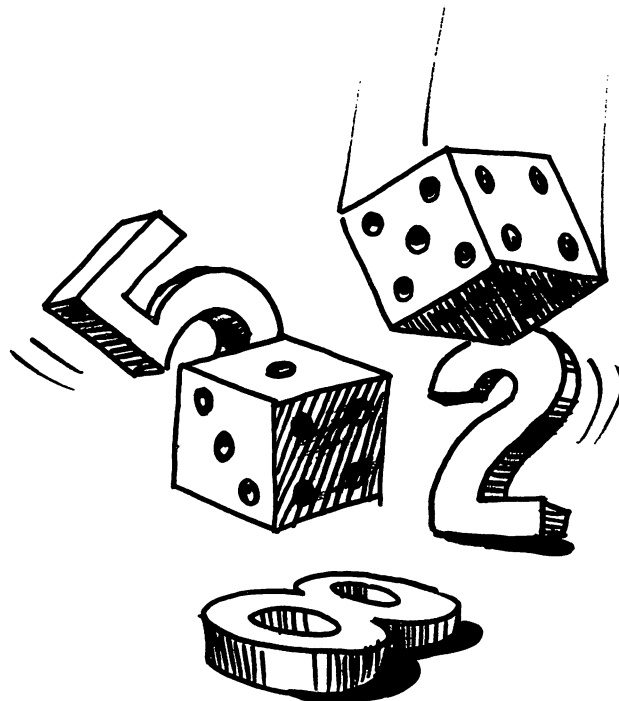
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 FORI=1TO9:IFLI(I)<>0THEN3015
3011 NEXTI:SYSPLOT,0,18:FORI=1TO4:PRINT"
{RVS OFF}
"
3012 SYSPLOT,2,18:PRINT"YOU GOT THEM ALL
!!!":FORPA=1TO1000:NEXT:RETURN
3015 GOSUB2200
3016 FORI=1TO9:L2(I)=0:NEXT
3020 TT=D1+D2:SF=0
3025 SYSPLOT,0,18:FORI=1TO4:PRINT"{RVS O
FF}
"
3026 NEXT
3027 SYSPLOT,2,18:PRINT"{GRY 3}<<< YOU M
UST GET A TOTAL OF"TT">>>"
3030 FORI=1TO9:IFLI(I)<>0THEN3035
3031 NEXTI:SYSPLOT,2,19:PRINT"{GRY 3}THE
RE IS NOTHING LEFT, AND":
3032 PRINT" YOU CANNOT REACH THE TOTAL.
"
3033 FORPA=1TO2000:NEXTPA:RETURN
3035 NP=0:GOSUB3100
3040 NU=LI(NP):GOSUB2110:FORPA=1TO100:NE
XT:GOSUB2100
3050 GETKEY$:IFKEY$=""THEN3040
3060 IFKEY$="{F1}"THENGOSUB3200:GOTO3040
3061 IFKEY$="{F3}"THENGOSUB3100:GOTO3040
3065 IFKEY$="{←}"THENRETURN
3070 IFKEY$"<>"{F7}"THEN3040
3072 NU=NP:GOSUB2110
3075 SF=SF+NP:LI(NP)=0:L2(NP)=1:IFSF>=TT
THEN3080
3076 SYSPLOT,2,19:PRINT"{GRY 3}YOU'VE GO
T"SF"{CSL},YOU NEED "TT-SF"... "":GOT
O3030
3080 IFSF<>TTTHEN3085
3081 SYSPLOT,2,20:PRINT"
"
3082 GT=GT+TT:SYSPLOT,2,20:PRINT"{GRY 3}
YOU GOT THAT ONE !!!"
3083 FORPA=1TO500:NEXTPA:GOTO3010
3085 FORI=1TO9:IFL2(I)<>0THENU=I:GOSUB2
100:LI(I)=I
3090 NEXTI:SF=0
3095 SYSPLOT,2,20:PRINT"{GRY 3}OOOPS, TR
Y AGAIN! YOU NEED"TT"... "

```

```

3096 FORPA=1T0500:NEXTPA
3098 GOT03025
3100 NP=NP+1:IFNP>9THENNP=1
3110 IFLI(NP)=0THEN3100
3120 RETURN
3200 NP=NP-1:IFNP<1THENNP=9
3210 IFLI(NP)=0THEN3200
3220 RETURN
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4005 SYSPLOT,2,21
4020 PRINT"{GRY 3}YOU GOT"GT"OUT OF A POSSIBLE 45."
4022 PRINT" THAT IS ";
4025 ONINT(GT/5)GOTO4040,4041,4042,4043,4044,4045,4046,4047,4048,4049
4040 PRINT"THE ABSOLUTE WORST !!!";:RETURN
4041 PRINT"EXTREMELY POOR !!!";:RETURN
4042 PRINT"TERRIBLE !!!";:RETURN
4043 PRINT"VERY BAD !!!";:RETURN
4044 PRINT"JUST SO-SO !!!";:RETURN
4045 PRINT"FAIR ...";:RETURN
4046 PRINT"PRETTY GOOD ...";:RETURN
4047 PRINT"GREAT !!!";:RETURN
4048 PRINT"FANTASTIC !!!";:RETURN
4049 PRINT"PERFECT !!!!!!";:RETURN
4990 RETURN

```





This one-man paddle game is a test of your dexterity and your ability to think ahead. As opposed to most computer games, the object of Point Attack is to maneuver your paddle so that the flying points do not collide with you. (And as usual, there is a little luck involved as well.) Let's take a look at some of the simple graphics in this program. You'll notice our all-purpose (and now familiar) machine language graphics routine in lines 2005-2026. Lines 2031-2033 draw the border around the playing field using only three lines of code. They do this simply by printing long rows of inverse (black) spaces. This is a good example of getting a maximum of results from a minimum of work, which is also a characteristic of good programming in general. The paddle is drawn in line 3060. Try taking out the reverse video character in the PRINT statement and see what happens!

The ball (as in line 3041) is drawn with a SHIFT-Q.

```

10 REM *****
11 REM ***          ***
12 REM **  POINT ATTACK  **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,12:POKE53280,12
1010 PRINT"{CLR}{BLK}"TAB(10)"*** POINT
ATTACK ***"
1020 PRINT:PRINT:PRINT
1030 PRINT"IN THIS GAME YOU WILL CONTROL
THE MOVE-"
1031 PRINT"MENT OF A PADDLE IN THE MIDL
E OF THE"
1032 PRINT"SCREEN BY PRESSING THE KEYS J
AND K ON THE KEYBOARD."
1033 PRINT:PRINTTAB(8)"J - LEFT      K -
RIGHT"
1040 PRINT:PRINT"THERE WILL BE A FLYING
POINT THAT WILL"
1041 PRINT"BOUNCE AROUND ON THE WALLS.
THE POINT"
1042 PRINT"WILL TRY TO HIT YOU, BUT DON'
T LET IT."
1050 PRINT:PRINT"AFTER AWHILE, ANOTHER P
OINT WILL BE"
1051 PRINT"ADDED TO THE FIELD OF PLAY, A
ND YOUR"
1052 PRINT"PADDLE WILL GROW LARGER.  AVO
ID BEING"
1053 PRINT"HIT FOR AS LONG AS YOU CAN."
1055 PRINT:PRINT"GET THE POINT?"
1060 PRINT:PRINT"PRESS RETURN TO BEGIN P
LAY!"
1065 GETANS$: IFANS$<>CHR$(13) THEN1065
1990 RETURN

```

```

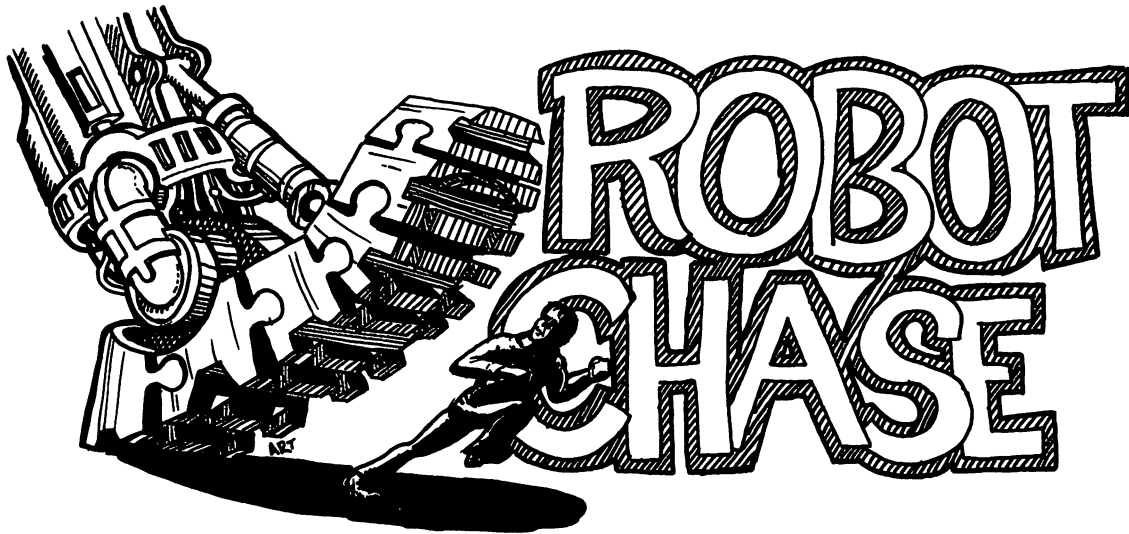
2000 :
2001 :REM *** SETUP
2002 :
2005 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXT:POKESI+5,9
2020 PLOT=49664:FORI=0T024:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,194,132
2022 DATA253,32,16,194
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 PRINT"{CLR}{BLU}";
2031 PRINT"{RVS ON}
";
2032 FORI=1T021:PRINT"{RVS ON} {RVS OFF}
{R
VS ON} {RVS OFF}";:NEXT
2033 PRINT"{RVS ON}
";
2035 DIMBP(10,2),BD(10,2)
2040 FORI=1T010:BP(I,1)=2:BP(I,2)=1+INT(
20*RND(0)):BD(I,1)=1
2045 BD(I,2)=2*INT(2*RND(0))-1
2050 NEXT:X=20
2130 PDL=49152:FORI=0T063:READC:POKEPDL+
I,C:NEXTI
2131 DATA162,1,120,173,2,220,141,0,193,1
69,192,141,2,220,169
2132 DATA128,141,0,220,160,128,234,136,1
6,252,173,25,212,157
2133 DATA1,193,173,26,212,157,3,193,173,
0,220,9,128,141,5,193
2134 DATA169,64,202,16,222,173,0,193,141
,2,220,173,1,220,141
2135 DATA6,193,88,96
2200 POKE650,128:REM ENABLE KEY REPEAT
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3005 PD$=" "
3010 FORH=1T030:PD$=PD$+" ":X=X-1:SYSPLO
T,2,24
3011 PRINT"{BLK}{RVS OFF}<<< SCORE ===>"
H"{CSL} >>>";:FORJ=1T025
3015 I=H:IFI>10THENI=10
3016 FORK=1TOI
3020 XX=BP(K,1):YY=BP(K,2):SYSPLOT,XX,YY

```

```

:PRINT "{RVS OFF} ";
3025 BP(K,1)=XX+BD(K,1):BP(K,2)=YY+BD(K,
2)
3026 XX=BP(K,1):YY=BP(K,2)
3030 IFXX<>1ANDXX<>38THEN3032
3031 BD(K,1)=-BD(K,1):GOSUB3300
3032 IFYY<>1ANDYY<>21THEN3035
3033 BD(K,2)=-BD(K,2):GOSUB3300
3035 IFPEEK(1024+XX+40*YY)=160THENRETURN
3040 PRINTMID$("{BLK}{WHT}{BLU}{YEL}{BLU
}{BRWN}{CYN}{LT GRN}{WHT}{GRY 2}",K,1);
3041 SYSLOT,XX,YY:PRINT "{RVS OFF}#";
3045 IFX<1THENX=1
3050 SYSLOT,X,10:PRINTPD$;
3052 REM SYSPDL:P=PEEK(PDL+257):IFP<128T
HENX=X+1:GOTO3056
3053 REM X=X-1
3054 GETX$:IFX$="J"THENX=X-1
3055 IFX$="K"THENX=X+1
3056 IFX>39-(H+2)THENX=39-(H+2)
3057 IFX<1THENX=1
3060 SYSLOT,X,10:PRINT "{RVS ON}{BLK}"PD
$;
3099 NEXTK:NEXTJ:NEXTH
3100 RETURN
3300 POKESID+1,INT(256*RND(0)):POKESID+2
4,15:POKESID+4,33
3310 FORPA=1TO10:NEXTPA:POKESID+1,0:POKE
SID+4,32:RETURN
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4010 FORPA=1TO500:NEXTPA
4015 PRINT "{CLR}{BLU}":SYSLOT,0,11:PRIN
T"THE GAME IS OVER !!!"
4020 PRINT"YOUR SCORE IS"H", CONGRATULAT
IONS !!!"
4990 RETURN

```



In this game you try to avoid being captured by killer robots. Actually, eluding the relentless robots is most difficult. Because the pursuit is entirely predetermined, it might be a good idea to chart your course out before your first move. Line 2031 merits a closer look:

```
DEF FNR(X)=INT(X*RND(0))+1
```

It uses the Commodore's DEF FN (DEFine FUNction) feature to create a random number generator. The equation on the right side of the equal sign is defined as function R. When a random number in a certain range is needed, as in line 2035, a statement like the following will substitute the value in parentheses for the X in function R:

```
RX=FNR(20) or RX=FNR(10)
```

In our case, FNR(20) will return a random number between one and twenty (since the RND function returns a ten-digit number between 0 and 1) and FNR(10) will return a random number between 1 and 10.

Most of the graphics in this game are pretty straightforward. For an example of the kind of changes you can make to see what's going on in the programs, change the PRINT statement in line 3060 from PRINT "(RVS-OFF)" to PRINT "(RVS-OFF):". Little experiments like these will explain exactly what's happening in each program line.

The club is produced with a SHIFT-X.

The spade is produced with a SHIFT-A.

The circle is produced with a SHIFT-Q.

The plus sign in 1105 is produced with a SHIFT-plus sign.

The arrow in line 3011 is made with SHIFT-C's and a ">" sign.

```

10 REM *****
11 REM ***          ***
12 REM **   ROBOT CHASE   **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53280,0:POKE53281,12
1010 PRINT"{CLR}{BLK}"TAB(10)"*** ROBOT
CHASE ***"
1030 PRINT:PRINT"IN ROBOT CHASE, YOU ARE
AN EXPLORER WHO"
1031 PRINT"HAS LANDED HIS SPACESHIP ON A
HOSTILE PLANET."
1040 PRINT:PRINT"SEVERAL PROTECTOR ROBOT
S ARE TRYING TO"
1041 PRINT"CAPTURE YOU. IF YOU CAN REAC
H A BASE,"
1042 PRINT"YOU WILL BE SAFE BEHIND ITS P
ROTECTIVE FORCE FIELD."
1050 PRINT:PRINT"HERE'S HOW THINGS WORK:
"
1051 PRINT
1052 PRINT" {BLU}BLUE {RVS ON} {RVS OF
F} = AN EXPLOSIVE FENCE (BAD!)"
1053 PRINT" {WHT}WHITE ♣ = YOU"
1054 PRINT" {LT GRN}GREEN ♠ = ATTACKING
ROBOT (BAD!)"
1055 PRINT" {BRWN}ORANGE ♣ = PROTECTIVE
BASE (GOOD!)"
1060 PRINT
1100 PRINT" {BLK}{RVS ON}3 2 1{RVS O
FF} THIS IS YOUR CHOICE"
1105 PRINT" {RVS ON}4 + 8{RVS OFF}
OF MOVEMENT"
1110 PRINT" {RVS ON}5 6 7{RVS OFF}
THROUGH THE MAZE"

```

```

1120 PRINT:PRINT:PRINT:PRINT"PRESS RETURN TO BEGIN PLAY!";
1130 GETANS$:IFANS$<>CHR$(13)THEN1130
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2005 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXT:POKESID+5,9
2020 PLOT=49152:FORI=0T024:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 DIMFI%(21,11)
2031 DEFFNR(X)=INT(X*RND(0))+1
2032 NR=4+FNR(5):DIMRO%(9,2)
2033 NO=INT((NR-4)/2)+1
2034 FORI=1TONR
2035 RX=FNR(20):RY=FNR(10):IFFI%(RX,RY)<
>0THEN2035
2040 FI%(RX,RY)=1:RO%(I,1)=RX:RO%(I,2)=R
Y:NEXTI
2045 FORI=1TONO
2050 OX=FNR(10):OY=FNR(10)
2055 IFFI%(OX,OY)<>0THEN2050
2060 FI%(OX,OY)=2:NEXTI
2070 YX=10+FNR(10):YY=FNR(10)
2075 FI%(YX,YY)=3
2090 FORI=0T021:FI%(I,0)=4:FI%(I,11)=4:N
EXTI
2095 FORI=0T011:FI%(0,I)=4:FI%(21,I)=4:N
EXTI
2096 CO$="{GRY 3}{LT GRN}{BRWN}{WHT}{BLU
}":RV$="{RVS OFF}{RVS OFF}{RVS OFF}{RVS
OFF}{RVS ON}":CH$="  中  "
2100 PRINT"{CLR}"TAB(10){BLU}*** ROBOT
CHASE ***"
2110 FORI=0T021:FORJ=0T011:SYSLOT,I+9,J
+3:K=FI%(I,J)+1
2111 PRINTMID$(CO$,K,1)MID$(RV$,K,1)MID$(
CH$,K,1);
2112 NEXT:NEXT
2200 DIMDI(8,2):FORI=1T08:READDI(I,1),DI
(I,2):NEXT
2210 DATA1,-1,0,-1,-1,-1,-1,0,-1,1,0,1,1
,1,1,0

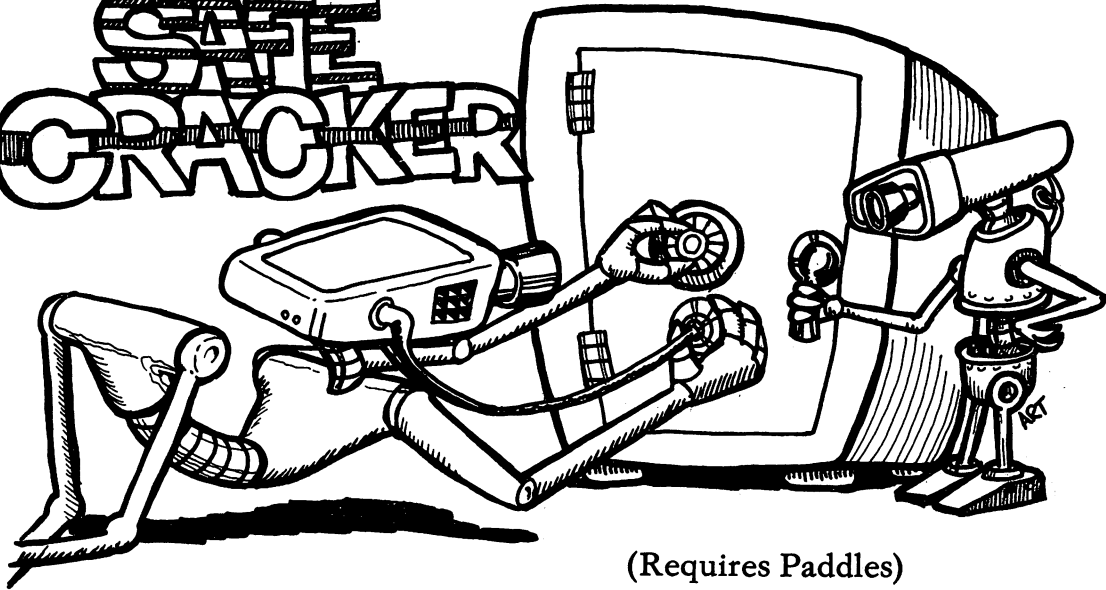
```

```

2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 GOSUB5000:PRINT" {BLK}{RVS ON}3 2 1
":PRINT" {RVS ON}4 + 8"
3011 PRINT" {RVS ON}5 6 7{RVS OFF}      WH
ICH DIRECTION ——> ";
3015 GETANS$:IFANS$=""THEN3015
3016 ANS=VAL(ANS$):IFANS<1ORANS>8THEN301
0
3020 GOSUB5000
3021 X2=YX+DI(AN,1):Y2=YY+DI(AN,2)
3025 SYSPLOT,YX+9,YY+3:PRINT" {RVS OFF} "
;:SYSPLOT,X2+9,Y2+3:PRINT" {WHT}#";
3030 O=FI%(X2,Y2):IFO=4THENWL=0:RETURN:R
EM FENCE
3031 IFO=1THENWL=0:RETURN:REM ROBOT
3032 IFO=2THENWL=1:RETURN:REM BASE
3035 FI%(YX,YY)=0:YX=X2:YY=Y2:FI%(YX,YY)
=3
3040 FORI=1TONR
3045 IFFNR(10)<7THENX2=FNR(3)-2:Y2=FNR(3
)-2:GOTO3055
3050 X2=SGN(YX-RO%(I,1)):Y2=SGN(YY-RO%(I
,2))
3055 X2=X2+RO%(I,1):Y2=Y2+RO%(I,2):O=FI%
(X2,Y2):IFO=1ORO=2ORO=4THEN3045
3060 SYSPLOT,RO%(I,1)+9,RO%(I,2)+3:PRINT
" {RVS OFF} ";:SYSPLOT,X2+9,Y2+3:PRINT" {L
T GRN}#";
3065 IFO=3THENWL=0:RETURN:REM HUMAN
3070 FI%(RO%(I,1),RO%(I,2))=0:RO%(I,1)=X
2:RO%(I,2)=Y2:FI%(X2,Y2)=1
3075 NEXTI:GOTO3010
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4010 GOSUB5000:PRINT" {YEL}THE GAME IS O
VER !!!"
4011 IFWL<>0THENPRINT" YOU'VE BEATEN THE
KILLER ROBOTS (YEAH!)":RETURN
4012 PRINT" THE KILLER ROBOTS GOT YOU !"
4990 RETURN
5000 SYSPLOT,0,18:FORI=1TO4:PRINT" {RVS O
FF}
"
5010 NEXT:SYSPLOT,0,18:RETURN

```


SAFE CRACKER



(Requires Paddles)

In this exciting paddle game, you try to discern the combination of a safe. Your ears, as well as your eyes, are important tools. The object of the game is to open the safe of an enemy agent before the thirty second delayed explosion blows up the safe and you! Once you grasp all of the rules, you will discover that being a safecracker is not too easy. Armed with a sophisticated and sensitive safe-cracking device, you try to detect the numbers in the combination one by one. Each time you pinpoint a number, you turn the paddle the other way until you pinpoint the next number in the combination. When you have identified the entire three-number combination, you will be informed that "THE PAPERS ARE YOURS!!!" and the explosion will be postponed. The safe itself is drawn in lines 2060-2075. The explosion (which you'll probably see quite a few times) is drawn in lines 4016-4020. By the way, if you don't think 30 seconds is enough time to crack the combination, change the value in line 3015 to some number higher than 1800 (add 60 to the number for each additional second you want — e.g., TI\$=2700 would give you 45 seconds).

The horizontal bars are produced with CTRL-C's.

The vertical bars are produced with SHIFT-minus signs.

The circle is produced with a SHIFT-W.

The legs are produced with a SHIFT-English pound sign and a SHIFT-asterisk.

The diamond is produced with SHIFT-Z.

```

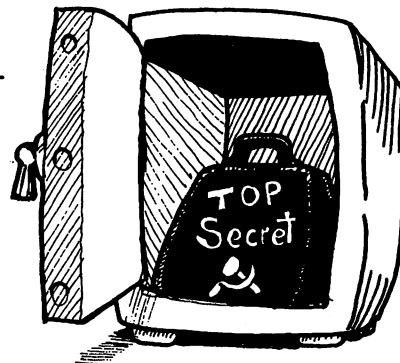
10 REM *****
11 REM ***          ***
12 REM **  SAFE CRACKER  **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,12:POKE53280,12
1010 PRINT"{CLR}"
1020 PRINT:PRINT:PRINTTAB(10)"{BLU}*** S
AFE CRACKER ***"
1030 PRINT:PRINT:PRINT"YOU ARE A GOVERNMENT SPY, AND YOU MUST"
1031 PRINT"RETRIEVE SOME CLASSIFIED DOCUMENTS WHICH";
1032 PRINT"WERE STOLEN BY FOREIGN AGENTS
."
1040 PRINT:PRINT"THE DOCUMENTS ARE KEPT IN A VAULT WHICH"
1041 PRINT"YOU MUST OPEN."
1050 PRINT:PRINT"YOU HAVE BEEN GIVEN A SOPHISTICATED SAFE";
1051 PRINT"CRACKING DEVICE.  WHENEVER THE TUMBLERS"
1052 PRINT"IN A SAFE CLICK INTO PLACE, THE DEVICE"
1053 PRINT"WILL ALSO MAKE A CLICKING SOUND.  IF YOU";
1054 PRINT"EITHER DIRECTLY HIT OR PASS BY A NUMBER"
1055 PRINT"IN THE COMBINATION, THEN THE DEVICE WILL MAKE A CLICK."
1056 PRINT:PRINT"PRESS RETURN WHEN READY TO CONTINUE"
1060 GETANS$:IFANS$<>CHR$(13)THEN1060
1070 PRINT"{CLR}"
1071 PRINT:PRINT:PRINTTAB(10)"{BLU}*** S

```

```

AFE CRACKER ***"
1080 PRINT:PRINT:PRINT"START BY TURNING
THE PADDLE ALL THE WAY"
1081 PRINT"TO THE LEFT (VALUE OF 0). TH
EN MOVE THE";
1082 PRINT"PADDLE TO THE RIGHT UNTIL YOU
GET THE FIRST NUMBER."
1090 PRINT:PRINT"WHEN YOU DISCOVER THE F
IRST NUMBER, THEN TURN TO THE LEFT UNTIL"
;
1091 PRINT" YOU GET THE 2ND NUMBER. ";
1092 PRINT"FINALLY, TURN THE DIAL BACK T
O THE RIGHT";
1093 PRINT" FOR THE THIRD AND LAST NUMBE
R."
1095 PRINT:PRINT"IF YOU GO PAST A NUMBER
, THEN YOU MUST"
1096 PRINT"TURN THE DIAL ALL THE WAY TO
THE LEFT,"
1097 PRINT"AND START OVER.":PRINT
1098 PRINT:PRINT"PRESS RETURN WHEN READY
TO CONTINUE"
1100 GETANS$:IFANS$<>CHR$(13)THEN1100
1110 PRINT"{CLR}"
1111 PRINT:PRINT:PRINTTAB(10)"{BLU}*** S
AFE CRACKER ***"
1120 PRINT:PRINT:PRINT"OH, BY THE WAY, O
NCE YOU HAVE TOUCHED"
1121 PRINT"THE SAFE, YOU WILL HAVE THIRT
Y SECONDS"
1122 PRINT"TO OPEN IT. WHEN THIRTY SECO
NDS HAVE"
1123 PRINT"PASSED, THE SAFETY MECHANISM
WILL CAUSE"
1124 PRINT"A TERRIBLE EXPLOSION."
1130 PRINT:PRINT"PRESS RETURN WHEN READY
TO CONTINUE"
1140 GETANS$:IFANS$<>CHR$(13)THEN1140
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2020 PLOT=49664:FORI=0TO24:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,194,132
2022 DATA253,32,16,194
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177

```



```

2030 N1=1+INT(50*RND(0))
2031 N2=1+INT(50*RND(0)):IFN2>=N1THEN203
0
2032 N3=1+INT(50*RND(0)):IFN3<=N2THEN203
0
2045 PRINT"{CLR}{BLU}";:PRINT:PRINTTAB(1
0)"*** SAFE CRACKER ***"
2050 PRINT:PRINT:PRINT
2060 PRINTTAB(12)"{RVS ON}{BLK}
"
2061 PRINTTAB(12)" {RVS ON}{BLK}
"
2062 PRINTTAB(12)" {RVS ON}{BLK}
"
2063 PRINTTAB(12)" {RVS ON}{BLK} {RVS OF
F}: {RVS ON}
"
2064 PRINTTAB(12)" {RVS ON}{BLK} | {R
VS OFF} {RVS ON} |
"
2065 PRINTTAB(12)" {RVS ON}{BLK} |
"
2066 PRINTTAB(12)" {RVS ON}{BLK} |
"
2067 PRINTTAB(12)" {RVS ON}{BLK} |
"
2068 PRINTTAB(12)" {RVS ON}{BLK} |
"
2069 PRINTTAB(12)" {RVS ON}{BLK} |
"
2071 PRINTTAB(12)" {RVS ON}{BLK} {RVS OF
F}: {RVS ON}
"
2072 PRINTTAB(12)" {RVS ON}{BLK}
"
2073 PRINTTAB(12)" {RVS ON}{BLK}
"
2074 PRINTTAB(12)" {RVS ON}{BLK}
"
2075 PRINTTAB(12)" "
2080 TI$="000000":F=1:D=1
2085 V2=10:H2=19
2130 PDL=49152:FORI=0TO63:READC:POKEPDL+
I,C:NEXTI
2131 DATA162,1,120,173,2,220,141,0,193,1
69,192,141,2,220,169
2132 DATA128,141,0,220,160,128,234,136,1
6,252,173,25,212,157
2133 DATA1,193,173,26,212,157,3,193,173,
0,220,9,128,141,5,193
2134 DATA169,64,202,16,222,173,0,193,141
,2,220,173,1,220,141
2135 DATA6,193,88,96

```

```

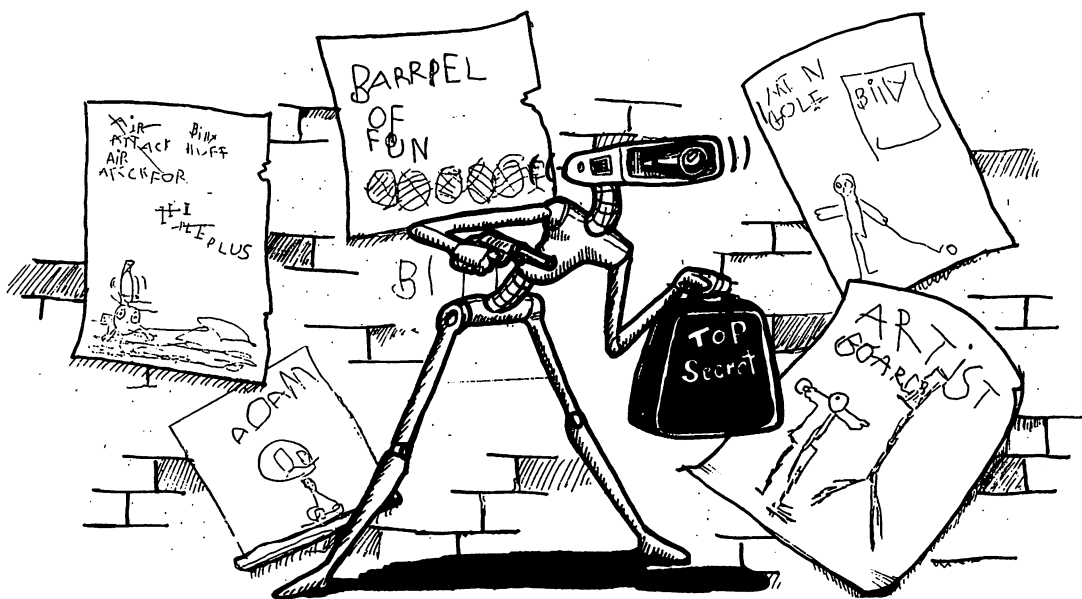
2200 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXT
2210 POKESID+5,88:POKESID+6,195:POKESID+
24,15
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 SYSPLT,16,3:PRINT"<RED>TIME: "RIGHT
$(TI$,2)
3015 IFTI>1800THENWL=0:RETURN
3020 SYSPDL:P=INT((255-PEEK(PDL+257))/5.
12)
3022 SYSPLT,19,9:PRINT"<BLK>"P"<CSL> "
3023 GOSUB3700
3024 IFP>0ANDFTHENSYSPLT,0,9:PRINT"<RVS
OFF>TURN DIAL":PRINT"TO THE LEFT":GOTO30
10
3025 IFP=0ANDFTHENF=0:SYSPLT,0,9:PRINT"<
RVS OFF>          ":PRINT"          ":GO
TO3010
3030 SYSPLT,19,9:PRINT"<RVS OFF><BLK>"P"
<CSL> "
3035 ONDGOTO3040,3050,3060
3040 IFP<N1THEN3010
3041 IFP>N1THENGOSUB3500:GOSUB3600:F=1:G
OTO3010
3042 GOSUB3500:D=2:GOTO3010
3050 IFP>N1THENGOSUB3600:F=1:D=1:GOTO301
0
3054 IFP>N2THEN3010
3056 IFP<N2THENGOSUB3500:GOSUB3600:F=1:G
OTO3010
3058 GOSUB3500:D=3:GOTO3010
3060 IFP<N2THENGOSUB3600:F=1:D=1:GOTO301
0
3064 IFP<N3THEN3010
3066 IFP>N3THENGOSUB3500:GOSUB3600:F=1:G
OTO3010
3068 GOSUB3500:WL=1:RETURN
3500 SYSPLT,30,5+D:PRINT"<LT GRN><CLICK>
"
3505 POKESID+24,15
3510 POKESID+1,20:POKESID+4,33:FORA=1TO2
0:NEXT:POKESID+4,32:POKESID+1,0
3515 POKESID+24,0
3520 RETURN
3600 FORY=1TO3:SYSPLT,30,5+Y:PRINT"<RVS
OFF>          ":NEXT:RETURN
3700 PP=P-4*INT(P/4):DNPP+1GOTO3701,3702

```

```

,3703,3704
3701 V=10:H=20:X$="{BLK}{RVS ON}|":GOTO3
705
3702 V=11:H=21:X$="{BLK}{RVS ON}—":GOTO3
705
3703 V=12:H=20:X$="{BLK}{RVS ON}|":GOTO3
705
3704 V=11:H=19:X$="{BLK}{RVS ON}—"
3705 SYSPLT,H2,V2:PRINT"{RVS ON}{BLK} ";
:SYSPLT,H,V:PRINTX$;:H2=H:V2=V:RETURN
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4010 IFWL=1THEN4040
4015 PRINT"{CLR}";
4016 FORI=1TO10:SYSPLT,0,I:PRINT"{RVS ON
}{RED}*****
*****"
4017 SYSPLT,0,21-I:PRINT"{RVS ON}*****
*****":NEXT
4020 SYSPLT,10,10:PRINT"{BLK}<<< B O O
M ! >>>"
4030 SYSPLT,0,21:PRINT"THE COMBINATION W
AS: "N1"—"N2"—"N3:RETURN
4040 SYSPLT,0,21:PRINT"{BLK}THE PAPERS A
RE YOURS !!!"
4050 PRINT"YOUR COLLEAGUES WILL BE QUITE
IMPRESSED."
4060 PRINT"{HOME}";
4990 RETURN

```





(Requires Paddles)

This is another two-player paddle game. Each player controls the up and down movements of a flying saucer. The object is to shoot down your opponent's ship. The first player to do so three times is the winner of the game. There is not always a direct shot to your opponent, so you may have to blast through a few stars as well! The stars that you blast through are created in lines 3015-3017. The FOR/NEXT loop is run through 30 times to create 30 stars in quasi-random locations on the screen. The X (horizontal) coordinate of a star may be any of the 40 horizontal screen positions. The Y (vertical) coordinate of a star, however, must not get in the way of the two saucers. In line 3015, the statement $Y = 5 + \text{FNR}(13)$ will return a number in the range of five through 17, which means that no stars will be placed in either the very top or very bottom portion of the screen. Remember that the RND function can be manipulated to return a value that is more suitable for your game than a fraction between 0 and 1. For a more complete discussion of the DEF FN command, see the commentary on the game ROBOT CHASE. One more thought: there's no sense in doing more work than you have to, right? That is why the flying saucer drawn in line 3600 is used for both players.

The ship in line 3600 is produced with the following key combinations:

CMDR-8, 1 {CSR}, CTRL-0, SHIFT-U, CMDR-E, SHIFT-1, 1 {CSD}, 5 {CSL}, CTRL-9, SHIFT-English pound sign, 1 space, SHIFT-W, 1 space, SHIFT-W, 1 space, SHIFT-asterisk, 1 {CSD}, 6 {CSL}, CTRL-0, CMDR-U, 1 space, CMDR-U, 1 space, CMDR-U.

The large X is produced with a SHIFT-V.

```

10 REM *****
11 REM ***          ***
12 REM ** SAUCER DUELS **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53280,12:POKE53281,12
1010 PRINT"{CLR}{BLK}"TAB(10)"*** SAUCER
DUELS ***"
1020 PRINT:PRINT
1025 PRINT"THIS IS A TWO-PLAYER GAME. E
ACH PLAYER"
1030 PRINT"CONTROLS A FLYING SAUCER BY U
SING THE PADDLES (PORT 2)."

```



```

NEXT:POKESID+4,9
2020 PLOT=49664:FORI=0T024:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,194,132
2022 DATA253,32,16,194
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2130 PDL=49152:FORI=0T063:READC:POKEPDL+
I,C:NEXTI
2131 DATA162,1,120,173,2,220,141,0,193,1
69,192,141,2,220,169
2132 DATA128,141,0,220,160,128,234,136,1
6,252,173,25,212,157
2133 DATA1,193,173,26,212,157,3,193,173,
0,220,9,128,141,5,193
2134 DATA169,64,202,16,222,173,0,193,141
,2,220,173,1,220,141
2135 DATA6,193,88,96
2200 DEFFNR(X)=INT(X*RND(0))
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3005 F1=0:F2=0:HI=0:L2=18:L4=18
3010 PRINT"{CLR}":POKE53280,0:POKE53281,
0
3015 FORI=1T030:X=FNR(40):Y=5+FNR(13):SY
SPLIT,X,Y
3016 PRINTMID$(" {WHT} {RED} {CYN} {PUR} {GRN
}{BLU} {YEL} {ORNG} {BRWN} {LT RED} {GRY 1} {G
RY 2} {LT GRN} {LT BLU} {GRY 3}",1+FNR(15),
1)"*";
3017 NEXT
3020 SYSPLIT,0,23:PRINT"{RVS ON}{WHT} PL
AYER #1          PLAYER #2 ";
3021 PRINT"
";
3022 POKE2023,160:POKE56295,1
3023 SYSPLIT,4,24:PRINT"{RVS ON}{WHT}"P1
;:SYSPLIT,33,24:PRINT"{RVS ON}{WHT}"P2;
3024 Y=0:GOSUB3600:Y=20:GOSUB3600
3025 PRINT"{HOME}";
3030 GOSUB3500:REM MOVE SHIPS
3040 GOSUB3800:REM MOVE SHOTS
3050 IFHI=0THEN3030
3060 IFHI=1THENP1=P1+1:GOTO3080
3070 IFHI=2THENP2=P2+1
3080 IFP1<3ANDP2<3THEN3005

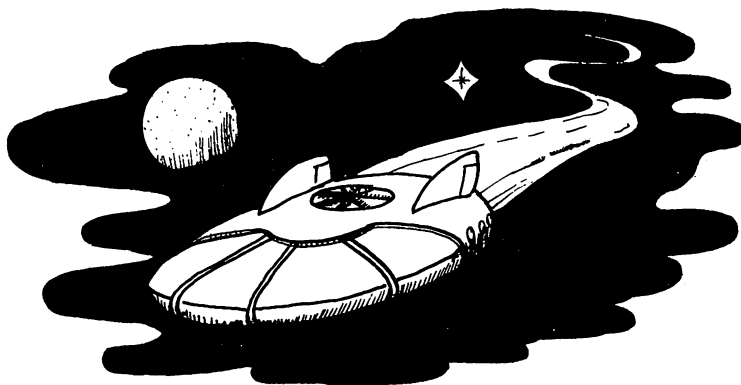
```

```

3090 RETURN
3500 SYSPDL:L1=INT((256-PEEK(PDL+258))/7
.5)
3501 L3=INT((256-PEEK(PDL+260))/7.5)
3505 B1=PEEK(PDL+261)AND4:B2=PEEK(PDL+26
1)AND8
3510 IFL2=L1THEN3550
3520 IFL2<L1THENSYSPLOTT,0,0:FORI=0TO2:PR
INTCHR$(148):NEXT:L2=L2+1:GOTO3550
3530 IFL2>1THENSYSPLOTT,0,0:FORI=0TO2:PRI
NT"{CSR}"CHR$(20):NEXT:L2=L2-1
3550 IFL3=L4THEN3590
3560 IFL4<L3THENSYSPLOTT,0,20:FORI=0TO2:P
RINTCHR$(148):NEXT:L4=L4+1:GOTO3590
3570 IFL4>1THENSYSPLOTT,0,20:FORI=0TO2:PR
INT"{CSR}"CHR$(20):NEXT:L4=L4-1
3590 RETURN
3600 SYSPLOTT,18,Y:PRINT"{GRY 3}{CSR}{RVS
OFF}  ( {CSD}{CSL}{CSL}{CSL}{CSL}{CSL}{R
VS ON}  ( {CSD}{CSL}{CSL}{CSL}{CSL}{C
SL}{CSL}{RVS OFF} - - -":RETURN
3800 IFF1<>0THEN3850
3805 IFB1<>0THEN3850
3810 F1=1:X1=L2+2:Y1=3:SYSPLOTT,X1,Y1:PRI
NT"{RVS OFF}{BLU}#";
3850 IFF2<>0THEN3870
3855 IFB2<>0THEN3870
3860 F2=1:X2=L4+2:Y2=19:SYSPLOTT,X2,Y2:PR
INT"{RVS OFF}{RED}#";
3870 FORI=1TO5
3871 IFF1=0THEN3880
3872 SYSPLOTT,X1,Y1:PRINT"{RVS OFF} ";
3873 Y1=Y1+1:IFY1>22THENF1=0:GOTO3880
3875 CH=PEEK(1024+X1+40*Y1):IFCH=32THENS
YSPLOTT,X1,Y1:PRINT"{RVS OFF}{BLU}#";:GOT
O3880
3876 IFCH=42THENSYSPLOTT,X1,Y1:PRINT"{RVS
OFF} ";:F1=0:GOSUB3900:GOTO3880
3878 XX=L4:YY=20:GOSUB3950:HI=1:RETURN
3880 IFF2=0THEN3890
3882 SYSPLOTT,X2,Y2:PRINT"{RVS OFF} ";
3883 Y2=Y2-1:IFY2<0THENF2=0:GOTO3890
3885 CH=PEEK(1024+X2+40*Y2):IFCH=32THENS
YSPLOTT,X2,Y2:PRINT"{RVS OFF}{RED}#";:GOT
O3890
3886 IFCH=42THENSYSPLOTT,X2,Y2:PRINT"{RVS
OFF} ";:F2=0:GOSUB3900:GOTO3890
3888 XX=L2:YY=0:GOSUB3950:HI=2:RETURN
3890 NEXT:RETURN
3900 POKESID+1,30:POKESID+24,15:POKESID+

```

```
4,33:FORPA=1T030:NEXT
3910 POKESID+4,32:POKESID+1,0:RETURN
3950 PRINT"{CYN}";:FORI=1T012
3955 X=XX+FNR(5):Y=YY+FNR(3):SYSPLOT,X,Y
3960 PRINT"X";
3970 POKESID+1,10:POKESID+24,15:POKESID+
4,33:FORPA=1T050:NEXT
3980 POKESID+4,32:POKESID+1,0:NEXT
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4010 PRINT"{CLR}":SYSPLOT,2,12:PRINT"{GR
Y 3}THE GAME IS OVER!!!"
4020 PRINT"  PLAYER NUMBER ";
4030 IFP1=3THENPRINT"1";
4035 IFP2=3THENPRINT"2";
4040 PRINT" IS THE WINNER !!!"
4990 RETURN
```





True to its name, the object of this game is to dodge the stars until you maneuver your ship near the bottom of the screen. Don't try to intercept the stars. If you do, you'll lose! This program makes use of special Commodore graphics characters to draw the stars and the ship. You may notice that there is not as much complexity in this program as there is in some of the others. List lines 3021-3025. These are the lines responsible for generating stars. Depending on the level of difficulty specified in line 1052, these lines will plot a number of stars on each row of the screen. Line 3021 will pick a random color (LIGHT GREY, BLUE, LIGHT GREEN, PINK, or YELLOW) and line 3025 will pick a random character from a four character string.

Where are all those spacy sounds coming from? LOAD Star Dodger again. (You may have to power down your computer to stop the last tone POKEd into your SID chip.) Before you RUN the program, delete line 3035 by typing in 3035 and pressing the RETURN key. Now if you RUN the program, you'll hear what space really sounds like — an aural vacuum!

The circle is produced with a SHIFT-W.
The diamond is produced with a SHIFT-Z.
The large plus sign is produced with a SHIFT-plus sign.
The large X is produced with a SHIFT-V.

The ship in line 3055 is produced with a CMDR-Q, SHFT-Q, CMDR-W.

The explosion in line 3130 is drawn with 5 asterisks, 5 CMDR-plus signs, 5 periods, 5 CMDR-English pound sign, and 5 CMDR-B's.

```

10 REM *****
11 REM ***          ***
12 REM **  STAR DODGER!  **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1003 POKE650,128:REM SET KEY REPEAT
1005 POKE53281,0:POKE53280,0:PRINT"{GRY
3}";
1010 PRINT"{CLR}":PRINTTAB(11)"*** STAR
DODGER ***"
1030 PRINT:PRINT:PRINT"IN THIS GAME, YOU
WILL BE USING THE KEY-BOARD TO ";
1031 PRINT"PILOT A SPACESHIP.  THE OBJEC
T IS TO MANEUVER THE SHIP ";
1032 PRINT"THROUGH A FIELD OF STARS, AND
TO SUCCESSFULLY REACH ";
1033 PRINT"THE BOTTOM OF THE SCREEN."
1035 PRINT:PRINT"YOU MUST TRY TO AVOID C
OLLIDING WITH THE";
1036 PRINT"STARS.  EACH COLLISION WITH T
HE STARS"
1037 PRINT"DRAINS POWER FROM YOUR SHIP'S
PROTECTIVE";
1038 PRINT"SHIELDS."
1040 PRINT:PRINT"YOU HAVE FIVE ATTEMPTS
TO REACH THE HOME";
1041 PRINT"BASE BEFORE YOUR SHIELDS GIVE
OUT.  AT THIS POINT, THE NEXT ";
1042 PRINT"COLLISION WILL FINDYOUR SHIP
WITH NO POWER AND THE SHIP"
1043 PRINT"WILL BE DESTROYED > BOOM! <"
1044 PRINT
1045 PRINT"USE THE FOLLOWING KEYS TO MAN
EUVER: "
1046 PRINT
1047 PRINT "          A - LEFT          K -
RIGHT":PRINT

```

```

1049 PRINT"PRESS RETURN TO CONTINUE...";
1050 GETX$:IFX$<>CHR$(13)THEN1050
1052 PRINT"{CLR}HOW DO YOU RATE YOURSELF
?":INPUT "1=POOR ... 10=GREAT. ";ANS
1055 IFANS<10RANS>10THEN1050
1990 RETURN
2000 :
2001 :REM *** SETUP
2002 :
2003 POKE53280,0:POKE53281,0
2005 DEFFNJOY(X)=(15-PEEK(56321)AND15)AND
D12
2010 DEFFNR(X)=1+INT(X*RND(O))
2015 DEFFNSC(X)=PEEK(1024+X+SX+40*SY)
2020 NT=5
2025 SID=54272
2026 FORI=SIDTOSID+24:POKEI,0:NEXT
2030 POKESID+5,88:POKESID+6,195
2120 PLOT=49664:FORI=0T024:READC:POKEPLT
+I,C:NEXTI
2121 DATA32,16,194,132
2122 DATA253,32,16,194
2123 DATA152,170,164,253
2124 DATA24,76,240,255
2125 DATA32,253,174,32
2126 DATA158,173,76,170,177
2200 PDL=49152:FORI=0T063:READC:POKEPDL+
I,C:NEXTI
2210 DATA162,1,120,173,2,220,141,0,193,1
69,192,141,2,220,169
2211 DATA128,141,0,220,160,128,234,136,1
6,252,173,25,212,157
2212 DATA1,193,173,26,212,157,3,193,173,
0,220,9,128,141,5,193
2213 DATA169,64,202,16,222,173,0,193,141
,2,220,173,1,220,141
2214 DATA6,193,88,96
2990 RETURN
3000 :
3001 :REM *** PLAY
3002 :
3010 PRINT"{CLR}{GRY 3}":SYSLOT,0,10:IF
NT>1THENPRINTNT"ATTEMPTS REMAIN !!!":GOT
03014
3012 PRINT"1 ATTEMPT REMAINS !!!"
3014 FORI=1T01000:NEXTI
3015 SX=FNR(38):SY=1:PRINT"{CLR}"
3017 X2=SX:Y2=SY
3020 SYSLOT,SX,SY:PRINT" ";SX=X2:SY=
Y2

```

```

3021 FORI=1TOANS/3:PRINTMID$("{GRY 3}{LT
  BLU}{LT GRN}{LT RED}{YEL}",FNR(5),1);
3025 SYSPLIT,FNR(4),24:PRINTMID$("o+x"
  ,FNR(4),1);:NEXT
3030 SYSPLIT,0,24:PRINT"{CSD}{LT BLU}";
3035 POKESID+24,15:POKESID,0:POKESID+1,4
  0+3*FNR(30)
3036 POKESID+4,33:FORI=1TO50:NEXT
3040 IFFNSC(0)<>32ORFNSC(1)<>32ORFNSC(2)
  <>32THEN3100
3041 GETX$
3046 IFX$="K"THENX2=SX+1:GOTO3050
3048 IFX$="A"THENX2=SX-1:GOTO3050
3050 IFX2<1THENX2=1
3051 IFX2>38THENX2=38
3055 SYSPLIT,SX,SY:PRINT" #+ ";:Y2=SY
3060 CNT=CNT+1:IFCNT=8THENCNT=0:Y2=Y2+1:
  IFY2=20THEN3200
3065 FORI=1TO50:NEXT
3070 GOTO3020
3100 SX=SX-2:IFSX<1THENSX=1
3110 SY=SY-2:IFSY<1THENSY=1
3115 POKESID+24,0:POKESID+1,0:POKESID+4,
  32
3120 FORI=1TO5:FORJ=SYTOSY+4:SYSPLIT,X2+
  I-1,J
3130 PRINTMID$("*****.....#####
  #", (I-1)*5+1,5):NEXTJ:FORK=1TO50:NEXT:NE
  XT
3160 NT=NT-1:IFNT>0THEN3010
3200 RETURN
4000 :
4001 :REM *** END
4002 :
4005 POKESID+1,0:POKESID+24,0
4010 PRINT"{CLR}":SYSPLIT,0,10
4020 IFNT=0THENPRINT"I'M SORRY BUT YOU L
  OST...":RETURN
4030 PRINT"YOU WON !!! CONGRATULATIONS !
  !!"
4990 RETURN

```




Grab a friend and get ready for some heated competition. The object of this game is to wall in your opponent and prevent him from being able to move. On each turn you will be asked for the coordinates of the square you wish to move into. Once you have moved, you will also be asked which square you would like to block out. The first player who is unable to make a move loses. This game actually combines some of the better elements of both chess and checkers. The game board itself is drawn on lines 2050-2065. The original player positions and colors are designated in lines 2070 and 2071. You may have noticed that there are a number of GOSUB 5000's in the program. If you look at the REM statement at line 5000, you'll see that this is the subroutine to fill in a square to which a player has moved. See if you can trace the path the program takes to erase the old position, find the new position, then draw the square in. (Hint: look at lines 3060-3070.)

```

10 REM *****
11 REM ***          ***
12 REM **    STRANDED    **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1003 POKE53281,12:POKE53280,6
1005 PRINT"{CLR}{BLU}"TAB(10)"*** STRANDED ***"
1010 PRINT
1020 PRINT"THIS IS A GAME FOR TWO PLAYERS. BOTH OF";
1021 PRINT"YOU WILL BE PLACED IN AN 8 X 8 MATRIX."
1030 PRINT"A PLAYER MAY MOVE IN ANY OF THE EIGHT"
1031 PRINT"DIRECTIONS. ENTER THE COORDINATES OF THE TARGET SQUARE."
1035 PRINT
1040 PRINT"AFTER YOU MOVE, YOU WILL BE ASKED FOR"
1041 PRINT"THE COORDINATES OF A SECOND SQUARE. THIS";
1042 PRINT"SQUARE WILL THEN BE BLACKED OUT, AND"
1043 PRINT"WILL BE UNENTERABLE."
1045 PRINT
1050 PRINT"THE FIRST PLAYER UNABLE TO MOVE LOSES"
1051 PRINT"THE GAME. THE TOP LEFT IS (1,1) AND THE";
1052 PRINT"TOP RIGHT IS (1,8).";
1055 PRINT:PRINT:PRINT"PRESS RETURN TO BEGIN PLAY!"
1060 GETANS$:IFANS$<>CHR$(13)THEN1060
1990 RETURN

```

```

2000 :
2001 :REM *** SETUP
2002 :
2005 SID=54272:FORI=SIDTOSID+24:POKEI,0:
NEXT:POKESID+5,9
2020 PLOT=49152:FORI=0T024:READC:POKEPLT
+I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2030 DEFFNC(X)=(X-1)*3+9:DEFFNR(X)=(X-1)
*2+2
2040 DIMM(8,8),P(2,3)
2041 M(4,1)=1
2042 M(5,8)=2
2043 P(1,1)=4
2044 P(1,2)=1
2045 P(2,2)=5
2046 P(2,3)=8
2050 PRINT"{CLR}":FORI=1T017:SYSPLOT,8,I
:PRINT"{RVS ON}{WHT}
"
2055 NEXTI
2056 SYSPLOT,9,0
2057 PRINT"{RVS OFF}{LT GRN} 1 2 3 4
5 6 7 8"
2058 SYSPLOT,8,1
2059 PRINT"{RED}{RVS ON}
"
2060 FORI=2T017STEP2:SYSPLOT,5,I
2061 PRINT"{BLK}"I/2"{RED}{RVS ON} {CSR}
{CSR} {CSR}{CSR} {CSR}{CSR} {CSR}{CSR} {
CSR}{CSR} {CSR}{CSR} {CSR}{CSR} {CSR}{CS
R} "
2062 SYSPLOT,8,I+1
2063 PRINT"{RED}{RVS ON}
"
2065 NEXT
2070 PRINT"{YEL}";:X=4:Y=1:GOSUB5000
2071 PRINT"{BLU}";:X=5:Y=8:GOSUB5000
2080 DIMDIR(8,2):FORI=1T08:READDIR(I,1),
DIR(I,2):NEXT
2090 DATA1,1,-1,-1,1,-1,-1,1,1,0,0,1,-1,
0,0,-1
2990 RETURN
3000 :
3001 :REM *** PLAY

```

```

3002 :
3010 FORP=1TO2
3020 GOSUB6100:PRINT"{WHT}*** PLAYER #"P
:INPUT"MOVE TO ({BLK}ROW,{LT GRN}COLUMN(
WHT})";X1,Y1
3025 IFX1<1ORX1>8ORY1<1ORY1>8ORX1<>INT(X
1)ORY1<>INT(Y1)THENGOSUB6000:GOTO3020
3030 X=P(P,P):Y=P(P,P+1)
3040 IF(X1-X)*(X1-X)+(Y1-Y)*(Y1-Y)>=4THE
NGOSUB6020:GOTO3020
3050 IFM(X1,Y1)<>0THENGOSUB6020:GOTO3020
3060 M(X,Y)=0:PRINT"{WHT}";:GOSUB5000
3070 M(X1,Y1)=P:PRINTMID$("{YEL}{BLU}",P
,1);:P(P,P)=X1:P(P,P+1)=Y1:X=X1:Y=Y1:GOS
UB5000
3100 GOSUB6100:INPUT"{WHT}BLOCK AT ({BLK
}ROW,{LT GRN}COLUMN(WHT))";X2,Y2
3105 IFX2<1ORX2>8ORY2<1ORY2>8ORX2<>INT(X
2)ORY2<>INT(Y2)THENGOSUB6000:GOTO3100
3110 IFM(X2,Y2)<>0THENGOSUB6020:GOTO3100
3120 M(X2,Y2)=3:PRINT"{BLK}";:X=X2:Y=Y2:
GOSUB5000
3200 K=3-P:FORJ=1TO8:X4=P(K,K)+DI(J,1):Y
4=P(K,K+1)+DI(J,2)
3210 IFX4<1ORX4>8THEN3250
3211 IFY4<1ORY4>8THEN3250
3220 IFM(X4,Y4)<>0THEN3250
3230 J=8:NEXTJ:GOTO3990
3250 NEXTJ:RETURN
3990 NEXT
3995 GOTO3010
4000 :
4001 :REM *** END
4002 :
4010 GOSUB6100
4020 PRINT"{WHT}THE GAME IS OVER.....":P
RINT"PLAYER"P"IS THE WINNER!"
4990 RETURN
5000 REM *** FILL IN A SQUARE
5001 SYSPLOT,FNC(Y),FNR(X):PRINT"{RVS ON
} ";:RETURN
6000 REM *** SHOW ERROR THEN PAUSE
6010 PRINT"COORDINATES MUST BE INTEGER B
ETWEEN 1 AND 8...":GOTO6050
6020 PRINT"ILLEGAL MOVE..."
6050 FORPA=1TO400:NEXTPA
6100 REM *** CLEAR BOTTOM OF SCREEN
6105 SYSPLOT,0,19:FORI=1TO5:PRINT"{RVS O
FF} "":
NEXT
6110 SYSPLOT,0,19:RETURN

```



This game requires good timing. A small moving target is the object of your marksman talent. The angle at which the ball is shot depends on the angle of the firing pad. Unlike most games, where the gameboard and other starting graphics are drawn in the subroutine beginning at line 2000, in Target, ALL of the graphics are drawn in the subroutine starting at line 3000. Line 3010 clears the screen and lines 3011-3014 draw a black border around the edge of the screen. Take a close look at line 3030. Unlike most FOR/NEXT loops, which are numbered from 1 to some higher number, the FOR/NEXT loop in this line goes from -2 to 2. This allows us to find the center point of the paddle (XP and YP) and plot two points to either side of it. To see exactly what is going on, change the -I to +I or the +I to -I. The paddle might still look the same, but it will soon become apparent that the game won't play the same way!

The ball (dot) is drawn with a SHIFT-Q.

```

10 REM *****
11 REM ***          ***
12 REM **    TARGET    **
13 REM ***          ***
14 REM *****
15 :
16 :
20 GOSUB1000:REM INSTRUCTIONS
30 GOSUB2000:REM SETUP
40 GOSUB3000:REM PLAY!
50 GOSUB4000:REM !END!
60 END
70 :
1000 :
1001 :REM *** INSTRUCTIONS
1002 :
1005 POKE53281,12:POKE53280,12
1010 PRINT"{CLR}{BLK}"TAB(13)"*** TARGET
    ***"
1025 PRINT:PRINT
1030 PRINT"IN THIS GAME YOU TRY TO HIT A
    MOVING"
1031 PRINT"TARGET.  BY PRESSING ANY KEY,
    A SMALL"
1032 PRINT"BALL WILL BE FIRED FROM THE P
    ADDLE ON  THE SCREEN."
1040 PRINT:PRINT"YOU MUST TIME THE RELEA
    SE SUCH THAT THE"
1041 PRINT"SMALL BALL HITS THE LARGE TAR
    GET.  THE"
1042 PRINT"DIRECTION AND DISTANCE WILL V
    ARY WITH"
1043 PRINT"EACH NEW TARGET. "
1050 PRINT:PRINT"PRESS RETURN TO BEGIN P
    LAY. "
1090 GETANS$:IFANS$<>CHR$(13)THEN1090
1990 RETURN
2000 PLOT=49152:FORI=0TO24:READC:POKEPLT
    +I,C:NEXTI
2021 DATA32,16,192,132
2022 DATA253,32,16,192
2023 DATA152,170,164,253
2024 DATA24,76,240,255
2025 DATA32,253,174,32
2026 DATA158,173,76,170,177
2990 RETURN

```

```

3000 :
3001 :REM *** PLAY
3002 :
3005 C=0
3010 PRINT "{CLR}";
3011 PRINT "{RVS ON} {BLU}
";
3012 FOR I=1 TO 21:PRINT "{RVS ON} {RVS OFF}
(R
VS ON} {RVS OFF}";
3013 NEXT
3014 PRINT "{RVS ON} {BLU}
";
3015 SYS PLOT, 10, 23:PRINT "{RVS OFF} {BLK} S
CORE: "SC" {CSL} SHOTS: "SH" {CSL} {HOME
}";
3020 IF SH=0 THEN 3025
3021 SYS PLOT, 14, 24:PRINT "{BLK} PERCENT: "
INT(SC/SH*100)" {CSL} % ";
3025 PM=2*INT(2*RND(0))-1:XP=10+INT(10*R
ND(0)):YP=7+INT(5*RND(0))
3026 SP=PM*INT(3*RND(0)):FLAG=0
3030 PRINT "{BLK} {RVS ON}";:FOR I=-2 TO 2:SY
S PLOT, XP-I*SP, YP+I:PRINT " ";:NEXT
3035 X2=25+INT(10*RND(0)):Y2=1:S2=1+INT(
2*RND(0)):X3=X2:Y3=Y2
3040 PRINT "{RVS OFF}";:FOR I=Y3 TO Y3+1:SYS
PLOT, X3-1, I:PRINT " ";:NEXT
3041 PRINT "{RVS ON} {LT GRN}";:FOR I=Y2 TO Y
2+1:SYS PLOT, X2-1, I:PRINT " ";:NEXT
3045 X3=X2:Y3=Y2:IF FL THEN 3070
3050 GETKEY$:IF KEY$="" THEN 3600
3055 FL=1:B1=XP+1:B2=YP:B3=B1:B4=B2:SH=S
H+1
3060 SYS PLOT, 10, 23:PRINT "{RVS OFF} {BLK} S
CORE: "SC" {CSL} SHOTS: "SH" {CSL} {HOME
}";
3065 SYS PLOT, 14, 24:PRINT "{BLK} PERCENT: "
INT(SC/SH*100)" {CSL} % ";
3066 S=PEEK(1024+B3+40*B4)
3067 IF B1<>B3 AND S<>B1 THEN 3095
3070 IF S=160 THEN 3095
3075 SYS PLOT, B3, B4:PRINT "{RVS OFF} ";:SY
S PLOT, B1, B2:PRINT "{WHT}#";:B3=B1:B4=B2:B
1=B1+1:B2=B2+SP
3080 IF B2<10 OR B2>20 THEN B2=B4:SP=-SP:GOTO 3
600
3085 IF B1>37 THEN SYS PLOT, B3, B4:PRINT "{RVS
OFF} ";:FL=0:SP=ABS(SP)*PM:GOTO 3600
3090 IF PEEK(1024+B1+40*B2)=32 THEN 3600

```

```
3095 SC=SC+1:GOTO3010
3600 C=C+1:IFC<>2THEN3045
3605 C=0:Y2=Y2+1:IFY2>20THENY2=1
3610 GOTO3040
3990 RETURN
4000 :
4001 :REM *** END
4002 :
4990 RETURN
```


GAMES COMMODORES PLAY

ALL THE GAMES IN THIS BOOK ON DISK

!!! Save hours of typing and de-bugging !!!
Only \$15.95 with this coupon.
Available ONLY to readers of this book.

- * Spend your time playing the games and writing your own versions.
- * Use the sound routines and graphics in your own programs.
- * Games may be copied, listed, and modified.
- * Let your fingers do the playing, not the typing.



Please RUSH me the GAMES COMMODORES PLAY diskette for \$15.95. Please include \$1.75 for shipping and handling. California Residents add 6.5% sales tax (\$1.04).

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE () _____

MASTER CARD VISA CHECK

CREDIT CARD NUMBER _____ EXPIRES _____

SIGNATURE (for charge cards) _____

 DATAMOST™

20660 Nordhoff St., Chatsworth, CA 91311-6152
(818) 709-1202

GAMES COMMODORES PLAY

WHAT BETTER WAY TO LEARN PROGRAMMING THAN BY PLAYING GAMES?

The authors have compiled a selection of classic computer games along with many clever new games. Written in Commodore BASIC, they are formatted in a way which lets you easily adapt any of the routines to your own programs. Each game is explained in simple, straightforward terms — allowing you to modify and customize to your heart's content!

You'll learn principles of text formatting, word games, data statements and input routines. Many of the games will help you understand how grids are constructed and graphics animated. Helpful techniques are given so you can dissect BASIC programs and learn what makes them tick.

You'll have hours of fun learning to program BASIC in this easy, enjoyable way. And when you're done, you'll have the games to play! Whoever thought that gaming could teach so much about programming and debugging?



ISBN 0-88190-121-0



20660 Nordhoff Street, Chatsworth, CA 91311-6152
(818) 709-1202