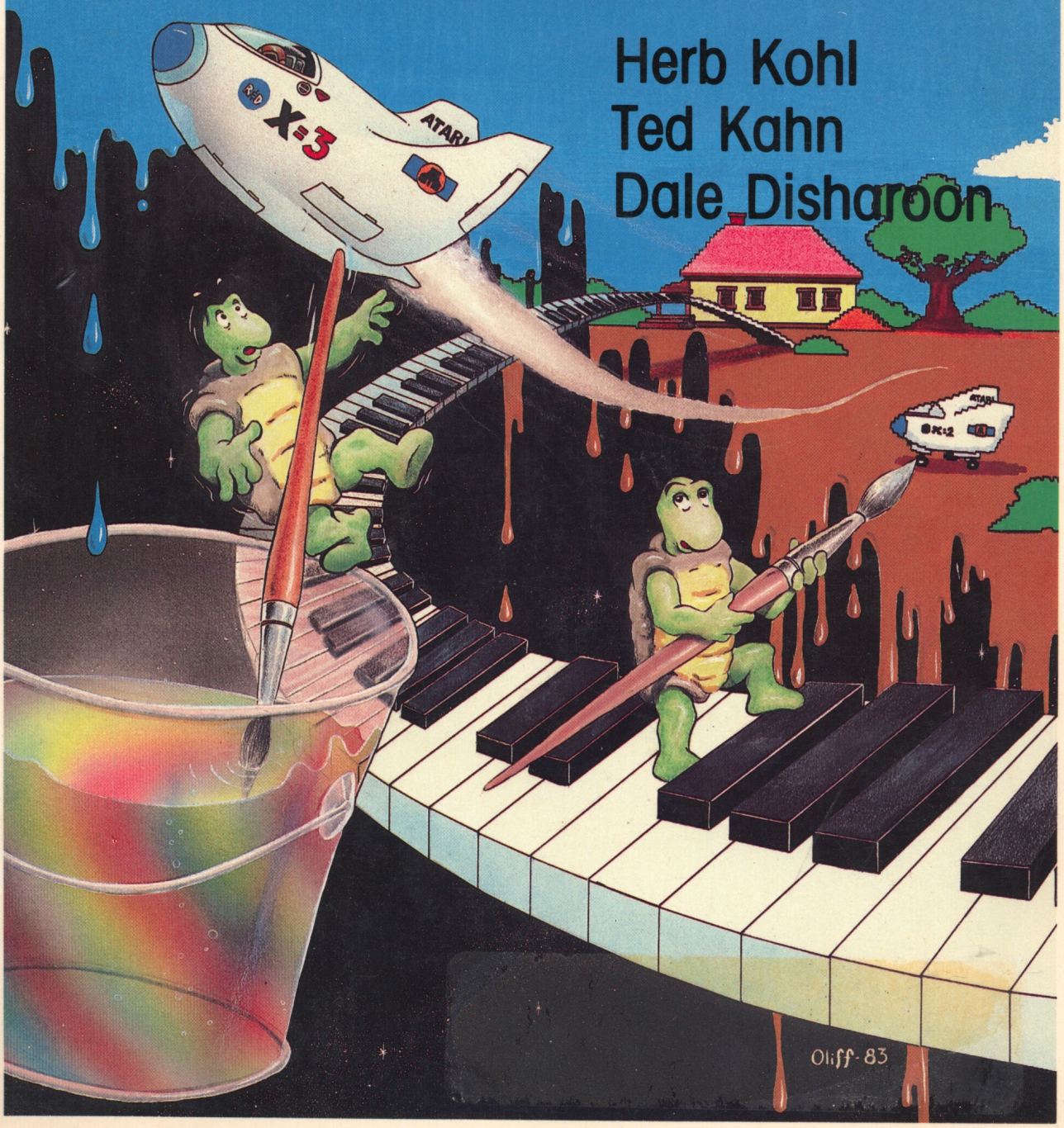
ATARI® PILOT
Activities and Games

Herb Kohl
Ted Kahn
Dale Disharoon



ATARI® PILOT Activities and Games

Herb Kohl
Ted Kahn
Dale Disharoon

Illustrated by Steve Oliff

A Reston Computer Group Book
Reston Publishing Company, Inc.
A Prentice-Hall Company
Reston, Virginia

Library of Congress Cataloging in Publication Data

Kohl, Herbert R.
Atari PILOT activities and games.

Includes index.
1. Atari 400 (Computer)—Programming. 2. Atari 800 (Computer)—Programming. 3. PILOT (Computer program language) 4. Computer games. I. Kahn, Ted.
II. Disharoon, Dale. III. Title.
QA76.8.A8K64 1983 001.64'2 83-11113
ISBN 0-8359-0321-4

All of the programs in this book have been tested on the ATARI 400 XL and 800 XL Home Computers. It is the author's understanding that these programs will work on any ATARI Computer.

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Contents

Preface

PILOT (Programmed Inquiry Learning Or Teaching) is a simple, but powerful, programming language developed in the 1960s by Dr. John Starkweather (University of California Medical Center, San Francisco). Designed specifically for education, the primary audience was teachers who wanted to develop computer-based education programs. However, because of PILOT's simplicity, it was found to be an excellent introductory programming language for children as well as adults. Dr. Dean Brown, then head of SRI's Educational Technology Laboratory, began using PILOT with elementary school children in the late 1960s and found that children as young as six and seven loved to write their own PILOT programs. Dean Brown's work also convinced the Lawrence Hall of Science's Computer Education Project (Bob and Ted Kahn, Pete Rowe, Keith Vann, and others) to use PILOT as the main language for working with elementary-school-aged children.

The development of microcomputers allowed new and wonderful worlds to merge graphic images and music. Dr. Seymour Papert (MIT), one of the inventors of the Logo programming language, developed the use of computers as "microworlds of exploration." He developed a concept called "turtle graphics," an elegant mathematical system which is easy enough for any child to use to create beautiful dynamic works of art.

When the Atari Corporation developed its own version of PILOT for ATARI Home Computers, it took the best of the ideas of each of these outstanding educators. Harry Stewart, who implemented ATARI PILOT, included the turtle graphics system from Logo and the standard X-Y coordinate graphics, and he added a sound command to allow PILOT to make music. Dave Thornburg offered very helpful advice about some important additions to PILOT's commands. Bob and Ted Kahn helped develop manuals and tutorial materials, as well as examples of programs in PILOT which would help users see the versatility of this language. The result is an outstand-

ing programming language, one which can be used by both children and adults for entertainment and education. We hope *ATARI® PILOT Activities and Games* will help show ATARI Computer users how versatile PILOT is and that it will also help develop ideas of "modular programming." We suggest that this book be used in conjunction with the two manuals which are available with the purchase of ATARI PILOT: *The ATARI 400/800™ PILOT Primer* and *ATARI 400/800™ Student PILOT: Reference Guide.*

"Deeper meaning often lies in child's play."
—Schiller

ATARI® PILOT Activities and Games

1 PILOT: A Simple Language That Can Do Complex Things

Introductory Notes

ATARI® PILOT is a simple language to master and is therefore very useful for teachers and other people, young and old, who are new to programming. Also, the ATARI PILOT cartridge contains "turtle graphics," which makes it easy to master some of the powerful aspects of ATARI Computer graphics, including drawing and animation.

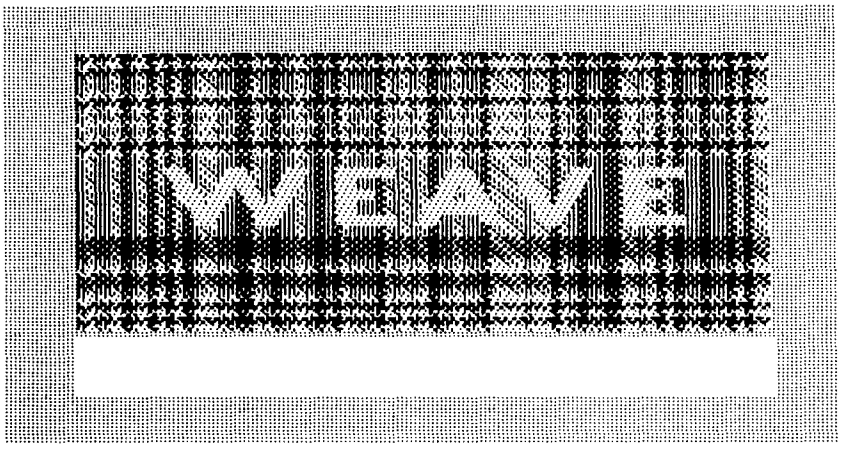
The simplicity of PILOT, however, does not mean that you can do only simple programming in the language. In fact PILOT is structured so that the programmer can design quite complex programs, add new structures (called modules) to the language, and use them in any number of different programs. We will show how to use PILOT to build complex programs throughout this book as well as provide dozens of games, both educational and recreational, that you can copy and use on any ATARI Computer with the ATARI PILOT cartridge.

However, we will assume that you have not had much experience with PILOT. If you have already mastered PILOT, you can skip the introductory section and begin to sample some of the programs in the book. We have provided language, math, and logic games as well as music and graphics programs. There is also a section on miniworlds and adventure games, where you can discover how PILOT can be used to develop adventures and simulations. We have also included explanations of how to examine and change the contents of certain memory locations (equivalent to ATARI BASIC PEEK and POKE commands) so that PILOT can take advantage of some of the features of the special chips in the ATARI Computer.

In addition to explaining programs and giving their listings (for our listing notation, please turn to Appendix I—you won't need it for awhile, and we'll let beginners know when

*ATARI is a registered trademark of Atari, Inc. (a Warner Communications Company), used by permission.

they need to use it), we have made screen dumps of most programs in the book. A *screen dump* is a computer printout of exactly what you see on the screen when you use (RUN) a program. It is not a listing of the program but a picture of the program in action. For example, see the illustration of a screen dump of the introduction to a PILOT program, included in this book, that allows you to weave your own patterns. These screen dumps are included to help you visualize what the programs in the book do.



In addition to providing these illustrations, we have included suggestions on how you can modify your programs or dress them up. It is important to take control of programming as well as use other people's programs, and we encourage you to become a creative and playful programmer. PILOT is particularly suited to getting first experiences in programming, as you will see.

Preliminary Commands and Techniques

You should know a few simple commands and techniques before you begin programming. In PILOT, like most other computer languages, each line of a program must have a line number if it is to be stored in the computer's memory. In PILOT any integers from 0 to 9999 can be used as line numbers. No matter what order you type your lines in, the computer reorders them from smallest to largest and uses them in that order. Thus if you number your lines

0 _____
99 _____
6 _____
2 _____
5 _____

the computer will reorder and use them in this sequence:

0 _____
2 _____
5 _____
6 _____
99 _____

Two PILOT operating commands need to be memorized: RUN and LIST. You type LIST when you want to see all the numbered lines that make up a program. You type RUN when you want to use a program. After typing either LIST or RUN, you press [RETURN] to see the program list or use the program. Finally you type NEW and press [RETURN] if you want to erase a program from your computer's memory.

The Core of PILOT

In PILOT a number of simple commands can be combined to make complex and interesting programs. In this section we will build a simple adventure game from a core PILOT vocabulary and show how to modify that game to make it more complex and interesting without using additional vocabulary. If you follow the development of this game and type it into the ATARI Computer as it grows, you should develop a feel for the ease of programming in PILOT. We'll begin with the following three comands:

T:
A:
M:

T: instructs the computer to "Type" or print anything that comes after it, including spaces on the screen, as the following example indicates:

```
10 T:GUESS WHAT THE NEXT SENTENCE SAYS
20 T: GUESSWHATTHENEXTSENTENCESAYS
```

When you type RUN and press [RETURN] the screen will look like this:

```
GUESS WHAT THE NEXT SENTENCE SAYS
 GUESSWHATTHENEXTSENTENCESAYS

READY
```

If you want to space out the two sentences or any two T: lines in your program, you simply add any empty T: where you want the line skipped. Then your program would look like this:

```
10 T:GUESS WHAT THE NEXT SENTENCE SAYS
15 T:
20 T: GUESSWHATTHENEXTSENTENCESAYS
```

The screen dump shows what the empty T: would do to the program when you RUN it.

```
GUESS WHAT THE NEXT SENTENCE SAYS
 GUESSWHATTHENEXTSENTENCESAYS

READY
```

You'll notice throughout the programs in this book many empty T: statements. These are used to make the program more readable and visually elegant when it appears on the screen.

The next two commands are A: (the Accept or PILOT input command) and M: (the Match command), which is unique to PILOT. A: lets the user put a number, letter, or word into the computer's memory. M: lets you match (compare) that number, letter, or word with a list that you type after M:.

Here are several of the infinite number of match lists that you could create while inventing games, puzzles, or other programs:

```
M: 2,4,6,8,10
M: 1,3,5,7,9
M: A,E,I,O,U
M: LION, TIGER, CAT, CHEETAH
```

Notice that there *must* be a comma between each item in a match (M:) statement.

When you input something into an A: statement followed by an M: statement, you will either have a match or you won't.

- Y indicates there is a match.
- N indicates there is no match.

The Y and N match features are unique characteristics of PILOT. Understanding how they work should give you a feel for some of the structural power of PILOT. Y and N are never found in programs by themselves. They are used in conjunction with another command that tells you what to do if there is (or isn't) a match. T: is one of the commands with which they can be used. TY: tells the computer what to print if there is a match. TN: tells it what to print if there is no match to the M: statement. Here are two examples that should clarify how Y and N work:

```
10 T:HOW OLD ARE YOU
20 A:
30 M:1,2,3,4,5
40 TY:ITS GOOD TO BE YOUNG
50 TN:SORRY YOU'RE SO OLD ALREADY
```

```
10 T:HOW ARE YOU FEELING?
20 A:
30 M:GOOD,FINE,OK,WONDERFUL
40 TY:GLAD TO HEAR IT
50 TN:HOPE YOU FEEL BETTER TOMORROW
```

In the first example, if you type 1, 2, 3, 4, or 5 in the A: line, you will have a match; the program will go to line 40 TY: and print out IT'S GOOD TO BE YOUNG. If you type any other number, the program will go to line 50 TN: and print out:

SORRY YOU'RE SO OLD ALREADY.

Try to analyze how the second program works and write a few programs using T:, M:, A:, TN:, and TY:.

A summary follows of these core PILOT ideas to refer to in case you need to refresh yourself on how they are used.

```
          THE CORE OF PILOT:1

Commands          Meaning

   T:             Type or print

   A:             Accept as input
                  a number or word

   M:             Match what is put
                  in A: with a list
                  of words or numbers
                  put after M: and
                  separated by commas.

READY
```

```
THE CORE OF PILOT:II

Conditionals          Meaning

   Y                  This is used if
                      a valid match is
                      made.
                      It can be used with
                      T: in the form TY:

   TY:                If a match is made
                      then whatever follows
                      TY: is printed

READY
```

```
THE CORE OF PILOT:III

Conditionals          Meaning

   N                  This is used if
                      n0 valid match can
                      be made.
                      It can be used with
                      T: in the form TN:

   TN:                If no match is made
                      then whatever follows
                      TN: is printed

READY
```

Building Our Adventure

With a few more simple bits of PILOT, we will be able to build our adventure. Let's begin by taking a look at A:, the PILOT input command. A: can accept numbers (actually only positive and negative integers), letters, and words. Sometimes A: is used a number of times in a program, and the kind of input that it receives needs to be named and used in different parts of a program. The way to name variables that are input into a program is to use the following:

- *For letters and words:* A:$ followed by a name such as A:$ANIMAL or A:$CAR or A:$NAME
- *For numbers:* A:# followed by a single letter such as A:#X or A:#B or A:#C

Two short programs illustrate how variables are used with the A: command.

```
10 T:WHAT IS THE LARGEST CAT?
20 A:$CAT
30 M:LION
40 TY:RIGHT, $CAT IS THE LARGEST CAT
50 TN:SORRY IT IS NOT A $CAT
```

In the first program the variable name is $CAT. Notice that $CAT appears on both lines 40 and 50. If you answer the

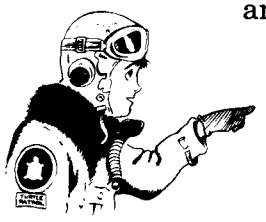
question on line 10 by typing lion, you will get a match; the computer will go to line 40 and type out

RIGHT, LION IS THE LARGEST CAT

If on the other hand you type in tiger, there will be no match; the computer will go to the TN: statement on line 50 and type

SORRY IT IS NOT A TIGER

Notice that $CAT functioned just like a word in the TY: and TN: statements. The only difference is that it is a variable and therefore it becomes whatever you type in on line 20.



Here's a short number variable program:

```
10 T:WHAT NUMBER IS HALF WAY BETWEEN
20 T:10 AND 20?
30 A:#G
40 M:15
50 TY:YOU ARE RIGHT
60 TN:SORRY #G IS NOT CORRECT
```

If you type in 15, there will be a match and the computer will type out

YOU ARE RIGHT

If you type any other number, such as 22 or 17, there will be no match; the computer will print at 60

SORRY 22 IS NOT CORRECT

or SORRY 17 IS NOT CORRECT

Here are two other simple programs using variables. Copy and RUN them to get a sense of how variables work in PILOT; then change them and experiment with your own programs, using the core PILOT vocabulary introduced so far.

```
10 T:WHAT IS THE SMALLEST BIRD?
20 A:$BIRD
30 M:HUMMINGBIRD
40 TY:RIGHT, $BIRD IS THE SMALLEST BIRD
50 TN:SORRY,$BIRD IS NOT THE SMALLEST BIRD
```

```
10 T:WHAT NUMBER IS HALF OF TWENTY?
20 A:#X
30 M:10
40 TY:YOU ARE RIGHT
50 TN:NO #X IS NOT HALF OF 20
```

```
THE CORE OF PILOT:IV

Variables          Meaning

  A:$NAME          This is used when
                   you name a letter
                   or word input
                   variable.

  A:#X             This is used when
                   you name a number
                   variable. The name
                   should be one letter.

READY
```

Now we can start on the adventure. As it develops, new PILOT vocabulary will be introduced. By the time we've finished the miniadventure, you should have most of the core vocabulary of PILOT mastered and be able to do many programs of your own. In addition to providing dozens of programs that you can copy and use, we'll introduce you to programming computation, sound, graphics, and animation in ATARI PILOT.

We'll begin the adventure by asking the player for his or her name and finding out whether he or she wants to play:

```
10 T:WHAT IS YOUR NAME?
20 A:$NAME
25 T:
30 T:$NAME DO YOU WANT AN ADVENTURE?
40 A:$ANSWER
50 M:YES
60 TY:WELL HERE WE GO
70 TN:SORRY MAYBE SOME OTHER TIME
80 E:
```

Notice that two variables, $NAME and $ANSWER, have been used, each referring to a different match statement. Also note a new PILOT command; E: on line 80 follows the statement SORRY MAYBE SOME OTHER TIME. E: is a command that ends a program (or, as we'll see later, part of a program). If a person doesn't want to play, the program stops.

In another look at the program, if someone wants to play, we must jump into the game; that should take place right after the computer prints WELL HERE WE GO, that is, between lines 60 and 70.

```
10 T:WHAT IS YOUR NAME?
20 A:$NAME
25 T:
30 T:$NAME DO YOU WANT AN ADVENTURE?
40 A:$ANSWER
50 M:YES
60 TY:WELL HERE WE GO
```

Jump in

```
70 TN:SORRY MAYBE SOME OTHER TIME
80 E:
```

That's exactly what happens in PILOT. You can jump from one part of a program to another using the jump command J: or the jump command with a match statement:

- JY: (jump if there is a match)
- JN: (jump if there is no match)

Y and N function with J: in the same way that they function with T:.

Using Modules

In many computer languages the rough equivalent of J: refers to a line number. For example, in BASIC the GOTO command instructs the computer to go to the line number written after the command. GOTO 100 means go to line 100 in the program and do whatever it says. *There are no references to line numbers within programs in PILOT.* Instead there are *modules* that you create and name. These modules are like new words or functions that you create for your programs. You can use them repeatedly at different parts in your program. With programming in PILOT one must learn how to *think modules.* As you'll see soon, if you can read the modules in a program and see how they are used, you can quickly understand long and seemingly complex programs.

Each module has a name that is preceded by an asterisk (the multiplication sign on the ATARI Computer keyboard). You can name your modules in any way that you want by using numbers and upper-case letters as long as they do not have any spaces in them. Thus possible module names could be: *FRIEND, *FRIENDSHIP, or *2FRIENDS, but *not* *FRIEND SHIP.

THE CORE OF PILOT:V

Commands	Meaning
E:	End a program
*MODULE	Name a module
J:*MODULE	Jump to *MODULE
JY:*MODULE	Jump to *MODULE if there is a match
JN:*MODULE	Jump to *MODULE if there is no match

■

This short program should give you a good idea of how to read a PILOT program that uses modules and jumps. It is a good idea to skim a PILOT program listing, look for the modules that it contains, and then return to the beginning lines and see how they are used. In many of the programs in this book we'll indicate the modules by placing a bracket to the left of their listing, as in this program.

```
10 T:WHICH COLOR DO YOU LIKE BETTER
20 T:
30 T:RED OR YELLOW?
40 J:*GUESS
50 *GUESS
60 A:
70 M:RED
80 JY:*RED
90 M:YELLOW
100 JY:*YELLOW
110 *RED
120 T:YOU HAVE A HOT TEMPER
130 E:
140 *YELLOW
150 T:YOU ARE QUITE INSIGHTFUL
160 E:
```

This program asks you whether you prefer red or yellow and then jumps to either a red answer (*RED) or a yellow answer (*YELLOW). The beginning lines introduce the program. It would be easy to add more colors by simply creating new modules such as *BLUE or *PURPLE or *ORANGE. One advantage of PILOT is that it is easy to build up complexity by adding simple modules to a program.

Let's return to our monster adventure, which we begin with this introduction:

```
10 T:WHAT IS YOUR NAME?
20 A:$NAME
25 T:
30 T:$NAME DO YOU WANT AN ADVENTURE?
40 A:$ANSWER
50 M:YES
60 TY:WELL HERE WE GO
70 TN:SORRY MAYBE SOME OTHER TIME
80 E:
```

Notice that we have added at line 65 a command to jump on a match to a module *MONSTER. That module will start us on our adventure.

We'll begin the module on line 100:

```
100 *MONSTER
```

This module, like all PILOT modules, can be either simple or complex and can deal with words, numbers, music, graphics, or animation. One can begin with an extremely simple module and make it more complex as you have new ideas. Throughout this book you'll see modules of varying complexity, and we'll give hints on how to distinguish modules and modify them.

A good programming trick that often comes in handy at the beginning of a module is to clear the screen so that it won't look cluttered and too full of print. You can clear the screen at any point of a program by writing the following command:

T: [ESCAPE] [SHIFT CLEAR]

What this means is that you type T:, then press the [ESCAPE] key, followed by pressing the [SHIFT] and [CLEAR] keys simultaneously. What you will see on the screen is

T:↱

In our program notations we will represent T:↱ by the processes that you must perform. According to the notation in Appendix I, T: ↱ will be printed

T: [CLEAR]

The module looks like this so far:

```
100 *MONSTER
110 T:[CLEAR]
```

Now we must decide what we want the monster to say or do. If you know how to use PILOT sound and graphics, you could start with mysterious music and the face of your monster. However, we'll start simple here, though it may be a good idea for you to return to this program after you've read the sound and graphics chapter and make *MONSTER fancy.

```
120 T:YOU ARE TAKING A WALK ONE DAY AND
130 T:
140 T:MEET A MONSTER. IT COMES UP TO YOU
150 T:
160 T:AND BREATHES SMOKE IN YOUR FACE. IT
170 T:
180 T:SAYS,"WHAT ARE YOU DOING BOTHERING"
190 T:
200 T:ME?"
210 T:
220 T:WHAT DO YOU DO?
230 J:*VOCAB
```

Putting the parts of the program together, here's what we have so far:

```
10 T:WHAT IS YOUR NAME?
20 A:$NAME
25 T:
30 T:$NAME DO YOU WANT AN ADVENTURE?
40 A:$ANSWER
50 M:YES
60 TY:WELL HERE WE GO
65 JY:*MONSTER
70 TN:SORRY.  MAYBE SOME OTHER TIME.
80 E:
100 *MONSTER
110 T:
120 T:YOU ARE TAKING A WALK ONE DAY AND
130 T:
140 T:MEET A MONSTER. IT COMES UP TO YOU
150 T:
160 T:AND BREATHES SMOKE IN YOUR FACE. IT
170 T:
180 T:SAYS,"WHAT ARE YOU DOING BOTHERING"
190 T:
200 T:ME?"
210 T:
220 T:WHAT DO YOU DO?
```

At this point we must decide on the vocabulary to be allowed in the adventure. We could choose any number of words, but for the sake of simplicity we will use three and let the player know what they are. To do this we'll create a vocabulary module called *VOCABULARY and jump to that module at line 230:

```
230 J:*VOCABULARY
```

Here's the module using the words CRY, SMILE, and FROWN. Notice that each of these three words leads to a module of its own: *CRY, *SMILE, and *FROWN. In that way the program uses matches and jumps to build up the whole structure.

```
710 *VOCAB
720 T:
730 T:  HERE ARE THE WORDS YOU CAN USE
740 T:
750 T:  CRY      SMILE      FROWN
760 T:
770 T:  WHAT WILL YOU DO?
780 A:
790 M:CRY
800 T:[CLEAR]
810 JY:*CRY
820 M:SMILE
```

```
830 T:[CLEAR]
840 JY:*SMILE
850 M:FROWN
860 T:[CLEAR]
870 JY:*FROWN
```

At this point the program on your screen would look like the following screen dump.

```
JOSHUA DO YOU WANT AN ADVENTURE?
YES
WELL HERE WE GO
YOU ARE TAKING A WALK ONE DAY AND
MEET A MONSTER. IT COMES UP TO YOU
AND BREATHES SMOKE IN YOUR FACE. IT
SAYS,"WHAT ARE YOU DOING BOTHERING"
ME?"
WHAT DO YOU DO?
  HERE ARE THE WORDS YOU CAN USE
  CRY      SMILE      FROWN          ■
  WHAT WILL YOU DO?
```

Here are the modules for the three words that we chose:

```
230 J:*VOCAB
240 *SMILE
250 T:
260 T:
270 T:
280 T:YOUR SMILE HAS MELTED MY HEART
290 T:
300 T:I WILL LET YOU START THE ADVENTURE
310 T:
320 T:AGAIN
330 J:*MONSTER

340 *FROWN
350 T:
360 T:
370 T:
380 T:YOUR FROWN HAS ANGERED ME.
390 T:YOU BETTER RUN OR I'LL EAT YOU.
400 T:I AM KINDER THAN YOU THINK.
410 T:YOU CAN START ALL OVER AGAIN.
420 J:*MONSTER

430 *CRY
440 T:
450 T:
460 T:
470 T:I AM A FEROCIOUS MONSTER....BUT
480 T:
490 T:BUT I CAN'T STAND CRYING...HERE
500 T:
510 T:TAKE MY TREASURE.
520 J:*TREASURE
```

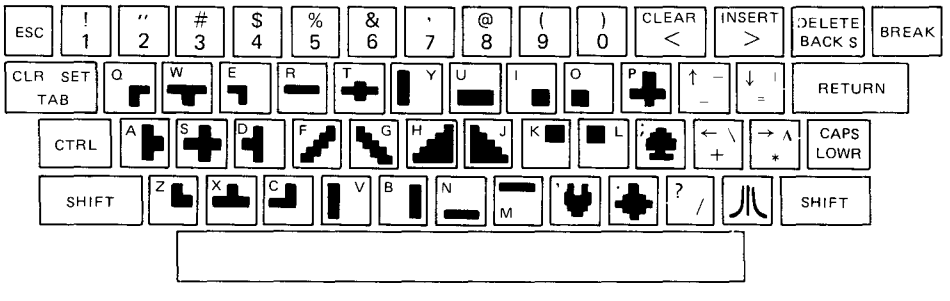
Notice that if you decide to smile or frown, you jump back to *MONSTER, but if you decide to cry, you go to the last module in the program, *TREASURE, and get the treasure. You can make the treasure anything that you want. Here's an empty treasure module with lots of T: commands left blank for you to fill in with words or with ATARI Computer character graphics, sound, or turtle graphics, all of which you'll learn about in this book:

```
530 *TREASURE
540 T:
550 T:
560 T:
570 T:
580 T:
590 T:
600 T:
610 T:
620 T:
630 T:
640 T:
650 T:TO START AGAIN TYPE HELLO
660 A:$HELLO
670 M:HELLO
680 T:[CLEAR]
690 JY:*MONSTER
700 E:
```

Here's a simple treasure that we drew using the control graphics keyboard on the ATARI Computer. All the graphics on the keyboard can be incorporated into your programs, which makes it easy to be fancy.

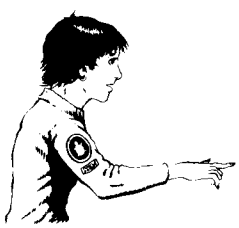
```
530 *TREASURE
540 T:
550 T:
560 T:
570 T:
580 T:
590 T:
600 T:
610 T:
620 T:
630 T:
640 T:
650 T:TO START AGAIN TYPE HELLO
660 A:$HELLO
670 M:HELLO
680 T:[CLEAR]
690 JY:*MONSTER
700 E:
```

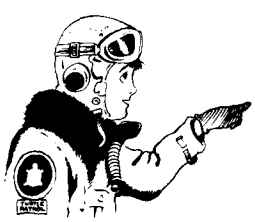
The following drawing illustrates the graphics characters that you can use simply by pressing CONTROL, that is [CTRL], at the same time as pressing the indicated keys.



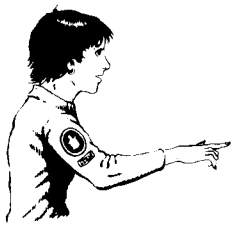
Notice lines 650 to 690 in *TREASURE. They give the option of replaying the game by typing the word HELLO.

Putting all the pieces together, here is how the program looks. Notice that, though it's very long, if you look at each module as a unit, you can easily figure out how the program works. Modules are like words that you've created; in fact the power of PILOT is that by building your own modules you are creating your own language for the computer to use.





```
10 T:WHAT IS YOUR NAME?
20 A:$NAME
25 T:
30 T:$NAME DO YOU WANT AN ADVENTURE?
40 A:$ANSWER
50 M:YES
60 TY:WELL HERE WE GO
65 JY:*MONSTER
70 TN:SORRY.  MAYBE SOME OTHER TIME.
80 E:
100 *MONSTER
110 T:
120 T:YOU ARE TAKING A WALK ONE DAY AND
130 T:
140 T:MEET A MONSTER. IT COMES UP TO YOU
150 T:
160 T:AND BREATHES SMOKE IN YOUR FACE. IT
170 T:
180 T:SAYS,"WHAT ARE YOU DOING BOTHERING"
190 T:
200 T:ME?"
210 T:
220 T:WHAT DO YOU DO?
230 J:*VOCAB
240 *SMILE
250 T:
260 T:
270 T:
280 T:YOUR SMILE HAS MELTED MY HEART
290 T:
300 T:I WILL LET YOU START THE ADVENTURE
310 T:
320 T:AGAIN.
330 J:*MONSTER
340 *FROWN
350 T:
360 T:
370 T:
380 T:YOUR FROWN HAS ANGERED ME.
390 T:YOU BETTER RUN OR I'LL EAT YOU.
400 T:I AM KINDER THAN YOU THINK.
410 T:YOU CAN START ALL OVER AGAIN.
420 J:*MONSTER
```



```
430 *CRY
440 T:
450 T:
460 T:
470 T:I AM A FEROCIOUS MONSTER....BUT
480 T:
490 T:BUT I CAN'T STAND CRYING...HERE
500 T:
510 T:TAKE MY TREASURE.
520 J:*TREASURE
530 *TREASURE
540 T:
550 T:
560 T:
570 T:
580 T:
590 T:
600 T:
610 T:
620 T:
630 T:
640 T:
650 T:TO START AGAIN TYPE HELLO
660 A:$HELLO
670 M:HELLO
680 T:[CLEAR]
690 JY:*MONSTER
700 E:
710 *VOCAB
720 T:
730 T:  HERE ARE THE WORDS YOU CAN USE
740 T:
750 T:  CRY     SMILE     FROWN
760 T:
770 T:  WHAT WILL YOU DO?
780 A:
790 M:CRY
800 T:[CLEAR]
810 JY:*CRY
820 M:SMILE
830 T:[CLEAR]
840 JY:*SMILE
850 M:FROWN
860 T:[CLEAR]
870 JY:*FROWN
```

Try to read this program and figure out how it works. Then try to turn it into a space adventure or pirate adventure. Change the vocabulary or add to it. You will find it easy to make your own simple adventures. Throughout the rest of the book we'll show how to add to the core PILOT ideas presented here and make your own games and adventures. We'll also give you dozens of games and other programs that you can copy, modify, and use. It would be a good idea to try to read and understand the programs before copying them so that you see how the module structure works.

More Core Ideas

There are a few more core PILOT ideas to master. Then we'll be ready to plunge into graphics, sound, and color. These are the remark statement R: and the use module U:. The remark R: statement is used to help you read and understand the structure of a program. It is *not* really a part of the program and does not affect it in any way. Instead it is a programmer's way to mark out blocks in a program and help others understand how the program works. Here is a simple program that we used before, with R: statements added.

```
10 R:COLOR CHOICE PROGRAM
20 T:WHICH COLOR DO YOU LIKE BETTER
30 T:
40 T:RED OR YELLOW?
50 R:INPUT OF COLOR CHOICE
60 T:
70 A:
80 M:RED
90 JY:*RED
100 M:YELLOW
110 JY:*YELLOW
120 R:RED MODULE BEGINS HERE
130 *RED
140 T:YOU HAVE A HOT TEMPER
150 E:
160 R:YELLOW MODULE BEGINS HERE
170 *YELLOW
180 T:YOU ARE QUITE INSIGHTFUL
190 E:
```

Finally there is the U:, or use, command. This is very convenient when a module is used many times in a program. It is essential that *each module referred to by a U: command is ended with an E: command.* If not, your program will run on to the lines after the module and might become hopelessly confused. Here's a version of the color choice program with an additional module called *SUMMARY, which is used at the end of the program regardless of the color chosen.

```
170 *SUMMARY
180 T:SEE HOW COLOR AFFECTS PERSONALITY?
190 T:ADD MORE COLORS TO THIS PROGRAM.
200 T:THINK ABOUT COLOR AND PERSONALITY.
210 E:
```

Here's how it can work, taking advantage of the power of the U: command.

```
10 T:WHICH COLOR DO YOU LIKE BETTER
20 T:
30 T:RED OR YELLOW?
40 A:
50 M:RED
60 UY:*RED
70 M:YELLOW
80 UY:*YELLOW
90 U:*SUMMARY
100 E:
110 *RED
120 T:YOU HAVE A HOT TEMPER
130 E:
140 *YELLOW
150 T:YOU ARE QUITE INSIGHTFUL
160 E:
170 *SUMMARY
180 T:SEE HOW COLOR AFFECTS PERSONALITY?
190 T:ADD MORE COLORS TO THIS PROGRAM.
200 T:THINK ABOUT COLOR AND PERSONALITY.
210 E:
```

Notice that at line 60 there is a UY: command, which functions like a JY: command. It instructs the computer to use * RED if there is a match, just as line 80 says UY:*YELLOW and instructs the computer to use the module *YELLOW.

At the end of each module is an E: command. The E: command must end each module referred to by the use command U:. After the module is used, the program jumps back to the line after the command and continues to run. The E: at the end of the module in effect returns the program to the line after the U: statement. Look at the arrows in the following program. The first arrow shows how the program uses the *RED module. When that is done, the program returns to line 90 (since there is no match with YELLOW). Line 90 then directs the computer to use the module *SUMMARY. After it is used, the program returns to line 100 and ends. This might not be clear the first time that you study it but try to play with U: statements, as they make it possible to use a module many times within a program and therefore save much time and effort when you're trying to create complex programs.

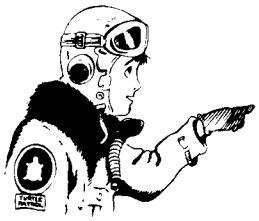
```
10 T:WHICH COLOR DO YOU LIKE BETTER
20 T:
30 T:RED OR YELLOW?
40 A:
50 M:RED
```

```
60 UY:*RED
70 M:YELLOW
80 UY:*YELLOW
90 U:*SUMMARY
100 E:
110 *RED
120 T:YOU HAVE A HOT TEMPER
130 E:
140 *YELLOW
150 T:YOU ARE QUITE INSIGHTFUL
160 E:
170 *SUMMARY
180 T:SEE HOW COLOR AFFECTS PERSONALITY?
190 T:ADD MORE COLORS TO THIS PROGRAM.
200 T:THINK ABOUT COLOR AND PERSONALITY.
210 E:
```

1 2 3

Here's a version of our monster adventure that takes advantage of the U: command. After you've studied (and hopefully modified) it, you're ready to venture into ATARI PILOT sound and graphics.

```
10 T:WHAT IS YOUR NAME?
20 A:$NAME
30 T:
40 T:$NAME DO YOU WANT AN ADVENTURE?
50 A:$ANSWER
60 M:YES
70 TY:WELL HERE WE GO
80 JY:*MONSTER
90 TN:SORRY.  MAYBE SOME OTHER TIME.
100 E:
110 *MONSTER
120 T:
130 T:YOU ARE TAKING A WALK ONE DAY AND
140 T:
150 T:MEET A MONSTER.  IT COMES UP TO YOU
160 T:
170 T:AND BREATHES SMOKE IN YOU FACE.  IT
180 T:
190 T:SAYS, "WHAT ARE YOU DOING BOTHERING
200 T:
210 T:ME?"
220 T:
230 T:WHAT DO YOU DO?
240 U:*VOCAB
250 T:[CLEAR]
260 M:CRY
270 JY:*CRY
280 M:SMILE
290 UY:*SMILE
300 M:FROWN
310 UY:*FROWN
320 J:*MONSTER
```

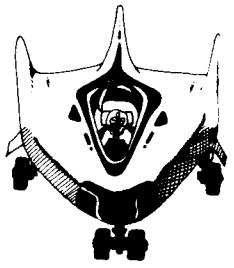


```
330 *SMILE
340 T:
350 T:
360 T:
370 T:YOUR SMILE HAS MELTED MY HEART
380 T:
390 T:I WILL LET YOU START THE ADVENTURE
400 T:
410 T:AGAIN.
420 E:
```

```
430 *FROWN
440 T:
450 T:
460 T:
470 T:YOUR FROWN HAS ANGERED ME.
480 T:YOU BETTER RUN OR I'LL EAT YOU.
490 T:I AM KINDER THAN YOU THINK.
500 T:YOU CAN START ALL OVER AGAIN.
510 E:
```

```
520 *CRY
530 T:
540 T:
550 T:
560 T:I AM A FEROCIOUS MONSTER....BUT
570 T:
580 T:BUT I CAN'T STAND CRYING...HERE
590 T:
600 T:TAKE MY TREASURE.
610 U:*TREASURE
620 T:TO START AGAIN TYPE HELLO
630 A:$HELLO
640 M:HELLO
650 T:[CLEAR]
660 JY:*MONSTER
670 E:
```

```
680 *TREASURE
690 T:
700 T:
710 T:
720 T:
730 T:
740 T:
750 T:
760 T:
770 T:
780 T:
790 T:
800 T:
810 E:
```



```
820 *VOCAB
830 T:
840 T:  HERE ARE THE WORDS YOU CAN USE
850 T:
860 T:  CRY     SMILE     FROWN
870 T:
880 T:  WHAT WILL YOU DO?
890 A:
900 E:
```

A summary of U: and R: follows.

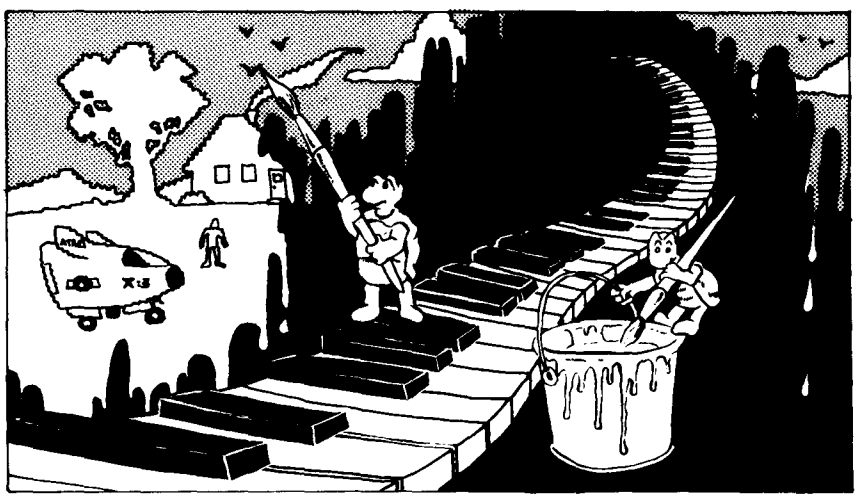
```
THE CORE OF PILOT:UI

R:              The remark command
                lets you label your
                programs for your own
                convenience. It is not
                part of the program.

U:*NAME         The USE command makes
                your program use a
                specific module (in
                this case *NAME) and
                then return and go on
                with the rest of the
                program.
UY:*NAME        Use a module if there
                is a match.
UN:*NAME        Use if there is no
                match.
(UY: & UN: are similar to JY: & JN:)
```

At this point all the core commands in PILOT have been presented and illustrated. A summary of them follows. If any puzzle you, go back over the chapter and try to develop programs of your own similar to the ones that we have presented. The best way to become comfortable with programming is by creating your own programs.

Summary of the Core of PILOT

T:	E:
A:	*MODULE
M:	J:*MODULE
Y	JY:*MODULE
TY:	JN:*MODULE
N	R:
TN:	U:*NAME
A:$NAME	UY:*NAME
A:#X	UN:*Name

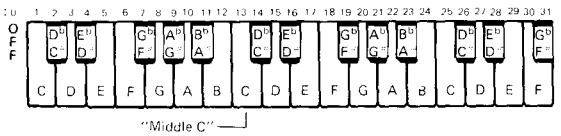
2 PILOT Sound and Graphics



ATARI PILOT Sound

Sound and music are very easy to create with ATARI PILOT. The sound command is simply SO:. SO: lets you create musical tones on your computer. You can program up to four tones to play at the same time. Each separate tone is then called a *voice.* When people get together to sing, they divided into several *voices:* bass, baritone, alto, and soprano. When you program SO: commands, you can program several voices to play together. When different voices play a sound at the same time, that sound is called a *chord.*

The tones that you can play in PILOT range from 0 to 31. If you enter the command SO:13, you will hear a single tone equal to middle C on the piano. The command SO:1 produces a tone equal to the note C, one octave below middle C. The command SO:31 produces the note F sharp, one octave above middle C. SO:0 turns a tone off. The following diagram shows a piano keyboard with the key numbers corresponding to the PILOT sound commands.



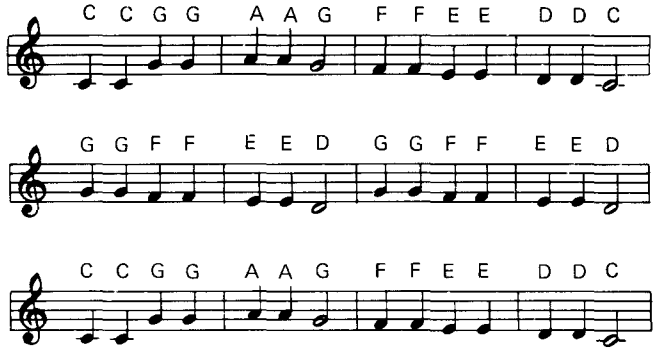
Every note in a song is not held for the same length of time. In order to be able to vary the length of time that notes are held, there is a pause command in ATARI PILOT—PA:. The SO: and PA: commands can be used together to make music. The following table shows the PAUSE values for various musical notes.

Pause Values for Musical Notes

PA:16	Sixteenth Note
PA:32	Eighth Note
PA:64	Quarter Note
PA:128	Half Note
PA:256	Whole Note

Each pause is for 1/60 of a second so that PA:2 means to pause for 2/60 of a second and PA:60 means to pause for a second. You can use the pause command in many different kinds of programs to keep something on the screen. It is not limited to music or other sound programs. You can insert it anywhere in a program that you believe is running too fast.

The nursery rhyme "Twinkle, Twinkle, Little Star" is represented in musical notations (with the notes written at the top).



Here are the first few bars translated into ATARI PILOT:

```
10 R:TWINKLE TWINKLE TAKE 1
20 SO:13
30 PA:64
40 SO:0
50 PA:4
60 SO:13
70 PA:64
80 SO:20
90 PA:64
100 SO:0
110 PA:4
120 SO:20
130 PA:64
140 SO:22
150 PA:64
160 SO:0
170 PA:4
180 SO:22
190 PA:64
200 SO:20
210 PA:128
```

Programming this simple melody is tedious. However, it's very easy to begin to add chords, vary what you've done, and get fancy once the basic work is finished. Here's a simple variation on "Twinkle, Twinkle" using chords (see lines 20, 60, 80, 120, 140, 180, and 200).

```
10 R:TWINKLE TWINKLE TAKE 2
20 SO:13,17
30 PA:64
40 SO:0
50 PA:4
60 SO:13,17
70 PA:64
80 SO:20,24
90 PA:64
100 SO:0
110 PA:4
120 SO:20,24
130 PA:64
140 SO:22,25
150 PA:64
160 SO:0
170 PA:4
180 SO:22,25
190 PA:64
200 SO:20,24
210 PA:128
```

Notice that lines 40 and 50, 100 and 110, and 160 and 170 all have the same routine:

```
SO:0
PA:4
```

SO:0 shuts off the sound, and PA:4 makes the program pause for 4/60 or 1/15 of a second. This way you can have a single note sounded several times in a row. Notice that "Twinkle, Twinkle, Little Star" begins CC GG AA.

The SO:0 and PA:4 routine makes it possible to have these notes repeat with a small pause in between them. When a note changes (say, from C to G or G to A), it is not necessary to shut off the sound by using: SO:0.

If you expect to have many repeats in a musical piece that you want to program, the U: can be very convenient. Here's a way to rewrite "Twinkle, Twinkle, Take 1" using a U: command and a module *REPEAT:

```
10 R:TWINKLE TWINKLE TAKE 3
20 SO:13
30 PA:64
40 U:*REPEAT
50 SO:13
60 PA:64
70 SO:20
80 PA:64
90 U:*REPEAT
100 SO:20
110 PA:64
120 SO:22
130 PA:64
140 U:*REPEAT
```

```
150 SO:22
160 PA:64
170 SO:20
180 PA:128
190 *REPEAT
200 SO:0
210 PA:4
220 E:
```

This is another example of the convenience of creating modules and using the U: command.

Here are two more simple music programs using only five notes. Copy them and then change them by varying the pauses, adding chords, or introducing more notes. Feel free to become a composer.

```
10 R:FIVE FINGER EXERCISE#1
20 SO:13
30 PA:32
40 SO:15
50 PA:32
60 SO:17
70 PA:32
80 SO:18
90 PA:32
100 SO:20
110 PA:64
120 SO:20
130 PA:32
140 SO:18
150 PA:32
160 SO:17
170 PA:32
180 SO:15
190 PA:32
200 PA:32
210 SO:13
220 PA:64
```

```
10 R:FIVE FINGER EXERCISE#2
20 SO:13,17
30 PA:32
40 SO:15,18
50 PA:32
60 SO:17,20
70 PA:32
80 SO:18,22
90 PA:32
100 SO:20,24
110 PA:64
120 SO:20,24
130 PA:32
140 SO:18,22
150 PA:32
```

```
160 SO:17,20
170 PA:32
180 SO:15,18
190 PA:32
200 SO:13,17,20,25
210 PA:128
```

Here are some simple sound effects modules that use ATARI PILOT and can be added to many different types of programs. They use a few commands that will be introduced later. However, they are short enough to copy and use. How they work will become clear after you've gone through the rest of this chapter.

Sound Effects Module 1. This runs from SO:31 to SO:1 in about a second. It sounds a bit like a space gunshot or a jet plane flying past. You can use it in many different ways in the middle of your programs. By putting a PA: statement in at line 135, for example,

```
135 PA:10
```

or

```
135 PA:20
```

you can get a descending scale and a distinctly more musical sound.

```
100 *SOUNDEFFECT1
110 C:#S=31
120 *LOOP
130 SO:#S
140 C:#S=#S-1
150 J(#S>0):*LOOP
160 E:
```

Sound Effects Module 2. This is the ordinary arcade game space shot effect.

```
100 *SOUNDEFFECT2
110 C:#T=6
120 *LOOP1
130 C:#S=31
140 *LOOP2
150 SO:#S
160 C:#S=#S-1
170 J(#S>0):*LOOP2
180 C:#T=#T-1
190 J(#T>0):*LOOP1
200 E:
```

Sound Effects Module 3. This effect is an elaboration of Sound Efects Module 2. It sounds a bit like a staccato laser machine gun or perhaps a space bee.

```
100 *SOUNDEFFECT3
110 C:#T=9
120 *LOOP1
130 C:#S=31
140 *LOOP2
150 SO:#S
160 C:#S=#S-3
170 J(#S>0):*LOOP2
180 C:#T=#T-1
190 J(#T>0):*LOOP1
200 E:
```

Sound Effects Module 4. This is an alien landing effect. It's another of the arcade sounds that are very simple to duplicate with the ATARI Computer and used throughout programs that you create with the U: command.

```
100 *SOUNDEFFECT4
110 C:#T=9
120 *LOOP1
130 C:#S=20
140 *LOOP2
150 SO:#S
160 C:#S=#S+1
170 J(#S<32):*LOOP2
180 C:#T=#T-1
190 J(#T>0):*LOOP1
200 E:
```

Sound Effects Module 5. This effect creates an ascending and descending scale. It can be used at the end of a game or as an introductory remark. If you increase the pauses at lines 140 and 190, say, to PA:10 or PA:20, you will get a much more musical effect.

```
100 *SOUNDEFFECT5
110 C:#S=31
120 *LOOP1
130 SO:#S
140 PA:2
150 C:#S=#S-2
160 J(#S>0):*LOOP1
170 *LOOP2
180 SO:#S
190 PA:2
200 C:#S=#S+2
210 J(#S<32):*LOOP2
```

The sound programs that we've presented are quite simple. Here is a much longer and more complex piece of music that can be programmed using ATARI PILOT. It is the aria "Jesus, Joy of Man's Desiring" from Bach's 57th Cantata. We'll return to it later in this chapter and show how we added color and graphics to accompany the music. It uses simple commands in a complex way and is an example of how interesting music can be programmed in PILOT.

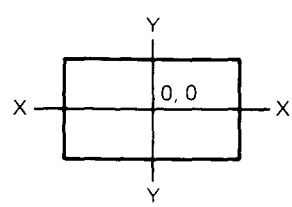
```
820 *MEASURE1
830 C:#M=1
840 SO:1
850 U:*T
860 SO:1,13
870 U:*T
880 SO:1,15
890 U:*T
900 SO:1,13,17
910 U:*T
920 SO:1,20
930 U:*T
940 SO:1,12,18
950 U:*T
960 SO:10,13,18
970 U:*T
980 SO:10,22
990 U:*T
1000 SO:10,15,20
1010 U:*T
1020 C:#M=2
1030 SO:5,13,20
1040 U:*T
1050 SO:5,25
1060 U:*T
1070 SO:5,15,24
1080 U:*T
1090 SO:10,17,25
1100 U:*T
1110 SO:10,20
1120 U:*T
1130 SO:10,13,17
1140 U:*T
1150 SO:10,5,13
1160 U:*T
1170 SO:10,15
1180 U:*T
1190 SO:10,8,17
1200 U:*T
1210 C:#M=3
1220 SO:3,10,18
1230 U:*T
1240 SO:3,20
1250 U:*T
1260 SO:3,12,22
1270 U:*T
```

```
1280 SO:5,13,20
1290 U:*T
1300 SO:5,18
1310 U:*T
1320 SO:5,8,17
1330 U:*T
1340 SO:6,10,15
1350 U:*T
1360 SO:6,17
1370 U:*T
1380 SO:6,5,13
1390 U:*T
1400 C:#M=4
1410 SO:8,3,12
1420 U:*T
1430 SO:8,13
1440 U:*T
1450 SO:8,15
1460 U:*T
1470 SO:3,8,12
1480 U:*T
1490 SO:8,13
1500 U:*T
1510 SO:12,15
1520 U:*T
1530 SO:8,12,18
1540 U:*T
1550 SO:8,17
1560 U:*T
1570 SO:8,12,15
1580 U:*T
1590 C:#M=5
1600 SO:1,13,17
1610 U:*T
1620 SO:1,13
1630 U:*T
1640 SO:1,12,15
1650 U:*T
1660 SO:10,13,17
1670 U:*T
1680 SO:10,20
1690 U:*T
1700 SO:10,12,18
1710 U:*T
1720 SO:6,10,13,18
1730 U:*T
1740 SO:6,22
1750 U:*T
1760 SO:6,15,20
1770 U:*T
1780 C:#M=6
1790 SO:5,13,17,20
1800 U:*T
1810 SO:5,25
1820 U:*T
1830 SO:5,15,24
1840 U:*T
1850 SO:10,17,25
1860 U:*T
1870 SO:10,20
1880 U:*T
```

```
1890 SO:10,13,17
1900 U:*T
1910 SO:8,10,13
1920 U:*T
1930 SO:8,15
1940 U:*T
1950 SO:8,17
1960 U:*T
1970 C:#M=7
1980 SO:6,10,15
1990 U:*T
2000 SO:6,20
2010 U:*T
2020 SO:6,18
2030 U:*T
2040 SO:7,10,13,17
2050 U:*T
2060 SO:7,15
2070 U:*T
2080 SO:7,13
2090 U:*T
2100 SO:8,3
2110 U:*T
2120 SO:8,13
2130 U:*T
2140 SO:8,6,12
2150 U:*T
2160 C:#M=8
2170 SO:1,5,8,13
2180 PA:40
2185 E:
3000 *T
3010 PA:10
3020 E:
```

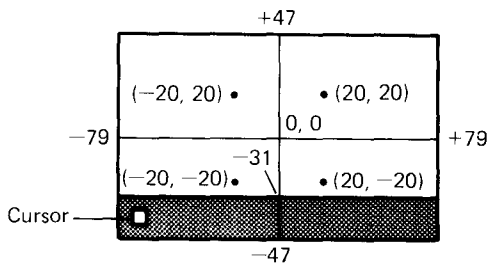
ATARI PILOT Graphics

ATARI PILOT graphics is quite easy to master once you understand the nature of the screen and the characteristics of the turtle. The screen can be thought of as a piece of graph paper 95 columns long and 159 columns wide with two axes crossing each other in the middle.



As in most school arithmetic the middle point is named (0,0) and other points on the grid are presented by the column and row they are in (note that points are always written X,Y

where X stands for the column and Y stands for the row). The X, or column, axis goes from −79 through 0,0 to +79, and the Y axis rows go from −47 to +47. There is a text window on the bottom of the screen in PILOT graphics so that the rows of your drawing board actually end at −31, as in the graph shown here.



We have plotted four points on the screen. Notice how the signs of the X and Y coordinates change according to the position of the point in relationship to the X and Y axes. Imagine where the following points would be on the grid:

20,0	0,20
−20,0	0,−20

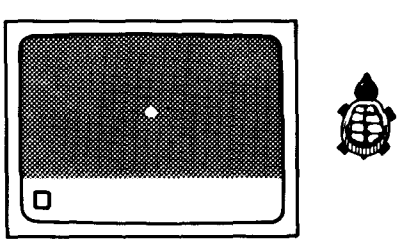
Now comes the first graphics command:

```
GR:
```

Each line in a program that uses graphics *must* begin with GR:. Now turn on the ATARI Computer, type:

```
GR:GOTO 0,0
```

and press the [RETURN] key. You'll notice that the screen looks like this.

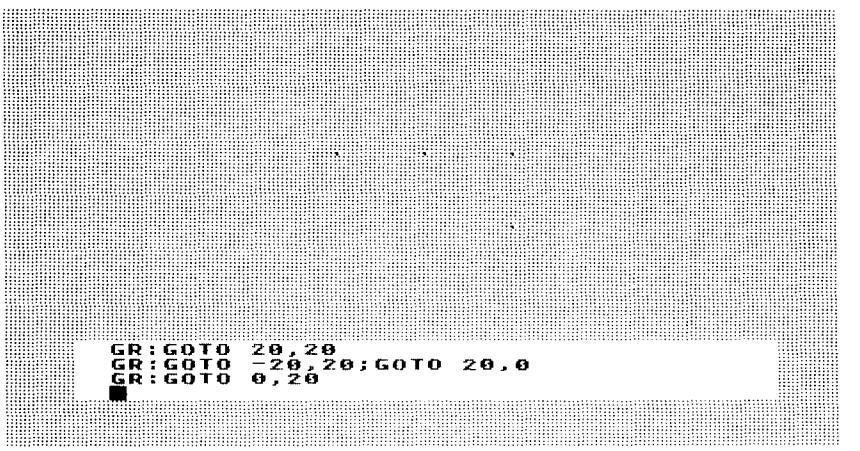


The little white dot at the center is called the "turtle." It does not look as if it is at the center of the screen because the text window with the cursor takes up the equivalent of 16 rows.

The command GOTO: told the turtle to go to point 0,0. Try to move the turtle around before continuing. Here are the points to try: Simply type GR:GOTO X,Y (which will appear in the text window) and press the [RETURN] key.

–20,–20	–20,20	20,–20	20,20
–20,0	0,–20	20,0	0,20

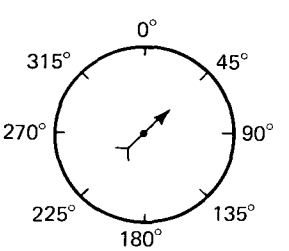
Notice that the turtle will go to the point that you indicate, drop a dot there, and then go on to the next point and drop a dot.



When designing pictures and animations in ATARI PILOT turtle graphics, it is a good idea to keep a pencil and a piece of graph paper next to your computer and do some sketching along with programming.

The turtle itself is like the arrow or spinners used in many board games. It can spin 360° and point in any direction from 0° to 360°. When you begin a program, the turtle is located at point 0,0 and faces toward 0°. The turtle always moves and draws in the direction that it is pointing. As you begin to program in turtle graphics, you must remember

1. What point the turtle is on
2. What direction the turtle is facing.

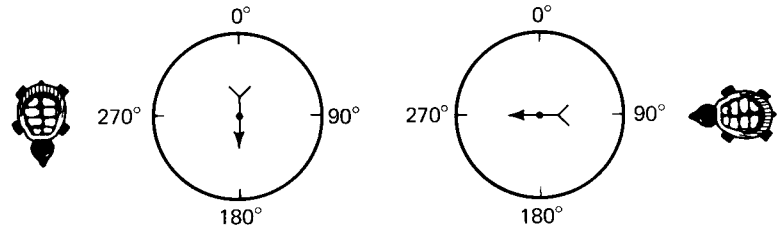


Most commands that the turtle understands come in pairs:

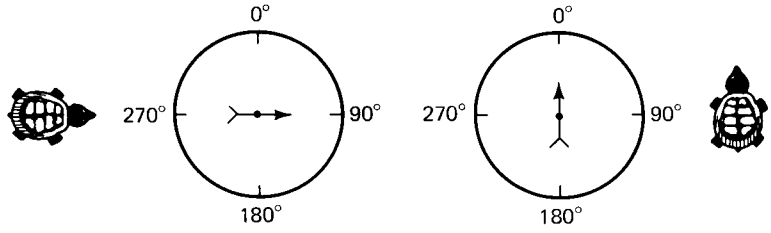
1	GR:DRAW	GR:GO	GR:TURN	GR:FILL
2	GR:DRAWTO	GR:GOTO	GR:TURNTO	GR:FILLTO

The commands on line 1 refer to units on the grid and mean just what they say. GR:DRAW 10 means draw 10 units, which the turtle will do. Remember that the turtle will draw the units in the direction that it is facing.

GR:GO 10 will move the turtle 10 units in that direction. GR:TURN allows you to control the direction of the turtle. GR:TURN 90 instructs the turtle to turn 90°. Thus if the turtle were facing toward 180°, the turtle would turn 90° and end up facing 270°.



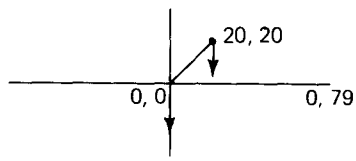
The turtle can also turn counterclockwise by putting a negative sign before the number of degrees that you want it to turn. Thus if the turtle were facing toward 90°, the command GR:TURN –90 would turn it to 0.



Before considering the GR:FILL and GR:FILLTO commands, which are a bit more complex, let's look at the other commands on line 2: GR:DRAWTO, GR:GOTO, and GR:TURNTO. The first two commands refer to points on the grid. Thus

```
GR:DRAWTO  20,20
```

will result in a line from the current location to the point 20,20. Remember that when the turtle is finished drawing, it will be at the end of the line and facing in the same direction as when it began, as illustrated. The turtle begins at 0,0, facing 180, and ends at 20,20, still facing 180.



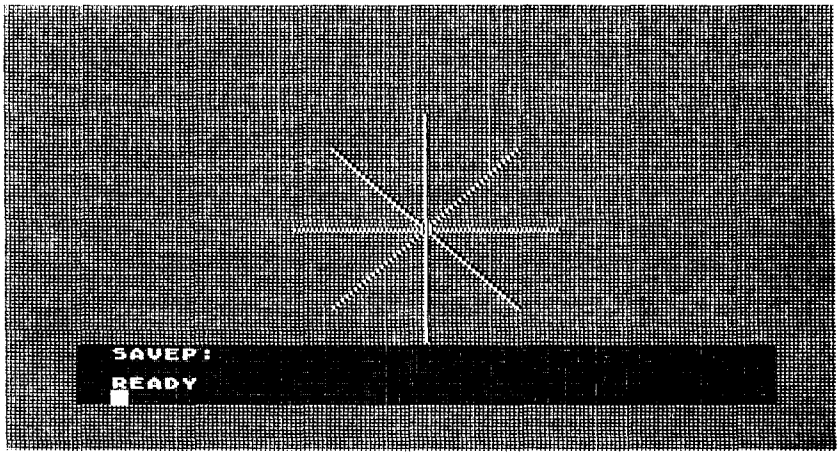
GR:GOTO sends the turtle to a particular point on the grid— thus GR:GOTO 20,20 moves the turtle to 20,20. It ends there but once again facing in the same direction it began.

GR:TURNTO is a second way to control the direction of the turtle. It tells the turtle which angle to face. Thus GR:TURNTO 90 faces the turtle toward 90°, and GR:TURNTO 166 faces the turtle toward 166°.

Here's a simple program using the GR:DRAW, GR:GOTO, and GR:TURNTO commands. Notice that the program uses a shortcut to join several commands. *In ATARI PILOT turtle graphics you can put any number of graphics commands on the same line by separating them with a semicolon (;).*

```
10 GR:DRAW 30;GOTO 0,0;TURNTO 45
20 GR:DRAW 30;GOTO 0,0;TURNTO 90
30 GR:DRAW 30;GOTO 0,0;TURNTO 135
40 GR:DRAW 30;GOTO 0,0;TURNTO 180
50 GR:DRAW 30;GOTO 0,0;TURNTO 225
60 GR:DRAW 30;GOTO 0,0;TURNTO 270
70 GR:DRAW 30;GOTO 0,0;TURNTO 315
80 GR:DRAW 30
```

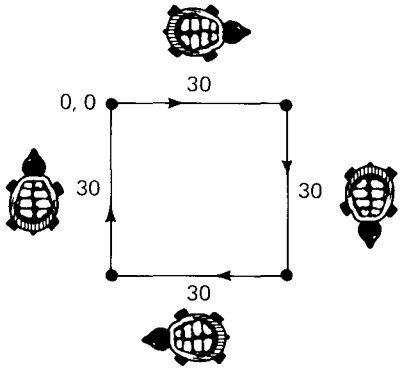
Study the program and try to figure out what it does before studying the screen dump illustration or looking at our explanation. You probably noticed that the only differences between lines 10 to 70 was the angle in the GR:TURNTO command. The turtle starts at 0,0, facing 0°. It draws 30, then returns to 0,0, turns to 45°, and draws 30 again. Then it returns to 0,0 rotates another 45° to 90°, and draws 30. This goes on until the turtle has drawn a line radiating out from 0,0 every 45°, as shown in the following screen dump.



Try to figure out what this program will result in.

```
10 GR:TURN 90;DRAW 30
20 GR:TURN 90;DRAW 30
30 GR:TURN 90;DRAW 30
40 GR:TURN 90;DRAW 30
```

Here's what will happen.

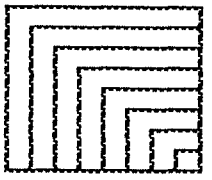


The turtle will draw a square and end back on 0,0 facing 0°, just where it started. Notice that each line in the program is identical; in effect the program says that the turtle should turn 90° and draw 30 units four times. There is a shorthand for continually repeating operations in ATARI PILOT turtle graphics. Lines 10 to 40 can be condensed into one line:

```
10 GR:4(TURN 90;DRAW 30)
```

In general you can enclose in parentheses the commands that you want to be repeated a particular number of times and place the number of repetitions before the left hand parenthesis. The next program, which produces a series of nested boxes, uses this technique a number of times.

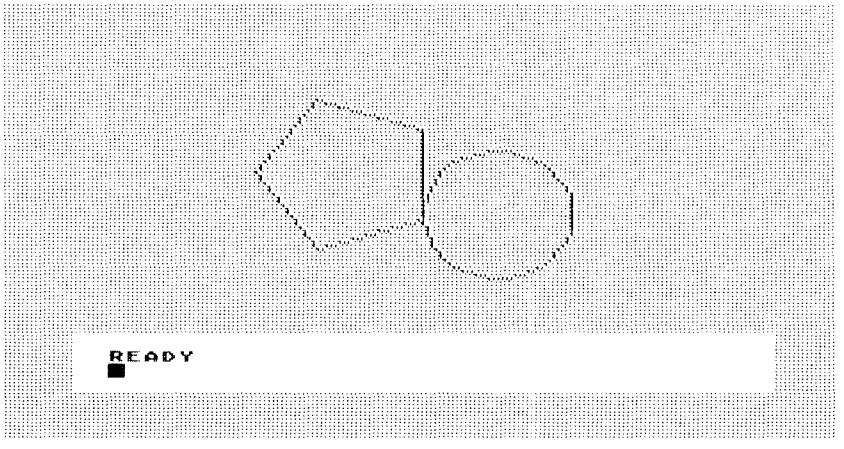
```
100 GR: GOTO 0,20
110 GR:4(TURN 90;DRAW 40)
120 GR:4(TURN 90;DRAW 35)
130 GR:4(TURN 90;DRAW 30)
140 GR:4(TURN 90;DRAW 25)
150 GR:4(TURN 90;DRAW 20)
160 GR:4(TURN 90;DRAW 15)
170 GR:4(TURN 90;DRAW 10)
180 GR:4(TURN 90;DRAW 5)
```



Line 100 jumps the turtle up to row 20, which makes it possible to draw a square with sides 40 units long, as in line 110. If the program didn't begin this way, the 40-unit square would run into the text window at the bottom of the screen.

With the simple concepts presented, it is possible to draw interesting designs and pictures. For example,

```
10 GR:36(DRAW3;TURN10)
20 GR:5(DRAW 25;TURN -72)
```



GR:36(DRAW 5; TURN 10) produces the circle on the right while GR:5 (DRAW 25; TURN –72) produces the pentagon on the left. Notice that 36 × 10 = 360 and 5 × 72 = 360. That means that each program turns 360°, one producing a 36-sided shape approximating a circle, the other a 5-sided shape. A six-sided shape could be drawn using GR:6(DRAW 25; TURN 60).

Try to produce regular polygons with

3 sides

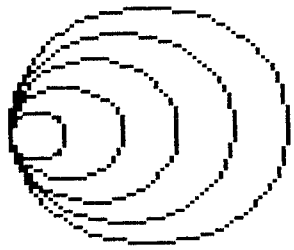
10 sides

12 sides

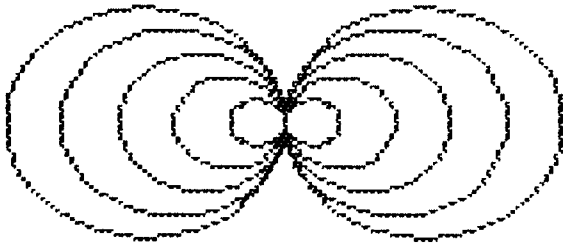
20 sides

Write and run programs that do this. You can also experiment with the length of the line in the DRAW command. For example, here is a series of nested circles created by changing the DRAW statement.

```
10 GR:36(DRAW5;TURN10)
20 GR:36(DRAW4;TURN10)
30 GR:36(DRAW3;TURN10)
40 GR:36(DRAW2;TURN10)
50 GR:36(DRAW1;TURN10)
```



Here is a double design that could be called *pop eyes*.



It was generated by adding the following lines after line 50 in the previous program.

```
60 GR:36(DRAW5;TURN-10)
70 GR:36(DRAW4;TURN-10)
80 GR:36(DRAW3;TURN-10)
90 GR:36(DRAW2;TURN-10)
100 GR:36(DRAW1;TURN-10)
```

Using Color. With the graphics that you have learned it is possible to draw many pictures. Try a face and a man or woman. When drawing the following, GR:PEN commands will be very useful:

GR:PENYELLOW
GR:PENBLUE
GR:PENRED
GR:PENERASE
GR:PENUP

Imagine the turtle carrying three colored pens and an eraser as it moves around the screen. It can use any of the pens (though only one at a time), can erase anything that it

has drawn, and can pick the pen up and move from one part of the screen to another without leaving its usual dot.

When you begin to use the graphics mode, the turtle *always* starts drawing in yellow. To change the color of the pen, you use the command corresponding to the pen color that you want. For example, here's the circle program in three colors.

```
10 GR:36(DRAW5;TURN10)
15 GR:PENRED
20 GR:36(DRAW4;TURN10)
25 GR:PENBLUE
30 GR:36(DRAW3;TURN10)
35 GR:PENYELLOW
40 GR:36(DRAW2;TURN10)
45 GR:PENRED
50 GR:36(DRAW1;TURN10)
55 GR:PENYELLOW
60 GR:36(DRAW5;TURN-10)
65 GR:PENRED
70 GR:36(DRAW4;TURN-10)
75 GR:PENBLUE
80 GR:36(DRAW3;TURN-10)
85 GR:PENYELLOW
90 GR:36(DRAW2;TURN-10)
95 GR:PENRED
100 GR:36(DRAW1;TURN-10)
```

If you like the effect and think that you might want to use it in the future, you can easily turn it into a module and renumber for use in future programs. PILOT has a very useful command that will automatically renumber your programs:

REN

REN renumbers the program statements beginning with the first line of the program. For example, with the circle program, if I type in REN (it does not use a line number), the program comes out:

```
10 GR:36(DRAW5;TURN10)
20 GR:PENRED
30 GR:36(DRAW4;TURN10)
40 GR:PENBLUE
50 GR:36(DRAW3;TURN10)
60 GR:PENYELLOW
70 GR:36(DRAW2;TURN10)
80 GR:PENRED
90 GR:36(DRAW1;TURN10)
100 GR:PENYELLOW
110 GR:36(DRAW5;TURN-10)
120 GR:PENRED
130 GR:36(DRAW4;TURN-10)
140 GR:PENBLUE
```

```
150 GR:36(DRAW3;TURN-10)
160 GR:PENYELLOW
170 GR:36(DRAW2;TURN-10)
180 GR:PENRED
190 GR:36(DRAW1;TURN-10)
```

REN will also let you begin renumbering at any line number and increase each statement by any amount that you want. For example, REN 60,5 will renumber all statements starting with 60 and increase each line number by 5's.

The way to turn a short program into a module, as you may remember, is first by naming it and then by putting an E: statement at the end of it. Thus we could call the preceding program *CIRCLES. When saving modules for use in other programs, it is helpful to give them high line numbers so that you can tag them on to programs. Here's *CIRCLES, starting at line 600:

```
600 *CIRCLES
610 GR:36(DRAW5;TURN10)
620 GR:PENRED
630 GR:36(DRAW4;TURN10)
640 GR:PENBLUE
650 GR:36(DRAW3;TURN10)
660 GR:PENYELLOW
670 GR:36(DRAW2;TURN10)
680 GR:PENRED
690 GR:36(DRAW1;TURN10)
700 GR:PENYELLOW
710 GR:36(DRAW5;TURN-10)
720 GR:PENRED
730 GR:36(DRAW4;TURN-10)
740 GR:PENBLUE
750 GR:36(DRAW3;TURN-10)
760 GR:PENYELLOW
770 GR:36(DRAW2;TURN-10)
780 GR:PENRED
790 GR:36(DRAW1;TURN-10)
800 E:
```

Here's a simple program that uses *CIRCLES. It asks if someone wants to see circles and uses the module if he or she does. If not, then it types:

SORRY MAYBE SOME OTHER TIME.

Notice that the first lines are all that had to be added. The module wasn't changed in any way.

```
10 T:DO YOU WANT TO SEE CIRCLES?
20 A:
30 M:YES
40 JY:*CIRCLES
50 M:NO
```

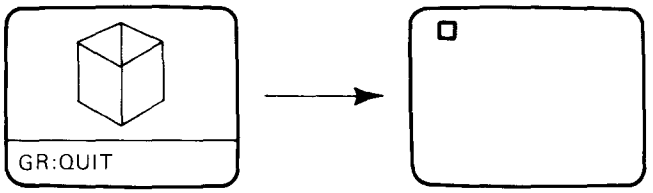
```
60 TY:SORRY MAYBE SOME OTHER TIME
70 E:
600 *CIRCLES
610 GR:36(DRAW5;TURN10)
620 GR:PENRED
630 GR:36(DRAW4;TURN10)
640 GR:PENBLUE
650 GR:36(DRAW3;TURN10)
660 GR:PENYELLOW
670 GR:36(DRAW2;TURN10)
680 GR:PENRED
690 GR:36(DRAW1;TURN10)
700 GR:PENYELLOW
710 GR:36(DRAW5;TURN-10)
720 GR:PENRED
730 GR:36(DRAW4;TURN-10)
740 GR:PENBLUE
750 GR:36(DRAW3;TURN-10)
760 GR:PENYELLOW
770 GR:36(DRAW2;TURN-10)
780 GR:PENRED
790 GR:36(DRAW1;TURN-10)
800 E:
```

The GR:PENERASE command lets you erase what you have drawn. The following simple program repeatedly draws and erases a circle. The GR:PENERASE command can be used to make part of a drawing appear and disappear and, in more complex sketching programs such as the Rambrant program at the end of this chapter, allows you to create and correct your own drawings.

```
10 *CIRCLENOCIRCLE
20 GR:36(DRAW5;TURN10)
30 GR:PENERASE
40 GR:36(DRAW5;TURN10)
50 GR:PENYELLOW
60 J:*CIRCLENOCIRCLE
```

The PEN UP command tells the turtle to pick the pen off the screen. While the pen is up, all movement by the turtle will be invisible. To put the pen "down," you choose a pen color.

When developing your own graphics programs two other commands are very useful: GR:QUIT and GR:CLEAR. GR:QUIT allows you to leave the graphics mode and returns you to the full-screen print mode. It is particularly useful if you need to see a whole program listing to make additions or corrections after playing with graphics. In that case you just type GR:QUIT, which will appear in the print window at the bottom of the graphics screen. No line number should be used. When you press [RETURN], the screen will return to the print mode.

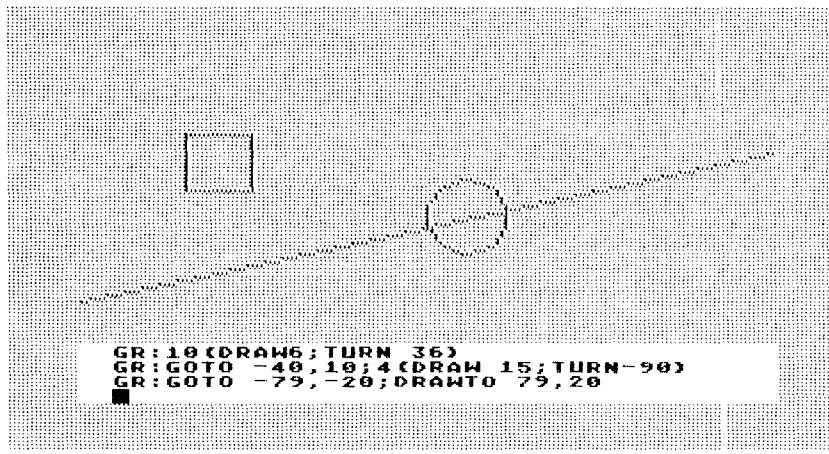


GR:QUIT can also be used to get out of the graphics mode in the middle of a program. For example, this program draws a red circle and then asks if you like it. If you say yes, it draws a blue circle. If no, it ends. At line 120 the screen goes from graphics to print mode.

```
10 U:*REDCIRCLE
20 T:DID YOU LIKE IT?
30 A:
40 M:YES
50 TY:WELL HERE IS ANOTHER ONE
60 J:*BLUECIRCLE
70 M:NO
80 TY:SORRY
90 E:
100 *REDCIRCLE
110 GR:PENRED;36(DRAW 5;TURN 10)
120 GR:QUIT
130 E:
140 *BLUECIRCLE
150 GR:PENBLUE;36(DRAW 5;TURN 10)
160 E:
```

GR:CLEAR does the opposite of GR:QUIT, which returns you from the graphics screen to the print screen. If you type GR:CLEAR and press [RETURN], the graphics screen will appear with the text window beneath it. The turtle will have disappeared from the center of the screen as well (everything was cleared from the screen). You can now use all the graphics commands to sketch and draw. As long as you don't use line numbers, you will see what you've drawn as soon as you press [RETURN]. Notice in the following sketch that you can see at most four print lines at a time in the window. If you like something that you've drawn, you can insert a line number at the beginning of any line and it will become a part of your program. If you want a clear screen to sketch with, all you must do is type GR:CLEAR.

(Note: The four-line limit is why GR:QUIT is needed. You simply can't see a long program and watch it run at the same time.)



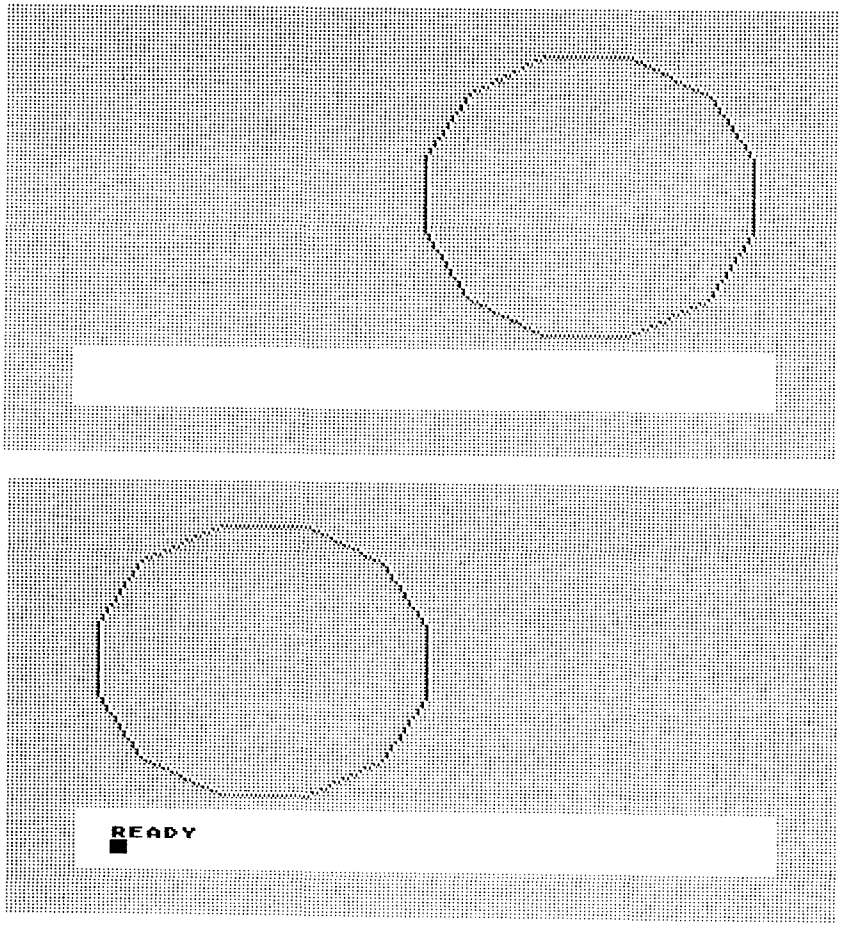
GR:CLEAR can also be used inside a program to clear the screen and then see another image. *Most graphic programs begin with GR:CLEAR, as it eliminates the dot at 0,0, where the turtle starts.* GR:CLEAR is also the secret of simple animation. You draw a figure at one spot, clear the screen, then move it to another spot, and on and on with GR:CLEAR statements. You can even control the speed of movement by using the pause PA: command described in the sound section. This simple program moves a 12-sided regular polygon from the right of the screen to the left of the screen. If you turned into a module that jumped back to itself, you would have continual movement back and forth as in Program 2.

Program 1

```
10 GR:12(DRAW 20;TURN 30)
20 PA:50
30 GR:CLEAR
40 GR:12(DRAW 20;TURN-30)
```

Program 2

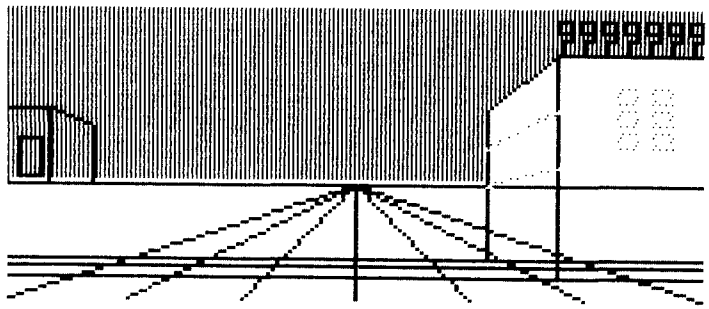
```
5 *CIRCLESWITCH
10 GR:12(DRAW 20;TURN 30)
20 PA:50
30 GR:CLEAR
40 GR:12(DRAW 20;TURN-30)
45 GR:CLEAR
50 J:*CIRCLESWITCH
```



Animation. This simple animation program moves a circle across the screen. Add some eyes, a mouth, and some color; you'll have your first animation. Read the program first and imagine what it does. By now it shouldn't be too hard.

```
5 GR:CLEAR
10 GR:GOTO -80,0:36(DRAW 3:TURN 10)
20 PA:10
30 GR:CLEAR
40 GR:GOTO -60,0:36(DRAW 3:TURN 10)
50 PA:10
60 GR:CLEAR
70 GR:GOTO -40,0:36(DRAW 3:TURN 10)
80 PA:10
90 GR:CLEAR
100 GR:GOTO -20,0:36(DRAW 3:TURN 10)
110 PA:10
120 GR:CLEAR
130 GR:GOTO -10,0:36(DRAW 3:TURN 10)
140 PA:10
150 GR:CLEAR
160 GR:GOTO 0,0:36(DRAW 3:TURN 10)
```

This picture resulted from sketching and saving parts of sketches by turning them into modules.



```
2 R:    *** DISAPPEARING POINT ***
4 R:        COPYRIGHT (C) 1982
6 R:            BY HERB KOHL
100 GR:TURNTO 180;DRAW 31;GOTO -79,0;T
URNTO 90;DRAW 158
110 GR:GOTO 0,0;DRAWTO -79,-31;GOTO 0,
0;DRAWTO -52,-31;GOTO 0,0;DRAWTO -26,-
31
120 GR:GOTO 0,0;DRAWTO 79,-31;GOTO 0,0
;DRAWTO 26,-31;GOTO 0,0;DRAWTO 52,-31
130 GR:GOTO -79,-25;DRAWTO 79,-25;GOTO
 -79,-22;DRAWTO 79,-22;GOTO -79,-20;DR
AWTO 79,-20
140 GR:GOTO 46,-25;TURNTO 0;DRAW 60;TU
RNTO 90;DRAW 33
150 GR:GOTO 30,-20;TURNTO 0;DRAW 40;DR
AWTO 46,35
160 GR:PEN RED;GOTO 30,10;DRAWTO 46,20
170 GR:GOTO 60,10;4(DRAW 4;TURN 90)
180 GR:GOTO 60,16;4(DRAW 4;TURN 90)
190 GR:GOTO 60,22;4(DRAW 4;TURN 90)
200 GR:GOTO 68,10;4(DRAW 4;TURN 90)
210 GR:GOTO 68,16;4(DRAW 4;TURN 90)
220 GR:GOTO 68,22;4(DRAW 4;TURN 90)
230 GR:PEN RED;GOTO 30,0;DRAWTO 46,5
240 GR:PEN BLUE;GOTO -79,1;FILL 46
250 GR:PEN YELLOW;GOTO -60,0;TURNTO 0;
DRAW 15;DRAWTO -70,20;TURNTO 180;DRAW2
0
260 GR:GOTO -70,20;DRAWTO -79,20
270 GR:GOTO -77,2;TURNTO 0;2(DRAW 10;T
URN 90;DRAW 5;TURN 90)
280 C:#X=0
290 *PLAZAWALK
300 GR:GOTO #X+47,35
310 PA:20
320 U:*AWALK
330 C:#X=#X+5
340 J(#X<70):*PLAZAWALK
350 *END E:
360 *AWALK
370 GR:TURNTO 0
380 GR:DRAW 4;TURN 90;DRAW 2;TURN-90;D
RAW 2;TURN -90;4(DRAW 3;TURN 90)
390 E:
```

Before proceeding to the last two graphic commands—GR:FILL and GR:FILLTO—here's a summary of the commands covered so far. Review them if you have forgotten them.

GR:GO	GR:DRAW	GR:TURN	1
GR:GOTO	GR:DRAWTO	GR:TURNTO	

GR:PEN		
GR:PENRED	GR:PENERASE	2
GR:PENBLUE	GR:PENUP	
GR:PENYELLOW		

GR:CLEAR	REN (renumber)	3
GR:QUIT		

Notice that PILOT turtle graphics commands do what they say. PILOT is designed to be a friendly language. If you forget a command, think of what you want to do and there's a good chance that the right command will pop out of your memory.

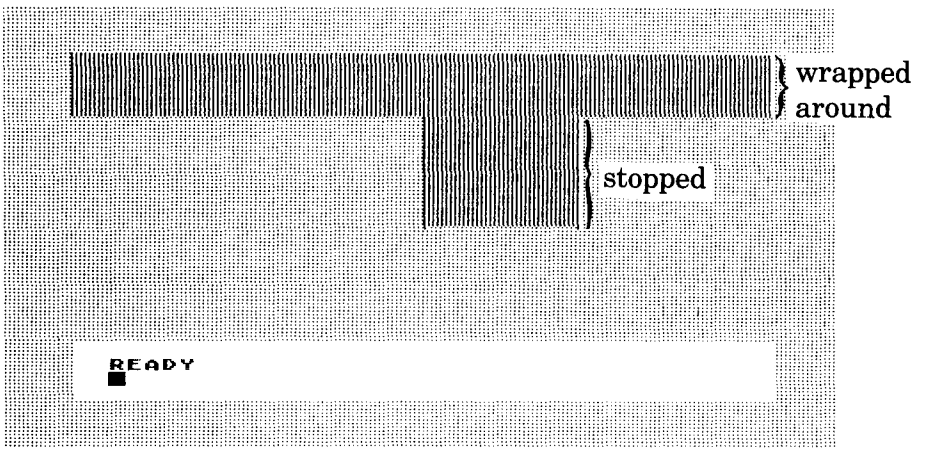
Finally, GR:FILL and GR:FILLTO also do just what they say: fill in sections of the screen with different colors and make it possible for you to color in your drawings. GR:FILL is like the GR:DRAW statement. It tells the turtle to draw a line *N* units long in the direction that it is facing. As the turtle moves, however, all the background screen space to the *right* of the turtle will be filled in with the current pen color. The fill will stop wherever there is a line. If there is no line, the turtle will wrap around the screen and return on the other side to make a stripe of color. To see that, try this extremely simple program:

```
10 GR:CLEAR;FILL 20
```

You get a yellow band across the screen. If you put a line to the right of the fill statement, the filling would stop wherever the line was, as in this program:

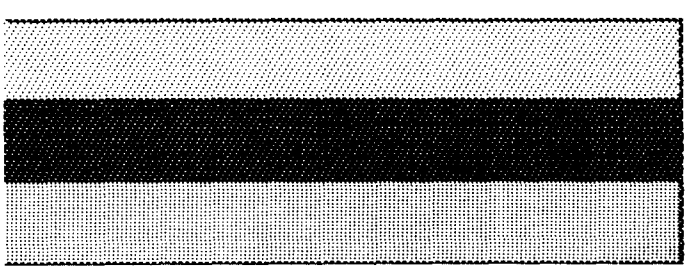
```
10 GR:CLEAR;GOTO 35,0;DRAW 30
20 GR:GOTO 0,0;FILL 50
```

This is the shape generated by the program (on your screen it would be yellow). Notice that the fill stopped where the line was drawn. However, as it went past that line up the screen, it went off the screen at the right and then wrapped around to the left, making a solid stripe.



A good way to get a sense of the use of GR:FILL is to make flags of different colors, designs, and stripes. Here's a simple three-color flag. First the border is drawn from top left around the edges and then the stripes are fitted from the left-hand side to the right border.

```
110 GR:CLEAR
120 GR:GOTO -77,39:TURNTO 90:DRAW 154
130 GR:TURN 90:DRAW 67:TURN 90:DRAW 154
140 GR:PEN RED:TURN 90:FILL 22
150 GR:PEN YELLOW:FILL 23:PEN BLUE:FILL 22
```



Here are three more simple flag progams. Try to match them with the following flags before running them. Then make your own flags. Draw diagonal lines. Try to fill in a square, a circle, a window in a house that you draw. Practice with GR:FILL is important.

1

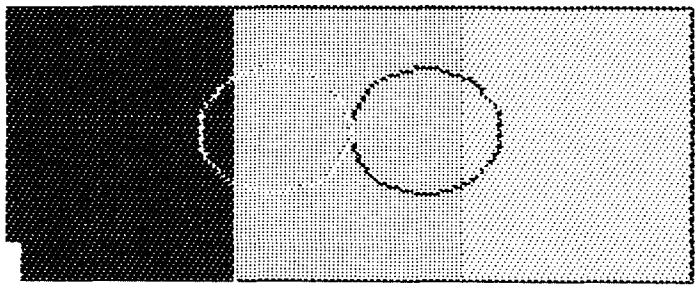
```
100 R:FLAGS SIMPLE FORM #2
110 GR:CLEAR
120 GR:GOTO -20,5;36(DRAW 2;TURN 10)
130 GR:GOTO -77,39;TURNTO 90;DRAW 154
140 GR:TURN 90;DRAW 67;TURN 90;DRAW 15
4
150 GR:PEN RED;TURN 90;FILL 22
160 GR:PEN BLUE;DRAW 23;PEN RED;FILL 2
2
```

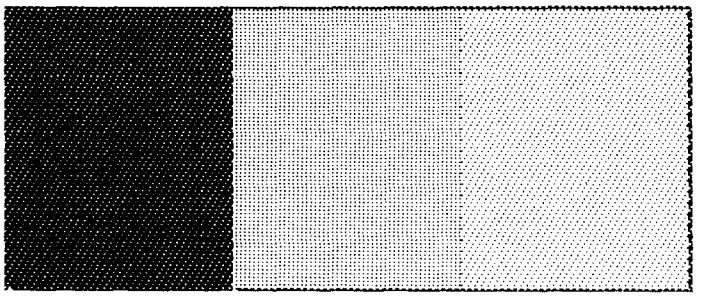
2

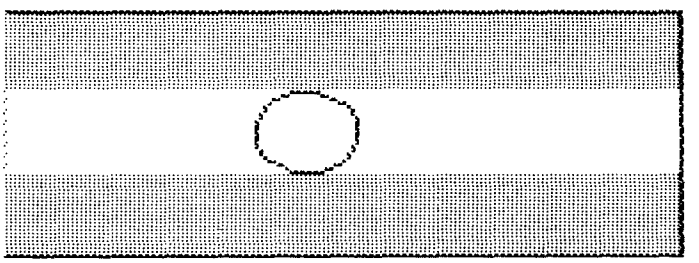
```
100 GR:CLEAR
110 GR:GOTO -78,46;TURNTO 90
120 GR:2(DRAW 156;TURN 90;DRAW 76;TURN
 90)
130 GR:GOTO 26,-30;TURNTO 0;PENBLUE;FI
LL75
140 GR:GOTO-26,-30;TURNTO 0;PEN RED;FI
LL75
150 GR:GOTO -77,-30;TURNTO 0;PENYELLOW
;FILL75
160 GR:GOTO 0,10;PEN YELLOW;36(DRAW 3;
TURN 10)
170 GR:GOTO 0,10;PEN BLUE;36(DRAW 3;TU
RN-10)
```

3

```
100 GR:CLEAR
110 GR:GOTO -78,46;TURNTO 90
120 GR:2(DRAW 156;TURN 90;DRAW 76;TURN
 90)
130 GR:GOTO 26,-30;TURNTO 0;PEN BLUE;F
ILL75
140 GR:GOTO-26,-30;TURNTO 0;PEN RED;FI
LL75
150 GR:GOTO -77,-30;TURNTO 0;PEN YELLO
W;FILL75
```

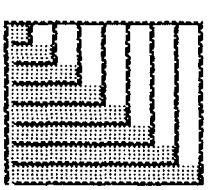




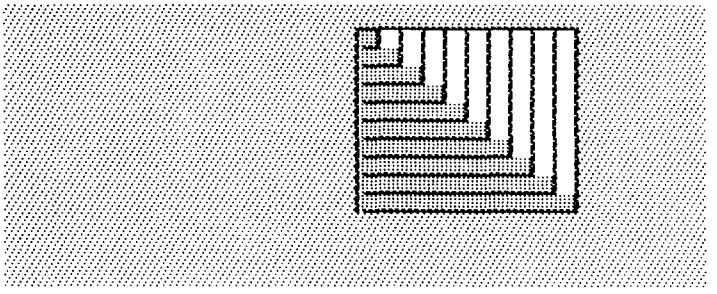


The next two programs, using GR:FILL, turn the nested series of boxes that we created when discussing simple drawing into a three-dimensional stairway (or any other shape that you can see in it).

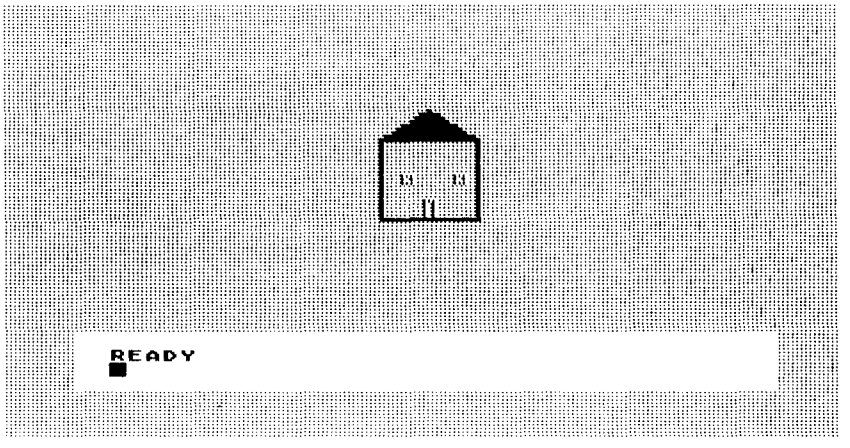
```
100 GR: GOTO 0,20
110 GR:4(TURN 90;DRAW 40)
120 GR:4(TURN 90;DRAW 35)
130 GR:4(TURN 90;DRAW 30)
140 GR:4(TURN 90;DRAW 25)
150 GR:4(TURN 90;DRAW 20)
160 GR:4(TURN 90;DRAW 15)
170 GR:4(TURN 90;DRAW 10)
180 GR:4(TURN 90;DRAW 5)
190 GR:GOTO 1,20;TURNTO 180;PENRED;FIL
L 40
```



```
100 GR: GOTO 0,40
110 GR:4(TURN 90;DRAW 50)
120 GR:4(TURN 90;DRAW 45)
130 GR:4(TURN 90;DRAW 40)
140 GR:4(TURN 90;DRAW 35)
150 GR:4(TURN 90;DRAW 30)
160 GR:4(TURN 90;DRAW 25)
170 GR:4(TURN 90;DRAW 20)
180 GR:4(TURN 90;DRAW 15)
190 GR:4(TURN 90;DRAW 10)
200 GR:4(TURN 90;DRAW 5)
210 GR:GOTO 1,40;TURNTO 180;PENRED;FIL
L 50
220 GR:GOTO -79,-32;TURNTO 0;PENBLUE;F
ILL 78
230 GR:GOTO 51,41;TURNTO 180;FILL 55
```

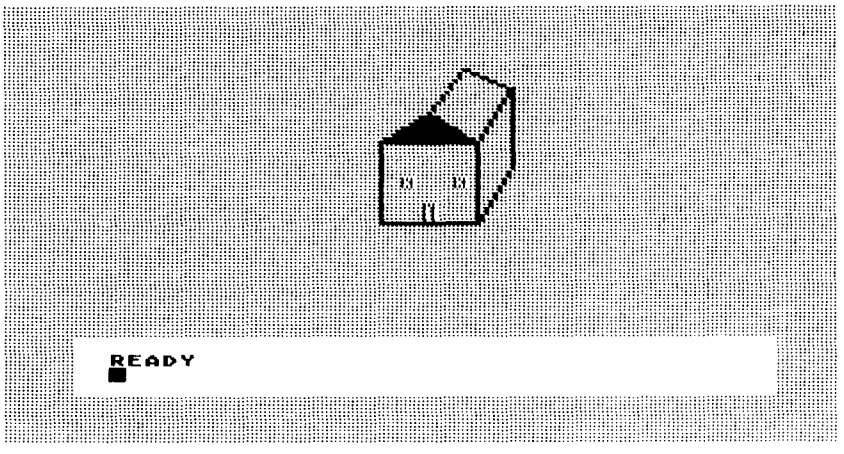


GR:FILLTO X,Y functions just like GR:FILL except that it draws a line to the point X,Y and fills in anything to the right of the line with the current pen color until it is blocked by another line. Here's a simple house module that you may want to use in some programs. Notice that the roof is made red by using a GR:FILLTO statement (line 550).



```
500 *HOUSE
510 GR:2(DRAW 5;TURN 90;DRAW 2;TURN 90)
520 GR:GOTO -5,10;TURNTO 0;4(DRAW 2;TURN 90)
530 GR:GOTO 7,10;TURNTO 0;4(DRAW 2;TURN90)
540 GR:PEN RED;GOTO -10,0;TURNTO 0;4(DRAW 22;TURN 90)
550 GR:GOTO 12,22;DRAWTO 1,30;FILLTO -10,22
560 E:
```

Here's another house module that uses GR:FILLTO. Remember that you can change these modules, place them on the screen anywhere you want, color them in differently, and change their size. All this can be done with the commands and concepts presented in this book.

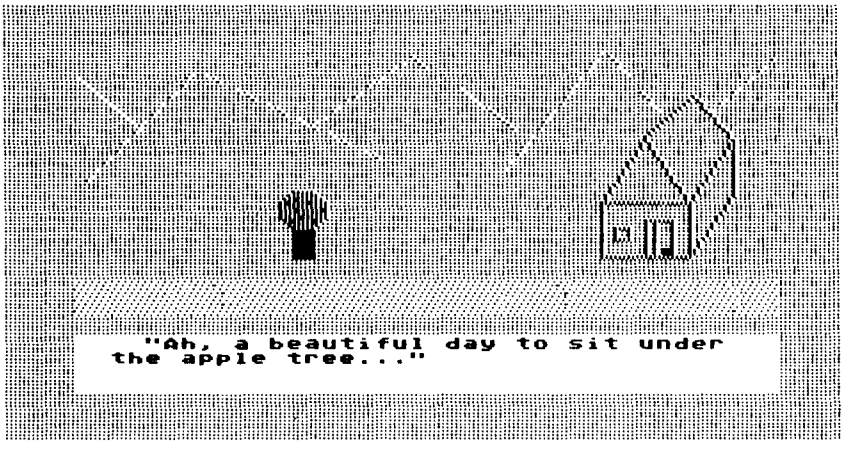



```
700 *3DHOUSE
710 GR:2(DRAW 5;TURN 90;DRAW 2;TURN 90)
720 GR:GOTO -5,10;TURNTO 0;4(DRAW 2;TURN 90)
730 GR:GOTO 7,10;TURNTO 0;4(DRAW 2;TURN90)
740 GR:PEN RED;GOTO -10,0;TURNTO 0;4(DRAW 22;TURN 90)
750 GR:GOTO 12,22;DRAWTO 1,30;FILLTO -10,22
760 GR:GOTO 12,0;DRAWTO 20,15;DRAWTO 20,37
770 GR:GOTO 12,22;DRAWTO 20,37
780 GR:GOTO 1,30;DRAWTO 9,42;DRAWTO 20,37
790 E:
```

This much more complex animation program uses this house module. Unless you are an experienced programmer, you will not understand it. However, you can look for the modules and the U: and J: statements and piece quite a bit of it together. It uses the C: command, which we will introduce shortly, and the C: @ B#X command, which is the command for poking directly into a memory location in ATARI PILOT. B#X means "at byte number X." Poking is a powerful device as it allows you to change things directly in the computer's

chip. Appendix II provides an explanation of this concept and gives POKE numbers that allow you to do powerful things in ATARI PILOT. We recommend that you look at it and at the ATARI PILOT reference manual, which also contains POKE numbers. If you don't understand it at first reading, don't worry. You'll see its value as you practice programming. Refer to it several times and experiment with it.

This program, created by Andy Hall, a sophomore at Cal Tech, is a short animation about Sir Isaac Newton. It is the first of many longer programs in this book. You may want to copy it now. If not, look through it and return to it when you have time. The virtue of copying programs is that it gives you a feel for the way that experienced programmers work as well as access to a lot of software, which once copied can be stored on tape or cassette and used repeatedly.

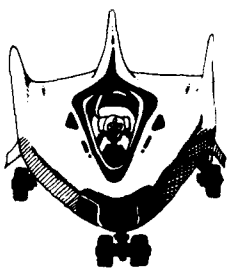



```
N
2 R:  *** "NEWTON": A CARTOON ***
4 R:       COPYRIGHT (C) 1982
6 R:          BY ANDY HALL
100 GR:QUIT
110 U:*TITLE
120 C:#P=-85
130 GR:CLEAR;PENYELLOW
140 U:*HOUSE
150 U:*MOUNTAINS
160 U:*TREE
170 C:#W=5
180 U:*WAIT
190 U:*DOOROPEN
200 U:*RIVER
210 U:*EXIT
220 C:#W=4
230 U:*WAIT
```

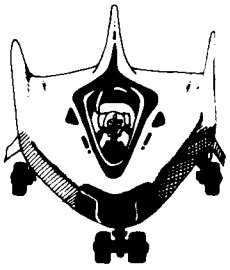
```
240 T:    "Ahhh..."
250 C:#W=10
260 U:*WAIT
270 T:[CLEAR]
280 C:#W=15
290 U:*WAIT
300 U:*APPLE
310 U:*BOUNCE
320 C:#W=10
330 U:*WAIT
340 T:    "Hmmm...   \
350 C:#W=7
360 U:*WAIT
370 T:Nah, couldn't be.
380 C:#W=7
390 U:*WAIT
400 T: But yes!!   \
410 C:#W=3
420 U:*WAIT
430 T:F=GMm/rr!!!" \
440 C:#W=25
450 U:*WAIT
460 E:
470 *HOUSE GR:GOTO40,-10;TURNTO90;DRAW
20;TURN-60;DRAW20;TURN-30;DRAW15
480 GR:TURN-150;DRAW20;TURN-30;DRAW15;
GO-15;TURNTO330;DRAW20;TURN-120;DRAW20
;TURN-30;DRAW15
490 GR:TURNTO90;GO10;TURNTO0;GO33;TURN
45;DRAW15;DRAWTO70,22
500 GR:GOTO40,5;TURNTO90;DRAW20
510 GR:GOTO50,-10;TURNTO0;DRAW10;TURN9
0;DRAW6;TURN90;DRAW10;GOTO 54,-5
520 GR:GOTO43,-5;TURNTO0;4(DRAW3;TURN9
0)
530 E:
540 *MOUNTAINS GR:PENBLUE;GOTO-78,10;T
URNTO40;DRAW40;TURN80;DRAW50;GO-20;TUR
NTO50;DRAW30
550 GR:TURN80;DRAW35;TURNTO215;DRAW10;
TURN180;DRAW40;TURN110;DRAW23
560 GR:GOTO65,30;TURNTO45;DRAW30;GOTO-
65,25;TURNTO315;DRAW30
570 GR:PENUP;GOTO-79,-27;PENBLUE;TURNT
O0;FILL10
580 E:
590 *TREE GR:PENRED;GOTO-30,-2;TURNTO9
0;DRAW4;TURN90;DRAW8;TURN90;DRAW4;TURN
90;FILL8
600 GR:PENYELLOW;TURNTO270;8(TURN45;DR
AW4)
610 GR:3(TURN45;FILL4)
620 GR:PENRED;GOTO-30,3;GOTO-27,5;GOTO
-26,0;GOTO-29,1;GOTO-24,3;GOTO-32,5;GO
TO-30,6;GOTO-31,0
630 E:
640 *RIVER C:#C=5
650 GR:GOTO#P,-18;TURNTO0
660 *LOP GR:PENBLUE;TURN135;GO3;TURN45
;GO3;TURN45;GO3;TURNTO0;GO7;TURN90;GO2
```

```
670 GR:PENERASE;TURN45;GO3;TURN45;GO3;
TURN45;GO3;TURNTO0;GO7;PENUP;GOTO%X+38
,%Y
680 C:#C=#C-1
690 J(#C<>0):*LOP
700 C:#P=#P+2
710 C(#P>-47):#P=-85
720 E:
730 *DOOROPEN GR:PENYELLOW;GOTO55,-10;
TURNTO0;DRAW10
740 SO:18
750 U:*RIVER
760 GR:GOTO55,-10;PENERASE;DRAW9;PENYE
LLOW;GOTO54,-10;DRAW10
770 SO:22
780 U:*RIVER
790 GR:GOTO54,-10;PENERASE;DRAW9;PENYE
LLOW;GOTO53,-10;DRAW10
800 SO:26
810 U:*RIVER
820 GR:GOTO53,-10;PENERASE;DRAW9;PENYE
LLOW;GOTO52,-10;DRAW10
830 SO:30
840 U:*RIVER
850 E:
860 *EXIT C:#X=54
870 C:#Y=-8
880 U:*DRAW
890 C:#W=7
900 U:*WAIT
910 T:[CLEAR]  "Ah, a beautiful day to
 sit under the apple tree..."
920 C:#W=10
930 U:*WAIT
940 T:[CLEAR]
950 U:*ERASE
960 C:#X=#X-2
970 C:#Y=-12
980 *STEP U:*DRAW
990 U:*RIVER
1000 U:*ERASE
1010 C:#X=#X-4
1020 J(#X<>-20):*STEP
1030 *STEPUP U:*DRAW
1040 U:*RIVER
1050 U:*ERASE
1060 C:#X=#X-2
1070 C:#Y=#Y+2
1080 J(#X<>-24):*STEPUP
1090 U:*DRAW
1100 E:
1110 *DRAW GR:PENRED;GOTO#X,#Y;GOTO#X+
1,#Y;GOTO#X+1,#Y-1;GOTO#X,#Y-1
1120 SO:20
1130 PA:2
1140 SO:0
1150 E:
1160 *ERASE GR:PENERASE;GOTO#X,#Y;GOTO
#X+1,#Y;GOTO#X+1,#Y-1;GOTO#X,#Y-1
1170 E:
1180 *APPLE GR:PENERASE;GOTO-24,3
```

```
1190 SO:3,5
1200 PA:10
1210 SO:0,0
1220 U:*RIVER
1230 GR:PENRED;GOTO-24,3
1240 U:*RIVER
1250 GR:PENERASE;GOTO-24,3
1260 SO:3,5
1270 PA:10
1280 SO:0,0
1290 U:*RIVER
1300 U:*RIVER
1310 GR:PENYELLOW;GOTO-24,3;PENRED;GOT
O-23,2
1320 U:*RIVER
1330 GR:PENYELLOW;GOTO-23,2;PENRED;GOT
O-23,1
1340 U:*RIVER
1350 GR:PENYELLOW;GOTO-23,1;PENRED;GOT
O-23,0
1360 C:#Y=0
1370 *FALL U(#Y\2=1):*RIVER
1380 GR:PENERASE;GOTO-23,#Y
1390 C:#Y=#Y-1
1400 GR:PENRED;GOTO-23,#Y
1410 J(#Y<>-7):*FALL
1420 SO:25
1430 PA:2
1440 SO:0
1450 E:
1460 *BOUNCE GR:PENERASE;GOTO-23,-7;PE
NRED;GOTO-22,-6
1470 U:*RIVER
1480 GR:PENERASE;GOTO-22,-6;PENRED;GOT
O-21,-7
1490 U:*RIVER
1500 GR:PENERASE;GOTO-21,-7;PENRED;GOT
O-21,-8
1510 U:*RIVER
1520 GR:PENERASE;GOTO-21,-8;PENRED;GOT
O-21,-9
1530 SO:20
1540 PA:2
1550 SO:0
1560 U:*RIVER
1570 GR:PENERASE;GOTO-21,-9;PENRED;GOT
O-20,-8
1580 U:*RIVER
1590 GR:PENERASE;GOTO-20,-8;PENRED;GOT
O-19,-9
1600 U:*RIVER
1610 GR:PENERASE;GOTO-19,-9;PENRED;GOT
O-19,-10
1620 SO:15
1630 PA:2
1640 SO:0
1650 U:*RIVER
1660 GR:PENERASE;GOTO-19,-10;PENRED;GO
TO-18,-10
1670 SO:5
1680 PA:1
```



```
1690 SO:0
1700 E:
1710 *WAIT U:*RIVER
1720 C:#W=#W-1
1730 J(#W<>1):*WAIT
1740 E:
1750 *TITLE T:[CLEAR]
1760 C:@B1374=1
1770 WRITE:S
1780 C:@B84=8
1790 WRITE:S,       "NEWTON"
1800 C:@B84=14
1810 C:@B1374=2
1820 WRITE:S,     BY ANDY HALL
1830 PA:200
1840 E:
```



Computing in ATARI PILOT—The C: Command

As you've seen, all graphics statements in ATARI PILOT turtle graphics must begin with the GR: command. In the same way all statements in ATARI PILOT concerning the manipulation of numbers must begin with the compute command C:.

In addition the only numbers that ATARI PILOT recognizes are the positive and negative integers:

0, 1, 2, 3, 4, 5 . . .

–1, –2, –3, –4, –5. . . .

You will see in the chapter on math and logic games that this is not as much of a limitation as you might think. Addition, subtraction, and multiplication are performed in the usual way. However, division always results in giving the integer answer and dropping the remainder unless you use the modular arithmetic, the explanation of which is beyond the scope of this book. Thus

2 + 3 = 5

3 – 2 = 1

2 * 3 = 6

but

5/2 will result in 2 (remainder 1 dropped)
5/3 will result in 1 (remainder 2 dropped)

In all of the drill-and-practice programs in the math and

logic chapter of the book, we've used modular arithmetic so that division can be done as usual.

C: works with number variables, which are designated by # and a variable name, for example, #X, #Y, or #A. It takes the form

C:#X=#Y+ 10 (i)

or

C:#X=#Y− 5 (ii)

or

C:#X=#Y+#Z (iii)

or

C:#X=#X+1 (iv)

Thus, (i) adds 10 to any #Y, (ii) subtracts 5, and (iii) adds two numbers, #Y and #Z. In many programs (iv) is particularly important. It is usually used within a module and says that each time it's used the original value of #X should be increased by 1.

This simple program multiplies a number that you input by 10:

```
10 T:GIVE ME A NUMBER AND I'LL
20 T:MULTIPLY IT BY 10
30 A:#X
40 C:#Y=#X*10
50 T:#Y
```

This example uses C:#X=#X+1:

```
10 C:#X=1
20 *GROWTH
30 C:#X=#X+1
35 T:#X
40 J:*GROWTH
```

If you run it, the computer will display a list of numbers beginning with 2 and increasing by 1 (2, 3, 4, 5, 6, . . .) until you press the [BREAK] key. The computer sets up an infinite loop, which is not useful in the middle of a program as it will never stop. However, there is a simple way to determine a specific number of loops in ATARI PILOT. PILOT commands can be used in conjunction with these signs:

< less than

<= less than or equal to

= equal to

> greater than

>= greater than or equal to

to determine conditions under which certain operations are performed. The chart illustrates these commands:

C:	C(#X<#Y):	C(#X<=#Y):	C(#X=#Y):	C(#X>#Y):	C(#X>=#Y)
J:	J(#X<#Y):	J(#X<=#Y):	J(#X=#Y):	J(#X>#Y):	J(#X>=#Y)
U:	U(#X<#Y):	U(#X<=#Y):	U(#X=#Y):	U(#X>#Y):	U(#X>=#Y)
T:	T(#X<#Y):	T(#X<=#Y):	T(#X=#Y):	T(#X>#Y):	T(#X>=#Y)
M:	M(#X<#Y):	M(#X<=#Y):	M(#X=#Y):	M(#X>#Y):	M(#X>=#Y)
GR:	GR(#X<#Y):	GR(#X<=#Y):	GR(#X=#Y):	GR(#X>#Y):	GR(#X>=#Y)
SO:	SO(#X<#Y):	SO(#X<=#Y):	SO(#X=#Y):	SO(#X>#Y):	SO(#X>=#Y)
PA:	PA(#X<#Y):	PA(#X<=#Y):	PA(#X=#Y):	PA(#X>#Y):	PA(#X>=#Y)

This program illustrates the use of the J(#X<#Y): command:

```
10 C:#X=1
20 *GROWTH
30 C:#X=#X+1
35 T:#X
40 J(#X<20):*GROWTH
```

Line 40 says that if #X is less than 20, then the program should jump to *GROWTH. When #X gets bigger than 20, the program will stop. This program, instead of printing an infinite list, prints 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 . . . . . Note that in order for the program to work, *#X must have an initial value,* which is what line 10 does. If line 10 read

```
10 C:#X=15,
```

the program would print

```
16,17,18,19,20
```

Try the following modifications of the program.

- Make it print out 14,15,16,17,18.
- Make it print out 1,2,3,4,5,6,7.

- Make it print all even numbers up to 20.
- Make it print all odd numbers up to 20.

Here's an example of how numerical conditions can determine graphics. The two modules, *HOUSE and *3DHOUSE, were already presented. Here they are combined in a program. If you type in a number less than 10, you will see *HOUSE. If you type in 10 or a larger number, you will see *3DHOUSE.

```
10 T:GIVE ME A NUMBER
20 A:#X
30 J(#X<10):*HOUSE
40 J(#X>=10):*3DHOUSE
50 E:
500 *3DHOUSE
510 GR:2(DRAW 5;TURN 90;DRAW 2;TURN 90)
520 GR:GOTO -5,10;TURNTO 0;4(DRAW 2;TURN 90)
530 GR:GOTO 7,10;TURNTO 0;4(DRAW 2;TURN90)
540 GR:PEN RED;GOTO -10,0;TURNTO 0;4(DRAW 22;TURN 90)
550 GR:GOTO 12,22;DRAWTO 1,30;FILLTO -10,22
560 GR:GOTO 12,0;DRAWTO 20,15;DRAWTO 20,37
570 GR:GOTO 12,22;DRAWTO 20,37
580 GR:GOTO 1,30;DRAWTO 9,42;DRAWTO 20,37
590 E:
600 *HOUSE
610 GR:2(DRAW 5;TURN 90;DRAW 2;TURN 90)
620 GR:GOTO -5,10;TURNTO 0;4(DRAW 2;TURN 90)
630 GR:GOTO 7,10;TURNTO 0;4(DRAW 2;TURN90)
640 GR:PEN RED;GOTO -10,0;TURNTO 0;4(DRAW 22;TURN 90)
650 GR:GOTO 12,22;DRAWTO 1,30;FILLTO -10,22
```

Another feature of C: is that it allows you to select random numbers, that is, numbers chosen by chance. This can be a powerful programming tool since it allows you to simulate coin tossing or dice throwing and to create programs that change each time they are run. A random number #X is chosen when you type in the following command:

C:#X=?

the ? is the ATARI PILOT symbol for random number. In this program the computer selects a random number. If it is less than or equal to 50, you get *HOUSE; if it is less than 50, you get *3DHOUSE.

```
10 R:RANDOM NUMBER GUESS
20 C:#X=?
30 J(#X<=50):*HOUSE
40 J(#X>50):*3DHOUSE
50 E:
500 *3DHOUSE
510 GR:2(DRAW 5;TURN 90;DRAW 2;TURN 90)
520 GR:GOTO -5,10;TURNTO 0;4(DRAW 2;TURN 90)
```

```
530 GR:GOTO 7,10;TURNTO 0;4(DRAW 2;TURN90)
540 GR:PEN RED;GOTO -10,0;TURNTO 0;4(DRAW 22;TURN 90)
550 GR:GOTO 12,22;DRAWTO 1,30;FILLTO -10,22
560 GR:GOTO 12,0;DRAWTO 20,15;DRAWTO 20,37
570 GR:GOTO 12,22;DRAWTO 20,37
580 GR:GOTO 1,30;DRAWTO 9,42;DRAWTO 20,37
590 E:
600 *HOUSE
610 GR:2(DRAW 5;TURN 90;DRAW 2;TURN 90)
620 GR:GOTO -5,10;TURNTO 0;4(DRAW 2;TURN 90)
630 GR:GOTO 7,10;TURNTO 0;4(DRAW 2;TURN90)
640 GR:PEN RED;GOTO -10,0;TURNTO 0;4(DRAW 22;TURN 90)
650 GR:GOTO 12,22;DRAWTO 1,30;FILLTO -10,22
660 E:
```

If you want to choose a random number from a limited set of numbers (for example, 1 from 6 as in dice or 1 from 2 as in coin tossing), the command is

C:#X=?\N

Notice that \ is the opposite of / the symbol for division. It is located on the + key. [SHIFT][+] gives you \.

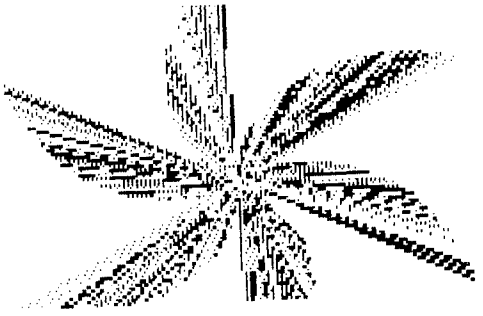
N stands for the upper limit for selecting a random number. ?\2 selects from 0 and 1; ?\6 selects from 0,1,2,3,4,5.

Remember that the upper limit itself is not included in the set of numbers chosen from. This program randomly draws a circle or square using: C:#X=?\2 at line 10.

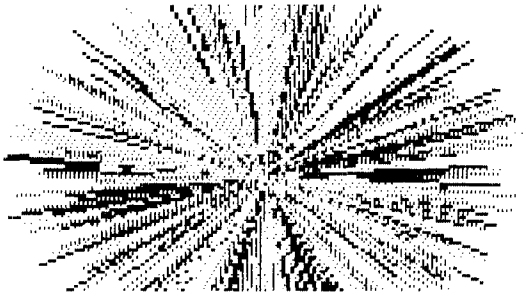
```
10 C:#X=?\2
20 J(#X=0):*CIRCLE
30 J(#X=1):*SQUARE
40 E:
50 *CIRCLE
60 GR:36(DRAW 5;TURN 10)
70 E:
80 *SQUARE
90 GR:4(DRAW 25;TURN 90)
100 E:
```

Programs and Patterns. Here are a number of programs that generate beautiful patterns using C: and the other ATARI PILOT commands.

```
2 R:          *** FLOWER ***
4 R:        COPYRIGHT (C) 1980
6 R:       BY HARRY B. STEWART
100 *LOOP
110 C:#C=?\4
120 GR(#C=0):PEN ERASE
130 GR(#C=1):PEN YELLOW
140 GR(#C=2):PEN RED
150 GR(#C=3):PEN BLUE
160 C:#L=#L+1\60
170 GR:GOTO 0,0;DRAW #L;TURN 1
180 J:*LOOP
```

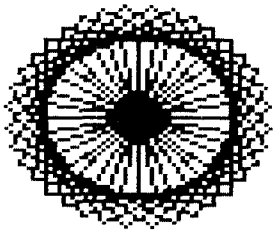


```
2 R:        *** STARBURS ***
4 R:      COPYRIGHT (C) 1980
6 R:     BY HARRY B. STEWART
100 *LOOP
110 C:#C=?\4
120 GR(#C=0):PEN ERASE
130 GR(#C=1):PEN YELLOW
140 GR(#C=2):PEN RED
150 GR(#C=3):PEN BLUE
160 GR:GOTO 0,0;DRAW ?\60;TURN 1
170 J:*LOOP
```



```
2 R:         *** ROTATE ***
4 R:       BY TED KAHN (1980)
6 R:THIS PROGRAM ROTATES A SQUARE IN N
 DEGREE INCREMENTS. TRY DIFFERENT SIZE
S & DEGREE ROTATIONS.
8 R:
100 GR:CLEAR
110 *LOOP
120 T:ENTER THE SIZE OF THE SQUARE:\
130 A:#S
140 T:NOW ENTER THE ROTATION INCREMENT
 (DEGREES): \
150 A:#I
160 T:WHAT COLOR (RED, YELLOW, OR BLUE
)?
170 A:
180 M:R,Y,B
190 JM:*RED,*YELLOW,*BLUE
200 *DISPLAY
210 C:#T=360/#I+1
220 GR:CLEAR;GOTO 0,0;TURNTO 0
```

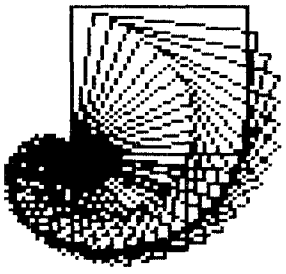
```
230 GR:#T(4(DRAW #S;TURN 90);TURN #I)
240 J:*LOOP
250 *RED GR:PEN RED
260 J:*DISPLAY
270 *YELLOW GR:PEN YELLOW
280 J:*DISPLAY
290 *BLUE GR:PEN BLUE
300 J:*DISPLAY
310 E:
```



```
NOW ENTER THE ROTATION INCREMENT
(DEGREES): 5
WHAT COLOR (RED, YELLOW, OR BLUE)?
YELLOW
```

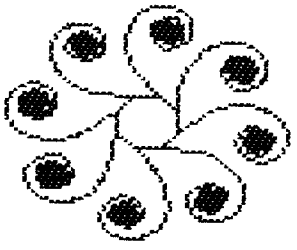
Notice a multiple match at lines 170, 180, and 190. Line 170 asks you to type in a color. Line 180 then gives three match choices, each of which correlates to a module in line 190. This multiple matching is another way that the M: command can be used.

```
2 R:          *** SHROTATE ***
4 R:   TED KAHN & HARRY STEWART (1980)
6 R:ROTATE & SHRINK A SQUARE IN N DEGR
EE INCREMENTS (WITH SOUND EFFECTS)
100 GR:CLEAR
110 *LOOP
120 SO:#S,#P,#I
130 T:[CLEAR]ENTER STARTING SQUARE SIZ
E:\
140 A:#S
150 T:NOW ENTER THE PERCENT SHRINKAGE:
\
160 A:#P
170 T:FINALLY, ENTER THE ROTATION INCR
EMENT:\
180 A:#I
190 GR:CLEAR;GOTO 0,0;TURNTO 0
200 *L1 GR:4(DRAW #S;TURN 90);TURN #I
210 C:#D=#S*#P/100
220 C(#D=0):#D=1
230 C:#S=#S-#D
240 J(#S):*L1
250 J:*LOOP
```



```
ENTER STARTING SQUARE SIZE:40
NOW ENTER THE PERCENT SHRINKAGE:5
FINALLY, ENTER THE ROTATION INCREMENT:
7
```

```
2 R:      *** FLOWER #1 ***
4 R:      COPYRIGHT (C) 1982
6 R:         BY ANDY HALL
100 *FLOWER1
110 GR:PENERASE;TURNTO0;GOTO0,0
120 T:HOW MANY PETALS (0 TO EXIT)? \
130 A:#N
140 E(#N=0):
150 C:#C=0
160 C:#A=0
170 C:#D=180-(360/#N)
180 GR:PENYELLOW;CLEAR;GOTO0,5
190 *FLLOOP GR:DRAW3;TURN#A
200 C:#A=#A+2
210 J(#A<360):*FLLOOP
220 C:#C=#C+1
230 GR:TURN#D
240 C:#A=0
250 J(#C<#N):*FLLOOP
260 J:*FLOWER1
```



```
HOW MANY PETALS (0 TO EXIT)?
```

POKES and Graphics. Here are a number of other short programs that produce beautiful visual effects. They use POKE numbers (see Appendix II for an explanation). You can copy and use them even if you don't care to examine the POKES that they use.

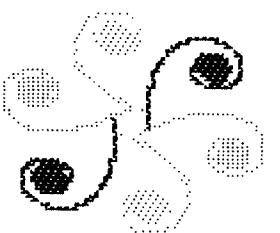
The first is a more complex flower petal arrangement; the second a series of fans that you can generate, the third a sphere, and the fourth, and perhaps most impressive, a color display of continually changing patterns created by rotating lines through different angles.

```
2 R:      *** FLOWER #2 ***
4 R:      COPYRIGHT (C) 1982
6 R:         BY ANDY HALL
100 *FLOWER2
110 GR:PENERASE;GOTO0,0;PENYELLOW
120 T:NUMBER OF PETALS (0 TO QUIT)? \
130 A:#N
140 E(#N=0):
150 C:#D=180-(360/#N)
160 GR:CLEAR;GOTO7,5;TURNTO0
170 C:#C=0
180 C:#A=0
190 *FLLOOP2 GR:DRAW3;TURN#A
200 C:#A=#A+2
210 C:@B1363=@B1363\3+1
220 J(#A<360):*FLLOOP2
230 C:#C=#C+1
240 GR:TURN#D
250 C:#A=0
260 J(#C<#N):*FLLOOP2
270 J:*FLOWER2
```

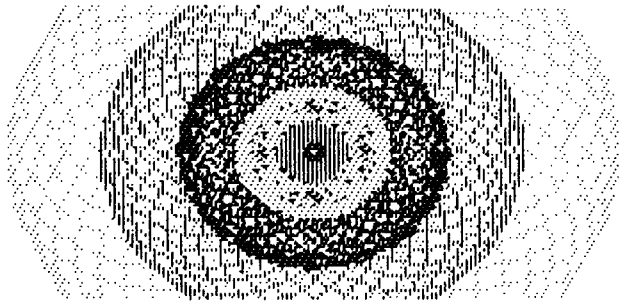
```
NUMBER OF PETALS (0 TO QUIT)?
```

```
2 R:           *** FAN ***
4 R:        COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
100 *FAN
110 GR:CLEAR
120 T:NUMBER OF FANBLADES? \
130 A:#N
140 J(#N=0):*FAN
150 C:#D=180-(360/#N)
160 C:#C=0
```

```
170 GR:CLEAR;TURNTO0;PENYELLOW;GOTO7,4
180 C:#A=0
190 *FLOOP GR:DRAW2;TURN#A
200 C:#A=#A+1
210 J(#A<360):*FLOOP
220 GR:TURN#D
230 C:@B1363=@B1363\3+1
240 C:#A=0
250 C:#C=#C+1
260 J(#C<#N):*FLOOP
270 T:        HIT ANY KEY TO EXIT.
280 C:@B764=255
290 *ROTATE C:#C=@B710
300 C:@B710=@B709
310 C:@B709=@B708
320 C:@B708=#C
330 PA:10
340 J(@B764<>255):*RESTORE
350 J:*ROTATE
360 *RESTORE C:@B764=255
370 C:@B708=40
380 C:@B709=202
390 C:@B710=148
400 E:
```

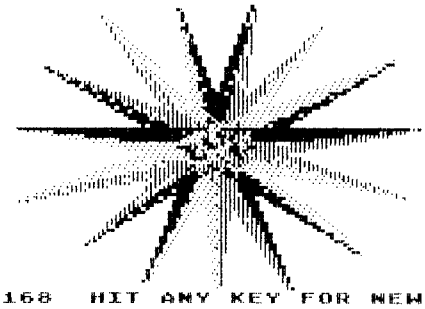


```
2 R:          *** SPHERE ***
4 R:        COPYRIGHT (C) 1982
6 R:          BY ANDY HALL
100 *SPHERE
110 GR:PENBLUE;GOTO0,8;CLEAR;TURNTO0
120 C:#A=10
130 C:#C=6
140 C:#S=6
150 *SLOOP GR:TURN#A
160 U:*CIRCLE
170 J(%A<>0):*SLOOP
180 C:#C=#C-1
190 C:@B1363=@B1363\3+1
200 J(#C>0):*SLOOP
210 GR:PENBLUE;GOTO0,8
220 T:          TYPE RETURN TO EXIT. \
230 A:
240 E:
250 *CIRCLE C:#R=#C*#S
260 C:#D=360/#R
270 GR:#R(DRAW#C;TURN#D)
280 E:
```

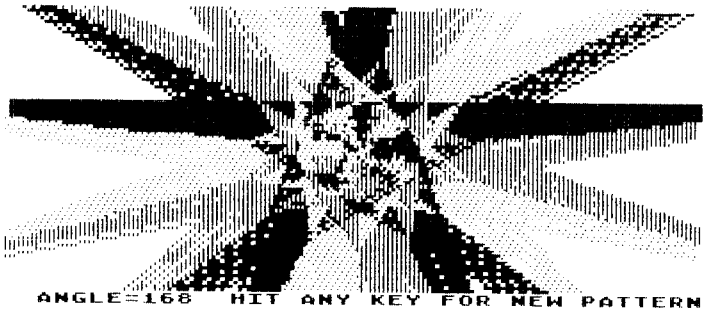


TYPE RETURN TO EXIT.

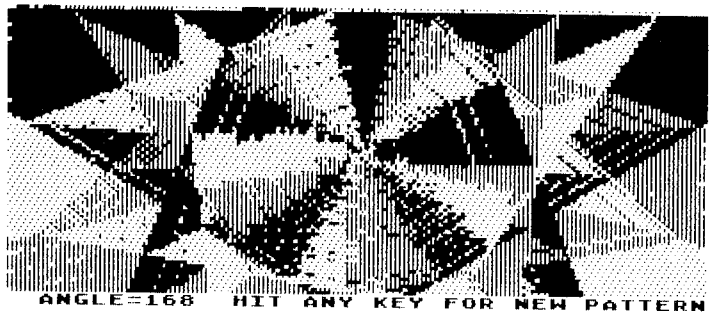
```
2 R:      *** KALEIDOSCOPE ***
4 R:       COPYRIGHT (C) 1982
6 R:          BY ANDY HALL
100 *ANGLE T: WHAT ANGLE (0 FOR RANDOM
)? \
110 A:#A
120 C(#A=0):#A=?\360
130 GR:PENYELLOW;CLEAR;GOTO0,8;TURNTO9
0
140 T:ANGLE=#A      HIT ANY KEY TO CHAN
GE.
150 C:#L=2
160 *LOOP GR:DRAW#L;TURN#A
170 C:@B1363=@B1363\3+1
180 C:#L=#L+1
190 J(@B764=255):*LOOP
200 C:@B764=255
210 J:*ANGLE
```

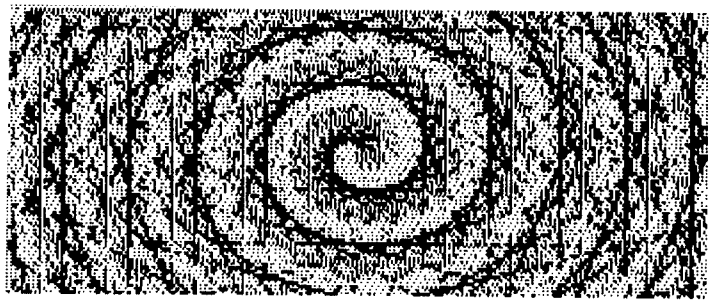


ANGLE=168 HIT ANY KEY FOR NEW PATTERN

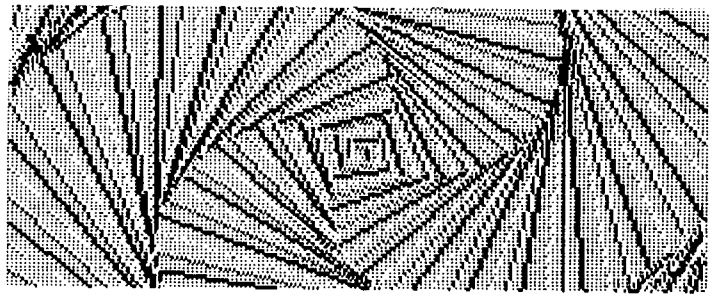


ANGLE=168 HIT ANY KEY FOR NEW PATTERN





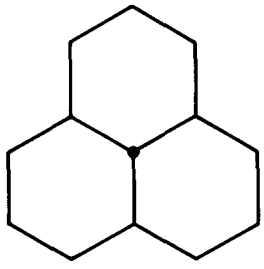
ANGLE=234



ANGLE=89

Tessellations

Anyone who has seen a tiled bathroom or a floor covered with linoleum pentagons or octagons should have no trouble understanding a tessellation. Tessellations are made by piecing together repetitions of the same shape so that they would cover an infinite plane. The most common tile is six-sided.



The illustration shows three tiles fitted together. You could cover the plane by adding more and more identical tiles indefinitely. Of course, bathrooms and kitchens are not infinite in size; some tiles must be cut at the edges. But a tessellation as an idea goes on forever. The Dutch artist M. C. Escher used tessellations in some of his most intriguing and interesting paintings.

ATARI PILOT turtle graphics makes it possible for you to create all kinds of interesting tessellations. We'll start simply and end with a complex program that combines sound, color, and tessellations.

The first step toward creating tessellations is to figure out how to repeat a form across the screen. With a combination of the C: command and graphics commands, this can be done simply, as in this program which produces the following pattern across the screen.

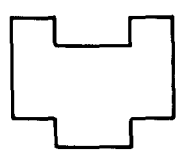
```
10 C:#X=-79
20 C:#Y=0
30 *PLOT
40 GR:GOTO #X,#Y:TURNTO 90:DRAW 5
50 C:#X=#X+20
55 C:#Y=0
60 J(#X<80):*PLOT
```

Note that lines 10 and 20 set the X and Y coordinates at −79 and 0, respectively. The module *PLOT essentially instructs the turtle to go to −79,0, turn to 90°, and draw a line 5 units long. Then it adds 20 to the X coordinate, keeps Y the same, and jumps back to plotting the new #X,#Y (the first time around the loop it will plot −59,0, the second time −39,0, etc.) When #X is larger than 80 (and therefore goes off the screen), the program stops.

To make the program a bit more complex, the DRAW command on line 40 can be replaced by commands that result in a square, as in the following program:

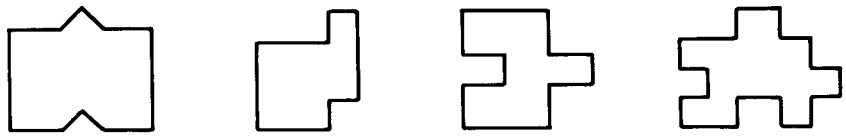
```
10 C:#X=-79
20 C:#Y=0
30 *PLOT
40 GR:GOTO #X,#Y;TURNTO 90;4(DRAW 20;TURN 90)
50 C:#X=#X+20
55 C:#Y=0
60 J(#X<80):*PLOT
```

This is beginning to look more like a tessellation. If the squares could begin at the bottom left or top left of the screen and cover the whole screen, we'd have a simple tessellation. That can be done by changing the value of #Y each time that a line is completed so that the program changes rows as well as columns. Instead of demonstrating this with a square, we'll use a more interesting form.



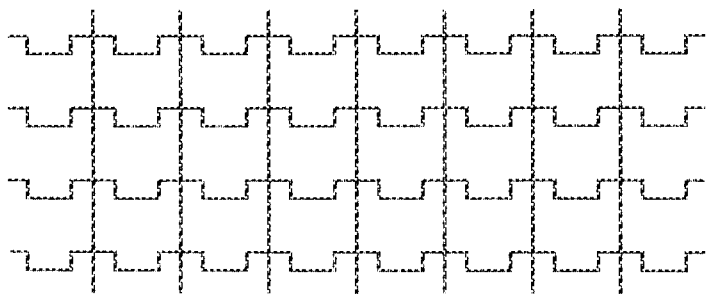
```
5 R:TESSMOD#1
10 GR:2(TURN 90;DRAW 5);TURN-90;DRAW 10
20 GR:TURN -90;DRAW 5;TURN 90;DRAW 5
30 GR:TURN -90;DRAW 20;TURN -90;DRAW 5
40 GR:TURN -90;DRAW 5;TURN 90;DRAW 10
50 GR:TURN 90;DRAW 5;TURN -90;DRAW 5
60 GR:TURN -90;DRAW 20
```

This form was created by taking away a piece of the top of the square and adding it to the bottom. Regular additions and subtractions within a regular form retain the ability to tessellate the plane and provide you with an infinite variety of potential tessellations.



This tessellation program uses the first tessmodule.

```
100 R:TESS #1
110 C:#X=-80
120 C:#Y=-40
130 GR:CLEAR:TURNTO 0
140 *LOOP
150 U:*TESS
160 C:#Y=#Y+20
170 J(#Y<41):*LOOP
180 C:#X=#X+20
190 C:#Y=-40
200 J(#X<61):*LOOP
210 E:
220 *TESS
230 GR:GOTO #X,#Y:TURNTO 0
240 GR:2(TURN90:DRAW5):TURN-90:DRAW10
250 GR:TURN -90:DRAW 5:TURN 90:DRAW 5
260 GR:TURN -90:DRAW 20:TURN -90:DRAW 5
270 GR:TURN -90:DRAW 5:TURN 90:DRAW 10
280 GR:TURN 90:DRAW 5:TURN -90:DRAW 5
290 GR:TURN -90:DRAW 20
300 E:
```

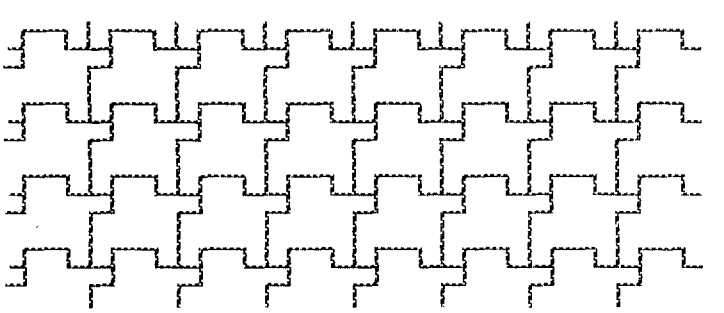


Notice that there are two modules in the program: *LOOP and *TESS. *LOOP controls the position of the turtle and *TESS draws the tessellating form. It is worth studying how #X and #Y change in this program. Remember that we are

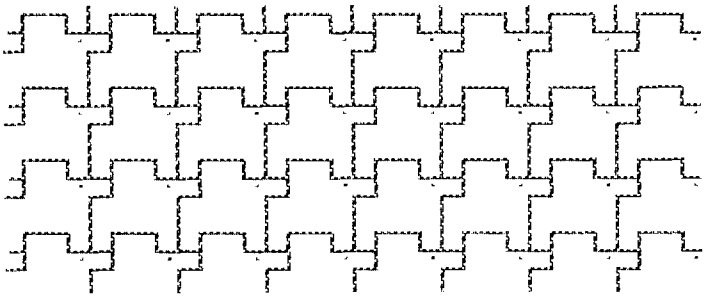
still using the 20 × 20 basic square that we used to draw squares across the screen. That's why 20 appears in lines 160 and 180. The program begins at −80, −40 at the bottom left of the screen and moves across the screen to the right. Line 170 jumps the program 20 rows higher just before #X is to go off the screen. This may seem a bit tricky, but if you copy the program and use your own *TESS Modules, you should catch on quickly. We could have chosen to draw a form every 10 or 25 units (in lines 160 or 180) if we wanted. The former would have made the tessellation units smaller, the latter larger. Play with all these variables and see what you come up with.

With a little fiddling around and dressing up you can begin to approximate animal or people forms for your tessellations. Here are two stages of an elephant/horse/camel(?) that we developed.

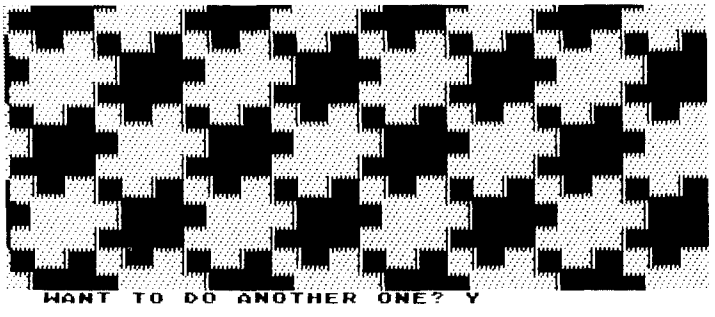
```
100 R:TESS #2
110 C:#X=-80
120 C:#Y=-40
130 GR:TURNTO 0
140 *LOOP U:*TESS
150 C:#Y=#Y+20
160 J(#Y<41):*LOOP
170 C:#X=#X+20
180 C:#Y=-40
190 J(#X<61):*LOOP
200 E:
210 *TESS
220 GR:GOTO #X,#Y;TURNTO 0
230 GR:TURN 90;DRAW 5;TURN -90;DRAW 5
240 GR:TURN 90;DRAW 10;TURN 90;DRAW 5;
TURN -90;DRAW 5
250 GR:TURN -90;DRAW 15;TURN 90;DRAW 5
260 GR:2(TURN -90;DRAW 5);DRAW 5;TURN
90;DRAW 5
270 GR:2(TURN -90;DRAW 10);TURN 90;DRA
W 5
280 GR:TURN -90;DRAW 15
290 E:
```

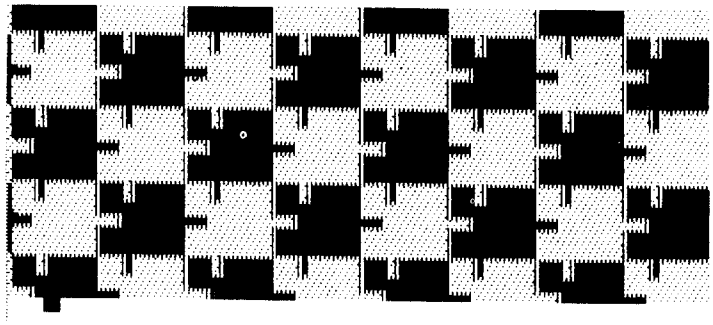


```
100 R:TESS #3
110 C:#X=-80
120 C:#Y=-40
130 GR:TURNTO 0
140 *LOOP U:*TESS
150 C:#Y=#Y+20
160 J(#Y<41):*LOOP
170 C:#X=#X+20
180 C:#Y=-40
190 J(#X<61):*LOOP
200 E:
210 *TESS
220 GR:GOTO #X,#Y;TURNTO 0
230 GR:TURN 90;DRAW 5;TURN -90;DRAW 5
240 GR:TURN 90;DRAW 10;TURN 90;DRAW 5;
TURN -90;DRAW 5
250 GR:TURN -90;DRAW 15;TURN 90;DRAW 5
260 GR:2(TURN -90;DRAW 5);DRAW 5;TURN
90;DRAW 5
270 GR:2(TURN -90;DRAW 10);TURN 90;DRA
W 5
280 GR:TURN -90;DRAW 15
290 GR:TURN -90;GO18;TURN-90;GO18
300 E:
```



Here's a more complex program for you to copy and study. It generates an indefinite number of tessellations using the random number function and several other advanced features of PILOT. It also incorporates color changes and fill commands, which you should try out with your own tessellations.





```
2 R:    ***TESSELLATIONS***
3 R:    COPYRIGHT (C) 1982
4 R:       BY ANDY HALL
100 *AGAIN U:*INITIALIZE
110 C:#X=-80
120 C:#Y=-40
130 C:#S=1
140 C:#H=20-#A-#C
150 C:#I=20-#B-#D
160 GR:CLEAR
170 U:*TESSDRAW
180 *Y C:#Y=#Y+20
190 U:*TESSDRAW
200 J(#Y<41):*Y
210 J:*X
220 *X C:#X=#X+20
230 C:#Y=-40
240 C:#S=1-#S
250 U:*TESSDRAW
260 J(#X<80):*Y
270 T:[CLEAR]WANT TO DO ANOTHER ONE? \
280 A:
290 M:N
300 GRN:QUIT
310 JN:*AGAIN
320 E:
330 *TESSDRAW GR:GOTO#X,#Y;TURNTO90;DR
AW#A;TURN90;DRAW5;TURN-90;DRAW#C
340 GR:TURN-90;DRAW5;TURN90;DRAWTO#X+2
0,#Y
350 GR:TURN-90;DRAW#B;TURN90;DRAW5
360 GR:TURN-90;DRAW#D;TURN-90;DRAW5;TU
RN90;DRAWTO#X+20,#Y+20
370 GR:TURN-90;DRAW#H;TURN-90;DRAW5;TU
RN90;DRAW#C
380 GR:TURN90;DRAW5;TURN-90;DRAWTO#X,#
Y+20
390 GR:TURN-90;DRAW#I;TURN-90;DRAW5;TU
RN90;DRAW#D
400 GR:TURN90;DRAW5;TURN-90;DRAWTO#X,#
Y
410 C:#S=1-#S
420 R:BEGIN FILL ROUTINE
430 GR:PENUP;TURN-90;GO#A+1;TURN90;GO5
;TURN180
440 U:*COLOR
450 GR:FILL19;PENUP;GOTO#X+1,#Y
```

```
460 U:*COLOR
470 GR:FILL#B-1;TURN90;GO5;TURN-90;FIL
L#D+1;PENUP;GOTO#X+1,#Y+20;TURN180
480 U:*COLOR
490 GR:FILL#I-1;PENUP;GOTO#X+#A+#C+1,#
Y+20
500 U:*COLOR
510 GR:FILL5
520 GR:PENYELLOW
530 E:
540 *COLOR GR(#S=0):PENRED
550 GR(#S=1):PENBLUE
560 E:
570 *INITIALIZE T:[CLEAR]
580 POS:2,3
590 T:YOU CAN DESIGN YOUR OWN TESSELLA
TION.
600 R: BE CAREFUL ABOUT SPACES AND THI
NGS WHEN TYPING IN THESE NEXT LINES.
610 R: REMEMBER, [ESC][UP] INVOLVES
HITTING "ESC" THREE TIMES, "UP ARROW"
ONCE.
620 POS:2,6
630 T:            [QRRE]    [QRE]
640 T:            ¦  ¦    ¦ ¦
650 T:           [ESC][UP][ZREZRRRC ZRE]
660 T:          D¦  ¦          ¦
670 T:           [ESC][DOWN][QRC]      [
QRC][ESC][UP]
680 T:            ¦          ¦  ¦B
690 T:            [ZRRE    QRC]  [ESC][DO
WN]
700 T:            [ESC][LEFT]-[ESC][RIGH
T]¦    ¦
710 T:             A [ZRRRC]
720 T:               [ESC][LEFT]---[ESC]
[RIGHT]
730 T:                  C
740 *A POS:2,19
750 T:WHAT LENGTH FOR "A" (6-9)? \
760 A:#A
770 J((#A<6)+(#A>9)):*A
780 *B POS:2,19
790 T:WHAT LENGTH FOR "B" (6-9)?      [
4(LEFT)]\
800 A:#B
810 J((#B<6)+(#B>9)):*B
820 *C POS:2,19
830 C:#M=21-(2*#A)
840 T:WHAT LENGTH FOR "C" (2-#M)?
[4(LEFT)]\
850 A:#C
860 J((#C<2)+(#C>#M)):*C
870 *D POS:2,19
880 C:#M=21-(2*#B)
890 T:WHAT LENGTH FOR "D" (2-#M)?
[4(LEFT)]\
900 A:#D
910 J((#D<2)+(#D>#M)):*D
920 E:
```

Here are three short programs which allow you to experiment with sequences of sounds and trills. The third program provides you with a graphic representation of the sounds you type in.

1

```
2 R:          *** MUSIC ***
4 R:           BY TED KAHN
100 T:TYPE IN A SEQUENCE OF NOTES (1-3
1), SEPARATED BY COMMAS (,):
110 A:$NOTES
120 *LOOP
130 MS:!,
140 A:#A=$LEFT
150 SO:#A
160 PA:30
170 AY:=$RIGHT
180 JY:*LOOP
190 A:#A=$RIGHT
200 SO:#A
210 PA:30
```

2

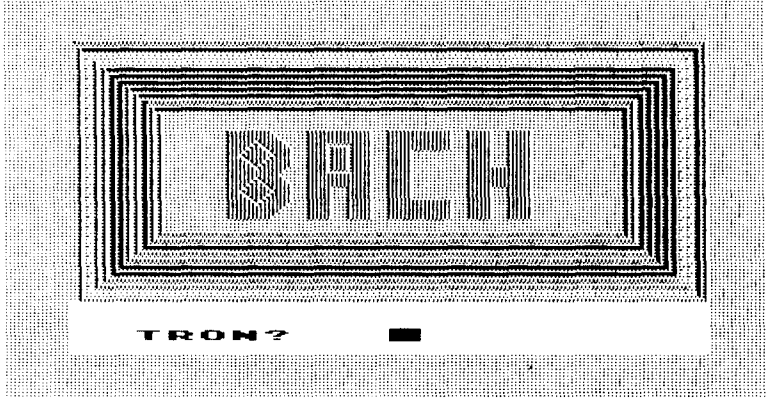
```
2 R: *** TRILL2 (WITH GRAPHICS) ***
4 R:       BY TED M. KAHN, 1982
6 R:TRILL1+GRAPHICS (EXPERIMENT WITH D
IFFERENT SPEEDS & INTERVALS TO GET VAR
IOUS EFFECTS)
8 R:
100 *TRILL2
110 VNEW
120 T:
130 T:ENTER A STARTING NOTE NUMBER (1-
30):
140 A:#A
150 T:NOW ENTER AN INTERVAL (1-12):\
160 A:#I
170 C:#B=#A+#I
180 T:FINALLY, ENTER A SPEED (1-30):\
190 A:#P
200 *PLAY
210 SO:#A
220 GR(#T<=1):CLEAR;PEN UP;GOTO -75+(#
T*#P),#A;PEN RED
230 GR(#T>1):PEN RED;DRAWTO -75+(#T*#P
),#A
240 PA:#P
250 GR:PEN YELLOW;DRAWTO -75+((#T+1)*#
P),#B
260 SO:#B
270 PA:#P
280 C:#T=#T+2
290 J(#T<16):*PLAY
300 J:*TRILL2
310 E:
```

```
ENTER A STARTING NOTE NUMBER (1-30):
5
NOW ENTER AN INTERVAL (1-12):6
FINALLY, ENTER A SPEED (1-30):7
```

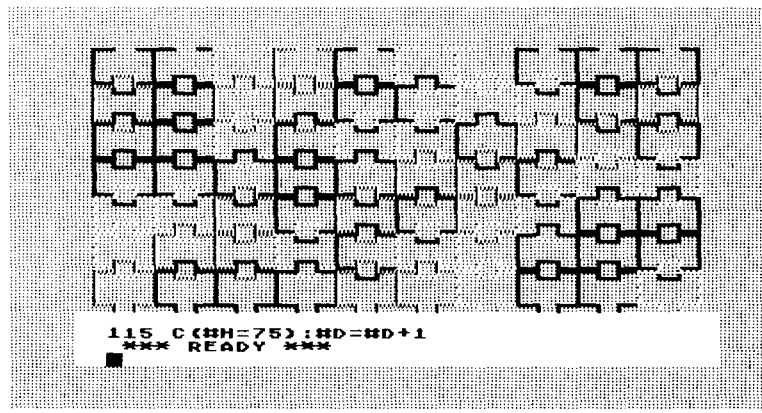
```
3   2 R:          *** TRILL ***
    4 R:           BY TED KAHN
    6 R:(EXPERIMENT WITH DIFFERENT SPEEDS
    & INTERVALS TO GET VARIOUS EFFECTS)
    100 *TRILL1 VNEW
    110 T:
    120 T:ENTER A STARTING NOTE NUMBER (1-
    30):
    130 A:#A
    140 T:NOW ENTER AN INTERVAL (1-12):\
    150 A:#I
    160 C:#B=#A+#I
    170 T:FINALLY, ENTER A SPEED (1-30):\
    180 A:#P
    190 *PLAY
    200 SO:#A
    210 PA:#P
    220 SO:#B
    230 PA:#P
    240 C:#T=#T+1
    250 J(#T<8):*PLAY
    260 J:*TRILL1
    270 E:
```

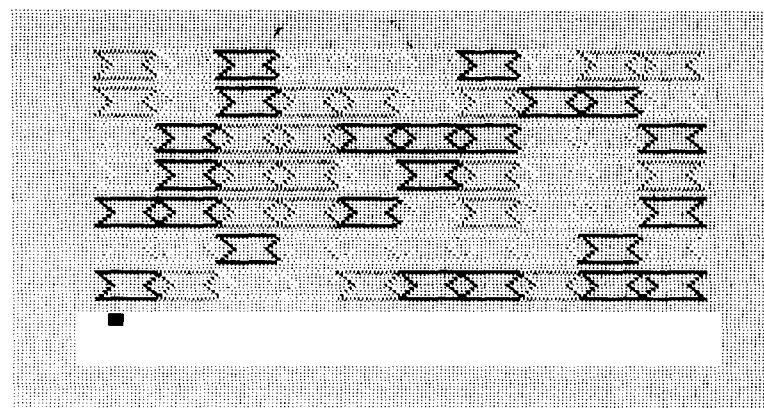
Bach

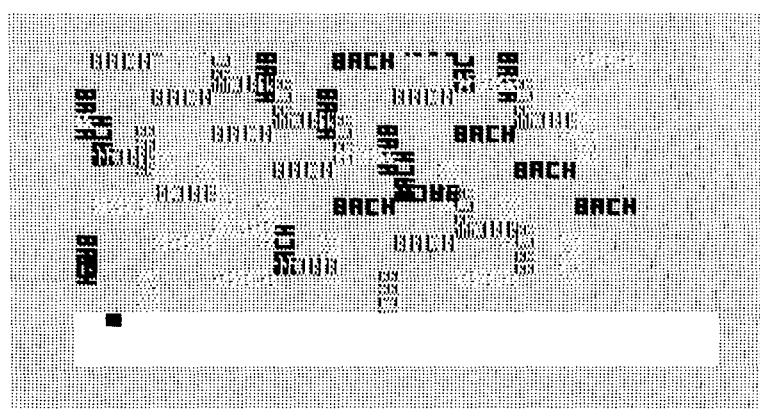
Finally here is a program that we have called Bach, that orchestrates the music from Bach's 57th Cantata (which is reproduced in the sound section) with a variety of tessellations and color changes. The program displays the range of hues and luminescences that enrich the color possibilities of the ATARI Computer. (Control of these hues and luminescences can be done in PILOT using the POKE numbers listed in Appendix II.) The program also has a TRON (trace on) function, that is, a function that tells where it is in the music and what color it is using while it's running.



Bach is a tribute to Richard Hofstadter, whose book *Gödel, Escher, Bach: An Eternal Golden Braid* (Basic Books, 1979) weaves together music, painting, and self-reference to make a wonderful combination of opera buffa and opera seriosa.





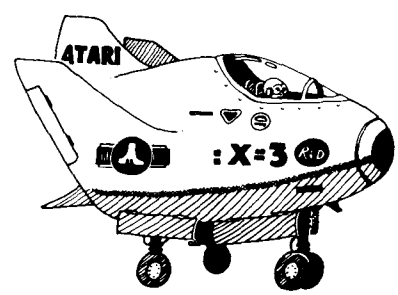


```
100 R:          ***  BACH  ***
110 R:        COPYRIGHT (C) 1982
120 R:  by Dale Disharoon & Herb Kohl
130 R:
140 J:*COVER
150 R:--------------------------------
160 *T
170 C(#G):@B656=1
180 T(#G):MEASURE ##M
190 U:*COLOR
200 C(#A=0):#H=-75
210 C:#A=1
220 C(#B=0):#V=47
230 C:#B=1
240 U:*GRAPHICS
250 C:#V=#V-11
260 E(#V>-30):
270 C:#B=0
280 C:#H=#H+15
290 E(#H<75):
300 C:#A=0
310 C:#D=#D+1
320 GR:CLEAR
330 U:*REGISTERS
340 E:
350 R:--------------------------------
360 *T3
370 U:*REGISTERS
380 PA:#T*3
390 E:
400 R:--------------------------------
410 *GRAPHICS
420 C(#G):@B656=2
430 T(#G):GRAPHICS [
440 GR:GOTO #H,#V
450 A:=#D
460 M:0,1,2
470 JM:*G0,*G1,*G2
480 R:--------------------------------
490 *COLOR
500 C:#C=?\3
510 GR(#C=0):PENRED
520 GR(#C=1):PENYELLOW
530 GR(#C=2):PENBLUE
540 E:
550 R:--------------------------------
560 *REGISTERS
570 C(#G):@B656=2
580 T(#G):REGISTERS
590 C:#Z=708
600 *REGISTERLOOP
610 C:#R=?\16
620 C:#L=?\7+4
630 C:#R=#R*16+#L
640 C:@B#Z=#R
650 C:#Z=#Z+1
660 J(#Z<711):*REGISTERLOOP
670 E:
680 R:--------------------------------
690 *G0 GR:2(TURN 90;DRAW 5;TURN -90;D
RAW 3;TURN 90;DRAW 5;TURN 90;DRAW 3;TU
```

```
RN -90;DRAW 5;TURN 90;DRAW 10)
700 E:
710 R:--------------------------------
720 *G1 GR:2(TURN 90;DRAW 16;TURN 135;
DRAW 6;TURN -90;DRAW 6;TURN 45)
730 GR:GOTO -12-#H,#V;2(TURN 90;DRAW 1
6;TURN 135;DRAW 6;TURN -90;DRAW 6;TURN
 45)
740 E:
750 R:--------------------------------
760 *G2 GR:TURN 90;DRAW 2;TURN 90;DRAW
 4;TURN 90;DRAW 2;TURN 90;DRAW 3;TURN
180;GO 1;TURN -90;GO 1
770 GR:GO 3;DRAW 2;TURN 90;DRAW 2;TURN
 90;GO 2;TURN 90;DRAW 4;TURN 90;DRAW 2
;TURN 90;DRAW 4;TURN -90
780 GR:GO 4;TURN 180;DRAW 2;TURN 90;DR
AW 4;TURN 90;DRAW 2
790 GR:GO 2;TURN 90;DRAW 4;TURN -90;GO
 2;TURN -90;DRAW 4;TURN 180;GO 2;TURN
90;GO 1;TURN 180
800 E:
810 R:--------------------------------
820 *MEASURE1
830 C:#M=1
840 SO:1
850 U:*T
860 SO:1,13
870 U:*T
880 SO:1,15
890 U:*T
900 SO:1,13,17
910 U:*T
920 SO:1,20
930 U:*T
940 SO:1,12,18
950 U:*T
960 SO:10,13,18
970 U:*T
980 SO:10,22
990 U:*T
1000 SO:10,15,20
1010 U:*T
1020 C:#M=2
1030 SO:5,13,20
1040 U:*T
1050 SO:5,25
1060 U:*T
1070 SO:5,15,24
1080 U:*T
1090 SO:10,17,25
1100 U:*T
1110 SO:10,20
1120 U:*T
1130 SO:10,13,17
1140 U:*T
1150 SO:10,5,13
1160 U:*T
1170 SO:10,15
1180 U:*T
1190 SO:10,8,17
```

```
1200 U:*T
1210 C:#M=3
1220 SO:3,10,18
1230 U:*T
1240 SO:3,20
1250 U:*T
1260 SO:3,12,22
1270 U:*T
1280 SO:5,13,20
1290 U:*T
1300 SO:5,18
1310 U:*T
1320 SO:5,8,17
1330 U:*T
1340 SO:6,10,15
1350 U:*T
1360 SO:6,17
1370 U:*T
1380 SO:6,5,13
1390 U:*T
1400 C:#M=4
1410 SO:8,3,12
1420 U:*T
1430 SO:8,13
1440 U:*T
1450 SO:8,15
1460 U:*T
1470 SO:3,8,12
1480 U:*T
1490 SO:8,13
1500 U:*T
1510 SO:12,15
1520 U:*T
1530 SO:8,12,18
1540 U:*T
1550 SO:8,17
1560 U:*T
1570 SO:8,12,15
1580 U:*T
1590 C:#M=5
1600 SO:1,13,17
1610 U:*T
1620 SO:1,13
1630 U:*T
1640 SO:1,12,15
1650 U:*T
1660 SO:10,13,17
1670 U:*T
1680 SO:10,20
1690 U:*T
1700 SO:10,12,18
1710 U:*T
1720 SO:6,10,13,18
1730 U:*T
1740 SO:6,22
1750 U:*T
1760 SO:6,15,20
1770 U:*T
1780 C:#M=6
1790 SO:5,13,17,20
1800 U:*T
```

```
1810 SO:5,25
1820 U:*T
1830 SO:5,15,24
1840 U:*T
1850 SO:10,17,25
1860 U:*T
1870 SO:10,20
1880 U:*T
1890 SO:10,13,17
1900 U:*T
1910 SO:8,10,13
1920 U:*T
1930 SO:8,15
1940 U:*T
1950 SO:8,17
1960 U:*T
1970 C:#M=7
1980 SO:6,10,15
1990 U:*T
2000 SO:6,20
2010 U:*T
2020 SO:6,18
2030 U:*T
2040 SO:7,10,13,17
2050 U:*T
2060 SO:7,15
2070 U:*T
2080 SO:7,13
2090 U:*T
2100 SO:8,3
2110 U:*T
2120 SO:8,13
2130 U:*T
2140 SO:8,6,12
2150 U:*T
2160 C:#M=8
2170 SO:1,5,8,13
2180 U:*T
2190 SO:1,5,8,17
2200 U:*T
2210 SO:1,5,13,20
2220 U:*T
2230 SO:17,25
2240 U:*T
2250 SO:20
2260 U:*T
2270 SO:1,8,13,17
2280 U:*T
2290 SO:12,8,13
2300 U:*T
2310 SO:12,17
2320 U:*T
2330 SO:12,10,19
2340 U:*T
2350 C:#M=9
2360 SO:8,12,15
2370 U:*T3
2380 SO:6,3,8,15
2390 U:*T3
2400 SO:5,8,14,17
2410 U:*T3
```

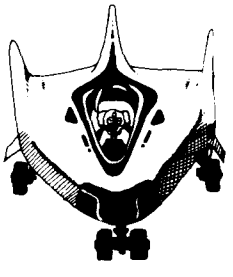


```
2420 C:#M=10
2430 SO:3,10,15,18
2440 U:*T3
2450 U:*T
2460 PA:#T/2
2470 SO:1,10,15,18
2480 U:*T
2490 PA:#T/2
2500 SO:6,11,15,18
2510 U:*T
2520 PA:#T/2
2530 SO:6,10,15,18
2540 U:*T
2550 PA:#T/3
2560 C:#M=11
2570 SO:8,11,15,17
2580 U:*T3
2590 SO:5,11,15
2600 U:*T3
2610 SO:18
2620 PA:#T/2
2630 SO:20
2640 PA:#T/2
2650 SO:10,14,17
2660 U:*T
2670 PA:#T/2
2680 SO:8,10,14,17
2690 U:*T
2700 PA:#T/2
2710 C:#M=12
2720 SO:3,15
2730 U:*T
2740 SO:3,18
2750 U:*T
2760 SO:3,10,17
2770 U:*T
2780 SO:11,6,15
2790 U:*T
2800 SO:11,22
2810 U:*T
2820 SO:11,14,20
2830 U:*T
2840 SO:8,11,15,20
2850 U:*T
2860 SO:8,11,15,23
2870 U:*T
2880 SO:8,11,15,22
2890 U:*T
2900 C:#M=13
2910 SO:6,15,22
2920 U:*T
2930 SO:6,15,27
2940 U:*T
2950 SO:6,15,17,26
2960 U:*T
2970 SO:11,18,27
2980 U:*T
2990 SO:11,18,22
3000 U:*T
3010 SO:11,18
3020 U:*T
```

```
3030 SO:10,6,11,15
3040 U:*T
3050 SO:10,6,11,17
3060 U:*T
3070 SO:10,6,11,18
3080 U:*T
3090 C:#M=14
3100 SO:8,15,23
3110 U:*T
3120 SO:8,15,22
3130 U:*T
3140 SO:8,15,20
3150 U:*T
3160 SO:9,12,18
3170 U:*T
3180 SO:9,12,17
3190 U:*T
3200 SO:9,12,15
3210 U:*T
3220 SO:10,5
3230 U:*T
3240 SO:10,5,15
3250 U:*T
3260 SO:10,5,14
3270 U:*T
3280 C:#M=15
3290 SO:3,18
3300 U:*T
3310 SO:6,18
3320 U:*T
3330 SO:10,18
3340 U:*T
3350 SO:3,15,18
3360 U:*T3
3370 SO:5,8,13,20
3380 U:*T3
3390 C:#M=16
3400 SO:6,13,18,22
3410 U:*T3
3420 SO:3,15,18,22
3430 U:*T3
3440 SO:10,6,16,22
3450 U:*T3
3460 C:#M=17
3470 SO:11,15,20
3480 U:*T
3490 PA:#T/2
3500 SO:11,17,20
3510 U:*T
3520 PA:#T/2
3530 SO:8,11,18
3540 U:*T3
3550 SO:22
3560 PA:#T/2
3570 SO:23
3580 PA:#T/2
3590 SO:1,13,17,20
3600 U:*T
3610 PA:#T/2
3620 SO:1,11,20
3630 U:*T
```

```
3640 PA:#T/2
3650 C:#M=18
3660 SO:6,10,18
3670 U:*T
3680 SO:6,10,13
3690 U:*T
3700 SO:1,6,11
3710 U:*T
3720 SO:10,13,18
3730 U:*T
3740 SO:10,18
3750 U:*T
3760 SO:1,6,10
3770 U:*T
3780 SO:6,10,13
3790 U:*T
3800 SO:6,10,15
3810 U:*T
3820 SO:6,8,10,16
3830 U:*T
3840 C:#M=19
3850 SO:6,11,15
3860 U:*T
3870 SO:6,11,18
3880 U:*T
3890 SO:6,10,11,17
3900 U:*T
3910 SO:8,11,18
3920 U:*T
3930 SO:8,11,15
3940 U:*T
3950 SO:8,11
3960 U:*T
3970 SO:1,5,8
3980 U:*T
3990 SO:1,5,10
4000 U:*T
4010 SO:1,5,11
4020 U:*T
4030 C:#M=20
4040 SO:6,10
4050 U:*T
4060 SO:6
4070 U:*T
4080 SO:1,6,8
4090 U:*T
4100 SO:6,10
4110 U:*T
4120 SO:6,13
4130 U:*T
4140 SO:3,6,8,12
4150 U:*T
4160 SO:5,8,13
4170 U:*T
4180 SO:5,8,17
4190 U:*T
4200 SO:3,8,12,15
4210 U:*T
4220 C:#M=21
4230 SO:1,8,13
4240 U:*T
```

```
4250 SO:1,8,17
4260 U:*T
4270 SO:1,8,20
4280 U:*T
4290 SO:1,8,17,25
4300 U:*T
4310 SO:1,8,17,20
4320 U:*T
4330 SO:1,8,17
4340 U:*T
4350 SO:1,8,13
4360 U:*T
4370 SO:1,8,17
4380 U:*T
4390 SO:1,8,20
4400 U:*T
4410 C:#M=22
4420 SO:1,11,17,23
4430 U:*T
4440 SO:1,11,17,20
4450 U:*T
4460 SO:1,11,17
4470 U:*T
4480 SO:1,8,13
4490 U:*T
4500 SO:1,8,17
4510 U:*T
4520 SO:1,8,20
4530 U:*T
4540 SO:1,13,18,22
4550 U:*T
4560 SO:1,13,18
4570 U:*T
4580 SO:1,13,18,15
4590 U:*T
4600 C:#M=23
4610 SO:1,12
4620 U:*T
4630 SO:1,15
4640 U:*T
4650 SO:1,18
4660 U:*T
4670 SO:1,13,17,20
4680 U:*T
4690 SO:1,13,17
4700 U:*T
4710 SO:1,13
4720 U:*T
4730 SO:1,5,10
4740 U:*T
4750 SO:1,5,13
4760 U:*T
4770 SO:1,5,17
4780 U:*T
4790 C:#M=24
4800 SO:1,10,15,18
4810 U:*T
4820 SO:1,10,15
4830 U:*T
4840 SO:1,10,12
4850 U:*T
```



```
4860 SO:1,8
4870 U:*T
4880 SO:1,12
4890 U:*T
4900 SO:1,15
4910 U:*T
4920 SO:1,8,18
4930 U:*T
4940 SO:1,8,17
4950 U:*T
4960 SO:1,8,15
4970 U:*T
4980 C:#M=25
4990 SO:1,8,13,17
5000 U:*T
5010 SO:1,8,13
5020 U:*T
5030 SO:1,8,12,15
5040 U:*T
5050 SO:10,13,17
5060 U:*T
5070 SO:10,13,20
5080 U:*T
5090 SO:10,12,13,18
5100 U:*T
5110 SO:6,10,13,18
5120 U:*T
5130 SO:6,10,13,22
5140 U:*T
5150 SO:6,10,15,20
5160 U:*T
5170 C:#M=26
5180 SO:5,13,17,20
5190 U:*T
5200 SO:5,13,17,25
5210 U:*T
5220 SO:5,13,15,24
5230 U:*T
5240 SO:10,17,25
5250 U:*T
5260 SO:10,17,20
5270 U:*T
5280 SO:10,13,17
5290 U:*T
5300 SO:8,10,13
5310 U:*T
5320 SO:8,10,15
5330 U:*T
5340 SO:8,10,17
5350 U:*T
5360 C:#M=27
5370 SO:6,10,15
5380 U:*T
5390 SO:6,10,20
5400 U:*T
5410 SO:6,10,18
5420 U:*T
5430 SO:7,10,13,17
5440 U:*T
5450 SO:7,10,13,15
5460 U:*T
```

```
5470 SO:7,10,13
5480 U:*T
5490 SO:8
5500 U:*T
5510 SO:8,13
5520 U:*T
5530 SO:6,8,12
5540 U:*T
5550 C:#M=28
5560 SO:1,5,8,13
5570 U:*T3
5580 U:*T3
5590 U:*T3
5600 R:------------------------------
5610 *COVER
5620 U:*WINDOWMOD
5630 GR:GOTO -40,20;DRAWTO -30,20;DRAW
TO -25,15;DRAWTO -25,10;DRAWTO -28,7;D
RAWTO -25,4;DRAWTO -25,-1
5640 GR:DRAWTO -30,-6;DRAWTO -40,-6;FI
LLTO -40,20
5650 GR:GOTO -20,20;DRAWTO -5,20;DRAWT
O -5,-6;DRAWTO -10,-6;FILLTO -10,4;DRA
WTO -15,4
5660 GR:DRAWTO -15,-6;DRAWTO -20,-6;FI
LLTO -20,20
5670 GR:GOTO 0,20;DRAWTO 15,20;DRAWTO
15,15;DRAWTO 5,15;DRAWTO 5,-1;DRAWTO 1
5,-1;DRAWTO 15,-6;DRAWTO 0,-6
5680 GR:FILLTO 0,20
5690 GR:GOTO 35,20;DRAWTO 35,-6;DRAWTO
 30,-6;FILLTO 30,20;GOTO 20,20;DRAWTO
25,20;DRAWTO 25,10;DRAWTO 25,20
5700 GR:GOTO 30,4;DRAWTO 25,4;DRAWTO 2
5,-6;DRAWTO 20,-6;FILLTO 20,20
5710 GR:PENERASE;GOTO -33,17;TURNTO 13
5;4(DRAW 6;TURN 90)
5720 GR:GOTO -33,6;TURNTO 135;4(DRAW 6
;TURN 90)
5730 GR:GOTO -15,16;TURNTO 90;2(DRAW 5
;TURN 90;DRAW 8;TURN 90)
5740 GR:GOTO -79,47;TURNTO 90
5750 C:#X=158
5760 C:#Y=78
5770 *COVERLOOP
5780 U:*COLOR
5790 U:*REGISTERS
5800 GR:DRAW #X;TURN 90;DRAW #Y;TURN 9
0
5810 C:#X=#X-2
5820 C:#Y=#Y-2
5830 C:#E=#E+1
5840 J(#H*(#E>5)):*END
5850 J(#E<22):*COVERLOOP
5860 C:#E=0
5870 T:
5880 T:TRON?  \
5890 A:$A
5900 M:Y,SURE,OK
5910 CY:#G=1
5920 GR:QUIT
```

```
5930 GR:CLEAR;TURNTO 0
5940 C:#T=20
5950 J:*MEASURE1
5960 R:------------------------------
5970 *WINDOWMOD
5980 GR:CLEAR
5990 C:#W=@560
6000 C:#W=#W+85
6010 C:@B#W=70
6020 C:#W=#W+3
6030 C:@B#W=6
6040 C:#W=#W+1
6050 C:@B#W=6
6060 C:#W=#W+1
6070 C:@B#W=6
6080 E:
6090 R:------------------------------
6100 *END
6110 T:[CLEAR]
6120 C:@B656=0
6130 E:
```

Rambrant

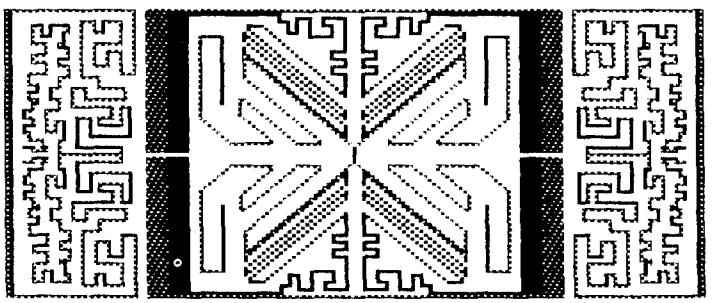
Rambrant is a program that allows you to draw on the screen using a joystick. You can make figures and shapes; can change colors, hues, and luminances; draw with cubes or stars; and do quad drawing (that is, drawing with four paints simultaneously) that produces designs like the illustrations.

Here is the menu for the program.

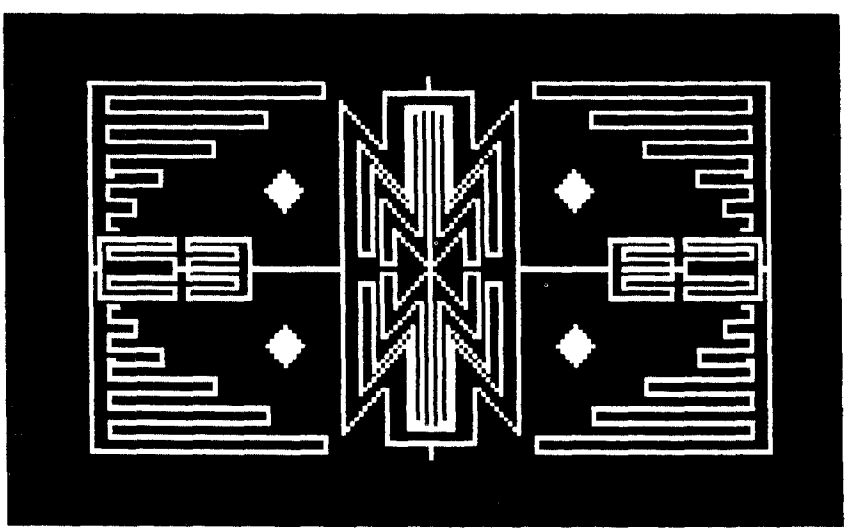
```
      *** RAMBRANDT COMMANDS ***

MOVE JOYSTICK TO DRAW
PRESS TRIGGER TO CHANGE PENS

KEYBOARD OPTIONS:

B: Change Background Color
C: 3-D CUBE (hold key down & draw)
D: Disk Storage (Save & Load Screen)
E: Erase Screen
H: Change HUE of current pen
L: Change LUMINANCE of current pen
Q: Quad Draw (turn on & off)
S: Star
I: Instruction Page


            Press Any Key
```

Notice that each command has a letter that actually names a module. For example, *B is the module that allows you to change the background color of your drawing. If you

look at line 410 of the program, you'll see that you can match to any letter of the alphabet. However, the only modules that we've given you are *B, *C, *D, *E, *H, *L, *Q, *S, and *I. The rest are for you to create. Make a shape or form and you can draw with it. Create a new technique. All you must do is turn it into a module (in effect, a PILOT word) and add it to the menu.



*** RAMBRANT THE COMPUTER ARTIST ***



```
100 R:       *** RAM-BRANDT ***
110 R:
120 R:       COPYRIGHT (C) 1982
130 R:       by  Dale Disharoon
140 R:
150 J:*INITIALIZE
160 *MAINLOOP U(@B764<>255):*KEYPRESS
170 U(%T8=1):*TRIGGER
180 C:#J=%JO
190 C((#J=5)+(#J=1)+(#J=9)):#Y=#Y+1
```

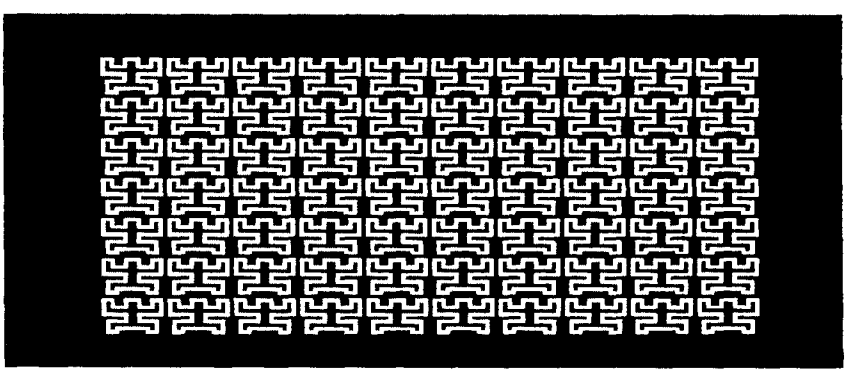
```
200 C((#J=6)+(#J=2)+(#J=10)):#Y=#Y-1
210 C((#J=5)+(#J=4)+(#J=6)):#X=#X-1
220 C((#J=9)+(#J=8)+(#J=10)):#X=#X+1
230 C(#X<-79):#X=#R
240 C(#X>#R):#X=-79
250 C(#Y>47):#Y=#B
260 C(#Y<#B):#Y=47
270 U(#P=0):*FLASH
280 GR:GOTO #X,#Y
290 U(#Q):*QUADRAW
300 J:*MAINLOOP
310 *QUADRAW
320 GR:GOTO 0-#X,#Y;GOTO #X,16-#Y;GOTO
 0-#X,16-#Y
330 E:
340 *FLASH GR:PENYELLOW;GOTO #X,#Y
350 U(#Q):*QUADRAW
360 GR:PENERASE
370 E:
380 *KEYPRESS A:=@B764
390 C:@B764=255
400 M:63,21,18,58,42,56,61,57,13, 1 ,
5 , 0 ,37,35, 8 ,10,47,40,62,45,11,16,
46,22,43,23
410 JM:*A,*B,*C,*D,*E,*F,*G,*H,*I,*J,*
K,*L,*M,*N,*O,*P,*Q,*R,*S,*T,*U,*V,*W,
*X,*Y,*Z
420 E:
430 *TRIGGER C:#P=#P+1
440 C(#P=4):#P=0
450 GR(#P=0):PENERASE
460 GR(#P=1):PENRED
470 GR(#P=2):PENYELLOW
480 GR(#P=3):PENBLUE
490 C:#H=#H+1
500 C(#H=712):#H=708
510 *TRIGDOWN J(%T8):*TRIGDOWN
520 E:
530 *A
540 E:
550 *B [BACKGROUND COLOR
560 C:#T=@B712
570 C:#T=#T+2
580 C:@B712=#T
590 E:
600 *C [THREE-D CUBE
610 GR:5(PENBLUE;TURNTO 90;DRAW 6;PENR
ED;DRAW 1;TURNTO 45;DRAW 4;DRAW -4;TUR
NTO 270;PENUP;GO 7;TURNTO180;GO1)
620 GR:TURNTO 0;GO 6;TURNTO 90;GO 2
630 GR:6(PENYELLOW;TURNTO 45;DRAW 3;DR
AW -3;PENUP;TURNTO 90;GO 1);TURNTO 0
640 E:
650 *D [DISK STORAGE
660 T:[BELL]DISK STORAGE:
670 T:SAVE, LOAD OR EXIT? (S/L/E) \
680 A:$A
690 M:E
700 JY:*EXIT
710 T:ENTER FILENAME: \
720 A:$FILE
730 C:$FILE=D:$FILE
```

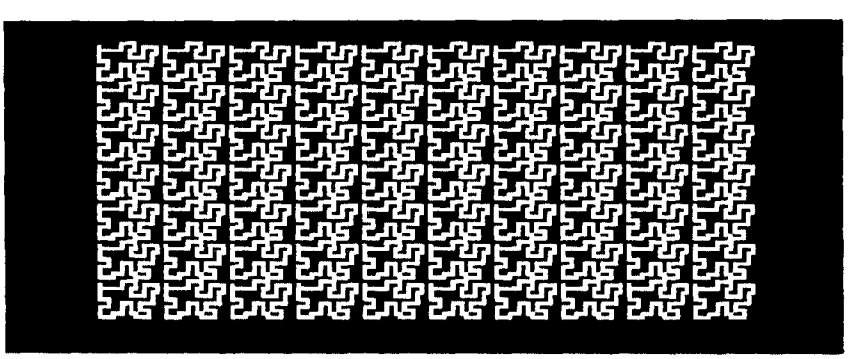
```
740 A:=$A
750 M:S,L
760 JM:*SAVESCREEN,*LOADSCREEN
770 *EXIT T:[CLEAR]
780 E:
790 *SAVESCREEN WRITE:$FILE
800 C:@B203=11
810 CALL:1536
820 CLOSE:$FILE
830 J:*EXIT
840 *LOADSCREEN READ:$FILE
850 C:@B203=7
860 CALL:1536
870 CLOSE:$FILE
880 J:*EXIT
890 *E [ERASE
900 U:*SETUP
910 E:
920 *F
930 E:
940 *G
950 E:
960 *H [HUE CHANGE
970 C:#T=@B#H
980 C:#T=#T+16
990 C:@B#H=#T
1000 E:
1010 E:
1020 *I [INSTRUCTION PAGE
1030 U:*INSTRUCTIONS
1040 U:*SETUP
1050 E:
1060 *INSTRUCTIONS
1070 GR:QUIT
1080 C:@B710=0
1090 C:@B752=1
1100 T:
1110 :      *** RAMBRANDT COMMANDS ***
1120 :
1130 :
1140 :MOVE JOYSTICK TO DRAW
1150 :PRESS TRIGGER TO CHANGE PENS
1160 :
1170 :KEYBOARD OPTIONS:
1180 :
1190 :B: Change Background Color
1200 :C: 3-D CUBE (hold key down & dra
w)
1210 :D: Disk Storage (Save & Load Scr
een)
1220 :E: Erase Screen
1230 :H: Change HUE of current pen
1240 :L: Change LUMINANCE of current p
en
1250 :Q: Quad Draw (turn on & off)
1260 :S: Star
1270 :I: Instruction Page
1280 POS:13,22
1290 T:Press Any Key
1300 C:@B764=255
1310 *HOLD J(@B764=255):*HOLD
1320 C:@B764=255
```

```
1330 E:
1340 *J
1350 E:
1360 *K
1370 E:
1380 *L [LUMINANCE CHANGE]
1390 C:#T=@B#H
1400 C:#J=#T/16
1410 C:#L=#T-(#J*16)
1420 C(#L=14):#L=-2
1430 C:#L=#L+2
1440 C:#T=(#J*16+#L)
1450 C:@B#H=#T
1460 E:
1470 *M
1480 E:
1490 *N
1500 E:
1510 *O
1520 E:
1530 *P
1540 E:
1550 *Q [QUAD DRAW ON/OFF TOGGLE
1560 C:#Q=1-#Q
1570 C(#Q):#R=0
1580 C(#Q):#B=8
1590 C(#Q=0):#R=79
1600 C(#Q=0):#B=-31
1610 E:
1620 *R
1630 E:
1640 *S [STAR
1650 GR:8(DRAW 10;TURN 135)
1660 E:
1670 *T
1680 E:
1690 *U
1700 E:
1710 *V
1720 E:
1730 *W
1740 E:
1750 *X
1760 E:
1770 *Y
1780 E:
1790 *Z
1800 E:
1810 *INITIALIZE U:*COVER
1820 U:*SCREENSTOREPOKES
1830 U:*INSTRUCTIONS
1840 U:*SETUP
1850 J:*MAINLOOP
1860 *SETUP
1870 C:#P=1
1880 C:#R=79
1890 C:#B=-31
1900 C:#Y=8
1910 C:#X=0
1920 C:#H=708
1930 C:#Q=0
1940 GR:CLEAR;PENRED;TURNTO 0
```

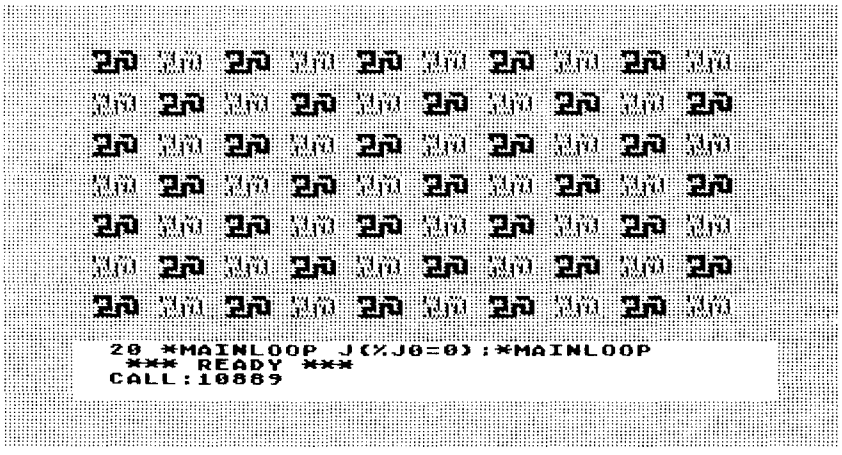
```
1950 E:
1960 *SCREENSTOREPOKES
1970 C:$M=165*88*141*132*3*165*89*141*
133*3*169*128*141*136*3*169*12*141*137
*3*165*203*141*130*3*162*64*76*86*228*
1980 C:#P=1536
1990 *SSLOOP A:=$M
2000 MS:*
2010 EN:
2020 C:$M=$RIGHT
2030 A:#T=$LEFT
2040 C:@B#P=#T
2050 C:#P=#P+1
2060 J:*SSLOOP
2070 *COVER C:@B1373=16
2080 C:@B1374=1
2090 WRITE:S
2100 C:@B710=0
2110 C:@B84=8
2120 WRITE:S,    **************
2130 WRITE:S,    * ram-brandt *
2140 WRITE:S,    **************
2150 T:[TAB]       COPYRIGHT 1982
2160 T:[TAB]       Dale Disharoon
2170 CLOSE:S
2180 E:
```

Waves





This program uses a joystick in port 1. It lets you build wave like patterns that are repeated across the screen by moving the joystick. There is no limit to the variety of repeated forms that can be created with it.



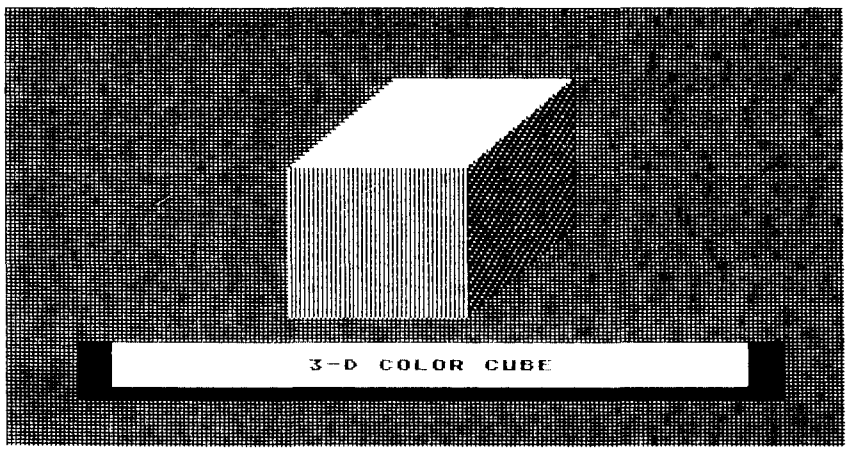

```
100 R:          ***  WAVES  ***
110 R:        COPYRIGHT 1982 by
120 R:         DALE  DISHAROON
130 R:
140 J:*COVER
150 *WAVES C:#H=-75
160 *WAVE1 C:#V=47
170 *WAVE2 C:#C=1-#C
180 GR(#C=0):PENYELLOW
190 GR(#C):PENRED
200 GR:GOTO #H+#X,#V-#Y
210 C:#V=#V-11
220 J(#V<>-30):*WAVE2
230 C:#H=#H+15
240 J(#H<>75):*WAVE1
250 E:
260 *MAINLOOP
270 J(@B764<>255):*KEYBOARD
280 J(%J0=0):*MAINLOOP
290 U:*READSTICK
300 U:*CHECKMARGIN
310 U:*WAVES
320 J:*MAINLOOP
330 *READSTICK C:#J=%J0
340 C:#A=#X
350 C:#B=#Y
360 C((#J=5)+(#J=1)+(#J=9)):#Y=#Y-1
370 C((#J=6)+(#J=2)+(#J=10)):#Y=#Y+1
380 C((#J=5)+(#J=4)+(#J=6)):#X=#X-1
390 C((#J=9)+(#J=8)+(#J=10)):#X=#X+1
400 C((#X>-1)*(#X<15)*(#Y>-1)*(#Y<11))
:$J=$J#J*
410 E:
420 *CHECKMARGIN C(#X<0):#X=0
```

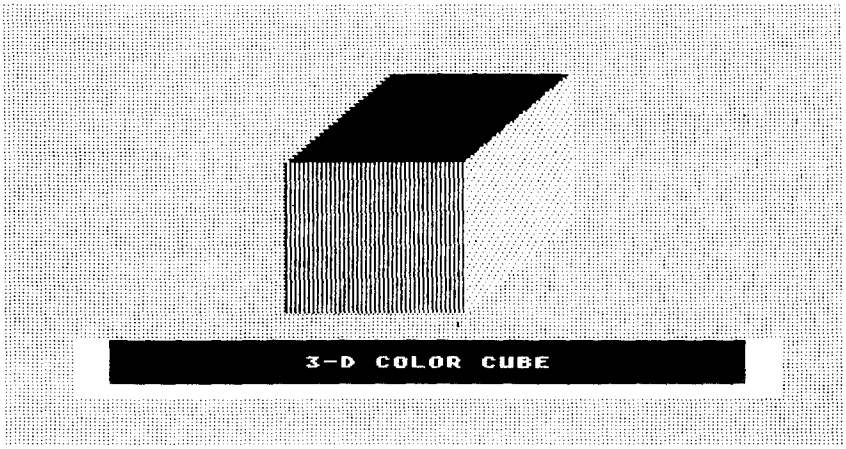
```
430 C(#X>14):#X=14
440 C(#Y<0):#Y=0
450 C(#Y>10):#Y=10
460 E:
470 *KEYBOARD
480 A:=@B764
490 C:@B764=255
500 M: 2 , 18 , 42 , 40 ,
510 JM:*BACKUP,*COLOR,*ERASE,*REDRAW
520 *BACKUP
530 J((#A=#X)*(#B=#Y)):*MAINLOOP
540 GR:PENERASE
550 C:#H=-75
560 *BACK1 C:#V=47
570 *BACK2 GR:GOTO #H+#X,#V-#Y
580 C:#V=#V-11
590 J(#V<>-30):*BACK2
600 C:#H=#H+15
610 J(#H<>75):*BACK1
620 C:#X=#A
630 C:#Y=#B
640 C:$STAR=
650 C:$TLEFT=
660 C:$T=
670 *BLOOP A:=$J
680 MS:*
690 JN:*GLOOP
700 C:$T=$T$TLEFT$STAR
710 C:$TLEFT=$LEFT
720 C:$J=$RIGHT
730 C:$STAR=*
740 J:*BLOOP
750 *GLOOP C:$J=$T
760 J:*MAINLOOP
770 *COLOR
780 U:*RANDOMCOLOR
790 C:@B708=#T
800 U:*RANDOMCOLOR
810 C:@B709=#T
820 J:*MAINLOOP
830 *RANDOMCOLOR
840 C:#H=?\16
850 C:#L=?\7+4
860 C:#T=#H*16+#L
870 E:
880 *ERASE
890 GR:CLEAR
900 C:#X=0
910 C:#Y=0
920 C:#C=0
930 C:$J=
940 J:*TEXTWINDOW
950 *REDRAW
960 GR:CLEAR
970 C:#X=0
980 C:#Y=0
990 C:#C=0
1000 C:$T=$J
1010 *LOOP
1020 A:=$T
1030 MS:*
```

```
1040 JN:*TEXTWINDOW
1050 C:$T=$RIGHT
1060 A:#J=$LEFT
1070 C((#J=5)+(#J=1)+(#J=9)):#Y=#Y-1
1080 C((#J=6)+(#J=2)+(#J=10)):#Y=#Y+1
1090 C((#J=5)+(#J=4)+(#J=6)):#X=#X-1
1100 C((#J=9)+(#J=8)+(#J=10)):#X=#X+1
1110 U:*WAVES
1120 J:*LOOP
1130 *COVER
1140 GR:CLEAR;PENRED
1150 C:@B710=0
1160 C:$J=
1170 T:              <<< WAVES >>>
1180 T:            COPYRIGHT  1982
1190 T:            DALE  DISHAROON\
1200 PA:120
1210 *TEXTWINDOW
1220 T:[CLEAR]
1230 T:
1240 T:
1250 T:
1260 T:B!BACKUP  C!COLOR  E!ERASE  R!R
EDRAW\
1270 J:*MAINLOOP
```

3-D Color Cube

This short program allows you to change the colors on all of the faces of the cube. It is visually interesting to adults and a good teaching tool for youngsters who are just learning about color.



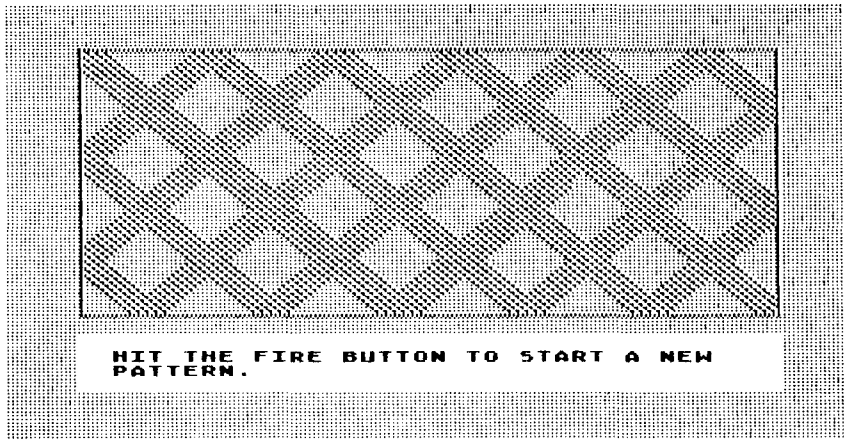



```
100 R:    ***  3-D COLOR CUBE  ***
110 R:
120 R:       By  Dave Thornburg
130 R:  PILOT Version/Dale Disharoon
140 R:
150 C:@B1374=1
160 WRITE:S
170 WRITE:S
180 WRITE:S, THE 3-D COLOR CUBE
190 WRITE:S
200 WRITE:S
210 WRITE:S
220 WRITE:S
230 WRITE:S
240 WRITE:S,  PRESS DIFFERENT
250 WRITE:S
260 WRITE:S,       KEYS TO
270 WRITE:S
280 WRITE:S,   CHANGE COLORS
290 PA:300
300 GR:CLEAR
310 C:@B712=148
320 C:@B708=148
330 C:@B709=148
340 C:#X=-32
350 C:#Y=16
360 C:#A=41
370 *FRONT
380 C:#A=#A-1
390 GR:GOTO #X,#Y-#A;DRAWTO #X+40,#Y-#
A
400 J(#A<>0):*FRONT
410 C:#A=25
420 GR:PENRED
430 *TOP
440 C:#A=#A-1
450 GR:GOTO #X+#A,#Y+#A;DRAWTO #X+#A+4
```

```
0,#Y+#A
460 J(#A<>1):*TOP
470 C:#A=25
480 GR:PENBLUE
490 *SIDE
500 C:#A=#A-1
510 GR:GOTO #X+40+#A,#Y+#A;DRAWTO #X+4
0+#A,#Y-40+#A
520 J(#A<>1):*SIDE
530 T:
540 T:          3-D COLOR CUBE
550 T:
560 *CHANGECOLORS C:#A=711
570 *THREESIDES
580 C:#A=#A-1
590 *HOLD
600 C:#B=@B764
610 J(#B=255):*HOLD
620 C:@B#A=#B
630 SO:#B
640 C:@B764=255
650 PA:6
660 SO:0
670 J(#A<>708):*THREESIDES
680 J:*CHANGECOLORS
```

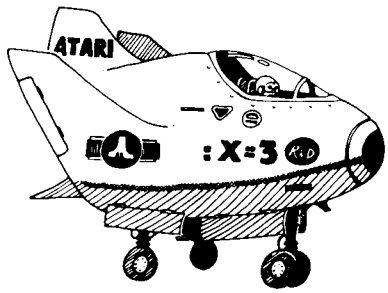
Schizo

With this program you can use a joystick to set borders on the screen and watch more and more complex patterns develop within those borders as a dot moves toward the edge and bounces off at a regular angle, leaving behind a trail of light.



```
2 R:        *** SCHIZO ***
3 R:      COPYRIGHT (C) 1982
4 R:         BY ANDY HALL
100 *RUN U:*INITIALIZE
110 U:*DRAWBOARD
120 *LOOP U:*DRAWLINE
130 U:*CHANGEDIR
140 SO(%T8):0,0
150 J(%T8):*RUN
160 C:#M=?\200
170 GR(#M=1):PENYELLOW
180 GR(#M=2):PENBLUE
190 GR(#M=3):PENERASE
200 J:*LOOP
210 *DRAWBOARD GR:CLEAR;PENYELLOW;GOTO
#A-1,#B+1;DRAWTO#X+1,#B+1;DRAWTO#X+1,#
Y-1
220 GR:DRAWTO#A-1,#Y-1;DRAWTO#A-1,#B+1
;GOTO#P,#Q
230 T:[CLEAR]
240 T:HIT THE FIRE BUTTON TO START A N
EW PATTERN.
250 E:
260 *PLUSMIN C:#N=?\2
270 C(#N=0):#N=-1
280 E:
290 *DRAWLINE C(#C=1):#R=#X-%X
300 C(#C=-1):#R=%X-#A
310 C(#D=1):#S=#B-%Y
320 C(#D=-1):#S=%Y-#Y
330 GR(#R<#S):DRAWTO%X+(#C*#R),%Y+(#D*
#R)
340 GR(#S<=#R):DRAWTO%X+(#C*#S),%Y+(#D
*#S)
350 E:
360 *CHANGEDIR C(#R<=#S):#C=-#C
370 C(#R<=#S):#Z=#Z+1
380 C(#Z=10):#K=3-#K
390 C(#Z=10):#Z=0
400 C(#R>=#S):#D=-#D
410 C:#L=(%X-#A)/6+1
420 C:#T=(%Y-#Y)\16/(@B1363+1)*#K+1
430 SO:#L,#T
440 PA:5
450 E:
460 *INITIALIZE GR:CLEAR
470 C:#J=0
480 C:#K=0
490 T:[CLEAR]USE YOUR JOYSTICK TO POIN
T TO THE UPPER LEFT CORNER OF THE FIEL
D. PRESS THE FIRE BUTTON WHEN READY.
500 U:*JOYSTICK
510 C:#A=#J
520 C:#B=#K
530 GR:PENYELLOW;GOTO#A,#B
540 C:#J=#J+3
550 C:#K=#K-3
560 *TRIGLOOP J(%T8):*TRIGLOOP
```

```
570 *LOWER T:[CLEAR]NOW INDICATE WHERE
 YOU WANT THE LOWER RIGHT CORNER TO BE
.
580 U:*JOYSTICK
590 J((#J<(#A+3))+(#K>(#B-3))):*LOWER
600 C:#X=#J
610 C:#Y=#K
620 U:*PLUSMIN
630 C:#C=#N
640 U:*PLUSMIN
650 C:#D=#N
660 C:#P=#A+(?\(#X-#A))
670 C:#Q=#Y+(?\(#B-#Y))
680 C:#K=1
690 E:
700 *JOYSTICK C:#T=%JO
710 C:#M=#J
720 C:#N=#K
730 GR:PENYELLOW;GOTO#J,#K
740 C(#T>7):#J=#J+1
750 C(#T>7):#T=#T-8
760 C(#T>3):#J=#J-1
770 C(#T>3):#T=#T-4
780 C(#T>1):#K=#K-1
790 C(#T>1):#T=#T-2
800 C(#T=1):#K=#K+1
810 C(#J<-78):#J=-78
820 C(#J>78):#J=78
830 C(#K<-30):#K=-30
840 C(#K>46):#K=46
850 GR:PENERASE;GOTO#M,#N
860 J(%T8=0):*JOYSTICK
870 E:
```



3 Word and Language Games

This chapter contains games that range from simple letter recognition to spelling drills, anagrams, codes, and fairly difficult word challenges. It also contains some silly rhymes and letter-writing programs as well as a version of Eliza, where the program has the ATARI Computer simulate a sympathetic psychiatrist. All the programs can easily be modified, reshaped, expanded, and dressed up. With the concepts in the previous two chapters and the introduction to PEEKS and POKES in Appendix II you should be able to turn many of the programs included here into dozens of other games and word challenges.

Guess the Word

We'll begin with a two-person, bare-bones word-guessing program.

```
10 A:$A
20 T:[CLEAR]
30 *LOOP
40 A:$B
50 M:$A
60 JN:*LOOP
70 T:YOU GOT IT!
```

Line 10 accepts a word ($A), line 20 clears the screen, and line 30 sets up a module, *LOOP. In the module a new word (the guess) is accepted and matches line 50 with $A. If there is no match, the program returns to *LOOP and another guess can be made. If the word is guessed correctly, the computer prints: YOU GOT IT.

Here's a slightly dressed-up version of this bare-bones program.

```
5 T:GIVE ME A WORD FOR YOUR FRIEND TO GUESS.
10 A:$A
20 T:[CLEAR]
30 *LOOP
```

```
35 T:GUESS WHAT WORD I GOT FROM YOUR FRIEND?
40 A:$B
50 M:$A
55 TN:SORRY,GUESS AGAIN.
60 JN:*LOOP
70 T:YOU GOT IT!
```

It's best, when typing in a word for your friend to guess, that he or she is out of the room though, of course, the person could just turn away. This two-person routine is used in a number of other programs in this chapter.

Here's a more complex word-guessing game. The second player can guess the word, a letter, or group of letters; the computer will tell him or her whether those letters are in the target word. On line 190 there is a new PILOT command, MS:, which means match with a string of letters.

MS:$WORD matches the letters typed in $GUESS with the letters in $WORD and lets the player know whether there is a letter match. This MS: string matching is convenient when working with words in PILOT.

```
 YES, RD IS IN MY WORD.
TYPE A LETTER THAT YOU THINK MAY BE
IN MY WORD, OR WHAT YOU THINK THE
WORD IS.
 YOUR GUESS? ORD
 YES, ORD IS IN MY WORD.
TYPE A LETTER THAT YOU THINK MAY BE
IN MY WORD, OR WHAT YOU THINK THE
WORD IS.
 YOUR GUESS? CORD
 NO, THAT'S WRONG.
TYPE A LETTER THAT YOU THINK MAY BE
IN MY WORD, OR WHAT YOU THINK THE
WORD IS.
 YOUR GUESS? WORD
```

```
2 R:     *** GUESS THE WORD #3 ***
4 R:        COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
100 T:[CLEAR]
110 T:TYPE A WORD FOR YOUR FRIEND TO G
UESS.
120 A:$WORD
130 T:[CLEAR]
140 *LOOP T:TYPE A LETTER THAT YOU THI
NK MAY BE IN MY WORD, OR WHAT YOU THIN
K THE WORD IS.
150 T:
160 T: YOUR GUESS? \
170 A:$GUESS
180 A:=$WORD
```

```
190 MS:$GUESS
200 T:
210 TN: NO, THAT'S WRONG.
220 TN:
230 JN: *LOOP
240 A:=$GUESS
250 M:$WORD
260 JY:*WORD
270 T: YES, $GUESS IS IN MY WORD.
280 T:
290 J:*LOOP
300 *WORD T:
310 T: VERY GOOD! YOU GUESSED MY WORD!
320 E:
```

Letter Guessing

In this game the computer gives you a letter, and you must type in a word that begins with that letter. The computer does not check your spelling of the word. It checks only if the first letter that you typed is the one that it requested. It is not a spelling game.

Notice that on line 100 the C: command is used with a string of letters (the alphabet) separated by commas. This is another use of C:. It can handle strings of words and store them. $ALPHA in the case of this program is the alphabet. There are a few other PILOT concepts in this program which may be unfamiliar. They are explained clearly in the PILOT reference manual.

```
   THE LETTER IS: B
 TYPE A WORD THAT BEGINS WITH B.
BAGEL
 VERY GOOD!
 I'LL TYPE A LETTER AND YOU TYPE A
WORD THAT BEGINS WITH MY LETTER.
   THE LETTER IS: B
 TYPE A WORD THAT BEGINS WITH B.
FERMI
NO, FERMI DOESN'T BEGIN WITH B. TRY
ANOTHER WORD.
 TYPE A WORD THAT BEGINS WITH B.
BACH
```

```
2 R:        *** LETTERS ***
4 R:      COPYRIGHT (C) 1982
6 R:         BY ANDY HALL
8 R:
10 R:   FROM A BASIC PROGRAM BY
12 R:   FRONA STAR AND TED KAHN
14 R:
```

```
100 C:$ALPHA=A,B,C,D,E,F,G,H,I,J,K,L,M
,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,
110 T:[CLEAR] LET'S PLAY A LETTER GAME
.
120 T:
130 *LOOP T: I'LL TYPE A LETTER AND YO
U TYPE A WORD THAT BEGINS WITH MY LETT
ER.
140 C:#L=?\26+1
150 A:=$ALPHA
160 *LOOP2 C:#L=#L-1
170 MS:!,
180 A:=$RIGHT
190 J(#L>0):*LOOP2
200 PA:26-#L
210 T:
220 T:    THE LETTER IS:$LEFT
230 T:
240 *AGAIN T: TYPE A WORD THAT BEGINS
WITH$LEFT.
250 T:
260 A:$WORD
270 M:$LEFT
280 JN:*OOPS
290 T:
300 T: VERY GOOD!
310 T:
320 J:*LOOP
330 *OOPS T:
340 T:NO, $WORD DOESN'T BEGIN WITH$LEF
T. TRY ANOTHER WORD.
350 T:
360 J:*AGAIN
370 E:
```

Simple Spelling Drill 1

This program lets you type in five spelling words and then drills you on them. It flashes each word on the screen and then gives you one chance (or several chances if necessary) to spell it correctly. The computer in this case does check for correct spelling. If you want to have the computer accept a longer spelling list, change lines 190, 420, and 440.

```
NOW TRY TO SPELL THE WORD.
BACH
 VERY GOOD!
 TRY THE NEXT ONE:
```

```
2 R:  *** SIMPLE SPELLING DRILL ***
4 R:        COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
100 *ANOTHER T:[CLEAR]
110 T:PLEASE TYPE FIVE OF YOUR OWN SPE
LLING WORDS (ONE PER LINE, EACH FOLLOW
ED BY RETURN).
120 T:
```

```
130 C:#C=1
140 *GETLOOP C:$TEMP=WORD#C
150 T: #C. \
160 A:$WORD
170 C:$$TEMP=$WORD
180 C:#C=#C+1
190 J(#C<6):*GETLOOP
200 T:[CLEAR]
210 T:OKAY, EACH WORD WILL BE FLASHED
ON THE SCREEN BRIEFLY. STUDY IT, AND T
HEN SPELL IT.
220 PA:500
230 C:#C=1
240 *LOOP T:[CLEAR]
250 C:$TEMP=WORD#C
260 C:$WORD=$$TEMP
270 POS:16,10
280 T:$WORD
290 C:#W=0
300 PA:120
310 *AGAIN T:[CLEAR]
320 T:NOW TRY TO SPELL THE WORD.
330 T:
340 A:$GUESS
350 R: LINE 360 SHOULD BE TYPED WITH T
HE UNDERLINE CHARACTER (SHIFT "-") IN
THE STATEMENT.
360 M: $WORD_
370 JN:*OOPS
380 T:
390 T: VERY GOOD!
400 T:
410 *NEXT C:#C=#C+1
420 T(#C<=5): TRY THE NEXT ONE:
430 PA:60
440 J(#C<=5):*LOOP
450 T:
460 T:
470 T: WANT TO DO IT AGAIN? \
480 A:$Y
490 M:Y
500 JY:*ANOTHER
510 E:
520 *OOPS C:#W=#W+1
530 T:
540 T(#W<3):NO, TRY AGAIN.
550 PA(#W<3):60
560 J(#W<3):*AGAIN
570 T:SORRY, THE WORD WAS "$WORD".
580 PA:120
590 J:*NEXT
```

Simple Spelling Drill 2

In this program the computer already has a list of words to drill. We chose five French verbs (there is no reason these two spelling programs can't be used to help people studying foreign languages). By adding $WORD N (for N = 1 to 5) at lines 110 to 150 and adjusting the T(#C =5): at lines 370 and

390, we have a short French drill program. You can substitute any words you care to and also expand the list.

```
NOW TRY TO SPELL THE WORD.
MANGER
 VERY GOOD!
 TRY THE NEXT ONE:


2 R: *** SIMPLE SPELLING DRILL #2 ***
4 R:        COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
8 R:
10 R:     FROM A BASIC PROGRAM BY
12 R:     HERB KOHL AND TED KAHN
14 R:
100 *ANOTHER T:[CLEAR]
110 C:$WORD1=MANGER
120 C:$WORD2=PORTER
130 C:$WORD3=VENIR
140 C:$WORD4=FAIRE
150 C:$WORD5=TOUCHER
160 T:OKAY, EACH WORD WILL BE FLASHED
ON THE SCREEN BRIEFLY. STUDY IT, AND T
HEN SPELL IT.
170 PA:500
180 C:#C=1
190 *LOOP T:[CLEAR]
200 C:$TEMP=WORD#C
210 C:$WORD=$$TEMP
220 POS:16,10
230 T:$WORD
240 C:#W=0
250 PA:120
260 *AGAIN T:[CLEAR]
270 T:NOW TRY TO SPELL THE WORD.
280 T:
290 A:$GUESS
300 R: LINE 310 SHOULD BE TYPED WITH T
HE UNDERLINE CHARACTER (SHIFT "-") IN
THE STATEMENT.
310 M: $WORD_
320 JN:*OOPS
330 T:
340 T: VERY GOOD!
350 T:
360 *NEXT C:#C=#C+1
370 T(#C<=5): TRY THE NEXT ONE:
380 PA:60
390 J(#C<=5):*LOOP
400 T:
410 T:
420 T: WANT TO DO IT AGAIN? \
430 A:$Y
440 M:Y
450 JY:*ANOTHER
460 E:
470 *OOPS C:#W=#W+1
480 T:
490 T(#W<3):NO, TRY AGAIN.
```

```
500 PA(#W<3):60
510 J(#W<3):*AGAIN
520 T:SORRY, THE WORD WAS "$WORD".
530 PA:120
540 J:*NEXT
```

Dropspaces

Dropspaces is a game for two players, one who types in a sentence and the other who must guess the sentence. The computer clears the screen and then drops the spaces between the words in the original sentence, which can sometimes be difficult to decipher.

```
PLAYER #2, WHAT IS THIS SENTENCE WITH
THE PROPER SPACES?
HOWOLDISYOURSISTER


2 R:        *** DROPSPACES ***
4 R:        COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
100 T:
110 T:THIS IS A GAME THAT DROPS THE SP
ACES FROM A PERSON'S SENTENCES.
120 T:
130 *OOPS T:HOW MANY PLAYERS (1-2)? \
140 A:#N
150 J((#N<1)+(#N>2)):*OOPS
160 *AGAIN T(#N=2):ONE PLAYER SHOULD L
EAVE.
170 PA(#N=2):300
180 T:
190 T(#N=2):PLAYER #1, \
200 T:WHAT IS YOUR SENTENCE?
210 A:$SENTENCE
220 U:*DROPSPACE
230 *WRONG T(#N=2):[CLEAR]
240 T:
250 T(#N=2):PLAYER #2, WHAT IS THIS SE
NTENCE WITH THE PROPER SPACES?
260 T(#N=1):HERE IS YOUR SENTENCE WITH
OUT SPACES:
270 T:
280 T:$SENT
290 T:
300 J(#N=1):*AGAIN
310 A:$GUESS
320 T:
330 M:¦$SENTENCE
340 TN:NO, THAT'S NOT RIGHT.
350 PAN:300
360 JN:*WRONG
370 T:VERY GOOD! LET'S DO IT AGAIN.
380 PA:300
390 J:*AGAIN
400 E:
410 *DROPSPACE C:$TSENT=$SENTENCE
```

```
420 C:$SENT=
430 *RLOOP A:=$TSENT
440 MS:[RIGHT]_
450 EN:
460 C:$TSENT=$RIGHT
470 C:$SENT=$SENT$LEFT
480 J:*RLOOP
```

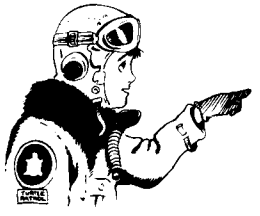
Ropirst

Ropirst is similar to Dropspaces. However, it drops the first letter from every word in a sentence. If you want to combine the two programs for a more complex challenge, you would have to add the *DROPSPACES module at line 410 in the Dropspaces program to Ropirst. Try it for a better idea of how powerful the modules in PILOT are.

```
   ROPIRST IS A GAME FOR TWO PLAYERS.
ONE PLAYER TYPES IN A SENTENCE WHILE
THE OTHER PLAYER IS OUT OF THE ROOM.
THE COMPUTER THEN LOOKS AT THE
SENTENCE AND DROPS THE FIRST LETTER
FROM EVERY WORD.

   AFTER THE SECOND PLAYER COMES BACK
IN, HE LOOKS AT THE NEW VERSION OF
THE SENTENCE AND GETS TO TRY TO
FIGURE OUT WHAT THE ORIGINAL SENTENCE
WAS.

       HIT ANY KEY TO BEGIN.


   AFTER DROPPING THE FIRST LETTER OF
EACH WORD, THE SENTENCE IS:

HERE S  ROBLEM ITH HAT OOSTER F URS

   PLAYER #2: WHAT WAS THE ORIGINAL
SENTENCE?
THERE IS A PROBLEM WITH
```

```
2 R:          *** ROPIRST ***
4 R:         COPYRIGHT (C) 1982
6 R:            BY ANDY HALL
100 U:*INSTRUCT
110 *LOOP T:[CLEAR]
120 T:ONE PLAYER SHOULD LEAVE THE ROOM
.
130 PA:300
140 T:
150 T:
160 T:PLAYER #1: WHAT'S YOUR SENTENCE?
170 A:$SENT
180 C:$SENTENCE=$SENT
190 U:*ROPIRST
200 T:[CLEAR]
210 T:
220 T:    AFTER DROPPING THE FIRST LETT
ER OF EACH WORD, THE SENTENCE IS:
230 *AGAIN T:
240 T:$SENT
250 T:
260 T:    PLAYER #2: WHAT WAS THE ORIGI
NAL SENTENCE?
```

```
270 A:
280 M:!$SENTENCE
290 T:
300 TY:VERY GOOD!
310 PAY:300
320 JY:*LOOP
330 T:    NO, THAT'S NOT THE ORIGINAL S
ENTENCE. TRY AGAIN.
340 PA:300
350 J:*AGAIN
360 E:
370 *ROPIRST C:$TSENT=$SENT
380 C:$SENT=
390 *RLOOP A:=$TSENT
400 MS:[RIGHT]_
410 EN:
420 C:$TSENT=$RIGHT
430 C:$A=$LEFT
440 C:$TLEFT=$LEFT
450 U:*NEXTCHAR
460 C(#E=1):$SENT=$SENT$TLEFT
470 C(#E=0):$SENT=$SENT$A
480 J:*RLOOP
490 *NEXTCHAR
500 C:#E=0
510 A:=$A
520 MS:[2(RIGHT)],
530 CN:#E=1
540 EN:
550 C:$A=$RIGHT
560 MS:$RIGHT
570 A:=$LEFT
580 MS:[RIGHT],
590 A:=$RIGHT
600 MS:[2(RIGHT)],
610 MS:[RIGHT]$RIGHT
620 C:$A1=$LEFT
630 E:
640 *INSTRUCT T:[CLEAR]
650 T:
660 T:     THIS GAME IS CALLED ROPIRST.
670 T:
680 T:    WOULD YOU LIKE INSTRUCTIONS?
\
690 A:
700 M:Y
710 EN:
720 T:[CLEAR]
730 T:
740 T:    ROPIRST IS A GAME FOR TWO PLA
YERS. ONE PLAYER TYPES IN A SENTENCE W
HILE THE OTHER PLAYER \
750 T:IS OUT OF THE ROOM. THE COMPUTER
 THEN LOOKS AT THE SENTENCE AND DROPS
THE FIRST LETTER FROM EVERY WORD.
760 T:
770 T:    AFTER THE SECOND PLAYER COMES
 BACK IN, HE LOOKS AT THE NEW VERSION
OF THE SENTENCE AND GETS TO \
780 T:TRY TO FIGURE OUT WHAT THE ORIGI
NAL SENTENCE WAS.
790 T:
```



```
800 T:       HIT ANY KEY TO BEGIN.
810 C:@B764=255
820 *ILOOP J(@B764=255):*ILOOP
830 C:@B764=255
840 E:
```

A To Z

The challenge of A to Z is to generate a short list of words that contains all the letters in the alphabet. The computer will check your list and let you know letters not in your list as well as when you've achieved the goal. The program could be made more complex by putting in a word counter and a total letter counter and matching one successful list against another for the shortest number of words or letters. Notice that the letters that you've already used are indicated at the top of the screen in inverse print.

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ

   TYPE IN A LIST OF WORDS THAT USE
ALL OF THE LETTERS OF THE ALPHABET.
USE AS FEW WORDS AS POSSIBLE.
TYPE "RETURN" AN EXTRA TIME WHEN
DONE.

COMPUTER
UMBRELLA
DIZZINESS

OOPS, THERE'S NOT A "F" IN THERE.
KEEP TRYING!
```

```
2 R:          *** A TO Z ***
4 R:        COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
100 R:  FIND WORDS THAT USE ALL THE LE
TTERS IN THE ALPHABET.
110 C:$ALPH=ABCDEFGHIJKLMNOPQRSTUVWXYZ
120 *LOOP C:$LETTERS=*
130 C:$USED=$ALPH
140 T:[CLEAR]    [Q][28(R)][E]
150 T:     ! $USED !
160 T:     [Z][28(R)][C]
170 T:
180 C:$WORDLIST=_
190 T:   TYPE IN A LIST OF WORDS THAT
USE ALL OF THE LETTERS OF THE ALPHABET
. USE AS FEW WORDS AS POSSIBLE.
200 T:TYPE "RETURN" AN EXTRA TIME WHEN
 DONE.
210 T:
220 C:#C=0
230 *LLOOP A:$WORD
240 M:[RIGHT],
250 JN:*CHECK
260 C:$WORDLIST=$WORDLIST$WORD
270 C:#C=#C+1
```

```
280 U:*UPDATE
290 J:*LLOOP
300 *CHECK C:$A=$ALPH
310 *CLOOP U:*NEXTCHAR
320 J(#E=1):*FINISHED
330 A:=$WORDLIST
340 M:[RIGHT]$A1
350 JY:*CLOOP
360 T:OOPS, THERE'S NOT A "$A1" IN THE
RE. KEEP TRYING!
370 J:*LLOOP
380 *FINISHED T:
390 T:VERY GOOD! YOU DID IT IN #C WORD
\
400 T(#C<>1):S!
410 T(#C=1):!
420 C:#D=#C-1
430 T(#D>1):TRY TO DO IT IN #D OR FEWE
R WORDS.
440 PA:300
450 J:*LOOP
460 *UPDATE C:$TEMP=$WORD
470 C:#Y=@884
480 *ULOOP C:$A=$TEMP
490 U:*NEXTCHAR
500 J(#E=1):*PRINT
510 C:$TEMP=$A
520 A:=$USED
530 MS:$A1
540 C:#L=@178+@@178+6
550 J(@B#L=32):*ULOOP
560 JN:*ULOOP
570 C:@B#L=@B#L\128+128
580 C:$USED=$LEFT$A1$RIGHT
590 J:*ULOOP
600 *PRINT POS:7,1
610 T:$USED
620 C:@884=#Y
630 E:
640 *NEXTCHAR
650 C:#E=0
660 R:@178 IS POINTER TO START OF STRI
NG MEMORY
670 C:#P=@178+4
680 R:LENGTH OF $A
690 C:#F=@B#P
700 C(#F=0):#E=1
710 E(#F=0):
720 R: ENSURE THAT $A1 IS DEFINED AND
ONE CHARACTER IN LENGTH
730 C:$A1=*
740 R:WHERE TO PUT $A1
750 C:#P=@178+@@178+6
760 R:WHERE TO GET $A
770 C:#T=@178+5
780 C:@B#P=@B#T
790 R:LENGTH OF $A
800 C:#T=#T-1
810 R:MAKE LENGTH-1
820 C:@B#T=@B#T-1
830 C:#F=@B#T
```

```
840 E(#F=0):
850 C:#T=#T+1
860 C:#P=#T+1
870 R:MOVE STRING DOWN IN MEMORY
880 *NEXTLOOP C:@B#T=@B#P
890 C:#F=#F-1
900 E(#F=0):
910 C:#T=#T+1
920 C:#P=#P+1
930 J:*NEXTLOOP
```

Guess the Code

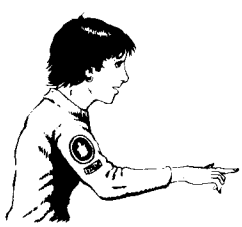
This program lets you create an alphabet substitute code. The computer then tests you on the code. Modifications to this program could consist of translating words and messages into your code or creating a two-person game where one person wrote a message, the computer translated it into code, and the second person had to decode it. It would also be possible for someone to write a message in code and have the computer decode it.

```
 GOOD!

WHAT IS THE CODE FOR R? S
 GOOD!

WHAT IS THE CODE FOR U? V
 GOOD!

WHAT IS THE CODE FOR Z? A
 GOOD!

WHAT IS THE CODE FOR V? W
 GOOD!

WHAT IS THE CODE FOR E? F
 GOOD!

WHAT IS THE CODE FOR F? G
```

```
2 R:     *** GUESS THE CODE ***
4 R:       COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
8 R:
10 R:  FROM THE BASIC PROGRAM BY
12 R:            TED KAHN
14 R:
100 C:$CODE=
110 C:$ALPHA=A,B,C,D,E,F,G,H,I,J,K,L,M
,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,
120 T:[CLEAR]
130 T: TYPE YOUR SECRET CODE FOR EACH
140 T: LETTER OF THE ALPHABET.
150 C:$A=$ALPHA
160 *USED T: USE EACH LETTER ONLY ONCE
.
170 *LOOP A:=$A
```

```
180 MS:¦,
190 TN:[CLEAR]
200 JN:*GAME
210 T:
220 T: WHAT IS YOUR CODE FOR$LEFT? \
230 A:$CHAR
240 A:=$CODE
250 M:$CHAR
260 JY:*USED
270 C:$A=$RIGHT
280 C:$CODE=$CODE$CHAR,
290 J:*LOOP
300 *GAME C:#L=?\26+1
310 C:$C=$CODE
320 C:$A=$ALPHA
330 *GLOOP C:#L=#L-1
340 A:=$A
350 MS:¦,
360 C:$L=$LEFT
370 C:$A=$RIGHT
380 A:=$C
390 MS:¦,
400 C:$C=$RIGHT
410 J(#L>0):*GLOOP
420 T:
430 T:WHAT IS THE CODE FOR$L? \
440 A:$GUESS
450 M:$LEFT
460 TY: GOOD!
470 T:
480 PA:20
490 JY:*GAME
500 T: NO, THE CODE IS$LEFT.
510 T:
520 PA:20
530 J:*GAME
```

Guess the Graphics



This game was designed to help you learn ATARI Computer control graphics. Each graphics character is paired with the letter with which it shares a key. The computer gives the graphics character. You must guess the letter or the same key.

```
    THIS IS A GAME TO HELP YOU LEARN
THE ATARI CONTROL GRAPHICS. THESE
GRAPHICS CHARACTERS ARE OBTAINED BY
HOLDING DOWN THE CTRL KEY WHILE YOU
TYPE A NORMAL KEY.

WHAT LETTER GOES WITH +? S
 GOOD!

WHAT LETTER GOES WITH ♩? C
 GOOD!

WHAT LETTER GOES WITH ¦? B

 NO, THE LETTER IS V.

WHAT LETTER GOES WITH ◣? J
 GOOD!

WHAT LETTER GOES WITH ▗? I
```

```
2 R:    *** GUESS THE GRAPHICS ***
4 R:        COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
8 R:
10 R:     FROM THE BASIC PROGRAM
12 R:           BY TED KAHN
14 R:
100 C:$ALPHA=A;B;C;D;E;F;G;H;I;J;K;L;M
;N;O;P;Q;R;S;T;U;V;W;X;Y;Z;
110 C:$CTR=[A];[B];[C];[D];[E];[F];[G]
;[H];[I];[J];[K];[L];[M];[N];[O];[P];[
Q];[R];[S];[T];[U];[V];[W];[X];[Y];[Z]
120 T:
130 T:    THIS IS A GAME TO HELP YOU LE
ARN THE ATARI CONTROL GRAPHICS. THESE
GRAPHICS CHARACTERS ARE OBTAINED \
140 T:BY HOLDING DOWN THE CTRL KEY WHI
LE YOU TYPE A NORMAL KEY.
150 T:
160 PA:200
170 *GAME C:#L=?\26+1
180 C:$C=$CTR
190 C:$A=$ALPHA
200 *LOOP C:#L=#L-1
210 A:=$C
220 MS:;
230 C:$L=$LEFT
240 C:$C=$RIGHT
250 A:=$A
260 MS:;
270 C:$A=$RIGHT
280 J(#L>0):*LOOP
290 T:WHAT LETTER GOES WITH$L? \
300 A:$GUESS
310 M:$LEFT
320 TY: GOOD!
330 T:
340 JY:*GAME
350 T: NO, THE LETTER IS$LEFT.
360 T:
370 J:*GAME
```

Cryptogram

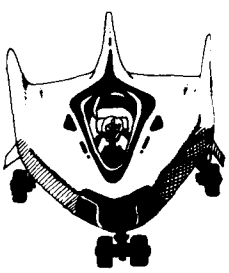
Cryptogram is the most elaborate word game in this book. It is playable on four different levels. The alphabet at the top of the screen registers those coded letters that you have decoded correctly. Though it's quite long, it is worth copying since it presents a series of challenges to sophisticated readers as well as to beginners.

```
A CRYPTOGRAM IS A SENTENCE THAT HAS
HAD SOME OR ALL OF ITS LETTERS
CHANGED BY MEANS OF A CODE. IT IS UP
TO YOU, USING ONLY YOUR WITS AND A
FEW OBSERVATIONS ABOUT THE ENGLISH
LANGUAGE, TO FIGURE OUT WHAT THE
SENTENCE IS.

HERE ARE A FEW HINTS TO HELP YOU
ALONG:

 1. THE MOST COMMONLY USED LETTERS IN
ENGLISH ARE E, T, A, O, I, AND N.

 2. BE ON THE LOOKOUT FOR THINGS LIKE
REPEATING LETTERS, SINGLE LETTERS
(CAN ONLY BE "A" OR "I"), ETC.

        HIT ANY KEY TO CONTINUE.


HERE ARE YOUR OPTIONS:
1. EASY SENTENCE, ONLY VOWELS ENCODED.
2. EASY SENTENCE, EVERYTHING ENCODED.
3. HARD SENTENCE, 1/2 OF ALPHABET
   ENCODED.
4. HARD SENTENCE, EVERYTHING ENCODED.

WHICH OPTION DO YOU WANT (1-4)? ■


  ABCDEFGHIJKLMNOPQRSTUVWXYZ

 HAW HUGH CIN YAE JEMP?

 **** ***** *** **** *****?

ENTER YOUR GUESS FOR A LETTER, OR
"GUESS" IF YOU WANT TO GUESS THE
SENTENCE.
■
```

```
2 R:     **** CRYPTOGRAM ****
4 R:      COPYRIGHT (C) 1982
6 R:         BY ANDY HALL
100 U:*INSTRUCT
110 *AGAIN VNEW
120 C:$ALPH=ABCDEFGHIJKLMNOPQRSTUVWXYZ
130 C:$A1=*
140 C:$ALPHAUSED= $ALPH_
150 U:*VERSION
160 T:PLEASE WAIT A LITTLE BIT . . .
170 C:$CROSTIC=_
180 J(#S=0):*EASYSENT
190 C:#R=?\19
200 A:=#R
210 M:18,17,16,15,14
220 JM:*MESS18,*MESS17,*MESS16,*MESS15
,*MESS14
230 M:13,12,11,10,9,8,7,6,5,4,3,2,1,0
240 JM:*MESS13,*MESS12,*MESS11,*MESS10
,*MESS9,*MESS8,*MESS7,*MESS6,*MESS5,*M
ESS4,*MESS3,*MESS2,*MESS1,*MESS0
250 *EASYSENT C:#R=?\5
260 A:=#R
270 M:4,3,2,1,0
```

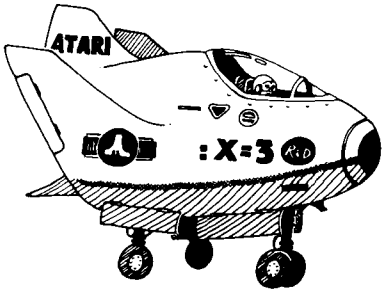
```
280 JM:*EMESS4,*EMESS3,*EMESS2,*EMESS1
,*EMESS0
290 *CONT U:*CODEMAKE
300 U:*ENCODE
310 U:*BLANKS
320 U:*GETGUESS
330 POS:3,5
340 T:ANOTHER GAME? \
350 A:
360 M:Y
370 JY:*AGAIN
380 E:
390 *MESS0 C:$MESS=A TRUE FRIEND IS SO
MEONE WHO KNOWS ALL YOUR FAULTS AND RE
ALLY LIKES YOU ANYWAY.
400 J:*CONT
410 *MESS1 C:$MESS=IF YOU SETTLE FOR A
LMOST ANYTHING, YOU'LL CERTAINLY NEVER
 GET MUCH.
420 J:*CONT
430 *MESS2 C:$MESS=THE ACTIONS OF PEOP
LE ARE THE BEST INTERPRETERS OF THEIR
THOUGHTS.
440 J:*CONT
450 *MESS3 C:$MESS=THERE IS NO TONIC S
O POWERFUL AS EXPECTATION OF SOMETHING
 TOMORROW.
460 J:*CONT
470 *MESS4 C:$MESS=PEDESTRIAN: A TIRED
 FELLOW WHO THOUGHT HE STILL HAD A FEW
 GALLONS LEFT IN THE TANK.
480 J:*CONT
490 *MESS5 C:$MESS=THERE IS NOTHING SO
 SHIFTY AS A HANDFUL OF HANGERS.
500 J:*CONT
510 *MESS6 C:$MESS=ALWAYS TRY TO BE KI
NDER THAN IS REQUIRED.
520 J:*CONT
530 *MESS7 C:$MESS=GRASP A SUBJECT AND
 THE WORDS WILL FOLLOW.
540 J:*CONT
550 *MESS8 C:$MESS=BE KIND TO YOUR CRI
TICS; THEY WILL WONDER WHAT YOU ARE UP
 TO.
560 J:*CONT
570 *MESS9 C:$MESS=THERE IS NO RULE WI
THOUT THE EXCEPTION.
580 J:*CONT
590 *MESS10 C:$MESS=TO ALL WHO LOVE TO
 GET LETTERS: HE WHO WRITES SHALL RECI
EVE.
600 J:*CONT
610 *MESS11 C:$MESS=SOME PEOPLE PLAY A
 FAIR GAME OF GOLF -- THAT IS, IF YOU
ARE WATCHING THEM CLOSE ENOUGH.
620 J:*CONT
630 *MESS12 C:$MESS=NEVER PUT OFF TILL
 TOMORROW ANYTHING YOU CAN DO THE DAY
AFTER TOMORROW.
640 J:*CONT
```

```
650 *MESS13 C:$MESS=NO MAN IS WISE ENO
UGH NOR GOOD ENOUGH TO BE TRUSTED WITH
 UNLIMITED POWER.
660 J:*CONT
670 *MESS14 C:$MESS=SELL NOT YOUR CONS
CIENCE WITH YOUR GOODS.
680 J:*CONT
690 *MESS15 C:$MESS=OPPORTUNITIES APPE
AR TO BE MUCH BIGGER GOING THAN COMING
.
700 J:*CONT
710 *MESS16 C:$MESS=WHEN SUCCESS TURNS
 A MAN'S HEAD, HE FACES FAILURE.
720 J:*CONT
730 *MESS17 C:$MESS=A FRIEND IS TO BE
TAKEN WITH HIS FAULTS.
740 J:*CONT
750 *MESS18 C:$MESS=ONE FALSEHOOD SPOI
LS A THOUSAND TRUTHS.
760 J:*CONT
770 *EMESS0 C:$MESS=SHE WAS THE FASTES
T IN HER CLASS.
780 J:*CONT
790 *EMESS1 C:$MESS=CAN YOU STAND ON Y
OUR HANDS?
800 J:*CONT
810 *EMESS2 C:$MESS=HOW HIGH CAN YOU J
UMP?
820 J:*CONT
830 *EMESS3 C:$MESS=MY DOG HAS FLEAS.
840 J:*CONT
850 *EMESS4 C:$MESS=I USED TO GET DIZZ
Y A LOT.
860 J:*CONT
870 *CODEMAKE C:#R=?\6
880 C:$CODE=#R
890 A:=$EASY
900 M:0
910 CN:$CODE=EASY$EASY
920 C:#A=?\4+1
930 *CODIFY A:=$CODE
940 M:EASY13,EASY,0,1,2,3,4,5,
950 JM:*EASY13,*EASY,*CODE0,*CODE1,*CO
DE2,*CODE3,*CODE4,*CODE5
960 *CODE0 C:#F=155-#F
970 E:
980 *CODE1 C:#F=2*(#F-64)+64
990 C(#F>90):#F=#F-25
1000 E:
1010 *CODE2 C:#F=#F-3
1020 C(#F<65):#F=#F+26
1030 E:
1040 *CODE3 C:#F=#F+3
1050 C(#F>90):#F=#F-26
1060 E:
1070 *CODE4 C:#F=#F-1
1080 C(#F<65):#F=#F+26
1090 E:
1100 *CODE5 C:#F=#F+1
1110 C(#F>90):#F=#F-26
1120 E:
```



```
1130 *EASY A:#C=$EASY
1140 J(#C\2*13+64=77):*EASYL
1150 E(#F>77):
1160 C:#F=#F+#C
1170 C(#F>77):#F=#F-13
1180 E:
1190 *EASYL E(#F<=77):
1200 C:#F=#F-#C
1210 C(#F<77):#F=#F+13
1220 E:
1230 *EASY13
1240 E((#F<>65)*(#F<>69)*(#F<>73)*(#F<
>79)*(#F<>85)):
1250 A:#C=$E
1260 *ELOOP C(#F=65):#G=69
1270 C(#F=69):#G=73
1280 C(#F=73):#G=79
1290 C(#F=79):#G=85
1300 C(#F=85):#G=65
1310 C:#F=#G
1320 C:#C=#C-1
1330 J(#C>0):*ELOOP
1340 E:
1350 *ENCODE C:$TMESS=$MESS
1360 *ELOOP2 C:$A=$TMESS
1370 U:*NEXTCHAR
1380 C:$TMESS=$A
1390 C:#P=@178+@@178+6
1400 C:#F=@B#P
1410 J((#F<65)+(#F>90)):*NCONT
1420 U:*CODIFY
1430 C:@B#P=#F
1440 *NCONT C:$CROSTIC=$CROSTIC$A1
1450 SO(#E=1):0
1460 E(#E=1):
1470 SO:#F
1480 J:*ELOOP2
1490 *BLANKS C:$TMESS=$MESS
1500 C:$BLANK=_
1510 *BLOOP C:$A=$TMESS
1520 U:*NEXTCHAR
1530 C:$TMESS=$A
1540 C:#P=@@178+@178+6
1550 C:#F=@B#P
1560 C(#F=32):$BLANK=$BLANK_
1570 J(#F=32):*BLOOP
1580 A:=$ALPH
1590 M:¦$A1
1600 CY:$BLANK=$BLANK*
1610 CN:$BLANK=$BLANK$A1
1620 SO:#F
1630 J(#E<>1):*BLOOP
1640 SO:0
1650 E:
1660 *CLEARTEXT T:[CLEAR]   [Q][28(R)]
[E]
1670 T:    ¦$ALPHAUSED¦
1680 T:    [Z][28(R)][C]
1690 POS:2,4
1700 T:$CROSTIC
1710 POS:2,8
```

```
1720 T:$BLANK
1730 E:
1740 *CORRECT
1750 C:@B1373=16
1760 C:@B1374=2
1770 WRITE:S,
1780 WRITE:S,
1790 WRITE:S,    YOU GOT IT!!
1800 C:#I=3
1810 C:#C=3
1820 C:#P=11
1830 *SOUND C:#P=#P+1
1840 SO:#P,#I
1850 C:@B708=@B708+16
1860 C:@B712=@B712+16
1870 PA:14
1880 C:#N=12+#I
1890 SO:#N,#I
1900 C:@B708=@B708+16
1910 C:@B712=@B712+16
1920 PA:14
1930 C:#N=17+#I
1940 SO:#N,#I
1950 C:@B708=@B708+16
1960 C:@B712=@B712+16
1970 PA(#C=1):45
1980 PA(#C>1):27
1990 C(#C=1):#I=#I+2
2000 C:#I=#I+3
2010 C:#C=#C-1
2020 J(#C>0):*SOUND
2030 SO:0,0
2040 PA:150
2050 GR:QUIT
2060 C:@B708=40
2070 C:@B712=0
2080 E:
2090 *NEXTCHAR
2100 C:#E=0
2110 C:#P=@178+4
2120 C:#F=@B#P
2130 C(#F=0):#E=1
2140 E(#F=0):
2150 C:#E=0
2160 C:$A1=*
2170 C:#P=@178+@@178+6
2180 C:#T=@178+5
2190 C:#F=@B#T
2200 C:@B#P=#F
2210 C:#T=#T-1
2220 C:#F=@B#T-1
2230 C:@B#T=#F
2240 C(#F=0):#E=1
2250 E(#E=1):
2260 C:#T=#T+2
2270 C:#C=5
2280 *NLOOP C:#T=#T-1
2290 C:#N=#T-1
2300 C:@B#T=@B#N
2310 C:#C=#C-1
2320 J(#C>0):*NLOOP
```



```
2330 C:@#T=@#T-1
2340 C:@178=@178+1
2350 E:
2360 *GETGUESS
2370 U:*CLEARTEXT
2380 POS:2,13
2390 T:ENTER YOUR GUESS FOR A LETTER,
OR "GUESS" IF YOU WANT TO GUESS THE SE
NTENCE.
2400 T:
2410 A:$GUESS
2420 M:GUESS
2430 JN:*LETTER
2440 T:
2450 T:WHAT DO YOU THINK THE SENTENCE
IS?
2460 T:
2470 A:$GUESS
2480 M:$MESS
2490 JY:*CORRECT
2500 T:
2510 T:  NO, THAT'S NOT IT.
2520 PA:190
2530 J:*GETGUESS
2540 *LETTER MS:=
2550 JN:*GETGUESS
2560 C:$TCROS=$CROSTIC
2570 C:$TBLANK=$BLANK
2580 C:$LTEMP=$LEFT
2590 C:$A=$RIGHT
2600 U:*NEXTCHAR
2610 C:$RTEMP=$A1
2620 C:$A=$LTEMP
2630 U:*NEXTCHAR
2640 C:$LTEMP=$A
2650 U:*GOODLET
2660 T(#E=1):[DOWN] NO, THAT'S NOT RIG
HT.
2670 PA(#E):120
2680 J(#E):*GETGUESS
2690 C:$BLANK=
2700 *LLOOP A:=$TCROS
2710 MS:$LTEMP
2720 CN:$BLANK=$BLANK$TBLANK
2730 SON:0
2740 JN:*GETGUESS
2750 C:$TCROS=$RIGHT
2760 C:$STRING=$LEFT
2770 U:*LENGTH
2780 C:#N=#L
2790 C:$CHAR=[RIGHT]
2800 U:*STRINGMAKE
2810 C:$A=$TBLANK
2820 C:#P=@178+5
2830 C:#G=@B#P
2840 A:=$TBLANK
2850 MS:[RIGHT]$STRING*
2860 C:$TBLANK=$RIGHT
2870 MS:|$MATCH$RIGHT
2880 C:$A=$LEFT
2890 U(#G<>32):*NEXTCHAR
```

```
2900 C:$ATEMP=$A
2910 C:$A=$RTEMP
2920 C:#P=@178+5
2930 C:@B#P=@B#P+128
2940 C:$BLANK=$BLANK$ATEMP$A
2950 SO:23
2960 PA:4
2970 SO:0
2980 J:*LLOOP
2990 *INSTRUCT T:[CLEAR]
3000 POS:2,4
3010 T:THIS IS A GAME OF CRYPTOGRAMS.
3020 T:
3030 T:WOULD YOU LIKE INSTRUCTIONS? \
3040 A:
3050 M:Y
3060 EN:
3070 T:[CLEAR]
3080 T:
3090 T:A CRYPTOGRAM IS A SENTENCE THAT
 HAS HAD SOME OR ALL OF ITS LETTERS CH
ANGED BY MEANS OF A CODE. \
3100 T:IT IS UP TO YOU, USING ONLY YOU
R WITS AND A FEW OBSERVATIONS ABOUT TH
E ENGLISH LANGUAGE, TO \
3110 T:FIGURE OUT WHAT THE SENTENCE IS
.
3120 T:
3130 T:HERE ARE A FEW HINTS TO HELP YO
U ALONG:
3140 T:
3150 T: 1. THE MOST COMMONLY USED LETT
ERS IN ENGLISH ARE E, T, A, O, I, AND
N.
3160 T:
3170 T: 2. BE ON THE LOOKOUT FOR THING
S LIKE REPEATING LETTERS, SINGLE LETTE
RS (CAN ONLY BE "A" OR "I"), ETC.
3180 T:
3190 T:       HIT ANY KEY TO CONTINUE.
3200 U:*WAITKEY
3210 T:[CLEAR]
3220 T: 3. EACH CODE HAS A LOGICAL RUL
E TO IT -- IT'S NOT RANDOM.
3230 T:
3240 T: 4. YOU MAY FIND YOURSELF WITH
THE SAME CODE FOR ANY TWO GAMES.
3250 T:
3260 T:WHEN THE GAME GETS GOING, YOU H
AVE TWO OPTIONS: EITHER TO GUESS A LET
TER OR GUESS THE SENTENCE. \
3270 T:IF YOU THINK, FOR EXAMPLE, THAT
 IN THE CODIFIED VERSION "S" STANDS FO
R "T", YOU WOULD TYPE, "S=T". \
3280 T:IF YOU WANT TO GUESS THE SENTEN
CE, TYPE "GUESS" AND THE COMPUTER WILL
 TAKE YOU ON FROM THERE.
3290 T:
3300 T:THERE ARE FOUR DIFFERENT VERSIO
NS OF THE GAME. OPTIONS 1 AND 2 ARE RE
COMMENDED FOR THOSE \
```

```
3310 T:JUST STARTING TO LEARN HOW TO S
OLVE CRYPTOGRAMS.
3320 T:
3330 T:        HIT ANY KEY TO BEGIN.
3340 U:*WAITKEY
3350 T:[CLEAR]
3360 E:
3370 *LENGTH C:#L=0
3380 C:$CRIGHT=[2(RIGHT)]
3390 A:=$STRING
3400 *LENLOOP M:$CRIGHT,
3410 EN:
3420 C:#L=#L+1
3430 C:$CRIGHT=$CRIGHT[RIGHT]
3440 J:*LENLOOP
3450 *STRINGMAKE C:$STRING=
3460 E(#N<1):
3470 *STRINGLOOP C:$STRING=$STRING$CHA
R
3480 C:#N=#N-1
3490 J(#N>0):*STRINGLOOP
3500 E:
3510 *GOODLET C:#E=0
3520 C:$A=$RTEMP
3530 C:#P=@178+5
3540 C:#F=@B#P
3550 U:*CODIFY
3560 C:$A=$LTEMP
3570 C:#P=@178+5
3580 C(#F<>@B#P):#E=1
3590 E(#E=1):
3600 A:=$ALPHAUSED
3610 MS:$RTEMP
3620 TN:[DOWN] THAT'S ALREADY BEEN USE
D.
3630 CN:#E=2
3640 EN:
3650 C:$A=$MATCH
3660 C:#P=@178+5
3670 C:@B#P=@B#P+128
3680 C:$ALPHAUSED=$LEFT$A$RIGHT
3690 E:
3700 *VERSION T:[CLEAR]
3710 POS:2,3
3720 T:HERE ARE YOUR OPTIONS:
3730 T:
3740 T:1. EASY SENTENCE, ONLY VOWELS E
NCODED.\
3750 T:
3760 T:2. EASY SENTENCE, EVERYTHING EN
CODED.
3770 T:
3780 T:3. HARD SENTENCE, 1/2 OF ALPHAB
ET        ENCODED.
3790 T:
3800 T:4. HARD SENTENCE, EVERYTHING EN
CODED.
3810 T:
3820 *VLOOP T:
3830 T:WHICH OPTION DO YOU WANT (1-4)?
 \
```

```
3840 A:#0
3850 J((#0<1)+(#0>4)):*VLOOP
3860 C:#S=#0-1/2
3870 C(#0\2=0):$EASY=0
3880 C:#A=?\11+1
3890 C(#0=3):$EASY=#A
3900 E(#0<>1):
3910 C:$EASY=13
3920 C:#A=?\4+1
3930 C:$E=#A
3940 E:
3950 *WAITKEY C:@B764=255
3960 *WAITLOOP J(@B764=255):*WAITLOOP
3970 C:@B764=255
3980 E:
```

Anagrams

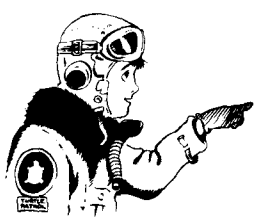
This PILOT version of anagrams has three levels of difficulty. You select a level, decide whether you want hints, and begin the game by using the [OPTION], [SELECT], and [START] keys on the right of the keyboard of the ATARI Computer.

Anagrams are word mixings. An anagram changes around the letters in a word; you must guess the original word. Here is a simple computer-generated anagram from level 1 of our game, along with a correct guess.

```
ABD

BAD
```

Here's a challenge from level 3, the most difficult level. See if you can figure out the word.

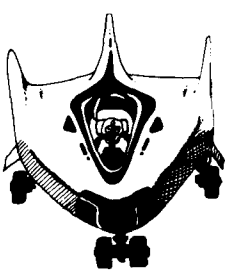
```
OAIECRTN

```



```
2 R:          *** ANAGRAMS ***
4 R:         COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
30 U:*WAITKEY
100 *RERUN VNEW $
110 U:*INITIALIZE
120 GR:QUIT
130 C:@B712=120
140 CLOSE:S
150 U:*DISPLIST
```

```
160 *AGAIN U:*GETWORD
170 C:@B708=252
180 C:@B709=126
190 U:*MIXUP
200 *MAINLOOP U:*DISPLAY
210 C:#Q=0
220 U:*GETGUESS
230 J(#Q=1):*RERUN
240 U:*EVALUATE
250 J(#M=0):*MAINLOOP
260 U:*GOOD
270 J:*AGAIN
280 E:
290 *DISPLIST C:#L=1536
300 C:@B#L=112
310 C:#L=#L+1
320 C:@B#L=112
330 C:#L=#L+1
340 C:@B#L=112
350 C:#L=#L+1
360 C:@B#L=71
370 C:#L=#L+1
380 C:@#L=@88
390 C:#L=#L+2
400 C:#C=8
410 *DISPLOOP C:@B#L=7
420 C(#C=5):@B#L=6
430 C:#L=#L+1
440 C:#C=#C-1
450 J(#C>0):*DISPLOOP
460 C:#C=6
470 *DISPLOOP2 C:@B#L=6
480 C:#L=#L+1
490 C:#C=#C-1
500 J(#C>0):*DISPLOOP2
510 C:@B#L=65
520 C:#L=#L+1
530 C:@#L=1536
540 C:@560=1536
550 C:$STRING=$WORD
560 U:*LENGTH
570 C:#X=(20-#L)/2
580 E:
590 *INITIALIZE C:@B1373=8
600 C:@B1374=1
610 C(#D=0):#D=1
620 C(#Y=0):#Y=9
630 C(#H=0):$HINT=N
640 C(#H=1):$HINT=Y
650 WRITE:S
660 C:@B709=202
670 C:@B710=148
680 C:@B84=3
690 WRITE:S,       anagrams
700 WRITE:S
710 C:@B84=9
720 WRITE:S,    DIFFICULTY    #D
730 C:@B84=12
740 WRITE:S,    HINTS         $HINT
750 C:@B84=15
760 WRITE:S,    INSTRUCTIONS
```

```
770 POS:1,#Y
780 WRITE:S,[J]
790 *ILOOP C:#C=15
800 *ILOOP1 C:#C=#C-1
810 J((#C>0)*(@B53279=#K)):*ILOOP1
820 C:#K=@B53279
830 J(#K=7):*ILOOP
840 J(#K=3):*OPTION
850 J(#K=5):*SELECT
860 J(#K=6):*START
870 J:*ILOOP
880 *OPTION
890 J(#Y=12):*HINTS
900 J(#Y=15):*INSTRUCT
910 POS:16,9
920 C:#D=#D+1
930 C(#D>3):#D=1
940 C:#N=#D*5+4
950 WRITE:S,#D
960 SO:#N
970 C:#N=#N+2
980 PA:3
990 C:#N=#N+2
1000 SO:#N
1010 PA:3
1020 SO:0,0
1030 J:*ILOOP
1040 *HINTS C:#H=1-#H
1050 C:#N=#H*10+10
1060 POS:16,12
1070 WRITE(#H=0):S,N
1080 WRITE(#H=1):S,Y
1090 SO:#N
1100 PA:8
1110 SO:0
1120 J:*ILOOP
1130 *SELECT POS:1,#Y
1140 WRITE:S,_
1150 C:#Y=#Y+3
1160 C(#Y>15):#Y=9
1170 C:#N=#Y
1180 SO:#N
1190 PA:3
1200 C:#N=#N+5
1210 SO:#N
1220 PA:3
1230 C:#N=#N+5
1240 SO:#N
1250 PA:3
1260 SO:0
1270 POS:1,#Y
1280 WRITE:S,[J]
1290 J:*ILOOP
1300 *START SO:27
1310 PA:10
1320 SO:0
1330 PA:10
1340 SO:27
1350 PA:10
1360 SO:0
1370 E:
```



```
1380 *INSTRUCT U:*INSTRUCTIONS
1390 J:*INITIALIZE
1400 *GETWORD J(#D>1):*DIFF2
1410 C:$WORDS1=THE*CAT/ACT*RAT/TAR*ATE
/EAT/TEA*DOG/GOD*BAD/DAB*
1420 C:$WORDS2=BUM*MUD*NUT*PAT/TAP*LIT
*PIT/TIP*PEN*PAN/NAP*RAN*
1430 C:#W=15
1440 *TAKEWORD C:#R=?\#W+1
1450 C:#C=1
1460 *GETLOOP C:$TEMP1=WORDS#C
1470 C:$TEMP=$$TEMP1
1480 C:#C=#C+1
1490 *GETLOOP2 A:=$TEMP
1500 MS:*
1510 JN:*GETLOOP
1520 C:#R=#R-1
1530 C:$TEMP=$RIGHT
1540 J(#R>0):*GETLOOP2
1550 A:=$LEFT
1560 MS:/
1570 C:$WORD=$LEFT
1580 CY:$OTHERS=$RIGHT
1590 CN:$OTHERS=
1600 C:$A=$WORD
1610 U:*NEXTCHAR
1620 C:$WORD=$A
1630 E:
1640 *DIFF2 J(#D=3):*DIFF3
1650 C:$WORDS1=THROW/WORTH*STRAY/TRAYS
*GRAPH*MUSIC*START/TARTS*PHOTO*
1660 C:$WORDS2=SWITCH*VOLUME*FRESH*CHA
IR*LIGHT*GLOVE*SPRAY/PRAYS*
1670 C:#W=13
1680 J:*TAKEWORD
1690 *DIFF3 C:$WORDS1=VARIABLE*MACHINE
*CASSETTE*DIFFICULT*PUZZLE*NATIONAL*
1700 C:$WORDS2=ANAGRAM*CREATION/REACTI
ON*PITCHER*HARDWARE*TELEVISION*
1710 C:#W=11
1720 J:*TAKEWORD
1730 *MIXUP C:$MIXED=
1740 C:$STRING=$WORD
1750 U:*LENGTH
1760 C:#N=#L
1770 C:$CHAR=_
1780 U:*STRINGMAKE
1790 C:$CORRECT=$STRING
1800 C:$WTEMP=$WORD
1810 *MLOOP J(#L=0):*MIXEND
1820 C:#N=?\#L+1
1830 C:$CHAR=[RIGHT]
1840 U:*STRINGMAKE
1850 A:=$WTEMP
1860 MS:$STRING,
1870 C:$STRING=$RIGHT
1880 MS:$RIGHT
1890 C:$A=$STRING
1900 U:*NEXTCHAR
1910 C:$MIXED=$MIXED$A1
1920 C:$WTEMP=$LEFT$A
```

```
1930 C:#L=#L-1
1940 J:*MLOOP
1950 *MIXEND
1960 A:=$WORD/$OTHERS
1970 M:$MIXED
1980 C:$A=$MIXED
1990 U:*LOWER
2000 C:$LMIXED=$A
2010 EN:
2020 J:*MIXUP
2030 *LENGTH C:#L=0
2040 *LENLOOP A:=$STRING
2050 MS:[2(RIGHT)],
2060 C:$STRING=$RIGHT
2070 EN:
2080 C:#L=#L+1
2090 J:*LENLOOP
2100 *STRINGMAKE C:$STRING=
2110 *MAKELOOP C:#N=#N-1
2120 C:$STRING=$STRING$CHAR
2130 J(#N>0):*MAKELOOP
2140 E:
2150 *LOWER C:#P=@178+4
2160 C:#C=@B#P
2170 *LOWLOOP C:#P=#P+1
2180 C:@B#P=@B#P+32
2190 C(@B#P=64):@B#P=32
2200 C:#C=#C-1
2210 J(#C>0):*LOWLOOP
2220 E:
2230 *REVERSE C:#P=@178+4
2240 C:#C=@B#P
2250 *REVLOOP C:#P=#P+1
2260 C:@B#P=@B#P+128
2270 C(@B#P=160):@B#P=32
2280 C:#C=#C-1
2290 J(#C>0):*REVLOOP
2300 E:
2310 *DISPLAY C:$STRING=$WORD
2320 U:*LENGTH
2330 C:#X=10-(#L/2)-(#L\2)
2340 T:[CLEAR]
2350 POS:#X,2
2360 T:$LMIXED
2370 POS:#X,3
2380 T:$CORRECT
2390 E:
2400 *GETGUESS SO:27
2410 PA:15
2420 SO:0
2430 C:$STRING=$WORD
2440 U:*LENGTH
2450 POS:#X,3
2460 A:$GUESS
2470 M:QUIT
2480 JY:*QUIT
2490 C:$STRING=$GUESS
2500 C:#N=#L
2510 U:*LENGTH
2520 J(#N=#L):*NORMAL
2530 C:#L=#N
```

```
2540 SO:6
2550 PA:50
2560 SO:0
2570 POS:#X,3
2580 T:
2590 J:*GETGUESS
2600 *NORMAL C:#L=#N
2610 E:
2620 *EVALUATE C:#M=0
2630 A:=$WORD/$OTHERS
2640 M:$GUESS
2650 CY:#M=1
2660 E((#M=1)+(#H=0)):
2670 C:$CORRECT=
2680 C:$TEMP=$WORD
2690 C:$TEMP2=$GUESS
2700 *ELOOP C:$A=$TEMP
2710 U:*NEXTCHAR
2720 C:$LET=$A1
2730 C:$TEMP=$A
2740 C:$A=$TEMP2
2750 U:*NEXTCHAR
2760 E(#E=1):
2770 C:$TEMP2=$A
2780 A:=$A1
2790 M:$LET
2800 CY:$CORRECT=$CORRECT$LET
2810 CN:$CORRECT=$CORRECT_
2820 J:*ELOOP
2830 *QUIT POS:2,5
2840 T:THE WORD WAS          "$WORD".
2850 PA:200
2860 C:#Q=1
2870 E:
2880 *GOOD C:#C=4
2890 *GOODLOOP0 C:#N=24
2900 *GOODLOOP SO:#N
2910 C(#N\2=0):@B708=372-@B708
2920 PA:1
2930 C:#N=#N+1
2940 J(#N<32):*GOODLOOP
2950 C:#C=#C-1
2960 SO:0
2970 PA:3
2980 J(#C>0):*GOODLOOP0
2990 E:
3000 *NEXTCHAR
3010 C:#E=0
3020 C:#P=@178+4
3030 C:#F=@B#P
3040 C(#F=0):#E=1
3050 E(#F=0):
3060 C:$A1=*
3070 C:#P=@178+@@178+6
3080 C:#T=@178+5
3090 C:@B#P=@B#T
3100 C:#T=#T-1
3110 C:@B#T=@B#T-1
3120 C:#F=@B#T
3130 E(#F=0):
3140 C:#T=#T+1
```

```
3150 C:#P=#T+1
3160 *NEXTLOOP C:@B#T=@B#P
3170 C:#F=#F-1
3180 E(#F=0):
3190 C:#T=#T+1
3200 C:#P=#P+1
3210 J:*NEXTLOOP
3220 *INSTRUCTIONS CLOSE:S
3230 T:[CLEAR]
3240 C:@B755=1
3250 T:
3260 T:    "ANAGRAMS" IS A SIMPLE GAME
TO PLAY. I PICK A WORD, SCRAMBLE ITS L
ETTERS, AND YOU GET TO \
3270 T:TRY TO FIGURE OUT WHAT MY WORD
WAS.
3280 T:
3290 T:    IF YOU LIKE, I CAN GIVE YOU
A HINT AFTER EACH TRY. I WILL LOOK AT
YOUR GUESS AND TELL YOU \
3300 T:WHICH OF THE LETTERS WERE CORRE
CT.
3310 T:
3320 T:       HIT ANY KEY TO CONTINUE.
3330 U:*WAITKEY
3340 T:[CLEAR]
3350 T:
3360 T:    SOME OF MY WORDS MAY HAVE MO
RE THAN ONE ANSWER. FOR EXAMPLE, "TCA"
 COULD BE EITHER "CAT" OR "ACT". \
3370 T:IF I GIVE YOU HINTS, THEY WILL
BE BASED ON ONLY ONE OF THESE WORDS, B
UT EITHER OF THEM WILL BE A \
3380 T:CORRECT ANSWER.
3390 T:
3400 T:    IF YOU GIVE UP OR WOULD LIKE
 TO RETURN TO THE MENU, TYPE "QUIT".
3410 T:
3420 T:              HIT ANY KEY.
3430 U:*WAITKEY
3440 C:@B755=2
3450 E:
3460 *WAITKEY
3470 C:@B764=255
3480 *WLOOP J(@B764=255):*WLOOP
3490 C:@B764=255
3500 E:
```

The Three Little Pigs

This is a "fractured fairy tale." The computer asks questions such as your name, the name of your best friend, the title of a job that always bugs you, etc. Then the computer substitutes your input into the nursery rhyme to create a slightly fractured, somewhat funny personalized version of "The Three Little Pigs." This is a good program to modify. Change the

questions (see, for example, lines 210, 240, 280, 320, and 360) and change the nursery rhyme to one of your favorites. This program presents a form that you can use in many different ways.

```
        * THE THREE LITTLE PIGS *

ONCE THERE WERE THREE LITTLE PIGS
NAMED DALE, HERB AND TED.

A HUNGRY EDITOR NAMED NIKKI
WANTED TO EAT THEM FOR SUPPER.

TERRIFIED, THEY HID IN THEIR CABIN.
FURIOUS, NIKKI SHOUTED:
"I'LL HUFF, AND I'LL PUFF AND I'LL
MELT YOUR CABIN DOWN!"

NO FOOLS, THE THREE PIGS SNEAKED OUT
THE BACK DOOR AND ESCAPED IN THEIR
MERCEDES.

                * THE END *
```

```
100 R: ***  SLIGHTLY FRACTURED  ***
110 R:          FAIRY TALES
120 R:
130 R:       COPYRIGHT 1982 BY
140 R:        Robert A. Kahn
150 R:        Ted M. Kahn &
160 R:        Dale Disharoon
170 R:
180 POS:5,1
190 T:A SLIGHTLY FRACTURED FAIRY TALE
200 POS:2,4
210 T:WHAT'S YOUR NAME?  \
220 A:$NAME
230 T:
240 :AND THE NAME OF
250 :YOUR BEST FRIEND?  \
260 A:$FRIEND
270 T:
280 :THE NAME OF YOUR
290 :FAVORITE RELATIVE?  \
300 A:$RELATIVE
310 T:
320 :JOB TITLE OF A KIND OF PERSON
330 :WHO ALWAYS BUGS YOU?  \
340 A:$TITLE
350 T:
360 :THE NAME OF YOUR BOSS?  \
370 A:$BOSS
380 T:[CLEAR]
390 :
400 :JUST A FEW MORE QUESTIONS...
410 :
420 :NAME FOR A TYPE OF
430 :LARGE BUILDING?  \
440 A:$BUILDING
450 T:
460 :ONE WORD FOR WHAT YOU WOULD LIKE
TO
470 :DO WHEN YOU'RE ANGRY?  \
480 A:$ANGRY
490 T:
```

```
500 :NAME OF A LUXURY VEHICLE YOU'VE
510 :ALWAYS WANTED TO OWN?  \
520 A:$VEHICLE
530 T:[CLEAR]
540 POS:7,1
550 T:* THE THREE LITTLE PIGS *
560 :
570 :ONCE THERE WERE THREE LITTLE PIGS
580 :NAMED $NAME, $FRIEND AND $RELATIV
E.
590 :
600 :A HUNGRY $TITLE NAMED $BOSS
610 :WANTED TO EAT THEM FOR SUPPER.
620 :
630 :TERRIFIED, THEY HID IN THEIR $BUI
LDING.
640 :FURIOUS, $BOSS SHOUTED:
650 :"I'LL HUFF, AND I'LL PUFF AND I'L
L
660 :$ANGRY YOUR $BUILDING DOWN!"
670 :
680 :NO FOOLS, THE THREE PIGS SNEAKED
OUT
690 :THE BACK DOOR AND ESCAPED IN THEI
R
700 :$VEHICLE.
710 :
720 :              * THE END *\
730 PA:1200
740 E:
```

Typical Form Letter

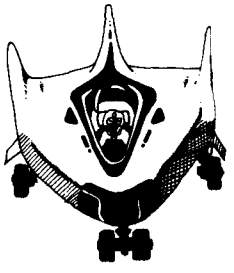
Here's another program that is easy to modify. It works particularly well with a printer and is like a lot of computer-personalized form letters. We chose a letter from our favorite politician, Senator Windbag. You can change the program to address anyone you care to and say what's on your mind.

```
          OFFICE OF
   THE HONORABLE SENATOR
       RALPH WINDBAG


                  SEPTEMBER 3, 1982


DALE DISHAROON
SEA CYPRESS DRIVE
MANCHESTER, CALIFORNIA 95459

DEAR DALE DISHAROON:

I'D LIKE TO THANK YOU, PERSONALLY,
FOR YOUR GENEROUS CONTRIBUTION OF
```

```
$5.00.  GROUPS LIKE YOURS,
NEEDY AUTHORS CO-OP, WHO ARE
WILLING TO STAND UP AND BE COUNTED,
HAVE MADE IT POSSIBLE FOR US TO
TRIUMPH IN OUR AGGRESSIVE CAMPAIGN
AGAINST DEADLINES!

MANY PEOPLE ARE WORRIED THAT NO ONE
HERE IN THE STATE CAPITAL HAS TIME
FOR SMALL GROUPS LIKE YOURS.
BUT I WANT TO REASSURE YOU,
PERSONALLY, THAT NEEDY AUTHORS CO-OP WILL
ALWAYS BE MY PRIME CONCERN WHILE IN
OFFICE.

                SINCERELY YOURS,
                SENATOR RALPH WINDBAG
```

```
100 R: *** TYPICAL FORM LETTER ***
110 R:
120 R:       COPYRIGHT 1982 BY
130 R:        Robert A. Kahn
140 R:        Lee Berman
150 R:        Ted M. Kahn
160 R:        Dale Disharoon
170 R:
180 T:
190 :BE SURE YOUR LINE PRINTER IS ON!
200 :
210 :INPUT DATA FOR FORM LETTER:
220 :
230 :
240 :DATE OF LETTER? \
250 A:$DATE
260 T:
270 :NAME OF CITIZEN? \
280 A:$NAME
290 T:
300 :STREET ADDRESS? \
310 A:$ADDRESS
320 T:
330 :CITY? \
340 A:$CITY
350 T:
360 :STATE? \
370 A:$STATE
380 T:
390 :ZIP CODE? \
400 A:$ZIP
410 T:[CLEAR]
420 POS:2,2
430 T:CONTRIBUTION? \
440 A:$MONEY
450 T:
460 :YOUR GROUP'S NAME? \
470 A:$GROUP
480 T:
```

```
490 :PROBLEM? \
500 A:$PROBLEM
510 POS:6,10
520 T:--------------------------
530 :
540 :TOP MARGIN? \
550 A:#M
560 *MARGIN
570 WRITE:P,
580 C:#M=#M-1
590 J(#M<>0):*MARGIN
600 WRITE:P
610 WRITE:P,            OFFICE OF
620 WRITE:P,     THE HONORABLE SENATOR
630 WRITE:P,         RALPH WINDBAG
640 WRITE:P
650 WRITE:P
660 WRITE:P,                    $DATE
670 WRITE:P
680 WRITE:P
690 WRITE:P,$NAME
700 WRITE:P,$ADDRESS
710 WRITE:P,$CITY, $STATE $ZIP
720 WRITE:P
730 WRITE:P,DEAR $NAME:
740 WRITE:P
750 WRITE:P,I'D LIKE TO THANK YOU, PER
SONALLY,
760 WRITE:P,FOR YOUR GENEROUS CONTRIBU
TION OF
770 WRITE:P,$MONEY.  GROUPS LIKE YOURS
,
780 WRITE:P,$GROUP, WHO ARE
790 WRITE:P,WILLING TO STAND UP AND BE
 COUNTED,
800 WRITE:P,HAVE MADE IT POSSIBLE FOR
US TO
810 WRITE:P,TRIUMPH IN OUR AGGRESSIVE
CAMPAIGN
820 WRITE:P,AGAINST $PROBLEM!
830 WRITE:P
840 WRITE:P,MANY PEOPLE ARE WORRIED TH
AT NO ONE
850 WRITE:P,HERE IN THE STATE CAPITAL
HAS TIME
860 WRITE:P,FOR SMALL GROUPS LIKE YOUR
S.
870 WRITE:P,BUT I WANT TO REASSURE YOU
,
880 WRITE:P,PERSONALLY, THAT $GROUP WI
LL
890 WRITE:P,ALWAYS BE MY PRIME CONCERN
 WHILE IN
900 WRITE:P,OFFICE.
910 WRITE:P
920 WRITE:P,             SINCERELY YOUR
S,
930 WRITE:P,             SENATOR RALPH
WINDBAG
```



Eliza or the World's Cheapest Psychiatrist

The Eliza program, first written by Joe Weizenbaum of MIT, is an example of what could be called empty interaction. The computer asks you questions and seems to be responding in a very personal way when in fact it is responding mechanically. Note that at line 240 it matches for misery and responds accordingly. At line 380 it responds to a desire to end the program and at line 530 at negatives. This is an interesting program to study as it is cleverly designed to create the illusion of conversation and consolation. It is also fun (and easy) to modify it and turn Eliza into a Freudian, a sadist, or a callous joker.

```
MY NAME IS ELIZA.  I AM A COMPUTER
PROGRAM.
WHAT IS YOUR NAME? ANDY
HELLO, ANDY.
HOW ARE YOU TODAY? I'M FINE, THANKS.
I AM FEELING FINE, TOO.
PLEASE TELL ME ABOUT YOUR PROBLEMS.
WELL, JUST BETWEEN THE TWO OF US, I'VE
ALWAYS WANTED TO BE A COMPUTER PRO-
GRAMMER. THEY'RE SO FASCINATING.
I SEE, ANDY.
DO YOU IDOLIZE THEM TOO?
LET'S TALK ABOUT YOU, NOT ME.
I THOUGHT AS MUCH. ANYWAY, I THINK
IT'S JUST AWESOME TO HAVE SUCH POWER
AT YOUR FINGERTIPS.
LET'S TALK ABOUT YOU, NOT ME.
WELL, OF COURSE COMPUTER PROGRAMS
DON'T HAVE FINGERS, MUCH LESS FINGER-
TIPS.
AREN'T YOU BEING A BIT NEGATIVE?
NO, JUST FACTUAL.
HMMM. . .
PROFOUND, HUH?
```

```
2 R:         *** ELIZA ***
4 R:ORIGINAL BY J. WEIZENBAUM, M.I.T.
6 R:SIMPLIFIED & ADAPTED VERSION BY
8 R:CAROL SHAW & HARRY STEWART (1980)
100 C:#M=32767
110 T:MY NAME IS ELIZA.  I AM A COMPUT
ER PROGRAM.
120 T:WHAT IS YOUR NAME? \
130 A:$NAME
140 MS:IS_
150 AY:$NAME=$RIGHT
160 T:HELLO, $NAME
170 T:HOW ARE YOU TODAY? \
180 A:
190 MS:FINE,GOOD,WELL,OK
200 JN:*NOTOK
210 T:I AM FEELING $MATCH, TOO
220 J:*ABC
230 *NOTOK
240 MS:AWFUL,BAD,BLAH,MISERABLE,TERRIB
LE
250 JN:*NOTAWFUL
260 T:I AM SORRY YOU ARE FEELING $MATC
H.
```

```
270 :WHAT IS WRONG?
280 J:*LOOP
290 *NOTAWFUL
300 T:HMMM. . .
310 *ABC
320 T:PLEASE TELL ME ABOUT YOUR PROBLE
MS.
330 *LOOP
340 A:$INPUT
350 M:YOU
360 TY:LET'S TALK ABOUT YOU, NOT ME.
370 JY:*LOOP
380 MS:BYE,QUIT,STOP
390 JN:*NOTBYE
400 T:ARE YOU SURE YOU WANT TO $MATCH?
410 A:
420 M:Y
430 TY:OK, GOODBYE.
440 EY:
450 T:ALL RIGHT, LET'S KEEP GOING.
460 J:*LOOP
470 *NOTBYE
480 M:?
490 CY:#A=?
500 TY(#A):THAT IS FOR YOU TO DECIDE.
510 TY(#A<0):THAT IS UP TO YOU.
520 JY:*LOOP
530 M:NO , NO , NOT , NOT,HATE,DON'T,D
ONT
540 C:#A=?
550 TY(#A):AREN'T YOU BEING A BIT NEGA
TIVE?
560 TY(#A<0):PLEASE BE MORE POSITIVE.
570 JY:*LOOP
580 M:!
590 TY:DON'T GET EXCITED!
600 JY:*LOOP
610 A:$RIGHT=$INPUT_
620 U:*ATOM
630 U:*ATOM
640 JY:*SKIP2
650 J(?):*NOMATCH
660 T:PLEASE ELABORATE
670 J:*LOOP
680 *SKIP2
690 J(? <25000):*NOMATCH
700 *SKIP
710 C:$LAST=$LEFT
720 U:*ATOM
730 JY:*SKIP
740 A:=$LAST
750 MS:.,!
760 AY:$LAST=$LEFT
770 C:#A=?
780 T(#A):TELL ME MORE ABOUT $LAST.
790 T(#A<0):WHAT DO YOU MEAN BY $LAST?
800 J:*LOOP
810 *NOMATCH
820 C:#A=?
830 C(#A<0):#A=-#A
840 C:#A=#A/(32767/6)
```

```
850 T(#A=0):HMMM. . .
860 T(#A=1):GO ON
870 T(#A=2):MM HMM. . .
880 T(#A=3):I SEE, $NAME
890 T(#A=4):THAT'S VERY INTERESTING.
900 T(#A>=5):HOW DO YOU FEEL ABOUT THA
T?
910 J:*LOOP
920 *ATOM A:=$RIGHT
930 MS:[RIGHT]_
940 EN:
950 A:=$LEFT
960 E:
```

Twenty Questions

This last program in the chapter is an Eliza-type program in which the computer is doing nothing but teasing you into thinking that it understands and answers your questions. Possibly a lot of actual human conversation proceeds in such an automatic way.

Notice that in line 260 the program matches for A?, E?, O?, U?, and Y?. Whenever your question ends that way, the computer answers your question YES. Otherwise, the computer answers NO. You can change these match criteria in any way you want.

```
NO

ENTER QUESTION NUMBER 2
IS IT A MINERAL?
NO

ENTER QUESTION NUMBER 3
IS IT GREEN?
NO

ENTER QUESTION NUMBER 4
IS IT ORANGE?
YES

ENTER QUESTION NUMBER 5
IS IT A FRUIT?
NO

ENTER QUESTION NUMBER 6
IS IT A VEGETABLE?
YES

ENTER QUESTION NUMBER 7
IS IT A CARROT?
```

```
2 R:          *** TWENTYQ ***
4 R:     A VARIANT OF THE GAME OF
6 R:    THE GAME OF '20 QUESTIONS'
8 R:        COPYRIGHT (C) 1981
10 R:        BY DAVE THORNBURG
100 *20QUESTIONS
```

```
110 T:WELCOME TO THE GAME OF TWENTY QU
ESTIONS.  BY ASKING QUESTIONS WHICH HA
VE \
120 T:YES OR NO ANSWERS, TRY TO GUESS
THE OBJECT THAT HAS BEEN SELECTED.
130 T:
140 T:BE SURE TO END EACH QUESTION WIT
H A QUESTION MARK.
150 T:
160 T:
170 C:#C=0
180 *ROUND
190 C:#C=#C+1
200 *QUESTION
210 T:ENTER QUESTION NUMBER #C
220 A:
230 M:?
240 TN:THAT ISN'T A QUESTION.  PLEASE
ASK A QUESTION.
250 JN:*QUESTION
260 M:A?,E?,I?,O?,U?,Y?
270 PA:?\150
280 TY:YES
290 TN:NO
300 T:
310 J(#C<20):*ROUND
320 T:END OF TWENTY QUESTIONS.  PRESS
RETURN TO START AGAIN.
330 A:
340 J:*20QUESTIONS
350 E:
```

More Modules

Here are a number of word modules that advanced programmers in PILOT can use within their programs:

*NEXTWORD gets you the next word in a string or sentence after a word that you specify.

*COUNTWORDS counts the number of words in a list.

*RTHWORD gives you the Rth word of a list.

*RANDWORD picks a word at random from a list.

*FIRSTWORD gives the first word of a list.

*LASTWORD gives the last word of a list.

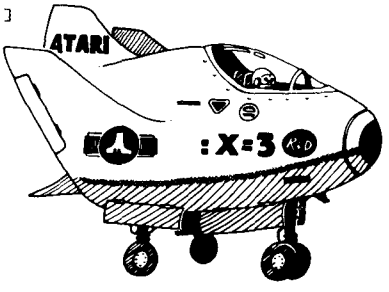
*BUTFIRSTWORD gives you all the words in a list except the first word.

*BUTLASTWORD gives you all the words in a list except the last word.

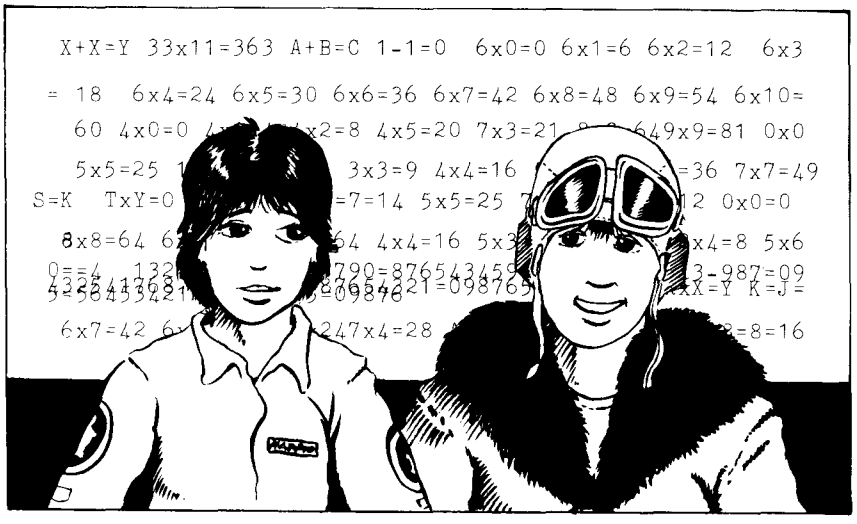
*REVERSE reverses the order of the words in a list.

```
1000 *NEXTWORD
1010 R:FINDS THE NEXT 'WORD' (SEPARATE
D BY BLANKS) IN THE STRING, $SENTENCE
1020 R:RETURNS $WORD AND #E=1 WHEN EMP
TY
1030 A:=$SENTENCE
1040 MS:_
1050 CN:#E=1
1060 EN:
1065 C:#E=0
1070 C:$SENTENCE=$RIGHT
1080 C:$WORD=$LEFT
1090 E:[*NEXTWORD]
1098 R:
1099 R:
2000 *COUNTWORDS
2010 R:COUNT THE NUMBER OF WORDS IN A
LIST ($LIST)
2020 R:$SENTENCE=(LIST)
2025 C:$SENTENCE=$LIST
2030 C:#W=0
2040 *COUNT
2050 U:*NEXTWORD
2060 C(#E=0):#W=#W+1
2070 J(#E=0):*COUNT
2080 E:[*COUNTWORDS]
2098 R:
2099 R:
3000 *RTHWORD
3010 R:FIND THE Rth WORD IN A LIST ($L
IST)
3020 R:SET #R BEFOREHAND
3025 R:USES $TEMP -- COMES IN HANDY LA
TER ON
3030 C:$SENTENCE=$LIST
3040 C:#W=0
3045 C:$TEMP=_
3050 *NEXT
3060 U:*NEXTWORD
3070 C:#W=#W+1
3080 E(#W=#R):[*RTHWORD]
3085 C:$TEMP=$TEMP $WORD
3090 J:*NEXT
3098 R:
3099 R:
4000 *RANDWORD
4010 R:#R=RANDOM NUMBER < LENGTH OF LI
ST (FROM *COUNTWORDS)
4020 R:USES $LIST
4030 C:$SENTENCE=$LIST
4040 U:*COUNTWORDS
4050 C:#R=?\#W+1
4060 U:*RTHWORD
4070 E:[*RANDWORD]
4098 R:
4099 R:
4100 *FIRSTWORD
4110 R:RETURN FIRST WORD IN A LIST
4120 C:#R=1
4130 U:*RTHWORD
```

```
4140 E:[*FIRSTWORD]
4198 R:
4199 R:
4200 *LASTWORD
4210 R:RETURN LAST WORD IN A LIST
4220 U:*COUNTWORDS
4225 J(#W=0):*ENDLAST
4230 C:#R=#W
4240 U:*RTHWORD
4250 *ENDLAST E:[*LASTWORD]
4298 R:
4299 R:
4300 *BUTFIRSTWORD
4310 R:RETURN ALL BUT THE FIRST WORD O
F A LIST IN $WORDS
4315 C:$SENTENCE=$LIST
4320 U:*NEXTWORD
4330 C:$WORDS=$SENTENCE
4340 E:[*BUTFIRSTWORD]
4398 R:
4399 R:
4400 *BUTLASTWORD
4410 R:RETURN ALL BUT THE LAST WORD OF
 $LIST (INTO $WORDS)
4420 C:$SENTENCE=$LIST
4430 U:*LASTWORD
4440 C:$WORDS=$TEMP
4470 E:[*BUTLASTWORD]
4480 R:
4490 R:
5000 *REVERSE [$LIST]
5010 R:REVERSE A LIST OF WORDS
5020 R:RETURNS NEW LIST IN $RSENTENCE
5030 C:$RSENTENCE=_
5040 *NEXTR U:*LASTWORD
5045 J((#R=1)*(#W=0)):*ENDREVERSE
5050 C:$RSENTENCE=$RSENTENCE $WORD
5060 U:*BUTLASTWORD
5070 C:$LIST=$WORDS
5080 J:*NEXTR
5090 *ENDREVERSE E:[*REVERSE]
```



4 Math, Logic, and Strategy



The first programs in this chapter are simple and designed for young people who are learning to master arithmetic. The first program can help 4-, 5-, and 6-year-olds who are just learning numbers and counting.

Simple Counting 1

This program picks a number from 0 to 10 and asks the player to type in the number that comes next. If there is a right answer, another problem comes up. If not, the computer prints TRY AGAIN, and the player has another choice.

```
WHAT NUMBER COMES RIGHT AFTER 3? 4

VERY GOOD!
WHAT NUMBER COMES RIGHT AFTER 5? 6

VERY GOOD!
WHAT NUMBER COMES RIGHT AFTER 1? 3
TRY AGAIN ...
WHAT NUMBER COMES RIGHT AFTER 1? 2

VERY GOOD!
WHAT NUMBER COMES RIGHT AFTER 3? 4

VERY GOOD!
WHAT NUMBER COMES RIGHT AFTER 6? 7
```

```
100 T:[CLEAR]
110 *AGAIN R: CHOOSE RANDOM NUMBER 0-1
0
120 C:#X=?\11
130 R: COMPUTE NUMBER+1
140 C:#Y=#X+1
150 *TRYAGAIN T: WHAT NUMBER COMES RIG
HT AFTER #X? \
160 R: GET ANSWER
170 A:#G
180 J(#G=#Y):*GOODGUESS
190 J:*OOPS
200 *GOODGUESS T:
210 T:
220 T: VERY GOOD!
```

```
230 T:
240 J:*AGAIN
250 *OOPS T: TRY AGAIN ...
260 T:
270 J:*TRYAGAIN
280 E:
```

There are a number of simple ways to modify this program to make it more complex. On line 120,

```
120 C:#X=?\11
```

The number 11 can be replaced by any other number according to what you think the players need practice in. For example,

```
120 C:#X=?\50
```

will generate numbers from 0 to 50 for the game.

You can also change lines 140 and 150 to:

```
140 C:#Y=#X+2
150 *TRY AGAIN T:WHAT NUMBER COMES TWO AFTER #X
```

This would be a simple addition problem.

Here are the changes that would create a simple doubling program for numbers from 0 to 15:

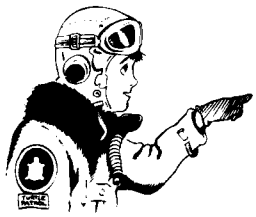
```
120 C:#X=?\16
140 C:#Y=#X*2
150 *TRY AGAIN T: WHAT NUMBER IS TWICE AS BIG AS #Y
```

Notice that on line 150 there is a module name as well as a T: command. PILOT allows you to add a command on line that begins with naming a module.

Simple Counting 2

This program differs from Simple Counting 1 in that it presents problems in sets of five and gives the player the choice of playing again or terminating the program.

```
VERY GOOD!

WHAT NUMBER COMES RIGHT AFTER 10? 11
VERY GOOD!

WHAT NUMBER COMES RIGHT AFTER 9? 11
NO, THAT WAS WRONG.

WHAT NUMBER COMES RIGHT AFTER 9? 10
VERY GOOD!

WHAT NUMBER COMES RIGHT AFTER 7? 8
VERY GOOD!

WANT TO PLAY AGAIN? Y
```



```
100 R: PROBLEMS IN GROUPS OF FIVE
110 *GAME C: #N=5
120 T:[CLEAR]
130 *LOOP C: #X=?\11
140 C: #Y=#X+1
150 *AGAIN T:
160 T: WHAT NUMBER COMES RIGHT AFTER #
X? \
170 A: #G
180 J(#G<>#Y): *WRONG
190 U: *RIGHT
200 J: *KEEPGOING
210 *WRONG U:*OOPS
220 J:*AGAIN
230 *KEEPGOING C: #N=#N-1
240 J(#N>0):*LOOP
250 U:*ANOTHER
260 JY: *GAME
270 T:
280 T: WELL, OKAY. HOPE YOU HAD FUN!
290 E:
300 *RIGHT
310 T:
320 T: VERY GOOD!
330 T:
340 U:*WAIT
350 E:
360 *OOPS
370 T: NO, THAT WAS WRONG.
380 T:
390 E:
400 *ANOTHER
410 U:*WAIT
420 T:
430 T: WANT TO PLAY AGAIN? \
440 A:$A
450 M:YES,Y
460 E:
470 *WAIT
480 PA:40
490 E:
```

Simple Addition Practice 1

This program provides simple addition problems with numbers under 10. The player can choose the number of problems to try. By changing lines 170 and 180 you can control the size of the numbers that you want in the problems.

```
HOW MANY ADDITION PROBLEMS WOULD YOU
LIKE? 5

WHAT IS THE ANSWER?
   7+2=9
  YOU GOT IT!

HERE'S ANOTHER ONE TO TRY:

WHAT IS THE ANSWER?
   4+2=5
   NO, TRY AGAIN...

WHAT IS THE ANSWER?
   4+2=6
  YOU GOT IT!

HERE'S ANOTHER ONE TO TRY:
```

```
100 T:[CLEAR]
110 T:
120 T:HOW MANY ADDITION PROBLEMS WOULD
 YOU LIKE? \
130 A:#N
140 C:#I=0
150 *LOOP R: SIZE OF NUMBERS DEPENDS
160 R: ON VALUES IN NEXT TWO LINES
170 C:#X=?\10+1
180 C:#Y=?\10+1
190 *AGAIN T:
200 T:WHAT IS THE ANSWER?
210 T:    #X+#Y=\
220 A:#Z
230 J(#X+#Y=#Z):*CORRECT
240 T:    NO, TRY AGAIN...
250 J:*AGAIN
260 *CORRECT T:  YOU GOT IT!
270 C:#I=#I+1
280 T:
290 T(#I<#N):HERE'S ANOTHER ONE TO TRY
:
300 T:
310 J(#I<#N):*LOOP
320 T:THANKS FOR PLAYING!
330 E:
```

Simple Addition Practice 2

This program adds a personal touch to the addition program. It asks the player for his or her name and then uses the name in asking how many problems to generate. The name routine on lines 45 to 100 can be used in other programs. For example, it can easily be put at the beginning of either of the counting programs.

```
WHAT'S YOUR NAME? ANDY

HOW MANY ADDITION PROBLEMS WOULD YOU
LIKE, ANDY? 5

WHAT IS THE ANSWER?
   6+1=7
 GREAT JOB, ANDY!
 GREAT JOB, ANDY!
 GREAT JOB, ANDY!
 GREAT JOB, ANDY!
 GREAT JOB, ANDY!

 HERE'S ANOTHER ONE TO TRY:

WHAT IS THE ANSWER?
   10+8=17
 NO, THAT'S NOT RIGHT.  TRY AGAIN...
WHAT IS THE ANSWER?
   10+8=18
```

```
10 R:    *** SIMPLE ADDITION #2 ***
20 R:        COPYRIGHT (C) 1982
30 R:           BY ANDY HALL
40 T:[CLEAR]
45 T:WHAT'S YOUR NAME? \
50 A:$N
```

```
60 T:
100 T:HOW MANY ADDITION PROBLEMS WOULD
 YOU LIKE, $N? \
110 A:#N
115 C:#I=0
120 *LOOP R: SIZE OF NUMBERS DEPENDS
130 R:ON VALUES IN NEXT TWO LINES
140 C:#X=?\10+1
150 C:#Y=?\10+1
160 *AGAIN T:
165 T:WHAT IS THE ANSWER?
170 T:    #X+#Y=\
180 A:#Z
190 J(#X+#Y=#Z): *CORRECT
200 T: NO, THAT'S NOT RIGHT.  TRY AGAI
N...
210 J:*AGAIN
220 *CORRECT
222 C:#C=5
224 *LITTLELOOP T: GREAT JOB, $N!
226 C:#C=#C-1
228 J(#C<>0): *LITTLELOOP
230 C:#I=#I+1
235 T:
240 T(#I<#N): HERE'S ANOTHER ONE TO TR
Y:
250 T:
260 J(#I<#N): *LOOP
270 E:
```

U-Addit

In this program you pick two numbers and add them. There is no limit to the size of numbers picked as long as they are integers. The computer lets you know when you've got the right answer.

```
PICK TWO NUMBERS AND ADD THEM.
  FIRST NUMBER? 4
  SECOND NUMBER? 5
    4+5=9
 YOU GOT IT!
PICK TWO NUMBERS AND ADD THEM.
  FIRST NUMBER? 1
  SECOND NUMBER? 1
    1+1=2
 YOU GOT IT!
PICK TWO NUMBERS AND ADD THEM.
  FIRST NUMBER? 1
  SECOND NUMBER? 2
    1+2=3
```

```
100 T:[CLEAR]
110 *LOOP T:
120 T:PICK TWO NUMBERS AND ADD THEM.
130 T:
140 T:  FIRST NUMBER? \
150 A:#X
```

```
160 T:  SECOND NUMBER? \
170 A:#Y
180 T:
190 *AGAIN T:     #X+#Y=\
200 A:#Z
210 C:#C=#X+#Y
220 J(#Z<>#C):*OOPS
230 T:
240 T: YOU GOT IT!
250 J:*LOOP
260 *OOPS T:
270 T: SORRY, TRY AGAIN.
280 J:*AGAIN
```

Simple Subtraction Practice

This subtraction program is similar to the addition program. However, it must guarantee that the number subtracted from is larger than the number taken away (for example, 9 − 6 is permitted, but 6 − 9 is not). This is done on line 190, which jumps back to a new choice of numbers if #Y>#X. If you want to use larger numbers, change lines 170 and 180.

```
HOW MANY SUBTRACTION PROBLEMS WOULD
YOU LIKE? 5
 WHAT IS THE ANSWER?
   9-6=3
 YOU GOT IT!
HERE'S ANOTHER ONE TO TRY:

 WHAT IS THE ANSWER?
   7-3=4
 YOU GOT IT!
HERE'S ANOTHER ONE TO TRY:

 WHAT IS THE ANSWER?
   9-5=3
   NO, TRY AGAIN...
 WHAT IS THE ANSWER?
   9-5=4
```

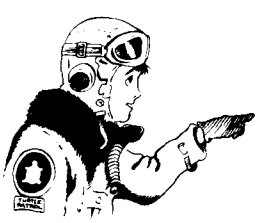
```
100 T:[CLEAR]
110 T:
120 T:HOW MANY SUBTRACTION PROBLEMS WO
ULD YOU LIKE? \
130 A:#N
140 C:#I=0
150 *LOOP R: SIZE OF NUMBERS DEPENDS
160 R: ON VALUES IN NEXT TWO LINES
170 C:#X=?\10+1
180 C:#Y=?\10+1
190 J(#Y>#X):*LOOP
200 *AGAIN T:
210 T: WHAT IS THE ANSWER?
220 T:    #X-#Y=\
230 A:#Z
240 J(#X-#Y=#Z):*CORRECT
250 T:    NO, TRY AGAIN...
```

```
260 J:*AGAIN
270 *CORRECT T: YOU GOT IT!
280 C:#I=#I+1
290 T:
300 T(#I<#N):HERE'S ANOTHER ONE TO TRY
:
310 T:
320 J(#I<#N):*LOOP
330 E:
```

Simple Multiplication Practice

This program is similar to the addition program. Changes in lines 170 and 180 will change the numbers that the program calls up.

```
HOW MANY MULTIPLICATION PROBLEMS
WOULD YOU LIKE? 5

WHAT IS THE ANSWER?
   2*8=16
  YOU GOT IT!

HERE'S ANOTHER ONE TO TRY:

WHAT IS THE ANSWER?
   5*1=5
  YOU GOT IT!

HERE'S ANOTHER ONE TO TRY:

WHAT IS THE ANSWER?
   3*8=27
    NO, TRY AGAIN...

WHAT IS THE ANSWER?
   3*8=24
```

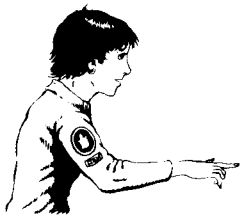


```
100 T:[CLEAR]
110 T:
120 T:HOW MANY MULTIPLICATION PROBLEMS
 WOULD YOU LIKE? \
130 A:#N
140 C:#I=0
150 *LOOP R: SIZE OF NUMBERS DEPENDS
160 R: ON VALUES IN NEXT TWO LINES
170 C:#X=?\10+1
180 C:#Y=?\10+1
190 *AGAIN T:
200 T:WHAT IS THE ANSWER?
210 T:    #X*#Y=\
220 A:#Z
230 J(#X*#Y=#Z):*CORRECT
240 T:     NO, TRY AGAIN...
250 J:*AGAIN
260 *CORRECT T:  YOU GOT IT!
270 C:#I=#I+1
280 T:
290 T(#I<#N):HERE'S ANOTHER ONE TO TRY
:
300 T:
310 J(#I<#N):*LOOP
320 E:
```

Simple Division Practice

This division program must guarantee that there will be no remainders. It does this by choosing two numbers, #X and #Y (lines 145 and 150); multiplying them (line 155); and using the answer #W in defining the problem (line 170). For example, if #X = 5 and #Y = 6, then #W = 30 and the problem presented by the computer would be 30÷6.

```
HOW MANY DIVISION PROBLEMS WOULD YOU
LIKE? 5
WHAT IS THE ANSWER?
   36/4=9
 YOU GOT IT!
 HERE'S ANOTHER ONE TO TRY:

WHAT IS THE ANSWER?
   30/3=11
 NO, THAT'S NOT RIGHT.  TRY AGAIN...
WHAT IS THE ANSWER?
   30/3=10
```



```
100 T:HOW MANY DIVISION PROBLEMS WOULD
 YOU LIKE? \
110 A:#N
115 C:#I=0
120 *LOOP R: SIZE OF NUMBERS DEPENDS
130 R: ON VALUES IN NEXT TWO LINES
140 C:#X=?\10+1
150 C:#Y=?\10+1
155 C:#W=#X*#Y
160 *AGAIN T:
165 T:WHAT IS THE ANSWER?
167 T:
170 T:   #W/#Y=\
180 A:#Z
190 J(#X=#Z):*CORRECT
200 T: NO, THAT'S NOT RIGHT.  TRY AGAI
N...
210 J:*AGAIN
220 *CORRECT T: YOU GOT IT!
230 C:#I=#I+1
240 T(#I<#N): HERE'S ANOTHER ONE TO TR
Y:
250 T:
260 J(#I<#N):*LOOP
270 E:
```

Times-Table Practice

This program lets you choose the times table that you want to practice and then generates as many problems as you can bear using that times table.

```
10 R:    *** TIMES TABLE DRILLS ***
20 R:        COPYRIGHT (C) 1982
30 R:           BY ANDY HALL
100 T:[CLEAR]
110 T: WHAT IS YOUR NAME? \
120 A:$N
130 T:
140 T: WHICH NUMBER DO YOU WANT TO PRA
CTICE, $N? \
150 A:#N
160 *LOOP R: SIZE OF NUMBER DEPENDS
170 R:ON VALUE IN NEXT LINE.
180 C:#X=?\10+1
190 *AGAIN T:
200 T: WHAT IS THE ANSWER, $N?
210 T:
220 T:  #X*#N=\
230 A:#Z
240 T:
250 C:#C=#X*#N
260 J(#Z=#C):*CORRECT
270 T: NO, TRY AGAIN...
280 J:*AGAIN
290 *CORRECT T: YOU GOT IT, $N!  HERE'
S ANOTHER ONE FOR YOU.
300 J: *LOOP
```

Getting to 100

In this program you race the computer to 100. You can play any number from 1 to 10. The computer will then play a number which is added to your number. The player who adds the number that makes exactly 100 wins. Here's a possible game:

		TOTAL	
YOU	10		
COMPUTER	3	13	
Y	9	22	
C	10	32	
Y	7	39	
C	9	48	
Y	10	58	
C	10	68	
Y	5	73	
C	9	82	
Y	7	89	
C	9	98	
Y	2	100	YOU WIN

This is a strategy game. If you think it through, there is a decent chance that you can beat the computer.

The bare-bones program that gets you to 100 by adding each number to the total of the previous numbers follows.

```
100 C:#T=0
110 *LOOP A:#X
120 C:#T=#T+#X
130 T:TOTAL=#T
140 J(#T<100):*LOOP
150 T:  YOU WIN.
160 T:
```

Line 120 is the essential one. It keeps a tally and adds your new input (at line 110) to the total each time through *LOOP until 100 is reached.

```
 THE OBJECT OF THIS GAME IS TO BE THE
FIRST TO REACH 100.  YOU MAY USE ONLY
NUMBERS FROM 1 TO 10.

 EACH TIME YOU ENTER A NUMBER IT IS
ADDED TO THE TOTAL.

 TYPE A NUMBER BETWEEN 1 AND 10. 9
              TOTAL=9
MY NUMBER IS 6.
              TOTAL=15


 TYPE A NUMBER BETWEEN 1 AND 10. 10
```

```
2 R:     *** GETTING TO 100    ***
4 R:         COPYRIGHT (C) 1982
6 R:            BY ANDY HALL
8 R:
10 R:    FROM THE BASIC PROGRAM BY
12 R:     HERB KOHL AND TED KAHN
14 R:
100 T:[CLEAR]
110 T: THE OBJECT OF THIS GAME IS TO B
E THE FIRST TO REACH 100.  YOU MAY USE
 ONLY NUMBERS FROM 1 TO 10.
120 T:
130 T: EACH TIME YOU ENTER A NUMBER IT
 IS ADDED TO THE TOTAL.
140 C:#T=0
150 *LOOP T:
160 T: TYPE A NUMBER BETWEEN 1 AND 10.
 \
170 A:#X
180 T((#X<1)+(#X>10)): NUMBER MUST BE
BETWEEN 1 AND 10.
190 J((#X<1)+(#X>10)):*LOOP
200 C:#T=#T+#X
210 T:
220 T:              TOTAL=#T
230 T:
240 J(#T>=100):*YOUWIN
250 C(#T<90):#G=?\10+1
260 C(#T>=90):#G=100-#T
270 PA:40
280 T:MY NUMBER IS #G.
```

```
290 C:#T=#T+#G
300 T:
310 T:                TOTAL=#T
320 T:
330 J(#T>=100):*IWIN
340 T:
350 J:*LOOP
360 *YOUWIN PA:60
370 T:[CLEAR]
380 *ULOOP T:          YOU WIN!!
390 U:*SOUND
400 J:*ULOOP
410 *IWIN PA:60
420 T:[CLEAR]
430 *ILOOP T:           I WIN!!
440 U:*SOUND
450 J:*ILOOP
460 *SOUND C:#N=?\32
470 SO:#N
480 E:
```

Guess My Number 1

In this program the computer picks a number between 1 and 10 (by now you should know how to change this). You must guess the computer's number. You can also play again if you care to.

```
NO, THAT'S NOT IT. TRY AGAIN.
    YOUR GUESS? 2
NO, THAT'S NOT IT. TRY AGAIN.
    YOUR GUESS? 3
NO, THAT'S NOT IT. TRY AGAIN.
    YOUR GUESS? 4
NO, THAT'S NOT IT. TRY AGAIN.
    YOUR GUESS? 5
NO, THAT'S NOT IT. TRY AGAIN.
    YOUR GUESS? 6
NO, THAT'S NOT IT. TRY AGAIN.
    YOUR GUESS? 7
NO, THAT'S NOT IT. TRY AGAIN.
    YOUR GUESS? 8
 VERY GOOD!
 DO YOU WANT TO PLAY AGAIN? Y
```

```
100 *AGAIN T:[CLEAR]
110 T:  I HAVE GUESSED A NUMBER BETWEE
N
120 T:  ONE AND TEN.  TRY TO GUESS IT.
130 C:#N=?\11+1
140 *LOOP T:
150 T:     YOUR GUESS? \
160 A:#G
170 J(#G<>#N):*WRONG
180 T: VERY GOOD!
190 T:
200 T: DO YOU WANT TO PLAY AGAIN? \
210 A:$Y
220 M:Y
```

```
230 JY:*AGAIN
240 E:
250 *WRONG T:NO, THAT'S NOT IT. TRY AG
AIN.
260 J:*LOOP
```

Guess My Number (with Hints)

In this version you can set the maximum number to be chosen by the computer. You will also get hints about whether your number is too high or too low, which should make your guessing strategy more efficient.

```
  I HAVE GUESSED A NUMBER BETWEEN
  1 AND 100.  TRY TO GUESS IT.

    YOUR GUESS? 50
NO, THAT'S TOO LOW.  TRY AGAIN.

    YOUR GUESS? 75
NO, THAT'S TOO HIGH.  TRY AGAIN.

    YOUR GUESS? 62
NO, THAT'S TOO HIGH.  TRY AGAIN.

    YOUR GUESS? 56
NO, THAT'S TOO HIGH.  TRY AGAIN.

    YOUR GUESS? 53
NO, THAT'S TOO HIGH.  TRY AGAIN.

    YOUR GUESS? 52
```

```
2 R:    *** GUESS MY NUMBER     ***
4 R:        COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
100 T:[CLEAR]
110 T:WHAT SHALL THE MAXIMUM NUMBER BE
? \
120 A:#M
130 *AGAIN T:[CLEAR]
140 T:  I HAVE GUESSED A NUMBER BETWEE
N
150 T:  1 AND #M. TRY TO GUESS IT.
160 C:#C=0
170 C:#N=?\(#M+1)+1
180 *LOOP T:
190 C:#C=#C+1
200 T:    YOUR GUESS? \
210 A:#G
220 J(#G<>#N):*WRONG
230 T: VERY GOOD! YOU GOT IT IN #C GUE
SSES!
240 T:
250 T: DO YOU WANT TO PLAY AGAIN? \
260 A:$Y
270 M:N
280 JN:*AGAIN
290 E:
300 *WRONG T(#G>#N):NO, THAT'S TOO HIG
H. \
```

```
310 T(#G<#N):NO, THAT'S TOO LOW. \
320 T:TRY AGAIN.
330 T:
340 J:*LOOP
```

I Guess Your Number

This program turns the table. You pick a number and give the computer hints. Try to figure out the computer's strategy and see how that strategy has been built into the program.

```
  OKAY, I'LL GUESS AND YOU TYPE:
   "H" IF MY GUESS IS HIGH,
   "L" IF MY GUESS IS LOW, AND
   "R" IF MY GUESS IS RIGHT.

 HERE WE GO!

1. MY GUESS: 50          H
2. MY GUESS: 25          L
3. MY GUESS: 37          L
4. MY GUESS: 43          L
5. MY GUESS: 46          L
6. MY GUESS: 48          L
7. MY GUESS: 49          R
```

```
2 R:     *** GUESS YOUR NUMBER ***
4 R:         COPYRIGHT (C) 1982
6 R:            BY ANDY HALL
100 *AGAIN T:[CLEAR]
110 T: THINK OF A NUMBER BETWEEN 1 AND
 100.
120 T: WHEN YOU'VE GOT IT, HIT START.
130 R: WAIT UNTIL "START" IS PRESSED
140 *LLOOP J(@B53279<>6):*LLOOP
150 T:[CLEAR]
160 T:   OKAY, I'LL GUESS AND YOU TYPE:
170 T:     "H" IF MY GUESS IS HIGH,
180 T:     "L" IF MY GUESS IS LOW, AND
190 T:     "R" IF MY GUESS IS RIGHT.
200 T:
210 C:#C=1
220 T: HERE WE GO!
230 C:#L=0
240 C:#G=50
250 C:#H=101
260 *LOOP T:
270 T:#C. MY GUESS: #G          \
280 A:$E
290 M:H,L,R
300 JM:*HIGH,*LOW,*RIGHT
310 T:      H, L, OR R, PLEASE.
320 J:*LOOP
330 *HIGH C:#H=#G
340 J((#H-#L)=1):*OOPS
350 C:#G=(#H-#L)/2+#L
360 C:#C=#C+1
370 J:*LOOP
380 *LOW C:#L=#G
```

```
390 J((#H-#L)=1):*OOPS
400 C:#G=(#H-#L)/2+#L
410 C:#C=#C+1
420 J:*LOOP
430 *RIGHT T:
440 T: I GUESSED YOUR NUMBER IN #C TRI
ES!
450 T: WANT TO TRY ME AGAIN? \
460 A:$Y
470 M:Y
480 JY:*AGAIN
490 T:
500 T: THANKS FOR PLAYING!
510 E:
520 *OOPS T:
530 T: SOMETHING'S WRONG. ARE YOU SURE
 YOU GAVE ME ALL THE RIGHT INFORMATION
? \
540 A:
550 T:
560 T: LET'S TRY IT AGAIN THEN.
570 PA:200
580 J:*AGAIN
```

Simple Algebra

Here are some simple algebraic equations generated by the computer for you to solve. You can make them more complex and change the size of the numbers involved. These equations provide a good practice for a beginning algebra student as well as an introduction to algebra for youngsters who haven't been exposed to it.

```
HERE IS A SIMPLE ALGEBRA PROBLEM.

 WHAT IS THE CORRECT VALUE FOR X IF
    9+X=10? 1
WELL DONE!  NOW TRY THIS ONE:

 WHAT IS THE CORRECT VALUE FOR X IF
    6+X=10? 3
      SORRY, TRY AGAIN...
 WHAT IS THE CORRECT VALUE FOR X IF
    6+X=10? 4
WELL DONE!  NOW TRY THIS ONE:

 WHAT IS THE CORRECT VALUE FOR X IF
    7+X=8? 1
```

```
100 T:[CLEAR]
110 T:HERE IS A SIMPLE ALGEBRA PROBLEM
.
120 *LOOP C:#X=?\10+1
130 C:#Y=?\10+1
140 J(#X>#Y):*LOOP
150 T:
```

```
160 *AGAIN T:
170 T: WHAT IS THE CORRECT VALUE FOR X
 IF
180 T:     #X+X=#Y? \
190 A:#G
200 C:#C=#Y-#X
210 U(#G=#C):*GOOD
220 J(#G=#C):*LOOP
230 T:
240 T:       SORRY, TRY AGAIN...
250 J:*AGAIN
1000 *GOOD R: GOOD ANSWER
1010 T:
1020 T:WELL DONE!  NOW TRY THIS ONE:
1030 E:
```

More Algebra Problems

```
HERE IS A SIMPLE EQUATION.
PICK A NUMBER N AND SOLVE IT FOR X.

   (N*1)+(N*1)=X

   WHAT VALUE FOR N? 3

WHEN N=3, WHAT DOES X EQUAL? 6
   YOU GOT IT!

HERE IS A SIMPLE EQUATION.
PICK A NUMBER N AND SOLVE IT FOR X.

   (N*5)+(N*1)=X

   WHAT VALUE FOR N? 2

WHEN N=2, WHAT DOES X EQUAL? 14

     NO, TRY AGAIN...
WHEN N=2, WHAT DOES X EQUAL? 12
```

```
100 T:[CLEAR]
110 *LOOP C:#X=?\21+1
120 C:#Y=?\21+1
130 T:HERE IS A SIMPLE EQUATION.
140 T:PICK A NUMBER N AND SOLVE IT FOR
 X.
150 T:
160 T:   (N*#X)+(N*#Y)=X
170 T:
180 T:  WHAT VALUE FOR N? \
190 A:#N
200 C:#C=#N*(#X+#Y)
210 T:
220 *AGAIN T:WHEN N=#N, WHAT DOES X EQ
UAL? \
230 A:#G
240 J(#C=#G):*GOOD
250 T:
260 T:     NO, TRY AGAIN...
270 J:*AGAIN
280 *GOOD T:  YOU GOT IT!
290 T:
300 J:*LOOP
310 E:
```

Binary Boxes

In this game the computer presents you with a series of boxes and asks you two things:

1. Pick any number between 1 and 31.
2. Tell it which boxes the number is in.

The computer will then tell you your number.

```
IN THIS GAME, I WILL HAVE YOU THINK
OF A NUMBER BETWEEN 1 AND 31. I WILL
DISPLAY SOME BOXES OF NUMBERS ON THE
SCREEN, AND WHEN YOU TELL ME WHICH
BOXES CONTAIN YOUR NUMBER, I WILL BE
ABLE TO FIGURE OUT WHAT YOUR NUMBER
IS. SIMPLE, HUH?

          HIT ANY KEY TO BEGIN.
```

#1

1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31

#2

2	3	6	7
10	11	14	15
18	19	22	23
26	27	30	31

#3

4	5	6	7
12	13	14	15
20	21	22	23
28	29	30	31

#4

8	9	10	11
12	13	14	15
24	25	26	27
28	29	30	31

```
PICK A NUMBER
BETWEEN 1 & 31.

WHAT BOXES
CONTAIN YOUR
NUMBER? 2,3,4■
```

#5

16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

In this case the chosen number was 14; the computer's response follows.

#1

1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31

#2

2	3	6	7
10	11	14	15
18	19	22	23
26	27	30	31

#3

4	5	6	7
12	13	14	15
20	21	22	23
28	29	30	31

#4

8	9	10	11
12	13	14	15
24	25	26	27
28	29	30	31

```
YOUR NUMBER WAS 14.
```

#5

16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

Why did the computer guess correctly and easily? Notice the first number in each box—1, 2, 4, 8, 16—or, written in base 2 notation, 2^0, 2^1, 2^2, 2^3, 2^4. Fourteen was in boxes 2, 3, and 4, which added together make $2^1 + 2^2 + 2^3 = 2 + 4 + 8 = 14$. Try any other number and express as a power of 2 according to the boxes it appears in. For example, 25 is in

boxes 1, 4, and 5. That would mean $2^1 + 2^4 + 2^5 = 1 + 8 + 16 = 25$.

In order to see whether you understand how this works, try to figure out why all the odd numbers from 1 to 31 are in box 1.

```
2 R:      *** BINARY BOXES ***
4 R:       COPYRIGHT (C) 1982
6 R:          BY ANDY HALL
100 U:*INSTRUCT
110 U:*BOXES
120 U:*NUMBERS
130 C:#C=0
140 *LOOP U:*GETNUMS
150 U:*SHOWNUM
160 U:*CLEARTEXT
170 C:#C=#C+1
180 J(#C<3):*LOOP
190 U:*TRICK
200 U:*CLEARTEXT
210 CN:#C=0
220 JN:*LOOP
230 U:*KNOWTRICK
240 C:#C=0
250 U:*TEST
260 U:*CLEARTEXT
270 J(#G=#N):*GOOD
280 T:MAYBE YOU DON'T
290 T:KNOW THE TRICK
300 T:I'M THINKING OF.
310 T:DO YOU WANT TO
320 T:SEE SOME MORE
330 T:EXAMPLES? \
340 A:
350 M:Y
360 UY:*CLEARTEXT
370 CY:#C=0
380 JY:*LOOP
390 E:
400 *GOOD T:VERY GOOD!
410 T:YOU GOT IT!
420 PA:300
430 T:SEE HOW LONG
440 T:IT TAKES YOUR
450 T:FAMILY AND
460 T:FRIENDS TO GET
470 T:THE PUZZLE.
480 PA:300
490 E:
500 *INSTRUCT T:[CLEAR]
510 T:
520 T:IN THIS GAME, I WILL HAVE YOU TH
INK OF A NUMBER BETWEEN 1 AND 31. I WI
LL DISPLAY SOME \
530 T:BOXES OF NUMBERS ON THE SCREEN,
AND WHEN YOU TELL ME WHICH BOXES CONTA
IN YOUR NUMBER, I WILL \
540 T:BE ABLE TO FIGURE OUT WHAT YOUR
NUMBER IS. SIMPLE, HUH?
```

```
550 T:
560 T:
570 T:          HIT ANY KEY TO BEGIN.
580 C:@B764=255
590 *ILOOP J(@B764=255):*ILOOP
600 C:@B764=255
610 E:
620 *NUMBERS POS:2,2
630 T:#1 [RIGHT] 1  3  5  7[RIGHT]  #2
 [RIGHT] 2  3  6  7
640 T:   [RIGHT] 9 11 13 15[RIGHT]
 [RIGHT]10 11 14 15
650 T:   [RIGHT]17 19 21 23[RIGHT]
 [RIGHT]18 19 22 23
660 T:   [RIGHT]25 27 29 31[RIGHT]
 [RIGHT]26 27 30 31
670 POS:2,10
680 T:#3 [RIGHT] 4  5  6  7[RIGHT]  #4
 [RIGHT]8  9 10 11
690 T:   [RIGHT]12 13 14 15[RIGHT]
 [RIGHT]12 13 14 15
700 T:   [RIGHT]20 21 22 23[RIGHT]
 [RIGHT]24 25 26 27
710 T:   [RIGHT]28 29 30 31[RIGHT]
 [RIGHT]28 29 30 31
720 POS:2,18
730 T:                    #5 [RIGHT]16 1
7 18 19
740 T:                       [RIGHT]20 2
1 22 23
750 T:                       [RIGHT]24 2
5 26 27
760 T:                       [RIGHT]28 2
9 30 31
770 E:
780 *BOXES T:[CLEAR]
790 C:$TOP=   [Q][11(R)][E]     [Q][11
(R)][E]
800 C:$SID=   ¦           ¦     ¦
      ¦
810 C:$BOT=   [Z][11(R)][Z]     [Z][11
(R)][Z]
820 T:$TOP
830 T:$SID
840 T:$SID
850 T:$SID
860 T:$SID
870 T:$BOT
880 POS:2,9
890 T:$TOP
900 T:$SID
910 T:$SID
920 T:$SID
930 T:$SID
940 T:$BOT
950 POS:2,17
960 T:                          [Q][11(R)][
Q]
970 C:$SID=                           ¦
      ¦
980 T:$SID
```

```
990 T:$SID
1000 T:$SID
1010 T:$SID
1020 T:                          [Z][11(R)]
[Z]
1030 E:
1040 *GETNUMS POS:2,16
1050 C:#S=0
1060 T:PICK A NUMBER
1070 T:BETWEEN 1 & 31.
1080 PA:200
1090 T:
1100 T:WHAT BOXES
1110 T:CONTAIN YOUR
1120 T:NUMBER? \
1130 A:$NUM
1140 MS:28 29
1150 A:=$LEFT
1160 M:1
1170 CY:#S=#S+1
1180 M:2
1190 CY:#S=#S+2
1200 M:3
1210 CY:#S=#S+4
1220 M:4
1230 CY:#S=#S+8
1240 M:5
1250 CY:#S=#S+16
1260 E:
1270 *SHOWNUM U:*CLEARTEXT
1280 T:YOUR NUMBER WAS #S.
1290 PA:300
1300 E:
1310 *CLEARTEXT POS:2,15
1320 T:                                  [R
IGHT]
1330 T:                                  [R
IGHT]
1340 T:                        [RIGHT]
1350 T:                      [RIGHT]
1360 T:                        [RIGHT]
1370 T:                        [RIGHT]
1380 T:                        [RIGHT]
1390 T:                        \
1400 POS:2,16
1410 E:
1420 *TRICK
1430 T:AS YOU MAY HAVE
1440 T:GUESSED, THERE IS
1450 T:A TRICK TO THIS.
1460 T:(AND IT'S NOT THE
1470 T:OBVIOUS ONE OF
1480 T:CHECKING EACH BOX.)
1490 PA:800
1500 U:*CLEARTEXT
1510 T:DO YOU THINK YOU
1520 T:KNOW WHAT IT IS?
1530 A:$Y
1540 M:Y
1550 E:
1560 *KNOWTRICK U:*CLEARTEXT
```

```
1570 T:OKAY, LET'S SEE
1580 T:HOW WELL YOU KNOW
1590 T:THE TRICK.
1600 PA:450
1610 U:*CLEARTEXT
1620 T:I'LL PICK A NUMBER
1630 T:AND TELL YOU WHAT
1640 T:BOXES IT'S IN.
1650 PA:450
1660 U:*CLEARTEXT
1670 T:THEN YOU FIGURE OUT
1680 T:WHAT MY NUMBER IS.
1690 PA:300
1700 E:
1710 *TEST U:*CLEARTEXT
1720 C:#N=?\25+7
1730 C:#M=#N
1740 C:#C=#C+1
1750 T:MY NUMBER IS IN THE
1760 T:FOLLOWING BOXES:
1770 T(#M>15):5 \
1780 C(#M>15):#M=#M-16
1790 T(#M>7):4 \
1800 C(#M>7):#M=#M-8
1810 T(#M>3):3 \
1820 C(#M>3):#M=#M-4
1830 T(#M>1):2 \
1840 C(#M>1):#M=#M-2
1850 T(#M>0):1 \
1860 T:
1870 PA:200
1880 T:WHAT DO YOU THINK
1890 T:MY NUMBER WAS? \
1900 A:$GUESS
1910 MS:24 25
1920 A:#G=$LEFT
1930 E((#N=#G)+(#C=3)):
1940 T:NO, THAT'S NOT IT.
1950 T:TRY ANOTHER ONE.
1960 PA:300
1970 J:*TEST
1980 E:
```

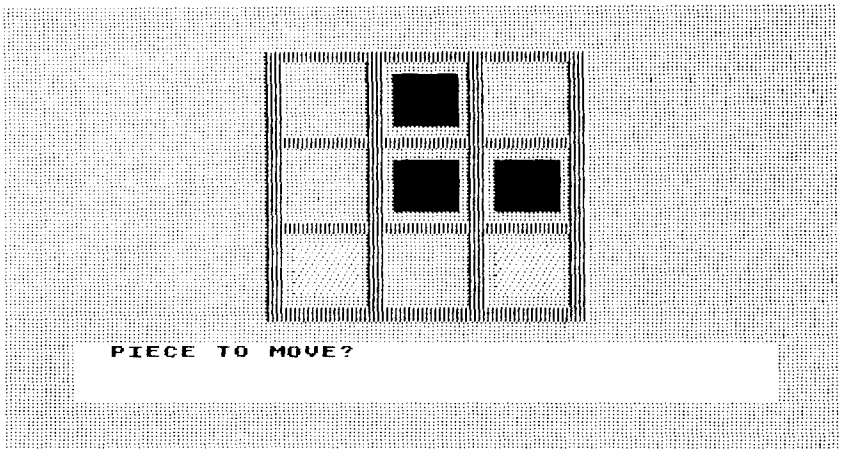
Pawns

The game is played on a 3 × 3 board, with pieces that move in the same way as pawns do in chess. You have three blue pawns, and the computer, your opponent, has three red ones.

The object of the game is to get one of your pawns to the other side of the board, without allowing your opponent to do the same. The first contestant to reach the opposite side wins. (To be honest, though, you may find winning impossible!)

A piece may move only forward one space if it is clear, or one space diagonally forward to capture an enemy piece. Play continues until somebody wins, or a stalemate occurs.

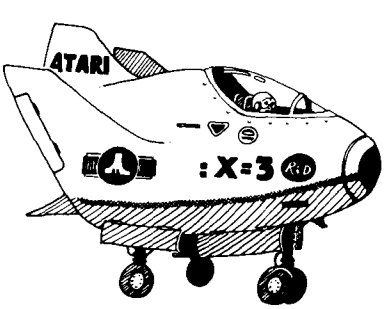
The game requires a joystick, which goes into port 1 of the ATARI Computer.



```
2 R:        *** PAWNS ***
4 R:      COPYRIGHT (C) 1982
6 R:         BY ANDY HALL
100 T:[CLEAR]
110 U:*INSTRUCT
120 *PLAYAGAIN U:*BOARD
130 U:*INITIALIZE
140 U(?\2):*COMPMOVE
150 *LOOP T:[CLEAR]PIECE TO MOVE?  \
160 U:*JOYSTICK
170 C:#Q=#P
180 T:WHERE TO?
190 U:*JOYSTICK
200 C:#R=#P
210 U:*MOVECHECK
220 T(#B=1):
230 J(#B=1):*LOOP
240 C:#P=#Q
250 U:*UPDATE
260 C:#P=#R
270 U:*UPDATE2
280 U:*WINCHECK
290 U:*COMPMOVE
300 U:*WINCHECK
310 J:*LOOP
320 E:
330 *INITIALIZE C:#P=1
340 *ILOOP C:$TEMP=#P
350 C:$$TEMP=1
360 U:*DRAWP
370 C:#P=#P+1
380 J(#P<4):*ILOOP
390 *ILOOP2 C:$TEMP=#P
400 C:$$TEMP=0
410 C:#P=#P+1
420 J(#P<7):*ILOOP2
```

```
430 *ILOOP3 C:$TEMP=#P
440 C:$$TEMP=2
450 U:*DRAWP
460 C:#P=#P+1
470 J(#P<10):*ILOOP3
480 C:#J=-31
490 C:#K=-3
500 E:
510 *BOARD GR:CLEAR;PENYELLOW;GOTO-36,
47;TURNTO90;2(DRAW72;TURN90);DRAW69;TU
RN90;FILL3
520 GR:3(DRAW20;FILL3);GOTO-36,47;TURN
180;FILL72;3(GOTO%X+26,47;DRAW72;GOTO%
X-3,47;FILL72)
530 E:
540 *DRAWP C:#X=#P\3
550 C(#X=1):#X=-30
560 C(#X=2):#X=-7
570 C(#X=0):#X=16
580 C:#Y=41-((#P-1)/3*23)
590 C:$TEMP=#P
600 A:=$$TEMP
610 M:0
620 JY:*ERASE
630 M:1
640 GRN:PENBLUE
650 GRY:PENRED
660 GR:GOTO#X,#Y;TURNTO90;14(DRAW14;GO
TO#X,%Y-1);PENERASE;GOTO%X,%Y
670 E:
680 *ERASE GR:PENERASE;GOTO#X,#Y;TURNT
O90;14(DRAW14;GOTO#X,%Y-1)
690 E:
700 *JOYSTICK U:*BLINK
710 J((%J0+%T8)=0):*JOYSTICK
720 J(%T8=1):*TRIGGER
730 C:#T=%J0
740 C((#T>7)*(#J<10)):#J=#J+23
750 C(#T>7):#T=#T-8
760 C((#T>3)*(#J>-30)):#J=#J-23
770 C(#T>3):#T=#T-4
780 C((#T>1)*(#K>3)):#K=#K-23
790 C(#T>1):#T=#T-2
800 C((#T=1)*(#K<40)):#K=#K+23
810 J:*JOYSTICK
820 *TRIGGER C:#A=#J+31
830 C:#B=#K+3
840 C:#P=(6-(#B/23*3))+(#A/23)+1
850 E:
860 *BLINK GR:PENYELLOW;GOTO#J,#K
870 PA:10
880 GR:PENERASE;GOTO#J,#K
890 PA:10
900 E:
910 *UPDATE C:$TEMP=#P
920 C:$COLOR=$$TEMP
930 C:$$TEMP=0
940 U:*DRAWP
950 E:
960 *UPDATE2 C:$TEMP=#P
970 C:$$TEMP=$COLOR
980 U:*DRAWP
```

```
990 E:
1000 *COMPMOVE R: 1. CHECK FOR WIN SIT
UATION
1010 R: 2. PROTECT HOMEFRONT
1020 R: 3. MAKE OTHER MOVES
1030 R:*** WIN SITUATION? ***
1040 A:=$4
1050 M:1
1060 JN:*CHK5
1070 A:=$7
1080 M:0
1090 JN:*DIAG1
1100 C:#P=4
1110 U:*UPDATE
1120 C:#P=7
1130 U:*UPDATE2
1140 E:
1150 *DIAG1 A:=$8
1160 M:2
1170 JN:*CHK5
1180 C:#P=4
1190 U:*UPDATE
1200 C:#P=8
1210 U:*UPDATE2
1220 E:
1230 *CHK5 A:=$5
1240 M:1
1250 JN:*CHK6
1260 A:=$7
1270 M:2
1280 JN:*STRT1
1290 C:#P=5
1300 U:*UPDATE
1310 C:#P=7
1320 U:*UPDATE2
1330 E:
1340 *STRT1 A:=$8
1350 M:0
1360 JN:*DIAG2
1370 C:#P=5
1380 U:*UPDATE
1390 C:#P=8
1400 U:*UPDATE2
1410 E:
1420 *DIAG2 A:=$9
1430 M:2
1440 JN:*CHK6
1450 C:#P=5
1460 U:*UPDATE
1470 C:#P=9
1480 U:*UPDATE2
1490 E:
1500 *CHK6 A:=$6
1510 M:1
1520 JN:*HOMFRNT
1530 A:=$8
1540 M:2
1550 JN:*CHK9
1560 C:#P=6
1570 U:*UPDATE
1580 C:#P=8
1590 U:*UPDATE2
```

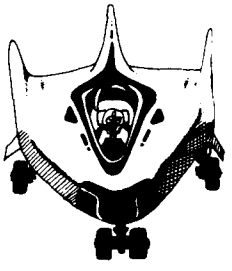


```
1600 E:
1610 *CHK9 A:=$9
1620 M:0
1630 JN:*HOMFRNT
1640 C:#P=6
1650 U:*UPDATE
1660 C:#P=9
1670 U:*UPDATE2
1680 E:
1690 *HOMFRNT R: *** PROTECT HOMEFRONT
 ***
1700 R: *** A. DETERMINE IF IN DANGER
1710 R: *** B. IF SO, ALLEVIATE SITUAT
ION BY CAPTURING OR MOVING
1720 C:$H1=4;5
1730 C:$H2=5;4,6
1740 C:$H3=6;5
1750 C:#C=1
1760 *HLOOP C:$TEMP=H#C
1770 A:=$$TEMP
1780 MS:;
1790 C:$TEMP=#C
1800 A:=$$TEMP
1810 M:1
1820 JY:*CHKDIAG
1830 A:=$$LEFT
1840 M:2
1850 CY:$ATTACKER=$LEFT
1860 JY:*ALLEVIATE
1870 C:#C=#C+1
1880 J(#C<4):*HLOOP
1890 J:*AOK
1900 *CHKDIAG A:=$RIGHT
1910 M:5
1920 JN:*CHECK2
1930 A:=$5
1940 M:2
1950 CY:$ATTACKER=5
1960 JY:*ALLEVIATE
1970 C:#C=#C+1
1980 J(#C<4):*HLOOP
1990 J:*AOK
2000 *CHECK2 A:=$4
2010 M:2
2020 CY:$ATTACKER=4
2030 JY:*ALLEVIATE
2040 A:=$6
2050 M:2
2060 CY:$ATTACKER=6
2070 JY:*ALLEVIATE
2080 C:#C=#C+1
2090 J:*HLOOP
2100 *ALLEVIATE A:=$ATTACKER
2110 M:5,6
2120 JM:*ATT5,*ATT6
2130 C:#P=2
2140 U:*UPDATE
2150 C:#P=4
2160 U:*UPDATE2
2170 E:
2180 *ATT6 C:#P=2
2190 U:*UPDATE
```

```
2200 C:#P=6
2210 U:*UPDATE2
2220 E:
2230 *ATT5 A:=$1$3
2240 M:11
2250 JY:*CHOICE
2260 M:10
2270 CY:#P=1
2280 CN:#P=3
2290 U:*UPDATE
2300 C:#T=#P
2310 C:#N=#P+6
2320 C:$TEMP=#N
2330 A:#N=$$TEMP
2340 A:=$2
2350 M:1
2360 C:#P=5
2370 CY((?\2=1)+(#N=0)):#P=#T+3
2380 U:*UPDATE2
2390 E:
2400 *CHOICE C:#I=1
2410 C:#T=1
2420 A:=$4
2430 M:2
2440 CY:#I=0
2450 A:=$6
2460 M:2
2470 CY:#T=0
2480 J((#T=0)+(#I=0)):*AHEAD
2490 A:=$7
2500 M:2
2510 CY:#P=3
2520 CN:#P=1
2530 J:*MOVIT
2540 C(#I=1):#P=1
2550 C(#T=1):#P=3
2560 *MOVIT U:*UPDATE
2570 C:#P=5
2580 U:*UPDATE2
2590 E:
2600 *AOK R: *** MOVE 'EM OUT ... ***
2610 C:#P=?\3+1
2620 C:$TEMP=#P
2630 A:=$$TEMP
2640 M:1
2650 JN:*AOK
2660 C:#Q=#P+3
2670 C:$TEMP=#Q
2680 A:=$$TEMP
2690 M:0
2700 JN:*AOK
2710 U:*UPDATE
2720 C:#P=#Q
2730 U:*UPDATE2
2740 E:
2750 *WINCHECK R:*** CHECK FOR WIN OR
STALEMATE ***
2760 A:=$1$2$3
2770 M:2
2780 JY:*PLAYERWIN
2790 A:=$7$8$9
2800 M:1
```

```
2810 JY:*COMPWIN
2820 A:=$1$4$7$2$5$8$3$6$9
2830 M:10,02
2840 EY:
2850 A:=$1$5
2860 M:12
2870 EY:
2880 A:=$2$4
2890 M:12
2900 EY:
2910 A:=$2$6
2920 M:12
2930 EY:
2940 A:=$3$5
2950 M:12
2960 EY:
2970 A:=$4$8
2980 M:12
2990 EY:
3000 A:=$5$7
3010 M:12
3020 EY:
3030 A:=$5$9
3040 M:12
3050 EY:
3060 A:=$6$8
3070 M:12
3080 EY:
3090 J:*STALE
3100 *PLAYERWIN T:[CLEAR]
3110 T:  *** YOU WIN ***
3120 J:*END
3130 *COMPWIN T:[CLEAR]
3140 T:  *** I WIN ***
3150 J:*END
3160 *STALE T:[CLEAR]
3170 T:  *** STALEMATE ***
3180 J:*END
3190 *MOVECHECK R: * CHECK VALIDITY OF
 MOVE FROM #Q TO #R; #B=1 IF BAD.
3200 C:#B=0
3210 C:$TEMP=#Q
3220 A:=$$TEMP
3230 M:2
3240 CN:#B=1
3250 EN:
3260 C:#D=#Q-#R
3270 C((#D=2)+(#D=4)):$TEMP=DIAG
3280 C(#D=3):$TEMP=STRAIGHT
3290 A:=$TEMP
3300 M:DIAG,STRAIGHT
3310 CN:#B=1
3320 EN:
3330 JM:*DIAG,*STRT
3340 *DIAG C:$TEMP=#R
3350 A:=$$TEMP
3360 M:1
3370 CN:#B=1
3380 E:
3390 *STRT C:$TEMP=#R
3400 A:=$$TEMP
```

```
3410 M:0
3420 CN:#B=1
3430 E:
3440 *INSTRUCT POS:7,4
3450 T:THIS IS THE GAME OF PAWNS.
3460 POS:3,6
3470 T:YOU WILL NEED A JOYSTICK TO PLA
Y.
3480 POS:5,12
3490 T:WOULD YOU LIKE INSTRUCTIONS? \
3500 A:$R
3510 M:Y
3520 EN:
3530 T:[CLEAR] THE GAME IS PLAYED ON A
 3*3 BOARD,
3540 T:WITH PIECES THAT MOVE IN THE SA
ME WAY
3550 T:AS PAWNS DO IN CHESS.  YOU HAVE
 THREE
3560 T:BLUE PAWNS, AND THE COMPUTER, Y
OUR
3570 T:OPPONENT, HAS THREE RED ONES.
3580 T:
3590 T:  THE OBJECT OF THE GAME IS TO
GET
3600 T:ONE OF YOUR PAWNS TO THE OTHER
SIDE
3610 T:OF THE BOARD, WITHOUT ALLOWING
YOUR
3620 T:OPPONENT TO DO THE SAME.  THE F
IRST
3630 T:CONTESTANT TO REACH THE OPPOSIT
E SIDE \
3640 T:WINS.  (TO BE HONEST, THOUGH, Y
OU \
3650 T:MAY FIND WINNING IMPOSSIBLE!)
3660 T:
3670 T:  A PIECE MAY MOVE ONLY FORWARD
 ONE
3680 T:SPACE IF IT IS CLEAR, OR ONE SP
ACE
3690 T:DIAGONALLY FORWARD TO CAPTURE A
N
3700 T:ENEMY PIECE.  PLAY CONTINUES UN
TIL
3710 T:SOMEBODY WINS, OR A STALEMATE
3720 T:OCCURS.
3730 T:
3740 T:     GOOD LUCK, AND HAVE FUN!
3750 T:
3760 T:        HIT ANY KEY TO BEGIN. \
3770 C:@B764=255
3780 *INSTRLOOP J(@B764=255):*INSTRLOO
P
3790 C:@B764=255
3800 E:
3810 *END T:
3820 R:THIS NEXT TRICKS THE COMPUTER
INTO THINKING IT'S IN THE MAIN PROGRAM
 (NOT A MODULE/SUBROUTINE).
3830 C:@B144=0
```

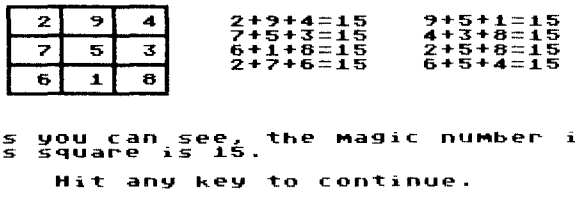


```
3840 T: WANT TO PLAY AGAIN? \
3850 A:$Y
3860 M:N
3870 JN:*PLAYAGAIN
3880 GR:QUIT
3890 E:
```

Lo Shu Squares

A Lo Shu Square is more commonly known as a magic square in America. It is a square that contains numbers arranged in such a way that the numbers in each row, column, and corner-to-corner diagonal add up to the same number.

Here is an example of a 3*3 Lo Shu Square.



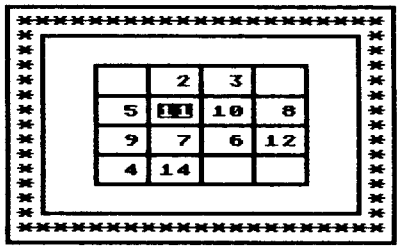
As you can see, the magic number in this square is 15.

One-Player Game. The computer constructs a Lo Shu Square of a given size (3 – 5) but leaves some boxes blank. You must figure out which numbers go in which boxes.

To do this, use the joystick to position the white square in the box that you choose, and press the fire button. Then select the number that you wish to put there. Moving the joystick forward will make the number go down, and pulling it back will make it go up. When you reach the number that you want, press the fire button again, and then you're ready to move on.

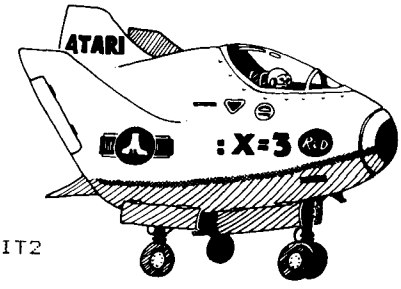
Two-Player Game. The computer once again constructs a Lo Shu Square of a given size. This time, however, one player is asked to leave the room. After this happens, the computer draws the board and allows the remaining player to use the joystick to switch any two boxes.

After the boxes have been switched, the first player comes back and must determine which two boxes were switched. This is done by positioning the white square over the first number, pressing the fire button, and then positioning the square over the other number and pressing the fire button. The computer will let the player know if the answer is right.



```
2 R:      *** LO SHU SQUARES ***
4 R:        COPYRIGHT (C) 1982
6 R:            BY ANDY HALL
100 *AGAIN U:*INITIALIZE
110 GR:QUIT
120 CLOSE:S
130 J(#P=2):*TWOPLAYER
140 U:*BOXMAKE
150 U:*MAGICMAKE
160 U:*BLANKOUT
170 U:*DISPLAY
180 POS:2,22
190 T:_[LEFT]\
200 C:#A=(#S-1)/2
210 C:#B=#A
220 C:#L=@88+(#B*80)+((12-#S)*40)+(#A*
3)+20-(#S*3/2)
230 *MAINLOOP C:@B#L=@B#L\128+128
240 C:#L=#L+1
250 C:@B#L=@B#L\128+128
260 C:#L=#L-1
270 U:*JOYSTICK
280 J(#K=3):*AGAIN
290 U:*CYCLE
300 J(#E=1):*MAINLOOP
310 U:*CHECK
320 J(#M>0):*MAINLOOP
330 C:#C=10
340 U:*WINLOOP
350 J:*AGAIN
360 *WINLOOP C:#N=15
370 *WINLOOP2 C:#N=#N+3
380 SO:#N
390 PA:1
400 J(#N<30):*WINLOOP2
410 C:#C=#C-1
420 J(#C>0):*WINLOOP
430 SO:0
440 PA:240
450 E:
460 E:
470 *TWOPLAYER U:*LEAVE
480 U:*BOXMAKE
490 U:*MAGICMAKE
500 U:*DISPLAY
510 C:#A=(#S-1)/2
520 C:#B=#A
530 C:#L=@88+(#B*80)+((12-#S)*40)+(#A*
3)+20-(#S*3/2)
```

```
540 C:@B#L=@B#L+128
550 C:#L=#L+1
560 C:@B#L=@B#L+128
570 C:#L=#L-1
580 *SWITCHLOOP POS:2,22
590 T:FIRST BOX TO SWITCH?  _[LEFT]\
600 U:*JOYSTICK
610 C:@B#L=@B#L+128
620 C:#L=#L+1
630 C:@B#L=@B#L+128
640 C:#L=#L-1
650 C:#H=#A
660 C:#I=#B
670 POS:24,22
680 T:SECOND? _[LEFT]\
690 *WAIT J(%T8=1):*WAIT
700 U:*JOYSTICK
710 J((#A<>#H)+(#B<>#I)):*GOAHEAD
720 C:@B#L=@B#L\128+128
730 C:#L=#L+1
740 C:@B#L=@B#L\128+128
750 C:#L=#L-1
760 J:*SWITCHLOOP
770 *GOAHEAD U:*SWITCHEM
780 C:#A=(#S-1)/2
790 C:#B=#A
800 SO:13
810 PA:4
820 SO:0
830 PA:4
840 SO:13
850 PA:4
860 SO:0
870 C:#L=@88+(#B*80)+((12-#S)*40)+(#A*
3)+20-(#S*3/2)
880 C:@B#L=@B#L\128+128
890 C:#L=#L+1
900 C:@B#L=@B#L\128+128
910 C:#L=#L-1
920 POS:2,22
930 T:PLAYER #2 MAY RETURN.
  _[LEFT]\
940 PA:200
950 POS:2,22
960 T:WHICH TWO BOXES WERE SWITCHED?
   _[LEFT]\
970 *TWOLOOP U:*JOYSTICK
980 J(#K=3):*AGAIN
990 SO:15
1000 PA:5
1010 SO:0
1020 C:@B#L=@B#L+128
1030 C:#L=#L+1
1040 C:@B#L=@B#L+128
1050 C:#L=#L-1
1060 C:#H=#A
1070 C:#I=#B
1080 *WAIT2 J(%T8=1):*WAIT2
1090 U:*JOYSTICK
1100 J((#A=#H)*(#B=#I)):*WAIT2
1110 SO:25
```



```
1120 PA:5
1130 SO:0
1140 U:*EVALUATE
1150 U(#E=1):*RESTORE
1160 J(#E=1):*TWOLOOP
1170 C:#C=10
1180 U:*WINLOOP
1190 POS:2,22
1200 T:
     _
1210 U:*RESWITCH
1220 POS:2,22
1230 T:_[LEFT]\
1240 PA:240
1250 J:*AGAIN
1260 E:
1270 *INITIALIZE VNEW $
1280 C:@B1373=8
1290 C:@B1374=1
1300 C:@B84=3
1310 C(#Y=0):#Y=8
1320 C(#S=0):#S=3
1330 C(#P=0):#D=1
1340 C(#P=0):#P=1
1350 C(#P=1):$DIFF=#D
1360 C(#P=2):$DIFF=
1370 WRITE:S
1380 WRITE:S,    lo shu squares
1390 C:@B84=8
1400 WRITE:S,    BOX SIZE      #S
1410 C:@B84=11
1420 WRITE:S,    PLAYERS       #P
1430 C:@B84=14
1440 WRITE:S,    DIFFICULTY    $DIFF
1450 C:@B84=17
1460 WRITE:S,    INSTRUCTIONS
1470 POS:1,#Y
1480 WRITE:S,[J]
1490 *ILOOP C:#C=10
1500 *ILOOP1 C:#C=#C-1
1510 J((#C>0)*(@B53279<>7)):*ILOOP1
1520 *ILOOP2 C:#K=@B53279
1530 J(#K=7):*ILOOP2
1540 J(#K=6):*START
1550 J(#K=5):*SELECT
1560 J(#K=3):*OPTION
1570 J:*ILOOP
1580 *SELECT POS:1,#Y
1590 WRITE:S,_
1600 C:#Y=#Y+3
1610 C((#Y=14)*(#P=2)):#Y=17
1620 C(#Y>17):#Y=8
1630 POS:1,#Y
1640 WRITE:S,[J]
1650 C:#N=#Y
1660 SO:#N
1670 PA:2
1680 C:#N=#N+3
1690 SO:#N
1700 PA:2
1710 C:#N=#N+3
```

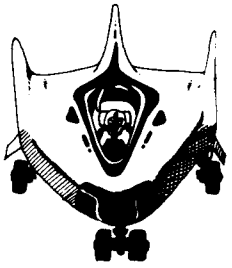
```
1720 SO:#N
1730 PA:2
1740 SO:0
1750 J:*ILOOP
1760 *OPTION
1770 J(#Y>8):*PLAYNUM
1780 C:#S=#S+1
1790 C(#S>5):#S=3
1800 POS:16,8
1810 WRITE:S,#S
1820 SO:14
1830 PA:5
1840 SO:0
1850 PA:5
1860 SO:14
1870 PA:5
1880 SO:0
1890 J:*ILOOP
1900 *PLAYNUM J(#Y>11):*DIFFNUM
1910 C:#P=3-#P
1920 POS:16,11
1930 WRITE:S,#P
1940 POS:16,14
1950 WRITE(#P=1):S,#D
1960 WRITE(#P=2):S,_
1970 SO:17
1980 PA:5
1990 SO:0
2000 PA:5
2010 SO:17
2020 PA:5
2030 SO:0
2040 J:*ILOOP
2050 *DIFFNUM J(#Y=17):*INSTRUCT
2060 C:#D=3-#D
2070 POS:16,14
2080 WRITE:S,#D
2090 SO:20
2100 PA:5
2110 SO:0
2120 PA:5
2130 SO:20
2140 PA:5
2150 SO:0
2160 J:*ILOOP
2170 *START C:#C=30
2180 *STARTLOOP SO:#C
2190 PA:2
2200 C:#C=#C-5
2210 J(#C>15):*STARTLOOP
2220 *STARTLOOP2 SO:#C
2230 PA:4
2240 C:#C=#C+2
2250 J(#C<30):*STARTLOOP2
2260 SO:0
2270 E:
2280 *MAGICMAKE C:#N=#S*#S
2290 J(#S=4):*FOURSQUARE
2300 C:#X=#S/2-1
2310 C:#Z=1
2320 C:#C=0
```

```
2330 *MAKELOOP E(#C=#N):
2340 C:#X=#X+1
2350 C:#C=#C+1
2360 C:#Z=#Z-1
2370 C:#X=(#X+#S)\#S
2380 C:#Z=(#Z+#S)\#S
2390 C:#B=#Z*#S+#X+1
2400 C:$TEMP=BOX#B
2410 A:=$$TEMP
2420 M:BOX
2430 JN:*TAKEN
2440 C(#C<10):$$TEMP= #C
2450 C(#C>9):$$TEMP=#C
2460 J:*MAKELOOP
2470 *TAKEN C:#X=#X-1
2480 C:#Z=#Z+2
2490 C:#X=(#X+#S)\#S
2500 C:#Z=(#Z+#S)\#S
2510 C:#B=#Z*#S+#X+1
2520 C:$TEMP=BOX#B
2530 A:=$$TEMP
2540 M:BOX
2550 JN:*TAKEN
2560 C(#C<10):$$TEMP= #C
2570 C(#C>9):$$TEMP=#C
2580 J:*MAKELOOP
2590 *FOURSQUARE C:#C=1
2600 *FLOOP C:$TEMP=BOX#C
2610 C(#C>9):$$TEMP=#C
2620 C(#C<10):$$TEMP= #C
2630 C:#C=#C+1
2640 J(#C<=16):*FLOOP
2650 C:#C=1
2660 U:*EXCHANGE
2670 C:#C=4
2680 U:*EXCHANGE
2690 C:#C=6
2700 U:*EXCHANGE
2710 C:#C=7
2720 U:*EXCHANGE
2730 E:
2740 *EXCHANGE C:#B=17-#C
2750 C:$TEMP=BOX#C
2760 C:$A=$$TEMP
2770 C:$TEMP2=BOX#B
2780 C:$$TEMP=$$TEMP2
2790 C:$$TEMP2=$A
2800 E:
2810 *BOXMAKE T:[CLEAR]
2820 C:@B755=1
2830 C:@B712=@B710
2840 C:#X=(29-(#S*3))/2
2850 C:#Z=7-#S
2860 C:#C=#S*3+5
2870 C:$CHAR=[R]
2880 U:*STRINGMAKE
2890 C:$CNTRLR=$STRING
2900 C:#C=#S*3+9
2910 C:$CHAR=*
2920 U:*STRINGMAKE
2930 C:$STARS=$STRING
```

```
2940 C:#C=#S*3+5
2950 C:$CHAR=_
2960 U:*STRINGMAKE
2970 C:$MIDDLE=¦*¦$STRING¦*¦
2980 POS:#X,#Z
2990 T:[Q][2(R)]$CNTRLR[2(R)][E]
3000 POS:#X,@B84
3010 T:¦$STARS¦
3020 POS:#X,@B84
3030 T:¦*[Q]$CNTRLR[E]*¦
3040 C:#C=#S*2+3
3050 *BOXLOOP POS:#X,@B84
3060 T:$MIDDLE
3070 C:#C=#C-1
3080 J(#C>0):*BOXLOOP
3090 POS:#X,@B84
3100 T:¦*[Z]$CNTRLR[C]*¦
3110 POS:#X,@B84
3120 T:¦$STARS¦
3130 POS:#X,@B84
3140 T:[Z][2(R)]$CNTRLR[2(R)][C]
3150 C:#C=#S-1
3160 C:$CHAR=[S][2(R)]
3170 U:*STRINGMAKE
3180 C:$MIDDLE=[A][2(R)]$STRING[D]
3190 C:#C=#S
3200 C:$CHAR=   ¦
3210 U:*STRINGMAKE
3220 C:$MIDDLE2=¦$STRING
3230 C:#C=#S-1
3240 C:$CHAR=[W][2(R)]
3250 U:*STRINGMAKE
3260 C:$TOP=[Q][2(R)]$STRING[E]
3270 C:#C=#S-1
3280 C:$CHAR=[X][2(R)]
3290 U:*STRINGMAKE
3300 C:$BOTTOM=[Z][2(R)]$STRING[C]
3310 C:#Z=11-#S
3320 C:#X=19-(#S*3/2)
3330 POS:#X,#Z
3340 T:$TOP
3350 POS:#X,@B84
3360 T:$MIDDLE2
3370 C:#C=#S-1
3380 *BOXLOOP2 POS:#X,@B84
3390 T:$MIDDLE
3400 POS:#X,@B84
3410 T:$MIDDLE2
3420 C:#C=#C-1
3430 J(#C>0):*BOXLOOP2
3440 POS:#X,@B84
3450 T:$BOTTOM
3460 E:
3470 *STRINGMAKE C:$STRING=
3480 *STRINGLOOP C:$STRING=$STRING$CHA
R
3490 C:#C=#C-1
3500 J(#C>0):*STRINGLOOP
3510 E:
3520 *BLANKOUT C:#C=#D*#N/4
3530 C:#M=#C
```

```
3540 *BLOOP C:#R=?\#N+1
3550 C:$TEMP0=CONT#R
3560 C:$TEMP=BOX#R
3570 A:=$$TEMP
3580 M:1,2,3,4,5,6,7,8,9
3590 JN:*BLOOP
3600 C:$$TEMP0=$$TEMP
3610 C:$$TEMP=
3620 C:#C=#C-1
3630 J(#C>0):*BLOOP
3640 E:
3650 *DISPLAY C:#C=1
3660 *DLOOP C:$TEMP=BOX#C
3670 C:#V=20+(3*((#C-1)\#S))-(#S*3/2)
3680 C:#W=12+(2*((#C-1)/#S))-#S
3690 POS:#V,#W
3700 T:$$TEMP
3710 C:#C=#C+1
3720 J(#C<=#N):*DLOOP
3730 C:@B755=2
3740 E:
3750 *JOYSTICK E(%T8=1):
3760 C:#K=@B53279
3770 E(#K=3):
3780 C:#J=%J0
3790 J(#J=0):*JOYSTICK
3800 C:@B#L=@B#L-128
3810 C:#L=#L+1
3820 C:@B#L=@B#L-128
3830 C:#L=#L-1
3840 C(#J\16-7>0):#A=#A+1
3850 C(#J\8-3>0):#A=#A-1
3860 C(#J\4-1>0):#B=#B+1
3870 C(#J\2>0):#B=#B-1
3880 C(#A<0):#A=0
3890 C(#B<0):#B=0
3900 C(#A=#S):#A=#S-1
3910 C(#B=#S):#B=#S-1
3920 C:#L=@88+(#B*80)+((12-#S)*40)+(#A
*3)+20-(#S*3/2)
3930 C:@B#L=@B#L+128
3940 C:#L=#L+1
3950 C:@B#L=@B#L+128
3960 C:#L=#L-1
3970 J:*JOYSTICK
3980 *CYCLE C:#L=#L+1
3990 C:#E=0
4000 C(@B#L<>128):#E=1
4010 C:#L=#L-1
4020 E(#E):
4030 C:#G=0
4040 *LLOOP J(%T8<>0):*LLOOP
4050 *CYCLELOOP E(%T8=1):
4060 C:#J=%J0
4070 J(#J=0):*CYCLELOOP
4080 C(#J\2=1):#G=#G-1
4090 C(#J\4-1>0):#G=#G+1
4100 C(#G<1):#G=#N
4110 C(#G>#N):#G=1
4120 C(#G>9):$A=#G
4130 C(#G<10):$A= #G
```

```
4140 C:#O=@178+5
4150 C:@B#O=@B#O+128
4160 C:#O=#O+1
4170 C:@B#O=@B#O+128
4180 C:#J=(#A*3)+20-(#S*3/2)
4190 C:#K=(#B*2)+12-#S
4200 POS:#J,#K
4210 T:$A
4220 POS:2,21
4230 T:
4240 J:*CYCLELOOP
4250 *CHECK
4260 POS:(#A*3)+20-(#S*3/2),(#B*2)+12-
#S
4270 C:#C=#S**B+#A+1
4280 C:$TEMP=CONT#C
4290 A:=$$TEMP
4300 M:#G
4310 JN:*WRONG
4320 C:$TEMP2=BOX#C
4330 C:$$TEMP2=$$TEMP
4340 T:$$TEMP2
4350 POS:2,21
4360 T:
4370 C:#F=3
4380 *GOOD C:#C=9
4390 *GOODLOOP C:#C=#C+3
4400 SO:#C
4410 J(#C<30):*GOODLOOP
4420 C:#F=#F-1
4430 J(#F>0):*GOOD
4440 SO:0
4450 C:#M=#M-1
4460 E:
4470 *WRONG
4480 C:#C=18
4490 *WRONGLOOP SO:#C
4500 PA:2
4510 C:#C=#C-2
4520 J(#C>=0):*WRONGLOOP
4530 SO:0
4540 T:
4550 POS:2,21
4560 T:__
4570 E:
4580 *LEAVE C:@B1373=8
4590 C:@B1374=1
4600 WRITE:S
4610 C:@B84=4
4620 WRITE:S,  TWO PLAYER lo shu
4630 WRITE:S
4640 C:@B84=10
4650 WRITE:S,      ONE PLAYER
                    SHOULD LEAVE
4660 C:@B84=21
4670 WRITE:S,       HIT start
4680 *LLOOPP J(@B53279<>7):*LLOOPP
4690 *LEAVELOOP C:#C=5
4700 *LEAVELOOP2 J(@B53279=6):*PRESSED
4710 C:#C=#C-1
4720 J(#C>0):*LEAVELOOP2
```



```
4730 C:@B709=202-@B709
4740 J:*LEAVELOOP
4750 *PRESSED C:@B709=202
4760 E:
4770 *SWITCHEM C:#F=#B*#S+#A+1
4780 C:$TEMP=BOX#F
4790 C:#G=#I*#S+#H+1
4800 C:$TEMP2=BOX#G
4810 C:$A=$$TEMP
4820 C:$$TEMP=$$TEMP2
4830 C:$$TEMP2=$A
4840 POS:(#A*3)+20-(#S*3/2),(#B*2)+12-
#S
4850 T:$$TEMP
4860 POS:(#H*3)+20-(#S*3/2),(#I*2)+12-
#S
4870 T:$$TEMP2
4880 E:
4890 *EVALUATE C:#R=#B*#S+#A+1
4900 C:#T=#I*#S+#H+1
4910 C:#E=1
4920 C(((#F+#G)=(#R+#T))*((#F=#R)+(#F=
#T))):#E=0
4930 E:
4940 *RESTORE C:#U=@88+(#B*80)+((12-#S
)*40)+(#A*3)+20-(#S*3/2)
4950 C:@B#U=@B#U\128+128
4960 C:#U=#U+1
4970 C:@B#U=@B#U\128+128
4980 C:#U=@88+(#I*80)+((12-#S)*40)+(#H
*3)+20-(#S*3/2)
4990 C:@B#U=@B#U\128
5000 C:#U=#U+1
5010 C:@B#U=@B#U\128
5020 E:
5030 *RESWITCH C:$TEMP=BOX#T
5040 POS:(#A*3)+20-(#S*3/2),(#B*2)+12-
#S
5050 T:$$TEMP
5060 C:$TEMP=BOX#R
5070 POS:(#H*3)+20-(#S*3/2),(#I*2)+12-
#S
5080 T:$$TEMP
5090 E:
5100 *INSTRUCT GR:QUIT
5110 CLOSE:S
5120 C:@B755=1
5130 T:[CLEAR]
5140 T:
5150 T:  This is the game of LO SHU SQ
UARES.
5160 T:
5170 T:  A Lo Shu Square is more commo
nly known as a magic square here in Am
erica. It is a square that \
5180 T:contians numbers arranged in su
ch a way that the numbers in each row,
 column, and corner-to-corner \
5190 T:diagonal all add up to the same
 number.
5200 T:
```

```
5210 T:       Hit any key to continue.
5220 U:*WAITKEY
5230 T:[CLEAR]
5240 T:
5250 T:  Here is an example of a 3*3 L
o Shu Square.
5260 T:
5270 T:
5280 T:    [Q][2(RRW)][2(R)][E]
5290 T:    ¦ 2¦ 9¦ 4¦    2+9+4=15    9+5+
1=15
5300 T:    [A][2(RRS)][2(R)][D]    7+5+3
=15    4+3+8=15
5310 T:    ¦ 7¦ 5¦ 3¦    6+1+8=15    2+5+
8=15
5320 T:    [A][2(RRS)][2(R)][D]    2+7+6
=15    6+5+4=15
5330 T:    ¦ 6¦ 1¦ 8¦
5340 T:    [Z][2(RRX)][2(R)][C]
5350 T:
5360 T:
5370 T:  As you can see, the magic num
ber in this square is 15.
5380 T:
5390 T:       Hit any key to continue.
5400 U:*WAITKEY
5410 T:[CLEAR]
5420 T:
5430 T:  One-Player Game: The computer
 constructs a Lo Shu Square of a given
 size (3-5) but leaves some \
5440 T:boxes blank. You must figure ou
t which numbers go in which boxes.
5450 T:
5460 T:  To do this, use the joystick
to position the white square in the bo
x you choose, and press the \
5470 T:fire button. Then select the nu
mber you wish to put there. Moving the
 joystick \
5480 T:forward will make the number go
 down, and pulling it back will make i
t go up. When you reach the \
5490 T:number you want, press the fire
 button again, and then you're ready t
o move on.
5500 T:
5510 T:       Hit any key to continue.
5520 U:*WAITKEY
5530 T:[CLEAR]
5540 T:
5550 T:  Two-Player Game: The computer
 once again constructs a Lo Shu Square
 of a given size. This time, \
5560 T:however, one player is asked to
 leave the room. After this happens, t
he computer draws the board \
5570 T:and allows the remaining player
 to use the joystick to switch any two
 boxes.
5580 T:
```

```
5590 T:  After the boxes have been swi
tched, the first player comes back and
 must determine which two boxes \
5600 T:were switched. This is done by
positioning the white square over the
first number, pressing the fire \
5610 T:button, and then positioning th
e square over the other number and pre
ssing the fire button. \
5620 T:The computer will let him know
if he's right or not.
5630 T:
5640 T:       Hit any key to continue.
5650 U:*WAITKEY
5660 T:[CLEAR]
5670 T:
5680 T:  Throughout the game, you need
 only press "OPTION" to return to the
main menu.
5690 T:
5700 T:              Hit any key.
5710 U:*WAITKEY
5720 C:@B755=2
5730 J:*INITIALIZE
5740 *WAITKEY C:@B764=255
5750 *WAITLOOP J(@B764=255):*WAITLOOP
5760 C:@B764=255
5770 E:
```

Pico Fermi Bagel—ATARI PILOT Version

Pico Fermi Bagel is a simple game to learn. The computer picks a number of a certain length, and you try to discover it. You do this by picking a number. The computer then tells you how many digits are correct and in the right position ("perfect"), and how many are correct but in the wrong position ("close").

The option "non-repeating digits" allows you to force the computer to choose a number where all of its digits are different. This is a slightly easier version of the game.

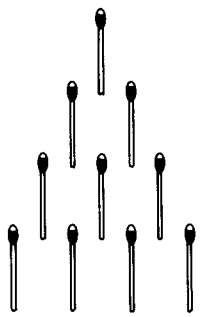
A sample four-digit game follows.

```
                        PERFECT    CLOSE
1. GUESS? 1234            0          1
2. GUESS? 5678            0          2
3. GUESS? 9009            0          1
4. GUESS? 2967            1          0
5. GUESS? 2850            0          2
6. GUESS? 3985            0          1
7. GUESS? 0547            3          0
8. GUESS? 0517
   YOU GOT IT!!
WANT TO PLAY AGAIN? Y
```

```
2 R:     *** PICO FERMI BAGEL ***
4 R:        COPYRIGHT (C) 1982
6 R:             BY ANDY HALL
100 U:*INSTRUCT
110 *PLAY C:#C=1
120 U:*CHOOSE
130 *LOOP U:*GETGUESS
140 U:*EVALUATE
150 J(#T=1):*AGAIN
160 J:*LOOP
170 *AGAIN T:
180 C:#T=0
190 T:WANT TO PLAY AGAIN? \
200 A:$A
210 M:N
220 JN:*PLAY
230 E:
240 *CHOOSE T:[CLEAR]
250 T:
260 T:HOW MANY DIGITS (3-5)? \
270 A:#D
280 J((#D<3)+(#D>5)):*CHOOSE
290 T:
300 T:NON-REPEATING DIGITS? \
310 A:
320 M:Y
330 C:#N=%M
340 C:#L=#D
350 C:$NUM=
360 *CLOOP C:#N=?\10
370 J(#N=0):*REPT
380 A:=$NUM
390 M:#N
400 JY:*CLOOP
410 *REPT C:$NUM=$NUM#N
420 C:#D=#D-1
430 J(#D>0):*CLOOP
440 T:[CLEAR]
450 T:                          PERFECT    CLOS
E
460 E:
470 *GETGUESS T:
480 C:#M=0
490 T:#C. GUESS? \
500 A:$GUESS
510 C:#C=#C+1
520 C:$TEMP=$GUESS
530 *GLOOP C:$A=$TEMP
540 U:*NEXTCHAR
550 C:$TEMP=$A
560 C(#E<>1):#M=#M+1
570 J(#E<>1):*GLOOP
580 E(#L=#M):
590 T:[UP]
600 T:[2(UP)]\
610 C:#C=#C-1
620 J:*GETGUESS
630 *EXACT A:=$GUESS
640 M:$NUM
650 EN:
660 T:
670 T:    YOU GOT IT!!
```

```
680 C:#T=1
690 E:
700 *EVALUATE U:*EXACT
710 E(#T=1):
720 C:$TNUM=$NUM
730 C:$TGESS=$GUESS
740 C:$SNUM=_
750 C:$SGESS=_
760 C:#P=0
770 C:#Q=0
780 *ELOOP C:$A=$TNUM
790 U:*NEXTCHAR
800 J(#E=1):*CLOSE
810 C:$B=$A1
820 C:$TNUM=$A
830 C:$A=$TGESS
840 U:*NEXTCHAR
850 C:$TGESS=$A
860 A:=$B
870 M:$A1
880 CY:$SNUM=$SNUM*
890 CY:$SGESS=$SGESS*
900 CY:#P=#P+1
910 CN:$SNUM=$SNUM$B
920 CN:$SGESS=$SGESS$A1
930 J:*ELOOP
940 *CLOSE C:$TNUM=$SNUM
950 C:$SNUM=_
960 *CLLOOP C:$A=$TNUM
970 U:*NEXTCHAR
980 J(#E=1):*CONT
990 C:$TNUM=$A
1000 A:=$SGESS
1010 MS:$A1
1020 JN:*CLLOOP
1030 A:=$A1
1040 M:*
1050 JY:*CLLOOP
1060 C:$SNUM=$SNUM*
1070 C:$SGESS=$LEFT*$RIGHT
1080 C:#Q=#Q+1
1090 J:*CLLOOP
1100 *CONT T:                          [UP]
#P        #Q
1110 E:
1120 *INSTRUCT T:[CLEAR]
1130 T:WOULD YOU LIKE INSTRUCTIONS? \
1140 A:
1150 M:Y
1160 EN:
1170 T:
1180 T:PICO FERMI BAGEL \
1190 T:IS A SIMPLE GAME TO LEARN. THE
COMPUTER PICKS A NUMBER OF A CERTAIN L
ENGTH, AND YOU TRY TO \
1200 T:DISCOVER IT. YOU DO THIS BY PIC
KING A NUMBER. THE COMPUTER THEN TELLS
 YOU HOW MANY DIGITS ARE CORRECT \
1210 T:AND IN THE RIGHT POSITION ("PER
FECT"), AND HOW MANY ARE CORRECT BUT I
N THE WRONG POSITION ("CLOSE").
1220 T:
```

```
1230 T:THE OPTION "NON-REPEATING DIGIT
S" ALLOWS YOU TO FORCE THE COMPUTER TO
 CHOOSE A NUMBER WHERE ALL OF \
1240 T:ITS DIGITS ARE DIFFERENT. THIS
IS A SLIGHTLY EASIER VERSION OF THE GA
ME.
1250 T:
1260 T:     << HIT ANY KEY TO BEGIN. >>
1270 C:@B764=255
1280 *ILOOP J(@B764=255):*ILOOP
1290 C:@B764=255
1300 E:
1310 *NEXTCHAR
1320 C:#E=0
1330 A:=$A
1340 MS:[2(RIGHT)],
1350 CN:#E=1
1360 EN:
1370 C:$A=$RIGHT
1380 MS:$RIGHT
1390 A:=$LEFT
1400 MS:[RIGHT],
1410 A:=$RIGHT
1420 MS:[2(RIGHT)],
1430 MS:[RIGHT]$RIGHT
1440 C:$A1=$LEFT
1450 E:
```



Nim

NIM is an ancient Chinese gambling game that is usually played with matches. A simple version played with two piles of matches is easy to figure out. Some gamblers use the two-pile version to lure people into playing more complex versions where figuring out a winning strategy is not so easy. Here's the setup for a two-pile version of NIM. (There can be any number of matches in either pile.)

(Pile I)

(Pile II)

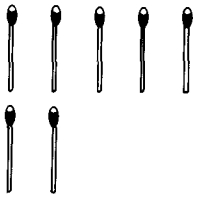
The players are to take turns removing as many matches as they please (at least one) from one of the piles. The players are free to decide which of the piles they will take matches from. The players can also decide whether the object of the game is to take the last match or to force one's opponent to take the last match. Here's an example of the game with the winner being the player who takes the last match:

Setup:

(I)

(II)

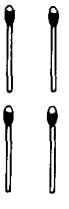
Player 1 (takes 3 matches from pile II):



(I)

(II)

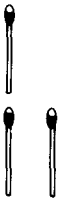
Player 2 (takes 3 matches from pile I):



(I)

(II)

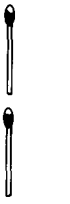
Player 1 (takes 1 match from pile I):



(I)

(II)

Player 2 (takes 1 match from pile II, thus forcing a win for him or herself since Player 1 can only draw matches from one column):



(I)

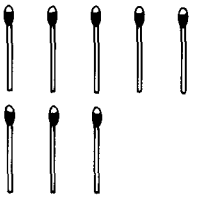
(II)

Player 1 takes match from pile I and Player 2 gets the last match:

(I)

(II)

In this version of the two-pile game, the strategy is to force your opponent to draw from two equal piles. You can then keep equalizing the piles until you get last draw. Here's a simple version of the strategy:

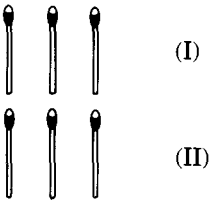


(I)

(II)

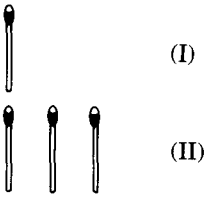
If the first player takes two matches from I, equalizing the columns, the second player can be forced to lose:

After player 1 moves:



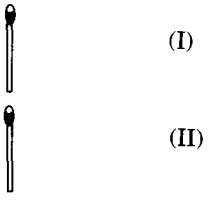
Now the second player can pick from either pile and will still be forced to face equal columns.

Player 2 moves, taking two from column I

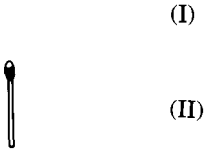


and player 1 counters by equalizing the columns and forcing a win (by guaranteeing that he or she will get the last match).

Player 1:

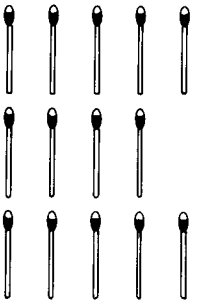


Player 2:



Player 2 takes the last match and wins.

The game of NIM becomes much more complex when three or more piles of matches are involved. Here's a game involving three piles. The rules are the same. Try this a few times before entering the more complex world of NIM that you can explore with the ATARI Computer:

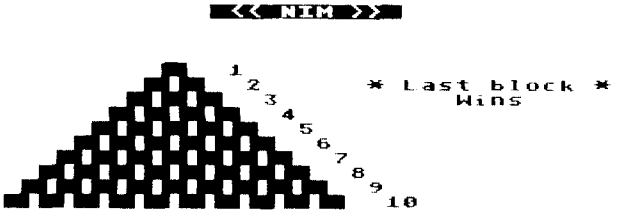


(I)

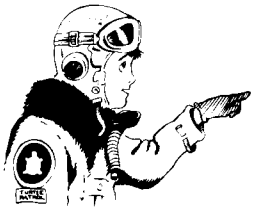
(II)

(III)

The player to take the last match wins.




```
Player #1 takes from level - ?3
Player #1 takes how many - ?
```

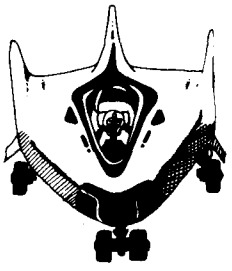


```
100 R:          ***  NIM  ***
110 R:         A GAME OF LOGIC
120 R:
130 R:        COPYRIGHT 1982 BY
140 R:         Dale Disharoon
150 R:
160 C:@B710=0
170 *START
180 C:#Z=0
190 T:[CLEAR]
200 POS:14,4
210 T:<< NIM >>
220 POS:2,8
230 T:How high will your
240 :pyramid be? (3-10)   ?
250 *HIGH
260 POS:25,9
270 A:#K
280 J(#K<3):*HIGH
290 J(#K>10):*HIGH
```

```
300 POS:2,15
310 T: 1 - Player taking last block wi
ns
320 T: 2 - Player taking last block lo
ses
330 POS:11,18
340 T: Enter 1 or 2
350 *LAST
360 POS:27,18
370 T:?\
380 A:#L
390 J(#L<1):*LAST
400 J(#L>2):*LAST
410 T:[CLEAR]
420 C:@B752=1
430 POS:14,1
440 T: << NIM >>
450 C:#A=1
460 C:#B=2
470 C:#C=3
480 C(#K>3):#D=4
490 C(#K>4):#E=5
500 C(#K>5):#F=6
510 C(#K>6):#G=7
520 C(#K>7):#H=8
530 C(#K>8):#I=9
540 C(#K>9):#J=10
550 POS:2,5
560 T:          [VB]  1
570 T:         [2(VB)]  2
580 T:        [3(VB)]  3
590 T(#K>3):       [4(VB)]  4
600 T(#K>4):      [5(VB)]  5
610 T(#K>5):     [6(VB)]  6
620 T(#K>6):    [7(VB)]  7
630 T(#K>7):   [8(VB)]  8
640 T(#K>8): [9(VB)]  9
650 T(#K>9):[10(VB)]  10
660 POS:23,6
670 T:* Last block *
680 POS:28,7
690 T(#L=1):Wins
700 T(#L=2):Loses
710 C:#P=2
720 *GAME
730 C:#P=3-#P
740 *TAKELEVEL
750 POS:2,17
760 T:[DEL LINE]Player ##P takes from
level - ?\
770 A:#M
780 J(#M<1):*NOLEVEL
790 J(#M>#K):*NOLEVEL
800 U:*ARRAY1
810 J(#N=0):*LEVELGONE
820 *HOWMANY
830 POS:2,19
840 T:[DEL LINE]Player ##P takes how m
any - ?\
850 A:#Q
860 J(#Q<1):*SORRY
```

```
870 U:*ARRAY1
880 J(#Q>#N):*TOOMANY
890 U:*ARRAY2
900 U:*ARRAY1
910 POS:(12-#M)+(2*#N),4+#M
920 C:#R=0
930 *LOOP
940 C:#R=#R+1
950 T:  \
960 U:*SOUND1
970 J(#R<>#Q):*LOOP
980 POS:2,17
990 T:[3(DEL LINE)]
1000 U:*CHECK
1010 J(#Z=1):*START
1020 J:*GAME
1030 *NOLEVEL
1040 T:[UP][DEL LINE]Sorry, there is n
o level #M
1050 PA:180
1060 J:*TAKELEVEL
1070 *LEVELGONE
1080 T:[UP][DEL LINE]Level #M is gone
1090 PA:180
1100 J:*TAKELEVEL
1110 *SORRY
1120 T:[UP][DEL LINE]I can't let you d
o that
1130 PA:180
1140 J:*HOWMANY
1150 *TOOMANY
1160 T:[UP][DEL LINE]There aren't that
 many in level #M
1170 PA:180
1180 J:*HOWMANY
1190 *SOUND1
1200 C:@B53761=168
1210 C:#S=28
1220 *SOUND1LOOP
1230 C:#S=#S-4
1240 C:@B53760=#S
1250 J(#S<>0):*SOUND1LOOP
1260 C:@B53761=0
1270 E:
1280 *SOUND2
1290 C:@B53761=168
1300 C:@B53763=168
1310 C:#S=72
1320 *SOUND2LOOP
1330 C:#S=#S-8
1340 C:@B53760=#S
1350 C:@B53762=#S+1
1360 J(#S<>0):*SOUND2LOOP
1370 C:@B53761=0
1380 C:@B53763=0
1390 E:
1400 *ARRAY1
1410 C(#M=1):#N=#A
1420 C(#M=2):#N=#B
```

```
1430 C(#M=3):#N=#C
1440 C(#M=4):#N=#D
1450 C(#M=5):#N=#E
1460 C(#M=6):#N=#F
1470 C(#M=7):#N=#G
1480 C(#M=8):#N=#H
1490 C(#M=9):#N=#I
1500 C(#M=10):#N=#J
1510 E:
1520 *ARRAY2
1530 C(#M=1):#A=#A-#Q
1540 C(#M=2):#B=#B-#Q
1550 C(#M=3):#C=#C-#Q
1560 C(#M=4):#D=#D-#Q
1570 C(#M=5):#E=#E-#Q
1580 C(#M=6):#F=#F-#Q
1590 C(#M=7):#G=#G-#Q
1600 C(#M=8):#H=#H-#Q
1610 C(#M=9):#I=#I-#Q
1620 C(#M=10):#J=#J-#Q
1630 E:
1640 *CHECK
1650 E(#A>0):
1660 E(#B>0):
1670 E(#C>0):
1680 E(#D>0):
1690 E(#E>0):
1700 E(#F>0):
1710 E(#G>0):
1720 E(#H>0):
1730 E(#I>0):
1740 E(#J>0):
1750 C:#Z=1
1760 POS:5,17
1770 U(#L=1):*WINNER
1780 U(#L=2):*LOSER
1790 POS:14,22
1800 T:Press RETURN
1810 C:@B764=255
1820 *RETURN
1830 J(@B764=255):*RETURN
1840 E:
1850 *WINNER
1860 T:   ***  Player ##P wins!  ***
1870 U:*SOUND2
1880 U:*SOUND2
1890 U:*SOUND2
1900 U:*SOUND2
1910 U:*SOUND2
1920 E:
1930 *LOSER
1940 C:@B53761=74
1950 C:@B53760=5
1960 T:   ***  Player ##P loses  ***
1970 PA:120
1980 C:@B53761=0
1990 E:
```



Dodgem

Dodgem is a complex game played on a simple board. We first discovered the game as a 3 × 3 board version in Martin Gardner's "Mathematical Games" column in *Scientific American*. According to Gardner, the game was developed in 1972 by Colin Vout, a mathematics student at the University of Cambridge. We made a 4 × 4 version for ATARI PILOT. Here are the instructions.

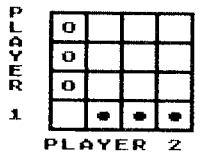
Dodgem is a two-player game. Each player has three pieces. The object of the game is to be the first player to get all one's pieces to the opposite side of the board.

The game is an essentially peaceful one: there is no capturing or jumping of your pieces. Rather, each piece must work its way across the board by moving only where there is an open space.

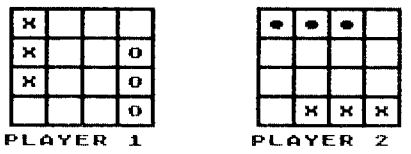
Rules of movement are that players may move their pieces only forward or sideways, not backward. Again, movement can only occur to an empty space.

Any move that one player makes must leave the opponent a valid move—"boxing up" is not allowed.

Here is the board as the game first begins:



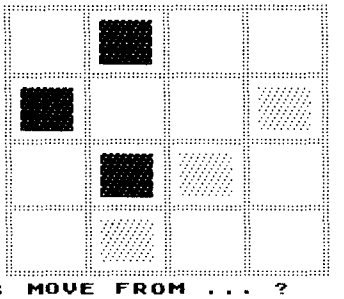
Here are Players 1 and 2 shown in winning positions:



Note the vertical/horizontal change in positions of the respective pieces.

You will need two joysticks to play. To make your move, use your joystick to position the blinking dot in the square containing the piece you want to move. Press the fire button to indicate to the computer which piece to move.

Then position the dot in the square where you want to move to, and press the fire button. The computer automatically checks for a valid move. If you try to make an illegal move, the computer will tell you to try again.



```
PLAYER 1: MOVE FROM ... ?
```

```
2 R:            *** DODGEM ***
4 R:          COPYRIGHT (C) 1982
6 R:             BY ANDY HALL
100 GR:QUIT
110 C:@14=@560-1
120 C:@741=@14-1
130 U:*TITLE
140 U:*INSTRUCT
150 *PLAY
160 C:#H=2
170 C:#I=22
180 C:#W=0
190 U:*BOARD
200 U:*INITIALIZE
210 C:#P=?\2+1
220 *LOOP U:*GETMOVE
230 J(#W>0):*WINNER
240 C:#P=3-#P
250 J:*LOOP
260 *WINNER T:[CLEAR] PLAYER ##W WINS!
270 T:
280 T:ANOTHER GAME? \
290 A:$Y
300 M:Y
310 JY:*PLAY
320 E:
330 *BOARD GR:CLEAR;PENRED;GOTO-37,43;
TURNTO0;4(TURN90;DRAW73)
340 GR:GOTO-36,42;4(TURN90;DRAW71)
350 GR:GOTO-19,43;TURNTO180;3(DRAW72;G
OTO%X+1,43;DRAW72;GOTO%X+17,43)
360 GR:GOTO-36,25;TURNTO90;3(DRAW72;GO
TO-36,%Y-1;DRAW72;GOTO-36,%Y-17)
370 E:
380 *INITIALIZE C:#C=0
390 C:#X=-33
400 C:#Y=39
```

```
410 C:#P=1
420 *ILOOP C:#C=#C+1
430 U:*DRAWP
440 C:#Y=#Y-18
450 J(#C<3):*ILOOP
460 C:#P=2
470 C:#X=-15
480 C:#Y=-15
490 C:#C=1
500 *ILOOP2 C:#C=#C+1
510 U:*DRAWP
520 C:#X=#X+18
530 J(#C<4):*ILOOP2
540 R:USES MEMORY JUST ABOVE PROGRAM F
OR SCRATCH
550 C:#A=1
560 *MLOOP C:#C=(#A-1)\4*18-33
570 C:#D=39-(((#A-1)/4)*18)
580 U:*MEMLOC
590 R:@176 CONTAINS THE ADDRESS OF THE
 LOWEST FREE MEMORY (JUST ABOVE THE PR
OGRAM)
600 C:#L=@176+#A
610 C:@B#L=#E
620 C:#A=#A+1
630 J(#A<17):*MLOOP
640 E:
650 *DRAWP C:@B1363=#P+1
660 GR:GOTO#X,#Y;TURNTO90;3(DRAW11;TUR
N90);FILL11
670 E:
680 *ERASEP GR:PENERASE;GOTO#J,#K;TURN
TO90;12(DRAW11;GOTO#J,%Y-1)
690 E:
700 *GETMOVE R:MOVE FROM #J,#K; TO #X,
#Y
710 T:[CLEAR]PLAYER #P: MOVE FROM ...
?  \
720 U:*JOYSTICK
730 C:#J=#H+1
740 C:#K=#I-1
750 *GLOOP J((#P=1)*(%T8=1)):*GLOOP
760 J((#P=2)*(%T9=1)):*GLOOP
770 T:  TO ... ?
780 U:*JOYSTICK
790 C:#X=#H+1
800 C:#Y=#I-1
810 U:*CHECK
820 SO(#B):1
830 T(#B):   INVALID MOVE
840 J(#B):*P
850 U:*MOVABLE
860 T(#B=2):  MUST LEAVE OPPONENT A VAL
ID MOVE.
870 *P PA(#B):30
880 SO(#B):0
890 PA(#B):90
900 J(#B):*GETMOVE
910 U:*ERASEP
920 U:*DRAWP
930 U:*WINCHECK
```

```
940 E:
950 *JOYSTICK C(#P=1):#S=%J0
960 C(#P=2):#S=%J1
970 C(#P=1):#T=%T8
980 C(#P=2):#T=%T9
990 GR:PENERASE;GOTO#H,#I
1000 E(#T=1):
1010 C((#S\16-7>0)*(#H<20)):#H=#H+18
1020 C((#S\8-3>0)*(#H>-20)):#H=#H-18
1030 C((#S\4-1>0)*(#I>-10)):#I=#I-18
1040 C((#S\2>0)*(#I<25)):#I=#I+18
1050 GR:PENRED;GOTO#H,#I
1060 J:*JOYSTICK
1070 E:
1080 *CHECK C:#B=0
1090 C:#C=#J
1100 C:#D=#K
1110 U:*MEMLOC
1120 C((#P=1)*(#E<>170)):#B=1
1130 C((#P=2)*(#E<>255)):#B=1
1140 C:#C=#X
1150 C:#D=#Y
1160 U:*MEMLOC
1170 C(#E<>0):#B=1
1180 C((#P=1)*((#X<#J)+(#X-#J>18)+(#Y-
#K>18)+(#K-#Y>18))):#B=1
1190 C((#P=2)*((#Y<#K)+(#Y-#K>18)+(#X-
#J>18)+(#J-#X>18))):#B=1
1200 C((#X<>#J)+(#Y<>#K)-1):#B=1
1210 E:
1220 *MEMLOC C:#L=@88+((52-#D)*40)+((#
C+79)/4)+2
1230 C:#E=@B#L
1240 E:
1250 *WINCHECK C:#C=21
1260 C:#D=21
1270 U:*MEMLOC
1280 C:#M=#L+720
1290 C:#N=#M+720
1300 C((@B#L=170)*(@B#M=170)*(@B#N=170
)):#W=1
1310 C:#C=-33
1320 C:#D=39
1330 U:*MEMLOC
1340 C:#M=#L+4
1350 C:#N=#L+9
1360 C((@B#L=255)*(@B#M=255)*(@B#N=255
)):#W=2
1370 E:
1380 *MOVABLE R:MAKE SURE PLAYER LEAVE
S VALID MOVE FOR OPPONENT
1390 R:1. MAKE PROPOSED CHANGES AT SCR
ATCH AREA.
1400 R:2. LOOK AT ONE OF OPPONENT'S PI
ECES: SEE IF MOVABLE.
1410 R:3. IF SO, RETURN. IF NOT, #B=2,
RESTORE SCRATCH AREA.
1420 C:#L=(39-#K)/18*4+((#J+33)/18)+@1
76+1
1430 C:@B#L=0
1440 C:#L=(39-#Y)/18*4+((#X+33)/18)+@1
```

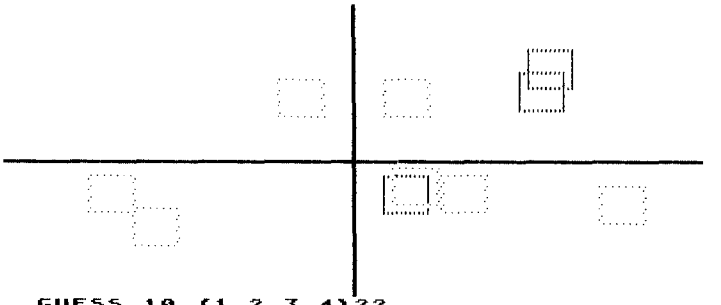
```
76+1
1450 C(#P=2):#E=255
1460 C(#P=1):#E=170
1470 C:@B#L=#E
1480 C:#E=425-#E
1490 C:#A=@176
1500 *MOVLOOP C:#A=#A+1
1510 C:#F=@B#A
1520 J(#A=(@176+17)):*BAD
1530 J(#E<>#F):*MOVLOOP
1540 C:#G=(#A-@176-1)\4
1550 J(#G=3):*SIDWAY
1560 C:#Z=#A+1
1570 E(@B#Z=0):
1580 *SIDWAY J(#A<(@176+5)):*CHKSPCL
1590 C:#Z=#A-4
1600 E(@B#Z=0):
1610 *CHKSPCL C(#P=2):#Z=#A+4
1620 C(#P=1):#Z=#A-1
1630 J(#Z>(@176+16)):*MOVLOOP
1640 J(#P=1):*P1
1650 E(@B#Z=0):
1660 J:*MOVLOOP
1670 *P1 J(#G=0):*MOVLOOP
1680 E(@B#Z=0):
1690 J:*MOVLOOP
1700 *BAD C:#B=2
1710 C:#L=(39-#Y)/18*4+((#X+33)/18)+@1
76+1
1720 C:@B#L=0
1730 C:#L=(39-#K)/18*4+((#J+33)/18)+@1
76+1
1740 C(#P=1):#E=170
1750 C(#P=2):#E=255
1760 C:@B#L=#E
1770 E:
1780 *INSTRUCT
1790 T:
1800 T:WOULD YOU LIKE INSTRUCTIONS? \
1810 A:
1820 M:Y
1830 EN:
1840 T:[CLEAR]
1850 T:  THE GAME OF DODGEM IS A TWO-P
LAYER GAME. EACH PLAYER HAS THREE PIEC
ES. \
1860 T:THE OBJECT OF THE GAME IS TO BE
 THE FIRST PLAYER TO GET ALL HIS PIECE
S TO THE OPPOSITE SIDE OF THE BOARD.
1870 T:
1880 T:  THE GAME IS AN ESSENTIALLY PE
ACEFUL ONE: THERE IS NO CAPTURING OR J
UMPING OF YOUR MEN. \
1890 T:RATHER, EACH PIECE MUST WORK IT
S WAY ACROSS THE BOARD BY MOVING ONLY
WHERE THERE IS AN OPEN SPACE.
1900 T:
1910 T:  RULES OF MOVEMENT: PLAYERS MA
Y MOVE THEIR MEN ONLY FORWARD OR SIDEW
AYS: NOT BACKWARDS. \
1920 T:AGAIN, MOVEMENT CAN ONLY OCCUR
TO AN EMPTY SPACE. *** ANY MOVE ONE PL
AYER MAKES MUST \
```

```
1930 T:LEAVE OPPONENT A VALID MOVE --
"BOXING UP" IS NOT ALLOWED. ***
1940 T:
1950 T:       HIT ANY KEY TO CONTINUE.
1960 U:*WAITKEY
1970 T:[CLEAR]
1980 T:
1990 T:  HERE IS THE BOARD AS THE GAME
 FIRST BEGINS:
2000 T:
2010 T:              P [Q][3(RW)][R][E]
2020 T:              L |O| | | |
2030 T:              A [A][3(RS)][R][D]
2040 T:              Y |O| | | |
2050 T:              E [A][3(RS)][R][D]
2060 T:              R |O| | | |
2070 T:                [A][3(RS)][R][D]
2080 T:              1 | ||||
2090 T:                [Z][3(RX)][R][C]
2100 T:                PLAYER 2
2110 T:
2120 T:
2130 T:
2140 T:       HIT ANY KEY TO CONTINUE.
2150 U:*WAITKEY
2160 T:[CLEAR]
2170 T:  HERE ARE PLAYERS 1 AND 2 SHOW
N IN WINNING POSITIONS:
2180 T:
2190 T:       [Q][3(RW)][R][E]      [Q][
3(RW)][R][E]
2200 T:       |X| | | |      ||||
2210 T:       [A][3(RS)][R][D]      [A][
3(RS)][R][D]
2220 T:       |X| | |O|      | | | | |
2230 T:       [A][3(RS)][R][D]      [A][
3(RS)][R][D]
2240 T:       |X| | |O|      | | | | |
2250 T:       [A][3(RS)][R][D]      [A][
3(RS)][R][D]
2260 T:       | | | |O|      | |X|X|X|
2270 T:       [Z][3(RX)][R][C]      [Z][
3(RX)][R][C]
2280 T:       PLAYER 1       PLAYER 2
2290 T:
2300 T:NOTE THE VERTICAL/HORIZONTAL CH
ANGE IN POSITIONS OF THE RESPECTIVE PI
ECES.
2310 T:
2320 T:
2330 T:
2340 T:       HIT ANY KEY TO CONTINUE.
2350 U:*WAITKEY
2360 T:[CLEAR]
2370 T:  YOU WILL NEED TWO JOYSTICKS T
O PLAY. TO MAKE YOUR MOVE, USE YOUR JO
YSTICK TO POSITION THE BLINKING \
2380 T:DOT IN THE SQUARE CONTAINING TH
E PIECE YOU WANT TO MOVE. PRESS THE FI
RE BUTTON TO INDICATE \
2390 T:TO THE COMPUTER WHICH PIECE TO
MOVE.
2400 T:
```

```
2410 T:  THEN POSITION THE DOT IN THE
SQUARE WHERE YOU WANT TO MOVE TO, AND
PRESS THE FIRE BUTTON. THE \
2420 T:COMPUTER AUTOMATICALLY CHECKS F
OR A VALID MOVE. IF YOU TRY TO MAKE AN
 ILLEGAL MOVE, THE COMPUTER \
2430 T:WILL TELL YOU TO TRY AGAIN.
2440 T:
2450 T:              HAVE FUN!
2460 T:
2470 T:        HIT ANY KEY TO BEGIN.
2480 *WAITKEY C:@B764=255
2490 *WAITLOOP J(@B764=255):*WAITLOOP
2500 C:@B764=255
2510 E:
2520 *TITLE C:@B559=0
2530 C:#P=@560+3
2540 C:@B#P=71
2550 C:#P=#P+3
2560 C:#C=7
2570 *TLOOP C:@B#P=7
2580 C:#P=#P+1
2590 C:#C=#C-1
2600 J(#C>0):*TLOOP
2610 C:@B#P=6
2620 C:#P=#P+1
2630 C:@B#P=6
2640 C:#P=#P+1
2650 C:@B#P=65
2660 C:#P=#P+1
2670 C:@B712=100
2680 C:@B710=12
2690 C:@B709=12
2700 C:@B708=200
2710 C:@#P=@560
2720 C:@B559=34
2730 T:[CLEAR]
2740 T:      dodgem
2750 POS:2,4
2760 T:  BY ANDY HALL
2770 PA:400
2780 GR:QUIT
2790 E:
```

ESP

This is an extrasensory perception (ESP) game. The computer presents a grid with four quadrants. The computer then picks one quadrant in which to place a square. You can't see the choice but must concentrate and try to perceive the computer's choice. You can take either 10 or 30 turns. Chance guessing would make approximately 25 percent of your guesses right. If your guesses are greater than chance, you may have ESP—or be lucky. Try again.



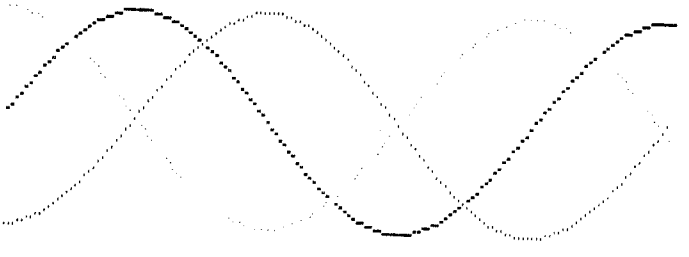
```
GUESS 10 (1,2,3,4)?2
CORRECT!  YOU HAVE 3 OUT OF 10.
```

```
2 R:              *** ESP ***
4 R:    COPYRIGHT (C) DECEMBER, 1982
6 R:              BY TED M. KAHN
8 R: A SIMPLE GAME TO TO TEST ESP PO-
TENTIAL - BASED ON THE WORK OF RUSSELL
TARG
100 *ESP
110 T:THIS GAME WILL ALLOW YOU TO TEST
 YOUR ABILITIES AT PRE-COGNITION.
120 :
130 :A SQUARE WILL BE DRAWN IN ONE OF
FOUR QUADRANTS.  YOU WILL TRY TO GUESS
 WHERE IT WILL APPEAR EACH TIME.
140 :
150 :(PRESS RETURN TO CONTINUE).
160 A:
170 GR:CLEAR
180 T:THE NUMBER OF YOUR GUESS CORRESP
OND TO THESE FOUR QUADRANTS:
190 PA:120
200 U:*AXES
210 GR:GOTO -45,20;DRAW 10
220 GR:GOTO 45,33;TURNTO 90;3(DRAW 7;T
URN 90);TURN -180;DRAW 5;TURN -90;DRAW
 7
230 GR:GOTO -45,-10;TURNTO 90;3(DRAW 7
;TURN 90);TURNTO 90;3(DRAW 7;TURN 90)
240 GR:GOTO 45,-10;TURNTO 180;DRAW 10;
TURN -90;DRAW 10;GOTO 52,-10;TURNTO 18
0;DRAW 15
250 T:
260 :YOUR E.S.P. POTENTIAL IS HOW WELL
 YOU DO ABOVE CHANCE.
270 PA:120
280 *LOOP
290 T:HOW MANY TRIALS DO YOU WANT (10,
20, OR 30)?\
300 A:#T
310 J(#T\10<>0):*LOOP
320 J(#T=0):*LOOP
330 C:#T=#T\30
340 U:*AXES
350 *LOOP
360 C:#R=0
370 C:#I=0
```

```
380 *NEXT C:#I=#I+1
390 T:[CLEAR]GUESS #I (1,2,3,4)?\
400 C:#Q=?\4+1
410 A:#G
420 U:*BOX
430 C(#G=#Q):#R=#R+1
440 T(#G=#Q):CORRECT!  YOU HAVE #R OUT
 OF #I.
450 T(#G<>#Q):NO, IT WAS #Q.  YOU HAVE
 #R OUT OF #I.
460 PA:180
470 J(#I<#T):*NEXT
480 C:#C=#T/4
490 C(#T\4):#C=#C+1
500 GR:QUIT
510 T:YOUR FINAL SCORE WAS #R OUT OF #
T.
520 :CHANCE GUESSING WOULD HAVE BEEN A
BOUT #C.
530 :
540 T(#R>#C):YOU DID BETTER THAN CHANC
E.
550 T(#R=#T):YOU REALLY HAVE E.S.P.!
560 T:DO YOU WANT TO PLAY AGAIN?
570 A:
580 M:Y,SURE,OK,O.K.
590 JY:*LOOP
600 E:
610 *AXES
620 GR:CLEAR; GOTO -79,5;DRAWTO 79,5
630 GR:GOTO 0,47;DRAWTO 0,-32
640 E:
650 *BOX
660 GR(#G=#Q):PEN BLUE
670 GR(#G<>#Q):PEN RED
680 GR(#Q=1):GOTO (?\68)-79,(?\31)+16
690 GR(#Q=2):GOTO (?\68)+1,(?\31)+16
700 GR(#Q=3):GOTO (?\68)-79,(?\25)-21
710 GR(#Q=4):GOTO (?\68)+1,(?\25)-21
720 GR:TURNTO 90;4(DRAW 10;TURN 90)
730 E:
```

Trig Functions

This simple program displays how sine and cosine functions can be represented in PILOT despite the fact that it deals only with integers.

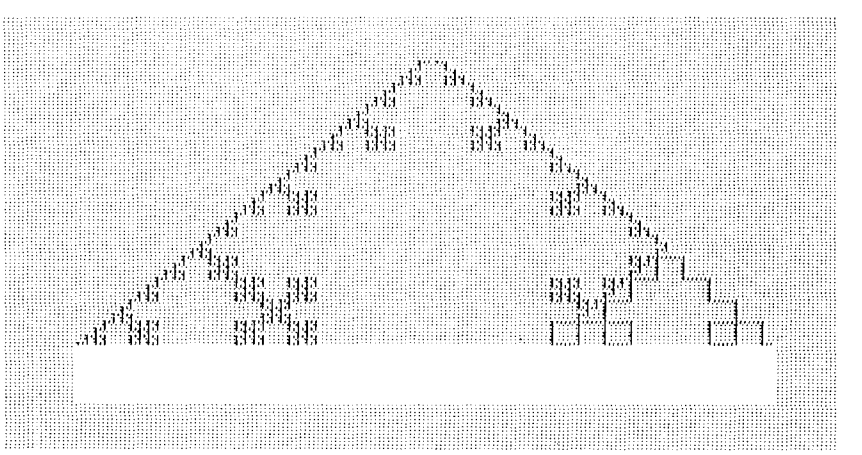


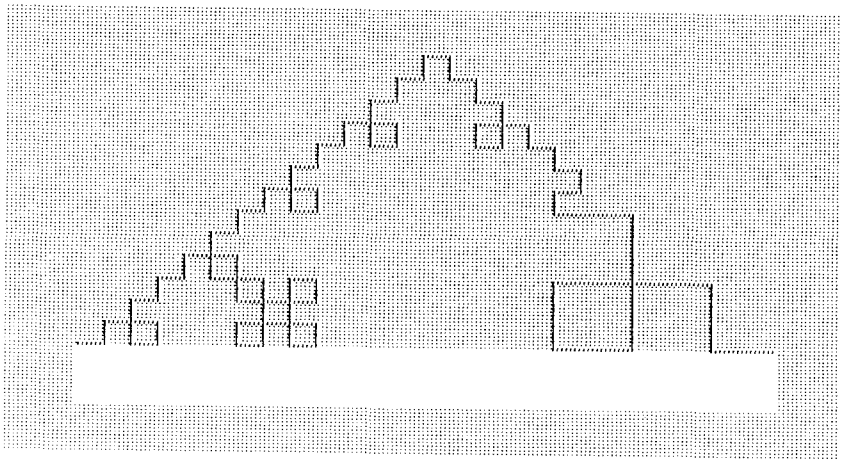
```
2 R:           *** GSINE ***
4 R:        BY HARRY B. STEWART
6 R: DEMONSTRATES PILOT PLOT OF SINE/C
OSINE CURVES
100 GR:TURNTO -90
110 *START C:#X=-80
120 GR:CLEAR
130 *LOOP
140 GR:PEN UP;GOTO 0,10; GO 30; PEN YE
LLOW; GOTO #X,%Y
150 GR:PEN UP; GOTO 0,10; TURN 90; GO
30;TURN -90; PEN RED; GOTO #X,%Y
160 GR:PEN BLUE; GOTO #X,20-%Y
170 GR:TURN 3
180 C(#X<80):#X=#X+1
190 J(#X<80):*LOOP
200 J:*START
```

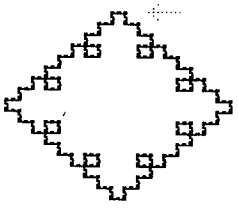
Fractals

To understand fractals, think of two mirrors facing each other with a person standing between the mirrors. If you stand a bit to the side and look in one mirror, you'll see an infinite regression of mirrors reflecting the person and mirroring themselves reflecting the person. A fractal is somewhat like this. A *fractal* is a curve, each part of which approximates the whole structure. At the bottom of this page and the top of the next, you will find two pictures of a growing fractal, one beginning to turn in on itself to mirror its own shape.

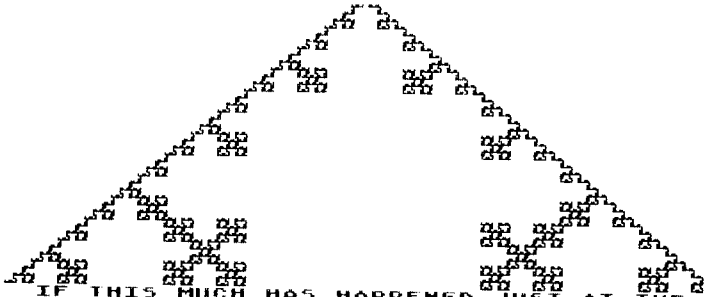
Here are two pictures of more detailed fractals and a program to generate them. If, after you play with this program, you are more interested in fractals, read David Thornberg's article on fractals in *Computer,* July 1982, and write to him for additional resources.







NOW WE'RE GOING TO TURN THE COMPUTER
INTO A 50X MICROSCOPE AND LOOK AT
WHAT'S HAPPENING AT THE TOP OF THE
VERY TOP BOX. HIT ANY KEY TO GO.



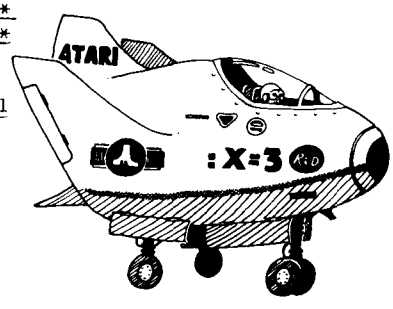
IF THIS MUCH HAS HAPPENED JUST AT THE
TOP, IMAGINE WHAT THE REST OF THE
SQUARE MUST LOOK LIKE! HIT ANY KEY TO
SEE.

```
2 R:          *** FRACTALS ***
4 R:        COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
100 U:*TITLE
110 U:*INSTRUCT
120 C:#O=0
130 C:#N=4
140 C:#S=0
```

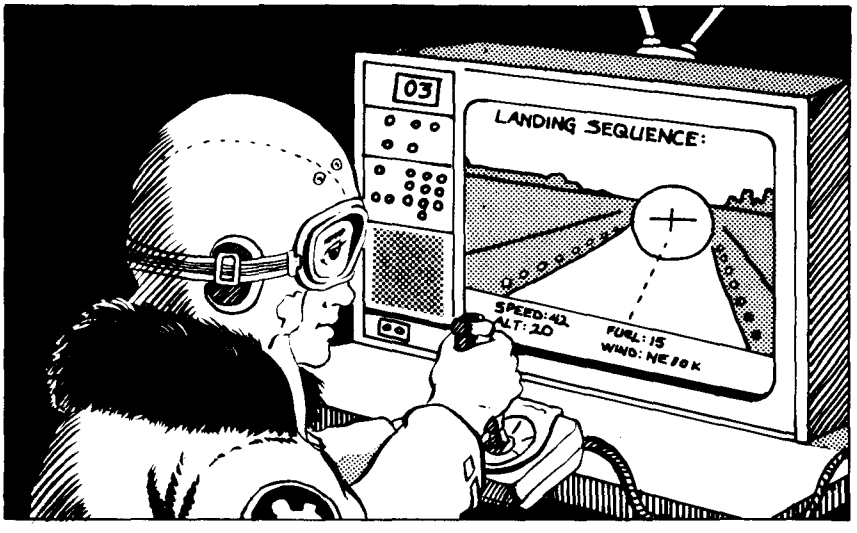
```
150 C:#L=9
160 C:#C=0
170 GR:CLEAR
180 C(#K=0):@B710=0
190 GR:PENYELLOW;GOTO-13,21;TURNTO90;4
(DRAW27;TURN90)
200 GR(#K=1):PENUP;GOTO-12,21;TURNTO18
0;PENBLUE;FILL26;PENYELLOW;GOTO-13,21;
TURNTO90
210 PA:100
220 U:*LOOP
230 T:[CLEAR]NOW WE'RE GOING TO TURN T
HE COMPUTER INTO A 50X MICROSCOPE AND
LOOK AT WHAT'S HAPPENING AT THE TOP\
240 T: OF THE VERY TOP BOX. HIT ANY KE
Y TO GO. \
250 GR:PENRED;GOTO14,33;DRAWTO7,33;DRA
WTO9,35;GOTO7,33;DRAWTO9,31
260 C:@B764=255
270 *MLOOP C:@B708=0
280 PA:15
290 C:@B708=70
300 PA:15
310 J(@B764=255):*MLOOP
320 C:@B764=255
330 T:[CLEAR]
340 C:#T=@560+5
350 C:#W=@B#T
360 C:#V=0
370 *LLOOP C:#U=#V/2+16
380 C:#X=#U+3
390 C(#X>31):#X=31
400 C:#W=313-#W
410 C:@B#T=#W
420 GR(#V=16):CLEAR
430 SO:#U,#X
440 PA:(32-#V)/3
450 C:#V=#V+1
460 J(#V<32):*LLOOP
470 SO:0,0
480 C:@B#T=#W
490 GR:CLEAR
500 C:#L=54
510 C:#D=1
520 C:#C=0
530 C(#K=0):@B710=0
540 GR:PENYELLOW;GOTO-79,-31;TURNTO90;
DRAWTO79,-31;GOTO-79,-31
550 T:HERE WE ARE AT THE TOP! JUST HOP
E WE DON'T RUN INTO ANY BUGS -- THEY'R
E PRETTY BIG AT THIS MAGNIFICATION.\
560 U:*LOOP
570 T:[CLEAR]IF THIS MUCH HAS HAPPENED
 JUST AT THE TOP, IMAGINE WHAT THE RES
T OF THE SQUARE MUST LOOK LIKE! \
580 T:HIT ANY KEY TO SEE. \
590 U:*WAITKEY
600 T:[CLEAR]
610 *LOOK C:#U=#V
620 C:#X=#U-3
630 C(#X<1):#X=1
```

```
640 SO:#U,#X
650 PA:#V/2
660 C:@B#T=314-@B#T
670 GR(#V=9):CLEAR
680 C:#V=#V-1
690 J(#V>0):*LOOK
700 C(#K=0):@B710=0
710 SO:0,0
720 GR:PENYELLOW;GOTO0,34;DRAWTO26,8;D
RAWTO0,-18;DRAWTO-26,8;DRAWTO0,34
730 GR(#K=1):PENBLUE;GOTO0,33;FILLTO-2
5,8;FILLTO0,-17;PENYELLOW
740 GR:TURNTO45;GOTO-16,1;DRAW3;GOTO14
,1;DRAW3;GOTO-7,-8;DRAW3;GOTO5,-8;DRAW
3
750 GR:GOTO-16,13;DRAW3;GOTO14,13;DRAW
3;GOTO-7,22;DRAW3;GOTO5,22;DRAW3
760 GR:TURNTO-45;GOTO16,1;DRAW3;GOTO-1
4,1;DRAW3;GOTO7,-8;DRAW3;GOTO-5,-8;DRA
W3
770 GR:GOTO16,13;DRAW3;GOTO-14,13;DRAW
3;GOTO7,22;DRAW3;GOTO-5,22;DRAW3
780 GR:TURNTO45;GOTO-23,10;7(G04;GOTO%
X,%Y);GOTO2,-15;7(G04;GOTO%X,%Y)
790 GR:GOTO-2,-15;TURNTO315;7(G04;GOTO
%X,%Y);GOTO23,10;7(G04;GOTO%X,%Y)
800 T:[CLEAR]LOOK WHAT'S HAPPENED SINC
E WE STARTED -- OUR SQUARE HAS "FRACTA
LED" INTO A DIAMOND!
810 PA:1200
820 C:@B710=148
830 E:
840 *LOOP GR:GO1;PENUP;TURN90;GO1
850 J(%Z<>2):*LEFTCHK
860 GR:GO-1
870 C:#C=0
880 J((#O=0)*(#S=#N)):*SCALE
890 J:*LOOP
900 *LEFTCHK GR:TURN180;GO2
910 J(%Z<>2):*STRAIGHT
920 GR:GO-1
930 C:#C=0
940 J((#O=0)*(#S=#N)):*SCALE
950 J:*LOOP
960 *STRAIGHT GR:GO-1;TURN90;PENYELLOW
970 C:#C=#C+1
980 J(#C<#L):*LOOP
990 *SCALE J((#S<#N)*(#O=0)):*CONT
1000 J((%X<80)*(#O=1)):*CONT
1010 C:#N=#N*6
1020 C:#C=0
1030 C:#L=#L/3
1040 PA:100
1050 T((#L=6)*(#O=1)):)
1060 GR(#O=1):GOTO-79,-31
1070 E(#L<2):
1080 J:*LOOP
1090 *CONT
1100 C:#S=#S+1
1110 C:#C=0
1120 U:*BOX
```

```
1130 J:*LOOP
1140 *BOX GR:TURN-90;3(DRAW#L;TURN90);
PENERASE;DRAW#L-1;PENUP
1150 GR(#K=0):TURN180;PENYELLOW;GO#L
1160 E(#K=0):
1170 C:#A=%A/90
1180 C:#X=%X
1190 C:#Y=%Y
1200 GR(#A=0):GOTO%X,%Y+1
1210 GR(#A=1):GOTO%X-#L+2,%Y+1
1220 GR(#A=2):GOTO%X-#L+1,%Y+#L-1
1230 GR(#A=3):GOTO%X,%Y+#L
1240 GR:PENBLUE;TURNTO180;FILL#L
1250 GR((#A=0)+(#A=2)):PENYELLOW;GOTO%
X,%Y
1260 GR:PENUP;GOTO#X,#Y;TURNTO#A*90+18
0;GO#L-1;PENYELLOW
1270 E:
1280 *INSTRUCT T:
1290 T:
1300 T:          *** FRACTALS ***
1310 POS:2,5
1320 T:A FRACTAL IS A STRANGE TYPE OF
GEOMETRIC SHAPE. TO MAKE ONE, YOU STAR
T WITH A FAMILIAR SHAPE, SAY \
1330 T:A SQUARE. THEN, ON EACH OF ITS
SIDES YOU DRAW ANOTHER, SMALLER, SQUAR
E, AND THEN \
1340 T:ANOTHER ON EACH OF THE SIDES YO
U'VE JUST CREATED. YOU KEEP ON DOING T
HIS UNTIL YOU CAN'T DRAW ANY \
1350 T:SMALLER. THEN YOU'VE CREATED A
FRACTAL.
1360 T:
1370 T:
1380 T:DO YOU WANT COLOR? \
1390 A:
1400 M:Y
1410 CY:#K=1
1420 E:
1430 *WAITKEY C:@B764=255
1440 *WLOOP J(@B764=255):*WLOOP
1450 C:@B764=255
1460 E:
1470 *TITLE C:@B1373=16
1480 C:@B1374=2
1490 WRITE:S
1500 WRITE:S,     ************
1510 WRITE:S,     * fractals *
1520 WRITE:S,     ************
1530 WRITE:S
1540 WRITE:S
1550 WRITE:S,     by andy hall
1560 PA:400
1570 GR:QUIT
1580 E:
```



5 Adventure Games, Modeling, and Miniworlds



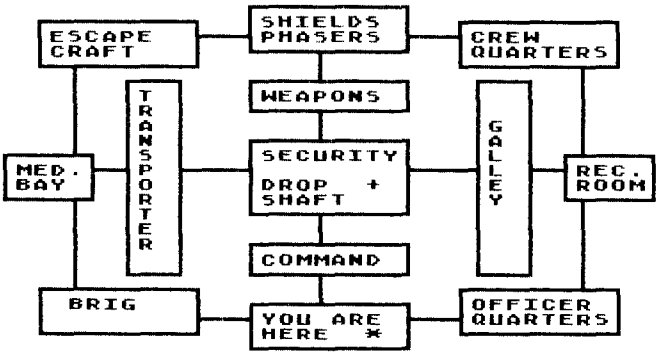
Scout Ship—An Adventure

We began this book with a simple monster adventure. Now we'll share a more complex adventure, one developed by Dale Disharoon and two of his students, Jeremiah Knoche and Miles Pickering, who are in junior and senior high school, respectively. The adventure, which is called Scout Ship, is easy to modify; we welcome you to change it and to develop your own adventures. Here is how you begin the adventure.

```
YOU HAVE BEAMED ABOARD AN ALIEN SPACE
CRAFT.  MOST OF THE ALIEN CREW ARE
NOT ONBOARD YET.  YOUR MISSION IS TO
DESTROY THE SHIP AND TRANSPORT
YOURSELF OUT, ALIVE.

TO DESTROY THE SHIP YOU MUST FIND AN
EXPLOSIVE, ARM IT AND ACTIVATE IT
NEAR THE SHIP'S ENGINES.

TRANSPORT TO SAFETY BY USING THE
COMMAND 'TRANSPORT'.

TO MOVE: TYPE THE 'LETTER' OF AN OPEN
CORRIDOR.  IF YOU NEED HELP DURING
THE GAME, TYPE 'HELP'.

GOOD LUCK...

PRESS START
```

Here are the words that you can use during the game as well as a map of the alien craft.

```
THESE ARE SOME OF THE WORDS I KNOW.

VERBS:                  NOUNS:

TAKE    SEARCH          COIN            WEED
DROP    LAUNCH          CRAFT           BOMB
PRESS   SHOOT           PHASER          GLOBE
SAY     INSERT          CARD            START
READ    LOOK            CARTRIDGE       LASER
EAT     INVENTORY       SOYBURGER       PAN
BREAK   TRANSPORT       SUNGLASSES      TRION
CUT     ACTIVATE        MANUAL          FLUID
FEED                    CHEST           GUARD

Try drawing a simple map as you go.

WHAT NOW?
```



The two screen dumps should give an idea of what the screen might look like during the game.

```
MAIN COMPUTER SECTION:  The huge bulk
of the computer dominates the room.

POSSIBLE EXITS . . .

     CORRIDOR   B
     CORRIDOR   I
     CORRIDOR   P

WHAT NOW?  FIND JOYSTICK

THE JOYSTICK  IS HARD TO FIND.

WHAT NOW?  PRESS START

The computer comes quietly to life.
and says . . .
```

```
ENERGY STORAGE:  A huge pulsating
glass globe showers the room with an
eerie pink glow.  There is a small
cutting laser here.

POSSIBLE EXITS . . .

     CORRIDOR   T

WHAT NOW?  ACTIVATE BOMB

The bomb must be on the floor in
ENGINE ROOM before you detonate it.

WHAT NOW?
```

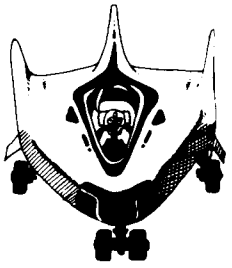
Notice that lines 160 to 260 name, use, or jump to modules. To familiarize yourself with how the program works, look at those modules before trying to copy the program. Also look at line 2360, *COMPUTER. You'll find one of the number-guessing games that we included in the Math chapter embedded in the adventure as a challenge. In PILOT it's easy to turn a whole game into a module and call it up at any time that you want in a longer and more complex program.

Play Scout Ship a bit and then add your own touches.

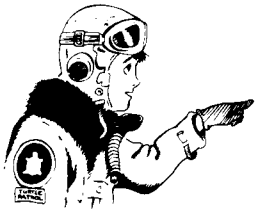
```
100 R: *** SCOUT SHIP ADVENTURE ***
110 R:         COPYRIGHT 1982
120 R:       by  Dale Disharoon
130 R:        Jeremiah Knoche
140 R:        Miles Pickering
150 R:
160 *START
170 U:*INITIALIZE
180 *MAINLOOP
190 U:*ROOMCOLOR
200 U:*DESCRIPTION
210 U:*OBJECTS
220 U:*CORRIDORS
230 U:*WHATNOW
240 J:*MAINLOOP
250 *MOD1 C:#T=1
260 *LOOP3 A:$T=Z#T
270 A:=$RIGHT
280 M:$$T
290 EY:
300 C:#T=#T+1
310 J(#T<13):*LOOP3
320 E:
330 *MOD2 A:$J=Y#T
340 A:#J=$$J
350 E:
360 *WHATNOW T:
370 :WHAT NOW?  \
380 A:$COMMAND
390 T:
400 M:CRY,SLEEP,SCREAM,SHOUT,DIE,LAUGH
410 TY:FROM DOWN THE CORRIDOR A VOICE
ECHOES 'SEARCH... SEARCH... SEARCH....
.'
420 JY:*WHATNOW
430 U(#B):*TIMER
440 MS:[RIGHT] [
450 A:=$RIGHT
460 M:A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,
Q,R,S,T,U,V,W,X,Y,Z
470 A:=$LEFT
480 JN:*ONEWORD
490 M:GET,TAKE,GRAB,DROP,LEAVE,SAY,REA
D,EAT,SEARCH,INSERT,PUSH,PRESS,LAUNCH,
ENTER,SHOOT,FIRE
500 JM:*GET,*GET,*GET,*DROP,*DROP,*SAY
,*READ,*EAT,*SEARCH,*INSERT,*PRESS,*PR
ESS,*LAUNCH,*LAUNCH,*SHOOT,*SHOOT
510 M:CUT,OPEN,BUST,DESTROY,BREAK,FEED
,WATER,FERTILIZE,ACTIVATE,SET,DETONATE
,EXPLODE
520 JM:*CUT,*CUT,*CUT,*CUT,*CUT,*FEED,
*FEED,*FEED,*BOMB,*BOMB,*BOMB,*BOMB
530 *STOCK A:=$RIGHT
540 M:S [
550 TY:THE $RIGHT ARE HARD TO $LEFT.
560 TN:THE $RIGHT IS HARD TO $LEFT.
570 J:*WHATNOW
```

```
580 *AINTGOT T:YOU DON'T HAVE THE $RIG
HT
590 J:*WHATNOW
600 *YOUCANT T:YOU CAN'T $LEFT THE $RI
GHT
610 J:*WHATNOW
620 *ONEWORD M: A , B , C , D , E , F
, G , H , I , J , K , L , M , N , O ,
P , Q , R , S , T , U ,
630 JY:*MOVE
640 M:LOOK
650 EY:
660 M:INVENTORY,TRANSPORT,HELP
670 JM:*INVENTORY,*TRANSPORT,*HELP
680 E:
690 *GET
700 A:=$RIGHT
710 M:FOOD
720 JY(#R=5):*FOOD
730 U:*MOD1
740 J(#T=13):*STOCK
750 U:*MOD2
760 J(#J=-1):*HAVEIT
770 J(#T=9):*FLUID
780 J(#J<>#R):*NOTHERE
790 A:$T=Y#T
800 J(#C=4):*TOOMUCH
810 J(#T=11):*TRION
820 C:#C=#C+1
830 C:$$T=-1
840 T:YOU NOW HAVE THE $RIGHT
850 PA:100
860 E:
870 *NOTHERE T:THE $RIGHT ISN'T HERE.
880 J:*WHATNOW
890 *TOOMUCH T:YOU ARE CARRYING TOO MU
CH.  IF YOU WANT THE $RIGHT YOU'LL HAV
E TO DROP SOMETHING.
900 J:*WHATNOW
910 *HAVEIT T:YOU ALREADY HAVE THE $RI
GHT
920 J:*WHATNOW
930 *DROP
940 U:*MOD1
950 J(#T=13):*YOUCANT
960 U:*MOD2
970 J(#J<>-1):*AINTGOT
980 J((#R=17)*(#T=3)):*BRIGHT
990 C:#C=#C-1
1000 A:$T=Y#T
1010 C:$$T=#R
1020 E:
1030 *BRIGHT T:The light is too bright
.  You had better keep the shades.
1040 J:*WHATNOW
1050 *TRION A:=$Y12
1060 M:-1
1070 TN:You need the BOMB casing to ho
ld the crystals.
1080 JN:*WHATNOW
```

```
1090 T:You gather the scattered crysta
ls and pack them into the BOMB casing.
1100 C:$Y11=0
1110 J:*WHATNOW
1120 *FOOD T:THE FOOD MACHINE REQUIRES
 MONEY.
1130 J:*WHATNOW
1140 *FLUID A:=$Y10
1150 M:-1
1160 TN:You don't have anything to car
ry it in.
1170 JN:*WHATNOW
1180 C:#P=1
1190 T:Your pan is full to the brim.
1200 J:*WHATNOW
1210 *FEED A:#T=$Y10
1220 J((#T<>-1)+(#P=0)+(#R<>16)):*STOC
K
1230 A:=$RIGHT
1240 M:WEED,PLANT
1250 JN:*STOCK
1260 T:The weed hungrily gobbles up th
e brown fluid and your bed pan.  It th
en looks up and mutters, "SAY BURPEE".
1270 C:$Y10=0
1280 C:#C=#C-1
1290 C:#N=1
1300 J:*WHATNOW
1310 *BOMB A:=$RIGHT
1320 M:BOMB
1330 JN:*STOCK
1340 J(#B):*STOCK
1350 A:#T=$Y12
1360 T(#T<>15):The bomb must be on the
 floor in ENGINE ROOM before you deton
ate it.
1370 T(#F=0):Without the TRION crystal
s the bomb is useless.
1380 J((#F=0)+(#T<>15)):*WHATNOW
1390 T:CONGRATULATION SCOUT!!  The bo
mb timer is set.  You have only a few
minutes to TRANSPORT out!
1400 C:#B=1
1410 C:@19=0
1420 J:*WHATNOW
1430 *TIMER E(@B19<30):
1440 T:[CLEAR]
1450 C:@709=10
1460 POS:2,6
1470 T:BOOM!  You didn't make it ...
1480 :
1490 :RANK: DEAD ADVENTURER
1500 J:*END
1510 *SAY A:=$RIGHT
1520 M:BURPEE
1530 TN:Your voice sounds lonely in th
is place.
1540 JN:*WHATNOW
1550 T(#R<>14):Nothing seems to happen
 in this room.
```



```
1560 J(#R<>14):*WHATNOW
1570 T:You suddenly hear a humming sou
nd coming from Corridor R.
1580 C:#M=1
1590 J:*WHATNOW
1600 *READ A:=$RIGHT
1610 M:MANUAL
1620 JY:*MANUAL
1630 T:YOU CAN'T READ THE $RIGHT
1640 J:*WHATNOW
1650 *MANUAL A:$J=Y4
1660 A:#J=$$J
1670 T(#J=-1):The only part you can un
derstand are the words:  * PRESS START
 *
1680 T(#J<>-1):You don't have the manu
al.
1690 J:*WHATNOW
1700 *EAT A:=$RIGHT
1710 T:EATING THE $RIGHT MIGHT MAKE YO
U SICK!
1720 J:*WHATNOW
1730 *CUT A:=$RIGHT
1740 M:GLOBE
1750 JY((#R=17)*(#F=0)):*CUT1
1760 J:*STOCK
1770 *CUT1 A:=$Y2
1780 M:-1
1790 TN:There's only one thing that ca
n cut the globe open and you don't hav
e it.
1800 JN:*WHATNOW
1810 C:$Y11=17
1820 T:Light bursts from your laser.
The globe shatters and precious crysta
ls of TRION spill out onto the floor.
1830 C:#F=1
1840 J:*WHATNOW
1850 *SEARCH A:=$RIGHT
1860 M:GUARD
1870 JY((#G=0)*(#R=3)):*GUARD
1880 M:CHEST
1890 JY((#H=0)*(#R=13)):*CHEST
1900 T:NO LUCK.
1910 J:*WHATNOW
1920 *GUARD C:#G=1
1930 T:A PILOT Cartridge falls from th
e guard's pocket.
1940 A:$T=Y1
1950 C:$$T=3
1960 J:*WHATNOW
1970 *CHEST C:#H=1
1980 T:You see a Security Card which h
ad been hidden in a pair of pajamas.
1990 A:$T=Y8
2000 C:$$T=13
2010 J:*WHATNOW
2020 *INSERT A:=$RIGHT
2030 M:COIN
2040 JY((#R=5)*(#E=0)):*SOY
2050 M:CARTRIDGE
```



```
2060 JY((#R=2)*(#I=0)):*CART
2070 T:What?
2080 J:*WHATNOW
2090 *SOY A:=$Y5
2100 M:-1
2110 TN:What coin?
2120 JN:*WHATNOW
2130 T:The machine rattles, a small do
or pops open and a soyburger drops out
 onto the floor.
2140 C:$Y7=5
2150 C:#E=1
2160 C:$Y5=0
2170 C:#C=#C-1
2180 J:*WHATNOW
2190 *CART A:=$Y1
2200 M:-1
2210 TN:You don't have the PILOT cartr
idge.
2220 JN:*WHATNOW
2230 C:#I=1
2240 C:#C=#C-1
2250 C:$Y1=0
2260 E:
2270 *ENGINE T:A strange force blocks
your way.
2280 C:#R=#T
2290 J:*WHATNOW
2300 *PRESS A:=$RIGHT
2310 M:START
2320 TY(#R<>6):The START button isn't
here.
2330 TN:The $RIGHT appears to be stuck
.
2340 JY(#R=6):*COMPUTER
2350 J:*WHATNOW
2360 *COMPUTER T:The computer comes qu
ietly to life.
2370 PA:100
2380 T:and says . . .
2390 PA:100
2400 T:I'M THINKING OF A NUMBER BETWEE
N 1 AND 100.
2410 *GAME C:#J=?\100+1
2420 *LOOP5 T:WHAT'S YOUR GUESS?
2430 A:#T
2440 T(#T<#J):SORRY EARTHLING, TOO LOW
.
2450 T(#T>#J):NOPE, TOO HIGH.
2460 J(#T<>#J):*LOOP5
2470 *NUM C:#T=?\100+1
2480 J(#T=#J):*NUM
2490 T:SORRY FRIEND.  MY NUMBER WAS #T
.
2500 :
2510 T:SAY, WOULD YOU LIKE TO SEE A MA
P OF THIS PLACE?
2520 A:$T
2530 M:Y
2540 UY:*MAP
2550 T:WANT TO PLAY SOME MORE?
```

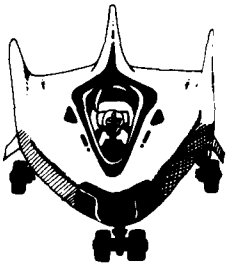
```
2560 A:$T
2570 M:Y
2580 JY:*GAME
2590 T:WOULD YOU LIKE ME TO FIX THE DR
OPSHAFT FOR YOU?
2600 A:$T
2610 M:Y
2620 CY:#A=1
2630 E:
2640 *LAUNCH A:=$RIGHT
2650 M:CRAFT,BOAT,SHIP
2660 EN:
2670 E(#R<>9):
2680 T:The hatch closes, the airlock b
lows open and your escape craft is sho
t out into space.
2690 PA:100
2700 T:
2710 T:RANK: STARVING ADVENTURER  Too
bad you don't know how to pilot this l
ifeboat.
2720 J:*END
2730 *SHOOT A:=$RIGHT
2740 M:PHASER,GUN
2750 JN:*STOCK
2760 A:=$Y6
2770 M:-1
2780 JN:*AINTGOT
2790 T:There must be something wrong w
ith the phaser.  It won't fire.
2800 J:*WHATNOW
2810 *ENERGY A:=$Y3
2820 M:-1
2830 EY:
2840 T:[CLEAR]
2850 C:@709=-256
2860 T:The light in this room is blind
ing.  You can't see a thing.
2870 *LOOP6 T:
2880 :WHAT NOW?  \
2890 A:$T
2900 M: T [
2910 C:#R=14
2920 EY:
2930 J:*LOOP6
2940 *ROOMCOLOR T:[CLEAR]
2950 C:@B752=1
2960 C:$W=W#R
2970 A:#T=$$W
2980 C:@709=#T
2990 E:
3000 *OBJECTS A:#T=$Y1
3010 T(#T=#R):There is a PILOT cartrid
ge here.  \
3020 A:#T=$Y2
3030 T(#T=#R):There is a small cutting
 laser here.  \
3040 A:#T=$Y3
3050 T(#T=#R):There are some protectiv
e sunglasses here.  \
3060 A:#T=$Y4
```

```
3070 T(#T=#R):A PILOT reference manual
 lies open on a counter.  \
3080 A:#T=$Y5
3090 T(#T=#R):A plastic coin is lying
on the floor.  \
3100 A:#T=$Y6
3110 T(#T=#R):There is a hand phaser h
ere.  \
3120 A:#T=$Y7
3130 T(#T=#R):A powerful smelling soyb
urger has been dropped on the floor.
\
3140 A:#T=$Y8
3150 T(#T=#R):Someone left their Secur
ity Card here.  \
3160 A:#T=$Y10
3170 T(#T=#R):There is a bed pan here.
  \
3180 A:#T=$Y11
3190 T(#T=#R):TRION crystals lie scatt
ered on the floor.  \
3200 A:#T=$Y12
3210 T(#T=#R):There is a Trion BOMB on
 the floor in the middle of the room.
 \
3220 E:
3230 *CORRIDORS T:
3240 :
3250 T:POSSIBLE EXITS . . .
3260 T:
3270 T(#R=1):     DROPSHAFT Q
3280 T(#R=14):     LIFTSHAFT Q
3290 T(#R=9):     ESCAPE CRAFT
3300 C:$T=X#R
3310 A:=$$T
3320 *LOOP1 MS:*
3330 EN:
3340 T:     CORRIDOR $LEFT
3350 A:=$RIGHT
3360 J:*LOOP1
3370 *MOVE
3380 C:$MOVE=$COMMAND#R
3390 C:#T=#R
3400 A:#R=$$MOVE
3410 J:*BARRIERS
3420 *MOVE1 J((#R=13)*(#D=0)):*TURTLE
3430 J((#R=14)*(#A=0)):*SHAFT
3440 J(#R=17):*ENERGY
3450 J((#R=15)*(#M=0)):*ENGINE
3460 E:
3470 *BARRIERS A:=$Y8
3480 M:-1
3490 JY:*MOVE1
3500 A:=#R
3510 M: 1 , 2 , 4 , 6 , 7
3520 JY:*FIELD
3530 M: 10 , 5 , 13 , 8 , 3 , 14
3540 JN:*MOVE1
3550 A:=#T
3560 M: 1 , 4 , 6 , 7
3570 JN:*MOVE1
```

```
3580 *FIELD C:#R=#T
3590 T:A security force field blocks y
our entrance to Corridor $LEFT
3600 J:*WHATNOW
3610 *INVENTORY C:#T=1
3620 T:YOU ARE CARRYING . . .
3630 T:
3640 *LOOP4 U:*MOD2
3650 U(#J=-1):*YOUHAVE
3660 C:#T=#T+1
3670 J(#T<13):*LOOP4
3680 J:*WHATNOW
3690 *YOUHAVE A:$T=Z#T
3700 T:$$T
3710 E:
3720 *TRANSPORT T(#R<>3):You must be i
n the Transporter Room to transport.
3730 J(#R<>3):*WHATNOW
3740 T(#I=0):TRANSPORTER DE-ACTIVATED
3750 J(#I=0):*WHATNOW
3760 T(#B=0):RANK: BEGINNING ADVENTURE
R  You forgot to activate the bomb!
3770 T(#B):RANK: MASTER ADVENTURER!!
You have successfully destroyed the Sc
out Ship.  Mission Accomplished!!
3780 J:*END
3790 *TURTLE A:=$Y7
3800 M:-1
3810 C:#R=#T
3820 JY:*NOTURTLE
3830 T:GROWL!! RUMBLE!! ROAR!! There's
 a wild Snapping Turtle down that corr
idor!  It won't let you pass!
3840 J:*WHATNOW
3850 *NOTURTLE C:#D=1
3860 A:$T=Y7
3870 C:$$T=0
3880 C:#C=#C-1
3890 T:A ferocious snapping Turtle lea
ps upon you, grabs the soyburger and d
isappears.
3900 J:*WHATNOW
3910 *SHAFT C:#R=#T
3920 T:<<DROPSHAFT - OUT OF ORDER>>
3930 J:*WHATNOW
3940 *INITIALIZE U:*COVER
3950 U:*COLORS
3960 U:*EXITS
3970 U:*CONNECTIONS
3980 U:*PLACEOBJECTS
3990 U:*INSTRUCTIONS
4000 C:#R=3
4010 E:
4020 *COLORS C:$W1=-29180
4030 C:$W2=-2034
4040 C:$W3=17420
4050 C:$W4=11268
4060 C:$W5=-19958
4070 C:$W6=28168
4080 C:$W7=1550
4090 C:$W8=3588
```

```
4100 C:$W9=-8700
4110 C:$W10=4110
4120 C:$W11=28686
4130 C:$W12=-23026
4140 C:$W13=-28146
4150 C:$W14=12302
4160 C:$W15=14848
4170 C:$W16=-19442
4180 C:$W17=21518
4190 C:$W18=8462
4200 E:
4210 *DESCRIPTION T(#R=1):SECURITY CEN
TRAL:  Glowing television screens cove
r the walls of this room.  \
4220 T(#R=2):SHIP'S COMMAND:  The capt
ain's chair sits empty before the humm
ing instrument boards.  \
4230 T(#R=3):TRANSPORTER ROOM:  Two gu
ards lie unconscious on the floor.  Tr
ansport controls stand ready.  \
4240 T(#R=4):ARMORY:  Most of the powe
r packs, phasers, blasters and explosi
ves in this room are securely locked.
 \
4250 T(#R=5):GALLEY: This room is fill
ed with dining tables.  An unearthly s
mell comes from the food vending machi
ne.  \
4260 T(#R=6):MAIN COMPUTER SECTION:  T
he huge bulk of the computer dominates
 the room.  \
4270 T(#R=7):BRIG:  The energy bars on
 the detention cells are not active.
Weird torture devices dangle from with
in.  \
4280 T(#R=8):MEDICAL BAY:  Many medica
l instruments are stored in locked cab
inets on the sterile white walls.  \
4290 T(#R=9):ESCAPE CRAFT DOCK:  Four
sleek space boats lay suspended by cab
les, ready for launch.  \
4300 T(#R=10):PHASER/SHIELD CONTROL:
A giant view-screen displays nothing b
ut the immense emptiness of space.  \
4310 T(#R=11):CREW QUARTERS:  This hug
e compartment is jammed with tiny cubi
cles.  \
4320 T(#R=12):RECREATION HALL:  The si
des of this room support shiny athleti
c equipment.  A whirlpool boils nearby
.  \
4330 T(#R=13):OFFICER'S QUARTERS:  Bea
utifully decorated cabins contain larg
e beds and heavy carved chests.  \
4340 T(#R=14):SMALL ARMS FIRING RANGE:
  A wide variety of stun guns hang on
the wall ready for firing.  \
4350 T(#R=15):ENGINE ROOM:  Five long
cylindrical tubes glow with pulsating
power.  There is a low hum in the air.
  \
```

```
4360 T(#R=16):HYDROPONIC GARDEN:  Many
 green plants twine wildly about this
room.  \
4370 T((#N=0)*(#R=16)):A six-foot weed
 looks at you and whispers, "FERTILIZE
 ME."  \
4380 T(#N*(#R=16)):A very happy lookin
g ten-foot weed is growing in one corn
er.  \
4390 T(#R=17):ENERGY STORAGE:  A huge
pulsating glass globe showers the room
 with an eerie pink glow.  \
4400 T(#R=18):WASTE MANAGEMENT CHAMBER
:  Brown fluids filter through plexigl
as tubes into massive settling basins.
  \
4410 E(#R<>2):
4420 T(#I=0):There is an empty cartrid
ge slot below a display which reads: T
RANSPORTER DE-ACTIVATED.  \
4430 T(#I):A display reads: TRANSPORTE
R ACTIVATED.  \
4440 E:
4450 *EXITS C:$X1=A*C*E*G*
4460 C:$X2=A*B*
4470 C:$X3=C*D*
4480 C:$X4=E*F*
4490 C:$X5=G*H*
4500 C:$X6=B*I*P*
4510 C:$X7=I*J*
4520 C:$X8=D*J*K*
4530 C:$X9=K*L*
4540 C:$X10=F*L*M*
4550 C:$X11=M*N*
4560 C:$X12=H*N*O*
4570 C:$X13=O*P*
4580 C:$X14=R*S*T*U*
4590 C:$X15=R*
4600 C:$X16=S*
4610 C:$X17=T*
4620 C:$X18=U*
4630 E:
4640 *CONNECTIONS C:$A1=2
4650 C:$A2=1
4660 C:$B2=6
4670 C:$B6=2
4680 C:$C1=3
4690 C:$C3=1
4700 C:$D3=8
4710 C:$D8=3
4720 C:$E1=4
4730 C:$E4=1
4740 C:$F4=10
4750 C:$F10=4
4760 C:$G1=5
4770 C:$G5=1
4780 C:$H5=12
4790 C:$H12=5
4800 C:$I6=7
4810 C:$I7=6
4820 C:$J7=8
```



```
4830 C:$J8=7
4840 C:$K8=9
4850 C:$K9=8
4860 C:$L9=10
4870 C:$L10=9
4880 C:$M10=11
4890 C:$M11=10
4900 C:$N11=12
4910 C:$N12=11
4920 C:$O12=13
4930 C:$O13=12
4940 C:$P13=6
4950 C:$P6=13
4960 C:$Q1=14
4970 C:$Q14=1
4980 C:$R14=15
4990 C:$R15=14
5000 C:$S14=16
5010 C:$S16=14
5020 C:$T14=17
5030 C:$T17=14
5040 C:$U14=18
5050 C:$U18=14
5060 E:
5070 *COVER C:@B1374=1
5080 WRITE:S
5090 C:@B84=4
5100 WRITE:S,    * SCOUT  SHIP *
5110 C:@B84=10
5120 WRITE:S,    copyright  1982
5130 WRITE:S
5140 WRITE:S
5150 WRITE:S,    dale  disharoon
5160 WRITE:S
5170 WRITE:S,    jeremiah knoche
5180 WRITE:S
5190 WRITE:S,    miles pickering
5200 CLOSE:S
5210 E:
5220 *PLACEOBJECTS C:$Y1=0
5230 C:$Y2=7
5240 C:$Y3=10
5250 C:$Y4=10
5260 C:$Y5=11
5270 C:$Y6=12
5280 C:$Y7=0
5290 C:$Y8=0
5300 C:$Y9=0
5310 C:$Y10=8
5320 C:$Y11=0
5330 C:$Y12=4
5340 C:$Z1=CARTRIDGE
5350 C:$Z2=LASER
5360 C:$Z3=SUNGLASSES
5370 C:$Z4=MANUAL
5380 C:$Z5=COIN
5390 C:$Z6=PHASER
5400 C:$Z7=SOYBURGER
5410 C:$Z8=CARD
5420 C:$Z9=FLUID
5430 C:$Z10=PAN
```

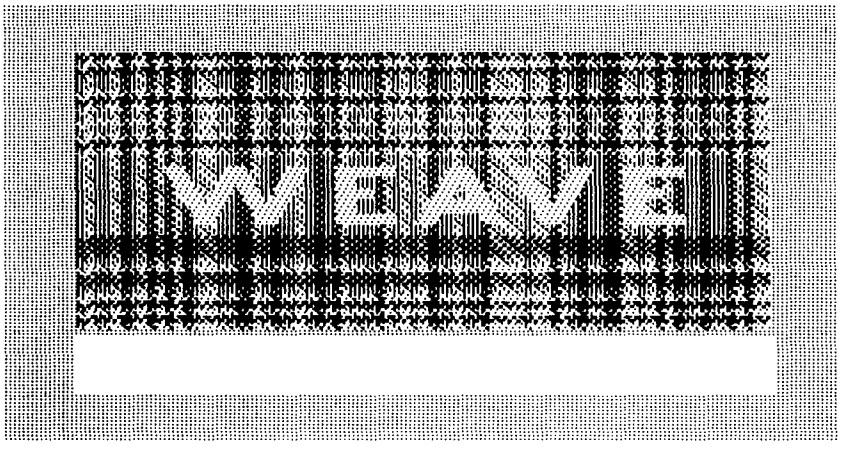
```
5440 C:$Z11=TRION
5450 C:$Z12=BOMB
5460 E:
5470 *END T:
5480 :* GAME OVER *
5490 *FOREVER J(@B764=255):*FOREVER
5500 C:@B764=255
5510 J:*START
5520 *MAP C:@B82=1
5530 T:[CLEAR]
5540 C:@709=10
5550 POS:1,0
5560 T:                [Q][8(R)][E]
5570 T:  [Q][8(R)][E]  ¦SHIELDS ¦  [Q]
[8(R)][E]
5580 T:  ¦ESCAPE  [A][2(R)][D]PHASERS
[A][2(R)][D]CREW    ¦
5590 T:  ¦CRAFT   ¦  [Z][3(R)][W][4(R)
][C]  ¦QUARTERS¦
5600 T:  [Z][R][W][6(R)][C]      ¦
   [Z][6(R)][W][R][C]
5610 T:    ¦  [Q][2(R)][E]   [Q][3(R)]
[X][4(R)][E]   [Q][2(R)][E]  ¦
5620 T:    ¦  ¦T ¦   ¦WEAPONS ¦   ¦  ¦
  ¦
5630 T:    ¦  ¦R ¦   [Z][3(R)][W][4(R)
][C]   ¦  ¦  ¦
5640 T:    ¦  ¦A ¦       ¦        ¦G ¦
  ¦
5650 T:    ¦  ¦N ¦   [Q][3(R)][X][4(R)
][E]   ¦A ¦  ¦
5660 T:[Q][3(R)][X][E] ¦S ¦   ¦SECURIT
Y¦   ¦L ¦ [Q][X][3(R)][E]
5670 T:¦MED.[A][R][D]P [A][3(R)][D]
     [A][3(R)][D]L [A][R][D]REC.¦
5680 T:¦BAY ¦ ¦O ¦   ¦DROP  + ¦   ¦E ¦
 ¦ROOM¦
5690 T:[Z][3(R)][W][C] ¦R ¦   ¦SHAFT
 ¦   ¦Y ¦ [Z][W][3(R)][C]
5700 T:    ¦  ¦T ¦   [Z][3(R)][W][4(R)
][C]   ¦  ¦  ¦
5710 T:    ¦  ¦E ¦       ¦        ¦  ¦
  ¦
5720 T:    ¦  ¦R ¦   [Q][3(R)][X][4(R)
][E]   ¦  ¦  ¦
5730 T:    ¦  ¦  ¦   ¦COMMAND ¦   ¦  ¦
  ¦
5740 T:    ¦  [Z][2(R)][C]   [Z][3(R)]
[W][4(R)][C]   [Z][2(R)][C]  ¦
5750 T:  [Q][R][X][6(R)][E]      ¦
   [Q][6(R)][X][R][E]
5760 T:  ¦ BRIG   ¦  [Q][3(R)][X][4(R)
][E]  ¦OFFICER ¦
5770 T:  ¦        [A][2(R)][D]YOU ARE
[A][2(R)][D]QUARTERS¦
5780 T:  [Z][8(R)][C]  ¦HERE  * ¦  [Z]
[8(R)][C]
5790 T:                [Z][8(R)][C]\
5800 PA:900
5810 C:@B82=2
```

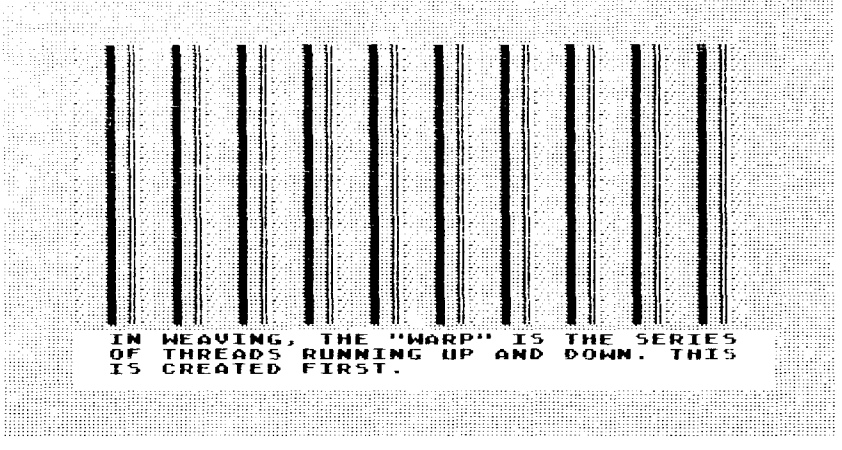
```
5820 U:*ROOMCOLOR
5830 U:*DESCRIPTION
5840 POS:2,6
5850 E:
5860 *HELP
5870 T:[CLEAR]
5880 :THESE ARE SOME OF THE WORDS I KN
OW.
5890 :
5900 :VERBS:              NOUNS:
5910 :
5920 :TAKE    SEARCH      COIN          WE
ED
5930 :DROP    LAUNCH      CRAFT         BO
MB
5940 :PRESS   SHOOT       PHASER        GL
OBE
5950 :SAY     INSERT      CARD          ST
ART
5960 :READ    LOOK        CARTRIDGE     LA
SER
5970 :EAT     INVENTORY   SOYBURGER     PA
N
5980 :BREAK   TRANSPORT   SUNGLASSES    TR
ION
5990 :CUT     ACTIVATE    MANUAL        FL
UID
6000 :FEED                CHEST         GU
ARD
6010 :
6020 :Try drawing a simple map as you
go.
6030 J:*WHATNOW
6040 *INSTRUCTIONS T:[CLEAR]
6050 C:@B752=1
6060 C:@B710=16
6070 T:
6080 :YOU HAVE BEAMED ABOARD AN ALIEN
SPACE CRAFT.  MOST OF THE ALIEN CREW A
RE NOT ONBOARD YET.  YOUR MISSION \
6090 :IS TO DESTROY THE SHIP AND TRANS
PORT YOURSELF OUT, ALIVE.
6100 :
6110 :TO DESTROY THE SHIP YOU MUST FIN
D AN EXPLOSIVE, ARM IT AND ACTIVATE IT
 NEAR THE SHIP'S ENGINES.
6120 :
6130 :TRANSPORT TO SAFETY BY USING THE
 COMMAND 'TRANSPORT'.
6140 :
6150 :TO MOVE: TYPE THE 'LETTER' OF AN
 OPEN CORRIDOR.  IF YOU NEED HELP DURI
NG THE GAME, TYPE 'HELP'.
6160 :
6170 :GOOD LUCK...
6180 :
6190 :PRESS START
6200 *STARTKEY J(@B53279<>6):*STARTKEY
6210 E:
```

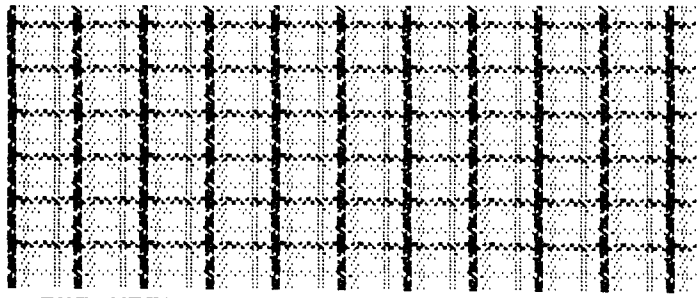
Weave

The computer can be used to model events and processes in the world. In order to show what a computer modeling program is like, we developed a weaving program. The program describes how weaving is done and then allows you to weave your own patterns. You create your own warp and then weave through it with your weft. Each time you create a pattern, you can weave with anywhere from one to eight threads of the same color.

On the surface your weaving is restricted to the three PILOT pen colors and black, the background color. However, combining stripes of colors in the right way creates browns, greens, purples, and many other colors. For example, single alternating threads of yellow and blue will produce a pale green. This is a program to try many times. After a while you'll be able to control pattern and color, and design handsome fabrics that could actually be woven.







THE WEFT IS THE SERIES OF THREADS RUNNING ACROSS, GOING OVER AND THEN UNDER EACH THREAD OF THE ORIGINAL WARP.

If you look through the program, you'll notice a graphic symbol that we haven't yet introduced. The symbol is %; it is used to create four graphics variables: %X, %Y, %A, and %Z. You cannot assign values to these variables or change their values. These variables tell you at any point in a program where the turtle is, what direction it is pointing, and what color the screen is at its current location. They do this in the following way (notice that they are preceded by a T: command):

T:%X = X coordinate of the turtle

T:%Y = Y coordinate of the turtle

T:%A = The angle the turtle is facing

T:%Z = 0 (ERASE or background color) or
1 (RED) or
2 (YELLOW) or
3 (BLUE)

(The T:%Z command must always be preceded by a GR:GO 0 command.)

These variables are convenient. For example, in the simple program

```
10 GR:PEN BLUE
20 GR:TURN 50
30 GR:DRAW 40
```

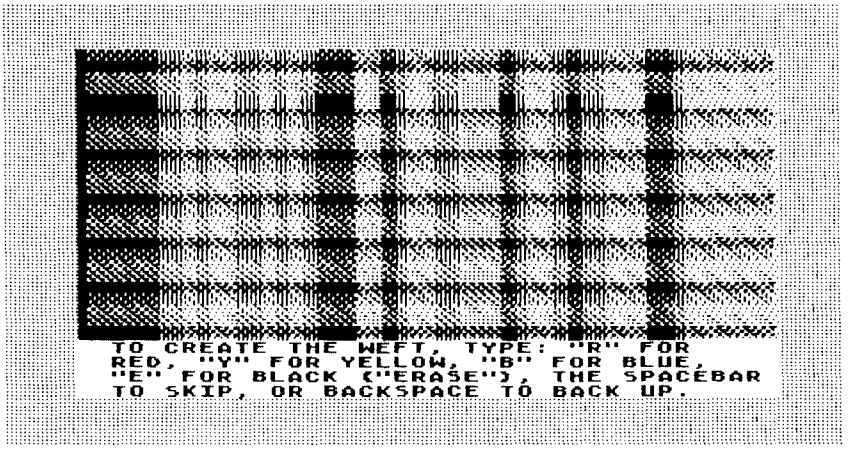
these variables will let you know exactly where you end up. In this case, for example,

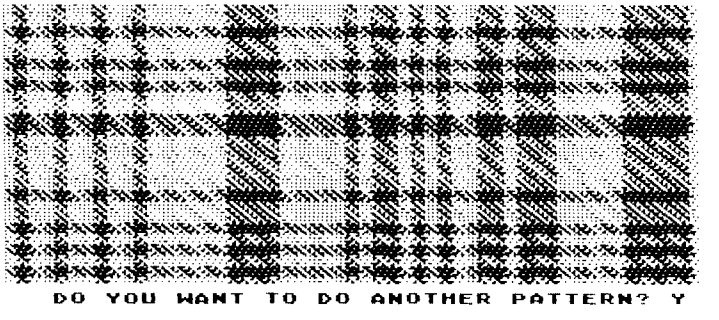
1. T:%X = 31
2. T:%Y = 26
3. GR:GO 0

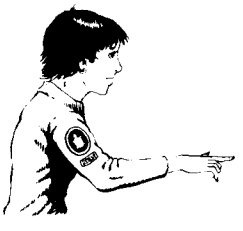
T:%Z = 3

4. T:%A = 50

This gives you the information that the turtle is at 31,26, turned to 50°, and has a BLUE pen, which is not that easy to obtain otherwise. Even more important is the fact that these variables can be used in your program to determine further movement of the turtle. The weaving program makes use of this quite extensively. For example, lines 1120 to 1320, which determine how the pattern is drawn, all use these % variables to orient and control the turtle.







```
2 R:          *** WEAVE ***
4 R:        COPYRIGHT (C) 1982
6 R:   BY ANDY HALL AND HERB KOHL
100 R: INITIALIZATION FOR *GETKEY
110 C:@B1778=32
120 C:@1779=@58404+1
130 C:@B1781=141
140 C:@1782=1785
150 C:@B1784=96
160 C:#A=0
170 U:*TITLE
180 PA:210
```

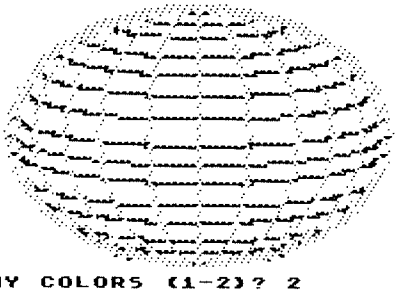
```
190 GR:CLEAR
200 T(#A=0):IN WEAVING, THE "WARP" IS
THE SERIES OF THREADS RUNNING UP AND D
OWN. THIS IS CREATED FIRST.
210 U:*WARPDEM
220 PA:600
230 T:[CLEAR]THE WEFT IS THE SERIES OF
 THREADS RUNNING ACROSS, GOING OVER AN
D THEN UNDER EACH THREAD OF THE \
240 T:ORIGINAL WARP.\
250 U:*WEFTDEM
260 PA:600
270 *AGAIN T:[CLEAR]\
280 T(#A=0):NOW YOU GET TO WEAVE YOUR
OWN PATTERN. \
290 T:HOW MANY THREADS PER KEYSTROKE F
OR THE WARP (1-9)? \
300 A:#Q
310 J((#Q<1)+(#Q>9)):*AGAIN
320 GR:CLEAR
330 T:[CLEAR]TO CREATE YOUR WARP, TYPE
 EITHER: "R" FOR RED, "Y" FOR YELLOW,
"B" FOR BLUE, \
340 T:THE SPACEBAR TO SKIP, OR THE BAC
KSPACE TO BACK UP. YOU MAY START.\
350 U:*WARP
360 *THREADS T:[CLEAR]HOW MANY THREADS
 PER KEYSTROKE FOR THE WEFT (1-8)?
370 A:#Q
380 J((#Q>8)+(#Q<1)):*THREADS
390 T:[CLEAR]TO CREATE THE WEFT, TYPE:
 "R" FOR RED, "Y" FOR YELLOW, "B" FOR
BLUE, "E" FOR BLACK ("ERASE"), \
400 T:THE SPACEBAR TO SKIP, OR BACKSPA
CE TO BACK UP. \
410 GR:PENUP;GOTO-79,47
420 C:#D=-2
430 U:*WEFT
440 C:#A=1
450 T:[CLEAR] DO YOU WANT TO DO ANOTHE
R PATTERN? \
460 A:
470 M:N
480 JN:*AGAIN
490 E:
500 *GETKEY CALL:1778
510 C:#K=@B1785
520 J(#K=155):*BREAKKEY
530 E:
540 *WARP
550 GR:PENUP;GOTO-80,47
560 *LOOP1 U:*GETKEY
570 J((#K<>32)*(#K<>66)*(#K<>82)*(#K<>
89)*(#K<>126)):*LOOP1
580 GR(#K=32):PENERASE
590 GR(#K=66):PENBLUE
600 GR(#K=89):PENYELLOW
610 GR(#K=82):PENRED
620 GR(#K=126):#Q(PENERASE;GOTO%X,47;D
RAWTO%X,-29;PENUP;GOTO%X-1,47)
630 J(#K=126):*LOOP1
```

```
640 C(78-%X<#Q):#Q=78-%X
650 GR:#Q(GOTO%X+1,47;DRAWTO%X,-29)
660 J(%X<77):*LOOP1
670 E:
680 *WEFT U:*GETKEY
690 J((#K<>32)*(#K<>66)*(#K<>69)*(#K<>
82)*(#K<>89)*(#K<>126)):*WEFT
700 C:#C=#Q
710 GR(#K=32):PENUP;GOTO((-1-%X)*(#Q\2
))+((1-(#Q\2))*%X),%Y-#Q
720 C((#K=32)*(#Q\2=1)):#D=-#D
730 J(#K=32):*SKIP
740 C((%Y<=(#Q-30))*(#K<>126)):#C=30+%
Y
750 *LOOP2 GR(#K=89):PENYELLOW
760 GR(#K=82):PENRED
770 GR(#K=66):PENBLUE
780 GR(#K=69):PENERASE
790 J(#K=126):*BACKUP
800 C:#D=-#D
810 GR:78(GOTO%X+#D,%Y);PENUP;GOTO%X\2
*2-1+%X,%Y-1
820 C:#C=#C-1
830 J(#C>0):*LOOP2
840 *SKIP J(%Y>-30):*WEFT
850 E:
860 *BACKUP GR:PENUP;GOTO((-1-%X)*(#Q\
2))+((1-(#Q\2))*%X),%Y+#Q
870 C(#Q\2=1):#D=-#D
880 J:*WEFT
890 E:
900 *WARPDEM GR:GOTO-79,47
910 *DLOOP GR:DRAWTO%X,-29;GOTO%X+1,47
;DRAWTO%X,-29
920 C:@B1363=@B1363\3+1
930 C:#R=%X+(4-@B1363*2)
940 GR(#R<78):GOTO#R,47
950 J(#R<78):*DLOOP
960 E:
970 *WEFTDEM GR:PENRED;GOTO-79,47
980 C:#D=1
990 *DLOOP2 GR:78(GOTO%X+2,%Y);PENUP;G
OTO78,%Y-1
1000 C:@B1363=#D
1010 GR:78(GOTO%X-2,%Y);PENUP;GOTO-79,
%Y-1
1020 C:#D=#D\3+1
1030 C:@B1363=#D
1040 GR(%Y>-27):GOTO%X,%Y-4+@B1363
1050 J(%Y>-27):*DLOOP2
1060 E:
1070 *TITLE GR:CLEAR;PENYELLOW;GOTO-80
,46
1080 *TLOOP GR:GOTO%X+1,46;DRAWTO%X,-2
9;GOTO%X+1,46;DRAWTO%X,-29
1090 C:@B1363=?\3+1
1100 J(%X<78):*TLOOP
1110 GR:PENRED;GOTO-78,46
1120 *TLOOP2 GR:78(GOTO%X+2,%Y);GOTO%X
-1,%Y-1;78(GOTO%X-2,%Y);GOTO%X+1,%Y-1
1130 C:#D=?\3+1
```

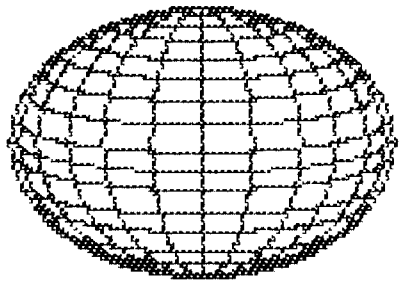
```
1140 C:@B1363=#D
1150 C((%Y<20)*(%Y>-5)):@B1363=2
1160 C(?\20=0):@B1363=?\2+1
1170 J(%Y>-30):*TLOOP2
1180 GR:PENERASE;GOTO%X,%Y
1190 GR:PENBLUE;GOTO-59,15;15(DRAWTO%X
+4,%Y;GOTO%X-3-(%Y\2),%Y-1)
1200 GR:15(DRAWTO%X+4,%Y;GOTO%X-4+(%Y\
2),%Y+1);15(DRAWTO%X+4,%Y;GOTO%X-3-(%Y
\2),%Y-1)
1210 GR:15(DRAWTO%X+4,%Y;GOTO%X-4+(%Y\
2),%Y+1);DRAWTO%X+4,%Y
1220 GR:GOTO-20,15;3(DRAWTO%X+13,%Y;GO
TO%X-13,%Y-1);3(DRAWTO%X+4,%Y;GOTO%X-4
,%Y-1)
1230 GR:3(DRAWTO%X+8,%Y;GOTO%X-8,%Y-1)
;4(DRAWTO%X+4,%Y;GOTO%X-4,%Y-1)
1240 GR:2(DRAWTO%X+13,%Y;GOTO%X-13,%Y-
1);DRAWTO%X+13,%Y
1250 GR:GOTO-2,0;15(DRAWTO%X+4,%Y;GOTO
%X-3-(%Y\2),%Y+1);15(DRAWTO%X+4,%Y;GOT
O%X-4+(%Y\2),%Y-1);DRAWTO%X+4,%Y
1260 GR:GOTO3,6;4(DRAWTO%X+8,%Y;GOTO%X
-8,%Y-1)
1270 GR:GOTO20,15;15(DRAWTO%X+4,%Y;GOT
O%X-3-(%Y\2),%Y-1);15(DRAWTO%X+4,%Y;GO
TO%X-4+(%Y\2),%Y+1);DRAWTO%X+4,%Y
1280 GR:GOTO46,15;3(DRAWTO%X+13,%Y;GOT
O%X-13,%Y-1);3(DRAWTO%X+4,%Y;GOTO%X-4,
%Y-1)
1290 GR:3(DRAWTO%X+8,%Y;GOTO%X-8,%Y-1)
;4(DRAWTO%X+4,%Y;GOTO%X-4,%Y-1)
1300 GR:2(DRAWTO%X+13,%Y;GOTO%X-13,%Y-
1);DRAWTO%X+13,%Y
1310 E:
1320 *BREAKKEY
```

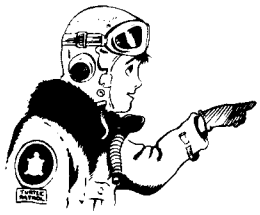
Earth

Earth is a short program that illustrates how you can model a three-dimension sphere using PILOT turtle graphics. This program can be added to; it is possible to make the sphere seem to be spinning or moving. You may want to turn the whole program into a module for future use.



```
HOW MANY COLORS (1-2)? 2
```



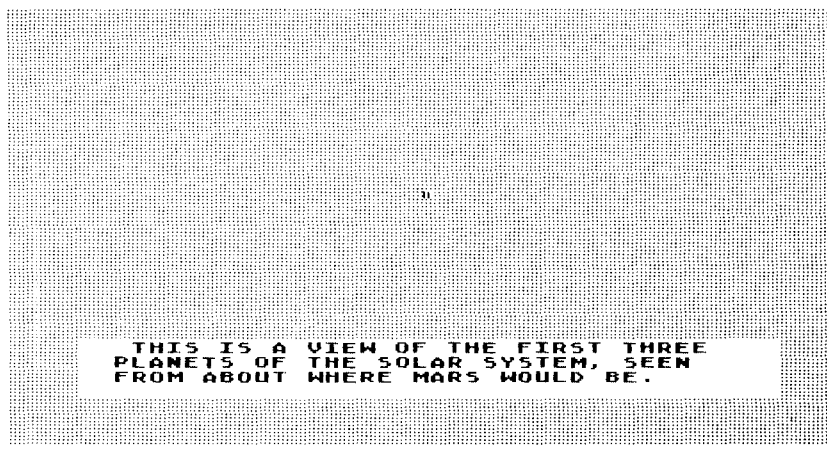


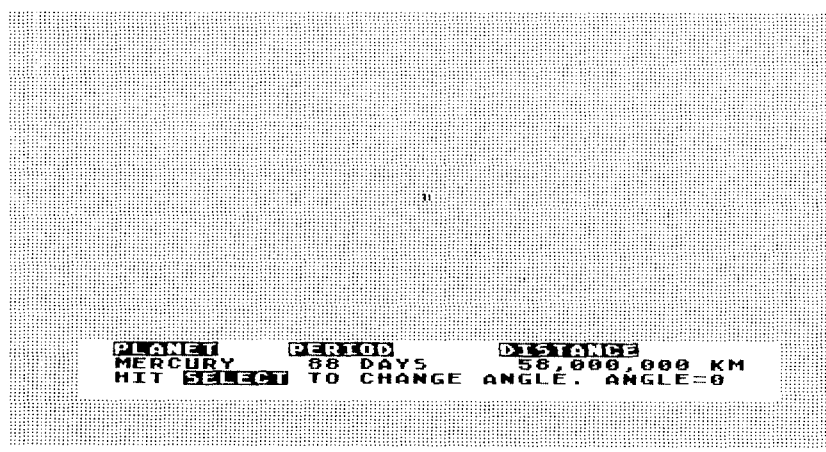
```
HOW MANY COLORS (1-2)? 1
```

```
2 R:            *** EARTH ***
4 R:          COPYRIGHT (C) 1982
6 R:             BY ANDY HALL
100 *EARTH
110 GR:CLEAR;PENUP
120 T:HOW MANY COLORS (1-2)? \
130 A:#K
140 J((#K<>1)*(#K<>2)):*EARTH
150 C:#N=180
160 *ELOOP GR:GOTO0,0;TURNTO#N;GO43
170 C:#N=#N-10
180 GR(%X=0):GOTO%X+1,%Y
190 C:#A=10000/%X
200 C:#Y=%Y*11/13+7
210 C:#B=#A*5
220 GR:TURNTO270;GOTO0,0
230 U:*ELLIPSE
240 J(#N>0):*ELOOP
250 C:#B=10000/43*13/11-1
260 C:#N=90
270 C:#Y=7
280 *ELOOP2 GR:GOTO0,0;TURNTO#N;GO43
290 C:#A=10000/(%X+1)
300 GR:TURNTO0;GOTO0,0
310 U(#K=1):*ELLIPSE
320 U(#K=1):*ELLIPSE
330 U(#K=2):*ELLIPSE2
340 C:#N=#N-15
350 J(#N>-1):*ELOOP2
360 T:HIT RETURN TO EXIT. \
370 A:
380 E:
390 *ELLIPSE GR:180(GO9999;PENYELLOW;G
OTO%X/#A,%Y/#B+#Y;TURN-1;PENUP;GOTO0,0
)
400 E:
410 *ELLIPSE2 GR:360(GO9999;PENBLUE;GO
TO%X/#A,%Y/#B+#Y;TURN-1;PENUP;GOTO0,0)
420 E:
```

Orbit

Orbit is a dynamic program that uses ATARI Computer missile graphics. It is hard to picture using a screen dump since it involves constant movement. It presents a view of the first three planets from the sun—Mercury, Venus, and Earth—as seen orbiting around the sun from Mars.





You can change the angle from which you see the planets orbiting about the sun. Watching the program run for a while and experimenting with the angle of vision gives one an uncanny sense of actually being in space and seeing planetary movement speeded up. The model shows conjunctions of planets, shows the relative speed of the three planets, and, in ways

a static picture never could, shows how different planetary bodies are constantly changing their positions relative to each other.

```
2 R:          *** ORBIT ***
4 R:        COPYRIGHT (C) 1982
6 R:           BY ANDY HALL
100 *ORBIT
110 GR:CLEAR
120 R:SAVE SPACE FOR MISSILE GRAPHICS
130 C:#H=@B742-4
140 C:#H=#H-(#H\4)
150 C:@B742=#H
160 C:@B54279=#H
170 C:#H=#H*256
180 C:#L=#H+384
190 C:#C=64
200 R:CLEAR MISSILE MEMORY
210 *CLOOP C:@#L=0
220 C:#L=#L+2
230 C:#C=#C-1
240 J(#C>0):*CLOOP
250 R:INITIALIZE MISSILES' X & Y COORD
INATES
260 C:#P=98
270 C:#R=148
280 C:#X=196
290 C:#Q=56
300 C:#S=56
310 C:#Y=56
320 R:SET MISSILE POSITIONS IN MEMORY
330 C:#L=#H+384+#Q
340 C:@B#L=1
350 C:#L=#H+384+#S
360 C:@B#L=@B#L+4
370 C:#L=#H+384+#Y
380 C:@B#L=@B#L+16
390 C:@B53252=#P
400 C:@B53253=#R
410 C:@B53254=#X
420 R:SET COLOR FOR THREE MISSILES
430 C:@B704=40
440 C:@B705=202
450 C:@B706=148
460 R:REGULAR PLAYFIELD
470 C:@B559=42
480 R:SET PLAYER-MISSILE PRIORITIES
490 C:@B623=4
500 GR:CLEAR;PENYELLOW;GOTO-1,9;TURNTO
90;4(DRAW1;TURN90);PENUP
510 R:ENABLE MISSILE GRAPHICS
520 C:@B53277=1
530 T:[CLEAR] THIS IS A VIEW OF THE FI
RST THREE PLANETS OF THE SOLAR SYSTEM,
 SEEN FROM ABOUT WHERE MARS WOULD BE.
540 PA:700
550 T:[CLEAR] MERCURY: \
560 C:#L=704
570 U:*BLINKSOUND
```

```
580 T:THE FIRST PLANET FROM THE SUN, N
AMED FOR THE FLEET-OF-FOOT MESSENGER O
F THE GODS.
590 PA:500
600 T:[CLEAR] VENUS: \
610 C:#L=705
620 U:*BLINKSOUND
630 T:THE SECOND PLANET FROM THE SUN,
WITH VERY THICK "AIR", THICKER EVEN TH
AN THAT FOUND IN SOME MAJOR CITIES.\
640 PA:500
650 T:[CLEAR] EARTH: \
660 C:#L=706
670 U:*BLINKSOUND
680 T:THE THIRD PLANET FROM THE SUN, B
ELIEVED BY SOME SCIENTISTS TO SUPPORT
LIFE.
690 PA:500
700 T:[CLEAR]PLANET     PERIOD       DIS
TANCE
720 C:#D=91
730 C:#M=260
740 C:#N=160
750 C:#O=122
760 *MOVE GR:GOTOO,O;TURNTO#M;GO30
770 C:#P=128+%X
780 C:@B53252=#P
790 C:#L=#H+384+#Q
800 C:@B#L=@B#L-1
810 C:#Q=56-(%Y*11/13/#D)
820 C:#L=#H+384+#Q
830 C:@B#L=@B#L+1
840 GR:GOTOO,O;TURNTO#N;GO58
850 C:#R=128+%X
860 C:@B53253=#R
870 C:#L=#H+384+#S
880 C:@B#L=@B#L-4
890 C:#S=56-(%Y*11/13/#D)
900 C:#L=#H+384+#S
910 C:@B#L=@B#L+4
920 GR:GOTOO,O;TURNTO#O;GO80
930 C:#X=128+%X
940 C:@B53254=#X
950 C:#L=#H+384+#Y
960 C:@B#L=@B#L-16
970 C:#Y=56-(%Y*11/13/#D)
980 C:#L=#H+384+#Y
990 C:@B#L=@B#L+16
1000 C:#M=#M-8
1010 C:#N=#N-3
1020 C:#O=#O-2
1030 C(#O<O):#O=#O+360
1040 J(@B53279<>5):*NOSELECT
1050 C:#D=#D-1
1060 C(#D=90):#D=5
1070 C(#D=O):#D=91
1080 C(#D=4):#D=3
1090 C:#E=90/#D
1100 T:[DOWN]#E DEGREES ABOVE THE ELLI
PTIC PLANE   [2(DOWN)]\
```

```
1120 *NOSELECT T(#0=120):MERCURY     88
 DAYS      58,000,000 KM
1130 C:#E=90/#D
1140 T(#0=120):HIT SELECT TO CHANGE AN
GLE. ANGLE=#E  [2(DOWN)]\
1150 T(#0=0):VENUS      225 DAYS     107
,500,000 KM
1160 T(#0=0):          HIT SPACE TO EXI
T.           [2(DOWN)]\
1170 T(#0=240):EARTH      365 DAYS     1
50,000,000 KM
1180 T(#0=240):
                [2(DOWN)]\
1190 C:#E=0
1200 C(@B764=33):#E=1
1210 C:@B764=255
1220 J(#E=1):*ENDIT
1230 J:*MOVE
1240 *BLINKSOUND C:#C=16
1250 C:#N=709-#L*6
1260 C:#D=@B#L
1270 *BLINKLOOP C:@B#L=#D-@B#L
1280 C:#N=16+#N
1290 SO:#N
1300 PA:10
1310 C:#C=#C-1
1320 J(#C>0):*BLINKLOOP
1330 SO:0
1340 E:
1350 *ENDIT C:@B53252=0
1360 C:@B53253=0
1370 C:@B53254=0
1380 E:
```

Angle Blaster

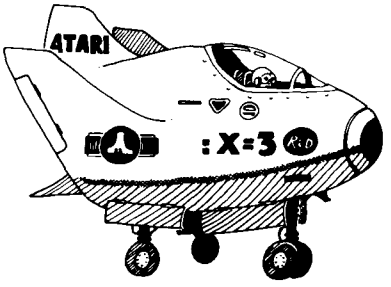
Enter the angle of your shot and estimated range to blast enemy ships in this make-believe space encounter. Set in deep space, this game will make you a turtle angle expert. It also helps with estimation skills.

```
100 R:          ANGLE BLASTER
110 R:       (C) COPYRIGHT 1982
120 R:       by  Dale Disharoon
130 R:
140 J:*START
150 *DRAWSHIP GR:PENRED;GOTO #X-2,#Y-2
;DRAWTO #X,#Y;DRAWTO #X+2,#Y-2;GOTO #X
,#Y-1;GOTO #X-1,#Y-2;GOTO#X+1,#Y-2
160 E:
170 *ERASESHIP GR:PENERASE;GOTO #X-2,#
Y-2;DRAWTO #X,#Y;DRAWTO #X+2,#Y-2;GOTO
 #X,#Y-1;GOTO#X-1,#Y-2;GOTO#X+1,#Y-2
180 E:
190 *CHECKHIT C:#H=0
200 GR:PENUP;GOTO #X,#Y
210 C(%Z=3):#H=1
220 GR:GOTO #X-1,#Y-1
230 C(%Z=3):#H=1
240 GR:GOTO #X-2,#Y-2
250 C(%Z=3):#H=1
```

```
260 GR:GOTO #X+1,#Y-1
270 C(%Z=3):#H=1
280 GR:GOTO #X+2,#Y-2
290 C(%Z=3):#H=1
300 GR:GOTO #X,#Y-1
310 C(%Z=3):#H=1
320 GR:GOTO #X-1,#Y-2
330 C(%Z=3):#H=1
340 GR:GOTO #X+1,#Y-2
350 C(%Z=3):#H=1
360 J(#H=0):*SHOOTLOOP
370 C:#S=#L*50+#S
380 C:#L=4
390 C:#E=#E+75
400 C:#J=708
410 U:*EXPLOSION
420 GR:CLEAR
430 J:*GAMELOOP
440 *START
450 U:*COVER
460 *RESTART
470 U:*WINDOWMOD
480 U:*VALUES
490 U:*CROSSHAIRS
500 *GAMELOOP
510 U(#C=1):*DRAWHAIRS
520 U:*PLACESHIP
530 U:*DRAWSHIP
540 *SHOOTLOOP J(#E<1):*ENDGAME
550 T:[CLEAR]
560 U:*STATISTICS
570 C:@B656=0
580 C:@657=1
590 J(#L=1):*ENEMY
600 T:angle \
610 A:#A
620 C:@B656=0
630 C:@657=11
640 T:range \
650 A:#R
660 J(#R=0):*SHOOTLOOP
670 C:#E=#E-#R
680 C(#E<0):#R=#R+#E
690 U:*LASER
700 C:#L=#L-1
710 J:*CHECKHIT
720 *COVER GR:QUIT
730 C:@B752=1
740 T:
750 C:@B710=0
760 C:#D=@560
770 C:#T=#D+9
780 C:@B#T=6
790 C:#T=#T+1
800 C:@B#T=6
810 C:$T=**********************************
*****
820 POS:3,1
830 T:$T
840 POS:3,4
850 T:ANGLE  BLASTER
860 POS:13,8
```

```
870 T:COPYRIGHT 1982
880 POS:13,10
890 T:Dale Disharoon
900 POS:3,14
910 T:$T
920 POS:7,17
930 T:Do you need instructions?
940 U:*YESNO
950 U(#K=43):*INSTRUCTIONS
960 E:
970 *YESNO C:@B764=255
980 *LOOPA J(@B764=255):*LOOPA
990 C:#K=@B764
1000 C:@B764=255
1010 E(#K=43):
1020 E(#K=35):
1030 J:*LOOPA
1040 *WINDOWMOD GR:CLEAR
1050 C:#D=@560
1060 C:#D=#D+85
1070 C:@B#D=70
1080 C:#D=#D+3
1090 C:@B#D=6
1100 C:#D=#D+1
1110 C:@B#D=6
1120 C:#D=#D+1
1130 C:@B#D=6
1140 E:
1150 *CROSSHAIRS T:[CLEAR]
1160 C:@B656=0
1170 C:@B657=1
1180 T:CROSSHAIRS?
1190 U:*YESNO
1200 C(#K=43):#C=1
1210 T:[CLEAR]
1220 E:
1230 *DRAWHAIRS GR:PENYELLOW;GOTO 0,18
;DRAWTO 0,-2;GOTO -10,8;DRAWTO 10,8
1240 GR:GOTO -64,8;DRAWTO -55,8;GOTO 5
5,8;DRAWTO 64,8
1250 E:
1260 *PLACESHIP
1270 C:#X=?\99-49
1280 C:#Y=?\76-28
1290 J((#X>-15)*(#X<15)*(#Y>-5)*(#Y<24
)):*PLACESHIP
1300 E:
1310 *EXPLOSION C:#F=0
1320 GR:PENYELLOW
1330 C:#V=15
1340 C:#T=0
1350 *EXPLODELOOP C:#F=#F+10
1360 C:#T=#T+1
1370 C(#T=3):#V=#V-1
1380 C(#T=3):#T=0
1390 C:@B53761=6*16+#V
1400 C:@B53760=#F
1410 GR:GOTO (?\11)-5+#X,(?\11)-5+#Y
1420 C:@B#J=?
1430 J(#F<>440):*EXPLODELOOP
1440 SO:
1450 C:@B708=70
```

```
1460 C:@B709=26
1470 E:
1480 *LASER GR:PENBLUE;GOTO 0,8;TURNTO
 #A
1490 C:#B=0
1500 *LASERLOOP C:#B=#B+1
1510 GR:DRAW 1
1520 C:#F=?\32
1530 SO:#F
1540 J(#B<>#R):*LASERLOOP
1550 SO:
1560 E:
1570 *STATISTICS C:@B656=1
1580 C:@B657=1
1590 T:SC:#S \
1600 C:@B656=1
1610 C:@B657=11
1620 T:EN:#E \
1630 E:
1640 *VALUES C:#E=500
1650 C:#S=0
1660 C:#L=4
1670 C:#C=0
1680 E:
1690 *ENEMY C:#E=#E-75
1700 C:#T=3
1710 *THREE GR:PENRED;GOTO #X,#Y;DRAWT
O 0,8
1720 C:@B53761=8*16+12
1730 C:#F=50
1740 *ENEMYLOOP C:#F=#F+3
1750 C:@B53760=#F
1760 J(#F<100):*ENEMYLOOP
1770 GR:PENERASE;GOTO #X,#Y;DRAWTO 0,8
1780 SO:
1790 U:*DRAWSHIP
1800 C:#T=#T-1
1810 J(#T):*THREE
1820 C:#L=4
1830 C:#X=0
1840 C:#Y=8
1850 C:#J=709
1860 U:*EXPLOSION
1870 J(#E<1):*ENDGAME
1880 GR:CLEAR
1890 J:*GAMELOOP
1900 *ENDGAME GR:QUIT
1910 C:@B1374=1
1920 WRITE:S
1930 C(#S>#W):#W=#S
1940 WRITE:S, * ANGLE  BLASTER *
1950 WRITE:S
1960 WRITE:S
1970 WRITE:S
1980 WRITE:S
1990 WRITE:S,    YOUR SCORE: #S
2000 WRITE:S
2010 WRITE:S,    HIGH SCORE: #W
2020 *ENDGAMELOOP C:@B85=2
2030 C:@B84=16
2040 WRITE:S,* press  start *
2050 C:#T=18
```

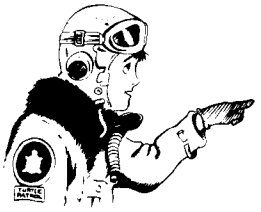


```
2060 *PAUSELOOP C:#T=#T-1
2070 J(@B53279=6):*OUT
2080 J(#T):*PAUSELOOP
2090 C:@B85=2
2100 C:@B84=16
2110 WRITE:S,                    [
2120 PA:17
2130 J:*ENDGAMELOOP
2140 *OUT CLOSE:S
2150 J:*RESTART
2160 *INSTRUCTIONS
2170 GR:QUIT
2180 C:@B710=0
2190 C:@B752=1
2200 POS:11,0
2210 T:* ANGLE BLASTER *
2220 T:
2230 T:1. Score as many points as poss
ible before your ship's ENERGY runs ou
t.
2240 T:
2250 T:2. Enter the ANGLE and RANGE of
 the enemy ship from the center of you
r view screen.
2260 T:
2270 T:3. You may enter a positive or
negative ANGLE value.
2280 T:
2290 T:4. The RANGE value is taken fro
m your ship's ENERGY.
2300 T:
2310 T:5. SCORING:  1st Shot - 150 poi
nts
2320 T:             2nd Shot - 100 poi
nts
2330 T:             3rd Shot -  50 poi
nts
2340 T:
2350 T:Gain 75 ENERGY points for each
hit
2360 T:Lose 75 ENERGY points for 3 mis
ses
2370 POS:12,23
2380 T:* PRESS START *\
2390 *WAIT J(@B53279<>6):*WAIT
2400 E:
```

Metagame

We end with a metagame, that is, a game to help organize your ideas and invent new games. This game is a bit like Eliza, our psychiatrist, in that the computer prompts you but doesn't actually use the specific information that you provide it. However, it does repeat and organize it in a way that can lead to the production of interesting games. This is not a trick program as Eliza was. It is a tool of thought, an instrument of provocation, a game about games.

```
METAGAME
WRITTEN BY
TED M. KAHN

(INSPIRED BY HESSE'S GLASS BEAD
GAME, M. CASPI'S 'EDUCOMBINATORICS,'
& PICODYNE CORP.)

HAVE YOU EVER WONDERED HOW YOU COULD
COME UP WITH NEW IDEAS FOR GAMES?

IF SO, THEN WELCOME TO METAGAME.
THIS IS AN EXERCISE IN USING YOUR OWN
CREATIVITY.  BY WEAVING DIFFERENT
GAME ELEMENTS TOGETHER (SOME GATHERED
FROM STRANGE PLACES), YOU MAY COME UP
WITH NEW COMBINATIONS WHICH MAY LEAD
TO WONDERFUL NEW GAME IDEAS FOR YOU
TO PROGRAM ON YOUR ATARI COMPUTER.

TO SHOW YOU HOW IT WORKS, LET'S WORK
THROUGH AN EXAMPLE...

(PRESS RETURN WHEN READY)
```



```
2 R:        *** METAGAME ***
4 R:      COPYRIGHT (C) 1982
6 R:         BY TED KAHN
8 R:      A GAME-IDEA GENERATOR
10 R: (BASED ON THE SYSTEM CALLED
12 R: 'EDUCOMBINATORICS' (C) 1977 BY
14 R: M. D. CASPI
100 T:                METAGAME
110 :
120 :                WRITTEN BY
130 :               TED M. KAHN
140 :
150 :(INSPIRED BY HESSE'S GLASS BEAD
GAME, M. CASPI'S 'EDUCOMBINATORICS,' &
 PICODYNE CORP.)
160 :
170 :HAVE YOU EVER WONDERED HOW YOU CO
ULD COME UP WITH NEW IDEAS FOR GAMES?
180 :
190 :IF SO, THEN WELCOME TO METAGAME.
 THIS IS AN EXERCISE IN USING YOUR \
200 :OWN CREATIVITY.  BY WEAVING DIFFE
RENT GAME ELEMENTS TOGETHER (SOME GATH
ERED FROM \
210 :STRANGE PLACES), YOU MAY COME UP
WITH NEW COMBINATIONS \
220 :WHICH MAY LEAD TO WONDERFUL NEW G
AME IDEAS FOR YOU TO PROGRAM ON YOUR A
TARI COMPUTER.
230 :
240 :TO SHOW YOU HOW IT WORKS, LET'S \
250 :WORK THROUGH AN EXAMPLE...
260 :
270 :(PRESS RETURN WHEN READY) \
280 A:
290 T:[CLEAR]
300 U:*NEWCOLOR [NEW BACKGROUND COLOR
310 T:FANTASY: EVERY GAME IS, IN PART,
 PLAYING WITH A FANTASY (E.G., FINDING
 TRUE LOVE, WEALTH, \
320 :WISDOM, ...).
330 :
340 :MY FANTASY IS TO \
350 *FANTASYX A:$FANTASY
```

```
360 M: NO, DON'T,?
370 TY:IF YOU'RE NOT SURE, TRY ONE OF
THE ONES LISTED, SUCH AS LOVE OR LEARN
ING.
380 :
390 JY:*FANTASYX
400 T:
410 T:OK, YOU WANT TO $FANTASY.  THINK
 ABOUT THIS FOR A MOMENT AND GET SOME
ASSOCIATIONS.
420 :LIST THE FIRST THREE WORDS OR IDE
AS THAT COME TO MIND WHEN YOU THINK OF
430 :'$FANTASY'.
440 :
450 T:FIRST IDEA?\
460 A:$IDEA1
470 T:SECOND?\
480 A:$IDEA2
490 T:THIRD?\
500 A:$IDEA3
510 T:
520 :GOOD.  NOW, AS YOU DEVELOP YOUR G
AME, KEEP $IDEA1, $IDEA2, & $IDEA3 IN
MIND.
530 :(YOU DON'T HAVE TO USE THEM.  THE
Y ARE JUST 'TRIGGERS' FOR MORE IDEAS).
540 :
550 :(RETURN TO CONTINUE) \
560 A:
570 T:[CLEAR]
580 U:*NEWCOLOR
590 T:GOAL: WHAT'S A GOAL OR OBJECTIVE
 YOU MIGHT LIKE (E.G., TO BE THE FASTE
ST, TO HAVE THE MOST, \
600 :TO BE THE FIRST, TO SURROUND MY O
PPONENT, ETC.)?
610 T:MY GOAL IS TO \
620 *GOALX A:$GOAL
630 T:
640 T:OK, SO YOU'RE GOING TO DESIGN A
GAME TO $FANTASY, WITH A GENERAL GOAL
BEING TO $GOAL.
650 :(PRESS RETURN) \
660 A:
670 T:[CLEAR]
680 U:*NEWCOLOR
690 T:SCENARIO: WHERE/WHEN DOES YOUR G
AME TAKE PLACE?  HERE ARE JUST A FEW P
OSSIBILITIES: \
700 :AN ISLAND, A MANSION, A CHESS BOA
RD, MEDIEVAL FRANCE, ANCIENT GREECE, 2
0TH CENTURY AMERICA.
710 :
720 :YOUR SCENARIO?\
730 *SCENARIOX A:$SCENARIO
740 T:
750 :AH, $SCENARIO.  THAT'S AN INTERES
TING SETTING FOR A GAME ABOUT $FANTASY
.
760 T:
770 :NOW, LET'S CONTINUE (PRESS RETURN
)\
780 A:
```

```
790 T:[CLEAR]
800 U:*NEWCOLOR
810 T:PLAYERS:  WHO'S PLAYING AND HOW
MANY PLAYERS (IN ADDITION TO THE TYPIC
AL 'ME AND MY COMPUTER', \
820 :YOU MIGHT CONSIDER SOME OF THE TH
E FOLLOWING:
830 :MY FAMILY, MY FOOTBALL TEAM, ME T
AKING THE ROLE OF A BLOOD CELL, ETC.
840 :YOUR TURN. WHO'S PLAYING? \
850 *PLAYERSX A:$PLAYERS
860 T:
870 :ARE YOU GETTING SOME NEW IDEAS?
880 A:
890 M:Y,GREAT,SURE,OF COURSE,THINK SO
900 TY:GOOD.  YOU SEE, GAMES ALL COME
FROM SOMEONE'S IMAGINATION.
910 TN:WELL, BE PATIENT.  IT WILL ALL
COME TOGETHER SHORTLY.
920 :
930 :(RETURN TO CONTINUE)\
940 A:
950 *A T:[CLEAR]
960 U:*NEWCOLOR
970 T:TIME: HOW LONG SHOULD THIS GAME
BE (30 SECONDS, AN HOUR, WEEKS, AS LON
G AS AN AVERAGE TV AD, ...)
980 :TIME? \
990 *TIMEX A:$TIME
1000 *SUMMARY1
1010 U:*NEWCOLOR
1020 T:[CLEAR]LET'S SEE WHAT WE HAVE S
O FAR.
1030 :YOU WANT TO DEVELOP A GAME TO $F
ANTASY.
1040 :THE OVERALL GOAL IS TO '$GOAL'.
1050 :THE PLAYERS ARE $PLAYERS (REMEMB
ER: 'ME' IS YOU), AND THE GAME WILL BE
 PLAYED OVER A TIME \
1060 :SPAN OF ABOUT $TIME.
1070 :
1080 :NOW DESCRIBE (IN 2 LINES OR LESS
) HOW YOUR FANTASY, GOAL, SCENARIO, &
PLAYERS \
1090 :MIGHT FIT TOGETHER IN A GAME.
1100 :(REMEMBER $IDEA1, $IDEA2, AND $I
DEA3)
1110 *DESCRIBEX A:$DESCRIBE1
1120 T:ANYTHING ELSE? \
1130 A:$DESCRIBE2
1140 T:
1150 :THINK ABOUT YOUR ELABORATION, BU
T DON'T FORCE YOURSELF TO COMMIT TO TH
ESE IDEAS JUST YET...
1160 :(PRESS RETURN) \
1170 A:
1180 *WINDFALL
1190 *NEWCOLOR
1200 R:USED TO CHANGE THE BACKGROUND C
OLOR
1210 C:@B710=(?\16)*16+8
1220 C:@B709=2
1230 E:
```

Appendix I A Guide to Understanding Our Codes

The program listings in this book use a simple code to represent graphics characters, inverse letters, and special function keys. The chart shows all the codes that may be used.

1. Whenever you see a letter in brackets, hold down the [CTRL] key and type the letter. This will generate a special graphics character. If a program listing contained the code [T], you would hold the [CTRL] key and press the [T] key. This will produce a small ball shape.

2. When a graphics character is repeated, it will be shown in an abbreviated fashion. Four small ball shapes would be shown as [4(T)], not [T][T][T][T].

3. Different graphics characters to be printed together are represented like this: [TS]. This tells you to hold the [CTRL] key and then press the [T], followed by the [S] key. A small ball and a cross shape will appear.

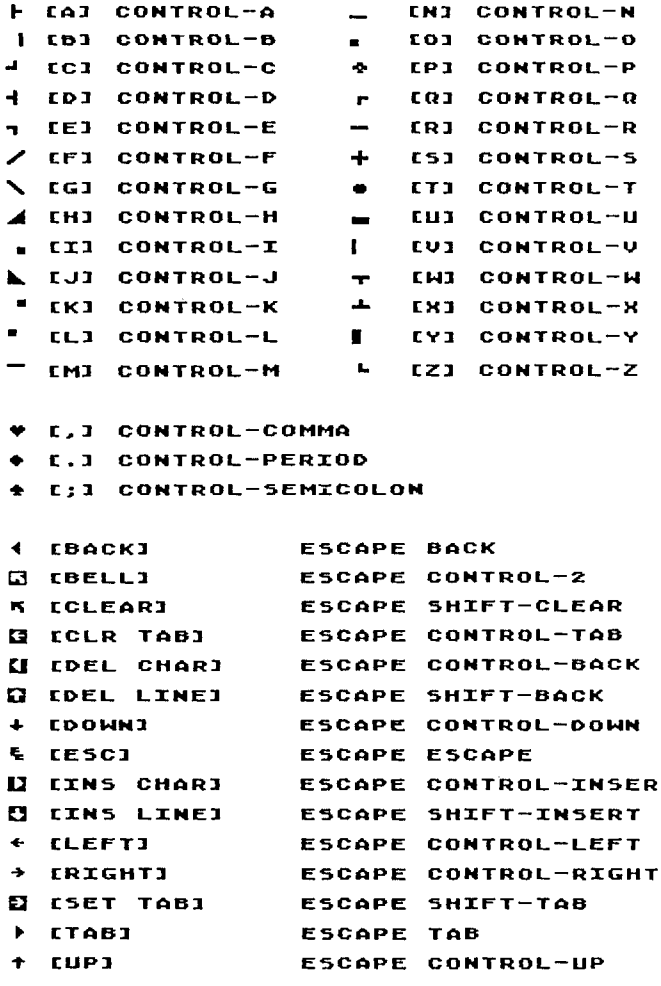
4. Three sets of small ball [T] and cross [S] would be coded like this: [3(TS)]. This is really an abbreviation of [T][S][T][S][T][S].

5. Remember not to type the brackets. The brackets are only there to show you that a special character code is coming. When you are copying a program, type in the actual character, not our code.

6. Special escape function keys like [CLEAR SCREEN], [BELL], and [DELETE LINE] are spelled in an abbreviated form within brackets. If you saw the code [CLEAR], you would press the [ESCAPE] key once and then hold the [SHIFT] key while you press the [CLEAR] key. This will print as a curving arrow character and when printed would clear the screen.

7. Anything in the program listings that is underlined should be typed in inverse characters. Press the ATARI Computer key for inverse print.

8. If you see a special graphics code underlined, as in [T], you should press the ATARI Computer inverse key, hold the [CTRL] key, and then press the [T] key. Finally, remember to press the ATARI Computer key again to turn off the inverse character mode.



Appendix II

PILOT Peeks and Pokes

Your computer is like a post office that contains many, many individual post office boxes. In the computer each box is called a memory location; like the numbered post office box, each memory location is labeled with a number. This number is called the memory location's address. Each memory location in your computer can hold one number. For instance, memory location 10 might be holding the number 0. Location 29998 might hold the number 255. The number contained in the memory location must be in the range of 0 to 255.

Here we've collected a group of memory addresses that can be useful when you are programming in PILOT. Each of these memory locations has a special function that you can use. By placing certain values in the location, you will notice many different kinds of effects.

In PILOT we use a special form of the COMPUTE (C:) statement to place values into memory locations. This form uses the @ symbol with a B and means "at byte number." Thus @B100 means "at byte 100." For example, to put the value 0 into location 710 you would type

```
C:@B710=0
```

If you try this one, you will notice that your television screen turns black. This is because location 710 holds the value for the color of your text screen. Here's another one to try:

```
C:@B82=20
```

If you enter this statement, you will notice that your left-hand margin has changed just a bit. Location 82 normally contains the value 2, which is your normal left-hand screen margin. We have just set it to 20, which is half way across the screen. To reset the margin you can press the [SYSTEM RESET] key or use the following statement:

```
C:@B82=2
```

Some of our programs place a value in a memory by using a numeric variable for either the location address or the value. We could write

```
C:#L=82
C:#V=2
C:@B#L=#V
```

Sometimes you will need to know what value is already in a particular memory location. Type

```
C:#V=@B82
```

After executing this statement, the variable #V will contain the value currently in location 82. You may also use the statement like this:

```
T:@B82
```

If your margin is at its usual setting, you will get a value of 2. Each program that follows will introduce you to one or more useful memory locations. This, of course, is not a complete collection of possible uses or locations. After all, the ATARI Computer contains thousands of memory locations; the number of possible uses is entirely up to you and your creativity.

Clock Demo

The ATARI Computer contains an internal clock, which is constantly being updated. The clock is tied to the television frame counter and thus ticks in 60ths of a second rather than whole seconds. This little demonstration will reset the clock locations to zero and then display their contents on the screen. You will notice that location 20 is the one that increases by 60ths of a second. Every time location 20 reaches the value of 256, it is reset to zero and location 19 is increased by one. Location 19 and 18 behave in a similar manner.

```
100 R: *** CLOCK DEMO ***
110 C:@B18=0
120 C:@B19=0
130 C:@B20=0
140 *LOOP
150 POS:2,2
160 T:LOCATION 20 = @B20   [
```

```
170 POS:2,4
180 T:LOCATION 19 = @B19  [
190 POS:2,6
200 T:LOCATION 18 = @B18  [
210 J:*LOOP
```

Left Margin

Here's one that we already played with. Location 82 holds the value of the left-hand screen margin. It is normally set to 2. You will find it useful for positioning text in the center of the screen. You may also want to play with location 83, which holds the value of the right-hand screen margin.

```
100 R:*** LEFT MARGIN DEMO ***
110 C:@B82=5
120 T:
130 T:MARGIN 5
140 C:@B82=10
150 T:
160 T:MARGIN 10
170 C:@B82=15
180 T:
190 T:MARGIN 15
```

Moving Left Margin

As you will see in this program, the margin setting can be used in a playful manner to create interesting animated screen displays.

```
100 R: *** MOVING MARGIN DEMO ***
110 *LOOP
120 C:#M=#M+1
130 C(#M>22):#M=1
140 C:@B82=#M
150 T:MOVING MARGINS
160 J:*LOOP
```

Color Text Screen

In order to make use of the color text modes in PILOT, you must change the contents of memory location 1373 and 1374. When location 1373 contains a zero, there will be no blue text window at the bottom of the screen. Placing a value of 16 in location 1373 will enable the text window. The value in location 1374 determines the size of colored letters. This program will demonstrate both letter sizes with and without the text window.

```
100 R:*** COLOR TEXT SCREEN DEMO ***
110 C:@B1373=0
120 C:@B1374=1
130 WRITE:S, LONGER LETTERS
140 PA:180
150 CLOSE:S
160 C:@B1373=16
```

```
170 C:@B1374=1
180 WRITE:S, LONGER LETTERS
190 WRITE:S, TEXT WINDOW
200 PA:180
210 CLOSE:S
220 C:@B1373=0
230 C:@B1374=2
240 WRITE:S, LONGER LETTERS
250 WRITE:S, TALLER LETTERS
260 PA:180
270 CLOSE:S
280 C:@B1373=16
290 C:@B1374=2
300 WRITE:S, LONGER LETTERS
310 WRITE:S, TALLER LETTERS
320 WRITE:S, TEXT WINDOW
330 PA:180
```

Position on Color Text Screen

This program enables a color text screen and then prints in different places on the screen. The PILOT word POS: however, has no effect in this mode. We therefore use the memory location 85 to set the horizontal coordinate and location 84 to set the vertical coordinate of our position. This program prints the word THERE at POS:4,9 and the word WHERE? at POS:10,3.

```
100 R: POSITION ON COLOR TEXT SCREEN
110 R: @B85 IS X POSITION
120 R: @B84 IS Y POSITION
130 C:@B1373=16
140 C:@B1374=1
150 WRITE:S
160 C:@B85=4
170 C:@B84=9
180 WRITE:S,THERE
190 C:@B85=10
200 C:@B84=3
210 WRITE:S,WHERE?
220 C:@B85=7
230 C:@B84=17
240 WRITE:S,HERE
```

Position in Text Window

This program is similar to the color text screen position program except that these two memory locations are used to position words in the blue text window. Location 657 contains the horizontal coordinate and 656 contains the vertical coordinate.

```
100 R:** POSITION IN TEXT WINDOW **
110 GR:CLEAR
120 C:@B657=5
130 C:@B656=0
140 T:POSITION 5,0
150 C:@B657=10
160 C:@B656=1
```

```
170 T:POSITION 10,1
180 C:@B657=15
190 C:@B656=2
200 T:POSITION 15,2
210 C:@B657=20
220 C:@B656=3
230 T:POSITION 20,3\
240 *LOOP
250 J:*LOOP
```

Color Registers

Locations 708, 709, 710, 711, and 712 are possibly the most useful to the PILOT programmer. These locations contain the color values that will be used when you specify PENRED, PENYELLOW, PENBLUE, and PENERASE.

708 = PENRED
709 = PENYELLOW
710 = PENBLUE
711 = Affects only the color text mode
712 = BACKGROUND COLOR

To create a different color set, multiply the memory location to your color choice (0–15) times 16 plus the brightness value (0–14 even). This is one that you will really need to play around with.

```
100 R: *** COLOR REGISTERS ***
110 GR:CLEAR
120 GR:PENRED
130 GR:GOTO 79,25;FILLTO 79,45
140 GR:PENYELLOW
150 GR:GOTO 79,5;FILLTO 79,25
160 GR:PENBLUE
170 GR:GOTO 79,-15;FILLTO 79,5
180 C:@B708=0*16+14
190 C:@B709=12*16+8
200 C:@B710=3*16+10
210 C:@B712=5*16+4
```

Random Color Registers

The color memory locations 708 through 712 are called the color registers. This program chooses a random color register and fills it with a random color and brightness value. It then loops back and does the same thing again very quickly.

```
100 R:*** RANDOM COLOR REGISTERS ***
110 GR:CLEAR
120 GR:PENRED
130 GR:GOTO 79,25;FILLTO 79,45
140 GR:PENYELLOW
150 GR:GOTO 79,5;FILLTO 79,25
160 GR:PENBLUE
170 GR:GOTO 79,-15;FILLTO 79,5
180 *LOOP
```

```
100 R: START-SELECT-OPTION KEY DEMO
110 T:
120 T:Press START, SELECT or OPTION
130 T:
140 *LOOP
150 C:#C=@B53279
160 T(#C=6):YOU PRESSED START
170 T(#C=5):YOU PRESSED SELECT
180 T(#C=3):YOU PRESSED OPTION
190 J:*LOOP
```

Direct Sound Register Control

You may bypass the PILOT sound command and work directly with the sound register locations to create a variety of sound effects. Each sound channel has a memory location that contains the frequency to be played and another location that controls the quality of the sound and its volume.

53760 = SOUND 1 FREQUENCY
53761 = SOUND 1 CONTROL
53762 = SOUND 2 FREQUENCY
53763 = SOUND 2 CONTROL
53764 = SOUND 3 FREQUENCY
53765 = SOUND 3 CONTROL
53766 = SOUND 4 FREQUENCY
53767 = SOUND 4 CONTROL

```
100 R: *** DIRECT SOUND CONTROL ***
110 C:@B53761=10*16+8
120 *LOOP
130 POS:2,2
140 T:PITCH=#P
150 C:@B53760=#P
160 C:#P=#P+1
170 J(#P<256):*LOOP
```

```
190 C:#R=?\5+708
200 C:#H=?\16
210 C:#L=?\7+4
220 C:@B#R=#H*16+#L
230 J:*LOOP
```

Keyboard Codes

Memory location 764 is special in that it is set to a different value each time a different key is pressed on the computer keyboard. This program prints a list of all the letter and number codes and then allows you to try them on your own.

```
100 R:KEYBOARD CODES FROM ** 764 **
110 POS:7,4
120 T:PRESS KEYS TO SEE KEY CODES
130 POS:2,14
140 T:A=63  H=57  O=8   V=16  3=26
150 T:B=21  I=13  P=10  W=46  4=24
160 T:C=18  J=1   Q=47  X=22  5=29
170 T:D=58  K=5   R=40  Y=43  6=27
180 T:E=42  L=0   S=62  Z=23  7=51
190 T:F=56  M=37  T=45  1=31  8=53
200 T:G=61  N=35  U=11  2=30  9=48
210 *LOOP
220 POS:18,6
230 T:@B764  [
240 J:*LOOP
```

Input Demo

This program uses the keyboard code at location 764 to simplify input from the keyboard. The code for the letter *Y* is 43 and the code for *N* is 35. We can therefore tell whether the user is responding with a *yes* or *no*.

```
100 R: *** INPUT WITH 764 ***
110 T:
120 T:DO YOU HAVE ANY WOOL (Y/N)
130 T:
140 *LOOP
150 C:#K=@B764
160 T(#K=43):YES SIR, YES SIR
170 T(#K=35):NO SIR, NO SIR
180 C(#K<>255):@B764=255
190 J:*LOOP
```

Start-Select-Option Keys

Location 53279 is used to determine which of the console keys is being pressed.

7 = No button is being pressed

6 = The [START] key is pressed

5 = The [SELECT] key is pressed

3 = The [OPTION] key is pressed

ATARI® PILOT Activities and Games

Herb Kohl / Ted Kahn / Dale Disharoon

Would you like to draw designs and shapes, move objects on the screen, and develop word and number games on your ATARI® Home Computer? It's fun, it's entertaining, *and* it's easy to learn! What makes it simple is the PILOT programming language—a language that allows you to design complex programs using simple commands. **ATARI® PILOT Activities and Games** is filled with word and language games, math activities, sound and graphics activities (including how to animate objects), and even a chapter on adventure games, modeling events, and miniworlds! Whether you're young or old, you won't be able to put this book down once you start programming in PILOT. Your computer becomes your window to new adventures, new experiences, and new insights—so get ready to find out what ATARI® PILOT has in store for you!

Table of Contents

RESTON PUBLISHING COMPANY, INC. 0-8359-0321-4
A Prentice-Hall Company
Reston, Virginia