

AVON



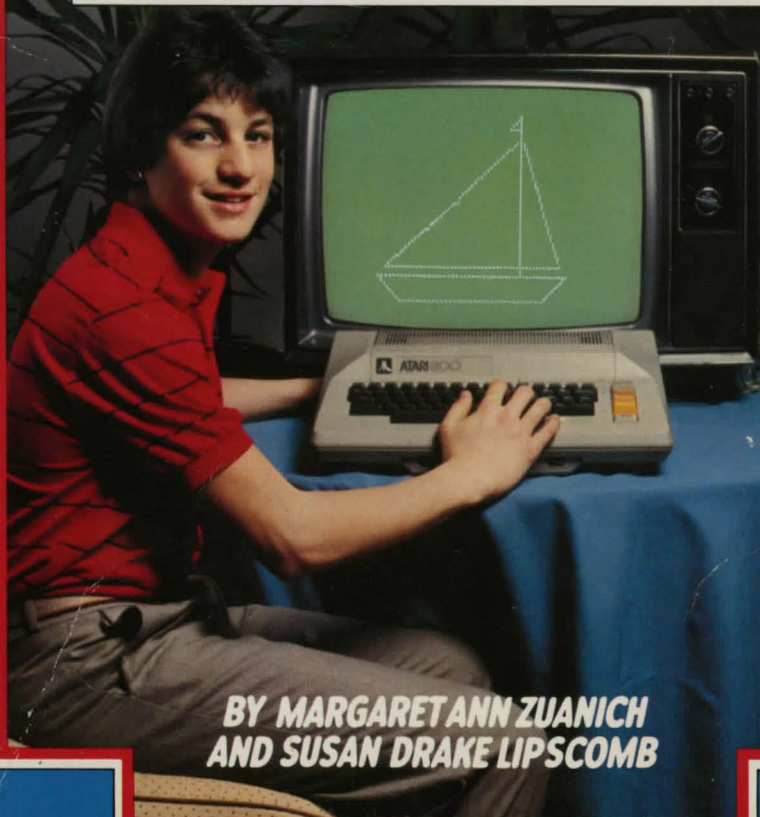
A guide to computer graphics

\$3.95

85050

BASIC FUN WITH GRAPHICS

THE  ***WAY***



***BY MARGARET ANN ZUANICH
AND SUSAN DRAKE LIPSCOMB***

*BASIC FUN
WITH GRAPHICS*

THE **ATARI®** *WAY*
COMPUTER

Other Avon Camelot Books by
Margaret Ann Zuanich and
Susan Drake Lipscomb

BASIC BEGINNINGS

BASIC FUN: COMPUTER GAMES, PUZZLES AND
PROBLEMS CHILDREN CAN WRITE

BASIC FUN WITH GRAPHICS: THE APPLE® COMPUTER WAY

BASIC FUN WITH GRAPHICS: THE IBM PC® COMPUTER WAY

MARGARET ANN ZUANICH and SUSAN DRAKE LIPSCOMB have a unique combination of skills that contributed to the creation of BASIC FUN WITH GRAPHICS. Margaret Zuanich's experience in the computer field has included everything from programming to management consulting. She earned her Master's Degree in Business and is now involved in computer systems training. Susan Lipscomb holds a Master's Degree in Education and has spent fourteen years in the area of language and learning disabilities. They both live in Palo Alto, California.

Avon Books are available at special quantity discounts for bulk purchases for sales promotions, premiums, fund raising or educational use. Special books, or book excerpts, can also be created to fit specific needs.

For details write or telephone the office of the Director of Special Markets, Avon Books, Dept. FP, 1790 Broadway, New York, New York 10019, 212-399-1357.

BASIC FUN WITH GRAPHICS

THE **ATARI®**
COMPUTER WAY

**MARGARET ANN ZUANICH
AND SUSAN DRAKE LIPSCOMB**

AN AVON  CAMELOT BOOK

5th grade reading level has been determined by using the Fry Readability Scale.

BASIC FUN WITH GRAPHICS: THE ATARI® COMPUTER WAY is an original publication of Avon Books. This work has never before appeared in book form.

This book is published by Avon Books which is not affiliated with Atari, Inc. and Atari is not responsible for any inaccuracies.

ATARI is a registered trademark of Atari, Inc. and 800 is a trademark of Atari, Inc.

AVON BOOKS

A division of
The Hearst Corporation
1790 Broadway
New York, New York 10019

Copyright © 1983 by Margaret Ann Zuanich and Susan Drake Lipscomb

Cover photograph by Aram Gesar

Published by arrangement with the authors

Library of Congress Catalog Card Number: 83-15528

ISBN: 0-380-85050-8

All rights reserved, which includes the right to reproduce this book or portions thereof in any form whatsoever except as provided by the U. S. Copyright Law. For information address Avon Books.

Library of Congress Cataloging in Publication Data

Zuanich, Margaret Ann.

Basic fun with graphics, the Atari computer way.

(An Avon/Camelot book)

Summary: Instructions for graphics programming on an Atari computer using BASIC.

1. Computer graphics—Juvenile literature.
2. Atari computer—Programming—Juvenile literature.
[1. Computer graphics. 2. Programming (Computers)
3. Atari computer—Programming. 4. Basic (Computer programming language)] I. Lipscomb, Susan Drake.
II. Title

T385.Z8 1983 001.55'3 83-15528
ISBN 0-380-85050-8

First Camelot Printing, November, 1983

CAMELOT TRADEMARK REG. U. S. PAT. OFF. AND IN
OTHER COUNTRIES, MARCA REGISTRADA, HECHO EN
U. S. A.

Printed in the U. S. A.

DON 10 9 8 7 6 5 4 3 2 1

TABLE OF CONTENTS

Introduction 7

How to use *BASIC Fun With Graphics:
The Atari Way* 9

1 PRINT:	11
1. Warmups	12
2. House	14
3. Ocean Liner	16
4. Space Fleet	17
5. Bugs	19
6. Rocket Ship	21
7. Bicycle	24
2 POSITION	25
1. Warmups	26
2. Face	28
3. Christmas Tree	30
4. Candy Jar	32
5. Tower	34
6. Grove	35
7. Hide-and-Seek	36
3 PLOT and DRAWTO	39
1. Warmups	40
2. Christmas Lights	45
3. Jet Plane	47
4. House	48
5. Flag	49
6. 3-D Room	51
7. Train	52
8. Skier	53

4 GRAPHICS MODES, COLOR, and SETCOLOR 55

1. Warmups 56
2. Fish 76
3. Car 77
4. Fishing Boat 79
5. Windmill 81
6. Snowflakes 83
7. Sailboat 85
8. Campground 87
9. Picture 89

5 ANIMATION 93

1. Warmups 94
2. Walking Figure 98
3. Flying Bird 100
4. Jumping Fish 102
5. Helicopter 104

6 FUN WITH GEOMETRY 107

1. Warmups 108
2. Apple Tree Harvest 115
3. Leaping Fish 118
4. Bird and Flower 120
5. Fancy Graphics 123

Glossary 125

INTRODUCTION

Are you ready for the fun and challenge of graphics programming? Do you want to use your computer to create your own pictures and designs, move objects around on the screen, or design your own video game? *BASIC Fun With Graphics: The Atari Way* gets you started, using the graphics programming statements on your computer. Each chapter includes exercises that show how the graphics statements work and simple BASIC programs that demonstrate exciting and fun ways to use these statements.

BASIC Fun With Graphics: The Atari Way presents a systematic yet entertaining approach to learning graphics programming. First you type in the exercises and programs exactly as they appear in the book. Then you experiment and change them, using your own ideas. Finally, you create your own designs and pictures, using those in the book as a guide.

BASIC Fun With Graphics: The Atari Way covers the beginning and intermediate graphics statements available in Atari Basic, as well as a chapter for "hotshots" that uses higher-level mathematical functions to create geometric figures and generate new and unique designs. It stops short of commands that directly address memory, like PEEK and POKE statements, and shape tables used in sophisticated graphics and animation. When you're ready for this level of programming, your computer's BASIC reference guide has a comprehensive discussion of how to proceed.

BASIC Fun With Graphics: The Atari Way is written in BASIC for the Atari 400 or 800 equipped with a BASIC cartridge. The programs will run on either a color or monochrome monitor. For visual appeal, SETCOLOR statements are included in many of the programs. To run on a monochrome or

black and white display, omit the SETCOLOR statements, but not the COLOR statements, from the program.

We found that experimenting with the graphics statements and creating designs and pictures on the screen was great fun. We hope you get as many "oohs," "aaahs," and chuckles as we have.

HOW TO USE BASIC FUN WITH GRAPHICS

1. Start up your Atari with the BASIC cartridge in it.
2. Type in each program line by line, including the line numbers, exactly as shown in the Program Listing. Press the RETURN key after each line.
3. Once you have a program entered onto your computer, type RUN and press Return. The computer will Run your program and display the results on the screen. It should look like the example in the book.
4. To check the programming statements, press RESET, type LIST and press Return. The computer will list your program.
5. Now try making the changes suggested in the text, or experiment with others, using ours as a guide.
6. When you're ready to save your program, use the SAVE or CSAVE command. To retrieve a program from the disk or cassette, use the LOAD or CLOAD command.
7. Before you enter another program, type NEW to erase anything currently in memory.
8. The Warmup exercises at the beginning of most chapters contain directions and examples using each graphics statement. Read these and work through the exercises before you proceed to the programs.
9. The pictures used as examples in this book were drawn using the screen display as a guide. The actual display on your screen may vary slightly from these examples.
10. If you have difficulties making a program work, check your program, line by line, against the listing in the book. All the programs and exercises were tested on an Atari 400 and an Atari 800.

1 PRINT

Use this program statement to tell the computer what to print on the screen or paper.

PROGRAMS

1. Warmups
2. House
3. Ocean Liner
4. Space Fleet
5. Bugs
6. Rocket Ship
7. Bicycle

WARMUPS

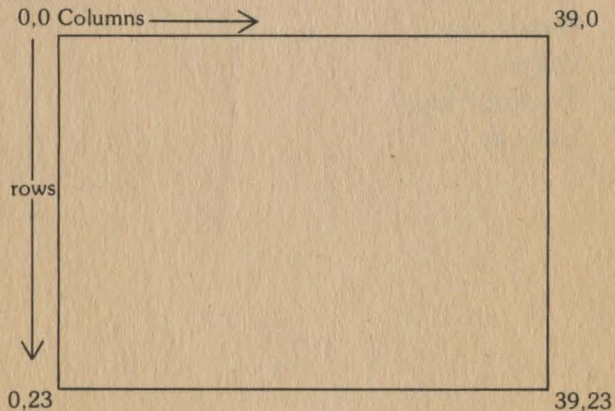
The format for the PRINT programming statement is:

PRINT " " or PR. " " or ? " "

You can use any combination of letters, numbers, and special characters in a PRINT statement.

Your computer has nine display modes: 3 text modes and 6 graphics modes. This chapter shows you how to use the GRAPHICS 0 text mode to draw shapes and objects.

In GRAPHICS 0, you have 24 rows and 40 columns. The rows are numbered from top to bottom, the columns from left to right, on your screen. If you think of your screen as a piece of graph paper, then the top left square is 0,0 and the bottom right square is 39,23. The Atari default margin in GRAPHICS 0 is 2,1 (column 2, row 1), where "READY" appears.



The graphics mode statement sets the display type. The format is:

GRAPHICS N or GR. N

where N is a number that determines the display type, as follows:

Type	Mode	Rows	Columns	Full Rows	Text Window
0	Text	24	40	24	No
1	Text	20	20	24	Yes
2	Text	10	20	12	Yes
3	Graphics	20	40	24	Yes
4	Graphics	40	80	48	Yes
5	Graphics	40	80	48	Yes
6	Graphics	80	160	96	Yes
7	Graphics	80	160	96	Yes
8	Graphics	160	320	192	Yes

The GRAPHICS statement can be abbreviated when you are typing to GR. N. GRAPHICS can be used either in a program or in immediate mode.

If you want to use the full screen in modes 1 through 8, thereby eliminating the text window, add 16 to the GRAPHICS N statement. Then make the last statement of your program a continuous loop, so you have to BREAK the program to end it. For example:

```
100 GRAPHICS 3 + 16
999 GOTO 999
```

These two statements will allow you to use 24 rows and 40 columns for your graphic commands in graphics mode 3.

Atari's graphic modes will be discussed in greater detail in Chapter 4.

HOUSE

Use PRINT to draw a house. Put this program in your computer by typing each line exactly as shown in the Program Listing.

Once you have this program up and running, try adding a chimney by using the Atari's edit keys (↑, ↓, ←, →, INS, DEL).

```

      / \
     /   \
    /     \
   /       \
  /         \
 I  I_I I_I  I         I
 I         I         I
 I         I   I_I   I
 I   I_I   I         I
 I   I_I   I         I
 I   I_I   I         I

```

PROGRAM LISTING

```

100 GRAPHICS 0
110 PRINT "      / \
120 PRINT "     /   \
130 PRINT "    /     \
140 PRINT "   /       \
150 PRINT "I  I_I I_I  I         I"
160 PRINT "I         I         I"
170 PRINT "I         I   I_I   I"
180 PRINT "I   I_I   I         I"
190 PRINT "I   I_I   I         I"

```

NOTES

Line 100: GRAPHICS 0 puts the program run in graphics mode 0.

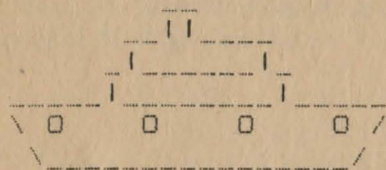
Lines 110-190: When the program runs, the PRINT statement tells

the computer to print out everything between the two " marks on each line exactly as it is shown in the program.

OCEAN LINER

Draw an ocean liner with this program.

Now add some portholes or another deck to make your own ship.



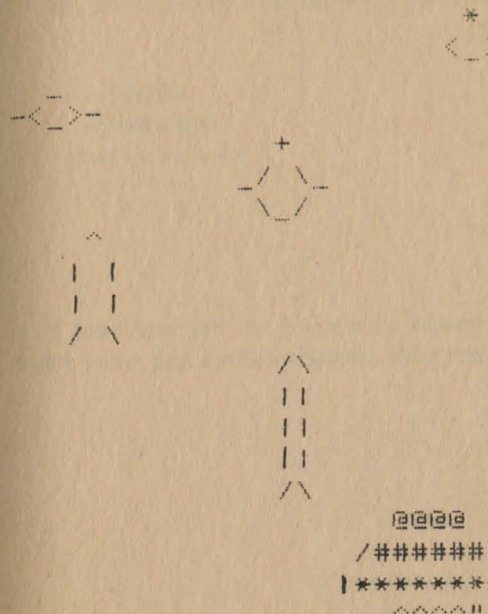
PROGRAM LISTING

```
100 GRAPHICS 0
110 PRINT "
120 PRINT "
130 PRINT "
140 PRINT "
150 PRINT "\ 0 0 0 0 /"
160 PRINT " \  "

```

SPACE FLEET

Create a fleet of different kinds of spacecraft.



PROGRAM LISTING

```
100 GRAPHICS 0
110 PRINT "
120 PRINT "
130 PRINT "
140 PRINT "-<->- "
150 PRINT "
160 PRINT "
170 PRINT "
180 PRINT " ^ "
190 PRINT " | | "

```




```

200 PRINT "  | |"
210 PRINT "  / \ "
220 PRINT "      /\ "
230 PRINT "      | |"
240 PRINT "      | |"
250 PRINT "      | |"
260 PRINT "      /\ "
270 PRINT "      @@@@ "
280 PRINT "      /#####\ "
290 PRINT "      |*****| "
300 PRINT "      ^^^^ "

```

NOTE

You can use any of the special characters on the keyboard in a PRINT statement. Experiment with these characters and make some of your own spaceships.

BUGS

Here is a swarm of bugs.
Can you come up with some species of your own?

```

      /\
      ( _ )
      /\

      ! !
      /\
      /\

      +
      \ ( ) /
      / ( _ ) \
      /      \

      ^ " ^
      / \ ( ) / \
      \ /      \ \
      v          v

      \ _ /
      ( . . )
      // /\ \ \

      @#####
      !!!!!!!

```

PROGRAM LISTING

```

100 GRAPHICS 0
120 PRINT "      /\ "
130 PRINT "      ( _ ) "
140 PRINT "      /\ "
150 PRINT "      ! ! "
160 PRINT "      / \ "
170 PRINT "      \ / "
180 PRINT "      + "
190 PRINT "      \ ( ) / "
200 PRINT "      / ( _ ) \ "
210 PRINT "      / \ "

```



```

225 PRINT "
230 PRINT "
240 PRINT "
250 PRINT "
260 PRINT "
270 PRINT "
280 PRINT "
290 PRINT "
300 PRINT "

```

```

      .
     \_/_/"
    (  .  )"
   //  \  \  "

```

```

      ^  "  ^
     /  \  /  \  "
    \  /  \  /  "
     v      v"

```

```

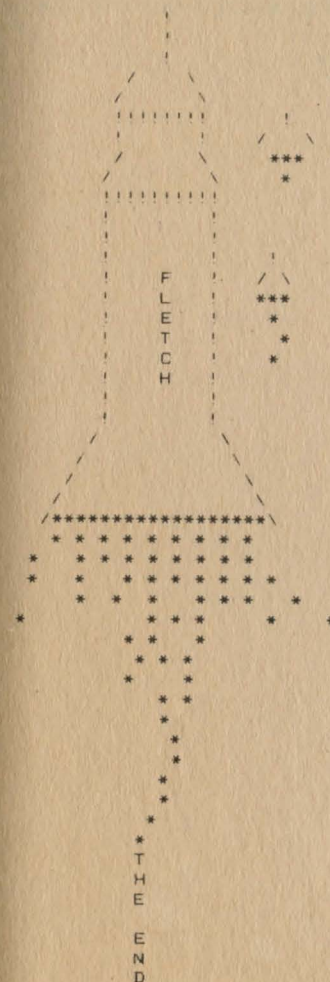
@#####"
!!!!!!!!!"

```

ROCKET SHIP

This rocket is drawn by the computer and then blasts off the screen, leaving a trail of fire.

You can do the same thing with your own spaceships.



PROGRAM LISTING

```

100 GRAPHICS 0
110 WIDTH 80
120 PRINT "      |"
130 PRINT "      |"
140 PRINT "      |"
150 PRINT "    /  \"
160 PRINT "   /    \"
170 PRINT "  /      \"
180 PRINT " /        \"
190 PRINT "/          \"
200 PRINT "/          \"
210 PRINT "/          \"
220 PRINT "/          \"
230 PRINT "/          \"
240 PRINT "/          \"
250 PRINT "/          \"
260 PRINT "/          \"
270 PRINT "/          \"
280 PRINT "/          \"
290 PRINT "/          \"
300 PRINT "/          \"
310 PRINT "/          \"
320 PRINT "/          \"
330 PRINT "/          \"
340 PRINT "/          \"
350 PRINT "/          \"
360 PRINT "/          \"
370 PRINT "/*****\"
380 PRINT "*****\"
390 PRINT "*****\"
400 PRINT "*****\"
410 PRINT "*****\"
420 PRINT "*****\"
430 PRINT "*****\"
440 PRINT "*****\"
450 PRINT "*****\"
460 PRINT "*****\"
470 PRINT "*****\"
480 PRINT "*****\"
490 PRINT "*****\"
500 PRINT "*****\"
510 PRINT "*****\"
520 PRINT "*****\"
530 PRINT "*****\"
540 PRINT "*****\"
550 PRINT "*****\"
560 PRINT "*****\"
570 PRINT "*****\"
580 PRINT "*****\"
590 PRINT "*****\"
600 PRINT "*****\"
610 PRINT "*****\"
620 FOR X=1 TO 40
630 PRINT
640 NEXT X

```

NOTE

Lines 620-640: This is a FOR . . . NEXT loop, which allows you to repeat part of your program as many times as you want. The FOR statement begins the loop and the NEXT statement ends it. In this program, it repeats Line 630, which PRINTs a blank line, 40 times, making the rocket move up the screen so it appears to blast off.

BICYCLE

Now you can create a ten-speed bicycle.
Can you design a racer or dirt bike of your own?

I -----
\ + &c
+-----+-----+-----+-----+-----+-----+
I \ / I
- - - - - -
I - \ / = I
+ I / +
I (X) == I
I / == I
I = I

PROGRAM LISTING

```

100 GRAPHICS 0
110 PRINT "  I---"
120 PRINT "    \          _ _ _ _ _"
130 PRINT "      ^          + 3."
140 PRINT "    -+--\      /  _ _ _ _ _"
150 PRINT "  -  I  - \  -  I  -"
160 PRINT "  -  I  -  \ /  =  I  -"
170 PRINT "    +----- I /  +-----"
180 PRINT "    I      (X)  == I"
190 PRINT "  -  I      /  -== I"
200 PRINT "  -  I  =  -  I  -"

```

2 POSITION

The POSITION statement tells the computer in which column and row to locate the cursor.

PROGRAMS

1. Warmups
2. Face
3. Christmas Tree
4. Candy Jar
5. Tower
6. Grove
7. Hide-and-Seek

WARMUPS

The format of the POSITION statement is:

POSITION column, row where
column is column position, 0 through 39 in GRAPHICS 0
row is row position, 0 through 23 in GRAPHICS 0

The values used in the POSITION statement vary with the screen format of each graphic mode for the Atari. The cursor doesn't actually move until a command which involves the screen is used. If you specify coordinates outside the available format for the mode you've selected, you will get an Error Message that says ERROR 141 (cursor out of range for particular mode) for the line following your POSITION statement which uses the screen, when you try to run your program.

EXERCISE 1

Use POSITION to put a character at various locations on your screen.

```
10 GRAPHICS 0
20 POSITION 19,12
30 PRINT "*"
```

*

This exercise puts an * at location column 19, row 12. Try placing the * at other locations on the screen.

EXERCISE 2

Use the POSITION statement to center text.

```
10 GRAPHICS 0
20 POSITION 9,11
```

```
30 PRINT "THIS SHOULD BE CENTERED"
40 POSITION 12,12
50 PRINT "IN 40 COLUMN TEXT"
60 PRINT "IN 40 COLUMN TEXT"
```

THIS SHOULD BE CENTERED
IN 40 COLUMN TEXT

POSITION 9,11 locates the cursor on column 9, row 11 when the program executes the following PRINT statement. POSITION 12,12 puts it on column 12, row 12 for the next PRINT statement.

FACE

Draw a girl's face and then add features. You can change her expression by altering Lines 210, 230, and 270.

```

  // // // \ \
  ||      ||
  || ^ ^ ||
  || 0 0 ||
  ||      ||
  || ^   ||
  ||  _  ||
  || \___/||
  ||      ||
  
```

PROGRAM LISTING

```

100 GRAPHICS 0
110 PRINT "      // // // \ \
120 PRINT "      ||      ||"
130 PRINT "      ||      ||"
140 PRINT "      ||      ||"
150 PRINT "      ||      ||"
160 PRINT "      ||      ||"
170 PRINT "      ||      ||"
180 PRINT "      || \___/ ||"
190 PRINT "      ||      ||"
200 POSITION 9,2
210 PRINT "^ ^"
220 POSITION 9,3
230 PRINT "0 0"
240 POSITION 10,5
250 PRINT "^"
260 POSITION 10,6
270 PRINT " _ "
280 POSITION 1,15
  
```

NOTES

Line 100: GRAPHICS 0 tells the computer to start the program run in graphics mode 0.

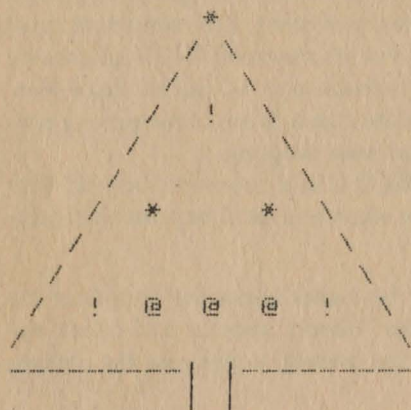
Lines 200, 220, 240, 260: POSITION tells the computer to put the cursor at the column, row position you want. This statement must correspond to the parameters of your screen mode (0-39 columns by 0-23 rows in GRAPHICS 0). Otherwise you will get an Error Message, ERROR 141, which means the cursor is out of range for a particular mode, when you try to run your program.

The default margin in GRAPHICS 0 is in column 2, row 1. You may need to take this into account when you set your position coordinates.

Line 280: POSITION 1,15 puts the cursor below the bottom of the picture, at row 15. Otherwise, the "Ready" prompt and cursor appear immediately below the last line printed, in this case the mouth.

CHRISTMAS TREE

Draw a Christmas tree, then go back and decorate it.
Add ornaments of your own to the program.

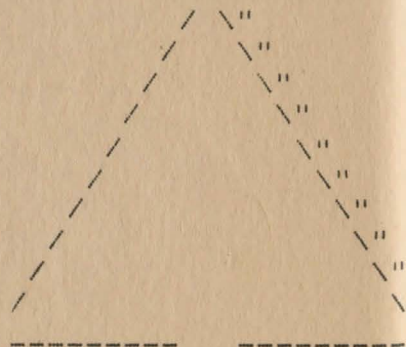


PROGRAM LISTING

```

100 GRAPHICS 0
105 PRINT
106 PRINT
110 PRINT "
120 PRINT "
130 PRINT "
140 PRINT "
150 PRINT "
160 PRINT "
170 PRINT "
180 PRINT "
190 PRINT "
200 PRINT "
210 PRINT "
220 POSITION 20,13
230 PRINT "| |"

```



```

240 POSITION 20,14
250 PRINT "| |"
260 REM *** DECORATE TREE
270 GOSUB 400
280 POSITION 21,1
290 PRINT "x"
300 GOSUB 400
310 POSITION 21,4
320 PRINT "!"
330 GOSUB 400
340 POSITION 18,7
350 PRINT "x"      "x"
360 GOSUB 400
370 POSITION 15,10
380 PRINT "!  @  @  @  !"
390 GOTO 430
400 FOR W=1 TO 400
410 NEXT W
420 RETURN
425 POSITION 1,20
430 END

```

NOTES

Line 270: GOSUB tells the computer to leave the program and go to a subroutine, in this case Line 400.

Line 390: GOTO tells the computer to go to a designated line number, in this case Line 430.

Lines 400-420: Subroutines are small programs within a larger program that are used over and over. When the subroutine is finished, a RETURN statement tells the computer to go back to where it came from in the main program. Here the subroutine is a Wait Loop, which simply makes the computer pause before it continues to decorate the tree.

Line 430: The END statement tells the computer that it is the last line in the program. It stops the Run.

CANDY JAR

This program draws a candy jar and fills it with jelly beans.
Design your own container and fill it with marbles.

```

      | |
      | |
    /   \
|000000|
|000000|
|000000|
|000000|
|-----|

```

PROGRAM LISTING

```

100 GRAPHICS 0
110 POSITION 2,7
120 PRINT "
130 PRINT "      | |"
140 PRINT "      | |"
150 PRINT "    /   \"
160 PRINT "      | |"
170 PRINT "      | |"
180 PRINT "      | |"
190 PRINT "      | |"
200 PRINT "      |-----|"
210 FOR Y=14 TO 11 STEP -1
220 FOR X=7 TO 12
230 POSITION X,Y
240 PRINT "0"
250 NEXT X
260 GOSUB 290
270 NEXT Y
280 GOTO 320
290 FOR T=1 TO 200

```

```

300 NEXT T
310 RETURN
320 POSITION 1,20
330 END

```

NOTE

Lines 210-270: These lines form two nested FOR . . . NEXT loops, to fill the jar from the bottom up. The Y loop sets the rows and the X loop sets the columns to print "0".

Line 210: The loop of lines is incremented by the number in the STEP statement, in this case -1, so the program counts backwards from 14 to 11. If there is no STEP statement, the loop is incremented by one's.

TOWER

Build a tower, using one of Atari's special ATASCII characters.

Can you change the program to build a pyramid, using a different character?



PROGRAM LISTING

```
100 GRAPHICS 0
110 FOR J=20 TO 10 STEP -1
120 FOR I=19 TO 21
130 POSITION I,J
140 PRINT CHR$(21);
150 NEXT I
160 NEXT J
170 POSITION 1,1
```

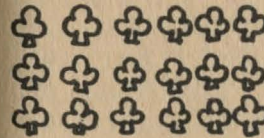
NOTE

Line 140: CHR\$(21) means the ATASCII character numbered 21.
On the Atari this character is a ■.

GROVE

"Grove" draws identical trees in evenly spaced columns and rows on your screen.

Can you change the orchard pattern or design of the trees?



PROGRAM LISTING

```
100 GRAPHICS 0
110 FOR J=5 TO 10 STEP 2
120 FOR I=10 TO 20 STEP 2
130 POSITION I,J
140 PRINT CHR$(16);
150 NEXT I
160 NEXT J
170 POSITION 1,20
```


HIDE-AND-SEEK

Here's a hide-and-seek game you play with the computer. Guess which tree X is hiding behind and the computer will show you if you're right.

```

      ^
     / 1 \
    -|-
      ^
     / 2 \
    -|-X
      ^
     / 3 \
    -|-
      ^
     / 4 \
    -|-
      ^
     / 5 \
    -|-
  
```

```

WHERE IS X HIDING?
BEHIND TREE 1,2,3,4 OR 5? 4
SORRY ... BEHIND TREE 2 !
YOU'RE STILL 'IT' !
WANT TO PLAY AGAIN (Y OR N)? N
  
```

PROGRAM LISTING

```

100 DIM N$(3)
110 GRAPHICS 0
120 A=INT(RND(1)*5+1)
130 FOR L=1 TO 5
140 X=INT(RND(1)*20+10)
150 Y=INT(RND(1)*15+1)
160 POSITION X,Y
  
```

```

170 PRINT " ^"
200 POSITION X,(Y+1)
210 PRINT "/" ; L ; "\"
220 POSITION X,(Y+2)
230 PRINT "-|- "
240 IF A=L THEN 270
250 NEXT L
260 GOTO 290
270 X1=X:Y1=Y
280 GOTO 250
290 POSITION 2,18
300 PRINT "WHERE IS X HIDING?"
310 PRINT "BEHIND TREE 1,2,3,4 OR 5?"
320 INPUT N
330 IF N=A THEN 370
340 PRINT "SORRY...BEHIND TREE";A;"!"
350 PRINT "YOU'RE STILL 'IT'!"
360 GOTO 380
370 PRINT "YOU'RE RIGHT!!!"
380 PRINT "THERE HE IS!"
390 X1=(X1+2):Y1=(Y1+1)
400 POSITION X1,Y1
410 PRINT "X"
420 POSITION 2,22
430 PRINT "WANT TO PLAY AGAIN (Y OR N)? ";
440 INPUT N$
450 IF N$="Y" THEN 110
460 END
  
```

NOTES

Line 100: String variables on the Atari must be dimensioned. DIM N\$(3) saves three spaces for N\$.

Lines 120, 140, 150: RND(1) or RND is a random-number function, which generates random numbers from 0 to .9999. INT is an integer function, which changes decimal numbers to whole numbers by dropping the fractional part of the number. The statement A=INT

(RND(1) * 5 + 1) tells the computer, "Let A be a random number between 1 and 5."

Lines 320, 440: The INPUT programming statement allows you to enter different numbers or words into your program each time it runs.

Lines 330, 450: The IF . . . THEN programming statement makes the computer decide which line number to execute next. IF the condition is true, THEN the program goes to the line number indicated. Otherwise it goes to the following line number.

3

PLOT and DRAWTO

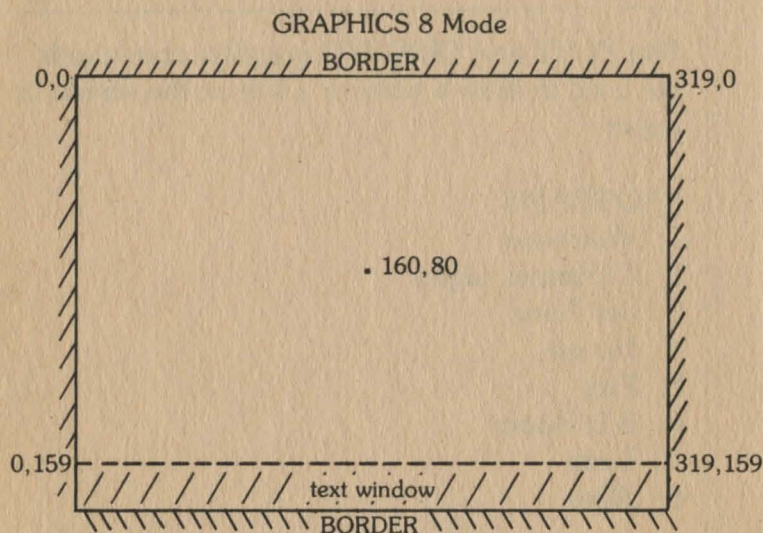
The PLOT and DRAWTO graphics commands are used to draw a point or a line on the screen or paper.

PROGRAMS

1. Warmups
2. Christmas Lights
3. Jet Plane
4. House
5. Flag
6. 3-D Room
7. Train
8. Skier

WARMUPS

Now you are ready to explore the graphics capabilities of your computer. The programs and exercises included in this chapter use the GRAPHICS 8 mode. This mode uses a grid of 310 columns across and 160 rows down, with a four-line text window at the bottom of the screen. If you think of the screen as an X,Y axis, then X corresponds to the columns and Y to the rows. The top left point on the screen is (0,0) the bottom right is (319,159). X increases as you move to the right across the screen, and Y increases as you move down.



The statement GRAPHICS 8 puts your computer into this resolution mode. Once you type this in, you will stay in GRAPHICS 8 until you press RESET or type in another GRAPHICS statement.

When BASIC is first powered up, all the color data is 0. Thus, when graphics statements are executed, nothing seems to happen. Correct this by typing COLOR 1 and hitting RETURN.

To examine your program listing, hit RESET or type GR. 0 to get back to a full-screen text mode. Then type LIST.

PLOT

The PLOT graphics statement plots a point on the screen. The format for the statement is:

PLOT X1,Y1 or PL. X1,Y1 where:

X1,Y1 are the X,Y coordinates of the point on the graphics screen.

You can abbreviate PLOT when typing to PL.

PLOT is most frequently used to determine the point on the screen where a shape begins. Then DRAWTO is used to draw the shape.

DRAWTO

The DRAWTO graphics statement draws a line on the screen. The format for the statement is:

DRAWTO X1,Y1 or DR. X1,Y1 where:

X1,Y1 are the X,Y coordinates of the ending points of the line.

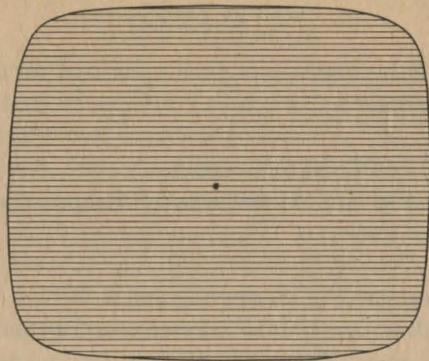
This uses the last point drawn as the starting point and X1,Y1 as the ending point.

DRAWTO is used in all modes on the Atari, and can be abbreviated when typing as DR. X1,Y1.

EXERCISE 1

Use PLOT to place a point in the middle of the screen.

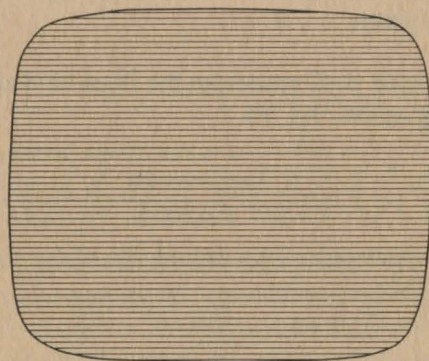
```
GRAPHICS 8
COLOR 1
SETCOLOR 1,1,14
PLOT 160,80
```

This example uses COLOR 1. If no COLOR is specified when you start up the system with BASIC, COLOR is 0, the color of the background, and appears invisible. Set COLOR 1,1,14 makes the point as bright as possible against the background. These statements are explained in detail in Chapter 4.

Now erase this point, using COLOR 0.

```
COLOR 0
PLOT 160,80
```



EXERCISE 2

Draw a line, using PLOT.

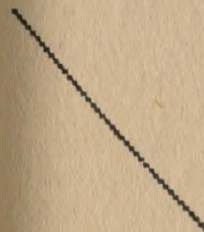
```
10 GRAPHICS 8
20 COLOR 1
30 SETCOLOR 1,1,14
40 FOR X=10 TO 50
50 PLOT X,50
60 NEXT X
```

Notice that you can use a numeric variable as the X or Y coordinate in PLOT.

EXERCISE 3

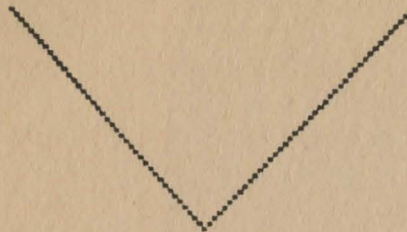
Use the PLOT and DRAWTO statement to draw a line on the screen.

```
NEW
GRAPHICS 8
COLOR 1
SETCOLOR 1,1,14
PLOT 140,100
DRAWTO 180,140
```



Now draw a line from the end of the first line to (220,100).

DRAWTO 220,100

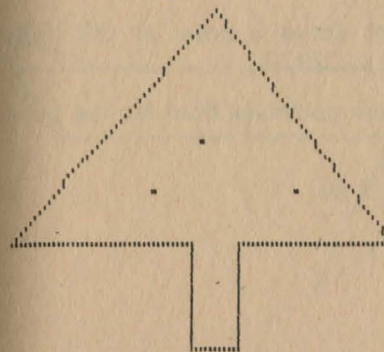


The DRAWTO statement uses the last point drawn on the screen as the starting point.

CHRISTMAS LIGHTS

Use this program to draw a small Christmas tree and put on some lights.

Add some more lights to the tree once you get this program up and running.



PROGRAM LISTING

```
100 GRAPHICS 8
105 COLOR 1:SETCOLOR 1,1,14
110 PLOT 80,100:DRAWTO 123,55
120 DRAWTO 160,100
130 PLOT 80,100:DRAWTO 118,100
140 DRAWTO 118,120
150 DRAWTO 128,120
160 DRAWTO 128,100
170 DRAWTO 160,100
180 PLOT 120,80
190 PLOT 110,90
200 PLOT 140,90
```


NOTES

Line 100: GRAPHICS 8 puts the computer into graphics mode 8.

Line 105: COLOR 1 makes the graphics points visible in GR. 8. SETCOLOR 1,1,14 makes them as bright as possible against the background. These statements are explained in detail in the following chapter.

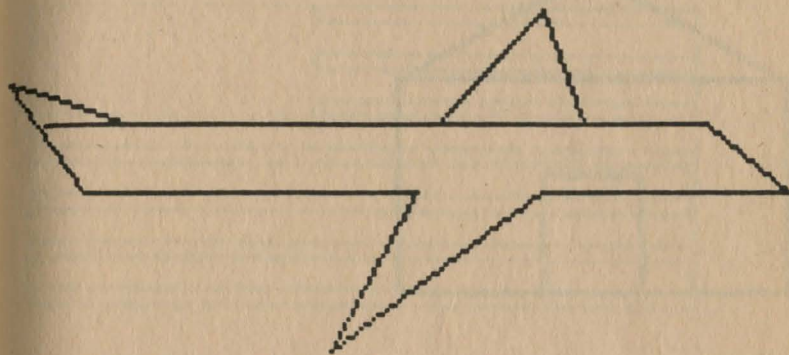
Line 110: The PLOT statement draws a point at (80,100). DRAWTO draws a line from there to 123,55.

Line 120: This DRAWTO statement continues from the last point drawn to 160,100.

Line 180: PLOT puts a point at 120,80.

JET PLANE

The computer can draw a jet liner with this program.
Can you improve upon the design?



PROGRAM LISTING

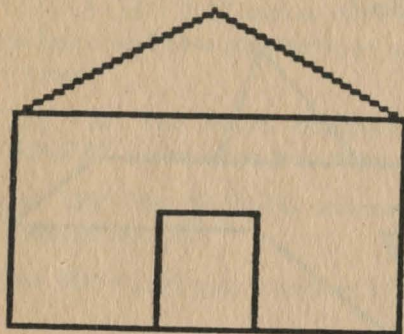
```
100 GRAPHICS 8:COLOR 1:SETCOLOR 1,1,14
110 PLOT 80,100:DRAWTO 240,100
120 DRAWTO 260,115
130 DRAWTO 200,115
140 DRAWTO 150,150
150 DRAWTO 170,115
160 DRAWTO 90,115
170 DRAWTO 70,90
180 DRAWTO 100,100
190 PLOT 175,100:DRAWTO 200,75
200 DRAWTO 210,100
```

NOTE

Lines 120-180: Notice that these DRAWTO statements all use the prior point as the starting point.

HOUSE

Here is a small house. See if you can add windows or a chimney.

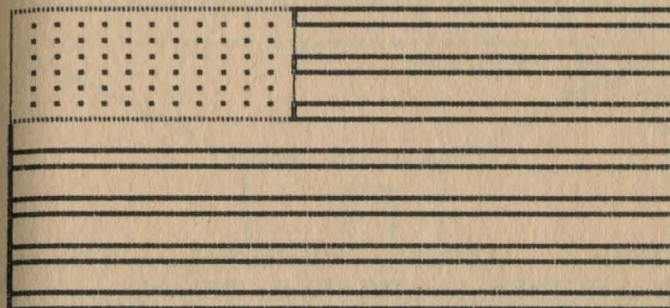


PROGRAM LISTING

```
100 GRAPHICS 8:COLOR 1:SETCOLOR 1,1,14
110 PLOT 80,100:DRAWTO 160,100
120 DRAWTO 160,140:DRAWTO 80,140
130 DRAWTO 80,100:DRAWTO 120,80
140 DRAWTO 160,100
150 PLOT 110,118:DRAWTO 130,118
160 PLOT 110,118:DRAWTO 110,140
170 PLOT 130,118:DRAWTO 130,140
```

FLAG

This program draws a flag with 66 stars and 13 stripes.
Can you change it to draw the America flag?



PROGRAM LISTING

```
100 GRAPHICS 8:COLOR 1
105 SETCOLOR 1,1,14
110 PLOT 100,100:DRAWTO 240,100
115 DRAWTO 240,157
117 DRAWTO 100,157:DRAWTO 100,100
120 PLOT 160,100:DRAWTO 160,121
125 DRAWTO 100,121
130 PLOT 160,100:DRAWTO 240,100
135 PLOT 160,103:DRAWTO 240,103
140 PLOT 160,109:DRAWTO 240,109
145 PLOT 160,112:DRAWTO 240,112
150 PLOT 160,118:DRAWTO 240,118
155 PLOT 160,121:DRAWTO 240,121
160 PLOT 100,127:DRAWTO 240,127
165 PLOT 100,130:DRAWTO 240,130
170 PLOT 100,136:DRAWTO 240,136
175 PLOT 100,139:DRAWTO 240,139
180 PLOT 100,145:DRAWTO 240,145
185 PLOT 100,148:DRAWTO 240,148
```



```

190 PLOT 100,154:DRAWTO 240,154
195 PLOT 100,157:DRAWTO 240,157
200 Y=0
210 FOR X=5 TO 55 STEP 5
220 PLOT 100+X,103+Y
230 NEXT X
240 Y=Y+3
250 IF Y>18 THEN 270
260 GOTO 210
270 END

```

NOTES

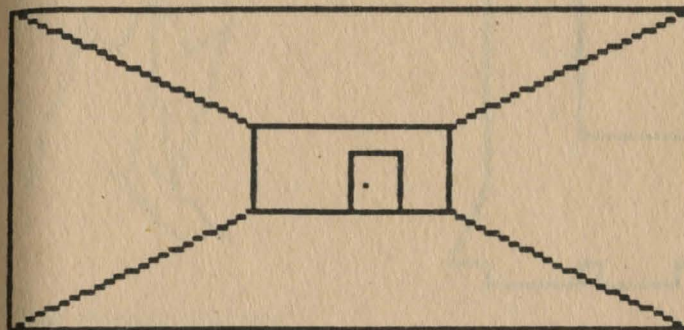
Line 250: This IF . . . THEN statement tells the computer, IF the row is greater than 18, THEN go to LINE 270, the END of the program. Otherwise, go to the next line, which sends the program back to print another row of stars.

Line 270: The END statement terminates the execution of the program and returns control to the user.

3-D ROOM

The computer draws a 3-dimensional room with a door in the far wall.

Change the program so the far wall seems even farther away.



PROGRAM LISTING

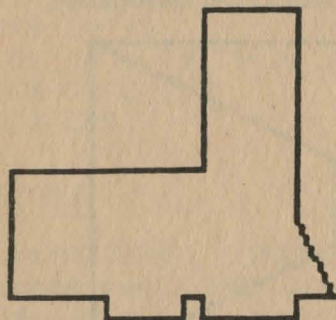
```

100 GRAPHICS 8:COLOR 1:SETCOLOR 1,1,14
110 PLOT 80,100:DRAWTO 220,100:DRAWTO 220,159
115 DRAWTO 80,159:DRAWTO 80,100
120 PLOT 130,122:DRAWTO 170,122
125 DRAWTO 170,138:DRAWTO 130,138
127 DRAWTO 130,120
130 PLOT 80,100:DRAWTO 130,122
140 PLOT 80,160:DRAWTO 130,138
150 PLOT 220,100:DRAWTO 170,122
160 PLOT 220,160:DRAWTO 170,138
170 PLOT 150,127:DRAWTO 160,127
172 DRAWTO 160,138
175 DRAWTO 150,138:DRAWTO 150,127
180 PLOT 153,133

```


TRAIN

Here is a steam engine ready to chug down the tracks.
Try designing a moden locomotive or a race car.

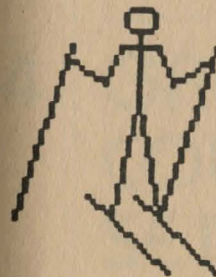


PROGRAM LISTING

```
100 GRAPHICS 8:COLOR 1:SETCOLOR 1,1,14
110 PLOT 0,46:DRAWTO 40,46
120 DRAWTO 40,16
130 DRAWTO 60,16
140 DRAWTO 60,56
150 DRAWTO 68,70
160 DRAWTO 60,70
170 DRAWTO 60,74
180 DRAWTO 40,74
190 DRAWTO 40,70
200 DRAWTO 36,70
210 DRAWTO 36,74
220 DRAWTO 20,74
230 DRAWTO 20,70
240 DRAWTO 0,70
250 DRAWTO 0,46
```

SKIER

This skier is set to take off down the slopes.
See if you can design a person on a sled to substitute for the skier.



PROGRAM LISTING

```
100 GRAPHICS 8:COLOR 1:SETCOLOR 1,1,14
102 REM ***DRAW HEAD
104 PLOT 150,62:DRAWTO 151,63
106 DRAWTO 152,64
108 DRAWTO 153,65:DRAWTO 152,66
110 DRAWTO 151,67
112 DRAWTO 150,68:DRAWTO 149,67
114 DRAWTO 148,66
116 DRAWTO 147,65:DRAWTO 148,64
117 DRAWTO 149,63
118 DRAWTO 150,62
120 REM ***DRAW BODY
125 PLOT 150,65:DRAWTO 150,78
130 DRAWTO 145,100
140 PLOT 150,78
150 DRAWTO 155,100
160 PLOT 150,68
170 DRAWTO 146,68
180 DRAWTO 143,75
```



```
190 DRAWTO 135,70
200 PLOT 136,68
210 DRAWTO 125,100
220 PLOT 150,68
230 DRAWTO 154,68
240 DRAWTO 157,75
250 DRAWTO 165,70
260 PLOT 166,68
270 DRAWTO 155,100
280 PLOT 140,96
290 DRAWTO 157,110
300 PLOT 150,96
310 DRAWTO 167,110
```

4 GRAPHICS MODES, COLOR, and SETCOLOR

Different GRAPHICS, COLOR, and SETCOLOR statements produce unique shapes, colors, and pictures on your Atari screen.

PROGRAMS

1. Warmups
2. Fish
3. Car
4. Fishing Boat
5. Windmill
6. Snowflakes
7. Sailboat
8. Campground
9. Picture

WARMUPS

The Atari has 128 possible color-luminance combinations with which to alter your screen display. Depending upon the graphics mode you select, not more than 5 colors can be displayed at any one time. Each graphics mode has its own set of default colors. In text modes GR. 0, 1, and 2, a different set of colors is enabled by the SETCOLOR statement. In graphics modes GR. 3 through GR. 8, the COLOR and SETCOLOR statements together set the color of the points to be plotted.

No SETCOLOR statement is required in modes GR. 0 through GR. 8 if the default set of colors is used. However, COLOR is required in modes GR. 3 through GR. 7 for any graphics statement to appear on the screen, as all the color data is set to 0 when BASIC is first powered up.

In GR. 0, GR. 1, and GR. 2, Atari's text modes, the COLOR statement determines the ATASCII character to be displayed by PLOT and DRAWTO statements.

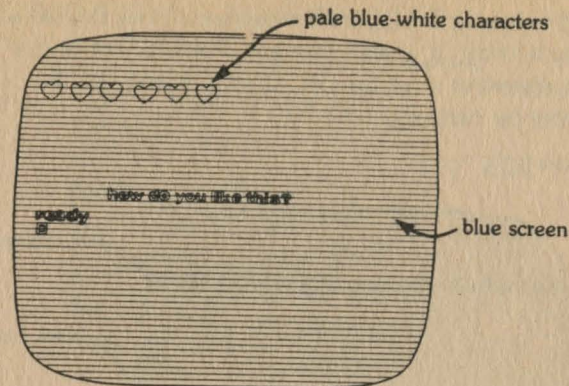
The format is:

COLOR X or C. X where:

X is the numeric variable for an ATASCII character stored in that location for that graphics mode.

EXERCISE 1

This program prints a row of hearts across the top of the screen.

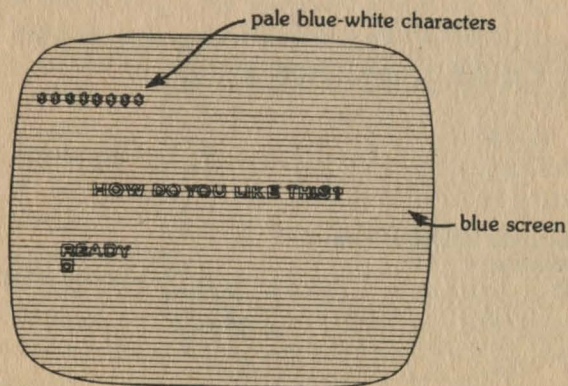


```
10 GRAPHICS 0
20 COLOR 0
30 PLOT 2,1:DRAWTO 17,1
40 POSITION 10,12
50 PRINT "HOW DO YOU LIKE THIS?"
```

In Line 20, COLOR 0 tells the computer to use the ATASCII character with the value of 0 in GR. 0, a heart, for the following PLOT and DRAWTO statement.

To get a row of \$ characters, change Line 20:

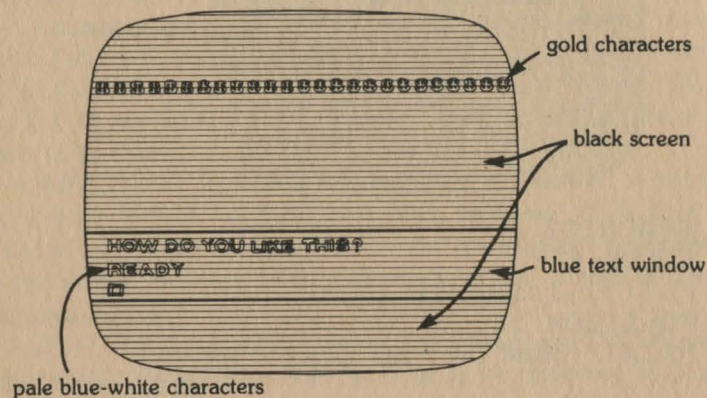
```
20 COLOR 36
```



Try other values in the COLOR statement. Your BASIC reference manual has a table of Atari's internal character set and a character/color assignment chart for GR. 0, GR. 1 and GR. 2.

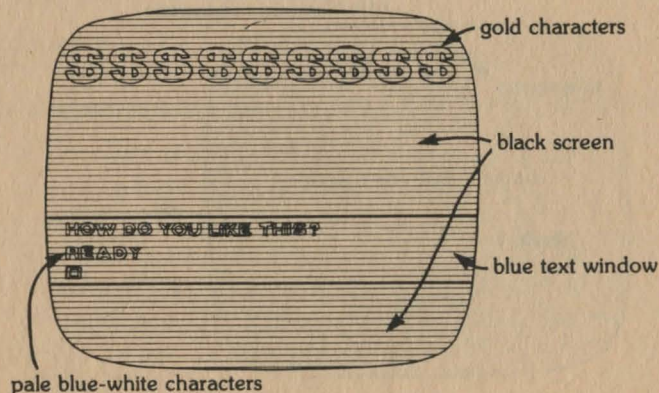
Experiment by changing Line 10:

10 GRAPHICS 1

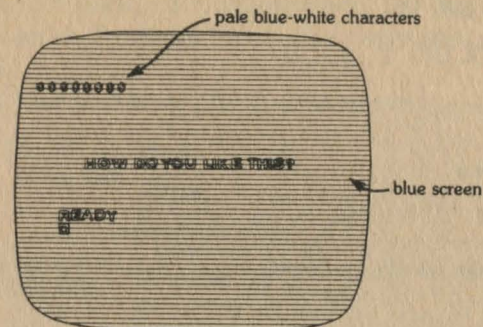


Watch the text window for these modes print at the bottom of the screen. Note how the size and shape of the characters change on the screen according to each screen mode's dimensions.

10 GRAPHICS 2



10 GRAPHICS 0



EXERCISE 2

The SETCOLOR statement determines the colors you can use in GR. 0, GR. 1, and GR. 2.

The format in GRAPHICS 0 is:

SETCOLOR X, Y, Z or SE. X, Y, Z where:

X is the color register:

When X = 1, the statement affects the luminance of the characters that are the same color as the background.

When X = 2, the statement affects the background.

When X = 4, the statement affects the border.

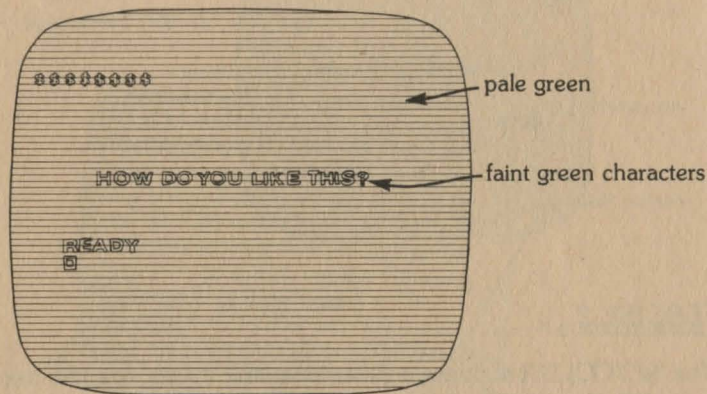
Y is a color variable from 0 to 15:

- | | |
|-----------------|-------------------|
| 0 - gray | 8 - blue |
| 1 - gold | 9 - light blue |
| 2 - orange | 10 - turquoise |
| 3 - red-orange | 11 - green-blue |
| 4 - pink | 12 - green |
| 5 - purple-blue | 13 - yellow-green |
| 6 - red | 14 - orange-green |
| 7 - blue | 15 - light orange |

Z is the luminance variable of the color, from 0 to 14 in even increments, where 0 is the duller and 14 is the brightest.

Let's add a background color of green to our previous program, Exercise 1, in GR. 0.

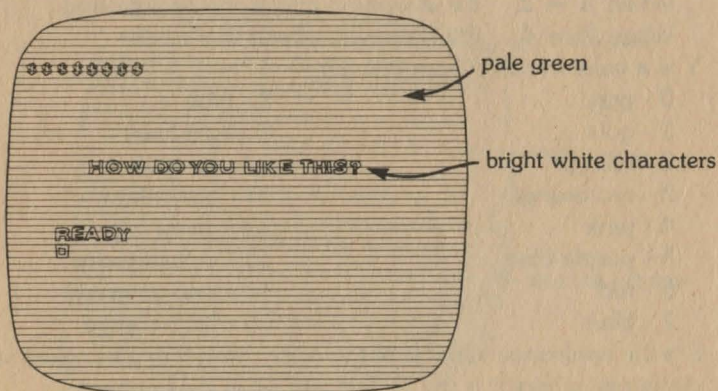
```
15 SETCOLOR 2,12,8
```



In Line 15 the SETCOLOR statement tells the computer to fill color register 2, which effects GR. 0 mode's background, with color 12 (green), with average luminance (8).

Now make the characters as bright as possible to stand out against the background. Add:

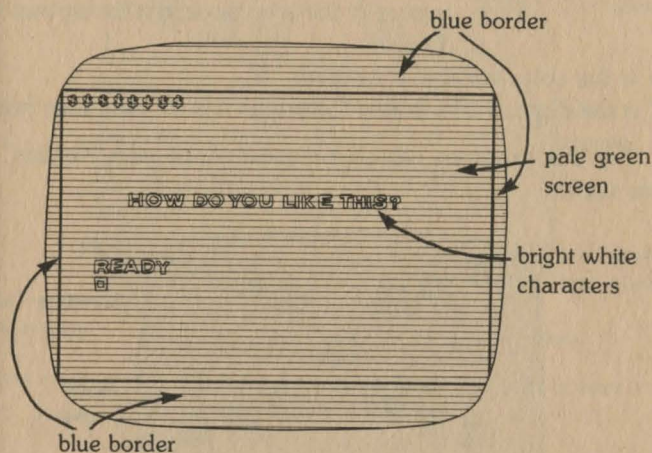
```
16 SETCOLOR 1,0,14
```



This SETCOLOR statement tells the computer to change the character luminance in register 1 to the brightest, 14. The color used is automatically set to the background color, regardless of the color-variable value.

Now add a border color of blue to your screen.

```
17 SETCOLOR 4,8,8
```



Here SETCOLOR tells the computer to fill color register 4, the border, in GRAPHICS 0, with blue (8), in average luminance (8).

Finally, experiment with other background and border colors and luminances in GRAPHICS 0 by changing the color and luminance variables in Line 15 and Line 17.

EXERCISE 3

In GR. 1 and GR. 2, the COLOR statement functions in the same way as in GR. 0. However, the color registers of the SETCOLOR statement affect different parts of the screen than in GR. 0.

In GR. 1 and GR. 2 the format is:

SETCOLOR X, Y, Z or SE. X, Y, Z where:

X is the color register.

When X = 0 the statement affects the color and luminance of the characters on the graphics screen.

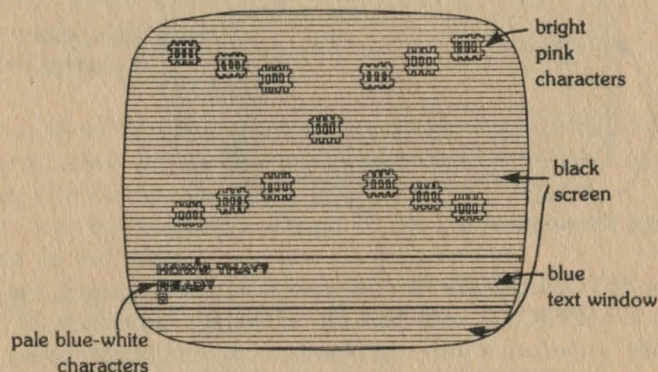
When X = 2 the statement affects the color and luminance of the background of the text window.

When X = 4 the statement affects the color and luminance of the graphic screen background and border.

Y is the color variable 0 through 15.

Z is the luminance variable 0 through 14, in even increments.

Try this program, which prints an X of pink "fences" (the number sign).



```
10 GRAPHICS 1
20 COLOR 35
30 SETCOLOR 0,4,14
40 PLOT 2,2:DRAWTO 18,8
50 PLOT 2,8:DRAWTO 18,2
60 PRINT "HOW'S THAT?"
```

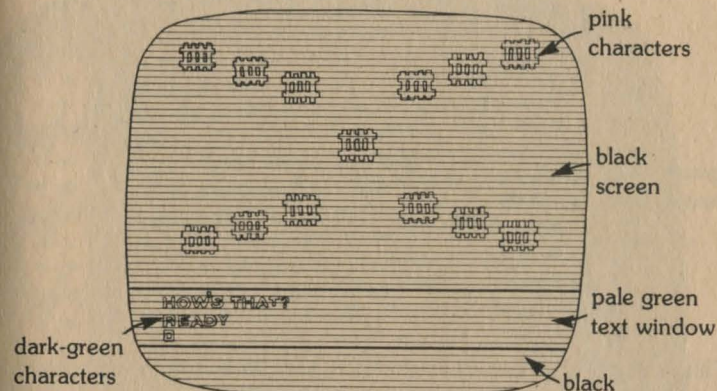
In Line 20, COLOR 35 tells the computer to use the ATASCII character 35 (# in GR. 1 and GR. 2) in the two following graphics statements, Lines 40 and 50.

In Line 30, SETCOLOR 0,4,14 tells the computer to fill the color

register which displays the graphics screen characters, 0, with bright pink.

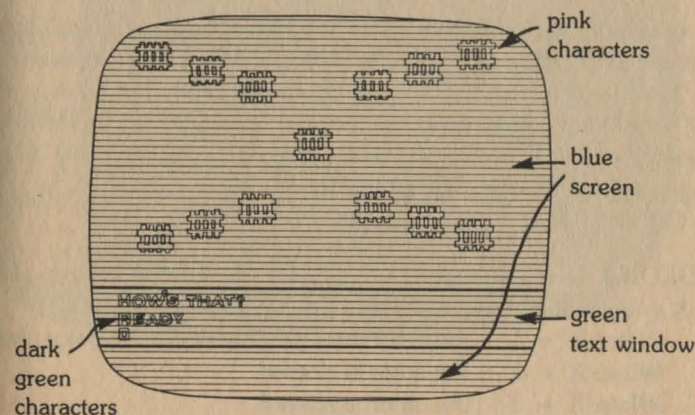
To change the text window's background and characters to green, add:

```
35 SETCOLOR 2,12,14
```

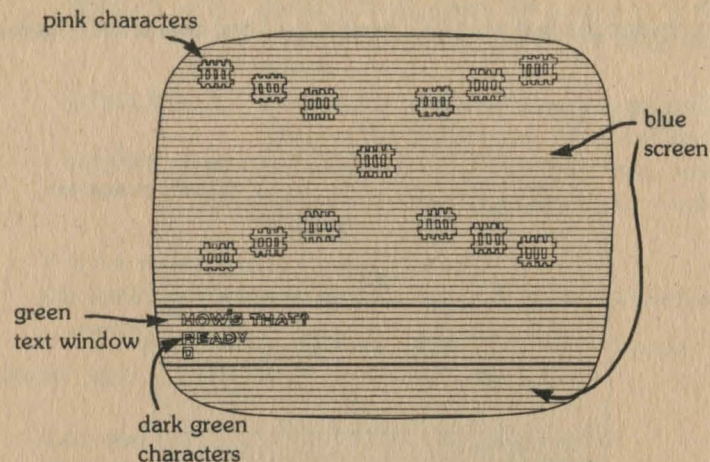


Now change the graphics screen background and border to blue.

```
37 SETCOLOR 4,8,12
```



To watch the dimensions of your characters change, alter Line 10.



Take the time to experiment with color and luminance combinations you like by changing those variables in Lines 30, 35, and 37.

EXERCISE 4

In graphics modes 3 through 8, the values used in the COLOR statement affect the color variable in the SETCOLOR statement. In GR. 3, 5 and 7, COLOR 1 enables the color variable in the SETCOLOR statement. COLOR 2 and COLOR 3 alter the color result and COLOR 0 makes the plotted points the same color as the background, so they appear invisible.

In GR. 3, 5, and 7 the format is:

COLOR X or C. X where:

X enables the color variable of SETCOLOR.

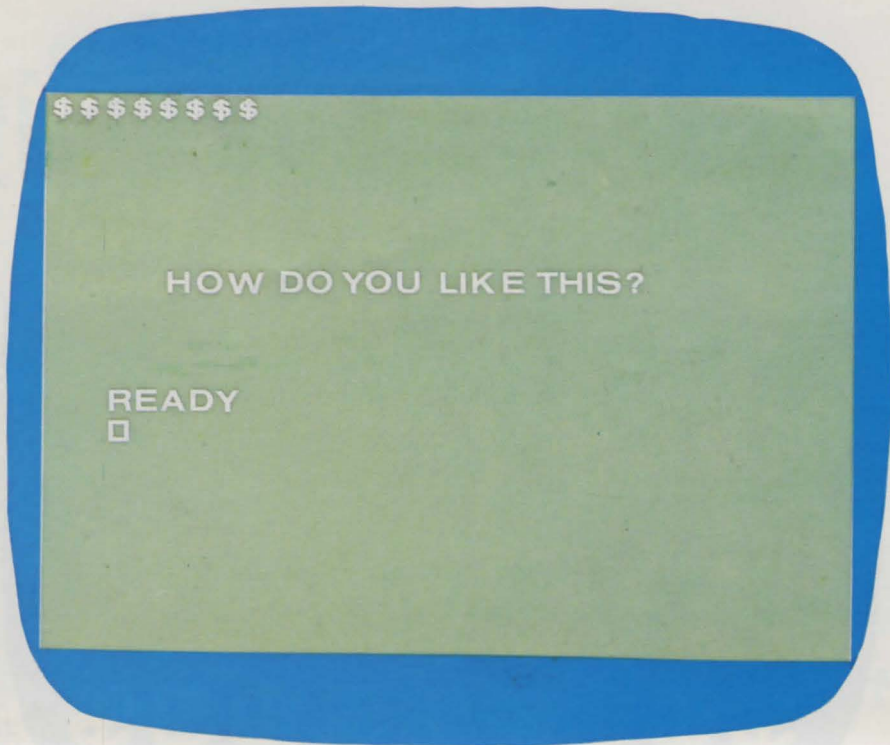
When X = 1 the color variable adheres to the color chart.

When X = 2 the color is altered.

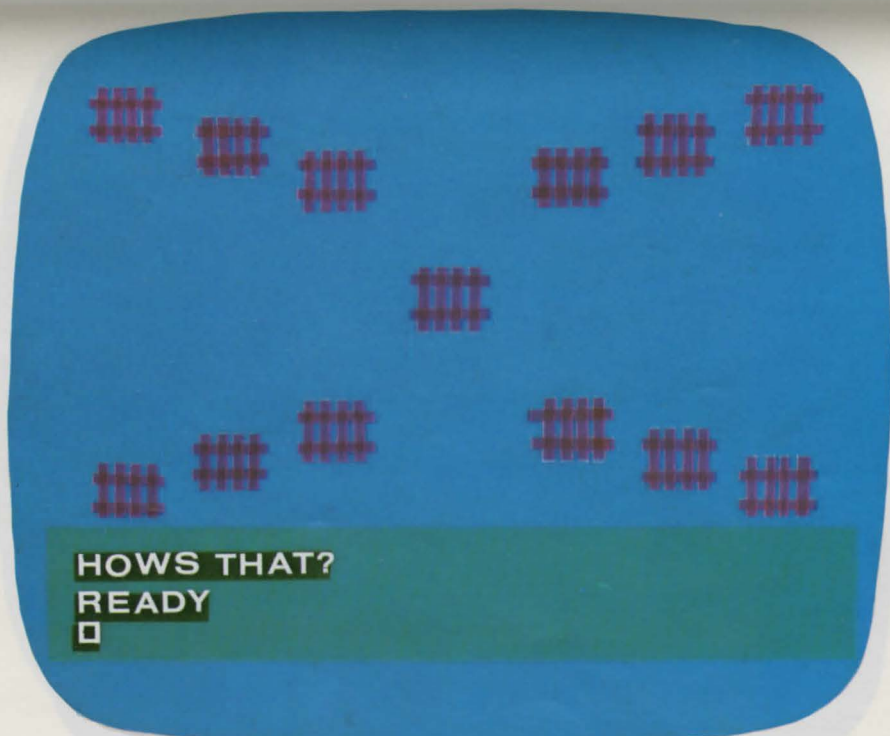
When X = 3 the color is altered.

When X = 0 the color is the same color as the background.





EXERCISE 2 from page 60



EXERCISE 3 from page 63



EXERCISE 4 *from page 68*



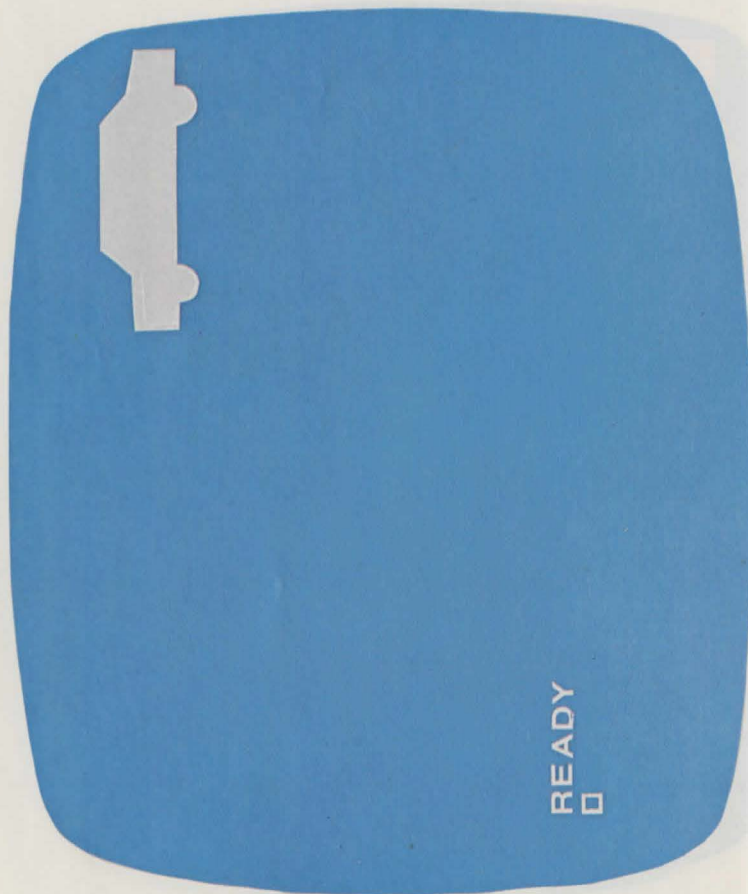
EXERCISE 5 *from page 69*



EXERCISE 5 *from page 72*



EXERCISE 6 *from page 75*



CAR from page 77

Using COLOR 1 in GR. 3, 5, or 7 affects the SETCOLOR statement as follows:

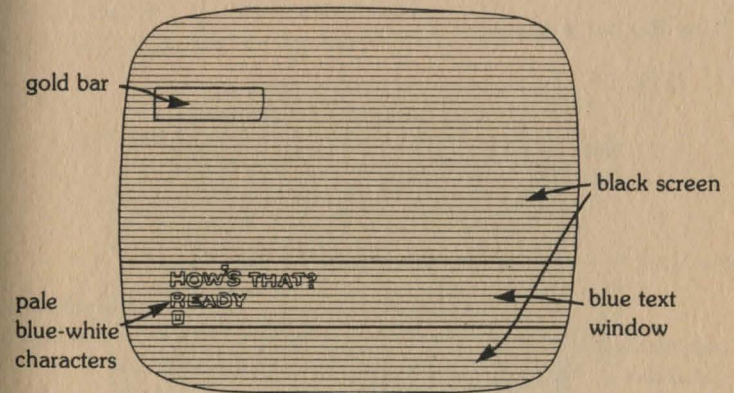
SETCOLOR X, Y, Z or SE. X, Y, Z where:

X is the color register.

- When X = 0 the graphics on the graphics screen are color Y, luminance Z.
- When X = 1 the colors are altered.
- When X = 2 the text window background and characters are color Y, luminance Z.
- When X = 3 the default colors are used.
- When X = 4 the graphics screen background and border are color Y, luminance Z.

Our programs use COLOR 1 and SETCOLOR statements with a color register variable of 0, 2, or 4—altering graphics characters, text window or graphic screen and border background and luminance.

This program demonstrates some of your color options:

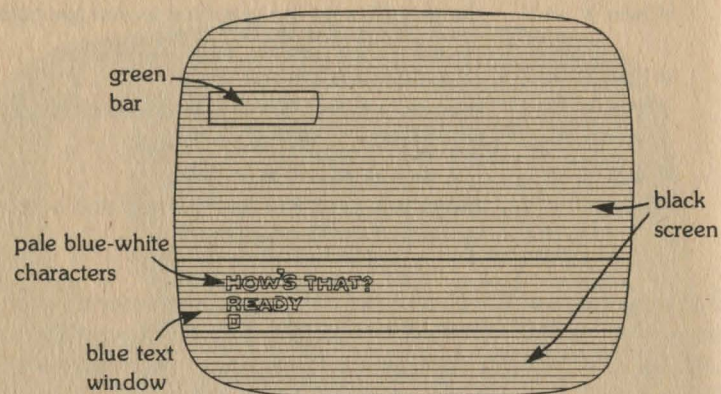


```
10 GRAPHICS 3
20 COLOR 1
30 PLOT 0,10:DRAWTO 20,10
40 PRINT "HOW'S THAT?"
```


A gold bar appears on the black graphics screen, with the print message in a blue text window.

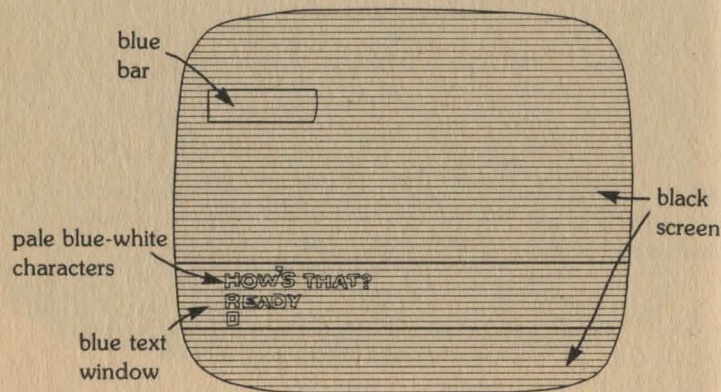
Change Line 20 and watch the results.

```
20 COLOR 2
```



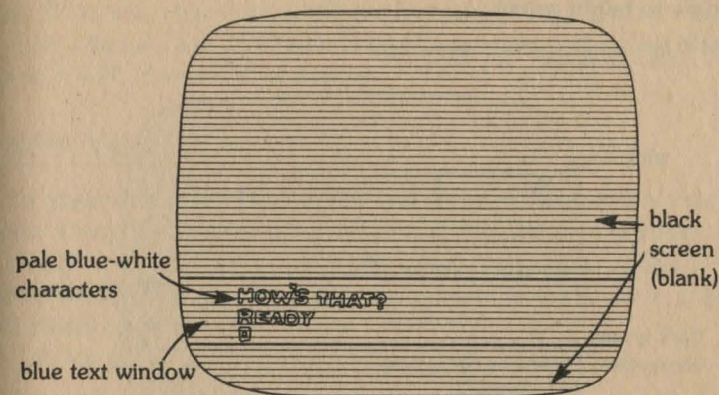
Now the bar is green.

```
20 COLOR 3
```



Now blue.

```
20 COLOR 0
```

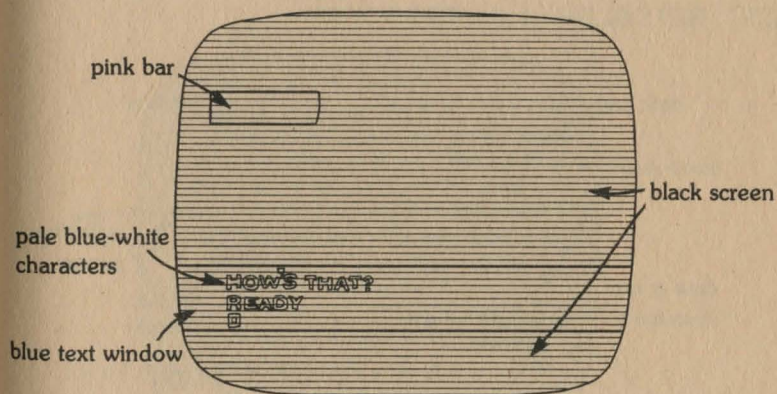


Now it is invisible, as it is the background color.

Make these changes to alter the bar's color according to the Atari hue color chart.

```
20 COLOR 1
```

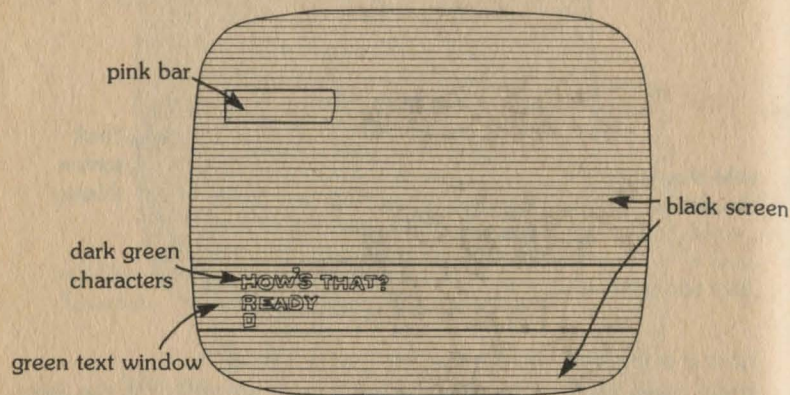
```
22 SETCOLOR 0,4,14
```



SETCOLOR tells the computer to put bright pink into register 0, which displays the graphics statements in GR. 3, 5, and 7. The bar is pink.

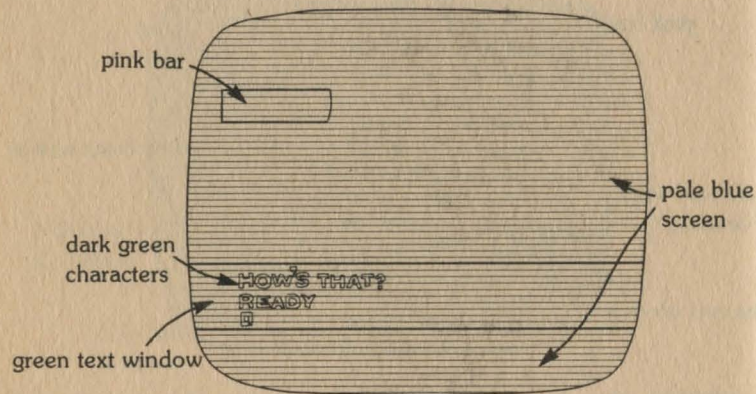
Add this to change the background and character color in the text window to bright green.

```
26 SETCOLOR 2,12,14
```



This SETCOLOR statement changes the graphics background and border to the color and luminance you specify, but also affects the hue of your graphics statements.

```
27 SETCOLOR 4,8,14
```



This makes the background and border blue, but also changes the bar to orange.

Now change the color and luminance variables in Lines 25, 26, and 27 to see what color combinations you prefer.

Also change Line 10 to see how your graphics look in the dimensions of GR. 5 and GR. 7 modes.

EXERCISE 5

In graphics modes 4 and 6 the SETCOLOR statement affects the same screen parameters as in GR. 3, 5 and 7, but only functions with COLOR 1 or COLOR 0. COLOR 0 sets the graphics point to the background color, so the points seem invisible. COLOR 1 enables the points to be seen.

The format in GR. 4 or GR. 6 is:

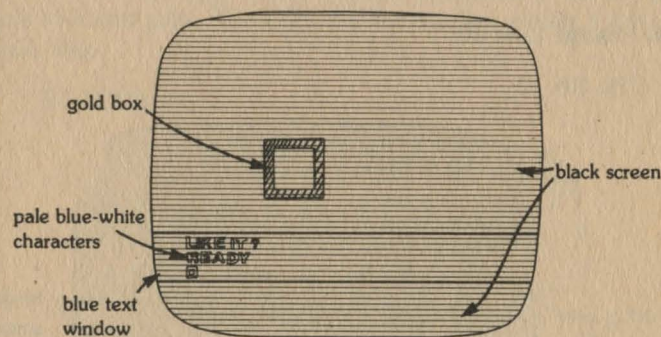
COLOR X or C. X where:

X enables color in the SETCOLOR statement.

When X = 0 the graphic points are the same color as the background and appear invisible.

When X = 1 the graphic points are visible.

This program displays a box in GRAPHICS 4's default colors.

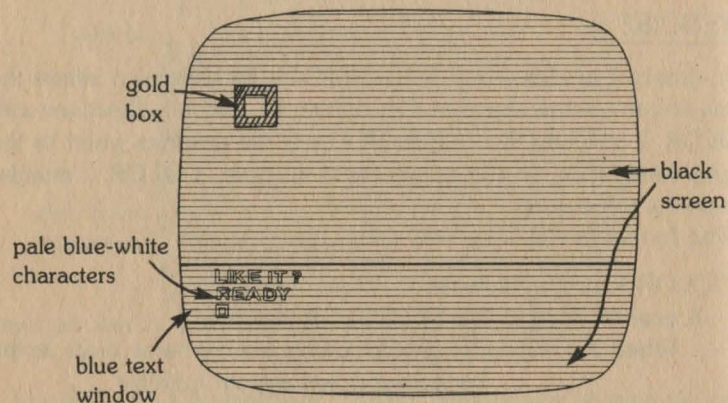


```
10 GRAPHICS 4
20 COLOR 1
30 PLOT 20,20:DRAWTO 30,20
40 DRAWTO 30,30:DRAWTO 20,30
50 DRAWTO 20,20
60 PRINT "LIKE IT?"
```


The box is outlined in orange on a black graphics screen background. The printed message appears in the blue text window.

Now change Line 10.

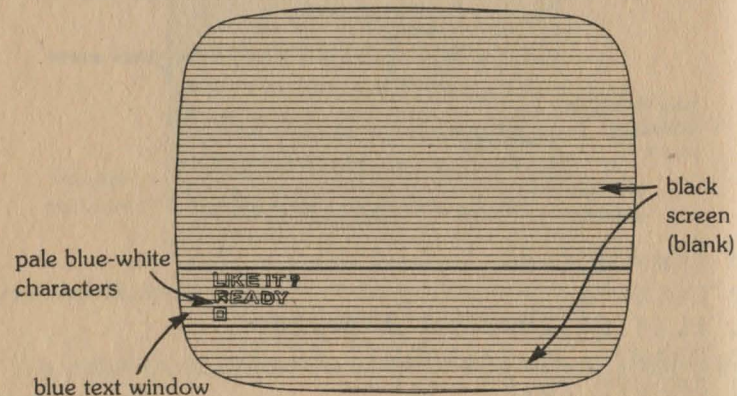
```
10  GRAPHICS 6
```



You should see the same program in GR. 6's screen mode parameters.

Now change Line 20.

```
20  COLOR 0
```

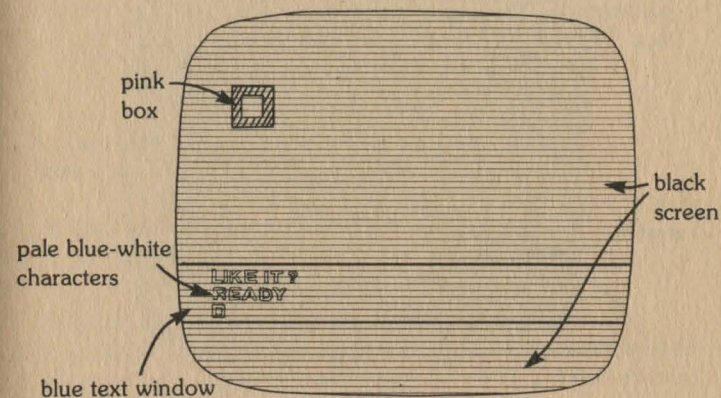


The graphic image disappears, as it is displayed in the background color, black.

Now make these changes and additions:

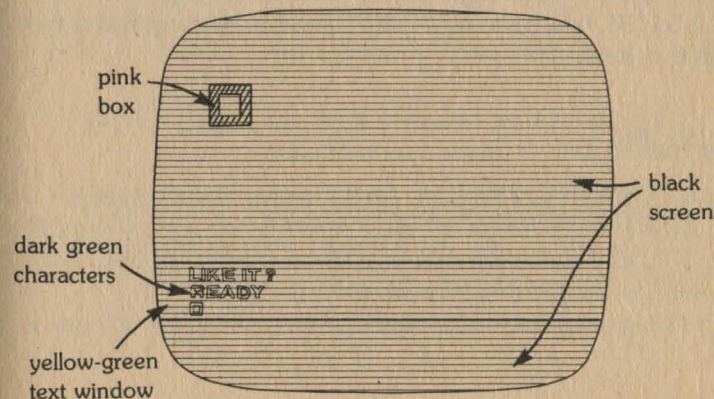
```
15  COLOR 1
```

```
20  SETCOLOR 0,4,14
```



The SETCOLOR statement tells the computer to set the graphic points to bright pink. The COLOR 1 statement makes them visible. Add this:

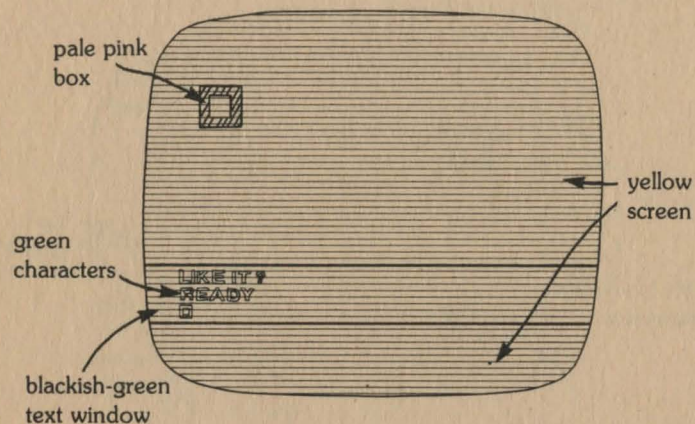
```
26  SETCOLOR 2,1,14
```



This SETCOLOR statement changes the text window background and characters to green.

Now add:

```
27 SETCOLOR 4,2,14
```



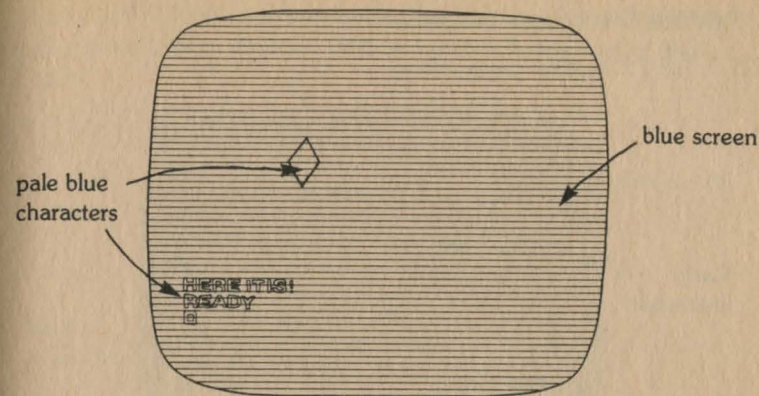
In Line 27, SETCOLOR changes color register 4, the graphics' screen background and border, to bright orange—which in turn affects the hue of the graphics points.

Experiment with the color and luminance variable in Lines 25, 26, and 27 to see what combinations you like. Change Line 10 from GR. 6 to GR. 4 and see how the dimensions of the graphics modes affect the result.

EXERCISE 6

In the GRAPHICS 8 mode, Atari's highest screen resolution, the COLOR statement serves the same function as in GR. 4 and GR. 6. The COLOR 1 statement enables a visible graphics display. COLOR 0 makes the plotted points appear invisible.

This program prints a faint diamond near the center of the screen.



```
10 GRAPHICS 8
20 COLOR 1
30 PLOT 100,80:DRAWTO 120,100
40 DRAWTO 140,80:DRAWTO 120,60
50 DRAWTO 100,80
60 PRINT "HERE IT IS!"
```

The graphics screen and text window in GRAPHICS 8 share the same background color. Here they are in blue, the default color.

In GRAPHICS 8, the SETCOLOR format is:

SETCOLOR X, Y, Z or SE. X,Y,Z

X is the color register.

When X = 1 the luminance of the plotted points is affected.

When X = 2 the graphic screen, points, text windows, and characters are set to color Y, luminance Z.

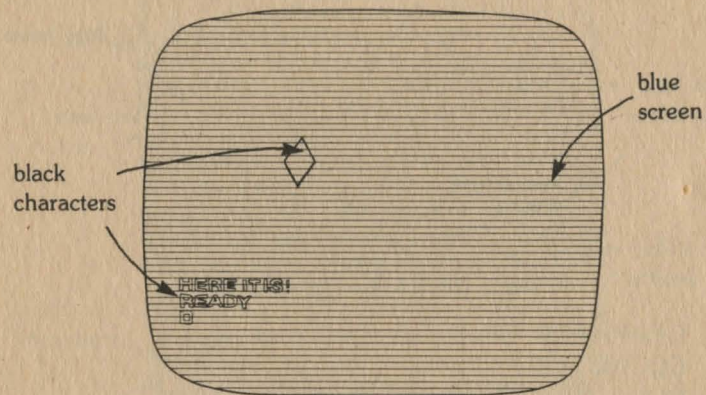
When X = 4 the border is set to color Y, luminance Z.

Y is the color variable, from 0 to 14.

Z is the luminance variable, from 0 to 14 in even increments.

Add this line to your program:

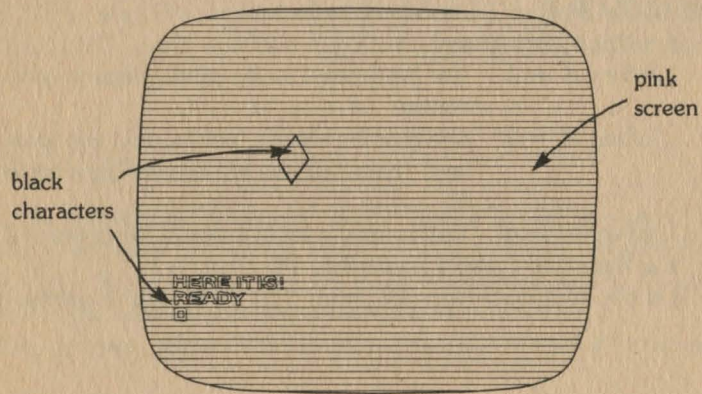
```
25 SETCOLOR 1,8,0
```



In GRAPHICS 8 the SETCOLOR statement affecting color register 1, regardless of the color variable, changes the luminance of the graphics points and the text characters. Here it makes them as dark as possible, to contrast with the background.

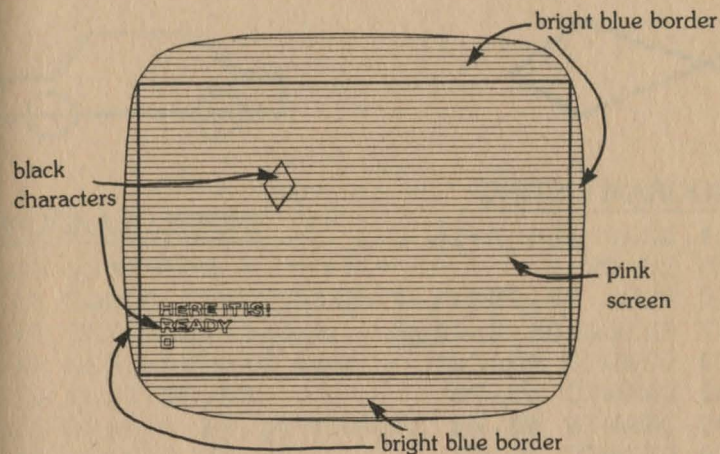
Now add this:

```
26 SETCOLOR 2,4,14
```



This SETCOLOR statement sets the graphic screen background, graphic points, text window, and text characters to bright pink. Add:

```
27 SETCOLOR 4,8,8
```



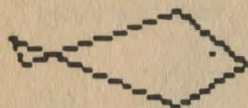
This last SETCOLOR statement fills the border, color register 4, with blue.

Try different color and luminance variables in Lines 26 and 27 to find the combinations that appeal to you.

FISH

Use this program to make a fish.

Can you change the size of this fish and add others to the picture?



PROGRAM LISTING

```
100 GRAPHICS 7+16
105 COLOR 1
110 PLOT 32,75
117 SETCOLOR 0,11,8
120 DRAWTO 48,70
130 DRAWTO 55,75
140 DRAWTO 48,79
150 DRAWTO 31,72
160 DRAWTO 32,75
170 PLOT 51,74
180 GOTO 180
```

NOTES

Line 100: GRAPHICS 7 + 16 tells the computer to use the full screen of graphics mode 7 to run this program.

Line 105: COLOR 1 says use COLOR 1 to enable SETCOLOR statements.

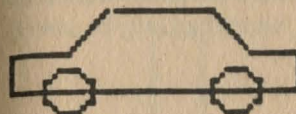
Line 117: SETCOLOR says fill the color register 0, the graphics points on the screen, with color 11, blue-green, with a luminance of 8 average brightness.

Line 180: GOTO tells the computer to keep the program in a continuous loop, enabling graphic mode 7's full screen, until you BREAK the run.

CAR

Here is a car ready to drive across the screen.

Try to design your favorite sports car or a racing model.



PROGRAM LISTING

```
100 GRAPHICS 7
105 COLOR 1:SETCOLOR 0,2,14
107 SETCOLOR 4,8,8
109 REM ***DRAW CAR
110 PLOT 50,30
120 DRAWTO 60,30:DRAWTO 65,25
130 DRAWTO 85,25
140 DRAWTO 90,30
150 DRAWTO 98,30
160 DRAWTO 98,35
170 DRAWTO 50,35
180 DRAWTO 50,30
185 REM ***DRAW TIRES
190 PLOT 56,35:DRAWTO 60,32
195 DRAWTO 64,35
200 DRAWTO 59,38:DRAWTO 56,35
210 PLOT 86,35:DRAWTO 90,32
220 DRAWTO 94,35
225 DRAWTO 89,38:DRAWTO 86,35
```

NOTE

Line 100: GRAPHICS 7 puts the program run in the regular graphics 7 mode, with a text window.

Line 105: SETCOLOR sets the color and luminance of the graphic points.

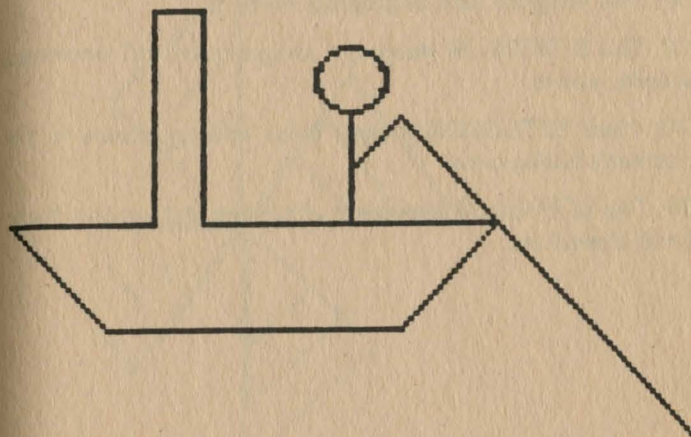
Line 107: Here SETCOLOR sets the color and luminance of the graphic screen's background.

Lines 109, 185: The Remark (REM) statement includes nonexecutable statements in the body of the program, which can only be seen in program listing, for the benefit of the user.

FISHING BOAT

This fisherman is hoping a fish will come along.

See if you can change the program to make his boat look like a cabin cruiser, or add some fish to the picture.



PROGRAM LISTING

```
100 GRAPHICS 5
110 COLOR 1:SETCOLOR 0,2,14
120 SETCOLOR 4,8,8
130 SETCOLOR 2,13,14
135 REM **DRAW BOAT
140 PLOT 20,20:DRAWTO 30,20
150 DRAWTO 30,7:DRAWTO 34,7
160 DRAWTO 34,20:DRAWTO 55,20
170 DRAWTO 49,26
180 DRAWTO 26,26
190 DRAWTO 20,20
195 REM **DRAW MAN
200 PLOT 45,12:DRAWTO 42,10
210 DRAWTO 45,8:DRAWTO 48,10
```



```

220 DRAWTO 45,12:DRAWTO 45,20
230 DRAWTO 45,17:DRAWTO 49,13
240 DRAWTO 78,36

```

NOTES

Line 100: This program runs in graphics mode 5.

Line 110: This SETCOLOR statement assigns color and luminance to the graphic points.

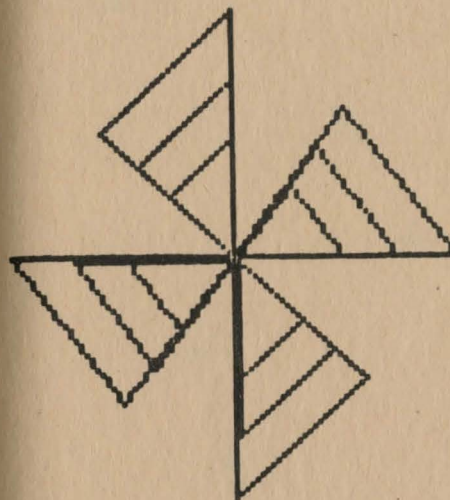
Line 120: Here SETCOLOR assigns color and luminance to the graphic screen's background.

Line 130: This SETCOLOR statement affects the text window background and characters.

WINDMILL

The computer draws the arms of a windmill.

Add to the program to make it a pinwheel or propellor.



PROGRAM LISTING

```

100 GRAPHICS 5
110 COLOR 1:SETCOLOR 0,4,14
120 SETCOLOR 4,8,10:SETCOLOR 2,3,14
130 PLOT 10,20:DRAWTO 50,20
140 DRAWTO 40,10:DRAWTO 20,30
150 DRAWTO 10,20
160 PLOT 30,5:DRAWTO 30,35
170 DRAWTO 42,30:DRAWTO 18,10
175 DRAWTO 30,5
180 PLOT 14,20:DRAWTO 22,28
190 PLOT 19,20:DRAWTO 25,25
200 PLOT 47,20:DRAWTO 39,12
210 PLOT 43,20:DRAWTO 35,14

```



```

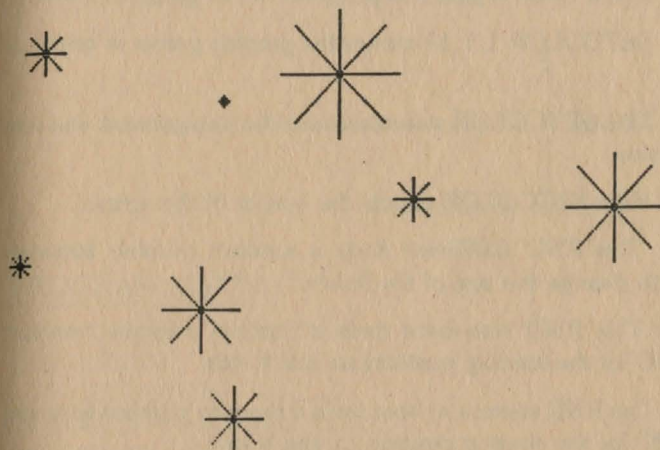
220 PLOT 30,9:DRAWTO 21,12
230 PLOT 30,12:DRAWTO 24,14
240 PLOT 30,33:DRAWTO 40,28
250 PLOT 30,30:DRAWTO 36,26

```

SNOWFLAKES

Fill the screen with different-sized snowflakes.

Change the program to print out even more random snowflakes on your screen.



PROGRAM LISTING

```

100 GRAPHICS 8
110 COLOR 1:SETCOLOR 1,1,14
112 SETCOLOR 2,8,2
114 SETCOLOR 4,4,14
115 FOR T=1 TO 8
120 L=INT(RND(1)*15+3)
130 X=INT(RND(1)*280+20)
140 Y=INT(RND(1)*120+20)
150 PLOT X-L,Y:DRAWTO X+L,Y
160 PLOT X,Y-L:DRAWTO X,Y+L
170 L=L-3
180 SX=X-L:SY=Y-L
200 PLOT SX,SY:DRAWTO SX+(2*L),SY+(2*L)

```



```

210 SY=Y+L
220 PLOT SX,SY:DRAWTO SX+(2*L),SY-(2*L)
230 NEXT T

```

NOTES

Line 100: GRAPHICS 8 puts this program run in graphics mode 8.

Line 110: SETCOLOR 1,1,14 makes the graphic points as bright as possible.

Line 112: This SETCOLOR statement sets the background and text window color.

Line 114: Here SETCOLOR affects the border of the screen.

Line 120: This RND statement finds a random number between 3 and 15 to change the size of the flakes.

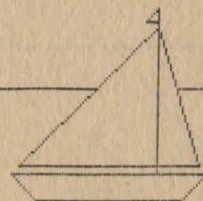
Line 130: This RND statement finds a random number between 20 and 280 for the starting position on the X axis.

Line 140: The RND statement here finds a random number between 20 and 120 for the starting position on the Y axis.

Line 150: PLOT is used to position the snowflakes. The X, Y coordinates are set to the numeric variables in Lines 130 and 140.

SAILBOAT

Draw a picture on your screen of a sailboat.
Can you add some sea gulls to the picture?



PROGRAM LISTING

```

100 GRAPHICS 5+16
102 COLOR 1:SETCOLOR 0,3,8
104 SETCOLOR 4,8,2
106 REM ***DRAW BOAT
110 PLOT 30,35:DRAWTO 65,35
115 DRAWTO 60,39
120 DRAWTO 35,39
125 DRAWTO 30,35
130 PLOT 53,35:DRAWTO 53,6
135 DRAWTO 50,9:DRAWTO 53,9
140 PLOT 53,10:DRAWTO 31,33
145 DRAWTO 64,33:DRAWTO 53,10
147 REM ***DRAW HORIZON
150 PLOT 0,20:DRAWTO 42,20
155 PLOT 58,20:DRAWTO 79,20
157 REM ***DRAW MOON
160 PLOT 9,2:DRAWTO 13,2
165 DRAWTO 15,3:DRAWTO 15,4
170 DRAWTO 13,6:DRAWTO 9,6
175 DRAWTO 7,5:DRAWTO 7,4:DRAWTO 9,3
180 GOTO 180

```

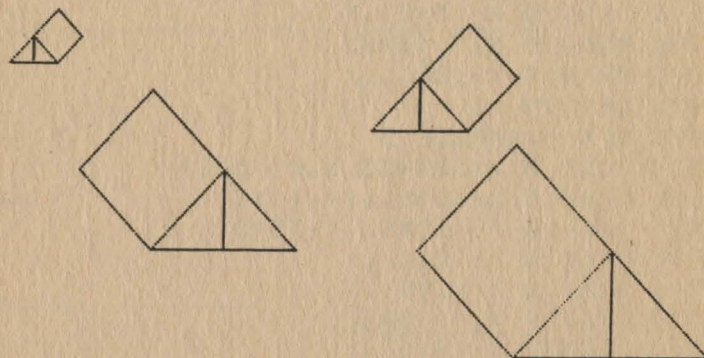

NOTE

Line 100: This program runs in the full screen of graphics mode 5. Since there is no text window, no SETCOLOR statement for color register 2 is needed.

CAMPGROUND

Here's a campground scene.

Try adding some figures or trees to the picture.



PROGRAM LISTING

```
100 GRAPHICS 7+16
110 SETCOLOR 1,9,14:COLOR 1
112 SETCOLOR 4,9,0
114 REM ***DRAW HORIZON
120 PLOT 0,20:DRAWTO 159,20
125 REM ***DRAW MOON
130 PLOT 125,3:DRAWTO 130,3
140 DRAWTO 134,5:DRAWTO 134,7
150 DRAWTO 130,10:DRAWTO 125,10
160 DRAWTO 121,7:DRAWTO 121,5:DRAWTO 125,3
165 X=25:Y=35:S=1
170 GOSUB 400
180 X=90:Y=45:S=2
190 GOSUB 400
```



```

200 X=55:Y=70:S=2.5
210 GOSUB 500
220 X=130:Y=90:S=2.75
230 GOSUB 500
240 GOTO 900
390 REM ***ANGLE 1
400 PLOT X,Y:DRAWTO X,Y-(5*S)
410 DRAWTO X+(8*S),Y-(11*S)
420 DRAWTO X+(15*S),Y-(5*S)
430 DRAWTO X+(8*S),Y
440 DRAWTO X-(5*S),Y
450 DRAWTO X,Y-(5*S)
460 DRAWTO X+(8*S),Y
470 RETURN
490 REM ***ANGLE 2
500 PLOT X,Y:DRAWTO X,Y-(5*S)
510 DRAWTO X-(8*S),Y-(11*S)
520 DRAWTO X-(15*S),Y-(5*S)
530 DRAWTO X-(8*S),Y
540 DRAWTO X+(5*S),Y
550 DRAWTO X,Y-(5*S)
560 DRAWTO X-(8*S),Y
570 RETURN
900 GOTO 900

```

NOTES

Line 100: This program runs in graphic mode 7's full screen.

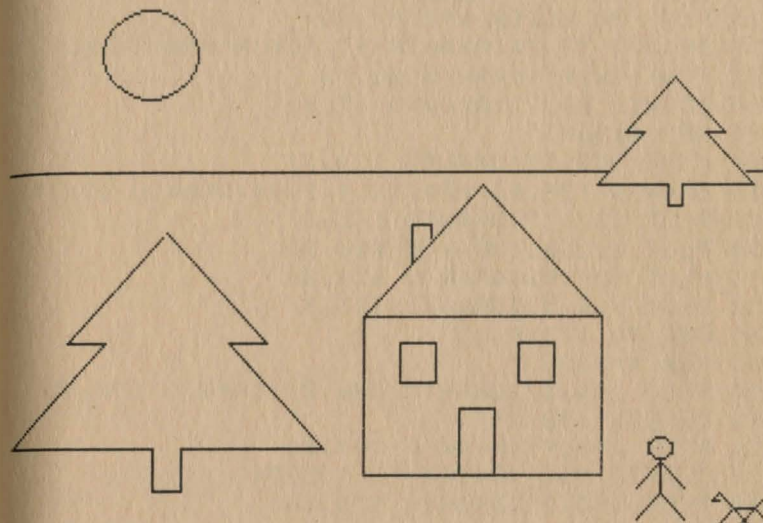
Lines 165, 180, 200, 220: X and Y are set as the starting coordinates of each tent. S is the size variable.

Lines 400-470, 500-570: These are subroutines, which draw the tents angled to face toward the left or the right.

PICTURE

Draw this picture with your computer.

What additions can you make to the scene?



PROGRAM LISTING

```

100 GRAPHICS 7+16
110 COLOR 1:SETCOLOR 1,0,0
112 SETCOLOR 4,8,2
115 REM **SET POSITION/SCALE OF TREES
120 X=120:Y=10:S=1
130 GOSUB 600
150 X=30:Y=50:S=1.75
160 GOSUB 600
200 REM **HORIZON
205 PLOT 0,25:DRAWTO 108,25
210 PLOT 133,25:DRAWTO 159,25
240 REM **HOUSE
250 PLOT 62,58:DRAWTO 107,58
260 DRAWTO 107,85:DRAWTO 62,85:DRAWTO 62,58

```



```

265 REM **WINDOWS & DOOR
270 DRAWTO 85,32:DRAWTO 107,58
280 PLOT 70,48:DRAWTO 70,42
290 DRAWTO 73,42:DRAWTO 73,45
300 PLOT 70,65:DRAWTO 77,65
310 DRAWTO 77,70:DRAWTO 70,70:DRAWTO 70,65
320 PLOT 90,65:DRAWTO 97,65
330 DRAWTO 97,70:DRAWTO 90,70:DRAWTO 90,65
350 PLOT 80,85:DRAWTO 80,73
360 DRAWTO 87,73:DRAWTO 87,85
390 REM **MAN
400 PLOT 120,77:DRAWTO 124,77
410 DRAWTO 124,80:DRAWTO 120,80:DRAWTO 120,77
420 PLOT 122,79:DRAWTO 122,87
430 PLOT 122,81:DRAWTO 118,86
440 PLOT 122,81:DRAWTO 126,86
450 PLOT 119,93:DRAWTO 122,87
460 DRAWTO 125,93
490 REM ***DOG
500 PLOT 140,87:DRAWTO 138,87:DRAWTO 140,85
510 DRAWTO 148,94
520 PLOT 140,94:DRAWTO 144,90
530 DRAWTO 150,90:DRAWTO 155,94
540 PLOT 145,94:DRAWTO 155,88
550 REM **SUN
560 PLOT 30,3:DRAWTO 34,3
565 X=34:Y=3
570 FOR I=1 TO 3:DRAWTO X+I,Y+I:NEXT I
572 DRAWTO 37,8
575 X1=37:Y1=8:FOR I=1 TO 3
576 DRAWTO X1-I,Y1+I:NEXT I
580 DRAWTO 30,11:X=30:Y=11
582 FOR I=1 TO 3:DRAWTO X-I,Y-I:NEXT I
585 DRAWTO 27,5
587 X=27:Y=5
590 FOR I=1 TO 3:DRAWTO X+I,Y-I
595 GOTO 900
600 REM ***TREES ROUTINE
605 PLOT X,Y:DRAWTO X-(10*S),Y+(10*S)
610 DRAWTO X-(5*S),Y+(10*S)
620 DRAWTO X-(16*S),Y+(18*S)
630 DRAWTO X-(2*S),Y+(18*S)

```

```

640 DRAWTO X-(2*S),Y+(22*S)
650 DRAWTO X+(2*S),Y+(22*S)
660 DRAWTO X+(2*S),Y+(18*S)
670 DRAWTO X+(16*S),Y+(18*S)
680 DRAWTO X+(5*S),Y+(10*S)
700 DRAWTO X+(10*S),Y+(10*S)
710 DRAWTO X,Y
720 RETURN
800 GOTO 900

```

5 ANIMATION

COLOR 1 and COLOR 0 statements make it easy to move objects around on the screen.

PROGRAMS

1. Warmups
2. Walking Figure
3. Flying Bird
4. Jumping Fish
5. Helicopter

WARMUPS

Moving a shape across the screen requires 3 steps:

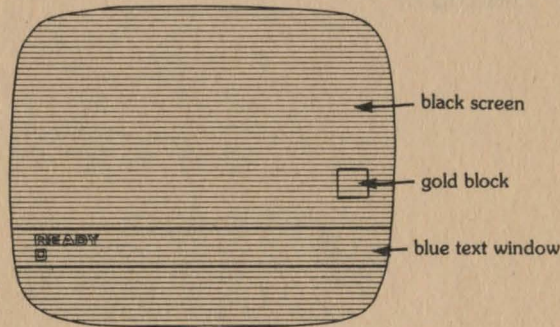
1. Draw the shape.
2. Erase the shape.
3. Draw it in a new location.

The faster the computer can do steps 1, 2, and 3, the faster the shape will move. If there is a delay between steps 2 and 3, the shape will flicker as it moves.

Any of the graphics statements can be used to draw the shape with COLOR 1. Then, to erase it, use the same statements, but set the COLOR statement to 0, the background color.

EXERCISE 1

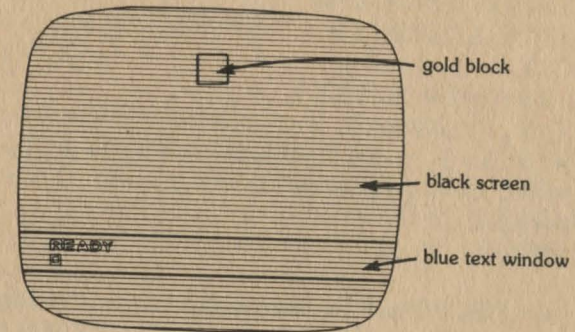
Use the PLOT statement to draw a block of color and then move it across the screen.



```
10 GRAPHICS 4
20 FOR X=1 TO 79
30 COLOR 1:PLOT X,30
40 FOR I=1 TO 50:NEXT I
50 COLOR 0:PLOT X,30
60 NEXT X
```

Take out the Wait Loop in Line 40 and see how fast the block moves across the screen.

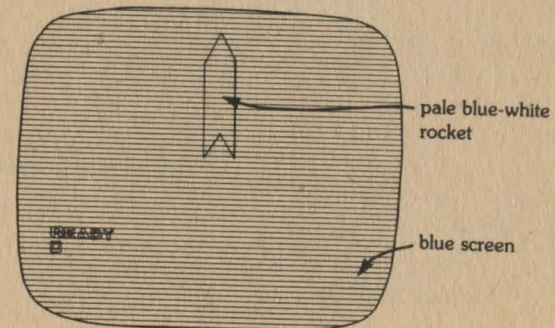
Change the program to move the block up the screen.



```
10 GRAPHICS 4
20 FOR Y=39 TO 1 STEP -1
30 COLOR 1:PLOT 40,Y
40 FOR I=1 TO 50:NEXT I
50 COLOR 0:PLOT 40,Y
60 NEXT Y
```

EXERCISE 2

Now draw a rocket in graphics mode 8 and make it blast up the screen




```

10 GRAPHICS 8:X=100
20 FOR Y=150 TO 40 STEP -10
30 COLOR 3:GOSUB 70
40 IF Y=100 THEN END
50 COLOR 0:GOSUB 70
60 NEXT Y
70 REM ***DRAW ROCKET
80 PLOT X,Y:DRAWTO X,Y-30
90 DRAWTO X+3,Y-35:DRAWTO X+6,Y-30
100 DRAWTO X+6,Y:DRAWTO X+3,Y-5
110 DRAWTO X,Y
120 RETURN

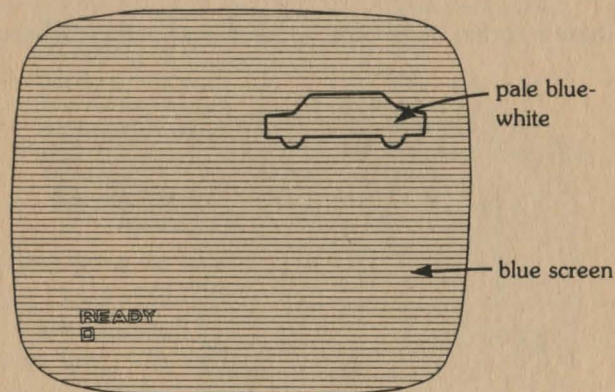
```

Line 30 sets COLOR to 1 and then calls the subroutine that draws the rocket. Line 50 erases the rocket by setting COLOR to 0, or the background color, before calling the same subroutine. Repeating these steps moves the rocket up the screen.

Add a Wait Loop between Lines 40 and 50 and see if you like the result.

EXERCISE 3

This exercise moves a car across the screen.



```

10 GRAPHICS 8
15 Y=100
20 FOR X=230 TO 25 STEP -15
30 COLOR 1:GOSUB 80
40 COLOR 0:GOSUB 80
50 NEXT X
60 END
70 REM **DRAW CAR
80 PLOT X,Y:DRAWTO X+20,Y
90 DRAWTO X+27,Y+8:DRAWTO X+38,Y+8
100 DRAWTO X+38,Y+15:DRAWTO X-22,Y+15
110 DRAWTO X-22,Y+8:DRAWTO X-10,Y+8
120 DRAWTO X,Y
130 PLOT X-12,Y+16:DRAWTO X-12,Y+18
140 PLOT X-11,Y+19:DRAWTO X-7,Y+19
150 PLOT X-6,Y+18:DRAWTO X-6,Y+16
160 PLOT X+18,Y+16:DRAWTO X+18,Y+18
170 PLOT X+19,Y+19:DRAWTO X+23,Y+19
180 PLOT X+24,Y+18:DRAWTO X+24,Y+16
190 RETURN

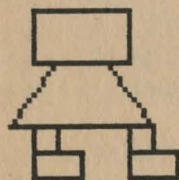
```


WALKING FIGURE

Make a simple cartoon character walk across the screen.

To make your own figure, just draw your character in two or more walking positions and then alternate the positions as you move your figure across the screen.

Try making an animal walk or run across the screen.



PROGRAM LISTING

```
1 GRAPHICS 0
100 GRAPHICS 8
110 COLOR 1:SETCOLOR 1,1,0
112 SETCOLOR 2,4,2
115 SETCOLOR 2,4,10
120 X=30:Y=50
150 FOR X=30 TO 250 STEP 15
160 COLOR 1:GOSUB 210
165 COLOR 0:GOSUB 210
180 COLOR 1:GOSUB 310
185 COLOR 0:GOSUB 310
190 NEXT X
200 STOP
205 REM ***DRAW POSITION 1
210 PLOT X-5,Y+8:DRAWTO X-10,Y+8
215 DRAWTO X-10,Y-2:DRAWTO X+10,Y-2
217 DRAWTO X+10,Y+8:DRAWTO X-5,Y+8
220 PLOT X-5,Y+8:DRAWTO X-15,Y+20
230 DRAWTO X+15,Y+20
240 DRAWTO X+5,Y+8
```

```
250 PLOT X-5,Y+20:DRAWTO X+5,Y+20
255 DRAWTO X+5,Y+25:DRAWTO X-5,Y+25
257 DRAWTO X-5,Y+20
260 PLOT X,Y+20:DRAWTO X+5,Y+20
265 DRAWTO X+5,Y+25:DRAWTO X,Y+25
267 DRAWTO X,Y+20
270 PLOT X-5,Y+25:DRAWTO X+10,Y+25
280 DRAWTO X+10,Y+30:DRAWTO X-5,Y+30
290 DRAWTO X-5,Y+25
300 RETURN
305 REM ***DRAW POSITION 2
310 PLOT X-5,Y+8:DRAWTO X-10,Y+8
315 DRAWTO X-10,Y-2:DRAWTO X+10,Y-2
317 DRAWTO X+10,Y+8:DRAWTO X-5,Y+8
320 PLOT X-5,Y+8:DRAWTO X-15,Y+20
330 DRAWTO X+15,Y+20
340 DRAWTO X+5,Y+8
350 PLOT X-5,Y+20:DRAWTO X-10,Y+20
355 DRAWTO X-10,Y+25:DRAWTO X-5,Y+25
357 DRAWTO X-5,Y+20
360 PLOT X-10,Y+25:DRAWTO X,Y+25
365 DRAWTO X,Y+30:DRAWTO X-10,Y+30
367 DRAWTO X-10,Y+25
370 PLOT X+10,Y+20:DRAWTO X+15,Y+20
375 DRAWTO X+15,Y+25:DRAWTO X+10,Y+25
377 DRAWTO X+10,Y+20
380 PLOT X+10,Y+25:DRAWTO X+20,Y+25
390 DRAWTO X+20,Y+30:DRAWTO X+10,Y+30
400 DRAWTO X+10,Y+25
410 RETURN
```

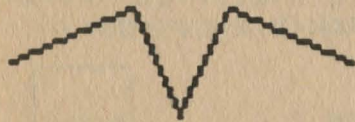
NOTE

Lines 160-185: The drawings of the figure are placed in subroutines starting on Lines 210 and 310. This helps the programmer separate the action part of the program from the initial drawings. First each position is drawn with COLOR 1, making it visible. Then it is drawn with COLOR 0, and disappears into the background color.

FLYING BIRD

Make a bird fly across the screen. This program uses three positions of the bird's wings in order to simulate motion.

Try making a bird or a butterfly of your own.



PROGRAM LISTING

```
100 GRAPHICS 8
110 COLOR 1:SETCOLOR 1,1,0
115 SETCOLOR 2,8,14
120 Y=40
130 REM ***MOVE ACROSS
140 FOR X=15 TO 200 STEP 15
150 COLOR 1:PLOT X,Y:GOSUB 195
155 COLOR 0:PLOT X,Y:GOSUB 195
160 COLOR 1:PLOT X-5,Y-5:GOSUB 250
165 COLOR 0:PLOT X-5,Y-5:GOSUB 250
170 COLOR 1:PLOT X-5,Y-10:GOSUB 310
175 COLOR 0:PLOT X-5,Y-10:GOSUB 310
180 NEXT X
190 STOP
195 REM ***POSITION 1
200 DRAWTO X+25,Y-10
210 DRAWTO X+35,Y+10
220 DRAWTO X+45,Y-10
230 DRAWTO X+70,Y
240 RETURN
250 REM ***POSITION 2
260 DRAWTO X+25,Y-10
270 DRAWTO X+30,Y+10
280 DRAWTO X+45,Y-5
```

```
290 DRAWTO X+70,Y-5
300 RETURN
310 REM ***POSITION 3
320 DRAWTO X+35,Y+10
330 DRAWTO X+70,Y-10
340 RETURN
```

NOTE

Lines 150-170: Each pair of GOSUB statements displays a different position of the bird's wings.

JUMPING FISH

This fish swims across the screen and also jumps out of the water until you hit the BREAK key.

Change the speed he swims by changing the step increments on Lines 160 and 300.



PROGRAM LISTING

```
100 GRAPHICS 8
110 COLOR 1:SETCOLOR 1,1,1
120 SETCOLOR 2,8,12
130 X=65:Y=140
145 REM ***DRAW WATER
150 COLOR 1:PLOT 0,80:DRAWTO 319,80
155 REM ***MOVE ACROSS
160 FOR X=65 TO 165 STEP 10
170 COLOR 1:PLOT X,Y:GOSUB 350
180 COLOR 0:PLOT X,Y:GOSUB 350
190 NEXT X
195 REM ***JUMP UP
200 FOR Y=140 TO 40 STEP -5
210 COLOR 1:PLOT X,Y:GOSUB 350
220 COLOR 0:PLOT X,Y:GOSUB 350
230 NEXT Y
235 REM ***JUMP DOWN
240 FOR Y=40 TO 140 STEP 5
250 COLOR 1:PLOT X,Y:GOSUB 350
260 COLOR 0:PLOT X,Y:GOSUB 350
270 NEXT Y
```

```
275 REM ***REDRAW FISH & WATER
280 COLOR 1:PLOT X,Y:GOSUB 350
290 COLOR 1:PLOT 0,80:DRAWTO 319,80
295 REM ***MOVE ACROSS
300 FOR X=175 TO 265 STEP 10
310 COLOR 1:PLOT X,Y:GOSUB 350
320 COLOR 0:PLOT X,Y:GOSUB 350
330 NEXT X
340 GOTO 145
350 REM ***DRAW FISH
360 DRAWTO X+31,Y-10:DRAWTO X+45,Y
370 DRAWTO X+31,Y+8:DRAWTO X-3,Y-5
380 DRAWTO X,Y:PLOT X+38,Y-2
390 RETURN
```

NOTES

Lines 155-190: Move the fish across the screen.

Lines 195-230: Make the fish jump up.

Lines 235-270: Make the fish jump down.

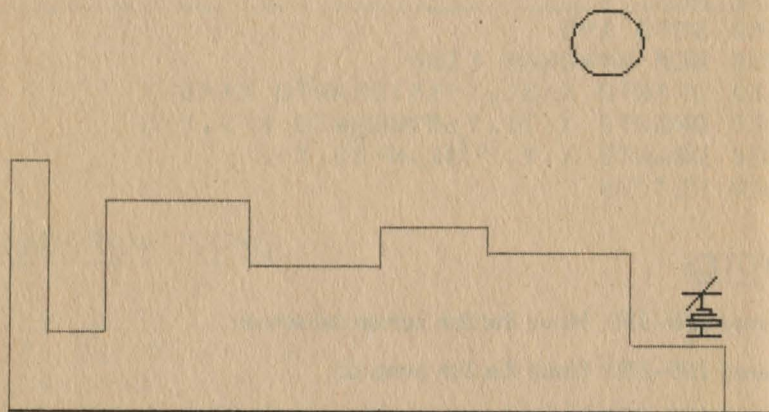
Lines 295-330: Move the fish the rest of the way across the screen.

Lines 350-390: Subroutine that draws the fish.

HELICOPTER

Watch your computer draw a helicopter flying above a city skyline and landing on a building in this program.

Once you get this program up and running, try adding some windows to the buildings.



PROGRAM LISTING

```
100 GRAPHICS 8:COLOR 1:SETCOLOR 1,1,14
115 SETCOLOR 2,9,0
117 REM ***DRAW SKYLINE
120 PLOT 0,150:DRAWTO 319,150
130 PLOT 0,55:DRAWTO 16,55
135 DRAWTO 16,120:DRAWTO 40,120
150 DRAWTO 40,70:DRAWTO 100,70
155 DRAWTO 100,95:DRAWTO 155,95
160 DRAWTO 155,80:DRAWTO 200,80
170 DRAWTO 200,90:DRAWTO 260,90
180 DRAWTO 260,120:DRAWTO 300,120
190 DRAWTO 300,150
200 REM ***MOON
```

```
210 PLOT 250,10
220 DRAWTO 260,10
230 DRAWTO 264,13
240 DRAWTO 266,16
250 DRAWTO 268,19
260 DRAWTO 268,21
270 DRAWTO 266,24
275 DRAWTO 264,27
280 DRAWTO 260,27
285 DRAWTO 250,27
290 DRAWTO 246,24
291 DRAWTO 244,21
292 DRAWTO 242,19
293 DRAWTO 242,16
294 DRAWTO 244,13
295 DRAWTO 246,11
296 DRAWTO 250,10
298 REM ***MOVE ACROSS
300 FOR X=20 TO 260 STEP 20
305 Y=30
310 COLOR 1:GOSUB 400
320 COLOR 0:GOSUB 400
325 NEXT X
327 REM ***MOVE DOWN
330 FOR Y=40 TO 90 STEP 10
335 X=280
340 COLOR 1:GOSUB 400
350 COLOR 0:GOSUB 400
360 NEXT Y
370 COLOR 1:GOSUB 400
380 STOP
390 REM ***DRAW COPTER
400 PLOT X,Y
410 DRAWTO X+6,Y-4
411 PLOT X,Y:DRAWTO X+8,Y
412 PLOT X,Y:DRAWTO X-8,Y
413 PLOT X,Y:DRAWTO X-6,Y+4
```



```
415 PLOT X,Y:DRAWTO X,Y+6
416 DRAWTO X-3,Y+6:DRAWTO X+6,Y+6
417 DRAWTO X+6,Y+8:DRAWTO X+8,Y+8
418 DRAWTO X-6,Y+8:DRAWTO X-6,Y+10
419 DRAWTO X-10,Y+10:DRAWTO X+10,Y+10
420 PLOT X,Y+10:DRAWTO X,Y+16
421 DRAWTO X-8,Y+16:DRAWTO X+8,Y+16
425 RETURN
```

6 FUN WITH GEOMETRY

You can use the built-in functions SIN and COS to draw many geometric shapes, such as a square, triangle, or circle. You can even make flowers and fancy designs.

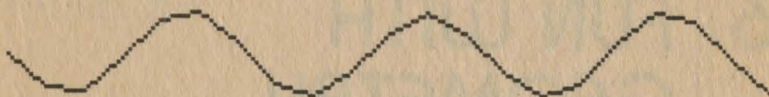
PROGRAMS

1. Warmups
2. Apple Tree Harvest
3. Leaping Fish
4. Bird and Flower
5. Fancy Graphics

WARMUPS

EXERCISE 1

Use the SIN function to draw a wave across the top of your screen.



```
100 GRAPHICS 8:COLOR 1
110 H=10: D=20: C=1: A=0: B=20
120 PLOT A,D
130 FOR X=A TO B STEP C
140 Y=SIN(X)*H+D
150 DRAWTO X*H,Y
160 NEXT X
```

The numeric variables on Line 110 are used as follows:

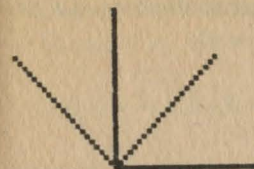
- H: height of wave
- D: location on Y axis
- A: beginning point on X axis
- B: last point on X axis
- C: size of step along X axis

Experiment with different values for these variables and see how the wave changes.

Replace the SIN function with the COS function on Line 140 and see how the wave changes.

EXERCISE 2

We can use the SIN and COS functions to draw any angle we want. This example draws 0 degrees, 45 degrees, 90 degrees, and 135 degrees.



```
100 GRAPHICS 8:COLOR 1
110 PRINT"INPUT LENGTH OF LINE,ANGLE"
120 INPUT R,A
130 AR=A*(3.14159/180)
140 X1=R*COS(AR)+100
150 Y1=-R*SIN(AR)+100
160 PLOT 100,100:DRAWTO X1,Y1
170 GOTO 120
```

Experiment with different values of R and A and see what angles you make.

Line 130 converts the angle A from degrees to radians, using the equation:

$$\text{Radians} = \text{degrees} * (\text{PI}/180) \text{ for PI} = 3.14159$$

Lines 140 and 150 determine the ending X,Y coordinate for the line and add these values to the point 100,100. This translates the 0,0 point from the top left point on the screen to 100,100.

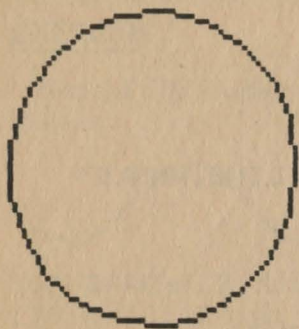
Line 150 uses a negative R because the Y coordinate on your screen is the reverse of the Y coordinate on a normal X,Y axis. The Y coordinate on your screen increases as you move down rather than up. Therefore, you must change the sign of Y in order to make it the same as a normal X,Y axis.

EXERCISE 3

Draw a circle using the SIN and COS functions. These equations:

$$\begin{aligned} X &= R * \text{COS}(I) + X_C \\ Y &= -R * \text{SIN}(I) + Y_C \end{aligned}$$

give the X,Y coordinates for any point on the circumference of a circle with a radius of R and center at XC,YC.



```
100 GRAPHICS 8:COLOR 1
110 XC=100: YC=75: R=30:
115 S=0:E=6.6:J=.3
120 PLOT XC+R,YC
130 FOR I=S TO E STEP J
140 X=R*COS(I)+XC
150 Y=-R*SIN(I)+YC
160 DRAWTO X,Y
170 NEXT I
```

The numeric variables on Line 110 are used as follows:

XC: center of circle X-axis
 YC: center of circle Y-axis
 R: radius of circle
 S: radians to begin drawing
 E: radians to end drawing
 J: increment to step while drawing

S and E determine which part of the circle is drawn. To make a complete circle, set S=0 and E=6.6. To draw a part of the circle, change S or E to any number between 0 and 6.6. Experiment with different combinations for S and E and see which part of the circle you make. Just be sure to make E larger than S

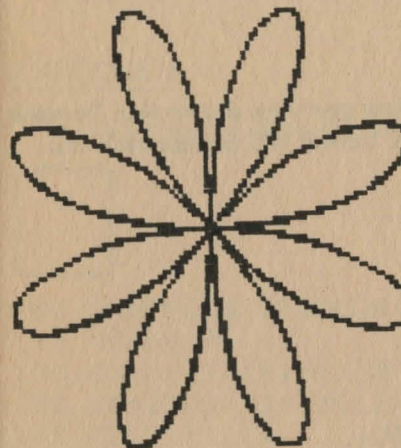
Change the value of J and see how it affects your circle.
 To make an ellipse, make R different for X than Y.

$$150 Y = -1.5 * R * \cos(I) + YC$$

Vary the shape of the ellipse by changing the radius R for X and Y.

EXERCISE 4

Draw a flower using the SIN and COS functions.



```
100 GRAPHICS 8:COLOR 1
110 N=4: A=50: PI=3.14159
120 M=N*PI*2.5
130 PLOT 100,100
140 FOR I=0 TO M STEP .05
150 R=A*SIN(N*I)
160 X=R*COS(I)+100
170 Y=-R*SIN(I)+100
180 DRAWTO X,Y
190 NEXT I
```


The numeric variables on Line 110 are used as follows:

A: size of the flower

N: number of petals drawn

If N is even, flower has $2 * N$ petals

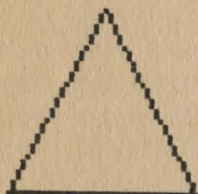
If N is odd, flower has N petals

Change the value of N and see what your flower looks like. Try using a decimal number for N and see what you get.

Line 150 uses an equation called "N-leaved rose" to find a value for R. Notice that R changes each time the FOR . . . NEXT loop executes.

EXERCISE 5

We can use these equations to draw geometric shapes also. To make a triangle, we need to draw 3 sides, turning 120 degrees each time.



```
100 GRAPHICS 8:COLOR 1
110 A=120*(3.14159/180)
120 R=40: AR=A
130 X2=R*COS(AR)+100
140 Y2=-R*SIN(AR)+100
145 REM**DRAW SIDE 1
150 PLOT 100,100:DRAWTO X2,Y2
160 AR=AR+A: X1=X2: Y1=Y2
170 X2=R*COS(AR)+X1
180 Y2=-R*SIN(AR)+Y1
185 REM**DRAW SIDE 2
190 PLOT X1,Y1:DRAWTO X2,Y2
195 REM**DRAW SIDE 3
200 PLOT X2,Y2:DRAWTO 100,100
```

The numeric variables are used as follows:

A: angle in degrees

AR: angle in radians

R: length of side

X1,Y1: X,Y coordinates for start of line

X2,Y2: X,Y coordinates for end of line.

Lines 130,140 and 170,180 add the new values of X,Y to the starting coordinates X1,Y1. This moves the origin each time to the starting point of the line.

EXERCISE 6

To draw a square, draw 4 sides and turn 90 degrees.



```
100 GRAPHICS 8:COLOR 1
110 A=360/4
120 AR=A*(3.14159/180)
130 X1=150: Y1=150: R=20
140 PLOT X1,Y1
150 FOR I=0 TO 3
160 X2=R*COS(AR*I)+X1
170 Y2=-R*SIN(AR*I)+Y1
180 DRAWTO X2,Y2
190 X1=X2: Y1=Y2
200 NEXT I
```

This program is a more efficient version of Exercise 5. The sides are drawn inside the FOR . . . NEXT loop on Lines 150-200.

EXERCISE 7

Now you can draw a N-sided shape just by changing Exercise 6 as follows:

```
105 INPUT N
110 A=360/N
150 FOR I=0 TO N-1
```

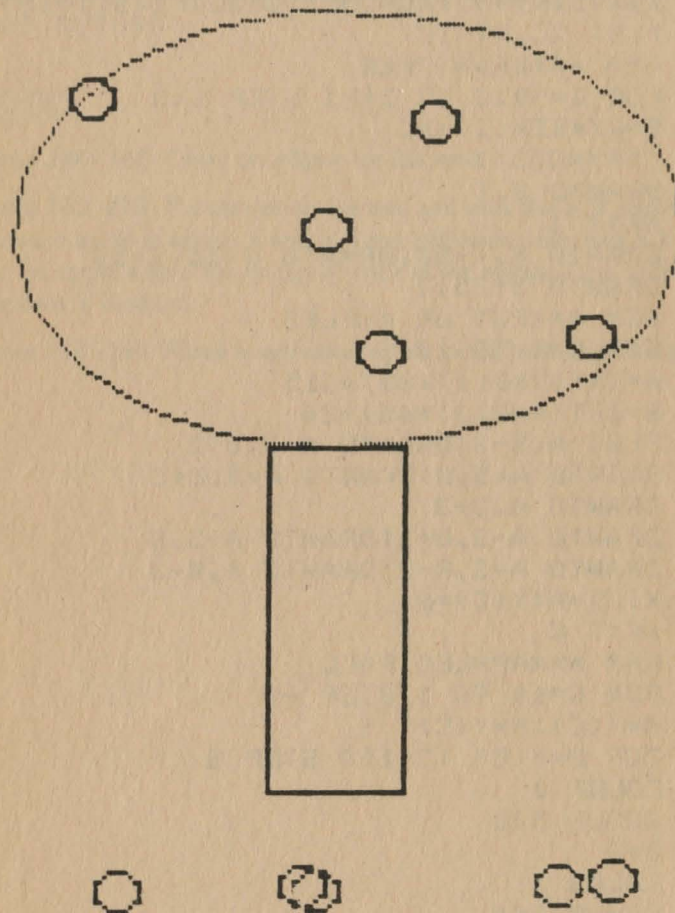
Try different values of N and see what shapes you make. How large must N be before your shape looks like a circle?

Now try using these ideas in some programs.

APPLE TREE HARVEST

Use the equations for an ellipse to draw an apple tree. This tree starts out with 10 apples, but some fall off.

Create your own picture by adding other figures to this one.



PROGRAM LISTING

```
100 GRAPHICS 8
102 COLOR 1:SETCOLOR 1,1,14
104 SETCOLOR 4,8,0
106 SETCOLOR 2,6,2
110 DIM X(10),Y(10)
120 RX=75:RY=45:XC=152:YC=55:PI=3.14159
130 PLOT XC,RY+YC
135 REM ***DRAW TREE
140 FOR I=-0.3 TO 2*PI STEP 0.3
150 S=RX*SIN(I)+XC
155 T=RY*COS(I)+YC
160 DRAWTO S,T
165 NEXT I
167 DRAWTO S,T+50:DRAWTO S+25,T+50
169 DRAWTO S+25,T
170 REM ***PUT ON APPLES
180 FOR C=1 TO 10
190 A=INT(RND(1)*85)+115
195 B=INT(RND(1)*65)+20
200 PLOT A,B-3:DRAWTO A+3,B-2
202 DRAWTO A+5,B:DRAWTO A+3,B+2
204 DRAWTO A,B+3
206 DRAWTO A-3,B+1:DRAWTO A-5,B
208 DRAWTO A-3,B-2:DRAWTO A,B-3
209 X(C)=A:Y(C)=B
210 NEXT C
212 REM ***APPLES FALL
215 FOR C=10 TO 1 STEP -2
216 A=X(C):B=Y(C)
220 FOR D=Y(C) TO 150 STEP 3
230 COLOR 0
240 GOSUB 300
241 B=D
245 COLOR 1
250 GOSUB 300
255 NEXT D
```

```
260 NEXT C
275 STOP
300 PLOT A,B-3:DRAWTO A+3,B-2
302 DRAWTO A+5,B:DRAWTO A+3,B+2
304 DRAWTO A,B+3
306 DRAWTO A-3,B+1:DRAWTO A-5,B
308 DRAWTO A-3,B-2:DRAWTO A,B-3
309 RETURN
```

NOTES

Lines 140-165: Draw an ellipse for the tree.

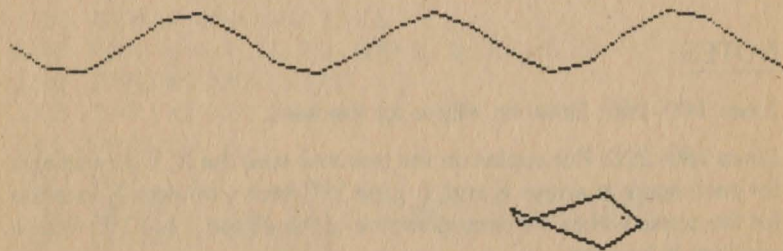
Lines 180-210: Put apples on the tree and save the X,Y coordinates for each apple in arrays X and Y. Line 190 finds a random X location for the apple within the circumference of the ellipse. Line 195 finds a random Y location.

Lines 215-260: Move every other apple to the bottom of the screen.

LEAPING FISH

Make our fish program from Chapter 5 more realistic by changing the line to a wave and making the fish jump in an arc instead of straight up.

Add some other fish or sea life and create your own underwater picture.



PROGRAM LISTING

```
100 GRAPHICS 8:COLOR 1:SETCOLOR 1,1,14
115 SETCOLOR 2,9,1
117 SETCOLOR 4,9,1
120 GOSUB 320
140 PI=3.14159:R=50
150 Y=80:YC=80
155 REM ***MOVE ACROSS
160 FOR X=10 TO 100 STEP 10
170 COLOR 1:GOSUB 390
175 COLOR 0:GOSUB 390
180 NEXT X
190 XC=X+50
195 REM ***JUMP UP
200 FOR I=3 TO 2 STEP -0.3
210 X=R*COS(I)+XC
220 Y=-R*SIN(I)*1.5+YC
230 COLOR 1:GOSUB 390
235 COLOR 0:GOSUB 390
```

```
240 NEXT I
245 Y=80
247 REM ***MOVE ACROSS
270 FOR X=160 TO 200 STEP 10
280 COLOR 1:GOSUB 390
285 COLOR 0:GOSUB 390
290 NEXT X
310 STOP
320 REM ***DRAW WAVE
330 PLOT 1,50
340 FOR X=1 TO 20 STEP 1
350 Y=SIN(X)*10+50
360 DRAWTO X*14,Y
370 NEXT X
380 RETURN
385 REM ***DRAW FISH
390 PLOT X,Y
400 DRAWTO X+31,Y-10:DRAWTO X+45,Y
405 DRAWTO X+31,Y+8:DRAWTO X-3,Y-5
408 DRAWTO X,Y
410 PLOT X+38,Y-2
420 RETURN
```

NOTES

Lines 200-260: Use the equations for a circle to make the fish jump in an arc.

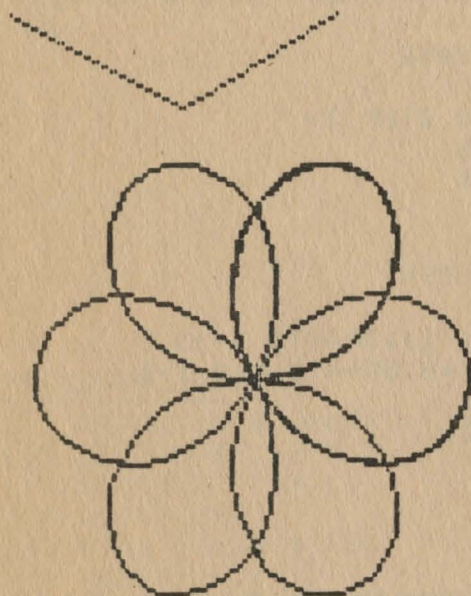
Lines 330-370: Use the SIN function to draw a wave across the screen.

Lines 390-420: Draw the fish.

BIRD AND FLOWER

Make the bird from Chapter 5 fly to a flower.

Change the value of N on Line 130 and see how the flower changes.



PROGRAM LISTING

```
100 GRAPHICS 8+16
110 COLOR 1:SETCOLOR 1,1,4
115 SETCOLOR 2,15,14
117 SETCOLOR 4,8,8
120 X=120:Y=120:PI=3.14159
130 N=1.5
140 REM ***DRAW FLOWER
145 PLOT X,Y
```

```
150 FOR I=0 TO 5*PI STEP 0.1
160 R=50*SIN(N*I)
170 C=R*COS(I)+X
180 D=-R*SIN(I)+Y
190 DRAWTO C,D
200 NEXT I
210 REM ***MOVE BIRD
220 B=30
230 FOR X=10 TO 90 STEP 4
240 Y=(6/15)*X+B
250 COLOR 1:GOSUB 300
255 COLOR 0:GOSUB 300
260 COLOR 1:GOSUB 360
265 COLOR 0:GOSUB 360
270 COLOR 1:GOSUB 420
275 COLOR 0:GOSUB 420
280 NEXT X
285 COLOR 1:GOSUB 420
290 GOTO 290
300 REM ***POSITION 1
305 PLOT X,Y
310 DRAWTO X+25,Y-10
320 DRAWTO X+35,Y+10
330 DRAWTO X+45,Y-10
340 DRAWTO X+60,Y
350 RETURN
360 REM ***POSITION 2
365 PLOT X-5,Y-5
370 DRAWTO X+25,Y-5
380 DRAWTO X+35,Y+10
390 DRAWTO X+70,Y-5
400 RETURN
420 REM ***POSITION 3
425 PLOT X-5,Y-10
430 DRAWTO X+35,Y+10
440 DRAWTO X+70,Y-10
450 RETURN
```


NOTE

Lines 230-280: Move the bird in a straight line from the point 10,30 to the top of the flower. Line 240 calculates the Y coordinate using the equation of a straight line as follows:

$$Y = M * X + B$$

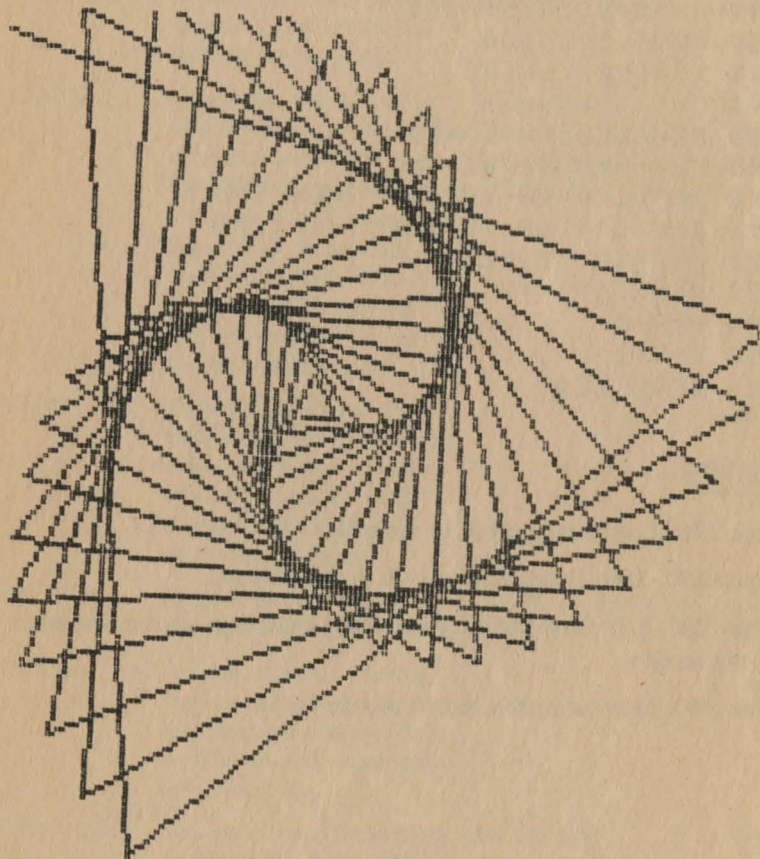
where M = slope of line, B = Y intercept.

In this case, $M = 6/15$ and $B = 30$.

FANCY GRAPHICS

Make your own spirals and fancy designs. Input 7, 123, 3 when this program runs to get this design.

Experiment with different input values and see what fancy designs you can make.



PROGRAM LISTING

```

100 GRAPHICS 8:SETCOLOR 1,1,14:COLOR 1
105 SETCOLOR 2,6,0
110 PRINT "INPUT SIDE, ANGLE, INCREMENT"
114 SETCOLOR 2,6,0
120 INPUT R,A,D
130 N=360/A
140 A=A*(3.14159/180)
150 PLOT 100,100
160 X1=100:Y1=100
170 I=0
180 X2=R*COS(A*I)+X1
190 Y2=-R*SIN(A*I)+Y1
200 IF X2<0 OR X2>319 THEN END
210 IF Y2<0 OR Y2>159 THEN END
220 DRAWTO X2,Y2
230 X1=X2:Y1=Y2
240 R=R+D
250 I=I+1
260 GOTO 180

```

NOTES

Line 140: Converts angle A to radians.

Lines 180, 190: Determine next X,Y coordinates.

Lines 200, 210: Stop the program when the design reaches the limits of the screen.

Line 240: Increments the length of side by D.

GLOSSARY

Term	Definition	Example	Page First Used
CHR\$(65)	Returns the character corresponding to the ATASCII code in the argument.	CHR\$ (65) = A	p. 34
COLOR 1 (C.)	Determines the data to be stored in display memory for all subsequent PLOT and DRAWTO statements until the next COLOR statement is executed. In graphics modes 0 through 2, the data is an ATASCII character. In graphics modes 3 through 8, the data enables color.	In Graphics 2, Color 36 = \$ In Graphics 3 COLOR 1 = gold	p. 56 p. 64
Cursor	Marker or symbol on the screen that marks where the user's next action will take effect or where the next character typed from the keyboard will appear.		p. 26
COS	Returns the trigonometric cosine of the argument.	COS(5) = .2836624	p. 108
Default	A value, action, or setting that is automatically used by the computer system when no other explicit information has been given.		p. 12
DIM A\$ (20) (DI.)	Sets aside space for a string variable (here called A\$), or for an array.	DIM A\$ (20)	p. 37

DRAWTO 25, 17 (DR.)	Causes a line to be drawn from the last point displayed by a PLOT statement, to the location specified by the DRAWTO statement.	DR. 220,100	p. 41
END	Terminates the execution of the program and returns control to the user.	END	p. 50
FOR I = 1 TO 4 (F.)	Allows you to repeat lines between the FOR and the NEXT statement four times.	FOR I = 1 TO 4	p. 33
NEXT I (N.)		NEXT I	
STEP	The loop of lines is incremented by the number in the STEP statement. If there is no STEP statement, the loop is incremented by one's.	FOR I = 1 TO 4 STEP -1 NEXT I	
GOSUB 250 (GOS.)	Executes the subroutine which begins at line 250.	GOSUB 250	p. 31
GOTO 200 (G.)	Branches to line 200.	GOTO 200	p. 31
GRAPHICS (GR.)	This command is used to select one of the nine Atari graphics modes.	GRAPHICS 3	p. 12
GRAPHICS 0	This command is also used as a clear screen command, either in Direct or Deferred mode. It terminates any previously selected graphics mode and returns the screen to the default mode (GR.0).	GRAPHICS 0	p. 12

IF. . . THEN	Executes or skips one or more statements, depending on the truth of the stated condition. In this example IF A = 10, THEN line 200 is executed next. Otherwise the line after the IF. . . THEN statement is executed.	IF A = 10 THEN 200	p. 38
INPUT A (I.)	Reads a line of input from the current input device.	PRINT "ENTER NUMBER" INPUT A	Ap. 38
INT (9.5)	Returns the largest whole number less than or equal to the argument.	INT (9.5) = 9	p. 37
PLOT (PL.)	Plots a single block at the specified location on the screen.	PLOT 10,20	p. 41
POSITION (POS.)	Places the invisible graphics window cursor at a specified location on the screen. The cursor does not actually move until an input or output command which uses the screen is issued.	POSITION 20,30	p. 26
PRINT A (PR. or ?)	Writes a line of output to the current display device. This example writes the value of A\$, B and C.	PRINT A\$,B,C	p. 12
REM	Includes remarks in the body of the program for the benefit of the user.	REM This is a remark	p. 78
RETURN (RET.)	Returns control from a subroutine to the statement following the GOSUB that called the subroutine	RETURN	p. 31

RND	Returns a random number between 0 and 1.	RND (1)	p. 37
SETCOLOR (SE.)	Chooses the particular hue and luminance to be stored in the specified color register.	SETCOLOR 1,1,114	p. 41 p. 59 p. 64
SIN	Returns the trigonometric sine of the argument.	SIN (°) = .90929	p. 108

COMPUTER CREATIONS WITH YOUR



Once you've explored the many capabilities of your Atari computer, you'll definitely want to take the next step to graphics. This book covers the beginning and intermediate graphics programming statements available in Atari BASIC. By following the exercises, you'll be able to decorate a Christmas tree, build a pyramid, play hide and seek, and create an amazing amount of complex and beautiful artwork. And after you've familiarized yourself with these programs, you can experiment and generate your own new and unique designs.

BASIC FUN WITH GRAPHICS

includes an introduction, an explanation on how to use this book, and warm up exercises at the beginning of each chapter. It also has a chapter that focuses on the color capabilities of your computer.

BASIC FUN WITH GRAPHICS

is written in BASIC for the Atari 400 or 800 Home Computer equipped with a BASIC cartridge. The programs will run on either a color or monochrome monitor.

The possibilities, discoveries, and fun are limitless. All you have to do is sit down at your computer and start creating!

AVON  CAMELOT

PUBLISHED BY AVON BOOKS

012-UP

